UNIVERSITE BRETAGNE LOIRE / MATHSTIC

ENSTA Bretagne

THESE DE DOCTORAT DE

L'ECOLE NATIONALE SUPERIEURE
DE TECHNIQUES AVANCEES BRETAGNE
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Automatique, Productique, Robotique*

Par

Thibaut NICO

Etude et développement de solutions de relocalisation d'objets sous-marins par des véhicules sous-marins hétérogènes

Thèse présentée et soutenue à Brest, le 13 juin 2019

Unité de recherche : Lab-STICC CNRS UMR 6285

Rapporteurs avant soutenance :

Vincent HUGEL	Professeur, Université de Toulon
Lionel LAPIERRE	Maître de conférence, LIRMM

Composition du Jury :

Examinateurs :	
Magali BARBIER	Ingénieure docteur, ONERA
Isabelle QUEINNEC	Directrice de recherche, CNRS
Gilles TROMBETTONI	Professeur, LIRMM
Co-directeurs de thèse :	
Benoit ZERR	Professeur, ENSTA-Bretagne
Luc Jaulin	Professeur, ENSTA-Bretagne
Encadrant industriel	
Hervé OTT	Ingénieur, ECA GROUP
Invité	
Benoît DESROCHERS	Ingénieur docteur, DGA-TN

Acknowlegdments

Firslty, I would like to thank ECA Group for their financial support through this industry-oriented doctoral thesis, named CIFRE (Conventions Industrielles de Formation par la Rercherche), in collaboration with ENSTA Bretagne (Ecole Nationale Supérieure des Techniques Avancées Bretagne, Lab-STICC UMR CNRS 6285).

On the ECA Group side, this thesis has been managed by Hervé OTT, head of engineering department at ECA Robotics in Toulon, subsidiairies of ECA Group, and Sébastien TAUVRY, marine data processing product manager at ECA Robotics in Brest. I would like to thank them for their guidance, time and advices along the three years, and their industrial point of view.

On the ENSTA Bretagne side, the thesis has been supervised by Benoit ZERR, professor at ENSTA Bretagne specialized in underwater acoustic imagery, and Luc JAULIN, professor at ENSTA Bretagne specialized in robotics. I am also grateful to their support, advices along the academic aspect of the thesis, time to read the articles and this long thesis. I would especially thank Luc for his help on robotics since it was not my initial engineering education. Moreover, Luc could answer to all my questions about the set-membership approach based on interval analysis when I faced issues on the concept.

I would like to thank Annick for all the administrative actions that have to be handled for a thesis at ENSTA Bretagne.

I want to express my aknowledgments to all members of the departement OSM (Ocean Sensing and Mapping) at ENSTA Bretagne for the interesting discussions around the underwater environment and more. It provided a great work environment during the three years.

I would like also to thank all the collegues from the small entity ECA Robotics in Brest for their support on the administrative aspect and the good moments shared.

I would express my gratitude to Benoit Desrochers who developed a Python library called **pyIbex** which is in part a binding of the IBEX library in C++ for the interval analysis during its thesis. The library is available on GitHub at:

https://github.com/benEnsta/pyIbex

Finally, I would like to thank my family for their support and their constant encouragement along the three years and in particular Amélie with whom I share my life before the beginning of the thesis.

Contents

List of Figures

List of Tables

Acronyms

ATR Automatic Target Recognition. 27, 54

AUV Autonomous Surface Vehicle. 49

AUV Autonomous Underwater Vehicle. 5, 6, 22, 27, 30, 43, 44, 49, 56, 71, 86, 305

DVL Doppler Velocity Log. 7, 9, 12, 22, 27, 28, 48, 59, 71, 86, 183, 305, 311

EKF Extended Kalman Filter. 16, 18

FLS Forward Looking Sonar. 8, 18, 19, 21, 22, 28, 35, 42, 43, 48, 49, 52, 53, 69, 100, 141, 142

FOV Field Of View. 141, 142, 174, 306

GPS Global Positioning System. 1, 86, 90, 305

INS Inertial Navigation System. 7, 9, 12, 20–22, 27, 28, 71, 86, 305

MBES MultiBeam EchoSounder. 20–23, 155, 182

MCM Mine Counter Measure. 2, 6, 17, 19, 25, 26, 28, 29, 33, 35, 49, 53, 54, 63, 64, 66, 69, 71, 87, 141, 305

MILCO Mine Like Contact. 26, 27, 29, 42, 63, 66, 87, 185

ROV Remotely Operated Vehicle. v, 5, 6, 49, 305

SAS Synthetic Aperture Sonar. 19, 27, 28, 42, 56, 64, 141, 182, 305

SIFT Scale-Invariant Feature Transform. 15, 16

SLAM Simultaneous Localization And Mapping. 14, 16, 18, 19, 23, 24, 55

SONAR SOund Navigation And Ranging. xiii, 15–17, 22, 305

SSS Sidescan Sonar. 9, 30, 182

SURF Speeded-Up Robust Feature. 15, 16

TERCOM TERrain COntour-Matching. 20, 21

TRN Terrain Relative Navigation. 14, 20–23

UUV Unmanned Underwater Vehicle. 4, 6, 13, 141

Chapter 1

Introduction

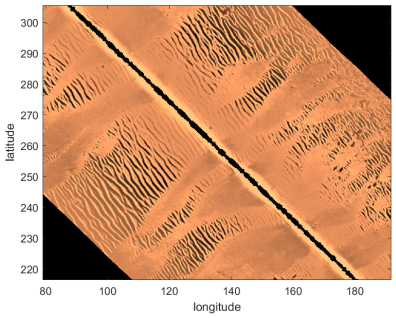
The autonomous navigation has been the topic of intensive researches during the last decades due the increasing ability of robot to perform complex tasks autonomously. More and more the Humans intend to replace some human tasks by robot to perform for example demanding jobs such as the work in the mines, replace some expensive work such the detection of underwater mines with surface vessels and to replace dangerous tasks such as the neutralization of bomb.

The level of autonomy of a system is determined by its ability to complete a mission by using smartly its energy, navigating in the environment and taking some decisions. The navigation autonomy can be defined as the ability of a robot to navigate precisely and estimating the position with little or non significant error over periods of time. Correctly estimating the position and the attitude within the environment is the key of the success in the different robotics applications. In outdoor environment, robotics applications can rely on the absolute positioning system based on the Global Positioning System (GPS) since 2000. Initially developed for military operations, it provides a centimeter accuracy of the position in terms of latitude and longitude. The GPS signal can be unreliable where the interferences are too important in mountainous environment, cities or when it faces jamming in battlefield. As in *GPS denied environment*, another navigation solution has to be developed to still have autonomous robot and the solution is not trivial.

The use of the GPS signal is not possible underwater due to the strong attenuation of the electromagnetic waves in submarine environments. The lack of the absolute position has to be overcome to provide an accurate navigation solution and to develop autonomous underwater robot. An analogy can be made in indoor environment, when people enter a building they don't know, they firstly look for a map in the entry hall to find the way to the destination. At any time they checked their smartphone to know their position. The problem is then to find a way to reach the destination. The solution adopted could be to minimize the path length but they could risk to be lost if some ways go through restricted areas. Another solution would be to walk between areas of interest such as the welcome desks of the different departments in any big structure like hospitals. By navigating between the different areas of interest called relocation areas and interacting with their environment, they will likely find their way. This thesis will be focused on this last solution to improve the autonomy level of robots in underwater environments. Obviously this problem could be extended to any *GPS denied environment* such as indoor and confined environment like underground galleries. Many applications could benefit of this human behaviour to find a way and complete a mission.

(a) Poor visibility of an orange buoy at 3 meters. Picture extracted from [274].

(b) Sealife leading to outliers. Picture extracted from [274].



(c) Unstructured aspect, sand ripples area on a sonar mosaic.

Figure 1.1: Challenging environment

1.1 The underwater context

1.1.1 Challenges

Water covers a major part of the Earth surface and according to the American National Oceanic and Atmosphere Administration (NOAA), 95% remains unknown by the Human. The recent advances in underwater technologies enable the Human to discover and understand more and more this environment feared during the navigation history and the discovery of new horizons. Inaccessible areas for Humans due to the high pressures of the column water are now explored with robots.

The success of an autonomous mission could be affected by the presence of strong currents. For example in the Mine Counter Measure (MCM) context, navigating perpendiculary to the direction of the current could degrade the image quality provided by a sonar during a survey [337] and some potential dangerous objects may not be detected by an operator or an algorithm.

Moreover the poor visibility in the deepest water due to the lack of light or opacities in shallow water make useless the use of cameras as depicted in Figure 1.1(a). The sea life may lead to outliers in the detection process of landmarks for example as shown in Figure1.1(b). The underwater environment is an unstructured environment compared to man-made environment as presented on a sonar mosaic depicted in Figure 1.1(c) which corresponds to a georeferenced sonar image.

However acoustics seems to be an interesting mean in underwater environment to overcome the communication and visibility issues. Indeed it has likely been the way of communicating for the marine animals for millions of years. Leonardo Da Vinci was among the first to develop the idea of underwater acoustics when he suggested [317]:

> "If you cause your ship to stop and place the head of a long tube in the water and place the outer extremity to your ear, you will hear ships at a great distance from you."

A major step in the knowledge of underwater acoustics has been established during an experiment on Lake Geneva in 1826, represented on figure[1] 1.2, where Daniel Colladon, a Swiss

[1]Picture extracted from J.D. Colladon, Souvernirs et Memoires, Albert-Schuchardt, Geneva, 1893.

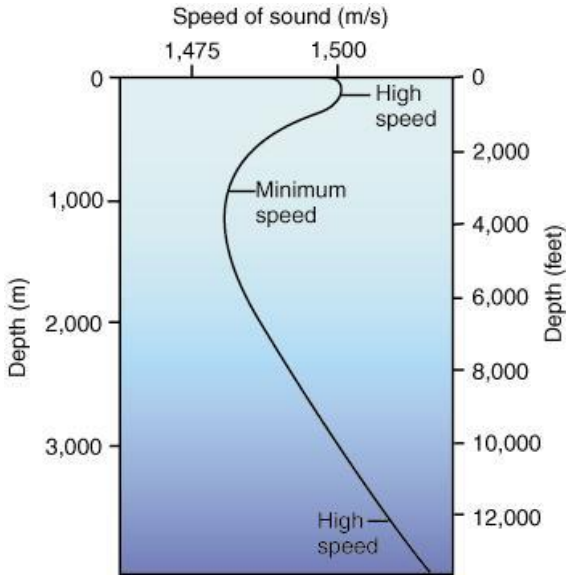


Figure 1.3: A sound speed profile: the speed varies according to the the depth

physicist, and Charles Sturm, a French mathematician, measured the sound speed at 1435 m/s over a 17 kms distance. It was later realized [57][199] that the speed in saline water was greater and it is affected by the temperature too. Usually the sound speed profile is represented as a function of depth as shown in Figure 1.3 but in reality at a given depth, it depends on the temperature, this depth and the salinity.

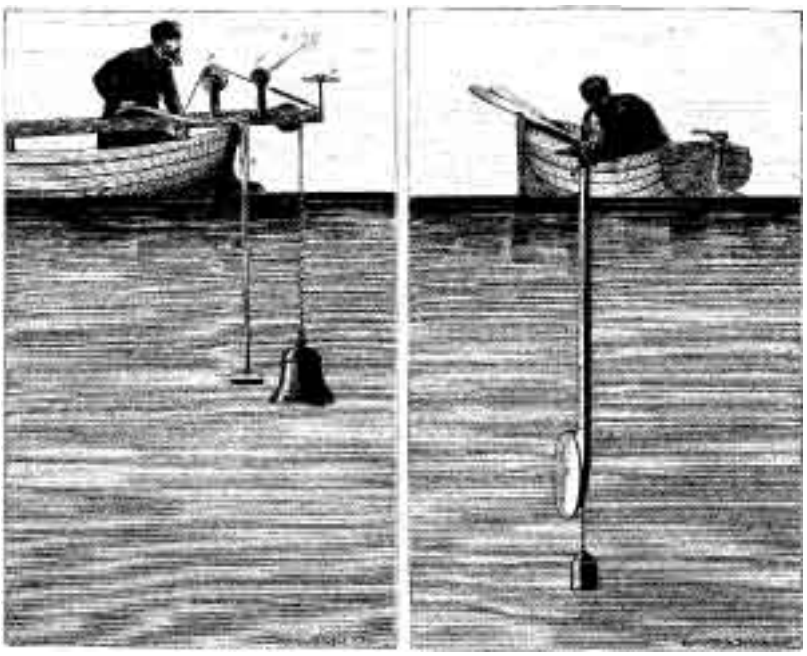


Figure 1.2: Experiment on Lake Geneva to determine the speed of sound in water. The bell was struck simultaneously with ignition of gunpowder on the left boat. The sound and the flash were observed 10 miles away on the right boat. The time between the flash and the sound reaching the boat enables to calculate the speed.

An acoustic wave is well suited to perceive distance between an emitter and any obstacles but the propagation depends on many factors. The direction of propagation is determined by the sound speed gradients. The propagation comes from the wave equation. Some models [146]

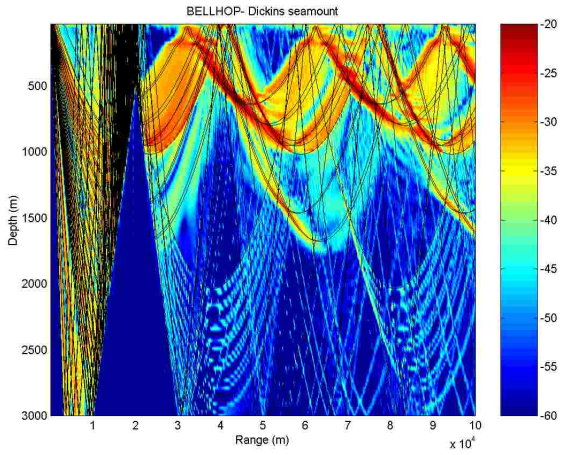


Figure 1.4: Paths taken by an acoustic signal. Some are reflected on the surface water and others on the seafloor. This figure is extracted from BELLHOP [273], an acoustic ray propagation tracing program.

have established under some simplifications the set of solutions for the propagation, as the ray theory shown in Figure 1.4, efficient at short range and high frequency. Therefore the sound signal usually doesn't propagate in a straight line.

1.1.2 Missions

Many sectors are interested by the use of underwater robots.

Defense

The defense is one of the first contributor in term of investment and research to the development of underwater autonomous robots. Indeed, underwater autonomous systems present many advantages to collect operational, tactical and strategic intelligence data for the commandmant. Moreover the ease of clandestine deployment and the small footprint and signatures make them an interesting ally in military operations. According to [36], the US Navy identified several missions for Unmanned Underwater Vehicle (UUV) and ordered them according to their priority degree in the field. The top five missions are listed below and can obviously be conducted without UUVs:

- Intelligence, Surveillance and Reconnaissance (ISR)
- Mine CounterMeasures (MCM)
- Anti Submarine Warfare (ASW)
- Inspection and Identification
- Oceanography

Among these missions, one can notice the MCM missions. The different steps in a MCM mission will be detailed in section 1.1.3. This will be the concern of this thesis. Moreover the U.S. navy mentioned the underwater mine removal as one of the most problematic mission for UUVs and to the NAVY in general [231].

Figure 1.5: Robots used for the exploration of the RMS *Titanic* wreck which sank in 1912. On the left, the unmanned ROV *Jason Jr.* taking some photographs, tethered by the manned submersible *Alvin* on the right picture.

Inspection and Identification missions are steps in MCM missions for mine identification and removal. However these missions could refer to clandestine operations such as spying an enemy ship with human divers setting up some electronic listening devices on a ship hull.

One of the listed mission is called Anti-Submarine Warfare (ASW). The main objective is to secure an area from submarines [324] and detect any intruder that would enter an harbour or an area desired to be safe.

Oceanography will be discussed in scientific missions.

All these operations can be conducted by an UUV.

Scientific missions

Oceanography takes today benefit of the use of Autonomous Underwater Vehicles (AUVs). Indeed AUVs can perform oceanographic reconnaissance in shallow and deep water area to collect various data. It is usually used to map the bottom, determine the profile of the bathymetry, profile the sub-bottom, classify the underwater environment with the acoustic and optical images, and characterize the water-column including the determination of the ocean current-profile (with tides), the temperature and salinity profiles. Oceanography is first-of-all highly related to military operations and globally to any subsea operations.

The marine archaelogy [197] has been highlighted in 1985 with the discovery and exploration of the wreck of the RMS[2] *Titanic*, lying on the seabed at 3800 meters, 600 kms south-southeast of Newfoundland. The ROV *Jason Jr.* controlled through a fiber optic cable from aboard the DSV (Deep Submergence Vehicle)*Alvin*, carrying usually two scientists and a pilot, explored the wreck and photographed the area where the submersible could not access. Figure 1.5 shows on the left *Jason Jr.* and on the right *Alvin*. The pictures are extracted from *Oceanus magazine*[3].

Recently, a 2400-year-old wooden vessel, likely a merchant greek vessel, has been discovered in the Black Sea at 2000 meters depth by the Black Sea Maritime Archaeology Project (MAP). The unusual preserved shipwreck, as shown in figure 1.6, has been photographed by a ROV

[2]Royal Mailing Ship

[3]http://www.whoi.edu/oceanus/feature/a-titanic-tale

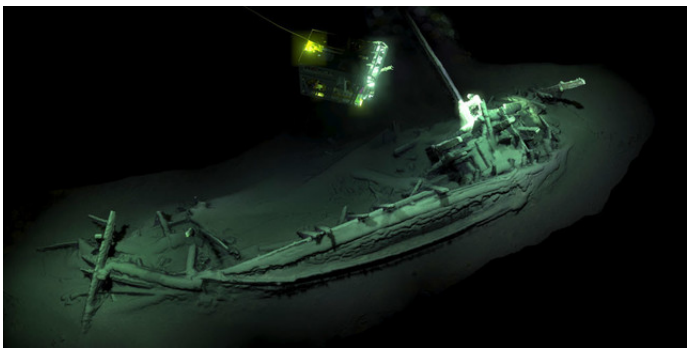


Figure 1.6: The black sea wreck found in an usual preservation state. © HO / Black Sea MAP / EEF expeditions / AFP

equipped with cameras. It shows some ship construction features, such as the mast and rowing benches, that have until now not been found in an intact conservation state on ships at this age.

Commercial missions

The main commercial mission occurs in the offshore oil and gas industry where they are looking for underground reservoirs to better extract the hydrocarbons. Instead of dragging long cables called streamers from surface vessels, which may be several kilometers long, a group of AUVs are able to perform gravitometry, magnetometry or seismic surveys to localize the underground reservoirs. Moreover, through the mapping of the seabed, it can help to determine pipeline routes to save costs as for the deployment of undersea cables for communication. It avoids the requirement of excessive length of cable from surface vessels in deep water. Moreover AUVs enable the inspection of pipelines and cables to check the reliability and localize potential failures.

In this thesis, we fill focus our application in the MCM context and the need to develop autonomous system for mine identification/neutralization. However, the concept presented can obviously fit to other missions.

1.1.3 Robots

Compared to the submersible like the DSV *Alvin*, an underwater robot is an unmanned underwater vehicle called UUV. As mentioned in the previous subsection, the UUVs proved to be successful in many kind of applications and sometimes necessary in certain situations. It is automatically propelled and is equipped with an energy source provided by embedded batteries or by another vehicle such as a surface vehicle. It can be remotely controlled by a cable in general or can perform autonomous mission. These robots are an interesting mean to collect fastly physical, acoustical and visual information that are stored in an embedded memory. The level of autonomy, briefly presented in the introduction, is based on the nature of the link to the surface. Generally, the UUVs are divided in two categories: the Remotely Operated Vehicle (ROV) and the Autonomous Underwater Vehicle (AUV).

1.1.3.1 Remotely Operated Vehicle (ROV)

The ROV is a Remotely Operated Vehicle controlled by a human pilot from a station on a boat. The robot is linked to the surface by a cable by which a lot of information transit such as the

Figure 1.7: Ocean One manipulating fragile objects

command to the engines or the data collected by the sensors. Due to the tension of the cable, it enhances some disturbances on the robot motion that affect the stability and makes it harder to complete the different tasks. However, the presence of a cable enables an easier recovery of the vehicle and avoids to loose the contact. Usually a ROV is equipped with exteroceptive sensors such as cameras and sonars to sense the environment and sometimes it is constituted of manipulator arms or hands for dexterous tasks such as marine archaeology which requires precise motions to avoid any degradation of ancient objects. As depicted in Figure[4]1.7, Ocean One has been developed by the robotics laboratory of Standford University, led by Oussama Khatib, in collaboration with the DRASSM (Département des Recherches Archéologiques Subaquatiques et Sous-Marines), to imitate the sense of touch of a human diver during underwater search.

The size of the ROV is important when exploring in shallow waters or cluttered spaces. Mini-ROVs are for example used in these difficult conditions thanks to their high maneuverability, however they suffer from a weak propulsion. Consequently, the tether cable has an high influence on the robot motion. To overcome this problem, a chain of mini ROVs is proposed in [177] where several robots are linked by a tether. A portion of the tether linking two successive robots, a leader and a follower, is managed by the follower robot to propose a visual regulation of the tether thanks to embedded cameras.

In MCM context, the K-ster from ECA Group, represented in Figure 1.8, has been developed for mine warfare. The model presented is the K-ster I, "I" for its identification purpose. The design is the same as for the K-ster C which is the mine killer version, also called MNV (Mine Neutralization Vehicle). Their head is tiltable to aim at the target for identification with a camera or for mine disposal with a shaped charge. In the case of mine disposal, the vehicle is intended to self detonate for time saving and neutralization guarantee. Therefore the design has to be low-cost. Some characteristics of this ROV can be found in Table 1.1. Expensive navigation sensors such as Inertial Navigation System (INS) and Doppler Velocity Log (DVL) which can provide an accurate estimation of the position are prohibited. However the command and the control system enable the robot to get fast and stable platform in various and difficult conditions such as strong currents or turbulences behind the object to be inspected.

Another ROV from ECA Group, called Seascan MK2, represented in Figure 1.9, has a more complete pannel of underwater tasks compared to the K-ster. Moreover it takes advantage of

[4]https://cs.stanford.edu/group/manips/ocean-one.html

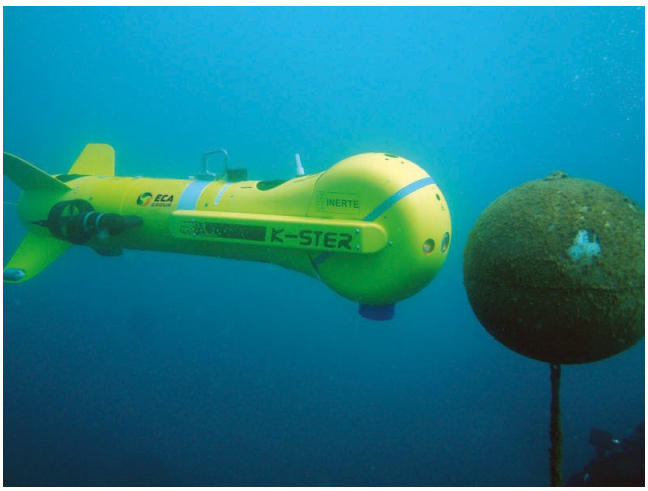


Figure 1.8: K-ster I (Identification) from Eca Group. It is designed for mine warfare. The K-ster on the picture has an inert head (yellow head). If the head was white, it would carry an explosive charge and is named mine killer.

real time data gathering through the optical fiber without the limitations of classical ROV towing an umbilical. The Seascan is quite performant in difficult conditions with strong currents and obstructed areas where an umbilical may be caught. Its long range may help for pipeline inspection.

Feature	**Specifications**
Hull	Length: 1500 mm Height: 430 mm Width: 500 mm Weigth in air: 50 kgs
Endurance	2 Hours
Max speed	6 knots (3 m/s)
Cruise speed	3 knots (1.5 m/s)
Operating depth	up to 300 m
Operational range	up to 2000 m
Batteries	Li ion
Sensors	Dual Frequency Forward Looking Sonar (FLS) Color video camera

Table 1.1: Some characteristics of the K-ster from ECA Group

1.1.3.2 Autonomous Underwater Vehicle (AUV)

The AUV is an Autonomous Underwater Vehicle that is equipped with localization and navigation devices to complete a mission autonomously. It's programmed before to follow a predefined mission such as the survey of an area with a *boustrophedon* pattern for example. It is equipped with its own source of energy with embedded batteries and it is usually possible to communicate with an acoustic link. Depending on the depth of the mission, the characteristics of the AUVs are not the same. Two types of AUV exist: the passively controlled and the dynamically controlled vehicles.

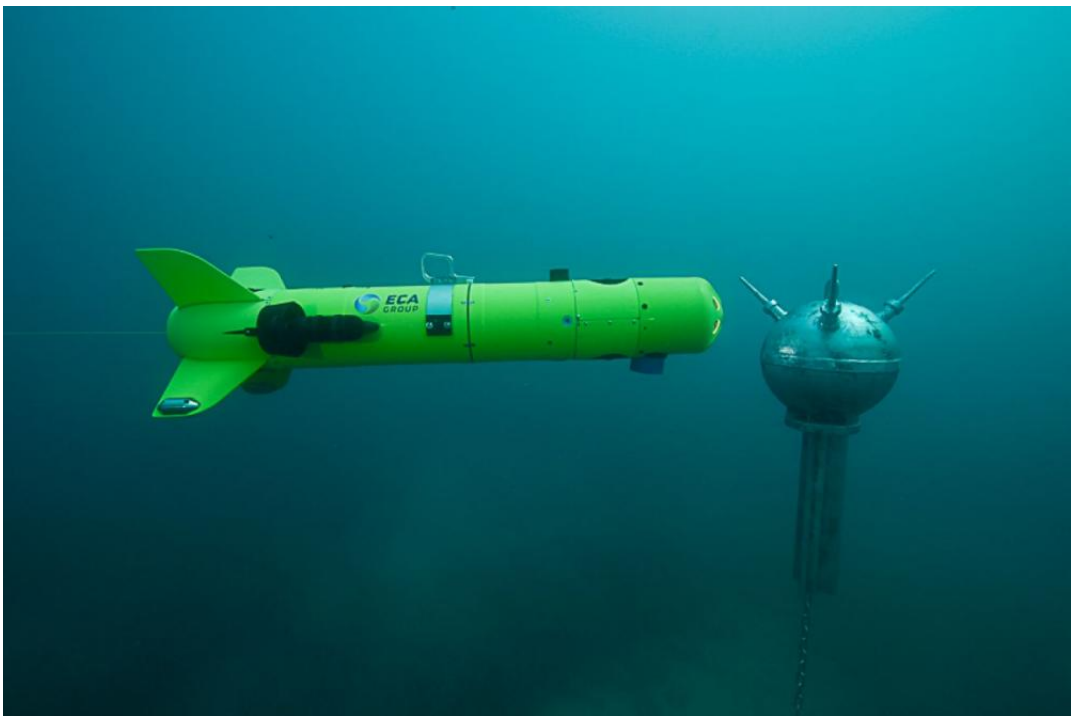
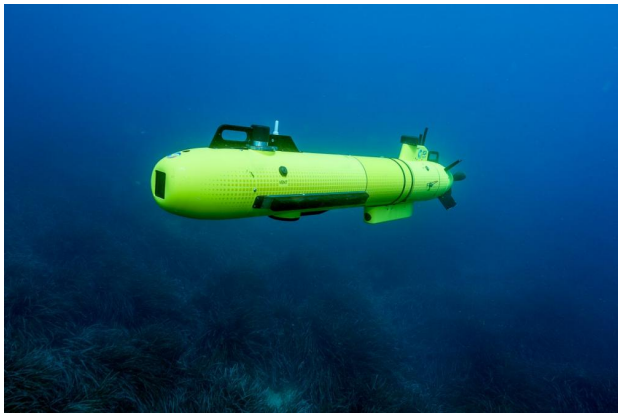


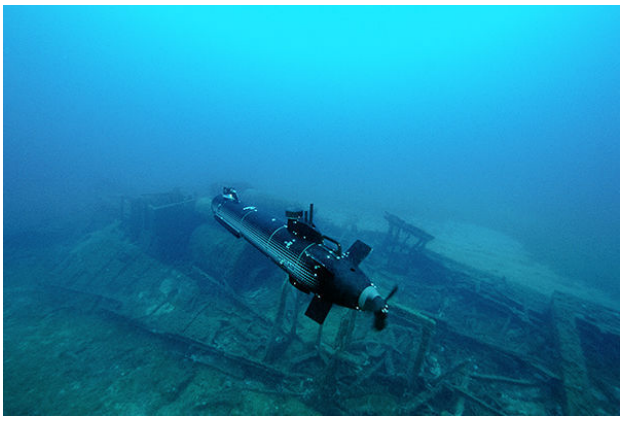
Figure 1.9: Seascan from Eca Group. It is dedicated to inspection missions.

Firslty, gliders [265][104] and buoys [189] are passively controlled autonomous underwater robot, as they use the ocean current to navigate for a long period and on long range to gather oceanic data. Buoys stay either on the surface or can maintain a constant depth during the mission. As it relies on the current, the prediction of the path is a little bit random. Gliders and wave gliders rely on wings to produce lift and forward motion, but their speed does not exceed (0.25-0.5 m/s) that can enhance an undesired path in strong currents.

Secondly, AUVs, as commonly known, are mostly torpedo shaped to minimize the drag so they do not require high power propulsion. Many models exist today, with a length varying between less than a meter up to 10 meters, and a diameter from a dozen centimeters to meters. The size depends on the mission which defines the payload embedded such as the sonars and the batteries. These elements are the biggest ones to carry. Recent works and technology advances reduce the size of the AUVs that can be now human portable with modules added depending on the mission desired. Many AUVs are available today on the market with different performances, sizes and prices. ECA Group, for example, proposes a multi task AUV called A9, represented in Figure 1.10. Several versions of the A9 exist today, such as the A9-E in Figure 1.10(a) or the A9-M in Figure 1.10(b) for military missions. Its color is more adapted to clandestine missions for mine disposal in the region around an harbour for instance. Many sensors are embedded in the vehicle including in particular an INS and a DVL that can provide an accurate navigation solution. These sensors will be discussed in the next section. Thanks to a reliable navigation, it can perform long term underwater mission without surfacing to get GPS fix. Moreover the interferometric Sidescan Sonar (SSS) provides simultaneously a 2D image of the seabed and the bathymetry (bottom profile). This AUV is well adapted for REA (Rapid Environment Assessment) missions thanks to its large coverage rate based on side-looking sonar.



(a) A9-E.



(b) A9-M, military version of A9.

Figure 1.10: A9 from ECA Group. On the left, the model E and on the right military version of A9 with its dark color.

Feature	**Specifications**
Hull	Length: 2 m Body diameter: 23 cm Weigth in air: 70 kgs
Endurance	up to 20 Hours
Max speed	5 knots (3 m/s)
Cruise speed	3 knots (1.5 m/s)
Operating depth	3-200 m
Sensors	Interferometric Sidescan Sonar Video CTD (Conductivity Temperature Depth) Environment sensors
Navigation	Inertial Navigation System (INS) Doppler Velocity Log (DVL) Global Positioning System (GPS)

Table 1.2: Some characteristics of the A9-E from ECA Group

1.2 UUV localization and navigation

The underwater environment is challenging due to many issues introduced in Subsection 1.1.1. Solving the localization problem is the key issue to complete any mission. The localization describes the process to estimate the vehicle's position in a given reference frame based on proprioceptive and exteroceptive measurements. The first ones provide information relative to the state of the robot through acceleration, heading, speed compared to the second ones that gather information about temperature, distance from beacons, distance from landmarks, *etc...*

A "good" localization system is important to collect meaningful data and to ensure the safety of the robot. Indeed, the robot should be able to avoid any obstacles and be recoverable after a mission for example. Contrary to aerial or terrestrial robots that can rely generally on an absolute positioning system (GPS), new solutions have to be developed for an UUV. Moreover, the localization system has to be embedded to conduct autonomous mission. This subsection provides an overview of the localization techniques and the sensors used to navigate in this

GPS denied environment.

The different localization procedures lay mainly in three categories:

- Inertial/Dead-reckoning: estimating the position from successive proprioceptive measurements.
- Acoustic beacons: estimating the position through the time of flight (TOF) of signals from beacons.
- Geophysical: estimating the position with external environmental information as reference.

1.2.1 Definition of the problem

Before introducing the different techniques, it is necessary to formalize the localization problem. As mentioned before, the sensors embedded gather different types of data to estimate the robot current state. It is generally defined in robotics community as state equations where a slight difference exist between bayesian and set-membership methods about the noise.

In the set-membership community, proposed in this thesis, the state equations are defined as follows:

$$\begin{cases} \dot{\mathbf{x}} & = & \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) & (\textit{evolution equation}) \\ \mathbf{z}(t) & = & \mathbf{g}(\mathbf{x}(t), \mathbf{m}) & (\textit{observation equation}) \\ \mathbf{x}(0) & \in & \mathbb{X}_0 & (\textit{initial state}) \end{cases} \tag{1.1}$$

In these equations, $t \in \mathbb{R}$ defines the time and $\mathbf{x} \in \mathbb{R}^n$ is the state vector of the vehicle: position, heading, speed, *etc...* it is called the *pose* of the vehicle. The *evolution equation* $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is differential and describes the state evolution of the robot based on the input vector $\mathbf{u} \in \mathbb{R}^m$ (generally corresponding to proprioceptive measurements). Due to the presence of noise, generally $\mathbf{u}(t)$ belongs to an interval $[\mathbf{u}(t)]$. The observation equation corresponds to exteroceptive measurements such as the distance to some beacons, the distance to landmarks or a part of the environment (shape) through a vector $\mathbf{z} \in \mathbb{R}^p$ related to the state of the vehicle $\mathbf{x}$ and the *observation function* $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^p$. Again a noise may be present, it is then defined as intervals or shapes. The notation m refers to any information about the environment, it could the position of a beacon or a map $\mathbb{M} \subset \mathbb{R}^q$ of the environment where q is the dimension of the workspace (generally two or three). The last equation describes the initial state of the problem at $t = 0$ where the position belongs to an initial domain $\mathbb{X}_0 \subset \mathbb{R}^n$. The functions $\mathbf{f}$ and $\mathbf{g}$ may be uncertain and non-linear.

For example, in a 2D environment, a robot can be described by $\mathbf{x} = (x_1, x_2, \theta)^\intercal$ where x_1 and x_2 are respectively the East and North positions, and θ corresponds to the heading of the vehicle.

1.2.2 Inertial/Dead-reckoning navigation

Dead-reckoning refers to the main mean of pose estimation of the robot when it cannot rely on absolute positioning system or acoustic beacons. The position estimation is updated upon the knowledge of its orientation and velocity or acceleration vector. These measurements are called proprioceptive measurements since it enables the estimation of the current robot position through the *evolution function*.

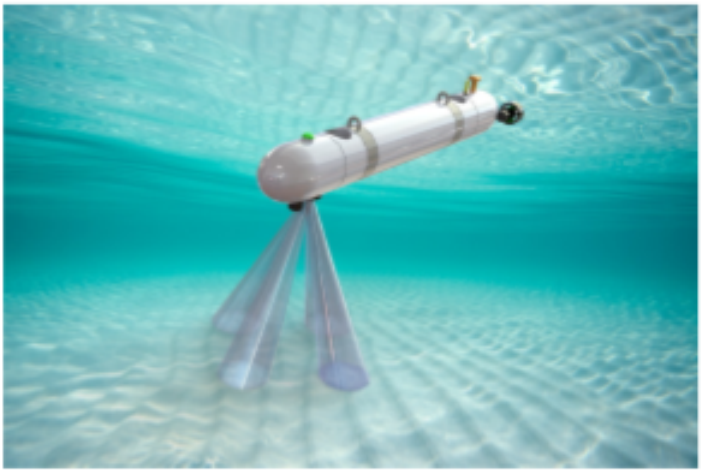


Figure 1.11: 4 downward looking beams DVL (picture extracted from [65])

An embedded IMU (Inertial Motion Unit) provides information on linear accelerations and rotation speeds of the system. Working with a magnetometer, the system is able to give the *Euler angles*: the *bank* ϕ, the *elevation* θ and the *heading* ψ. These angles are generally known as *roll* ϕ, *pitch* θ, *yaw* ϕ. The Euler angles or rates are then in general inputs to an AHRS (Attitude and Heading Reference System), which generates a stable estimation of vehicle orientation. The stabilized roll, pitch, and yaw are then used by an INS that combines other sensors information to provide a robot's state estimation based on algorithms such as the famous Kalman filter [152] in the bayesian community.

In underwater environment, the DVLs provide information about the ground referenced vehicle speed by emitting acoustic beams toward the seabed and using the acoustic Doppler effect. A change in sound pitch is proportional to the relative radial velocity between the source and the receiver. Most of the commercially DVLs use the time dilation to compute the velocity from a set of discrete pings, in general four downward-looking acoustic transducers and each oriented at 30° from the vertical. In this configuration, it results in four measurements of beam-component velocities representing each a scalar of the sensor velocity projected along each beam axis. A representation of the four downward looking beams is depicted in Figure 1.11.

The main disadvantage of the dead-reckoning method is that errors are cumulative, and consequently the robot position grows unbounded with the distance travelled. The errors may have various origins: noise, wrong calibrations of the units, *etc...* Moreover the local water current has to be taken into account in the kinematics model of the robot. This current can be measured with an ADCP (Acoustic Doppler Current Profiler) or can be obtained with the velocity measurement of the DVL. The ocean current can also be obtained by ocean prediction from current models [265].

The best INS system can achieve a drift of 0.1% of the distance traveled [94] for the most expensive system but generally the drift is estimated at $2-5\%$ of the distance travelled for mostly affordable systems.

The maximum vehicle time during subsea operations will be dictated by dead-reckoning/inertial navigation accuracy. Poor dead-reckoning quality will enhance frequent surfacing to get GPS fix. Vehicles operating near the coast and requiring a high surfacing frequency may collide a surface vessel. Surfacing frequently is not really possible for deep water applications and even

impossible for ice-covered oceans. Moreover in MCM operations, surfacing is a risk of being spotted by the enemy.

1.2.3 Acoustic-based navigation

Acoustic transponders can be used as beacons to guide the motion of the UUV without the need of resurfacing. Localization is achieved by measuring ranges from the TOF (Time Of Flight) of acoustic signals.

Most of the techniques fall into the following categories:

LBL (Long Baseline)

LBL is a solution where the beacons are placed over a wide mission area on the seabed. The localization is computed by triangulation of acoustic signals. In general, the beacons are georeferenced before the mission by a surface ship [170] or another AUV [323]. The AUV sends out an interrogation signal and the beacons reply in a predefined sequence. The two way travel time of the acoustic signal enables to compute the range. Alternatively the baseline can be synchronized and emits pings at the same time and therefore does not require the vehicle interrogation. It is similar to the solution of GPS intelligent buoys, where the beacons are placed at the surface of the ocean and benefit of the absolute positioning system GPS. The drawbacks of LBL system are the cost and the time needed to set up the beacons network. Moreover the range is limited and the range measurements are affected by the variability of the sound velocity profile of the water column due to different temperature, salinity, depth, *etc*...

Despite these limitations, it is a widespread system for robust, reliable and accurate localization device. It is particularly used in high-risk missions such as under-ice surveys [167].

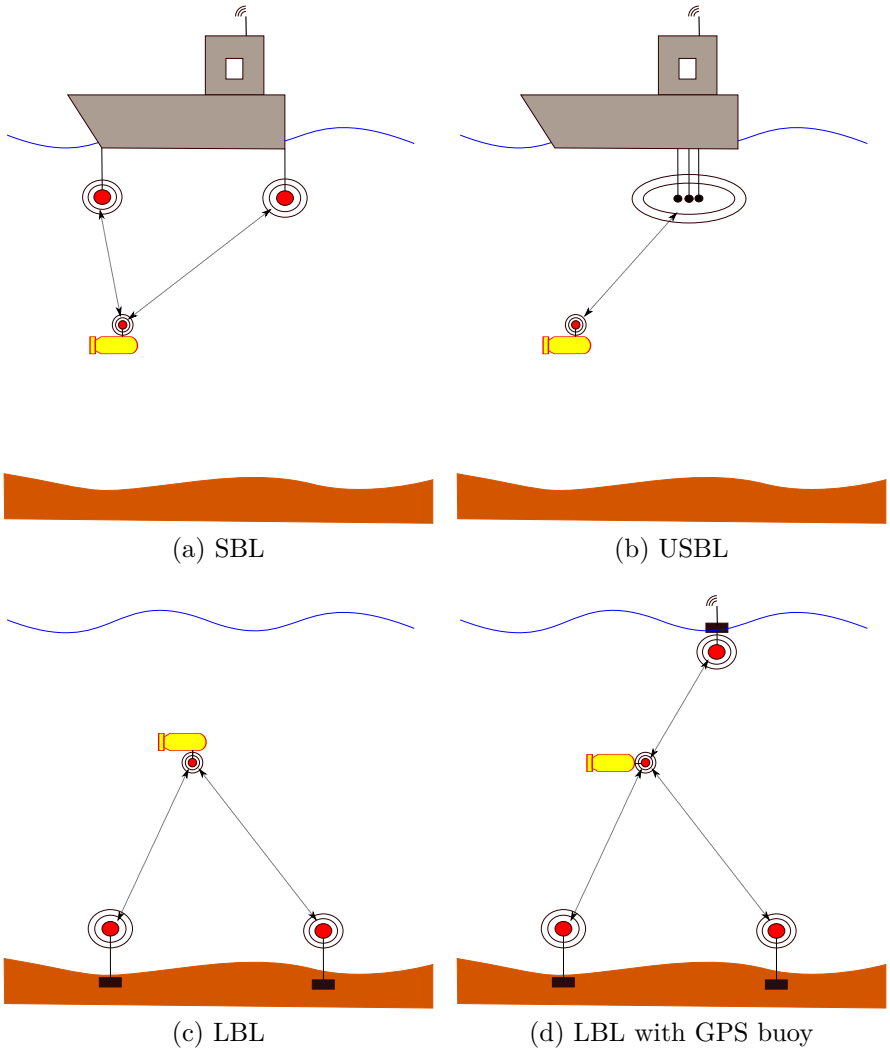
SBL (Short BaseLine) and **USBL** (Ultra Short BaseLine)

SBL and USBL navigation enable an AUV to localize itself relative to a surface vessel. In the USBL method, the relative range is computed by the TOF and the relative bearing is computed by phase differencing across an array of transducers that are about 10cm distant. The major limitation is the range. In SBL system, the transceivers are placed at the extremity of the surface ship and the same concept of triangulation as SBL system. The longer the surface vessel is, the better the accuracy of the position will be. It is better adapted to long ship rather than small boats.

A detailed review of these techniques, their advantages and their drawbacks has been discussed in [325].

1.2.4 Geophysical-based navigation

Geophysical navigation methods are based on the observation of physical features (or features variation) to perform localization procedure. The measured geophysical properties could be used as references for localization. As stated in [193], the idea is to match sensor measurements with the reference map to determine the vehicle position within the map, under the assumption that the parameters being measured are sufficiently variable to provide an accurate localiza-



(a) SBL (b) USBL

(c) LBL (d) LBL with GPS buoy

Figure 1.12: Acoustic-based navigation configuration

tion. The main advantage compared to acoustic-based navigation is the direct availability of the information and it does not require to set beacons. Moreover this kind of navigation is completely on-board, without the need of GPS, and enables operations in larger environment [211] such as clandestine mission in military context. The reference map can be built before the mission: it is called an *a priori* map, if some data are available, or it can be built while the robot is navigating: it is called Simultaneous Localization And Mapping (SLAM) method.

Geophysical-based navigation is commonly known in the robotics community as Terrain Relative Navigation (TRN) or TBN (Terrain Based Navigation) or even TAN (Terrain Aided Navigation). All the acronyms refer to the same principle. Underwater TRN has been the focus of a huge number of research efforts due to its potentially powerful solution for long-term and long-range navigation solution for AUVs as they can use directly the information available *in situ* through exteroceptive sensors. It's comparable to the human behaviour where a human to navigate in a known or unknown environment uses his senses including the "traditional" five senses, i.e. the sight, the hearing, the taste, the smell and the touch. This capacity that provides data for perception of the environment will enable him to evolve in a static or dynamic environment, react to avoid an obstacle for example, and localize himself in this environment.

Considering the sensors used in an UUV, TRN methods may refer to featureless and feature-

based navigation. Bathymetric-based and Gravity/Geomagnetic-based navigation refer to featureless navigation compared to acoustic image based navigation or optical image (called vision) based navigation which refer to feature-based navigation. Acoustic-navigation is classified as a featureless-navigation when a ranging SONAR is used and is called acoustic-image based method when it refers to imaging SONAR with features extraction. An exception can be made for acoustic image based navigation when the images are registered through iconic registration which is a featureless approach.

Two problems can be considered in both type navigation:

- Localization: with the aid of known *a priori* maps of the terrain, it consists in localizing the AUV within these maps. It is commonly known as *pose tracking* problem. It considers the map and the initial condition known.
- SLAM: when the maps are not available, it consists in mapping the environment and localizing itself within the map.

A third problem could be considered: *the initial localization* problem where only the map is available and the initial state must be revovered. This problem has already been tackled with Particle Filter (PF) methods such as Monte Carlo Localization (MCL) [69] or Rao-Blackwellised Particle Filter (RBPF)[82] where the map is represented by geometric features. The convergence and robustness of these algorithms remains a challenging task compared to interval methods [141] which are able to overcome these issues. The *initial localization* problem will not be discussed more in this thesis.

Some solutions in the two first problems in the underwater context will be discussed.

The following paragraphs refer to:

- Optical: vision-based navigation.
- Sonar: acoustic image-based navigation called "Imaging Sonar" and acoustic range-based navigation (bathymetric-based navigation) called "Ranging sonar".
- Magnetic and gravity: Gravity/Geomagnetic-based navigation.

1.2.4.1 Optical

Visual odometry is the technique to estimate the robot pose based on a sequence of successive images. In underwater environment, embedded stereo or monocular cameras can provide the images. Compared to monocular cameras, the full 6 DOF (Degree Of Freedom) transformations between successive images pair can be computed with the stereo cameras. The visual odometry can be estimated through optical flow technique for example. Many feature detectors have been applied such as Scale-Invariant Feature Transform (SIFT)[202], Speeded-Up Robust Feature (SURF) [19] and the alternative algorithm of SIFT-SURF called ORB [279]. Proving closing loops in the trajectory by associating non-consecutive images is one the most important problem in visual odometry, which is necessary to a have a bounded localization error.

The main limitations of visual odometry in underwater environment are the limited range of the cameras due the lack of visibility, the appearance of speckle noise and the presence of outliers such as fishs. It requires the use of artificial lights in dark water and to navigate close to the seabed. This technique is well-suited for feature-rich environments as it depends on feature detection and feature matching between successive images. It can be relevant in docking

operations.

In [92], the author proposes the new concept of visually augmented navigation (VAN) to solve the underwater visual SLAM problem. It solves a view-based Extended Kalman Filter (EKF)-SLAM fusing 6 DOF relative pose camera measurements from monocular overlapping seafloor imagery with classical dead-reckoning navigation sensors. The camera-derived relative pose measurements provide the spatial constraints for visual odometry and loop closure. The VAN has then been extended in [91] to an Information filter SLAM form where it maintains a sparse information matrix without approximations or pruning. It has been applied for the inspection of the RMS *Titanic* with SIFT and Harris [123] detectors for features extraction. In [285], a feature-based SLAM applied in underwater environment is proposed where the map constructed contains a set of 3D points and their associated SIFT/SURF descriptors coming from stereo camera. The SIFT/SURF features present many advantages because they are invariant to image rotation, translation, scaling and moreover they are not sensitive to changes in illumination, presence of noise and clutter, and distortions. The 3D points with the feature descriptors are stored as a local submap. It provides finally a large scale 3D reconstruction of the seabed from the reconstruction of many aligned submaps.

In the work [115][113][114], a mosaic-based navigation (MBN) for an AUV with a monocular camera is proposed. The mosaic is firstly computed *offline* and then it is directly used *online* for real-time navigation. The mosaic is built by taking into account the spatial pairwise constraints and by assuming a flat seabed. Consequently it estimates only the robot motion through the computation of homographies along 4 DOFs. It performed well on small areas (approximately 65 m^2) but it is not well-suited for large areas because it suffers from inconsistencies coming from the extended planar scene assumption. Moreover the localization method is only vision based localization and it does not fuse other sensors based navigation measurements.

1.2.4.2 Sonar

SONAR systems are the most spread exteroceptive sensors used in underwater environment as it propagates acoustic signals. Many types of SONAR are available today on the market, however they can be classified into two main categories: ranging SONAR and imaging SONAR. Both are intended to be used at specific frequencies depending on the range and resolution desired. More detailed information about imaging SONAR will be discussed in Chapter 2.

a) Imaging SONAR

The ensonified area for the different imaging SONARs are depicted in Figure 1.13. The table below sums up the different characteristics, with some applications and products examples.

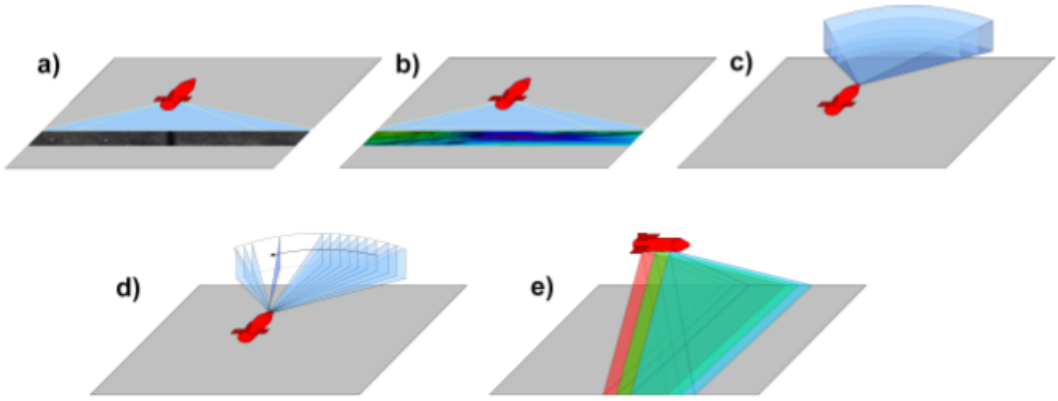


Figure 1.13: Sonar sensor swaths: (a) Sidescan Sonar (SSS) (b) Multibeam Echo Sounder (MBES) (c) Forward Looking Sonar (FLS) (d) Mechanical Scanning and Imaging Sonar (MSIS) (e) Synthetic Aperture Sonar (SAS) (pictures extracted from [242]).

Sonar	**Description**	**Figure**	**Applications**	**Products**
SSS	Sidescan Sonar, side-looking sonar, beams are perpendicular to the direction of the motion, 2D image of the seabed based on the intensity of the backscattered signal obtained after several ping emissions along a straight motion	1.13(a)	Area coverage in MCM mission for mine detection	Klein 5000
SAS	Synthetic Aperture Sonar, same configuration as SSS but coherent processing of consecutive displaced returns to create a virtual array, equivalent to SAR[5] in radar	1.13(e)	Area coverage in MCM mission for mine detection, better resolution than SSS but also constant resolution over range	Kraken MINSAS
2D FLS	Forward Looking Sonar, 2D image of the seabed obtained with one multi-beams ping emission oriented forward	1.13(c)	Obstacle Avoidance and Nadir gap filler	BlueView P900
MSIS	Mechanical Scanned Imaging Sonar, a single beam is rotated through the desired angle of view mechanically, beams oriented forward in general	1.13(d)	Obstacle Avoidance and area scanning	Tritech Micron

Table 1.3: Imaging SONAR

The different types of Sonars will be more detailed now.

SSS: SideScan Sonar

It is a side-looking sonar. Usually in REA (Rapid Environment Assessment) missions, an AUV is equipped with two side-looking sonars to form a star-board and a port image. Due to this configuration, the *nadir* is the area just below the AUV where no data are available, it's usually called *nadir gap*. A sidescan sonar image is described by an along-track (cross-range) and a cross-track (along-range) resolutions. The different resolutions are linked to the ping emission frequency, the velocity of the vehicle and the range desired but more detailed will be discussed in Chapter 2. The acoustic return from the seabed, called the backscattered signal, depends on the bottom type and is recorded in a sequence of cross-track slices. The sequency is then composed of the acoustic returns from each ping emitted at a specific frequency. Each ping is also composed of a multi-beam emission. The resulting image is called a *water-fall* image that represents the ping emitted in function of the range samples, which assemblies port and star-board images. In general, the detection algorithms are looking for an echo-shadow pair in the image, representative of an object protruding from the seabed. These objects are called landmark point and can be used by any SLAM algorithm to relocate the position of the robot. SLAM with the sidescan sonar has been firstly presented in [280] where an augmented EKF is used to create a stochastic map. The Rauch-Tung-Striebel smoother is used to filter the pose estimation when a loop closure occured to update all the previous poses. As precised in [282], the choice of the landmarks (targets) and a suitable data association are vital in sonar data where false returns called outliers are present. The automated feature detection is not trivial and data association based on MHTF filter (Multi Hypothesis Tracking Filter) [268] is adapted to sonar data. It consits in reducing the number of hypotheses when the data association is performed based on the exteroceptive measurements, the stochastic map and the vehicle navigation. Compared to FLS images, where the areas covered by the sensor swath as it moves overlap, there is little or no overlap in sidescan imagery. Usually the detection is done after a predefined sequence of pings and the data association is performed based on the vehicle poses historical. Sometimes it is meaningful to wait the construction of the sidescan image along a straight motion line to observe constellations of objects, homogeneous texture regions or large objects such as wrecks transformed after the detection into a symbolic format [344]. Then a fuzzy relaxation algorithm is used to perform the matching of sets. This concept introduces a delay between the real time navigation and the matching results. This delayed data association is known as image registration [348] in the image processing community. The method presented is defined as symbolic registration due to the extraction of element in the image such as geographical coordinates of individual objects or contour line of a textured region. Symbolic registration is a two step algorithm: firstly images are segmented and classified, and secondly a spatial transformation is estimated. It has been widely used in side-looking sonar imagery [49][322]. As precised earlier, sidescan sonar image can be registered through featureless registration, it is called iconic registration. This method consists in finding the spatial transformation between the pixels of two gray images [50] based on similarity metrics. Contrary to symbolic methods, it does not require any time consuming segmentation and is quite robust to illumination variations, occultations and noise highly present in sonar imagery. Moreover in featureless environment symbolic methods would fail whereas iconic methods would still provide a registration. Among iconic methods, a Fourier-based registration method has been applied in [235] to registrer parallel tracks in MCM context for change detection [234] between high resolution sonar images.

In [135][143], the sidescan SLAM is proposed as an interval constraint satisfaction problem

involving set-memberhip method. Even if the data association is done manually, it could be overcome with interval propagation involving integer variables (identity of a landmark). Moreover, since the high presence of outliers in underwater environment, an interval approach would generally return an empty set and then no trajectory estimation would be available. An alternative method could be to maximize the number of constraints satisfied [171][145] to be more robust to outliers. Recently, due to the unstructured aspect of the underwater environment, a shape based SLAM [71] has been proposed involving Thick Set [75] representation for the shapes. The problem is solved in a set-membership manner with contractors and separators. More details will be discussed in Chapter 4. This method enables to construct a map with shapes describing sets such as an area of texture like sand ripples. In [12], a selective submap joining SLAM is considered where a cascaded Haar classifier [328] for object detection is used. The difficulty with sidescan imagery is to design properly the features since it will be built differently upon a revisit [341].

FLS: Forward Looking Sonar

This system is mainly used for obstacle avoidance and as nadir gap filler in association with a sidescan sonar. It is commonly used in hovering AUV able to approach underwater structure. In [330], a feature based SLAM is proposed for ship hull inspections. The problem of feature extraction in FLS images is detailed in [147]. A feature based navigation (FBN) solution using a FLS in the context of revisiting a previously mapped environment with a sidescan sonar is proposed in [93]. It maintains a set of match hypotheses in parallel until the map matching score exceeds a threshold. This map matching score is called NAPS (Negative And Positive Scoring) incorporates Positive and Negative information. Positive information refers to information of prior features detected by the FLS sonar which are used in data association algorithm such as JCBB (Joint Compatibility Branch and Bound)[232]. A priori features were detected previously by the the sidescan sonar survey. Negative information [209] refers to information of prior features that were expected but undetected by the FLS. It consists to give a lower weigth to the trajectory (an hypothesis during the multi-hypotheses procedure) where it expected to detect a feature that was actually not detected [101]. The aim of the project in [93] is to propose the revisit for a low-cost vehicle with a FLS, by using an *a priori* map built with detections from sidescan sonar images. As indicated, this low-cost vehicle is intended to self detonate in MCM operations for mine removal. The idea of revisiting behind this project is close to the work developed in this thesis.

As in sidescan sonar imagery, symbolic and iconic methods can be used to register the images and reduce the uncertainty. In [133], a Fourier-based methods based on phase correlation technique is used for registration in order to create a FLS mosaic [326].

SAS: Synthetic Aperture Sonar

This system enables to improve the resolution in sonar imagery by using the same principle as SAR (Synthetic Aperture Radar) imagery in radar applications. It uses the motion of the sensor, i.e. the along-track motion of the vehicle, to create a large virtual array. As for the use of the sidescan sonar, the optimal employment of the Synthetic Aperture Sonar (SAS) is along a straight motion. The resulting resolution is comparable to the dimension of the transducer and is independent of the range between the sensor and the target. Due to this range independance, this system can be used at lower frequency, which enables longer range detection. However the vehicle has to move precisely along a straight motion with a tightly prescribed speed, and it enhances more complex image processing. Further details on the SAS concept can be found in

[121], [126] with an application on the HUGIN AUV [122].

MSIS: Mechanical Scanned Imaging Sonar

This system will not be discussed in this thesis, but the reader may refer to the work at the University of Girona for further details [271][270].

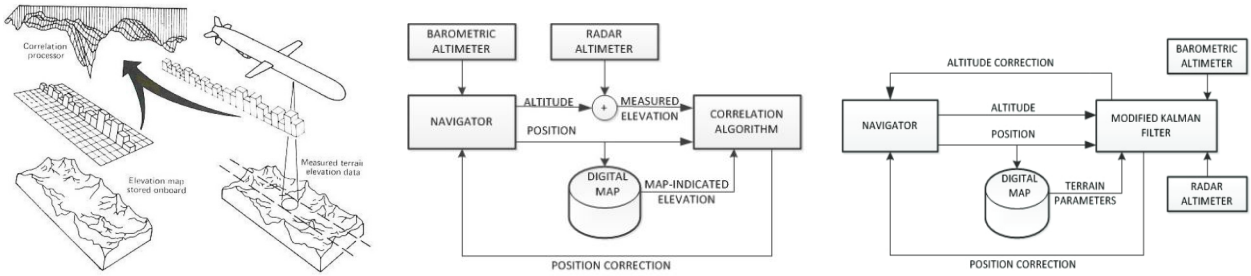
b) Ranging SONAR

It refers to bathymetric-based navigation as stated above. This method uses the spatial variation of the terrain to bound the growing error of inertial navigation by comparing the terrain measurements to maps stored in the robot. The physical features of the seabed are stored in a DTM (Digital Terrain Model), or in french MNT (Modèle Numérique de Terrain), known as bathymetric map. The accuracy of this method relies on the accuracy of the motion sensors, the quality of the maps, the presence of variation in the morphology of the seabed and the sensitivity of the ranging sensor to changes in AUV pose [211].

Bathymetric TRN was initially developed for military purposes, such as missile guiding [84] or aerial military vehicles, when the absolute positioning system (GPS) was not yet reliable enough. Moreover TRN methods are robust against interferences and jamming, and can be used under all weather conditions and at any time (night or day). The development of high-resolution DTMs provided by satellites plays an important role in the increasing interest of such methods. Nowadays TRN methods are spreadly used in both civil and military applications such as lunar landing [148] or aircraft system [63]. The first algorithms dedicated to TRN methods are TERrain COntour-Matching (TERCOM) and SITAN (Sandia Inertial Terrain Aided Navigation) methods.

Due to this dual origin of the bathymetric-TRN method, it was firstly divided into two categories: Batch methods and Sequential methods. The difference are schematically represented by the structure of the algorithms, depicted in Figure 1.14. In the case of TERCOM, due to the acquisition of the terrain profiles by a set of readings from the altimeter sensor and then the simultaneous process, it is called a Batch method. However, according to [211], this system is successful when it is coupled with a highly accurate INS or a dense range sensor such as MultiBeam EchoSounder (MBES) since the motion of the vehicle is not taken into account during the different measurements acquisition. Initially, TERCOM was developed to provide positional fixes to update the INS by taking the best match from the correlation between the measured terrain profile and the DTM. In the literature, many metrics can evaluate the degree of similarity such as the Cross-COrrelation (COR), the Absolute Square Distance (ASD), the Mean Absolute Difference (MAD) or even the Minimum Square Distance (MSD). If the *a priori* map is large, some methods [13] use the dead-reckoning estimate to bound the area when the correction is performed to reduce the computational cost. On the contrary, the SITAN method, which is sequential, considers each new measurements independently and feeds a Kalman filter. Consequently it updates directly the navigation solution. The main advantage of SITAN is the reduced computational cost compared to TERCOM due to the continuous position updates. Nevertheless, the sequential method needs an accurate initial position of the robot, the missile or the vehicle compared to batch processing techniques. Various algorithms based on these two concept were proposed later [318], and even hybrid solution such as TERPROM (TERrain PROfile Matching). The reader may refer to [318] or [215] for more detailed information.

An atlernative classification of underwater TRN is proposed in [7] where the methods are



(a) TERCOM principle (b) TERCOM algorithm (c) SITAN algorithm

Figure 1.14: TERCOM and SITAN algorithms. Pictures extracted from [107].

decomposed in Search Area methods and Gradient-based methods instead of Batch and Sequential methods. TERCOM would refer to Search Area method and SITAN to Gradient-Based method. Another alternative classification of TRN is proposed in [118] where it differences between the degree of integration with the INS, defining as loosely and tightly coupled integrations. In the latter approach, the bathymetric measurements are fused directly with all the other measurements in the filter with the INS. On the contrary, in the loosely coupled approach, the bathymetric measurements are processed in a filter apart until convergence, and then the position estimate is sent back to the INS.

Underwater TRN is a recent research domain compared to previous aerial techniques. The difference lies in the sensors used and the vehicle dynamics. Compared to aerial TRN which are more focused on the use of single beam sensor such as radar or altimeters, underwater TRN methods are interested in using multi-beam sensors such as MBES or 3D FLS to provide a large coverage of the seabed in high resolution.

One of the first work in underwater TRN is presented in [23]. A correlation position is estimated between some MBES measurements and a *a priori* known bathymetric map. The position estimation is then updated in a KF based on the correlation position. Later in [204] it proposes to match high resolution local depth maps on a large low resolution map by using a multi-scale analysis and some invariant points. In [237][236], the correlation is based on the Maximum Likelihood Estimator (MLE) under the assumption of large time between measurements. When the evolution and/or observation function are non-linear, EKF has been used in [294] with measurement provided by the best estimation of a matching precedure between common features of the observed map and the stored map, in order to refine the heading and the position of the vehicle. However EKFs provide "good" estimate only when the uncertainty is small, otherwise it diverges. Due to the non-linear aspect of the bottom profile, the TRN problem is highly non linear. Therefore non linear Bayesian methods were also of great interest to track the vehicle pose. In [236], a navigation filter based on the Unscented Kalman Filter (UKF) is proposed but it is concluded to be a not-optimal method for TRN. Later the attention was focused on non-parametric form of the Bayes filters such as Particle Filter (PF) or Point Mass Filter (PMF) for their simplicity.

PFs are recursive filters for solving the Bayesian estimation problem with non-linear motion and/or measurement models and without linearization techniques. A set of random "samples" are used to represent the probability distribution and are generated according to its probability density function (PDF). Each particles are then weighted based on the most recent observation. It has been used as TRN navigation solution in several works [213][211][250].

Not as popular as the PF, the Point Mass Filter assumes a posterior density represented by a set of point masses ordered in a grid. The continuous PDF can be otained by integrating over the masses in the grid. The size of the grid is usually fixed but there are some algorithms to adapt the grid mesh, which could be interesting for bottom profile where a high gradient PDF occurs. In [118], the PMF is applied as the TRN solution for AUVs at low altitude.

According to [212] and [8], the PMF is more robust and accurate than the PF, but the PF enables a higher dimensional search. However these filters show their limitations when the state space increases which can be the case when dealing with sensor-limited systems. Rao-Blackwellised Kalman Filter (RBPF), a combination of a PF and a KF, is an alternative approach to overcome these limitations when the model contains a linear sub-structure. The states following a linear model can be estimated with the KF whereas the non-linear states are estimated with the PF. RBPF has been used as TRN solution in [306] to merge measurements of bottom range from an altimeter, a 3D FLS and a MBES with dead-reckoning data provided by a DVL and an IMU (Inertial Motion Unit). In the proposed approach, the state vector is composed of the 2D position of the robot and a 2D velocity bias due to the unknown water currents. It proposes the Smooth Kernel Particle Filter (SKPF) to obtain more consistent results and it improves the robustness against outliers and flat terrains. This latter, usually known as information poor terrain, may lead to false fixes. In [68], a new method is proposed to avoid this false fixes. It proposes to adjust the filter weighting to depend on the relative amounts of map error, sensor error and terrain information. The method is similar to the variance adjustments used in robotics [309] to reduce the likelihood of overconfidence.

In many experimental cases, the MBES is usually coupled with high grade INS. However, in recent works, the focus was on TRN techniques applied to sensor limited systems [65]. These systems are usually composed of low accuracy inertial sensors and low-information exteroceptive sensors like DVLs or altimeters [81]. Due to the expensive cost of an accurate and complete INS, this concept is legitimate.

The main bathymetric sensors are mentioned in the table 1.4. The DVLs are not listed in Table 1.4 but as previously mentioned it corresponds in general to a set of four single beams pointing downward.

Sonar	**Description**	**Figure**	**Applications**	**Products**
Echo Sounder	Single and narrow beam to determine the depth, comparable to altimeter	not yet	Altitude estimation	not yet
SBP	Sub Bottom Profiler, low frequency echo sounders to penetrate the seabed, provides information about the layers of sediment	not yet	Geological and geophysical exploration survey, route survey for pipeline laying, marine construction	Kongsberg SPB 120/300
MBES	Multi-Beam Echo SOunder, provides bathymetric maps from TOF returns	1.13(b)	Large coverage area, seabed survey, REA mission	not yet

Table 1.4: Ranging SONAR

Bathymetric SLAM

Bathymetric SLAM in underwater environment presents many advantages since it does not require features detection and tracking, and is applicable to unstructured environment. A Bathymetric Particle SLAM (BPSLAM) is proposed in [16] based on a featureless FastSlam [224][298] implementation. Each particle maintains a current state estimation of the vehicle state and the 2D bathymetric map. The difficulty with such a large state space is the resampling process, vital element of PF techniques.

In [71], a shape-based SLAM approach is proposed in a set-membership manner based on interval computation. The bathymetric profiles generated by a MBES are decomposed in level layers such as isobaths. These different layers represent shapes of the acquired terrain profiles. This enables to have several 2D images representing the level layers. It proposes an inter-temporal SLAM with these layers to bound the error and reduce the uncertainty along the trajectory. In [274], only based on the proprioceptive measurements, it proves loop closure events with contractors in a set-membership context. Based on this guaranteed loop closure events, a contractor based on the altimetric measurement is proposed in order to reduce the error.

For detailed reviews on underwater bathymetric-based navigation, considering *a priori* known reference map and SLAM problem, the reader may refer to [215] and [42].

1.2.4.3 Magnetism and Gravity

Magnetism based navigation is similar to the problem of TRN The difference occurs in the data manipulated, which are here Earth's Geomagnetic field. This concept is quite new and only few works can be found. The problem has been adressed to underwater environment in [227][331]. In [272] experimental results were published. The concept of maps composed of invariant gradients of the geomagnetic field has been discussed in [306] where measuring the vertical gradient could help to reduce the errors. In [56], the authors proposed to use the magnetic field for mapping and navigation.

Analogously, Gravity Field navigation could be considered [191] by measuring the local gravity field using gradiometers but for now it is highly expensive to make the distinction between the vehicle acceleration, Coriolis acceleration and the gravity itself.

1.2.4.4 How to map the environment?

As previously stated, environments could be featureless such as bathymetric maps or feature-based such as landmark, segments,... An environment composed of geometric structures such as points, lines, parametric curves are called structured environment. Man-made environment is a meaningful example of structured environment. On the contrary, unstructured environment is composed of random shapes that cannot be parametrized such as the underwater environment. It is still possible to approximate a rock as a punctual landmark but a sand ripple region for instance is usually defined by a random contour.

An environment can be described by topological or metric maps according to [307].

A topological map is a simplified map where unnecessary information are removed and only vital information remains. Scales and distances are not considered such as on the map of a tube in London or Paris.

Metric maps considers the coordinates of the objects in the environment allowing the computation of distances and angles. Two version can be considered [309]: feature-based and location-based maps.

Feature-based maps
Feature-based maps are composed of elements described by features such as the cartesian coordinates, a pixel surface[282], a parametric structure such as a line or any other parametric shapes [10]. In the SLAM context, these features are usually stored in a state augmented space [223].

Location-based maps
In location-based maps, the space is covered by labels. A famous location-based map representation is called occupancy map [86]. In general the occupancy maps are represented by a regular grid and then an occupancy value (label) is affected to each cell that informs the presence or not of an obstacle. This value can be a Boolean number (0 or 1) or a probability of presence. Handling the sensor and the pose uncertainties in grid-based methods is the most challenging problem when updating the grid. For example, exteroceptive sensors that partially observe the environment such as cameras or sonars, have to be correctly modelled. Recently, the notion of *shape* has been introduced [71]. A shape is a defined as a subset of $\mathbb{R}^q$ where q is smaller than the dimension of the workspace and corresponds to a part of the environment that is seen by some sensors. Shape intervals are defined to take into account the uncertain "contour". In conclusion, a shape separates the environment into two complementary parts that can be for example the space free of obstacles and the space including obstacles or in the underwater context it could the space defined by sand ripples and the complementary part that does not contain sand ripples.

1.2.5 Spreadly embedded sensors for underwater environment

This subsection provides a brief overview of the sensors usually embedded in underwater applications with some performance and prices. Some previously introduced sensors are also sum up in the following table. The values are given based on the analysis of different commercial products.

Name	Description	Performance	Cost
3-axis Compass	Compass provides bounded heading reference. Gyrocompass are sometimes used in marine applications.	Accuracy: $\approx 2°$	$\approx$ 100 €
Pressure sensor	Underwater depth	Accuracy: $\leq$ 10 cm	$\approx$ 100 €
DVL	Velocity of the vehicle relative to the seabed, it determines the surge, the sway and the heave velocities.	Accuracy: 0.3 to 0.8 cm/s	$\approx$ 30 K€
GPS	Surface GPS fix	Accuracy: Commercial off-the-shelf $\approx$ 10 m, DGPS $\approx$ 0.3 to 10 m and RTK 0.05 to 0.5m	from hundreds to thousands €
IMU	Vehicle's orientation, velocity and gravitational forces. • Gyroscopes: angular rates (presence of drift in Euler angles) • Accelerometer: proper acceleration relative to free fall	• Gyroscopes: drift can vary from $0.0001°/hr$ (ring laser) to $60°/hr$ (MEM[1]) • Accelerometer: Bias range from 0.01 mg (MEMS) to 0.001 mg (Pendulum)	from hundreds €(MEMS) to hundreds K€(ring laser, FOG[2])

Table 1.5: Spreadly used underwater sensor

1.3 Thesis context

Among the different applications in underwater environment proposed in the section 1.1, this thesis is focused on Mine Counter Measure missions called MCM missions. Mines played an important role during the two World Wars and are still intensively used today. Mines are real threats for naval navigation and maritime forces. Indeed it may affect the commercial trafic, the navigation of submarines and could prevent the invasion of an harbour in the case of a conflict. Many military operations are interested in the development of reliable solutions to detect and remove these mines. Moreover, the dangerosity and the development of new mines

[1]MicroEletroMechanical System
[2]Fibre Optic Gyroscope

(a) Tripartite class Mine Hunter Andromede (M643)

(b) Human divers setting up a charge for mine removal

Figure 1.15: Mine hunting operations

demand an increasing effort of developing new solutions to "hunt" these mines. Mine hunting consists in detecting all the mines in a search field, geolocalized each of them and finally destroy them. The detection phase in the mine hunting process was previously conducted by surface ships called *mine hunters*, as depicted in Figure 1.15, where the design of the boats was special. Indeed, to avoid triggering any mines, the surface vessels were built with low acoustic and magnetic signatures. Moreover the *mine hunter* is equipped with specific sonar systems called Hull-Mounted Sonar. Side-looking sonar, such as sidescan sonar, are embedded to cover fastly a large area of the seabed and to build an image of the seabed. Onboard, a human expert analyses the image to detect some potential dangerous objects. The signature of an object (potentially a mine) in the sonar images is specific and will be discussed in Chapter 2. When an object is classified by the operator as a potential dangerous object, it is necessary to revisit this object to identify it. This identification phase was usually conducted by human divers, as depicted in Figure 1.15, and finally the object is neutralized if it is really a mine. These different phases are sum up in Figure 1.16 and gives an overview of what is called a MCM mission.

The different steps are described below:

- Survey Mission Planning: it consists in planning a mission to cover an area and having a high level of area coverage.
- Data Acquisition: during the survey, the exteroceptive sensors collect data and store them in a memory.
- Mosaicking and Stitching: usually the data collected are exploited after the survey mission to transform the raw data in exploitable data. This step includes preprocessing step and georeferencing of the data (mosaicking).
- Mine detection/Object classification: An operator or an algorithm analyzes the images to detect potential dangerous object (mine) and classify them as Mine Like Contact (MILCO) or NON MILCO.
- Re-acquisition and Re-identification: when a object is classified as MILCO, the determination as to whether the object is a mine or a non-mine can only be done through visual identification. This phase can be conducted by a human diver or a robot.
- Mine disposal: when the object is identified as a mine, the mine has to be removed.

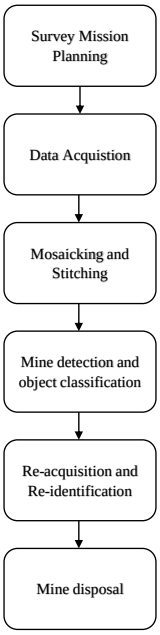


Figure 1.16: MCM mission overview

The increasing development in underwater robotics enables today the use of AUVs to perform the survey mission. An example of such AUV was presented in Figure 1.10 with the embedded sensors presented in Table 1.2. In general, these AUVs are equipped with sidescan sonars or SAS to collect high resolution sonar images to cover a large area on the seabed. These AUVs can benefit of a reliable navigation system based on a INS coupled with a DVL that drifts a little over time. The *boustrophédon* pattern, usually called lawn-mowing pattern [54], is used to cover optimally an area of search with side-looking sonars. The space between the tracks depends on the range of the side-looking sonars. Usually the distance inter-tracks is lower than the range of the sonar to have an overlap on two consecutive tracks. Different methods are proposed in the coverage path planning domain [2][1] to propose a complete coverage. However, as stated in [54], the coverage will be complete according to the proposed motion planning but not in the operating research field. In [244], an adaptive online coverage planning algorithm is proposed based on the actual coverage area that accounts for pose uncertainty of the AUV. The certainly explored area of a drifting AUV equipped with a side-looking sonar is computed in [73] in a guaranteed manner based on a constraint satisfaction problem.

Once the data are collected, some automatic algorithms are able to detect the potential dangerous objects. These algorithms are called Automatic Target Recognition (ATR). This topic will be treated in Chapter 2. These detections reveals the presence of a suspicious object. For each of these suspicious objects, seen once or more times (multi-view), a classification algorithm outputs the tested object as MILCO or NON MILCO. If it looks like a mine, the object has to be revisited to identify it. This phase is called "Re-acquisition/identification". If the revisited object is actually a mine, it has to be removed. This phase is called "Mine disposal". These two tasks were usually conducted by human divers as precised earlier but it is intended to be replaced by AUVs. Indeed it faces many constraints such as the poor visibility, the diver disorientation, the duration of the actual diver search, the time to find a search team, the securing of the ship, many time consuming tasks that can be handled easily with an underwater

robot. Moreover this task is not free of risk for human divers since they have to approach at the maximum the mine. Due to the need to return to site, these two tasks are called "Revisit phase".

Due to the development of the autonomy of underwater robots, a fleet of heterogeneous robots is proposed in [207] to conduct a complete MCM mission and to remove more and more the necessity of an expert human in the decision making. From the detection to the neutralization phase, a collaboration between the different robots is vital. In uncertain environment, as the underwater environment, the decision making to achieve a goal (neutralization) with cooperating heteregeneous vehicles that require communication to perform complete autonomous mission is not trivial [14].

1.3.1 The revisit concept

Many applications can be interested in a revisit of some areas of interest. For example, in [340], a study about the long-term effects of climate change and human activities on the benthos is conducted. The benthos designates the organisms living on or near the seabed. By collecting observations of the same place with AUVs at regular temporal intervals, it provides the ecological data for the study. In the context of MCM missions, revisiting a potential dangerous object may help to the classification process by getting other aspects of the targets. Indeed, in [21][20], a reacquire-identify (RID) pattern inspired by the *boustrophédon* pattern is proposed to revisit a set of geolocalized targets with a side-looking sonar. This pattern is defined to reacquire the targets at different angles of view called aspects to increase the probability of detection [343] and the classification performance [262][229]. The results were then used [233] to determine a safe naval transit. Similarly, in [61], a route optimization algorithm is proposed to get additional looks in MCM missions that takes into account the motion and the imaging constraints. The problem is reduced to a travelling salesman problem (minimize the travelling distance) with desired aspects of some targets and Dubins paths to model the non-holonomic constraint of the AUV. In the presence of water currents, in [337], it proposes to adapt the AUV survey heading to ensure quality SAS data [335] and to adapt its route just after the survey pattern with the remaining battery to reinspect some suspicious objects detected during the survey at additional aspects.

However in the methods listed above, the uncertainty on the pose of the robot is not taken into account in the planning process as the AUVs rely on high grade navigational sensors such as an expensive INS coupled with a DVL.

This thesis addresses the difficulty to revisit a particular target in the context of MCM mission with a "low" cost robot usually called "sensor-limited" vehicle [65]. Due to many cost limitations and strategic reasons explained in this specific mine removal context, the AUV is only equipped with low grade navigational sensor meaning the absence of an INS and a DVL. Moreover it is equipped with a lower resoluted imaging exteroceptive sensor such as a FLS or a camera to reacquire/identify the target. An altimeter can be used to use the bathymetric profile of the seabed to follow isobaths [139] or for localization purpose [213] for example. A depth sensor is obvisouly embedded. In this MCM context, the problem of reacquisition/identification in the underwater environment has been solved with heavy multi-hypotheses data association [93] but it requires to send the AUV in a features field environment to perform the data association of uncertain located punctual objects and finally to control the heading of the vehicle to go to the target. Wouldn't it be possible to follow a particular strategy to improve the data association and reduce its computational complexity ? Based on the different sensors embedded

in the vehicle and the map of the environment, may it be possible to propose a strategy in underwater environment to guarantee the *a priori* revisit of the desired target ? These questions highlight the interest developed in the thesis. Obviously this problem is not only dedicated to MCM missions and can be exploited to other missions that require a revisit of a previous surveyed area. This environment is said to be perfectly or partially known environment. The solution proposed through this thesis takes into account the difficulties of the heterogeneous means of detection during a survey and a revisit mission in the underwater environment, and the difficulties to find a motion planning strategy for a low grade navigational robot that has to rely on exteroceptive measurements to relocate itself. Moreover, for clandestine missions and some operational constraints, the AUV could dive far from the desired target locations. An overview of the solution proposed is detailed in the following section.

1.3.2 System overview

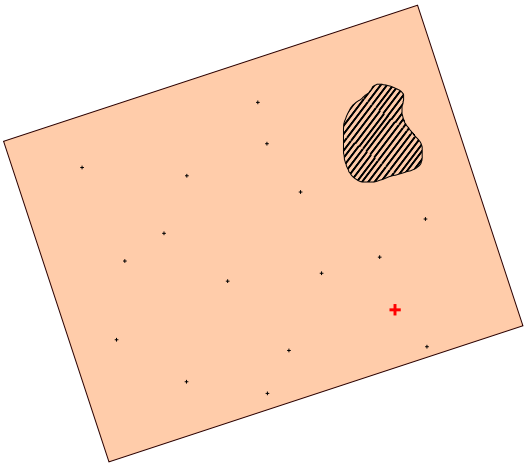
The MCM task is globally decomposed into two main operations: a survey mission and a revisit mission. Figure 1.17 depicts the solution adopted and will be detailed by making references to the Chapters where it will be discussed. The different steps of the process have already been basically explained but it provides an overall aspect of the project.

1.3.2.1 Survey mission

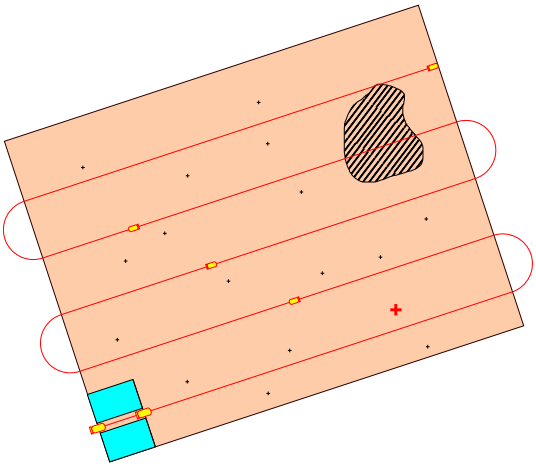
In the context of MCM mission, Figure 1.17(a) depicts an environment that has to be surveyed to detect any potential dangerous objects, called mines. An AUV conducts an autonomous mission to collect data in this area following a *boustrophédon* pattern or any coverage pattern 1.17(b). The diagram 1.18 depicts the steps to build a data base of the information gathered in this surveyed area. This data base is said to be "adapted" to the survey system since it enabled to collect the data with its own exteroceptive system. The part surrounded in blue will be the topic of the Chapter 2. From the data base, a "contact" is classified as MILCO and needs to be revisited. It will be called the "target".

1.3.2.2 Revisit mission

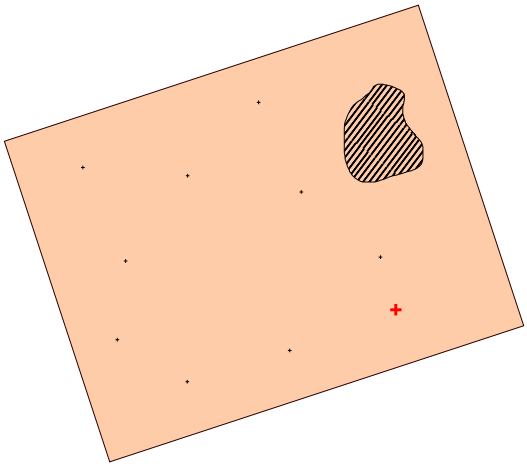
Based on the parameters of the revisit system such as the altitude of navigation of the AUV, its exteroceptive sensors, its velocity, its manoeuvrability, etc... the data base created with the survey system is adapted to the revisit system. This concept is depicted in Figure 1.17(c) where some landmarks have been removed due to the inability to detect them with the revisit system. The diagram 1.19 depicts the process of the data base adaptation, discussed in Chapter 2 and the creation of the registration maps, discussed in Chapter 5. These registration maps will be used by a motion planner in Chapter 6 where it finds an high level strategy taking into account the navigation parameters of the AUV. Figure 1.20 shows the different steps. It may be possible that no path is found. Two strategies are proposed in Figure 1.17(d) and 1.17(e) where any shaped landmarks can be considered.



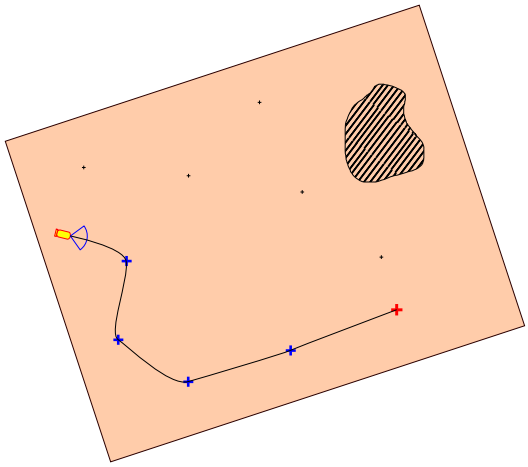
(a) Environment composed of landmarks such as rocks (punctual landmark) and a sand ripples area (2D landmark), and a mine (red cross)



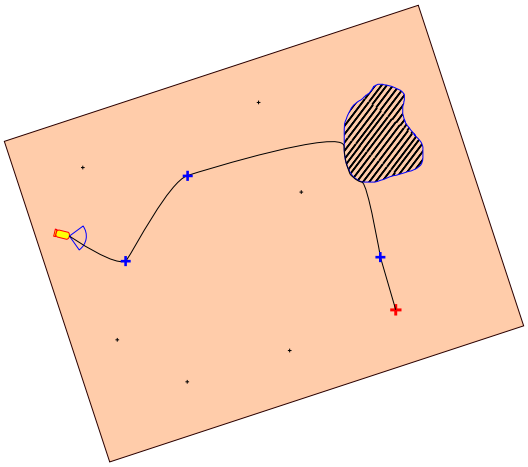
(b) *Boustrophédon* survey of the area with SSS and detections. Cyan blue area depicts the explored area between two configurations of the AUV along a straight path.



(c) Filtering to keep the landmark detectable by the revisit system



(d) Guaranteed revisit strategy 1. The blue pie shows the ensonified area of the exteroceptive sensor.



(e) Guaranteed revisit strategy 2. The blue pie shows the ensonified area of the exteroceptive sensor.

Figure 1.17: Overview of the solution proposed

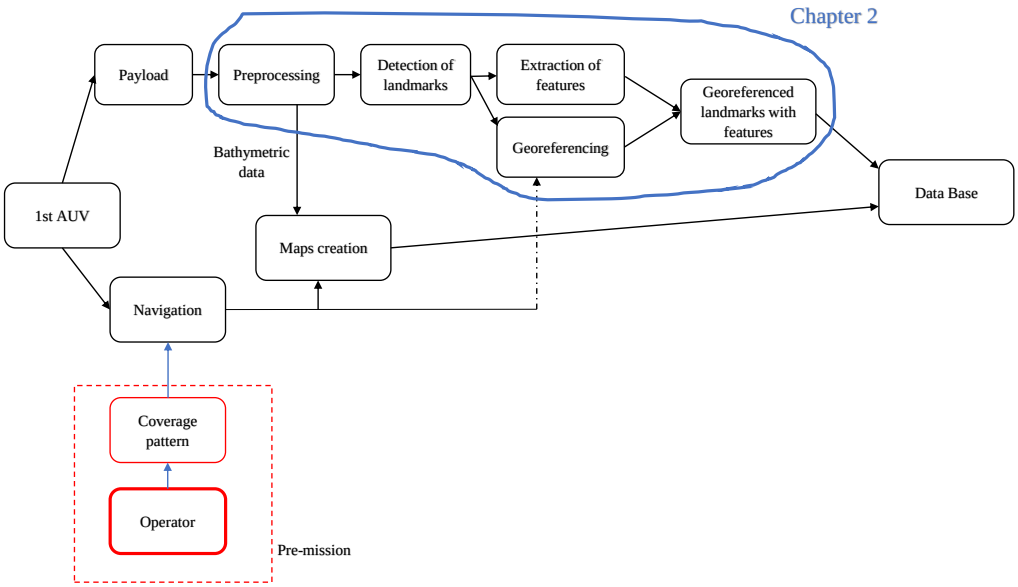


Figure 1.18: Diagram of the survey mission

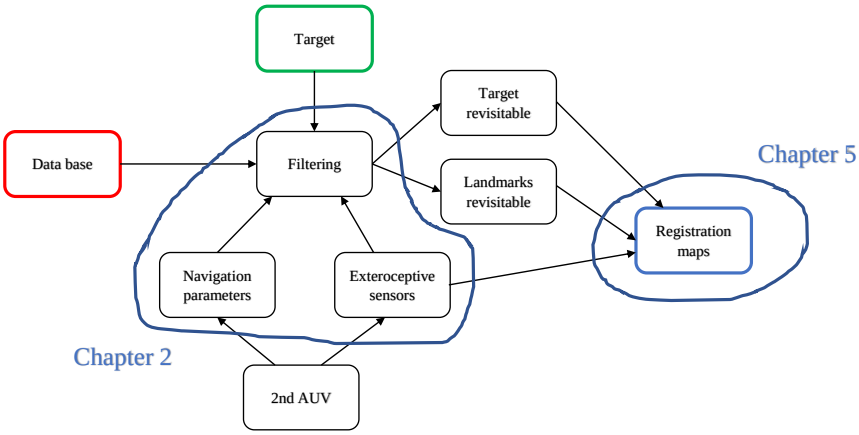


Figure 1.19: Diagram of the adaptation to the revisit system

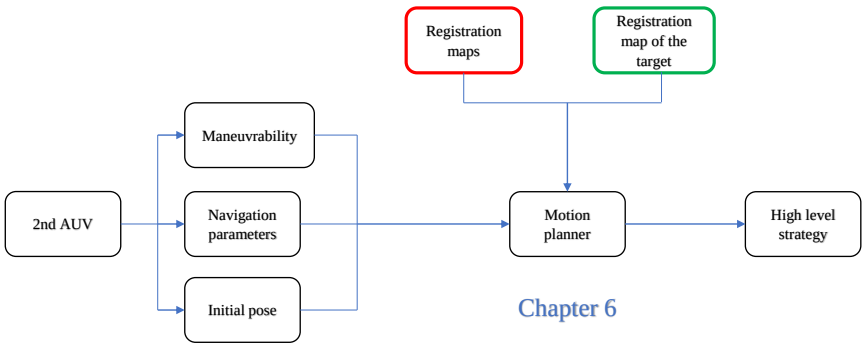


Figure 1.20: Diagram of the high level strategy finding

1.4 Document roadmap

In this introduction (**Chapter 1**), the underwater environment and its difficulties have been presented. The unstructured aspect, presence of outliers (fishs,...) and the absence of absolute positioning system (GPS) make it a difficult environment for an autonomous robot to reliably navigate and localize itself. After presenting different types of mission with various robots, it provides different ways for localization and navigation. Among them, the geophysical navigation seems to be of great interest since it enables to perform long-term and long-range mission without surfacing for an AUV with the aid of an exteroceptive sensor to perceive the environment. In the context of MCM mission, the neutralization of a mine is vital and it is necessary to renavigate the environment to recover the mine previously detected. Renavigating is a challenging task for a sensor-limited vehicle and the solution adopted in this thesis was roughly explained with illustrations.

The remainder of the thesis is organized as follows:

Chapter 2
This chapter provides the knowledge of sonar imagery, one of the most used sensor in underwater environment due to its capacity to work in dark area and at longer range than optical sensors. Depending on the characteristics of the sensor, it can image the seabed to provide information about potential dangerous objects such as mines. When using different exteroceptive sensors during different missions, the extracted information are not the same. Indeed the features extracted may not be detected by another sensor. Therefore, based on the physical properties, a filtering step is proposed to keep only the most likely detectable features according to the sensor used.

Chapter 3
This chapter provides an overview of the state of the art on motion planning techniques, with and without uncertainties. Some applications on our problem are proposed with the state of the art algorithms proposed.

Chapter 4
This chapter provides the set-membership tools that are used in this thesis. The set-membership approach appears as a guarantee alternative to Bayesian tools. It has already been used for localization but it has not been widely used for motion planning under uncertainty.

Chapter 5
This chapter introduces the first contribution of the thesis which is the definition of the relocation area based on the sensor used and the characteristics of the landmark observed. Based on the set-membership tools it provides the set of robot position able to detect any landmark considering the visibility area of a sensor-limited vehicle based on the range and/or the opening (aperture) angle.

Chapter 6
This chapter presents the second contribution which is a motion planner under uncertainty based on interval analysis and the relocation areas introduced in Chapter 5. It proposes an approach to guarantee the reachability of a target geolocalized in the environment, which is detectable by the sensor embedded. Based on the known position of other landmarks, the algorithm finds a sequence of intermediate revisits enabling relocation process before reaching for sure the target. It constructs an hyper-graph to find the links between the different sets

defining the relocation areas. The path sequence proposed minimizes a cost function.

Chapter 7

The last chapter summarizes the contributions of this thesis and the remaining issues that need to be solved for future applications. Moreover it proposes some perspectives.

Chapter 2

Sonar imagery

2.1 Introduction

Sonar is the acronym of *SOund NAvigation and Ranging* and uses the propagation of a mechanical wave, called acoustics wave, to transmit and receive information in underwater environment. The acoustics propagation is four-five times faster than in the air. Contrary to the strong attenuation of the electromagnetic waves, such as the light, enhancing a limited use of optical system to navigate in underwater environment, acoustics system proved to be well adapted in this context to navigate and transfer information (emit and receive). It enables to improve the knowledge in the underwater environment with two types of sonar: passive and active sonar. A passive sonar doesn't emit any sound wave, it is only listening the environment. It could be used to detect enemy vessels for the surveillance for example or could be used to study the underwater fauna and flora. Acoustics is the main mean of communication for the dolphins and the whales [316]. On the contrary an active sonar emits a signal and waits for the reflection on the target. In both cases, an antenna enables the reception of the signal, and then the sonar data are processed to detect particularities or to form an image. Due to the improvement of technologies, sonar systems are used in many applications, as introduced in Chapter 1, from scientific to commercial operations including obviously military operations.

In this thesis, a focus on the sidescan sonar, usually denoted as SSS, and Forward Looking Sonar (FLS) is presented. Indeed, due to its high coverage, the sidescan sonar can provide high resoluted image of the seabed by following a straight path to detect some potential dangerous objects called mines. In the context of Mine Counter Measure (MCM) missions it is vital to detect any mines to guarantee the safety of surface vessels and to enable military operations. As presented in Chapter 1, FLS provides directly an acoustic image of the seabed and is particularly used in the revisit phase in the context for re-acquisition/identification purpose with optical sensors. Due to the limitations of these latters, the FLS enables to renavigate this challenging environment and to perform the localization task before approaching the mine for identification with the camera. Both sonar systems provide an acoustic image, i.e. 2D image, of the seabed coming from the backscattered signal from the seabed. This representation makes the interpretation of the seabed easier, and can reveal the presence of protruding objects above the seabed such as mines.

In this chapter, the general principles of imaging sonars will be briefly explained. Then a focus on the sidescan imagery and FLS imagery will provide the knowledge required to understand the filtering step explained in the process of adaptation of the data between the survey mission and the revisit mission.

2.2 General principles of imaging sonar

2.2.1 From emission to reception...

An imaging sonar is an active sonar, that means it can emit and receive a signal. The information is carried by sound waves that are able to propagate in underwater environment.

The architecture of a sonar is composed of different modules:

- Emission module: generate an electrical signal that is intended to be propagated in the underwater environment.
- Antenna module: transform the electrical signal into a sound wave with specific characteristics of the antenna such as the directivity and transform the acoustics signal received back into an electrical signal.
- Reception module: some processing before the storage of the exploitable data.

Emission

A pulse is generally generated at a frequency called f_0 ranging from 300 to 900 kHz in the case of imaging sonars [32]. The emitted signal is also determined by the duration of the pulse T and a bandwidth B (frequency spread). Two types of signals are commonly used: a monochromatic pulse, referring to gated Continuous Wave (CW) pulse, and a chirp which is modulated in frequency. These parameters have direct consequences on the performance of the sonar. Indeed the range of the sonar is determined by the pulse frequency according to Table 2.1. These values are computed based on the sonar equation. In the case of an active sonar, the equation from an energetic point of view is:

$$SL - 2TL + TS - NL + DI + PG > DT \tag{2.1}$$

which means that the SNR (signal noise ratio) is above a threshold.

In this equation, the following terms are defined as follows:

- SL (Source Level): the power generated by the active sonar.
- NL (Noise Level): the noise of the system including the noise of the seabed.
- DI (Directivity Index): the antenna defined by its directivity.
- PG (Processing Gain): the processing gain.
- DT (Detection Threshold): the threshold enabling the detection.
- TL (Transmission Loss): the absorption of the energy by the underwater environment and the geometric divergence.
- TS (Targer Strengh): the echo power of the target.

Frequency	**Range**
10 kHz	>10 km
50 kHz	5 km
100 kHz	1000 m
500 kHz	150 m
1 MHz	50 m

Table 2.1: Maximal range of detection depending on the frequency.

Since the sound wave is propagating, its energy may be decreased with the distance to source: it is called the geometric divergence. Moreover the sound wave energy is absorbed by the underwater environment. This decreasing energy is known as the transmission loss TL expressed by:

$$TL = 20log_{10}(r) + \alpha \cdot r \tag{2.2}$$

where r is the range and α the coefficient of absorption of the environment.

The higher the frequency is, the more details will be observed in the image. However, the range of the sonar will be reduced due to the coefficient of the absorption α that appears in the transmission loss (TL). This coefficient increases approximately with the square of the frequency of the sonar. It is estimated at 30 dB/km at the frequency of 100 kHz enabling a range detection at 1000 m. At 1 MHz, this coefficient is at 500 dB/km reducing the maximum range detection at 50 m. The transmission loss (TL) is expressed as $2TL$ in the equation 2.1 due to the two-way travel.

The electrical signal generated by the waveform generator, modulated and amplified is then converted in the antenna module into an acoustic wave to propagate inside the underwater environment.

Antenna

The antenna is composed of electroacoustic transducers that can convert an electric energy into an acoustic energy and vice-versa. The module is in a mode called "emission" when the electric signal generated will be transmitted to the underwater environment. The emitted pulses are directed in a slant direction toward the seabed.

An antenna can be represented by two angles as shown in Figure 2.1:

- an azimuth angle θ: the angle between the axis of the propagation and an horizontal reference axis.
- an elevation angle ϕ: the angle between the axis of the propagation and a vertical reference axis.

The performance of an antenna is represented by the radiation pattern, also called antenna pattern or far-field pattern. It refers to the directivity index (DI) which can be defined by a directivity function $D(\theta, \phi)$ at a particular frequency. This function computes the energetic response in the far field of the transducer in function of the orientation. It is normalized by the maximum of the function. This representation, as depicted in Figure 2.2, shows a main lobe and side lobes. The main lobe is characterized by its aperture at -3 dB which corresponds to the width of the beam at $D(\theta, \phi) = 0.5$. This width along the θ axis is denoted as $2\theta_3$ and is called the azimuth aperture angle or horizontal beam-width. Similarly the width along the ϕ axis is denoted as $2\phi_3$ and is called the elevation aperture angle or the vertical beam-width. The higher the frequency is, the narrower the width will be but the higher the level of the side lobes will be too. These side lobes disrupt the measurements. Depending on the application, the frequency will have to be well adjusted based on the aperture angles desired and the level of the side lobes.

The formation of the sonar image consists in registering the backscattered acoustic waves in a slant direction. The dimension of the pixel depends on the sonar characteristics.

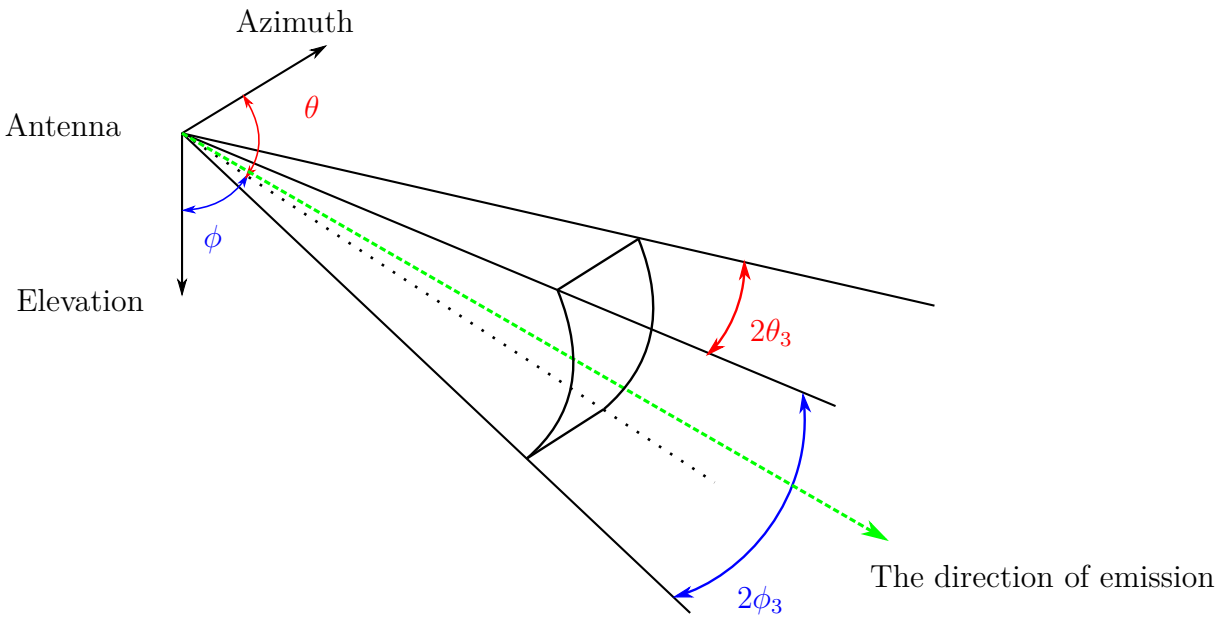


Figure 2.1: Geometry of a punctual antenna.

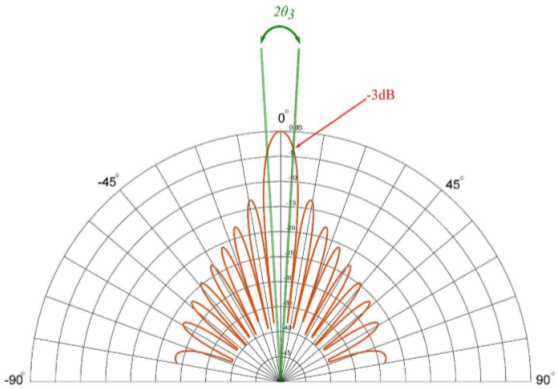


Figure 2.2: Example of antenna pattern in polar coordinates in dB around the 0° direction for the azimuth angle. Extracted from [254].

The azimuth resolution, known as angular resolution, corresponds to the minimal distance between two objects to see them separately on the backscattered signal. This resolution is defined as:

$$dx = 2r\theta_3 = \frac{r\lambda}{La} = \frac{rc}{f_0 La} \tag{2.3}$$

where r is the distance to sonar, $2\theta_3$ is the aperture of the antenna, La is the length of the antenna, c is the sound velocity in the water (approximately 1500 m/s) and λ is the wave length at the central frequency f_0 of the sonar. The angular resolution depends on the distance to sonar.

The range resolution, known as slant range resolution due to the slant range acquisition, corresponds to the minimal distance along the propagation direction between two objects to see them separately on the backscattered signal. This resolution is defined in the slant range direction as:

$$dr = \frac{c}{2B} \tag{2.4}$$

where B is the bandwidth of the signal and c is the sound velocity in the water.

The resolution of a pixel in the slant range direction at the distance r is then $dx \times dr$. It is possible to know the surface of the seabed highlighted by the sonar by projecting the dr on the seabed. This resolution, known as ground range resolution, is defined as follows:

$$dy = \frac{c}{2B\sin(\phi)} \tag{2.5}$$

where ϕ is the grazing angle.

These same transducers will convert the backscattered acoustic energy from the seabed or from any target such as mines when it will be in mode "reception". The parameters of the antenna in this mode are the same as in the "emission" mode.

Reception

When the backscattered signal is converted into an electrical signal through the transducers, the signal has to be amplified due to the transmission loss in the underwater environment. Moreover many noises are added to the signal:

- ambient noise: maritime trafic, surface trouble, rain, thermal agitation,...
- reverberation: it is the sum of the multiple backscattering contributions such as the volume, the surface (non uniform surface) and the seabed (not flat seabed), etc...
- multi-path: the backscatter signal from the seabed and the surface.
- electronic: electronic processing.
- vehicle carrying the sonar: motion, propeller,...

After the amplification, many systems propose a TVG (Time Varying Gain) correction to reduce the dynamic of the signal by compensating the transmission loss at the different ranges since the dependence in range according to the equation 2.2.

Finally the signal is digitized to be treated for any image processing such as detections. This step depends on a spatial and temporal sampling during the beamforming which defines the

real resolution of the output sonar image. Indeed, in general, the dimension of the pixel in the output image is smaller than the resolutions described before in Equation 2.3 and Equation 2.4 to avoid any information loss. It induces an oversampling effect on the ensonified area by the sonar but it improves the analysis of the images. The gray level of a pixel in the image indicates an amplitude which is the sum of the backscattered waves by the different elements inside the environment in this resolution cell.

In sonar systems, the antenna is composed of an array of transducers instead of an unique transducer. This enables to improve the directivity by the beamforming and then to reduce the noisy contributions from non desired directions.

All these operations enhance some changes to the raw signal which can be degraded. Moreover it can be seen in the sonar images the apparition of a noise named speckle.

2.2.2 What is the speckle noise?

When looking at a high resolution sonar image, one may notice a granular aspect of the seabed. This is representative of the so-called speckle noise. This noise is inherently present in every coherent system such as Synthetic Aperture Radar (SAR) images or in medical ultrasound images. This noise appears when an electromagnetic wave or an acoustic wave interacts with a rough surface made of many diffusers that are smaller than the wave length of the signal [112]. These diffusers reflect then a wave with the same wave length but with a random amplitude, phase and direction. The transducers register the sum of all the different contributions including these interferences coming from these specific diffusers inside the same resolution cell. The reverberation is the combination of destructive (low amplitude compared to the mean level) and constructive (high level compared to the mean level) information provided by all coherently but randomly phase-shifted backscattered waves due to the roughness of the seabed (smaller size than the wave length).

Different statistical modelisations of this phenomena have been proposed. The most famous statistical distribution in sonar images is the Rayleigh distribution. It is widely used to model the link between the seabed, the sonar characteristics and the distribution of the gray levels in a sonar image under some hypotheses. It has been proved under this distribution modelisation of the reverberation inside a same cell resolution that the speckle noise appears as a multiplicative noise [315]. This modelisation performs well when the size of the roughness is smaller than the resolution cell. Other modelisations are proposed such as the K distribution or the Weibull distribution when the seabed is composed of rocks or sand ripples which have similar and greater dimension than the resolution cells [188].

All these statistical distributions enable to model the possible distribution of the gray levels in the sonar images.

2.2.3 Presence of features

Features such as objects can be revealed in sonar images if their size are greater than the azimuth and range resolutions. In the sonar image, an acoustic shadow indicates the presence of an object protruding from the seabed as can be seen in Figure 2.3. This shadow is sometimes more relevant than the direct echo of the object due to the complexity of the acoustic backscattering from this object.

The presence or the lack of seabed reverberation may delimit the shadow area associated to

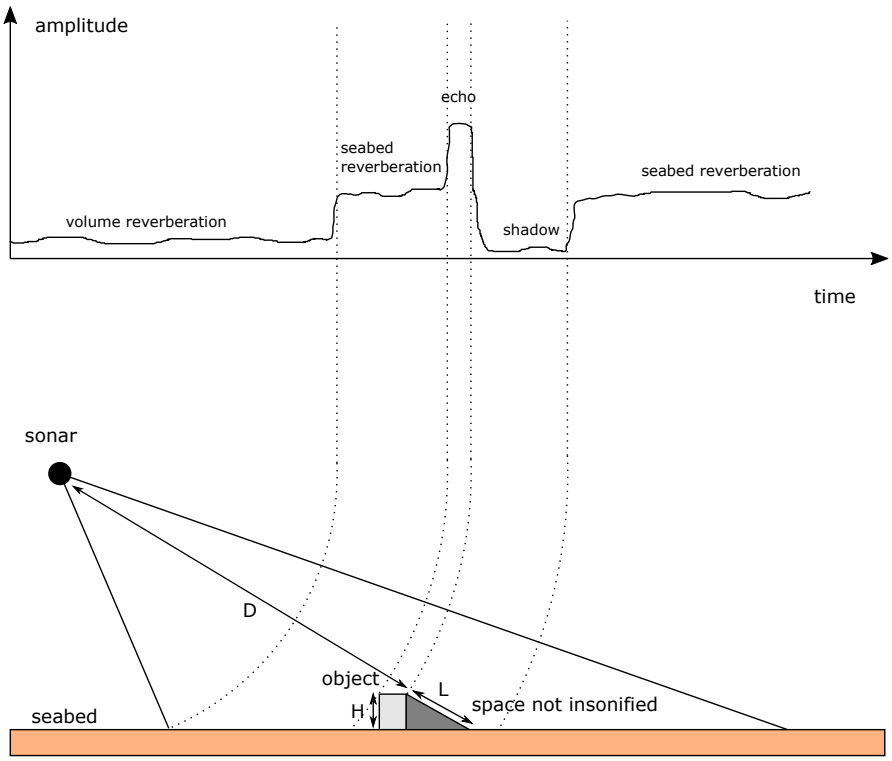


Figure 2.3: Received sonar signal for one pulse emission in the vertical plane.

the object. In Figure 2.3, one may see different amplitudes of the signal in the received time [32]:

- a low amplitude due to the volume reverberation (called the water column).
- a mean level corresponding to the seabed reverberation.
- a strong level revealing the echo of the object.
- a low level due to the shadow area of the object, it corresponds to volume reverberation, multipath, ambient noise,...
- a mean level corresponding to the seabed.

The backscattered level of the object (echo) depends on the shape of the object and its composition. Indeed the different materials do not react samely to the acoustic waves and if the object has an ensonified orientation surface perpendicular to the sonar, the received signal will be stronger. The rougher the surface of the object is, the more isotropic the direction of the backscattered signal on this object will be. As can be seen in Figure 2.4, the shadow of the object may indicate the height of the object, or at least the protruding height from the seabed, with the following expression:

$$h = \frac{HL}{L + D} \tag{2.6}$$

where H is the altitude of the sonar above the seabed, D the distance to the beginning of the shadow in the image (approximately the slantrange to the object) and L the length of the shadow.

The higher the object is, the longer the shadow in the sonar image will be due to the grazing aspect of the sonar.

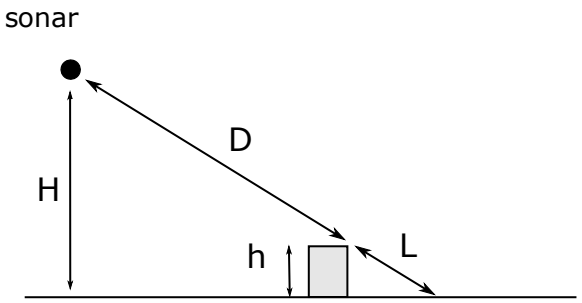


Figure 2.4: Height of the object based on the length of the shadow.

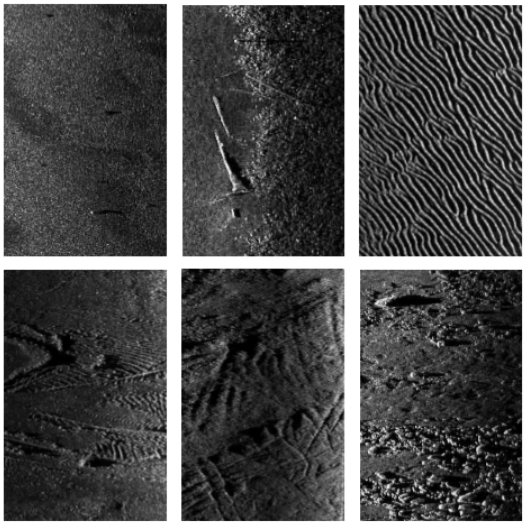


Figure 2.5: Mosaics of different seabed types. One may see sandy area, sand ripples, rocks, dredging activities or objects protruded from the seabed. Pictures extracted from [254].

In the past, the resolution of the sonar did not enable to detect correctly an echo. The classification of the objects, such as for the mine as Mine Like Contact (MILCO) or non-MILCO, was only based on the shadow of this object which is bigger due to the acquisition geometry. Nowadays, high resolution sonars are available on the market such as Synthetic Aperture Sonar (SAS) which provides very high resolution images.

The amplitude of the signal globally decreases with the range. Indeed it comes from the transmission loss, the absorption of the acoustic wave in the underwater environment and the grazing angle.

The analysis of the backscattered signal above indicates the presence of an object that may appear punctual due to the representation in the vertical axis but in reality many features can be seen in sonar, in both sidescan sonar images and FLS images. Many shaped or shapeless features can be observed on sonar images based on this principle as depicted in Figure 2.5. Sandy area is observable on the top-left image where it only appears noise due to the presence of thin sand. On the contrary, on the top-right figure, one may see some sand ripples with a geometric pattern. Textures may appear on high resolution sonar images revealing the type of seabed. However it is difficult to give a definition [125]. A texture can be described by a region in the image with a variable intensity and a repeated pattern based on a certain random level. These textures depend on the physical properties of the seabed that may vary in function of the meteorological conditions, seasons or geographical places.

In conclusion, depending on the acquisition parameters of the sonar, different types of information can be extracted from the seabed. These information can be classified into the following categories:

- energetic: the backscattered signal.
- structure: presence of structures such as rocks, sand ripples, objects,...
- geometric: direction and orientation of sand ripples...
- multi-scale: different size of the structures.

Moreover the size of the objects detectable by a sonar system are directly linked to the sonar parameters and the geometry of the acquisition.

Now a focus on particular imaging sonars will be provided such as the sidescan sonar acquisition and geometry. Then a short presentation of the 2D FLS will be introduced.

2.3 Particular imaging sonars

In the previous section, a general introduction about acoustic imagery has been proposed. It mainly introduced the basic knowledge about sonars, the cell resolution and what can be seen in sonar images depending on the size of the features. Now the sidescan sonar and the FLS will be briefly presented to understand the difference of acquisition geometry.

2.3.1 Sidescan Sonar (SSS)

2.3.1.1 Presentation

As previously presented in Chapter 1, the SideScan Sonar (SSS) is a specific imaging sonar well adapted to cover a large area of the seabed. In general, the system is composed of two antennas, each positioned on the sides of the vehicle embedding the sonar. It could be towed by a vehicle, embedded in an Autonomous Underwater Vehicle (AUV), ship-hull mounted in a boat... This system is called side-looking due to the geometry of acquisition as depicted in Figure 2.6. This sonar is scanning the area perpendicularly to the motion direction of the vehicle. The elevation aperture, in the vertical plane, is usually large to cover a large area sideways and the azimuth aperture, in the horizontal plane, is narrow to obtain fine resolutions. Two images result from the sonar processing, a starboard and a port image. The process will be explained in the following subsection.

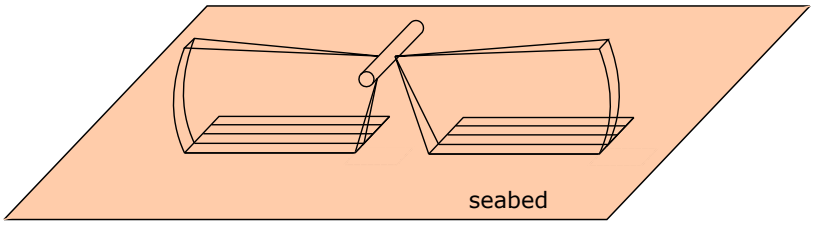


Figure 2.6: Geometry of data acquisition for a sidescan sonar (SSS).

2.3.1.2 Image formation

At every pulse emission, due to the narrow horizontal aperture, the signal received can be considered as 1D signal as depicted in Figure 2.3. This signal is received by both antennas: starboard and port antenna, that means there are actually 2 signals for one pulse emission. A pulse emission is called a ping. This scanning process can be considered as a sweep exploration [73]. This narrow acoustic beam illuminates only a narrow stripe of the seabed. By considering a straight motion of the boat or the AUV, at every ping emission, a narrow stripe of the seabed is covered and by concatenating these successive stripes, a wide area of the seabed is covered. As previously presented, the main parameters affecting the range resolution called across track resolution is the length of the antenna and the frequency of the acoustic beams. The along track resolution (azimuth) depends on the speed of the platform and on the ping emission frequency. Traditionally the data coming from the starboard and the port antenna are treated simultaneously. The general representation of a sidescan sonar image is a 2D image with port and starboard images gathered. Every row in the image corresponds to one ping emission (it depends on the motion) and every column corresponds to the received signal in the time. That means at every ping emission, a row is added to the image, and the columns of this row correspond to the time sampling of the received signal according to the 1D representation in Figure 2.3. Notice that this representation is not in Cartesian coordinates but in time coordinates. In such images, the vertical axis corresponds then to the time at which the beam (or ping) was emitted from the sonar and the horizontal axis corresponds to the time of flight of the pulse in the across track direction (perpendicular to the direction of the motion). This representation is called a water-fall image and an example is depicted in Figure 2.7. The image is getting darker on both sides starting from the middle due to the transmission loss. The TVG (Time Varying Gain) correction has not been applied. The narrow dark band in the middle corresponds to the water column which depends on the altitude, the roll of the platform, and on the direction of the antennas. A box containing an object is represented in red in the port image, the shadow is on the left as it comes after the echo. The null temporal reference along the samples is in the middle of the image due to the concatenation of the port and starboard images.

Moreover some artefacts that can be corrected are inherent to sonar images and will be preprocessed such as TVG correction before any image segmentation or classification.

2.3.1.3 Image preprocessing

In [29], three stages are proposed to convert raw sonar data into usable images:

- preprocessing: preparation of the raw sonar data by the cleaning of the navigation and the sensor's attitude, and the conversion between formats.
- processing: transformation to usable images or grids, with radiometric and geometric corrections. Grid interpolation and mosaicking are considered too.
- post-processing: not necessary for correct sonar interpretation but can be a plus, it can be visual enhancement and speckle removal for example.

These three stages is called in our case the preprocessing step before the exploitation of the sonar data for detection/classification. That means all these stages are gathered.

Some file formats to store the sonar data, the navigation and attitude of the platform are commercially proposed. Among them, the XTF (eXtended Triton Format) file format which is used in our case. It enables to record different types of sonar, navigation, and bathymetry

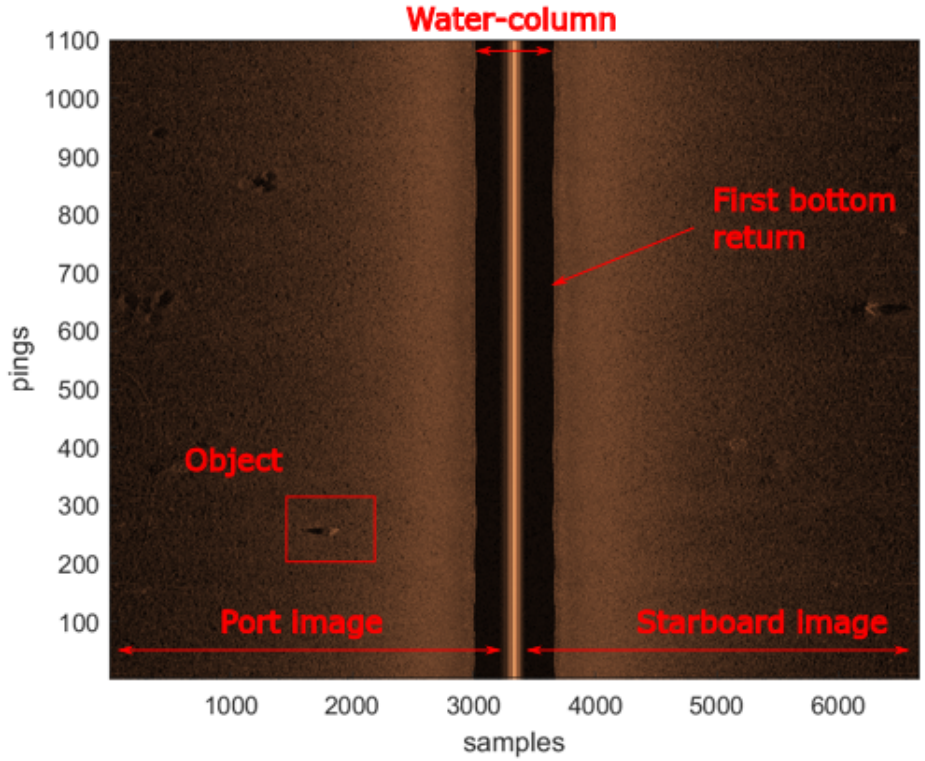


Figure 2.7: Water-fall image for a sidescan sonar.

measurements in only one file.

Firstly, the navigation and attitude information are usually filtered to remove outliers and spikes based on smoothing filters such as the Kalman filter [151] and interpolation with polynomial or spline functions.

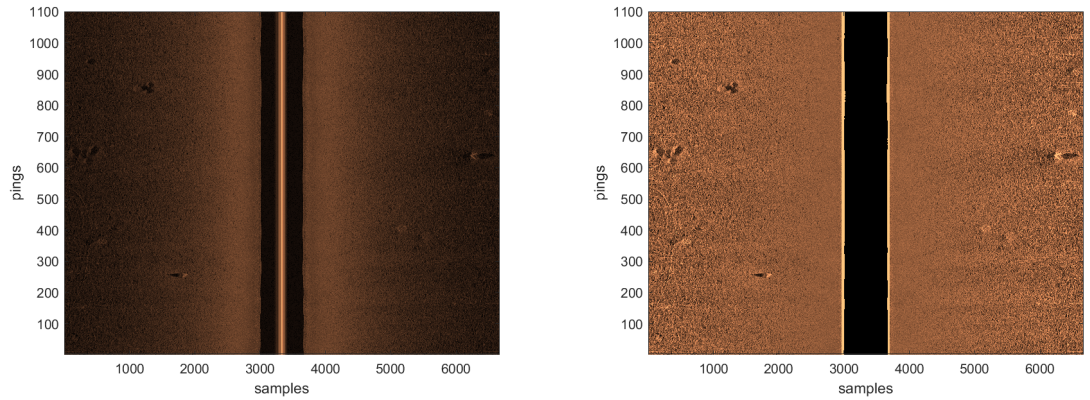
Secondly, the sonar data are processed with radiometric and geographic corrections [45][44].

The radiometric corrections refer to the following corrections:

- Requantization: it consists in adapting the output from the sonar hardware to follow a quantization schemes (8-bit quantization).
- Across-track correction: known as TVG (Time Varying Gain) to compensate the attenuation of the backscattered signal with distance.
- Along-track correction: it comes from acquisition problems within the transducer or loss of data. It results in lines shifted across track and can be compensated with averaged values of adjacent lines.

TVG correction

A TVG correction is applied to compensate the transmission loss of the acoustic energy with distance. In case of a flat seabed and a constant altitude for the acquisition platform, the range and angular dependency factors for the correction can be treated together. In [9], it is proposed to compute the average brightness on some number of pings. After removing the water column, the brightness is expressed based on the grazing angle. It provides a 1D curve with length equal to the number of samples contained in one ping. Every ping is multiplied by



(a) Original sonar image.

(b) With TVG correction.

Figure 2.8: Comparison between a non corrected (left) and a TVG corrected image (right).

the inverse of this 1D curve and its mean level to maintain the mean level of the image. The result of this TVG compensation is proposed in Figure 2.8.

Moreover the backscatter intensity varies in function of the grazing angle and the type of the seabed (sediment)[347].

In case of variable altitude of the platform or not flat seabed, it results in a variation of the width of the water column compared to the images presented in Figure 2.8. It is particularly common in shallow water and low altitude survey. Therefore the range and angular dependency should be treated separately [39]. Based on a bottom track to estimate the sensor altitude at each ping emission, a resampling strategy is proposed in [40] to compute an angular dependency correction factor. This method is quite interesting when no a priori knownledge about the sonar system is available. It can be classified as an heuristic method. In case of known acquisition parameters and environment [188], a compensation can be computed based on simple models such as the Lambert model [221].

Slant range correction

This correction comes from the slant range representation of the sonar image which is not representative of the real distance between objects on the seabed. This process consists in remapping the pixel from their apparent position to their true position by taking into account the range (elapsed receiving time) and the altitude of the platform. It results in giving the ground range of the pixels. Due to the grazing angle, the remapping is only important at short range and the error is small at far range. By assuming a flat seafloor, the ground range distance is:

$$D_i = \sqrt{(\frac{cT_i}{2})^2 - h^2} \tag{2.7}$$

where $R_i = \frac{cT_i}{2}$ is the slant range distance of the pixel i, c is the local sound speed and h is the altitude of the sonar platform. It assumes a flat seabed and some artefacts could appear in case of relief. If a bathymetry map is available, it should be taken into account.

After the slant range correction, the sonar image is corrected across-track but it still has to be corrected along-track. This process is called *anamorphis* [29]. It consists in having square pixel, which results in the same along-track and across-track resolutions, by sub-sampling or

replicating ping lines to have ground features correctly represented.

The following steps are classified as post-processing and non necessary according to [29] but it improves the interpretation of the sonar images.

Visual enhancement

The distribution of the different gray levels in the image can be plotted through histograms. The dynamic range is defined as [333]

$$D = I_{max} - I_{min} \tag{2.8}$$

with I_{min} and I_{max} the minimum and maximum gray levels.

The contrast is defined as:

$$C = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \tag{2.9}$$

The larger the contrast is, the easier the interpretation of the sonar image is.

The visual enhancement operation does not increase the "quality" of the image but increases the separation between different regions. Histogram operations [333] are used to improve the interpretation. Histogram sliding consists in adding or subtracting a fixed value to all pixels. If the offset is too large, some gray levels will be forced to the minimum range (under-saturation) or to the maximum (saturation). Histogram stretching consists in redistributing the pixel values to increase the dynamic range and the contrast by multiplying the values by a constant and rescaling. For a n bits, each new gray level will be:

$$I_{new} = \frac{I_{old} - I_{max}}{I_{max} - I_{min}} \times (2^n - 1) \tag{2.10}$$

It results in a wider dynamic range and is well adapted to poorly contrasted image with a small dynamic range. These two processes refer generally to histogram equalization.

Speckle removal

Speckle, as explained earlier, is a high frequency noise. Mean filters were proposed to average the speckle but it lowers the resolution of the image [29]. Others are based on the minimization of the mean quadratic error but the analysis window size is usually fixed. This issue is overcome with wavelets decomposition that seems to be an interesting filter since the useful information is generally gathered on few wavelet coefficients [190].

The reflection from the sea surface may appear on sonar images. There are some algorithms to remove this artefact.

All these operations are necessary for correct interpretation of the sonar images. Moreover the reliability of any automatic detection/classification algorithms depends on the "good" formation of the sonar images.

2.3.1.4 Mosaicking

Once the preprocessing phase is achieved, the final step is the mosaicking of the sonar image. It consists in georeferencing the sonar image. The pixels at this moment are still located with

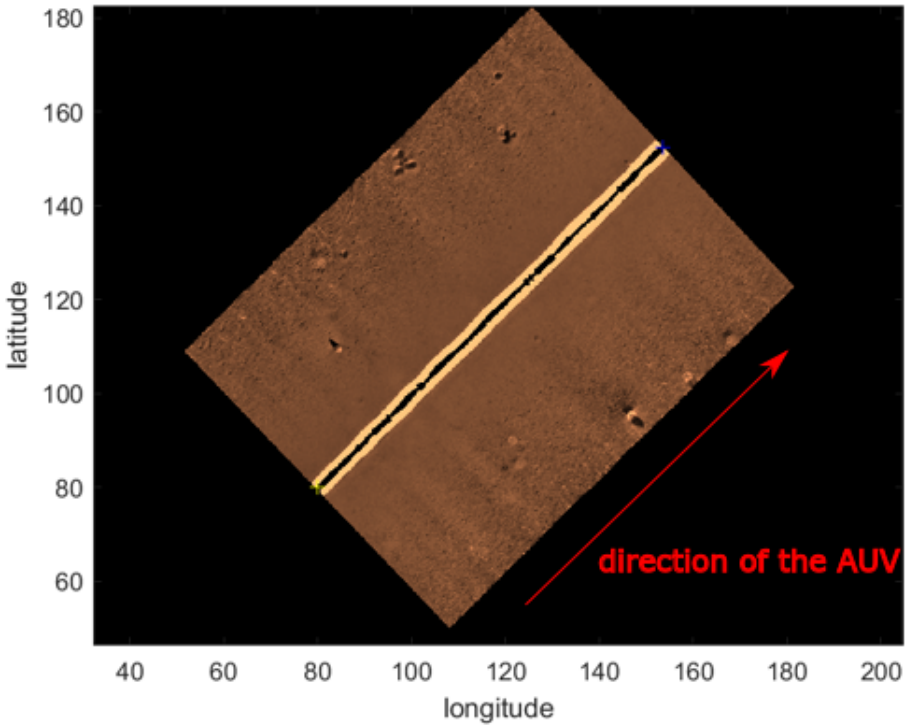


Figure 2.9: Water-fall image for a sidescan sonar.

relative coordinates depending to the sonar platform. Georeferencing is the transformation of these relative coordinates into an absolute coordinates system such as latitude and longitude. A particular map projection has to be defined such as Mercator, UTM,... The georeferencing process starts generally by defining a regular grid as a map with a desired resolution. Then based on the navigation, the attitudes and the slant range correction, an interpolation algorithm fills the grid with the pixel value of the sonar image. Simple interpolation algorithms average overlapping pixels with the mean or the median whereas more complex ones are based on polynomial or spline fitting techniques. The result of a simple median interpolation is proposed in Figure 2.9 in the case of the sonar image proposed.

The transformation from the relative coordinates to the absolute coordinates system is computed based on the rotation matrix and the position of the AUV.

$$\begin{pmatrix} Long_{ij} \\ Lat_{ij} \\ z_{ij} \end{pmatrix} = R(\phi(i), \theta(i), \psi(i)) \begin{pmatrix} 0 \\ \pm\sqrt{R(j)^2 - h^2} \\ h \end{pmatrix} + \begin{pmatrix} x_{AUV}(i) \\ y_{AUV}(i) \\ z_{AUV}(i) \end{pmatrix} \tag{2.11}$$

where i the i^{th} ping emitted (along-track) and j is the j^{th} samples (across-track). The value h refers to the altitude estimated by the bottom track or by a Doppler Velocity Log (DVL), and $R(j)$ is the slant range of the j^{th} sample. The sign $\pm$ is $+$ if the image is on the starboard side and $-$ if it is on the port side. Only the longitude $Long_{ij}$ and the latitude Lat_{ij} are used for the mosaicking.

2.3.2 2D Forward-Looking Sonar (FLS)

Contrary to the sidescan sonar, the 2D FLS does not need the motion of the vehicle to form a 2D acoustic image of the seabed. Generally this sonar is used for obstacle detection and

avoidance due to its forward looking aspect [251]. It is mounted at the front of an AUV or can be embedded in a Remotely Operated Vehicle (ROV) or an Autonomous Surface Vehicle (AUV) due to the weak dimension and power requirements of the sonar in case of lower resolution FLS. Recently due to the improvement of the characteristics, the FLS has been used for seabed mosaicking [326] and even intended to real-time mosaicking [100]. An application in MCM context has been proposed in [99] with the use of a real-time mosaicking building. The acoustic principle is the same for all sonar system. As precised earlier, the transducers emit the acoustic wave spanning its beam width in the azimuth (θ) and elevation (ϕ) directions. Therefore this sonar directly provides a range-bearing acoustic image of the seabed. However it is not possible to disambiguate the elevation angle of the backscattered signal at a particular range and bearing, meaning that the 3D information is lost during the projection into a 2D image. Due to the acquisition nature, it results in a polar coordinates image. This latter is composed of the number of beams in the angular direction and the number of range samples in the range axis. This polar representation can be converted to a 2D image in Cartesian coordinates for a true representation. Due to the non-uniform resolution as the range increases, one pixel in polar coordinates may be mapped to multiple pixels in the Cartesian coordinates with the same intensity. FLS is not really adapted to image around seamounts or abrupt terrain changes with strong 3D changes but it is interesting to be used on a regular seafloor or plane surfaces. The underlying surface should be approximately planar to obtain a correct imaging angle. The surface can obviously be a slope inclined at any angle since it is possible to tilt the sonar to image the underlying surface as a plane.

Three parameters are used to adjust the ensonified area on the seabed:

- the altitude to the plane a.
- the tilt angle ϕ_a of the sonar with respect to the plane.
- the sonar ranges (minimum R_{min} and maximum R_{max}).

Usually the roll of the vehicle is well controlled to gather exploitable acoustic data, consequently only these 3 parameters are needed.

It is difficult in real scenario to adjust the image limits corresponding to minimum and maximum set ranges as depicted in Figure 2.11. It shows a K-ster from Eca Group introduced before with a tiltable head. The ideal case would be the ranges R_{min} and R_{max} corresponding respectively to $\phi_a - \phi_3$ and $\phi_a + \phi_3$ with ϕ_3 the half elevation aperture angle, as shown in Figure 2.10. The detection of the actual leading edge and trailing edge can be inaccurate as presented in gray in Figure 2.11. As it can be expected, due to the large size of the first object ensonified, it appears greater in the sonar image.

Some configurations based on these three parameters can be imagined to understand the differences in the image perceived by the sonar, it is inspired by the work in [326]. In the following figures, the robot configuration is presented on the left and the possible sonar image in Cartesian coordinates is represented on the right. The sonar image is not the real one, it is just an indication of what should be expected in the image.

Choice of the altitude

If the altitude of the robot is high, the tilt angle has to be large, and due to the elevation aperture angle the ensonified area on the seabed will be small as depicted in Figure 2.12. The seabed is ensonified but no features can be extracted due to the small coverage.

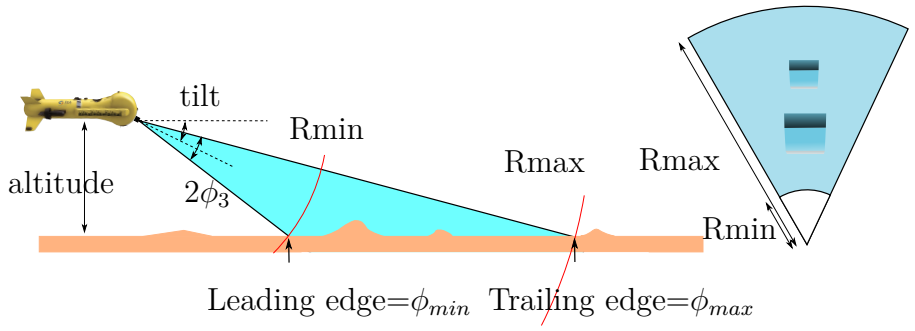
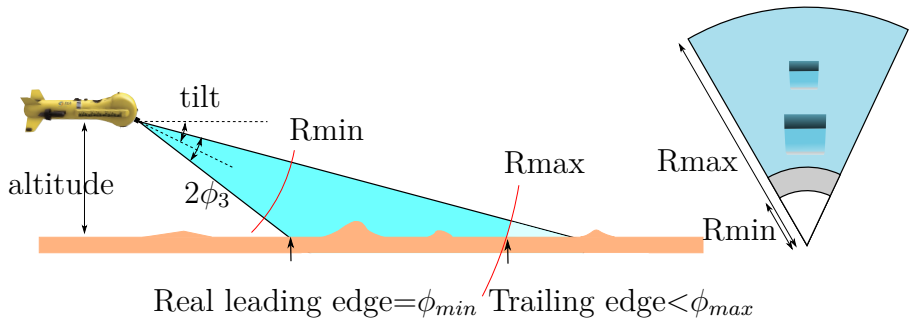


Figure 2.10: The ranges R_{min} and R_{max} correspond to the leading and trailing image edges, at the extremum ϕ_{min} and ϕ_{max}. The image on the right is the image in Cartesian coordinates seen by the sonar. The two protruding object appear with a white echo and a black shadow.



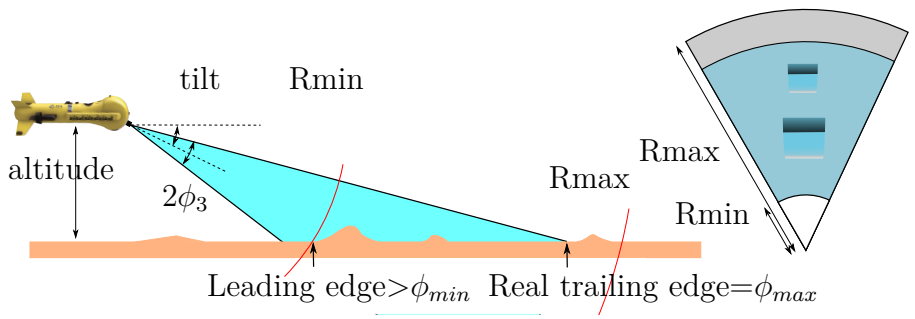


Figure 2.11: The ranges R_{min} and R_{max} do not correspond to the extremum ϕ_{min} and ϕ_{max}. On the top figure, the trailing edge does not correspond to ϕ_{max} and R_{min} does not correspond to the real leading edge. This latter can be detected in the image (gray part). On the bottom figure, the leading edge has an angle bigger than ϕ_{min} that is supposed being at R_{min}. The real trailing edge is at ϕ_{max} but it does not correspond to R_{max}. The trailing edge can be detected in the image (gray part).

Figure 2.12: Small coverage area with an high altitude and a large tilt angle.

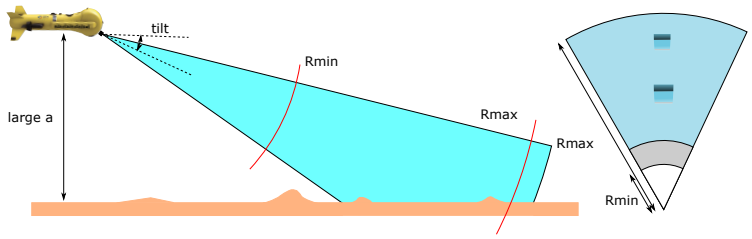


Figure 2.13: Large coverage area with an high altitude and large ranges.

To cover a larger area at high altitude, the idea is to increase the grazing angle, and therefore the sonar maximum range has to be larger to observe the seabed as shown in Figure 2.13. As presented earlier the leading can be detected.

If the sonar maximum range is not enough, the seabed may not be ensonified with this small grazing angle as depicted in Figure 2.14. It would result in a blind image.

Even if the longest ranges reach the seabed, the resulting image would have a lower resolution (less good resolution) than imaging at lower altitude. Consequently, imaging at a low altitude presents the advantages of larger coverage due to the small tilt angle and higher resolution due to closer ranges as depicted in Figure 2.15. But as it can be seen, the set of ranges needs to be correctly defined to increase the visibility in the sonar image.

However, the vehicle has to navigate at a minimum altitude to avoid any crash. Usually the altitude above the seabed is approximately 2 to 5 meters. The altitude is now fixed, the last two parameters (the tilt angle ϕ_a and the ranges R_{min} and R_{max}) have to be choiced.

Choice of the tilt angle

A too tilted angle reduces the coverage area (Figure 2.16) whereas a small grazing angle

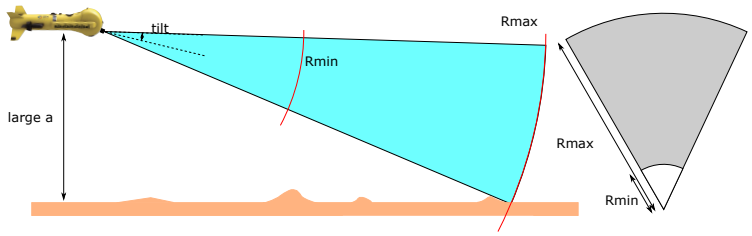


Figure 2.14: No seabed due to high altitude and too short ranges.

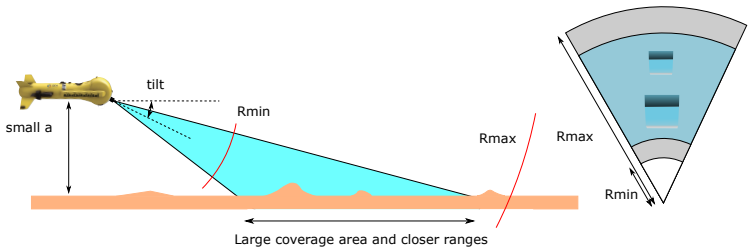


Figure 2.15: Large coverage area and high resolution images.

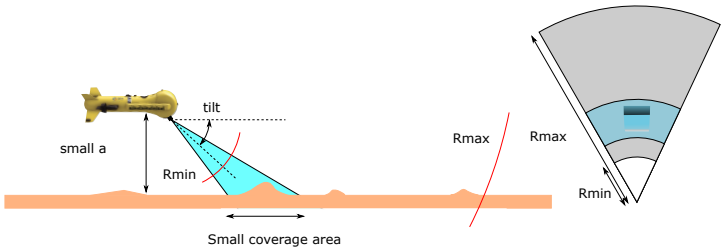


Figure 2.16: Small coverage area with a small altitude of the robot due to large tilt angle.

ensures a large coverage area (Figure 2.15)

A small grazing angle, handled by the tilt angle, would create too long shadows. Consequently the objects at further distances could be occluded by these shadows. Approximately 10-20 degrees seems to be a good compromise [326].

Choice of the ranges

A small tilt angle at low altitude provides a large coverage area with high resoluted images. However, to avoid at maximum the blind areas depicted in gray in the different images, the set of ranges has to be well defined. Indeed, the blind areas come from the leading and trailing edges. The idea is to increase the minimum range R_{min} until the first returns and reduce the maximum range R_{max} to the limit of the last returns as depicted in Figure 2.17 where no blind areas are present in the sonar image. This case was cited as ideal case before.

In conclusion, the tilt angle, the altitude and the ranges have an effect on the resolution in the FLS image, as expected according the general principles presented in the first subsection.

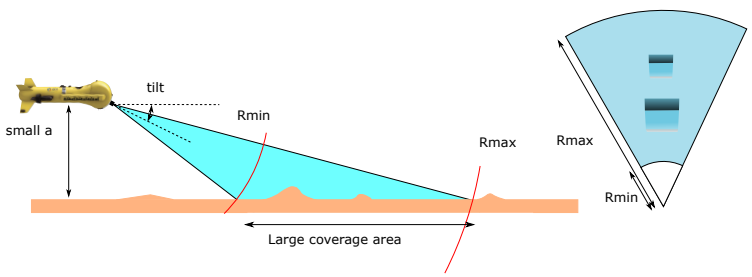


Figure 2.17: Large coverage area without blind zone in the sonar image.

In this subsection, it deals with a vehicle equipped by a tiltable head that can change the elevation angle for the ensonification of the seabed. However, this tilt angle can be fixed and then the vehicle just has to change its pitch angle to ensonify toward the seabed. Usually in case of navigation, the FLS is mounted in order to ensonify toward with a small tilt angle between 10-20°. The roll angle should be managed to keep the sonar as parallel as possible to the imaged plane.

2.4 Detection in sonar images

In the previous subsection, the basic knowledge to understand the sonar imagery has been introduced with special presentations of the sidescan sonar and the FLS. Indeed, the sidescan sonar is widely used for seabed mapping due to its high resolution and some algorithms have been developed to detect automatically objects or textures, especially in the case of MCM context where an operator, in the past, was looking at the sonar image to detect a mine. Due to the necessity of a reliable navigation to compensate the drift in position of the robot, the relocate process of the robot and to avoid being lost, the geophysical navigation presented in Chapter 1 requires the detection of elements in the environment the robot is navigating. The automatic detection of these elements, called landmarks, is vital for the autonomy of the different missions. These landmarks are naturally present in the underwater environment such as rocks, frontiers between different types of seafloor, wrecks, etc... In this subsection, the main features detection techniques in sonar images will be presented.

2.4.1 Landmarks detection

Landmarks in underwater environment are sometimes called "seamarks". They may be of various natures, from rocks to sand ripples going through man-made objects and wrecks. Many algorithms exist today for automatic detection but they differ in the way of detecting. As presented in the subsection 2.2.3, the presence of protuding objects is characterized by the an echo-shadow pair in the sonar images. Indeed due to the nature of the sonar image creation, the pixels representing a protuding object has an intensity higher than the average intensity and the shadow a lower intensity than the average. It may be possible than the echo is not detectable due to a weak or isoptropic backscattering effect. A protruding objet could be a mine, a rock, a wreck,... something that is lying on the seabed. The detection of such landmarks depends on the ability of detection of this echo-shadow pair. This will be the subject of the next subsection. However, in the underwater environment, these landmarks revealing an echo-shadow pair are not the only landmarks a navigation system with an imaging sonar could rely on to relocate itself. Indeed, the seabed is composed of various sediment areas such as sand ripples, posidonia, sand, mud, rocks region, etc... If the sonar system is able to detect all these areas, it can actually relocate the robot with the frontier between these sediment regions. These regions are known as textures. This will be briefly explained after the object detection. Even if these two detections process appear separately in the way of detecting, having the knownledge of the different sediment regions could help to the detection of the object inside these regions [254].

Generally these algorithms can be described by this architecture:

- Detection: detections in the sonar image.
- Extraction of features: features that characterize the detections.
- Classification: What is it?

2.4.1.1 Objects detection

The detection of objects in underwater environment with sonars has been widely treated in the context of MCM missions. Indeed, the automatic detection in such missions refers to Automatic Target Recognition (ATR) algorithms which aim to detect, classify and identify a contact. The detection process for these algorithms can be used to detect any objects on the seafloor that can constitute a landmark. The detection phase is intended to find some *regions of interest* (ROI) in the sonar images where an object could be present.

Matched filter

The presence of an object has been well defined in the section 2.2.3 due to the presence of an echo corresponding to a high level of the backscattered signal and a shadow corresponding to a low level. The dimension of the shadow depends on the grazing angle and on the height of the object. A 1D filter has been proposed in [299] for the detection of this kind of responses. This 1D presence is revealed after the convolution of the filter and the initial signal. It has been extended to 2D in [78].

Segmentation

The segmentation is the most known process for detection in the image processing community. In the case of objects detection, the segmentation aims usually to divide the sonar images into three classes: echo, shadow and seabed.

It has been shown in [79] that the distribution of the pixels (intensity) associated to an object is not the same compared to the distribution associated to the seabed. The author proposed to use an adaptive threshold to highlight the echo of an object and then was looking for the associated shadow.

Simple clustering techniques such as histogram thresholding [263] are the most famous segmentation algorithms. An adaptive thresholding technique based on the local mean is proposed in [313]. Fuzzy k-means [117] is also proposed. These techniques perform well on flat seabed with high signal-to-noise ratio (SNR) but fails on complex environments such as sand ripples.

A fuzzy logic approach has been proposed in [97] to separate echo and shadow from the background. Based on an iterative technique, a fuzzy function has a likelihood that increases with the number of neighbouring pixels of the same class and decreases as the pixel intensity value is far from the typical intensity value of the class.

Markov-Random Fields (MRF) is a statistical segmentation that uses *a priori* information about the intensity values of different classes (shadows, echos and background), and the intensity values of the neighbouring pixels to label all pixels belonging to the predefined classes (shadows, echos and background). In other terms, a pixel surrounded by shadow pixels is most likely to belong to the shadow class. MRFs were applied in medical images [346] and on sonar images in [58][217][218] but the segmentation was long and the computation demanding. In [266][267], the MRF algorithm were improved to increase the performance of the segmentation by using the spatial *a priori* knowledge on the size and the geometry of the objects to divide the image into the three classes (echo, shadow, seabed). This method was reviewed in [66] with graph-cuts[33] technique to increase the speed of the MRF segmentation. The main difficulty in MRF segmentation is the estimation of the parameters characterizing the *a priori* knownledge of the environment. These parameters are the noise model parameters defined by Probability

Density Functions (PDF) for each class and the Markovian parameters (known as MRF prior parameters or cliques parameters). The Markovian parameters have the effect to gather pixels. All these parameters are usually estimated with an Iterative Conditional Estimation (ICE)[286] that enables the estimation of the Markovian parameters based on a Least Squares Estimator (LSQR) described in [70] for example and the estimation of the noise parameters based on a Maximum Likelihood Estimator (MLE) for example. An "initial" segmentation needed for the distribution estimation can be performed simply with a K-means algorithms as in [220]. A Gaussian law is usually defined to describe the distribution of the shadow or the echo, and a Rayleigh distribution to model the speckle noise [219]. A Weibull distribution has been used in [95][96]. The Markovian parameters can be fixed *a priori*. After the parameters estimation, an Iterated Conditonal Modes (ICM) [24] is used to optimize the energy function of the markovian model by looking for the Maximum A Posteriori (MAP). The Markovian segmentation developped in the work [267] was used in Simultaneous Localization And Mapping (SLAM) problem with a sidescan sonar in [280] to provide information in a stochastic map about the echo-shadow detections and perform a data association based on the Joint Compatibility Test (JCT) with a Mahalanobis distance to keep the consistency of the stochastic map.

Snakes are statistical active contours that assume the regions have a continuous edge [155]. It is a closed curve defined by a list of nodes. It tries to minimize a cost (energy) function by deforming the contour [53]. It results then in a two-classes segmentation, a target region and a background region. Snakes have been widely used in medical image segmentation [60]. It can be seen as a matching technique between a deformable model and the object to detect in the image based on the minimization of an energy function. The main drawback of snakes is the requirement of an initial position or a contour to start moving from. The result of a previous segmentation could be used to start the deformation of the initial snake. Due to the dual nature of the object signature in the image (echo+shadow), a co-operating statistical snake has been developped in [266] to extract features based on the results of a MRF segmentation. Actually two snakes are used to segment both the echo and the associated shadow. It uses the *a priori* information between the shadow and the echo of an object to constrain the motion of the snakes. Samely it has been used in [96]. Contrary to MRF segmentation, snake algorithms do not need *a priori* information, only the intensity of the pixels influences the segmentation result.

Region growing [4] have been applied for classification of seabed in [154] and echo/shadow detection in [87].

In morphological methods, watershed algorithm interprets the image as a relief or a topographic map and then finds the lines that separate the image in homogeneous areas based on the morphological gradients [216].

Saliency detectors

The saliency can highlight some region where there are some differences between a textured background and its neighbourhood. In other terms, an object is said to be salient if it stands out from its background and all other objects.

In [11], a scale saliency object detection is proposed using the Shannon entropy. The scale defines the size of the local neighbourhood in the image. The shadows and echos have a stronger entropy than the seabed at specific scales. The algorithm is then looking for a peak in the entropy function by varying the scales.

In [339][336] [334] a detector based on the integral image is proposed. Based on different sized sliding windows, an echo and shadow map is generated in a cascaded architecture. The

approach is able to make near real-time detection in sonar data onboard of an AUV and has been used on high resolution SAS images. Due to the fast computation ability in sonar images, this detector has been implemented to detect some objects on the sonar data in this thesis. The principle of this detector will be the subject of the subsection 2.4.2. The integral image has also been used for object detection considering the local clutter in Haar-cascaded classifier [288] initially developped in image processing [327].

2.4.1.2 Environment characterisation

The texture detection relies mainly on simultaneous segmentation and classification techniques with features extraction to characterize the different textures representing the seabed [269]. The classification can be supervised or unsupervised to differentiate the classes.

In [190], the seafloor is separated into 5 classes: rocks, sand ripples, sand, mud and shadows. A supervised classification based a Nearest Neighbour (NN) is proposed after a Linear Discriminant Analysis (LDA) [210] closed to the Principal Component Analysis (PCA) principle. The parameters are obtained with a set of Gabor filters [134] to detect the local presence of a frequency. The size of the windows is an important choice.

In [67] the seabed is classified into three categories: homogeneous, anisotropic and complex regions. This idea comes from the fact that the false alarm rate increases in regions where the environment is highly textured. An anistropic region refers to the sand ripples and the complex region to a large number of rocks inside a region. On the contrary the homogeneous regions are regions generally composed of sand and mud. Consequently the detection of a protruding object on an homogeneous seabed will lead to an easier detection. These regions are categorized based on complexity and anisotropy coefficients. These coefficients can be estimated with Gabor filters [67], Haar filters [15], complex wavelet analysis [105] or with the monogenic signal [254][255].

2.4.2 Integral image detector

A proposed implementation of the algorithm based on the integral image in [336] due to its rapid computation, even near real time, is provided. The object detection is proposed on the port image of the sidescan sonar image depicted in Figure 2.7 after the different corrections explained. Moreover the image integral will be used for contractor image programming in Chapter 4 about the set membership tools.

An integral image [329] is an image representation that enables fast computation of rectangular, Haar-like features at any scale or location in constant time. The sonar image is then converted directly into an integral image.

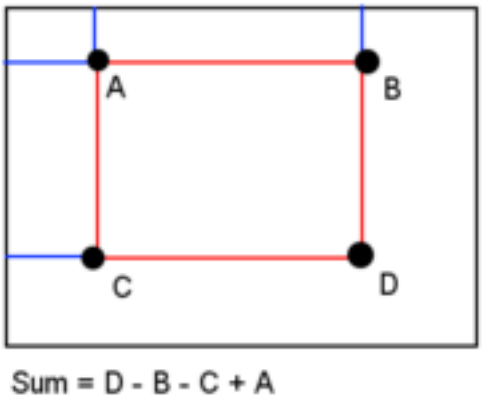


Figure 2.18: Easy computation of the sum of the pixels inside a rectangular area (A,B,C,D) based on the integral image.

The construction of the integral image I corresponding to an original image A is defined as follows:

$$I(x,y) = \sum_{x' \leq x, y' \leq y} A(x',y') \tag{2.12}$$

For a pixel location at (x,y) in the integral image it corresponds to the sum of the pixels above and to the left of (x,y) in the image A. It can be recursively computed with:

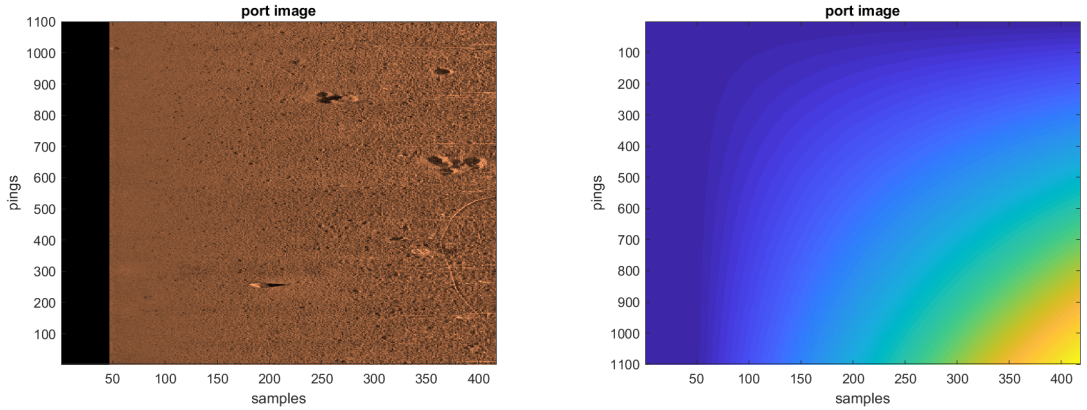
$$I(x,y) = I(x-1,y) + z(x,y) \tag{2.13}$$

with $z(x,y)$ is the cumulative sum of pixels in a row of the original image A computed by:

$$z(x,y) = z(x,y-1) + A(x,y) \tag{2.14}$$

The integral image corresponding to the port image is depicted in Figure 2.19. Notice that the pixels belonging that the water-column are set to zero. The boarders are usually handled by replicating the pixels (mirror) or just zero padded. Other techniques could have obviously been envisaged. Once the integral image is computed, it is easy to compute any sum of pixels inside a rectangle as shown in Figure 2.18. If the integral image is called Y for an original image X the sum of the pixels inside the rectangle (A,B,C,D) can be computed easily with:

$$\sum_{x_A < x' \leq x_C, y_A < y' \leq y_C} X(x',y') = Y(A) + Y(D) - Y(B) - Y(C) \tag{2.15}$$



(a) Original port image after TVG correction and water-column removal.

(b) Integral image. Blue corresponds to low values and yellow to high values.

Figure 2.19: Integral image computation on the port water-fall sonar image.

Background map

Firstly, based on the integral image computation, a background map B_{map} is generated. This map is intended to represent the reverberation level of the seabed. As previously mentioned the reverberation is highly dependent on the nature of the seabed. The background map is estimated based on a split-window template composed of three equally sized horizontal rectangles due to the nature of sidescan imagery. Indeed, in this kind of imagery, the convention is to have a representation with the echo coming first. The horizontal axis corresponds to the across-track axis and the vertical axis is the along-track one. These rectangles are juxtaposed in the along-track direction. The top and bottom rectangles contribute to the background estimation and not the middle one. The idea is to identify the surrounding of an object. The size of the bands is defined in along-track and in across-track direction. These sizes depends on the length of the object research, they are user-defined parameters. Indeed the idea is to fix the size in order that the middle rectangle fully contain the object. The background score at the location (x, y) corresponds to the mean level between the top and the bottom rectangle. This score is calculated as follows:

$$\begin{aligned} B_{map}(x,y) &= 0.5 * (I_{top}(x,y) + I_{bottom}(x,y)) \\ &= a_b^{-1} \Big[I(x - 3\tfrac{dx}{2}, y - \tfrac{dy}{2}) - I(x - 3\tfrac{dx}{2}, y + \tfrac{dy}{2}) \\ &\qquad - I(x - \tfrac{dx}{2}, y - \tfrac{dy}{2}) + I(x - \tfrac{dx}{2}, y + \tfrac{dy}{2}) \\ &\qquad + I(x + \tfrac{dx}{2}, y - \tfrac{dy}{2}) - I(x + \tfrac{dx}{2}, y + \tfrac{dy}{2}) \\ &\qquad - I(x + 3\tfrac{dx}{2}, y - \tfrac{dy}{2}) + I(x + 3\tfrac{dx}{2}, y + \tfrac{dy}{2}) \Big] \end{aligned} \tag{2.16}$$

with dx and dy corresponding respectively to the number of pixels in the along-track and in across-track direction. dx and dy are based on the resolution and the size of the rectangle desired. a_b is the number of pixels involved in the sum of the two rectangles. The contribution of the middle rectangle is not considered, because if an object was present, it would bias the background estimation.

The result of the background estimation is proposed in Figure 2.20. The background estimation around a real target is like doubled due to the estimation of the bottom rectangle when the estimation is computed above the target and the estimation of the top rectangle when the estimation is computed under the target.

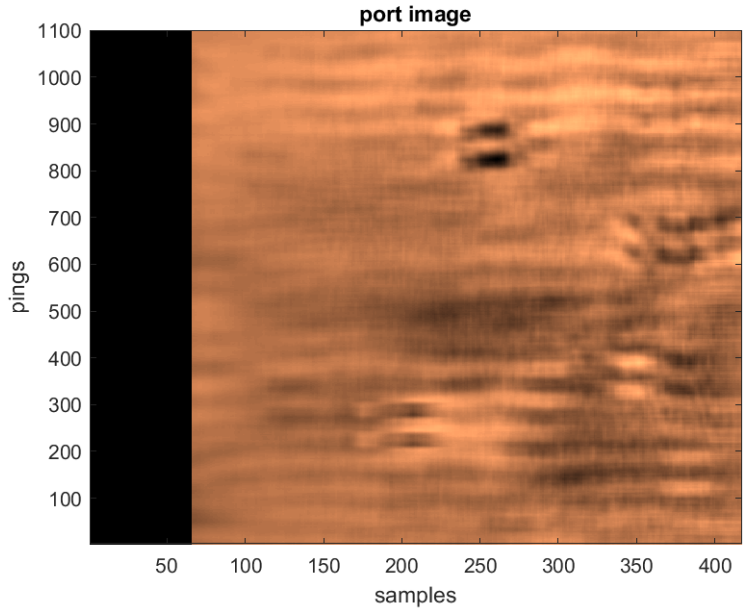


Figure 2.20: Background map.

Shadow map

The second step is the shadow map S_{map} estimation to identify area in the image where an object protruding from the seabed is present. Contrary to the background estimation, only one rectangle around the position (x, y) is used to directly compute the presence. As mentioned in the subsection 2.2.3, based on the explanation in Figure 2.4 and Equation 2.6, the length of the shadow depends on the height of the object, the altitude of the sonar and the slant range between the sonar and the object. Due to this range dependence, the size of the rectangle for the shadow calculation will depend on the range between the sonar and the (x, y) position. From Equation 2.6, the length of the shadow is:

$$L = \frac{hD}{H - h} \tag{2.17}$$

with H the altitude of the sonar, D the slant range, and h the height of the object. H is measured by a DVL or can be estimated with a bottom tracking algorithm and h is user-defined parameter. h is set at $20cm$ for example.

The along-track dimension s_x and the minimum height of the object h are user-defined parameters for the rectangle. The across-track dimension s_y of the rectangle is defined by Equation 2.17 ($s_y = L$). The shadow score is then computed with one rectangle around (x, y) position as follows:

$$\begin{aligned} S_{map}(x, y) = a_s^{-1} \Big[& I(x - \tfrac{dx}{2}, y - \tfrac{dy}{2}) - I(x - \tfrac{dx}{2}, y + \tfrac{dy}{2}) \\ & - I(x + \tfrac{dx}{2}, y - \tfrac{dy}{2}) + I(x + \tfrac{dx}{2}, y + \tfrac{dy}{2}) \Big] \end{aligned} \tag{2.18}$$

with as the number of pixels in the rectangle considered. dx and dy correspond respectively to the number of pixels for the along-track dimension s_x and the across-track dimension s_y. The result of the shadow map is depicted in Figure 2.21.

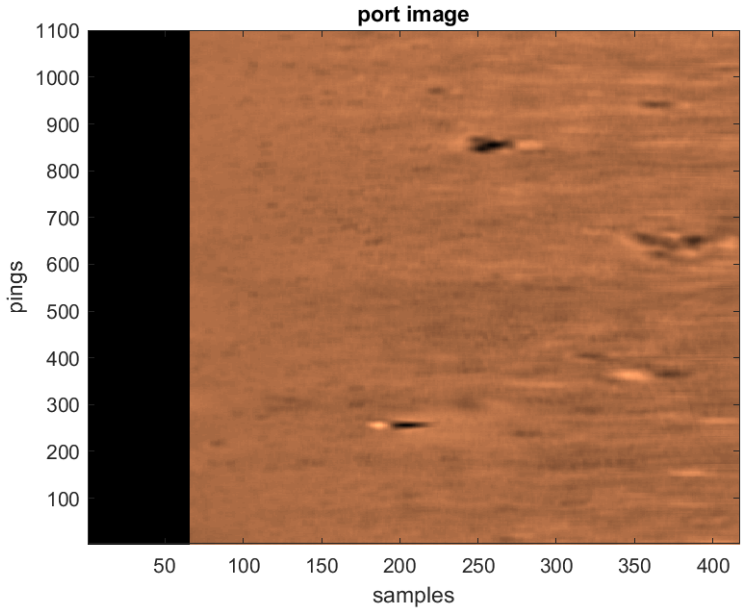


Figure 2.21: Shadow map.

The advantage of the range dependent size of the rectangle is to take into account the geometry of the problem. In general, matched-filtering methods cannot use this range dependence.

Region of interest

Based on the background map B_{map} and the shadow map S_{map}, the regions that may contain targets of interest can be determined. A pixel belongs to a shadow when the value at this pixel is lower than a certain amount of the background map estimated at this pixel. A pixel located at (x, y) belongs to a shadow if:

$$S_{map}(x, y) < \alpha B_{map} \tag{2.19}$$

with α a threshold fixed. The result of the binary map obtained is depicted in Figure 2.22. It enables to reduce the size of the image that has to be analysed and it highlights the regions of interest where an object with a minimum height h may be present.

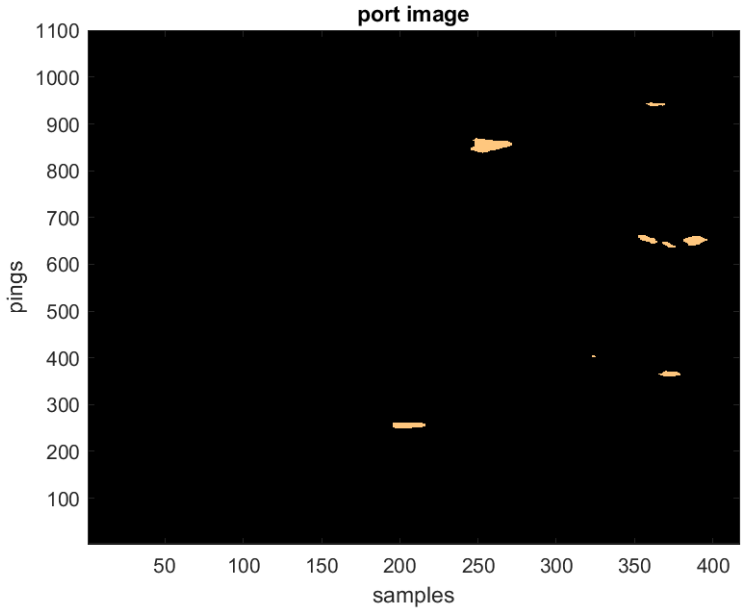


Figure 2.22: Regions of interest.

Based on the result of the regions of interest, a snake algorithm [51] is applied to extract the real shadow region. The result for one shadow is depicted in Figure 2.23. As mentioned earlier, the initialization of the snake is based on the result of the regions of interest. The resulting shadow are filtered based on the height of the object.

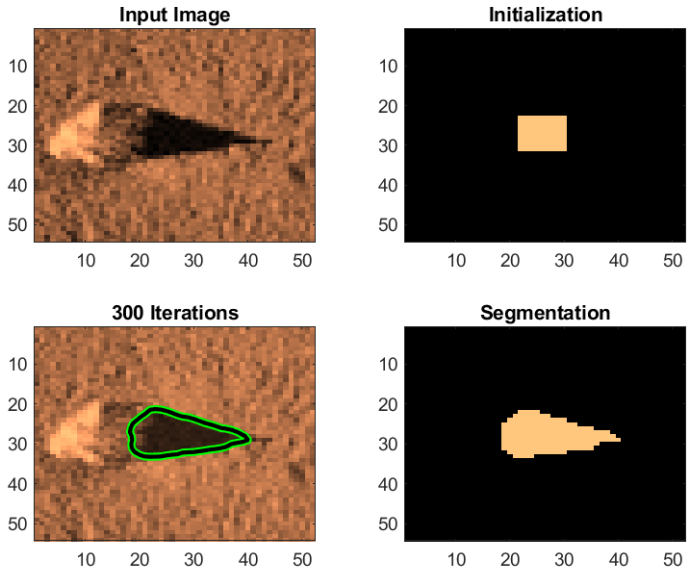
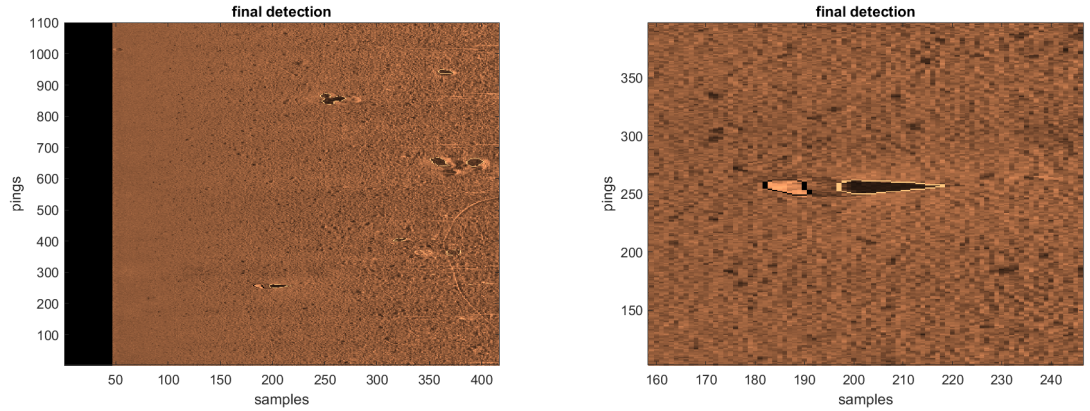


Figure 2.23: Snake algorithm to extract the shadow.

Echo detection

The presence of echos is only estimated at a shorter range than the shadows detected previously with the snakes. Similarly the echos are revealed with a threshold on the background map as the pixels belonging to echos have a higher value than the background map. The result of the echo detection and the shadow associated is represented in Figure 2.24.

Again a snake algorithm can be computed to extract the echo.



(a) All detections in the port image.

(b) Zoom on a detection result.

Figure 2.24: Final detections with contours surrounding the shadow and echo detections.

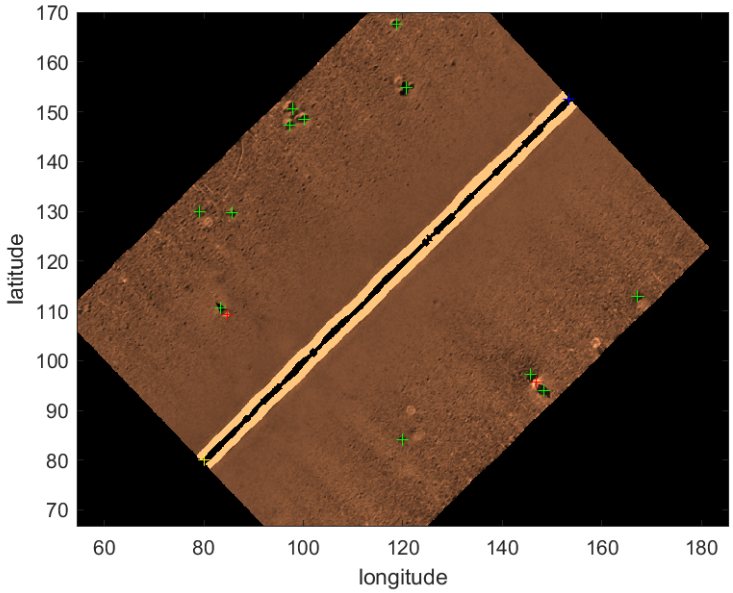


Figure 2.25: Detections on the mosaic. Red crosses indicate echo detection and green crosses shadow detections.

The different detections are projected on the mosaic in Figure 2.9 previously computed. The result is depicted in Figure 2.25.

Conclusion

Through this algorithm, it is possible to compute fastly object detections in sonar images. The advantages of this method is the range dependence that is inherently present in this kind of image due to the sonar geometry. Moreover a filtering step on the minimum height is provided based on the length of the shadow. When using data from different sonars, it may be difficult to associate the different data. Some features extracted from the different objects may help in the data association process for relocation when an object is seen again at different times.

2.4.3 Extraction of features

Features extraction is usually the step just after the region of interest extraction, as explained before, and just before the classification. The features describing a "contact" may help in the

definition of the different object detected in the sonar image. It may indicate if the contact is a MILCO or not a MILCO [254]. Some features extracted may help in the data assocation process when navigating in a environment to disambiguate some association based on the nature of the object and reduce the pose uncertainty. Obviously the position of the different landmarks are the primary features but sometimes it is not sufficient to have an *a priori* indication of the position. In [282], the position, the size of the target and the first invariant moment descriptors [281] are taken into account in the stochastic map and for the MHTF (Multi Hypotheses Tracking Filter) data association [268].

Statistical features may be extracted from the detections. They represent the intensity distribution of the different object. It can be simple features such as the mean, the variance, the energy or the entropy, or more complicated ones based on moments theory [132]. These moments enable to compute some features invariant to translation, rotation and scale. In [262], Zernike moments [305] are extracted for classification purpose on single look sonar image.

Fourier descriptors are proposed in [262] where the contour of an object can be described in the frequency domain through a Fourier transformation. These descriptors are invariant to translation, rotation, dilation or shift of the contour. Only a limited number of Fourier descriptors can describe the object.

Finally, geometric features such as the length of the shadow, its width or its orientation can be extracted in sonar images. The length of the shadow is directly linked in sonar images to the protruding height of the object from the seabed.

More details on the features that can be extracted for classification with sidescan sonar can be found in [98].

Features describing the different textures can also be determined as for example the orientation and the spatial frequency of sand ripples regions.

2.5 Notion of aspects

The notion of aspects, i.e. point of view, is an important notion in underwater environment and sonar imagery. Indeed, due to the shape of an object and the grazing angle of the sonar, the backscattered signal from this object is not the same at every point of view in sonar imagery, as mentionned in the subsection 2.2. Moreover the materials defining the objects, man-made or natural objects, may not react the same way to acoustic waves. These aspects are particularly solicited in the classification of underwater objects and especially in MCM context. The classification with a single-view may not be sufficient due to complex geometric shapes sometimes. A classification based on a multi-view process may remove an ambiguity [262] and is solved by fusing single-view classification in [98]. In [229], the multi-view classification problem is modeled as a POMDP (Partially Observative Markov Decision Process) problem where it determines which additional point of view would reduce the uncertainty in classification of different targets such as cylinders, truncated cones (Manta), rocks, or wedges. This multi-view classification leads to the problem of revisiting some targets explained in the section 1.3.1 and the necessity to reacquire some point of view of targets to increase the classification performance. In [345] using simulated sonar images of various objects and height profiles as features, the highest classification performance was achieved with an angular increment of 90° between the two images. This result was then confirmed on real data [62].

Moreover viewing the same *a priori* texture at different point of view does not mean that the sonar will be able to detect it for sure. For example, due to the strong orientation dependence of sand ripples, when scanning the seabed with a side-looking sonar such as SAS or a sidescan sonar, the sand ripples region appears as a flat seabed when the orientation of the sand ripples is in the along-track direction of the vehicle [338]. In others terms, when the acoustic waves are oriented perpendicularly to the direction of the sand ripples, it may appear as a flat seabed. In MCM context, the presence of sand ripples increases the false alarms rate in mine detection [116] because the targets are hidden in the shadows of ripples. It could be interesting to revisit this sand ripples region at a more favorably heading where the sand ripples appear as a flat seabed to detect the mines hidden lying on the sand ripples seabed [338][64]. However, in case of navigation, if the sand ripples region is considered as a landmark, the relocation process is possible only if this region is detected. Therefore the orientation of the sand ripples has to be taken into account in the path planning of the robot to be sure that it will detect it.

The notion of aspects is defined as an angle of view in case of objects or an orientation in case of an anisotropic region such as sand ripples. In the MCM context, a survey mission provides the data thanks to a sidescan sonar. Consequently the geometry of acquisition indicates the angle of view of the detected objects in the post-processing of the sonar data. Moreover, if sand ripples are present, it may be detectable by some algorithms such the ones in [255][338][336]. The aspects is illustrated in Figure 2.26.

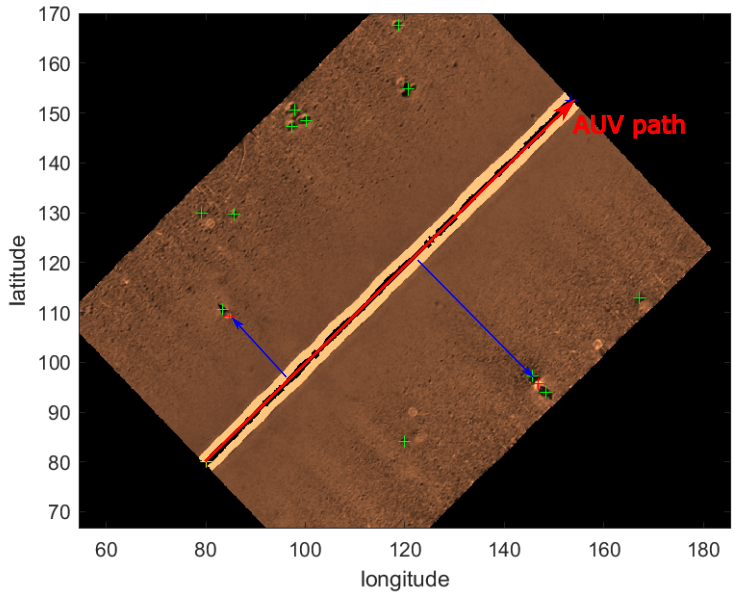


Figure 2.26: Aspects of the different detections depending on the AUV path and the sidescan sonar geometry. The path of the AUV is depicted by a red arrow indicating the direction of the AUV. The blue arrows show the echo detections according to the AUV path.

The angle of view is defined in the sidescan imagery as:

$$\theta_{detection} = Heading_{AUV} \pm 90^{\circ} \tag{2.20}$$

with $Heading_{AUV}$ in degrees and $\pm$ depending if the detection is on port or starboard side. $\theta_{detection}$ is the same reference as the heading of the vehicle.

Consequently, the different landmark detections can be represented in the reference according to Figure 2.27. $\theta_{detectionSS}$ refers to the angle of view at the moment of the detection. An interval around this angle of detection is introduced as it is quite impossible to revisit an object at the exact same point of view. This interval provides a flexibility on the angle of view as in [61].

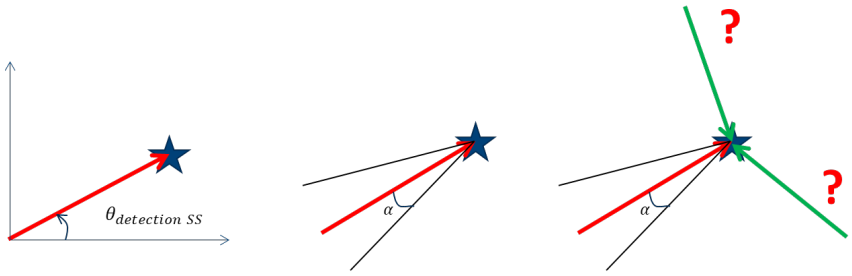


Figure 2.27: Aspects of detections.

This angular flexibility is then defined as follows:

$$\theta_{detection} = [\theta_{SS} - \alpha, \theta_{SS} + \alpha] \tag{2.21}$$

with θ_{SS} referring to the angle of detection $\theta_{detectionSS}$ and α the angular flexibility. Moreover, introducing a low angular flexibility does not change the classification performance [61]. Along this thesis α will be 10 or 15°. Detecting an object at a specific angle of view does not mean that the object is detectable at every point of view as stated at the beginning of this subsection.

2.6 Heterogeneous imaging sonars

Using heterogeneous sensors refer to multimodal/multisensor registration in image registration process. SAR (Synthetic Aperture Radar) images are registered with optical images of urban areas in [192] based on features extraction and Mutual Information (MI) registration, of various environments in [198] based on a contour matching algorithm or on medical images of brains in [256] with diffusion maps. Registering heterogeneous sensor images is not an easy task due to the difference in intensities and geometry acquisition such as in sonar images. Indeed, the sonar images are highly dependent on the altitude of the AUV, the angular aperture of the antenna, the frequency of the sonar, the ping emission frequency, the speed of the vehicle, etc... All these parameters influence directly the ability of detection of a sonar as previously mentioned in the section 2.2. Due to the heterogeneity of information gathered during the survey and the revisit mission, a filtering step is proposed to adapt the data base created during the survey and the revisit system parameters. Obviously it is possible to use the same system for both mission.

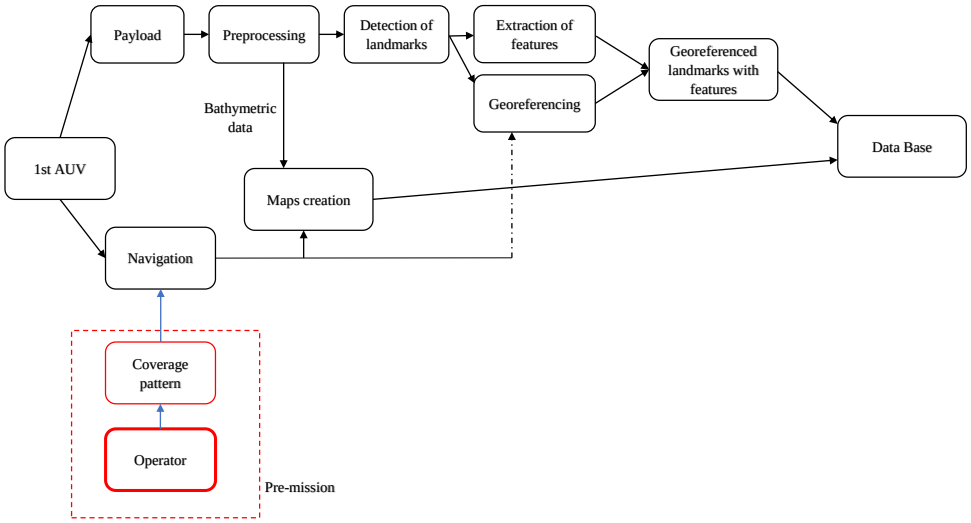
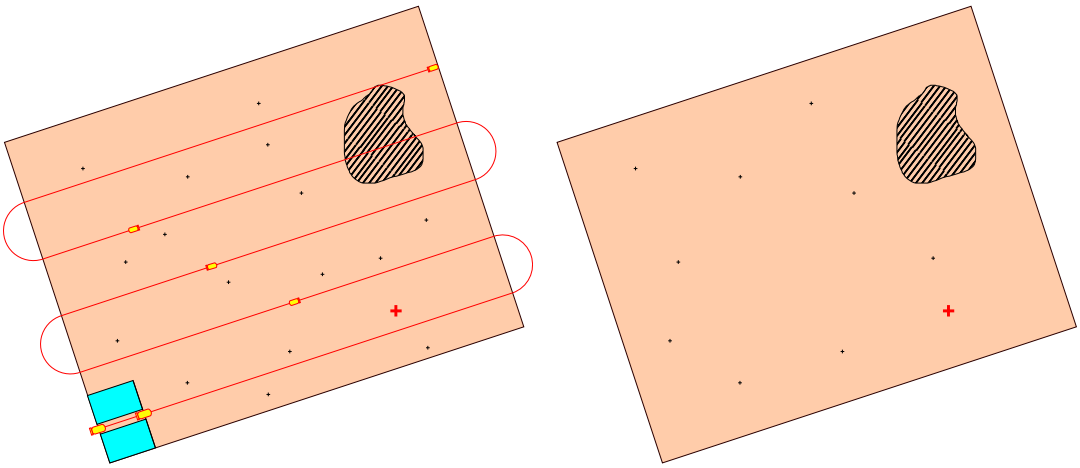


Figure 2.28: Creation of a data base after the survey mission.

Post-processing the survey data collected leads to some detections with automatic algorithms presented earlier or manual detections by an operator according to Figure 2.28. This data base contains all the landmarks characteristics including the positioning, the angle of view and some features such as geometrical or other types of features. Moreover it can store an eventual classification of the objects or textures such as rocks or sand ripples region with their orientation and spatial frequency. A texture region can be stored as a shape as presented in [71]. Due to the acquisition geometry, this data base is said to be only adapted to the survey system. If afterwards another system is navigating this environment, the information may differ from this data base. In the context of this thesis, the aim is to revisit an object classified as potentially dangerous object (MILCO) in MCM missions. The objective is to guarantee the revisit of this object with relocation process based on the map created in the data base. Based on the characteristics of the revisit system (altitude of the vehicle, sonar, etc...), the filtering step keeps only the landmarks *a priori* detectable by the revisit system. A simple filtering step based on the dimensions of the landmark (width, length, height) is proposed in case of objects. Indeed, the resolutions of the sonar images can be *a priori* computed based on the different characteristics presented in the subsection 2.2. Assuming an altitude of the vehicle, a desired range of detection to observe the landmarks and the shadows, the idea is to remove all landmarks that may not be detectable by the revisit system as presented in Figure 2.29.



(a) All the landmarks detected by the sidescan sonar along a *Boustrophédon* survey.

(b) Landmarks detectable by the revisit system.

Figure 2.29: Filtering of the data to adapt to the revisit system.

According to the sonar principle and considering an object with a width W, a length L and a height h (protruding height from the seabed), the following inequalities have to be respected in order to detect this object:

$$W > rd\theta \tag{2.22}$$

$$L > dr \tag{2.23}$$

$$SL > dr \tag{2.24}$$

with $d\theta = \frac{\lambda}{La}$ and $dr = \frac{c}{2B}$ the cell resolutions. The azimuth resolution depends on the range r between the sonar and the object. SL corresponds to the length of the shadow in the sonar images which is given by:

$$SL = \frac{rh}{H - h} \tag{2.25}$$

with H the altitude of the sonar and h the height of the object, assuming a flat seabed. If the altitude is fixed *a priori* and as the height of the different objects are available in the data base, it is possible to know the shadow length at different ranges r. In other terms, the cell resolution should be smaller than the size of the object. Moreover, according to the Johnson's criteria [149], at least one pixel is needed for the detection purpose.

Several pixels are needed for the height estimation, therefore the length of the shadow of an object should be sufficient. The resolutions in the sonar image do not correspond to the cell resolution due to the over-sampling criteria for the conversion of the data not to loose data.

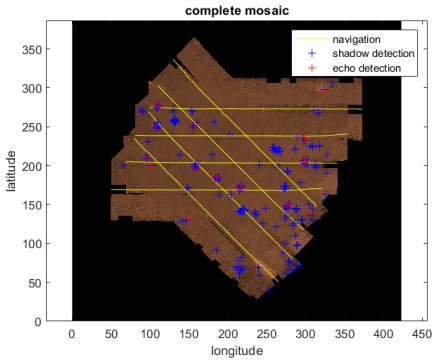
The set of ranges r has to be defined, and usually a detection in the mid range is better.

A study on the resolution needed for classification and identification has been proposed in [241] for further details.

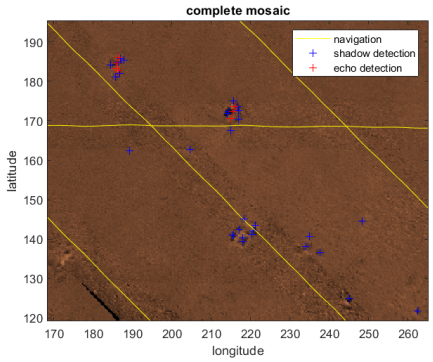
This filtering step is proposed in Figure 2.30 where a mission was conducted by an A9 from ECA Group with a sidescan sonar. The sonar images presented earlier came from this mission. The AUV performed a *boustrophédon* pattern as it can be noticed by the yellow parallel tracks. Only the navigation during the sonar acquisition along a straight line is represented (not the turns). Four E-W tracks and four NW-SE tracks (N: North, S: South, E: East and

W: West) are drawn in yellow. The result of the integral image detector is presented in Figure 2.30(a) where the blue crosses are the shadows detection and the red crosses are the echos detection associated to the shadows if they exist. The sonar images have not been registered and then the same landmark may appear at different locations. Due to the low uncertainty on the robot position thanks to a reliable navigation, these locations (of the same landmark) are really closed to each others (tens of centimeters). A zoom is provided in Figure 2.30(b) where the median has been used for the complete mosaic representation to highlight the shadows. It can be noticed that the environment is only composed of "punctual" landmarks on a sandy area. Some landmarks are described by an echo and a shadow. The filtering step only kept some landmarks as depicted in Figure 2.30(c) with a strong echo and a good size. A zoom is provided in Figure 2.30(d) where the maximum has been used to create the complete mosaic to highlight the echos (high gray values). The difference in location for the same landmarks can be seen by the different red crosses. This final map will be used by the planner proposed in this thesis to find a strategy for the revisit system since these landmarks are *a priori* detectable by the revisit system. A manta mine is present at the coordinates $(160, 250)m$.

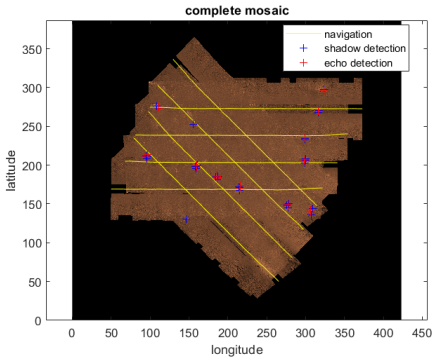
Notice that some "holes" are present at longer ranges in the sonar mosaic, it was an acoustics communication where the pixels have been removed.



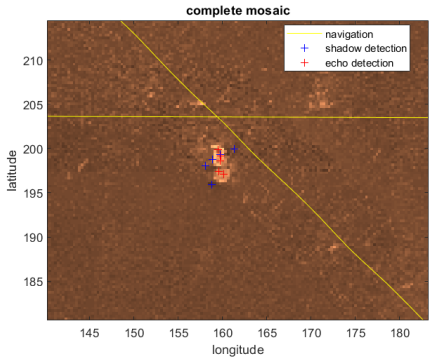
(a) All the landmarks detected by the integral image detector.



(b) Zoom on some detections.



(c) Landmarks kept for the revisit with a strong echo.



(d) Zoom on a single landmark. Notice that this landmark is described by several red crosses since the registration has not been realized.

Figure 2.30: Filtering step on real data. The complete mosaic of some sonar passes on the survey area. The sonar images have not been registered.

The automatic detection (may be manual) of the landmarks and the filtering step are nec-

essary since it provides the map for the planning algorithm that will be proposed in Chapter 6. This map is adapted to the revisit system. Indeed, the revisit vehicle will be able to relocate its position thanks to these geolocalized landmarks. Moreover, the target is also detected automatically or manually and is an input of the planning algorithm.

2.7 Conclusion

In this Chapter, the sonar imagery has been introduced with specific details on sidescan sonar and FLS imagery. The sidescan sonar is generally used for survey mission to cover a large area on the seabed by constructing an acoustic image of the seabed by using the displacement of the sonar and providing an high resolution sonar image. Some automatic algorithms have been developed to detect objects such as mines in the context of MCM missions by considering firstly the shadow created by the object due to the sonar geometry. Later more complex algorithms have been developed to take into account the environment surrounding a mine. The seabed is composed of many landmarks that can be extracted in sonar images and could be used to relocate a robot navigating in underwater environment. Renavigating on a *a priori* surveyed area may lead to different data in case of heterogeneous sensors or degraded sonar image. A simple filtering step has been proposed to keep only the landmark that may be detectable by a revisit system considering its performance, the dimension and the nature of the landmark detected during the survey mission. These filtered landmarks are considered as geolocalized landmarks in the rest of the document, and will be assumed that they are detectable by the revisit system. These landmarks can be of any shape. A motion strategy has then to be found to revisit the potentially dangerous object by revisiting some of these geolocalized landmarks.

Chapter 3

Motion planning

3.1 Introduction

This Chapter provides a state-of-the-art in the motion planning. The terms *motion planning* and *trajectory planning* define the conversion of high-level tasks from humans into low-level descriptions of how to move. It usually refers to the well known *Piano Mover's Problem* or can be transposed in everyday life when moving a sofa to another room requires a plan before starting to carry the sofa. Therefore, motion planning consists in determining a sequence of robot configuration that will result in completing a task mission. In many Autonomous Underwater Vehicle (AUV) applications, a plan is defined by a set of waypoints to be visited. These waypoints are defined assuming a closed loop controller is able to stabilize the vehicle and track the reference path. These waypoints can be pre-planned *a priori* as for survey mission with *boustrophédon* pattern [21] or can be updated during the mission to revisit some targets for example [337]. Alternative approaches to waypoints could be isobath following [139], maintaining an altitude over the seafloor, avoiding obstacles, following a structure such as a pipe... In these cases an interaction with the environment is necessary to achieve an autonomous mission. Many AUV applications have a reliable navigation system based on an Inertial Navigation System (INS) coupled with a Doppler Velocity Log (DVL) for position estimation and waypoints following, but the cost is expensive. Even if this system is not drift free, it can achieve long survey mission and an interaction with the environment would even increase its autonomy. The development of low-cost systems such as the mine killer in Mine Counter Measure (MCM) context enhances the improvement of new localization techniques and the necessity to include uncertainties at the planning phase. These uncertainties can come from imperfect knowledge of the environment, imperfect motion and imperfect sensing.

Any planning algorithm is defined by a *state space* that describes all possible states of the robot, it is commonly known as configuration space $\mathbb{X}$. This configuration space can be partitioned into a free configuration space $\mathbb{X}_{free}$ that the robot is able to reach without colliding obstacles and an obstacle configuration space $\mathbb{X}_{obs}$ (occupied configuration space) that results in a contact with an obstacle. The following relations are always true: $\mathbb{X} = \mathbb{X}_{free} \cup \mathbb{X}_{obs}$ and $\mathbb{X}_{free} = \mathbb{X} \backslash \mathbb{X}_{obs}$. The state of the system can be defined by a position, an orientation, a velocity, etc... It has the dimension of the number of degrees of freedom of the robot. To fully describe an AUV, six degrees of freedom are required: $q = (x, y, z, \phi, \theta, \psi)$ where x, y and z represent the 3D position of the robot and ϕ, θ and ψ are the Euler angles. In this thesis, it will be assumed a 2D environment, then the system can be described by $q = (x, y, \psi)$ with $\theta = \theta_{ref}$, $z = z_{ref}$ and $\phi = \phi_{ref}$. The aim of the robotics path planning is to compute a continuous function $f : [0, 1] \rightarrow \mathbb{X}_{free}$ connecting an initial state $q_{start} = f(0)$ to a goal state $q_{goal} = f(1)$. The initial and goal state can obviously be a set of states, and are not reduced to singleton.

The following properties are inherently used to define any planning algorithms:

- feasibility: find a feasible plan (collision free path) that leads the robot to the goal state regardless its efficiency.
- optimality: find a feasible plan that optimizes (minimizes or maximizes) some objectives and reaches the goal state.
- completeness: it is complete if it always finds a solution if one exists or determines that no solution exists in finite time.

Sometimes a planning algorithm is defined as *online* or *offline* if the plan is modified during the mission or precomputed before. The *workspace* refers to the physical world, it could the two or three dimensional Euclidean space in which the robot is moving.

A robot is defined by a state transition equation:

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

where x belongs to the state space.

In motion planning, some actions u are applied to make evolving the system from an initial state $x_{init} \in \mathbb{X}_{init}$ to a final state $x_{goal} \in \mathbb{X}_{goal}$. A robot has to deal with configuration constraints by the presence of obstacles or joint limits such as tractor-trailer angles for example. The dynamical constraints restrict the value of differential quantities such as velocity, acceleration or path curvature...The dynamical bounds are defined as inequalites on the maximum allowed speed or path curvature for example. The differential constraints are equalities that restrict the velocity to a reduced set of possible state space directions such as railway tracks or rolling constraints. Generally the differential constraints are classified as holonomic when they can be fully integrable and nonholonomic constraints when they are not completely integrable [311]. In most planning algorithms, the mechanics of a robot is not considered [183]. Nonholonomic planning was introduced in [180] to describe the problem of motion planning for a wheeled mobile robots. In other words, a system is said to be holonomic when the number of controllable degrees of freedom is equal to the total number of degrees of freedom (dimension of the state space), otherwise it is nonholonomic. Omnidirectional wheels are an holonomic system contrary to cars or tricycles that are not. In some ways, a robot that can move in any direction is holonomic and a robot that has a limited path curvature is nonholonomic. Nonholonomic planning is more complex than holonomic planning and can be computationally demanding.

Planning in discrete and continuous state space is possible through techniques that will be described in this Chapter. When some algorithms are dedicated to nonholonomic planning, it will mentioned with few references.

3.2 Graph-search

Firstly, the main graph-search algorithms are presented as they will be used in some planning algorithms to find a feasible or optimal path between the initial configuration and the final configuration. When looking for a feasible path, the breadth-first and the depth-first methods can be mentioned but will not be discussed here.

3.2.1 Definition of a graph

A graph is generally defined as an ordered pair (2-tuples) $G = (V, E)$ with V a set of vertices called nodes and E a set of edges. This latter is associated to two nodes in the set V. The graph can have undirected edges when an edge coming from A and going to B is identical to coming from B and going to A, and directed edge when the inverse case is not possible. The nodes of the graph refer to a discrete representation of the robot configuration and the edges to elementary motions between two states (nodes) inside this graph. Once a representation of this graph is built, a graph-search strategy enables to find a path in this graph between the initial state and the final state of the robot. The path is then a concatenation of states inside the graph with edges linking the different states corresponding to elementary motions. The path is found based on an optimality criteria that can be the length of the path for example. This optimal criteria is to minimize or maximize a cost function. A famous graph example is the graph of the cities and the roads, which is an undirected graph. The cities represent the nodes of the graph and the roads the edges between the cities. The length of the roads between the cities are mentioned in the edges. Looking for the shortest path in your GPS between a starting location (Brest) to a final destination (Paris) is a graph-search to find the path that minimizes the distance travelled.

3.2.2 Dijkstra

Dijkstra algorithm [77] is the most famous graph-search algorithm that optimizes a cost function. A description of this algorithm is provided in 1. The graph $G = (\mathbb{S}, \mathbb{A})$ is defined by a set of nodes $\mathbb{S}$ corresponding to possible states of the robot and a set of edges $\mathbb{A}$ corresponding to a motion command between two nodes (states of the system) and a weight (for example the euclidean distance between the two nodes). The initial state is denoted as x_{init} and the final state to reach is x_{goal}. A queue $\mathbb{Q}$ is defined that contains at the beginning all the nodes to be visited excepted x_{init} which is the starting node of the algorithm. An initialization step is described in Algorithm 1 between the lines 2 and 5, where an infinite cost is defined for all nodes excepted the starting node that has a null cost. The cost function is called f. It stores the different costs updated at every nodes. The predecessor $prev$ of each node are not yet known. Then the algorithm consists in visiting all the nodes in the queue $\mathbb{Q}$ by selecting the node u in line 7 with the minimum cost defined by Algorithm 2. This node u is then removed as it is visited (line 8 in Algorithm 1). For each neighbour node v of u a test with the cost (weight) of the edge inside $\mathbb{A}$ added to the cost at u is done to check if a shortest path can be found. If it is true, the cost (line 11 Algorithm 1) and the predecessor (line 12 Algorithm 1) are updated. The algorithm stops when no more nodes are in the queue $\mathbb{Q}$. Finally the path is returned by Algorithm 3 by starting from the final node x_{goal} and by using successively the predecessor until it reaches the starting node x_{init}.

Actually it is not necessary to wait that the queue $\mathbb{Q}$ is empty, if the final node x_{goal} is visited the algorithm can stop. The Dijkstra search is an isotropic search due to the visit of all nodes in the graph with the minimum cost at each time. Figure 3.1(a) depicts the Dijkstra graph search where a grid is considered. Each point in the grid is surrounded by 8 neighbours excepted for the points on the boarder and around obstacles shown in black. The green points are the visited nodes of the graph and the path minimizing the distance is represented in red.

3.2.3 A*

A* algorithm [124] is an updated version of the Dijkstra algorithm where an heuristic is used to obtain a result more fastly. This algorithm is choiced when the computing time is preferred

Algorithm 1 Dijkstra(**in** : $G(\mathbb{S}, \mathbb{A})$, x_{init}, x_{goal}, **out** : path[])

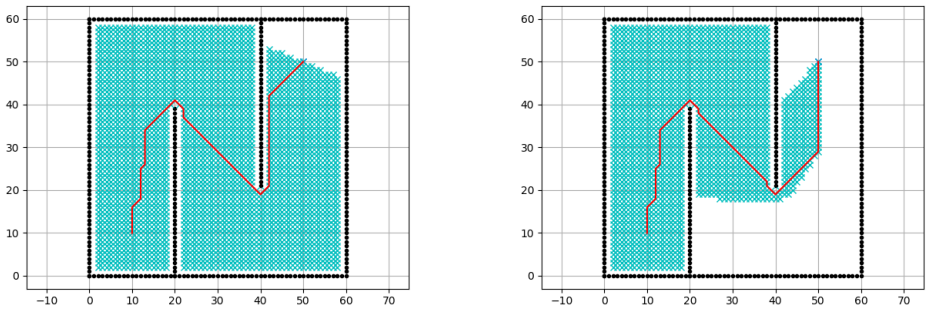
```
1: Q = S\{x_init}                                  ▷ Q is the queue
2: for all x ∈ S do                                ▷ Initialization
3:     f[x] ← ∞
4:     prev[x] ← undefined
5: f[x_init] ← 0
6: while Q ≠ ∅ do
7:     u ← DijkstraMin(Q)                          ▷ Node with least distance will be selected
8:     Q ← Q\{u}                                   ▷ u is removed from the queue
9:     for all neighbours v of u do                ▷ v is still in Q
10:        if d[u]+cost(u,v)<d[v] then             ▷ A shorter path going to v is found
11:            f[v] ← f[u] + cost(u, v)
12:            prev[v] ← u
13: Return DijkstraPath(x_goal,prev[])
```

Algorithm 2 DijkstraMin(**in** : $\mathbb{Q}$, **out** : u)

```
1: d_min ← ∞
2: u ← ∅
3: for all v ∈ Q do
4:     if d[v] < d_min then
5:         u ← v
6:         d_min ← d[v]
7: Return u
```

Algorithm 3 DijkstraPath(**in** : x_{goal}, prev[], **out** : path[])

```
1: path[] ← ∅
2: u ← x_goal
3: while prev[u] is defined do
4:     insert u at the beginning of path
5:     u ← prev[u]
6: Return path[]
```



(a) Dijkstra propagation.

(b) A* propagation.

Figure 3.1: Comparaison between Dijkstra and A* algorithm. Examples computed based on the work in [283].

to the optimality of the path. The heuristic enables to visit nodes that are closer to the final node x_{goal} by taking into account the cost at these nodes. The A* algorithm uses two lists: the OPEN list which gathers the candidate nodes and the CLOSED list that contains the visited nodes. The OPEN list is similar to the queue $\mathbb{Q}$ in the Dijkstra's algorithm. Two functions g and h are defined:

- $g(x_i)$: real cost from the starting node x_{init} to the node x_i.
- $h(x_i, x_{goal})$: estimation of the cost from the node x_i to the goal node x_{goal}.

The function $f(x_i) = g(x_i) + h(x_i, x_{goal})$ is an estimation of the cost from x_{init} to x_{goal} by going through the node x_i. $h(x_i, x_{goal})$ can be written $h(x_i)$ since the heuristic is always computed according to the final node. This heuristic can be the Euclidean distance. The function f is used in the algorithm to select the node to be visited. A pseudo-code is proposed in Algorithm 4.

The Dijkstra's algorithm can be seen as an A* algorithm with a null heuristic, i.e $f(x_i) = g(x_i)$ and $h(x_i) = 0$. Therefore the A* algorithm works similarly. An initialization is proposed in Algorithm 4 between the lines 1 to 7 similarly to the initialization of the Dijkstra's algorithm. Then all the nodes in the OPEN list are visited based on the minimum of the function f, similarly to Algorithm 2, until the current node is the goal node x_{goal}. If this condition is true (line 10), the same algorithm for the construction of the path as the Dijkstra's path (Algorithm 3) is used to construct the path (line 11 in Algorithm 4). Figure 3.1 depicts the difference in the propagation search between the two algorithms where the A* search is more straight-forward to the final node. The resulting path is slightly different to the Dijkstra's path.

An admissible heuristic never overestimates the cost from a current node to the final node, it is always less or equal than the true cost between this node and the final node. A monotonic heuristic satisfies the triangle inequality $h(x) \leqslant d(x, y) + h(y)$ [228].

3.2.4 D*

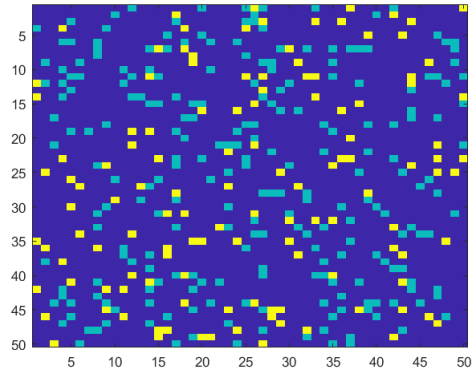
If the A* algorithm is well adapted to find fastly a path from an initial state to a final state if the environment is *a priori* known, when dealing with dynamic or partially known environment, an A* algorithm should be computed again. Indeed when the robot faces an unknown situation along the path such as an unknow obstacle, an A* algorithm is run again to find a path between the current state and the goal state, which can be time comsuming. D* algorithm [296] has

Algorithm 4 A*(**in** : $G(\mathbb{S}, \mathbb{A})$, x_{init}, x_{goal}, **out** : path[])

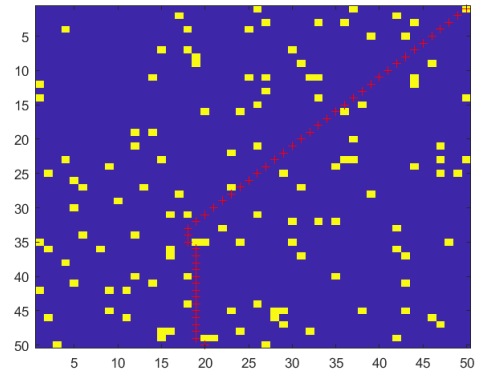
$OPEN \leftarrow \{x_{init}\}$
$CLOSED \leftarrow \emptyset$
$prev \leftarrow$ empty map
$g \leftarrow$ map with ∞ value
$f \leftarrow$ map with ∞ value
$g[x_{init}] \leftarrow 0$
$f[x_{init}] \leftarrow h(x_{init}, x_{goal})$
while $OPEN \neq \emptyset$ **do**
    $u \leftarrow argmin_f\{OPEN\}$ ▷ Node with least distance will be selected according to f
    if $u = x_{goal}$ **then**
        Return ReconstructPath(x_{goal},$prev$[])
    $OPEN \leftarrow OPEN \backslash \{u\}$
    $CLOSED \leftarrow CLOSED \cup \{u\}$
    for all neighbour v of u **do**
        if $v \in CLOSED$ **then**
            continue
        $testscore \leftarrow g[u] + cost(u, v)$
        if $v \notin OPEN$ or $testscore < g[v]$ **then**
            $prev[v] \leftarrow u$
            $g[v] \leftarrow testscore$
            $f[v] \leftarrow g[v] + h(v, x_{goal})$
            if $v \notin OPEN$ **then**
                $OPEN \leftarrow OPEN \cup \{v\}$
Return No solution

been developed to plan a global path as previously with known information and then locally circumvent unknown obstacles when detected by a sensor by modifying locally the path. It is a generalization of the A* to dynamic environment. Contrary to A* or Dijkstra, firstly it perfoms the search from the goal to the starting state (backward search). A backward search reduces the influence of changes to be propagated. Originally in the D* algorithm, the backward search was isotropic, like running a backward Dijkstra. Then the focussed D* algorithm was improved to focus the propagation [297] such as a backward A*. A D* Lite version has then been developed in [164] based on a Lifelong Planning A* (LPA*)[163] that has similar properties to A* and rigorous proofs of efficiency. Moreover the implementation requires fewer lines of code. The reader must refer to the work in [164] for a detailed version of the algorithm. However an example of the D* Lite is provided in Figure 3.2 where a complete implementation has been realized. The map is composed of a 50×50 grid with a resolution of one meter. The states of the system is then discretized as 2500 possible (x, y) positions. The yellow squares represent *a priori* known obstacles in this environment. The cyan squares are unexpected obstacles. Based on the *a priori* known information in the terrain, a backward search provides a path painted in red in Figure 3.2(b) where the initial state is at the bottom. The robot is equipped with a sensor that can see at 3 meters around. When the robot is following the path, some unexpected obstacles (cyan squares) are detected and the path is locally updated to avoid these obstacles. The final path avoiding these unexpected obstacles is depicted in red in Figure 3.2(c). The unexpected obstacles detected are marked by a blue asterisks. One may notice the detour at the beginning of the mission and then it follows the path normally.

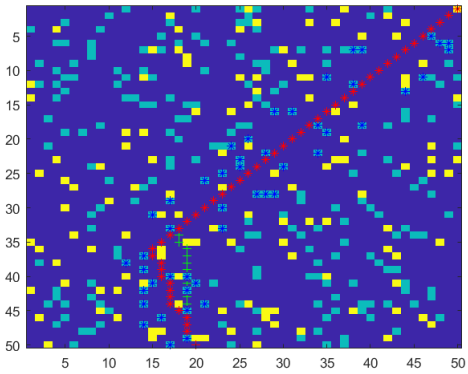
The D* Lite algorithm is pretty interesting due to the fast update of the path when unexpected events occur in the path planned. It has been successfully implemented in [110] that takes into account the uncertainty propagation along the path. Moreover in [109], the algorithm can find fastly a new path when the landmark is not detected by joining another landmark region. This will be more discussed in planning with uncertainty section.



(a) Real map with yellow squares as known obstacles and cyan squares as unknown obstacles.



(b) Backward search for a path in red based on the *a priori* information about the terrain (obstacles in yellow). The initial state is at the bottom and the goal state in the top right corner.



(c) Path updated with D* Lite based on the detection of the new obstacles in cyan. The modified path is represented in red and the cyan obstacles marked by blue asterisks are obstacles detected by the sensor. Old path is in green crosses.

Figure 3.2: D* Lite algorithm.

3.3 Planning

Many motion planning methods are today available by considering obstacles. Firstly, a state-of-the-art of the algorithms is proposed here where the uncertainties, that could come from the motion of the robot, the map or the measurements are not considered. Indeed, at the beginning the algorithms were developed without considering uncertainties and then many of them have been adapted to take into account some uncertainties.

3.3.1 Combinatorial planners

The combinatorial planners are among the first planner and are exact planner contrary to sampling-based and grid-based planners due to the fact it finds paths in the continuous configuration space without any approximations and considering the boundary of polygonal obstacles

$\mathbb{X}_{obs}$[1]. Moreover these algorithms are complete meaning it will find a solution if one exists. These methods can be performant when dealing with 2D environment and a robot able to move in translation. Most of these methods cannot be extended to high dimension. All combinatorial motion planning techniques consist in building a roadmap along the way to solve queries. The graph G is called a roadmap and it satisfies the *accessiblity* and *connectivity* properties. The roadmap provides therefore a discrete representation of the continuous motion planning problem. The query (x_{start}, x_{goal}) is solved by connecting each query point to the roadmap and a simple graph search strategy presented in Section 3.2 can be used.

If the robot is not considered as a point robot, the Minkowski difference has to be applied to the robot and the obstacle environment $\mathbb{X}_{obs}$.

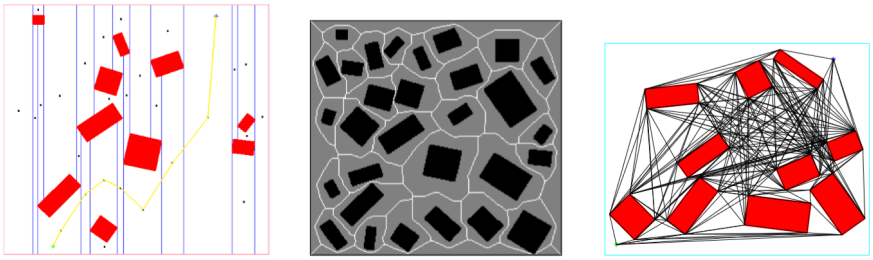
The first algorithm is called cell decomposition and especially vertical cell decomposition [52] which partitions the space into a set of smaller cells by a left-to-right ordering of the vertices of all polygonal obstacles and finally a sweep from left to right adding a vertical line when it meets a vertex. A single sample point such as the centroid of the cell is chosen to represent each cell and the roadmap is obtained by connecting the samples from neighbouring cells. A representation of such roadmap is presented in Figure 3.3(a). The major differences between cell decomposition are the methods to generate the cells and their shapes. The *Boustrophédon* cell decomposition has been introduced in [54].

Maximum Clearance roadmaps [37] present the advantage of minimizing the risk of collision. It usually refers to Generalized Voronoi Diagram (GVD). Each point along a roadmap edge is equidistant from two points on the boundary of $\mathbb{X}_{obs}$. The roadmap vertices are then the intersections of at least two roadmap edges. A simple way to compute GVD is to assign a circle with a small radius to each point of a grid representing $\mathbb{X}_{free}$ and then increase the radius of the circle until it reaches one obstacle. If it reaches only one obstacle, it does not correspond to the roadmap. A representation of the result is depicted in Figure 3.3(b).

Shortest-path roadmaps refer to reduced visibility graph [179] and is close to the principle of GVD because the shortest path tends to graze the corners in $\mathbb{X}_{obs}$. The robot is actually allowed to touch or graze the obstacles but not to collide it. Therefore the actual path is slightly longer that the path found. Considering an obstacle environment, a reflex vertex is defined as a polygon vertex with an interior angle greater than 180°. All vertices of a polygon obstacle are reflex vertices. These reflex vertices correspond to the nodes V of the graph G. Firstly, the reflex vertices that form edges of a polygon obstacle are added to the roadmap. Secondly, if a line (called bitangent) can be drawn between a pair of reflex vertices without intersecting any obstacle, it is added to the roadmap as an edge. Finally the roadmap is obtained with graph-search strategy as presented in Section 3.2. Figure 3.3(c) presents an application of this kind of path.

These algorithms requires polygon obstacles and if the obstacles are more complicated than squares it can be hard to find a solution. Moreover this assumption in underwater environment is not acceptable due to the unstructured aspect. However these algorithms could be interesting for a revisit of different textures areas on the seabed to make some relocation if they are *a priori* detectable by a sensor. In this case $\mathbb{X}_{obs}$ is not considered anymore and the robot can cross the frontier of a texture.

[1] A polygonal region has no holes.



(a) Cell decomposition method. (b) Generalized Voronoi diagram. (c) Visibility graph.

Figure 3.3: Main combinatorial planners. Pictures extracted from [243] and [183].

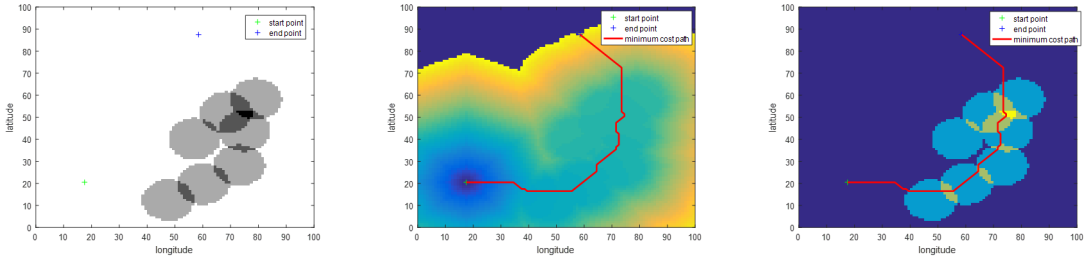
3.3.2 Sampling-based planners

In high dimensional problems, sampling-based methods appear as an interesting alternative method where computing the configuration space is not feasible [179]. Moreover when the environment is partially known or uncertain combinatorial methods are not well suited. Sampling methods avoid to construct explicitly $\mathbb{X}_{obs}$ and conduct a search in the sampled configuration space. A collision detector module enables to remove dangerous path. Compared to combinatorial methods, sampling algorithms are not complete due to the weaker guarantee that the problem will be solved. However, the higher the density of samples is, the higher the probability of finding a path converges to one. Deterministic approaches such as grids are resolution complete. Methods based on random sampling such as Rapidly-exploring Random Tree (RRT) are probabilistically complete. Even if these algorithms are not complete, the probability that the planner fails to return a solution, if one exists, decays to zero as the the number of samples approaches infinity [17]. The convergence of such algorithms is difficult to establish. Some of the main algorithms are proposed here. Some issues and proofs for the two main random sampling-based planners, i.e. Probabilistic Roadmaps (PRM) and Rapidly-exploring Random Tree (RRT), are proposed in [153][200].

3.3.2.1 Grid-based cost map

In grid-based path planning, a regular grid [109][228] is superimposed on the environment the robot will be navigating. The costmap contains some cost proportional to the traversability of the environment at that particular place called cell. A typical cost map representation is the Occupancy Grid [86] introduced in Chapter 1. In indoor environment, the floor is usually considered uniformally traversable and obstacles are represented as binary cost regions. A cost of one can be used for free space and infinity for obstacles. In outdoor environment, height information are incorporated in the map [226]. In [109], based on the terrain type such as paved road, grass, water, etc... a cost is associated. Roads are preferred for driving robot compared to water. In [310], each 2D location was classified as occupied, drivable or unknown using a probabilistic test on the height distribution of neighbouring points.

The aim is to find the minimum cost path based on the value in the costmap. This grid is considered as a graph G where the set of nodes (vertices) V corresponds to the center or the corners of the grid cells. Usually each node has a 4 or 8-connexity which imposed a small number of neighbours for each node. This 4 or 8-connexity defines the edges in the graph. The problem of this reduced connexity is the restiction on motion with increments of 45 degrees in case of 8 connexity. Moreover the nonholonomic constraint [180] of a vehicle cannot be taken into account in grid-based planners. The cost function takes as input the value in the cost map and a graph search strategy as presented in Section 3.2 that optimizes this cost function in



(a) Relocation areas based on the landmark locations and the range of the sensor.

(b) Dijkstra's search.

(c) Final path minimizing a cost function.

Figure 3.4: Grid-based path planning with a cost map.

then computed to find the shortest path.

Considering an environment composed of punctual landmarks (certainly located) and a robot equipped with an isotropic sensor that can sense the environment up to 10 meters, a grid-based path planning is proposed in Figure 3.4. As the landmarks are certainly located they can for example reduce the uncertainty in the robot configuration and therefore a cost map based on the density of the landmarks taking into account the ability of sensing is proposed. The cost for each displacement proposed is:

$$cost = e^{-n} d_{travelled} \tag{3.2}$$

with n the number of landmarks detectable in the cell and $d_{travelled}$ the distance travelled from cell to cell. The resolution is fixed at $1m$, so the distance travelled is 1 or $\sqrt{2}$ meters. An exponential function is used to prioritize regions where the number of landmarks is high. The path minimizing this cost function is obtained with a Dijkstra's graph search strategy and is depicted in red in Figure 3.4(c). The idea behind this cost function is to go through landmarks field to reduce a potential uncertainty on the robot configuration and make some relocation process. The uncertainty on the robot has not been considered here.

Path planning with cost map will be used again when planning with uncertainty will be presented.

3.3.2.2 Fast Marching

The fast Marching (FM) method [290] is a particular case of the Level Set methods [291] enabling to track and model a physical wave interface propagating. It computes the time u that a wave needs to reach every point of the space and can be generated from one or several sources ($T = 0$). At a given point x, the motion of the front is described by the Eikonal equation [290]:

$$\begin{aligned} |\nabla u(x)| = 1 \backslash f(x) \quad \text{for} \quad x \in \Omega \\ u(x) = 0 \quad \text{for} \quad x \in \partial\Omega \end{aligned} \tag{3.3}$$

with $u(x)$ the minimum amount of time it would take to reach $\partial\Omega$ starting from x with the speed $f > 0$ in the normal direction at the point x. ∇ denotes the gradient and $|.|$ the Euclidean norm. This describes the evolution of a closed surface as function of the time u. The FM method is a numerical method to solve the boundary value problem of this Eikonal equation. It uses a first order numerical approximation of the Eikonal equation on a 2D grid map in [291] inspired by the work [312] where the numerical scheme converges to a correct continuous

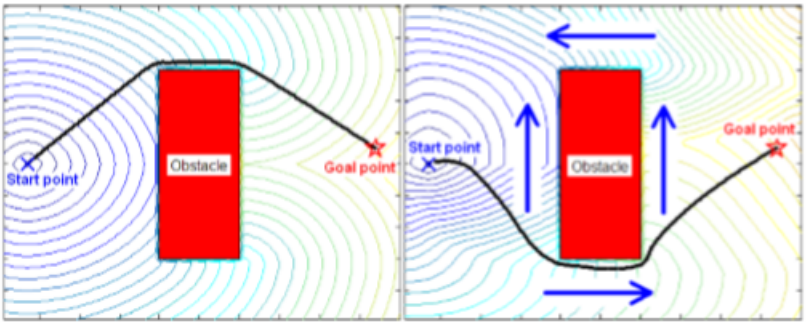


Figure 3.5: On the left an isotropic FM as the Dijkstra's algorithm and on the right the proposed anisotropic version in [252] taking into account the currents represented by blue arrows around an obstacle.

solution. The concept is basically a continuous version of the Dijkstra's algorithm to find a shortest path. This method aims at finding a control law for a given system such that a certain optimality criteria is achieved: this is the definition of optimal control. The Eikonal equation is a special case of the Hamilton-Jacobi equation.

Fast marching method has been used for AUV path planning in underwater environment in [253][252] where it proposes a FM* version of FM method as a continuous version of A* with an heuristic. This anisotropic search comes from the dynamic nature of ocean currents. The cost function takes as input the directional constraints imposed by the currents. And finally the vehicle kinematics have been taken into account to propose paths with some curvatures. The results of this planner is proposed in Figure 3.5

The ocean currents have been taken into account in [201] with level-set methods to propose a path planning method for gliders.

3.3.2.3 Potential Fields

Potential functions [160][162] can be generated when the obstacles and the goal location are *a priori* known. The paths can then be obtained by following the negative gradient of the potential function. The potential function is mainly composed of two terms: the goal is considered as an attractive force and the obstacles as a repulsive force. The potential function at any point in the space is then the sum of the positive and the repulsive force [55]. The attractive force is highly dependent on an heuristic [179]. This heuristic can be difficult to determine. Usually an high resolution grid is used for the state representation where the repulsive and the attrative field can be computed, and finally giving the combined field. Parameters can be defined to control the distance to obstacles and the attractivity of the goal location. The negative gradient corresponds to the direction that is progressing towards the goal. The main drawback of this method is the potential presence of local minima that do not correspond to the goal location, where the sum of the vectors is zero. Some methods are used to escape from these local minima by choosing a random direction during a certain number of iterations. This is called random walk and is similar to stochastic search and simulated annealing. It refers to Randomized Potential Fields or Randomized Path Planner [179][156]. However it does guarantee that an optimal path is found based on the potential function. Even if it can solve problems in high dimension, the random walk may generate long path that are not always adapted to the problem and the definition of the potential function requires some parameters to be well adjusted for increasing the effectiveness of the heuristic.

Potential fields method has been applied for AUV path planning in [332] around obstacles to provide a safe path.

The wavefront planner [18] works similarly to the potential fields. The potentials at each configuration in the free space are computed by simulating a wave propagating from the goal location to the start location. The space is usually represented as a grid and the nodes are marked as visited or unvisited based on the wave propagation. Iteratively it finds unvisited neighbours of visited cells and increments the value of one greater than the visited cell. When the start node is reached, a gradient descent from the start location enables to get the path. Contrary to potential fields, no local minima is possible. However, it can be computationally more expensive and can only be computed off-line.

3.3.2.4 Rapidly-exploring Random Tree (RRT)

The Rapidly-exploring Random Tree (RRT) algorithm, introduced in [182], builds a tree of feasible trajectories from an initial configuration $x_{init} \in \mathbb{X}_{init}$, called the root. It is a single query algorithm. A presentation of the algorithm is proposed in Algorithm 5. The algorithm is initialized with a graph that includes only the initial state and no edges. At each iteration, a sample $x_{rand} \in \mathbb{X}_{free}$ is generated. A connection is tested with the nearest neighbour $v \in V$ in the tree (graph) to this new sample x_{rand}. An input $\mathbf{u}$ is selected among a set of inputs $\mathbb{U}$ to join $x_{nearest}$ to x_{rand}. By integration of the evolution equation $\dot{x} = f(x, \mathbf{u})$ during a time interval Δt with $\mathbf{u}$ constant a new state x_{new} is generated. If the path from $x_{nearest}$ to x_{new} is feasible, i.e. collision free, based on the input $\mathbf{u}$ and the time interval Δt an edge is created. In case of an holonomic robot, $f(x, \mathbf{u})$ can be replaced by $\mathbf{u}$. The RRT algorithm is probabistically complete as the number of samples tends to infinity [166][184].

Algorithm 5 RRT (**in** : $x_{init} \in \mathbb{X}_{init}$, $x_{goal} \in \mathbb{X}_{goal}$, $\mathbb{X}$, $N \in \mathbb{N}$, Δt, **out** : G)

1: $G.init()$
2: $G.addVertex(x_{start})$
3: $i \leftarrow 0$
4: $x_{test} \leftarrow \emptyset$
5: **while** $x_{test} \notin \mathbb{X}_{goal}$ and $i \leq N$ **do**
6: $\quad x_{rand} \leftarrow \text{RandomState}(\mathbb{X}_{free})$
7: $\quad x_{nearest} \leftarrow NearestNeighbour(G, x_{rand})$
8: $\quad u \leftarrow SelectInput(x_{nearest}, x_{rand})$
9: $\quad x_{new} \leftarrow Steer(x_{nearest}, u, \Delta t)$
10: $\quad i \leftarrow i + 1$
11: $\quad$ **if** $CollisionFreePath(x_{nearest}, x_{new}, u, \Delta t)$ **then**
12: $\quad\quad G.addVertex(x_{new})$
13: $\quad\quad G.addEdge(x_{nearest}, x_{new}, u)$
14: $\quad\quad x_{test} \leftarrow x_{new}$

RRT algorithms are well-suited for handling differential constraints present in nonholonomic robot [246][247][34][183][181][184]. An RRT algorithm has been used for AUV mission planning in [128].

Different version of RRT exist today to increase the performance of the original RRT. Instead of taking a random sample from $\mathbb{X}_{free}$ through the *RandomState* function, RRT Goalbias [184] returns the final configuration $x_{goal} \in \mathbb{X}_{goal}$ with a probability p and others samples from

$\mathbb{X}_{free}$ with a probability $(1-p)$. Even if p is small ($p \approx 0.05$), the algorithm converges faster. If p is too big, it may be trapped in local minima as in randomized potential fields method. RRT Goalzoom [184] is an improvement of RRT Goalbias where a sample is selected with the probability p in a disk centered in x_{goal} with a radius equals to the smallest distance between x_{goal} and the RRT nodes in the tree at any iteration. The radius of the disk decreases until converging to x_{goal}. Similarly it may be trapped in local minima. Some of the sampling methods presented in Probabilistic Roadmap planners (PRM) may be used. A bidirectional RRT, called dual-RRT, is also proposed by growing two RRTs, one from x_{init} and one from x_{goal}. A solution is found when the two RRTs meet.

A Rapidly-eploring Random Graph (RRG) has been proposed in [153] to build a connected roadmap containing potential cycles. Contrary to RRT, when a new node x_{new} is added to the vertex set (V) of the tree, connections to other vertices of the graph in a ball of variable radius centered at x_{new} are attempted. If a connection is successful, an edge is created. Consequently the RRT graph (directed tree) is a sub-graph of the RRG graph (undirected tree). They share the same node set but the edge set of the RRT is a subset of the RRG edge set. Some cycles may then belong to the RRG graph. K-nearest RRG can also be considered with a variable K.

A more efficient version of RRT called RRT* is proposed to converge to the optimal path in [153] by removing edges that are not part of a shortest path from the root (x_{init}) of the tree to any nodes in the graph and consequently avoiding the formation of cycles present in RRG. Actually it is an improvement of RRG. Similarly a variable radius of a ball or a variable K-nearest neighbour can be considered.

3.3.2.5 Ariadne's Clew algorithm

The Ariadne's Clew algorithm [208][26] is a single query algorithm that grows a tree as a RRT algorithm from an initial configuration x_{init} toward the goal configuration x_{goal}. It is composed of two phases: EXPLORE and SEARCH. The exploration phase consists in proposing a reachable sample x_{new} far from the other nodes in the graph G to explore at the maximum the configuration space. The search phase then tries to reach the goal state x_{goal} from the proposed sample x_{new}. A genetic algorithm is used to optimize the exploration by proposing a sample reachable and far from the others. However this method requires a lot of parameters to be defined and the choice of an heuristic is difficult.

A similar algorithm was proposed in [131] where it tries to grow a tree in unexplored areas in $\mathbb{X}_{free}$. It picks a node that has a probability inversely proportional to the number of samples in the graph G (tree) in a ball of a certain radius centered on this node. Consequently, the most isolated point are selected.

3.3.2.6 Probabilistic Roadmap planner (PRM)

Probabilistic Roadmap Planner [158], also called Probabilistic Path Planner [302], builds probabilistic roadmaps, known as a topological graph, by randomly selecting configurations in the free configuration space and connecting the samples by simple feasible paths. As mentionned earlier, this method is probilistically complete. The algorithm works in two phases: the roadmap construction phase and the query phase [183].

The first phase, also known as preprocessing or learning phase, samples the configuration space according to a sampling strategy (some strategies will be explained later) and builds an

undirected graph $G = (V, E)$ with samples in the free space $\mathbb{X}_{free}$. The samples (x_{rand}) constitute the vertices V of the graph G. The feasible paths, corresponding to the edges E and known as local paths, connect the different samples by a local planner. This latter simply connects two samples (vertices) if the local path between them does not collide an obstacle (collision-free) and respects the constraints of the robot. In many applications, it selects the shortest path to build the graph between the different samples leading in general to collision-free path. However other methods, presented in [183], can be proposed such as selecting the K-nearest samples, meaning that for a specific sample x_{rand} it looks for the K-nearest samples in the graph that leads to feasible path connected to this sample x_{rand}. Taking the samples in a ball of radius r centered at the sample x_{rand} and having a collision-free relation could be another solution. In this case an upper limit is usually used to prevent from too many connections. Another solution for connecting the samples would be based on the visibility leading to a variant of the PRM called visibility roadmap [293]. In this variant, two kinds of vertices are defined: guards and connectors based on the visibility region defined at each samples.

The second phase is the query phase. It assumes that the graph G is sufficiently complete with samples and edges to answer many queries. These latters give the initial x_{init} and final x_{goal} configurations which are tried to be connected to nodes in the graph with the same local planner that checks the collision and the vehicle's constraints. If both are successfully connected a simple graph search strategy such as the ones presented in Section 3.2 may be used to connect x_{init} to x_{goal}. If the connection fails for both, it cannot be determined if a solution exists and the query fails. This algorithm is said to be multiple-query.

A description of this algorithm is given in Algorithm 6. The construction of the roadmap is mixed with the query phase. The graph G is initialized with the initial x_{init} and final x_{goal} position. A parameter N is fixed by the user to constraint the possible number of samples. The construction of the map is realized in the while loop and the query phase is expressed in the condition 'No path exists from x_{start} to x_{goal} in G' where it checks if the initial and final configurations are linked to any samples in the roadmap (graph G). At each iteration, a random sample is choiced and tested if it belongs to $\mathbb{X}_{free}$. The variable i is incremented only if a sample belongs to $\mathbb{X}_{free}$. For this new sample, it looks for the neighbouring samples in the graph G according to the strategies developed above such as the K-nearest neighbours (K-NN). Finally the *CONNECT* function checks if a link can be made between the new sample and the selected neighbouring samples. A link is possible if it does not collide any obstacles and respect the constraint of the vehicle.

Algorithm 6 PRM (**in** : x_{init}, x_{goal}, $\mathbb{X}$, N(**samples number**), **out** : path[])

```
1:  G.init()
2:  G.addVertex(x_start) and G.addVertex(x_goal)
3:  i ← 0
4:  while No path exists from x_start to x_goal in G and i < N do
5:      x_rand ← random samples ∈ X
6:      if x_rand ∈ X_free then
7:          G.addVertex(x_rand)
8:          i ← i + 1
9:          for all v ∈ neighbourhood(x) in G and v ≠ x do
10:             if connect(x_rand, v) then
11:                 G.addEdge(x_rand, v)
```

PRM has been proposed for holonomic robots in [157]. It has been applied to nonholonomic robots such as car-like robots that can move forwards and backwards or only forwards in [300][172][102]. The probabilistic completeness for nonholonomic robots is proven in [301].

Different variants of PRM exist today and mainly differ in the way of sampling as proposed just below.

How to sample in $\mathbb{X}$?

The sampling strategy has not yet been discussed. Uniform sampling is the main sampling strategy but other methods exist to smartly sample the environment. The other main approaches are importance sampling and adaptive sampling.

Firstly, considering importance sampling methods, sampling on the boundary of $\mathbb{X}_{free}$ is considered in [6] to avoid wasting time with samples in large areas in $\mathbb{X}$. Similarly, gaussian sampling is considered around the boundary of $\mathbb{X}_{free}$ in [30]. In the case of corridors, bridge-test sampling is preferred in [130] to select the middle sample along a line segment where the extremities belong to $\mathbb{X}_{obs}$. On the contrary medial-axis sampling forces the samples to be far from any obstacles in [129]. A sensor uncertainty sampling is proposed in [127] to provide sample x_{new} where greater information gain is expected when a sensor measurement is taken at x_{new}. It should then provide samples where a maximum localization accuracy of the vehicle is possible. This strategy is based on the concept of the Sensor Uncertainty Field (SUF) [304]. SUF will be discussed in motion planning with uncertainty section.

Secondly, adaptive sampling provides samples based on information gained from previous samples. It refers to the visibility roadmap presented in [293] discussed above.

Recently, a more computationally version of PRM called PRM* has been proposed in [153] where some proofs on PRM and PRM* are provided. Basically, it considers a variable radius r when looking for closest nodes in the graph to the tested sample x_{rand}. The radius decreases with the number of samples. The idea of a variable radius has already been proposed in [183]. It also proposed to consider a K-nearest PRM* where the number of neighbours K is not constant and depends on the cardinality of the roadmap.

3.4 Planning under uncertainty

In the previous section, the main motion planning algorithms have been presented where the notion of uncertainties has not been taken into account. These classical path planning algorithms aim at finding the best path between two configurations (or locations) assuming the position of the robot is known at every moment. However many uncertainties could be present: the system itself, the motion of the robot, the sensing or the environment. When a system can not access to an absolute positioning such as Global Positioning System (GPS), an uncertainty on the position is induced. GPS position is not available in underwater environment and if the robot does not use any acoustic positioning system (USBL,...), a drift in position occurs even if the system is equipped with an expensive INS and DVL. When using these latters, it is still possible to plan a mission with the techniques presented above due to the low drift in time but the mission can not be too long due to this inherent drift and ocean currents. However the AUV can make some *online* replanning if possible with a localization process based on a

a priori map or on the map that is building (SLAM). In the case of low cost robot, as for mine identification/neutralization in MCM missions, the drift in position is large and a planning method that does not consider uncertainty will likely fail. Indeed, revisiting a target or reaching a specific position from a far initial configuration might lead to the loss of the robot if it is not equipped by a recovery system. As mentioned earlier, the source of uncertainties could come from the motion, the map and the measurements. A planning method that considers all the three uncertainties is more likely to propose a strategy that will succeed for sure. In real applications, these uncertainties are unavoidable and especially in underwater applications. In the context proposed in this thesis, it is vital to propose a strategy that guarantees the revisit of the designated Mine Like Contact (MILCO) for identification and therefore an overview of the planning methods that consider uncertainties is proposed here with some applications to the context proposed. Moreover, when considering obstacles, the pose uncertainty may lead to collision, and the methods proposed have to guarantee the safety of the robot.

Due to the uncertainties present in the robot's motion, sensory readings and map, the true robot state is not available. One of the previous work on motion planning strategies with uncertainty is the work in [89]. Partially Observable Markov Decision Process (POMDP)[150] is a general framework to deal with these uncertainties.

3.4.1 POMDP problem

A Partially Observable Markov Decision Process (POMDP) is a generalization of the Markov Decision Process (MDP) where both states and actions are uncertain. It models an agent where the system dynamics is handled by a MDP but this agent cannot directly observe the state, it is called partial state observability. The plans are expressed in information state space known as belief space. In other words, it maintains a probability distribution over the set of possible states. If the robot has an imperfect knownledge of its state, selecting the next action must take into account all possible states that are consistent with its observations. Contrary to classical planners that generate a sequence of actions, POMDP produces a universal policy for action selection. A conditional probability function models the state transition from x to x_{new} after some action a according to the Markov property, meaning that the new state only depends on the previous state and the action. A reward function is defined to describe the desired behaviour of the robot. At each time step, the robot takes some action and gets a reward. The goal is to find a policy for selecting actions based on the current belief. An exact solution to the POMDP problem proposes the optimal action for each belief over the configuration state space. This optimality criteria maximizes the expected reward (or cost) of the agent over a possible infinite horizon. Even if some efforts have been made to increase the efficiency [259] of these approaches in large state space, it remains challenging. Due to its significant complexity, solving the POMDP problem is often intractable for realistic problem [150]. Therefore some approximate methods to the POMDP problem appear that carry the robot state and the associated uncertainty.

3.4.2 Some methods...

3.4.2.1 Pre-image backchaining

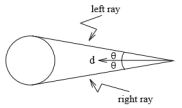
One of the first planning method dealing with uncertainty is called preimage back-chaining [203]. The method consists in planning fine motions in the presence of uncertainty in a polygonal world. A famous example of this approach is the *peg-in-hole* problem where the peg has to be inserted into a hole slightly larger. The idea is to plan a sequence of motion plans that guarantees the reachability of a goal region. This plan is built by starting from the goal region

and then searching backwardly until the initial pose is included in a preimage. It results in N open-loop strategies (composed of closed-loop motions) called f_1, f_2, :., f_N leading to:

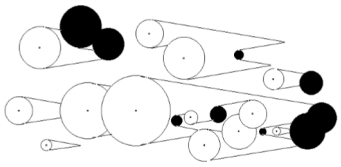
$$f_N(...f_2(f_1(x_{init}))...) \in \mathbb{X}_{goal} \tag{3.4}$$

Pratical techniques to compute the preimages can be found in [179][178]. The notion of preimage has been extended in [186][187][185] to propose a landmark based robot navigation on the concept of preimage backchaining. The method assumes that the uncertainty on the position of the robot can be modelled by a disk with a radius increasing linearly with the distance travelled. Therefore some landmark regions, where sensing is perfect, are defined by disks. The backprojection of a circular region is then a cone in a particular direction. The uncertainty based on the distance travelled is modelled by a directional uncertainty as illustrated in Figure **??**(a). If this backprojection does not collide any obstacles, it considers that if the robot belongs to this backprojection, it can for sure reach this circular landmark region with the particular command associated despite the directional uncertainty as shown in Figure 3.6(b). The cone enables to extend the reachable regions. Based on the backchaining process, it builds a motion plan to reach the goal region assuming also circular obstacles, depicted in Figure 3.6(c). The arrows show the direction the robot has to follow to reach the next landmark region at the point considered.

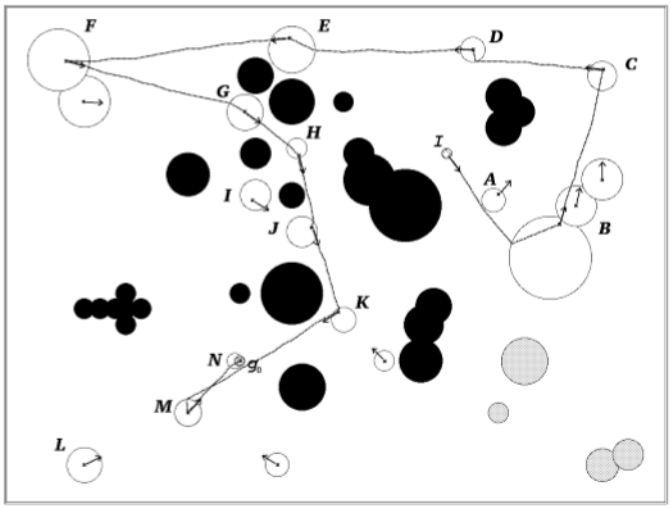
The problem of this method remains in the definition of these safe circular areas where an absolute positioning is available. Moreover the assumption of the linearly growing uncertainty has still to be justified. This assumption is usually made in robotics and underwater robotics. However this method seems interesting in our problem since it enables to find a guaranteed plan under this assumption to reach a goal region. It will be the subject of Chapter 6 where the landmark and goal regions are extended to unstructured shapes highly present in underwater environment based on the measurement abilities of the sensor embedded in AUVs. Moreover the problem will be solved in an elegant set-membership manner. More specifications about the algorithm presented here will be detailed in Chapter 6.



(a) Uncertainty propagation based on uncertain direction θ.



(b) Complex backprojections with obstacles in black.



(c) Plan found to reach the goal region g_0 from the initial position I. The black circles are obstacles and white disks are landmark regions. The directional uncertainty θ is defined at 5°.

Figure 3.6: Preimage backchaining algorithm. Images extracted from [186]

3.4.2.2 Sensory Uncertainty Field

Sensory Uncertainty Field (SUF) has been proposed in [304][303] and could be considered as a registration map.

Indeed, it indicates for every robot configuration, discretised by a grid, a relocation ability by making an exteroceptive measurement for all these configurations based on a sensor model and matching the measured data against an environment model. In other words it indicates the uncertain pose the robot would have if it senses the environment at this particular pose with a given map of the environment. The SUF is generated before the planning algorithm based on simulated and real measurements for a laser/camera range sensor in each configuration. The computation of such a map is proposed in [3] for cameras. Once the map is created, a Dijkstra (or A*) algorithm is applied to find a path minimizing an objective function combining expected errors and path length. However it does not consider the propagation of the uncertainty along the path. It does not guarantee that the landmark considered is reachable and could miss it due to a large uncertain pose of the robot. It proposes only to go through areas where a relocation is possible and to optimize the use of the sensor.

The SUF has been extended in [239][240] by considering the uncertainty along the path and the characteristics of the sensor when measuring the environment with different types of landmarks. It enables the localization on walls and other polygonal obstacles with cameras or

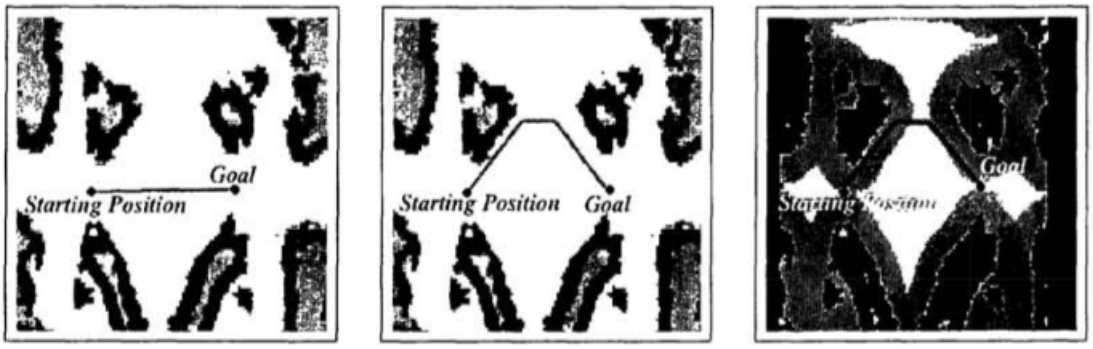


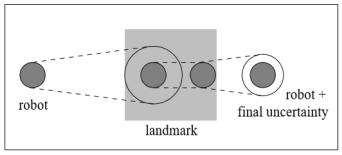
Figure 3.7: The black elements depict the coastal environment. On the left, a path minimizing simply the path length. In the middle, a path optimizing the path and the relocation ability where the gray areas indicate a possible relocation. On the right, the SUF is represented. Picture extracted from [278].

other sensors.

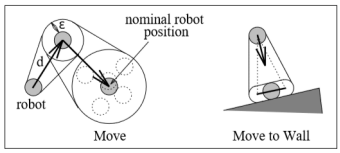
The concept of SUF has been used in [277][278] to propose a coastal navigation, depicted in Figure 3.7, inspired by traditional navigation ships when GPS is not available. Navigating close to the land enables the sailors to determine where they are with accuracy. The POMDP problem is solved by considering the uncertainty of the robot as a state variable, it is called extended state space due to the integration of the uncertainty in the state vector. Based on a grid, the aim of this algorithm is to minimize the uncertainty at the goal position. The uncertainty is modelled as a single parameter which is the entropy of a Gaussian distribution and then use Value Iteration [259] to find an optimal policy in the belief space. This Value iteration tries to find the policy that maximizes the long term reward.

3.4.2.3 Contact sensor

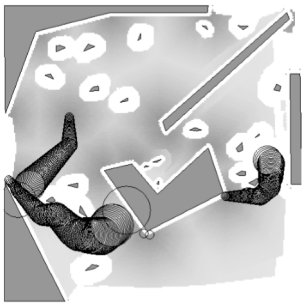
Planning with contact sensor is proposed in [5][31][161] where a potential field method is used to find a path in a polygonal environment. The state space is represented by a grid and the propagation of the uncertainty is realized with a wave propagation that gives to any nodes in the grid an uncertainty. This latter is represented by a radius of a circle (a ball) around the state. The uncertainty grows linearly with the distance when it is going far from an object where it can relocate and is reset when the robot enters a relocation area as shown in Figure 3.8(a). The ball of uncertainty is deformed when it reaches a wall, i.e. an obstacle, as depicted in Figure 3.8(b). The obstacles are used as landmark and the corner between two walls is seen as a reliable landmark. A plan consists in following a sequence of command such as "*Follow the wall*", "*Follow until it reaches an obstacle*", "*Follow to corner*" or "*Switch Wall*". An example of a plan generated is depicted in Figure 3.8 (c). The circles represent the growing uncertainty along the path when no landmark is available. Firstly the path consists in joining a wall for sure, then to follow the wall until the corner to reduce the uncertainty. Then the path proposes to reach the obstacle in the middle by avoiding another obstacles by checking if the ball of uncertainty does not collide an obstacle along the path. Once the wall is reached, it follows until the corner again, move along the wall to reach again another corner. Finally it can reach the desired configuration without collision along the path. The algorithm proposes a sort of high level strategy based on simple commands such as "*follow the wall*" or "*follow the wall until the corner*" for example. The uncertainty is reset when a corner or a landmark region is reached for sure. The solution proposed in Figure 3.8(c) aims at minimizing the path. Another solution



(a) Reset of the uncertainty when the robot enters for sure a relocation area shown by the gray square.



(b) Deformation of the ball of uncertainty when it reaches a wall.



(c) Path found with the uncertainty propagation.

Figure 3.8: Planning algorithm taking into account the uncertainty based on the wavefront potential field method. Images extracted from [31].

could to minimize the uncertainty at the goal location.

Another contact sensor method is proposed in [194][195][196][238] to find a path with only a compass, a map and a contact sensor. The robot is able to move in any direction and the uncertainty on the compass is bounded. The concept of the algorithm is inspired by the notion of preimage [203] presented earlier and the error propagation in a cone shape. This work is highly inspired by the work in [90] where the localization problem is addressed for a blind robot such as a vaccum cleaner robot which is equipped only with contact sensors. The algorithm provides a high level strategy similar to the problem of the coastal navigation [277] by navigating between landmark regions.

3.4.2.4 Sampling based planners with uncertainty

It refers to planning methods based on sampling strategies such PRM, RRT or grid to construct a graph, and that considers the uncertainty propagation between the edges, i.e. the path. Sometimes the uncertainty is added as a state variable and then it is called extended state space. Some of the algorithms presented in the previous subsection belong to these kind of methods such as the coastal navigation [277] for example.

One of the first outdoor planning methods considering uncertainty was proposed in [119][120] on a rough or difficult terrain. The path is found by a propagation algorithm on a grid that consists in a wavefront expansion of a numerical potential similarly to [31](see contact sensor approaches with uncertainty). Some visibility regions where a landmark is detectable are defined. Out of the visibility regions, the uncertainty grows proportionally to the distance and the difficulty of the terrain. Inside these regions, the uncertainty depends firstly on the same model as outside the region and secondly on a localization procedure based on the maximal distance of detection. The potential given at each node is obtained from the cost of a bitmap that computes the slope and the roughness at this point and the uncertainty is computed based on the models explained. However this method is unable to choose higher path cost (due to the difficult terrain) to achieve a lower uncertainty at the goal location.

A grid-based planning algorithm is proposed in [111][109][108] where the uncertainty increases linearly with the distance travelled and considers a 2σ safe path by assuming a bounded error on the direction (heading). It shows how to simplify a Gaussian error model to a single

parameter that increases linearly with the distance travelled at a specific rate $(2-10\%)$. Based on a cost map, the algorithm finds a path using a A* search to satisfy the uncertainty condition at the goal point. It uses some relocation areas where the robot (a car) can detect something in the map like a pole or a tree to reduce the uncertainty. The landmarks are extracted from a high resolution optical image and are considered as *point* when it deals with poles or trees and *linear* when it deals with walls for example. To avoid any ambiguities with point landmarks, i.e. wrong data association, it considers only *unique detection* regions. Considering a sensor that can detect isotropically at a range R, the *unique detection* region for a point landmark located in (x, y) is a circle centered at (x, y) with a radius R. The intersection between different detection regions has to be removed to consider only *unique detection* region. Consequently if the 2σ position of the robot (assuming containing all the posible location) is totally included in a *unique detection* region, it guarantees the detection of the point landmark and then the uncertainty can be reduced to a defined level. Moreover it proposes a replanning approach, called RPUP (Replanning with Uncertainty in Position), based on a D* algorithm to modify locally the graph if a landmark is not detected and finds a new path if possible. As presented earlier, it consists in a backward graph search (starting from the end) until the starting point is visited. When dealing with linear landmark, the full covariance matrix is needed. Entropy is used as in [277] to reduce the dimension of the search space. Indeed entropy is proportional to the product of the semi-axis of the covariance matrix [109]. However when comparing two ellipses with identical semi-axes that are rotated to each other (same entropy then), one can be in collision with an obstacle. Therefore it uses the concept developed in [43] to keep the full covariance matrix by using an incremental binning approach called the "grow" operator.

The algorithm presented here will be the subject of the next section with an application to our problem.

PRM algorithm have been used to deal with uncertainty in [261], it is called Belief RoadMap (BRM). It explicitly addresses the problem of observability by simulating measurements along candidate paths and selecting the path that has the minimal uncertainty at the goal location or with a limited uncertainty along the entire path. However it assumes the mean of the system is fully controllable at each time meaning that the controller is always able to drive the state estimate to the desired path. This method can be considered when the vehicle is moving slowly.

PRM, also called PPP, are used in [103][102] to find a safe path for a car-like robot considering uncertainties on two parameters related to the control of the robot: the travelling distance and the steering angle. An elaborated model of uncertainty propagation is then provided considering separately the uncertainty coming from the longitudinal and the steering controls. The heading is added to the state variable to consider non-holonomic path. It considers circular areas of relocation (named field of influence of the landmarks) where the uncertainty is reduced to a certain level. During the learning phase of the PPP, the nodes are spread in the relocation areas of the landmarks and the edges are checked to be robust, i.e. collision free.

PPP is used with a Kalman filter in [172] to estimate the state of the system. A perception uncertainty field is proposed to estimate the localization abilities of different landmarks and a sensor simulator is used to select the best features. However the uncertainty is not directly included in the search space.

The concept of "Towers of uncertainty" is introduced in [173] which allows multiple covariance at the same node of a graph proposed by a PPP. An A* graph search enables to find the shortest path considering Dubins path distance between the samples. The state of the robot is determined by a Kalman filter for the prediction and the update based on a sensor simulation

provided in [174]. The estimation is done at each time step. An A* graph search is also used to find the shortest path in [174] based on a grid discretization and considering observations where pertinent features can be detected, feeding a Kalman filter. The trajectory is smoothed with Bézier curve for nonholonomic robot.

The idea defended in [173][174] is to use the same localization procedure during the planning and the execution phases, here it is a Kalman filter.

A PPP taking into account for uncertainty will be proposed in the next subsection.

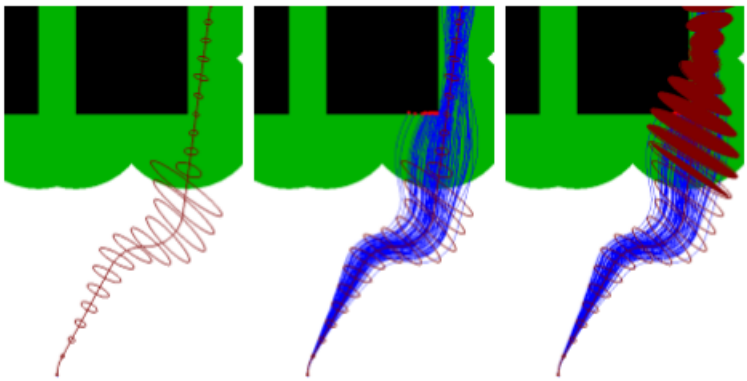
RRT algorithms have also been extended to deal with uncertainty. In [249] a RRT algorithm is used for a car-like robot with uncertainty propogation using a Kalman filter. Sensor measurements coming from telemetric sensor simulation provided in [174] is used around the mean position and feed the Kalman filter for the localization procedure. A box-RRT is provided in [245][246][248] to propose a robust collision free path in case of bounded errors with a guaranteed method of integration. The measurements are not taking into account in the update procedure.

A particle RRT is proposed in [214] where the state of the robot is represented by a discret set of states. The accuracy of the system depends on the number of particle used.

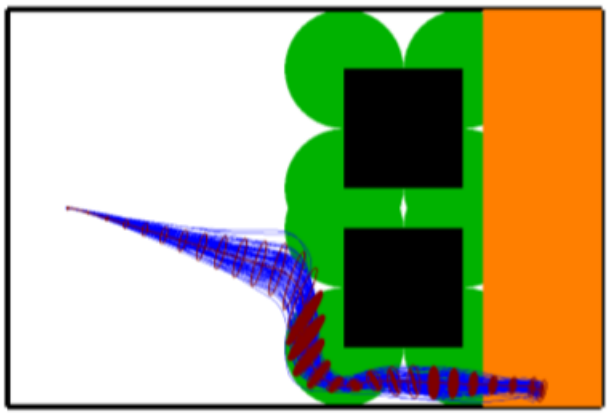
Based on many candidate paths generated with a RRT algorithm, an LQP-MP (Linear Quadratic Gaussian Motion Planning)[319] is used to evaluate all these candidate paths to select the best path by considering the sensors and the controller that will be used during the execution of the path. This motion planner is based on a linear quadratic controller with a gaussian model of uncertainty and estimates the *a priori* probability distributions of the state along the path. In other words, it evaluates the probability of success of a given trajectory instead of building an optimal one. This framework has been extended to solve a POMDP problem in [320][321] but still requires an initialization with a RRT algorithm.

An optimal solution is proposed in the algorithm called RRBT (Rapidly-exploring Random Belief Tree)[34][35] that shows interesting results in planning with nonholonomic constraint. It combines a variant of LQG-MP with RRT* to provide a globally optimal solution considering the controller. It builds incrementally a roadmap for a set of beliefs minimizing the uncertainty at the goal while respecting the chance-constraints [28] (defined by a threshold) along the path. The path provided by the algorithm is called a nominal path. The controller, when position measurements are available, has to relocate the position of the robot onto this path considering the nonholonomic constraints if there are some. Figure 3.9 shows the path found with this algorithm and proves that the set of closed-loop trajectories around the nominal path avoids the obstacle and uses the beacons measurements to relocate onto the nominal path found in red.

The approach has been improved in [257][258] where the samples are not anymore generated uniformly but in areas where a localization is possible (around a beacon) reducing the number of nodes and egdes.



(a) On the left, a path found where the predicted covariance ellipses using a Kalman filter are highly reduced when measurements are received and the robot seems to pass safely the obstacle. In the middle, a set of closed-loop trajectories (blue) found with the previous nominal path. When these trajectories enter the green area, the robot can relocate itself and join the nominal path. However due to the time for the controller to pull the robot back to the nominal path it results in collision. On the right, the solid ellipses show the closed-loop distribution with correct prediction of the collision.



(b) The path deviates to move parallely to the obstacle (black square) in order to get the measurements for the localization and to stabilize onto the nominal path to reach the goal area in orange.

Figure 3.9: RRBT algorithm with Dubins dynamics and beacons measurements. The green regions indicate the robot position where a range and a bearing measurement from the beacon (corner) is available. The nominal path is depicted in red and the red ellipses show the predicted covariance. The blue paths are closed-loop trajectories. Images are extracted from [34]

A review of some of sampling based motion planning techniques can be found in [85] considering or not uncertainty.

3.4.2.5 Conclusion

Many planning algorithms considering uncertainty are available today but few are applied to the underwater environment due to the presence of currents and the possibility to loose the robot. This environment is then very challenging for a low cost robot to find a path at a long distance that guarantees the reachability of an objective. Indeed, in the context of this thesis, the solution proposed, in Chapter 6, is to guarantee the revisit or the renavigation of a previously

mapped object (mine). Due to the unstructured aspect of the underwater environment, many planning algorithms show their limits. Moreover handling the uncertainties from motion and sensing (and possibly the map) is a difficult task for a motion planner that has to provide the guarantee of a successful mission.

3.5 Planning methods applied to our context

3.5.1 Introduction

In this section, two planning algorithms considering uncertainty, introduced in the previous section, are applied to the context of this thesis to understand the difficulty of the motion planning problem to provide the guarantee of the revisit of an object when localization procedures are involved. It will be assumed a robot equipped with an exteroceptive sensor (a sonar) that can sense the environment and some algorithms presented in Chapter 2 that can detect automatically some landmarks. The planners will deal only with *punctual* landmarks for simplification reasons and no obstacle will be present. The first algorithm, called *Gonzalez planner*, is an implementation of the algorithm proposed in [109] where the motion planner searches a path in an extended state space represented by a grid and an A* graph search. The second algorithm, called *Lambert-Gruyer planner*, will be an adaptation of the algorithm proposed in [173] where the Probabilistic Path Planner (PPP) is used with an A* graph search considering uncertainty. In both algorithms, some punctual landmarks are present and then some visibility regions based on the exteroceptive ability of the sensor embedded.

3.5.2 Gonzalez planner

This planner is inspired by the work in [109]. It presents a motion planner on a grid considering uncertainty as a state variable and tries to find a path that satisfies an uncertainty constraint at the goal location. The algorithm is originally based on a cost-map as it considers an outdoor environment for a driving robot. The values in the cost-map depend on the nature of the terrain, if it is easy or not to drive on it.

The robot is described by simple Dubins vehicle dynamics:

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = w \end{cases} \tag{3.5}$$

where the state of the robot is represented by $\mathbf{q} = (x, y, \theta)$ (the x-position, the y-position and the heading respectively), and the inputs to the model by $\mathbf{u} = (v, w)$ (the linear or longitudinal speed and the angular speed or rate of change in heading). Equation 6.3 can be expressed generally as:

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases} \tag{3.6}$$

where $\mathbf{x}_0$ is the initial configuration of the robot.

The drift is usually modelled as errors on the inputs [274] where w_v is the error in v and w_w is the error in w. This gives then:

$$\begin{cases} \dot{x} = (v + w_v)\cos\theta \\ \dot{y} = (v + w_v)\sin\theta \\ \dot{\theta} = (w + w_w) \end{cases} \tag{3.7}$$

which can be given in discrete time:

$$\begin{cases} x(k+1) = x(k) + (v(k) + w_v(k))\cos(\theta(k))\Delta t \\ y(k+1) = y(k) + (v(k) + w_v(k))\sin(\theta(k))\Delta t \\ \theta(k+1) = \theta(k) + (w(k) + w_w(k))\Delta t \end{cases} \tag{3.8}$$

with Δt a small time step.

Assuming zero-mean Gaussian errors, the Extended Kalman Filter (EKF) gives for the discret system in Equation 3.8 the following covariance prediction:

$$\Sigma(k+1) = \mathbf{F}(k)\Sigma(k)\mathbf{F}(k)^T + \mathbf{L}(k)\mathbf{Q}(k)\mathbf{L}(k)^T \tag{3.9}$$

where:

$$\mathbf{Q}(k) = \frac{1}{\Delta t}\begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_w^2 \end{pmatrix} \tag{3.10}$$

with σ_v and σ_w the errors in the longitudinal speed and angular rate respectively.

Moreover $\Sigma(k) = E(\hat{\mathbf{x}}(k)\hat{\mathbf{x}}(k)^T)$.

F and L correspond to Jacobians where $F_{ij} = \frac{\partial f(\mathbf{x}_i(k), u_j(k))}{\partial \mathbf{x}_i}$ and $L_{ij} = \frac{\partial f(\mathbf{x}_i(k), u_j(k))}{\partial \mathbf{u}_j}$ that give:

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & -v\Delta t \cdot sin(\theta(k)) \\ 0 & 1 & v\Delta t \cdot cos(\theta(k)) \\ 0 & 0 & 1 \end{pmatrix} \tag{3.11}$$

and

$$\mathbf{L} = \begin{pmatrix} \Delta t \cdot cos(\theta(k)) & 0 \\ \Delta t \cdot sin(\theta(k)) & 0 \\ 0 & \Delta t \end{pmatrix} \tag{3.12}$$

According to [159], a straight line trajectory (along the x-axis for example) maximizes the error terms for these equations. As suggested in [109], this result can be used as an upper bound on the error for any trajectory and it shows that the dominant error term is the error in the initial heading which varies linearly with the distance travelled. It considers a 2σ contour as a single isometric Gaussian parameter, which remains a good approximation of the 2σ contours of the full EKF model. Planning in extended state space (known as augmented state space too) considers uncertainty as additional dimensions, the configuration-uncertainty space is then:

$$\mathbf{q} = (\mathbf{x}, \epsilon) = (\mathbf{p}, \theta, \epsilon) \tag{3.13}$$

where $\mathbf{p} = (x, y)$ is the position of the robot and θ is the heading of the robot. The parameter $\epsilon = (\epsilon_1, \epsilon_2, ..., \epsilon_n)$ defines the uncertainties of the robot.

Considering a Gaussian uncertainty model $\mathbf{x}_k \sim \mathcal{N}(\mu_k, \Sigma_k)$, the most likely configuration is given at time step k by $\mu_k = (\mu_{x_k}, \mu_{y_k}, \mu_{\theta_k})$ and the full covariance matrix is given by:

$$\Sigma_k = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta\theta} \end{pmatrix}_k \tag{3.14}$$

The full covariance matrix contains 6 independent parameters. Consequently a full extended state space $\mathbf{q}$ would have a dimension equal to 9, meaning planning in 9 dimensions. Under

the assumption that changes in direction do not affect the heading error, it is not necessary to include θ in the planning space. Due to the high dimension of the extended state space, a simplest 2D isometric Gaussian is preferred where $\mathbf{x}_k = (x_k, y_k)$ and $\mathbf{x}_k \sim \mathcal{N}(\mu_k, \Sigma_k)$. In this case, the standard deviation is given by $\sigma_k = \sigma_{x_k} = \sigma_{y_k}$. The position of the robot is then given by the following probability distribution:

$$p_{\mu_k,\sigma_k}(\mathbf{x}_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} exp(-\frac{1}{2}\frac{(\mathbf{x}_k - \mu_k)(\mathbf{x}_k - \mu_k)^T}{\sigma_k^2}) \tag{3.15}$$

By defining:

$$\epsilon_k = 2 \cdot \sigma k \tag{3.16}$$

the boundary of the uncertainty centered at μ_k with a radius ϵ_k, it corresponds to the 2σ contour of the Gaussian distribution. As presented earlier, this assumption is valid if the dominant term in the error propagation is the error in the initial heading. This model is a conservative estimate of the true error.

The extended state space is then:

$$\mathbf{q} = (x, y, \epsilon) \tag{3.17}$$

where the error term ϵ increases linearly with the distance travelled at a specific rate. The following equation defines the propagation of the uncertainty:

$$\epsilon_{k+1} = \epsilon_k + \alpha d(\mu_k, \mu_{k+1}) \tag{3.18}$$

where α is a user-defined parameter and $d(\mu_k, \mu_{k+1})$ is the distance between the two configurations. Generally α belongs to the interval $[1\%, 10\%]$.

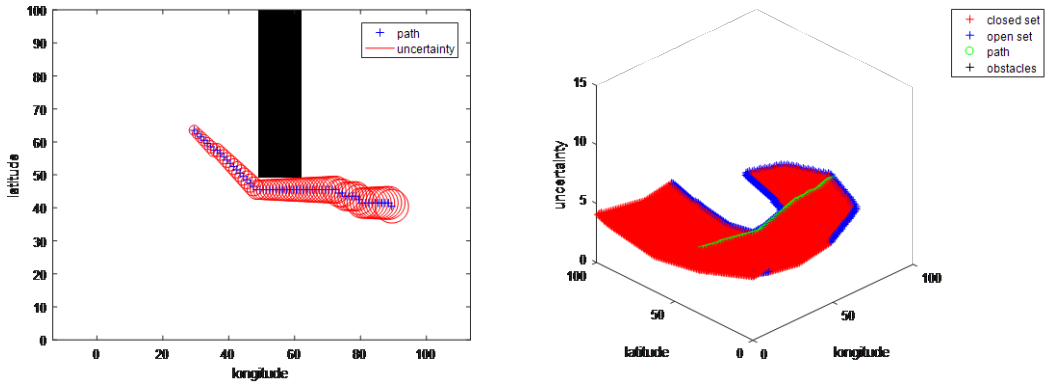
The planning method is then based on a 3D grid with deterministic transitions where a modified A* graph search is used to find the path. To reduce the computational burden in the 3D search space, the successors of each state (node in the grid) are computed only in the 2D plane. In other words, for each state $\mathbf{x}_k = (x_k, y_k)$ (2D point in the grid), the successors are the 8-connected samples in the plane. The motion depends on the resolution of the grid in the $(x - y)$ plane. Usually a regular grid is used for the $(x - y)$ plane. The new uncertainty is computed for each neighbour according to Equation 3.18. It provides an augmented state variable for each neighbour $\mathbf{q}_{k+1} = (x_{k+1}, y_{k+1}, \epsilon_{k+1})$. To determine if a successor $\mathbf{q}_{k+1}$ should be placed in the OPEN list of the A* algorithm, a pruning strategy based on state dominance is used to reduce the complexity of the path finding. When a successor is added to the OPEN list, the state is expanded. A successor respecting one of these following conditions is added:

- no state with (x, y) coordinates has been expanded.
- the expected cost from $\mathbf{q}_0$ to $\mathbf{q}_{k+1}$ is lower than the cost of any state with the same (x, y) coordinates.
- the computed uncertainty ϵ_{k+1} is lower than the uncertainty of any other state with the same (x, y) coordinates.

The A* algorithm runs until the goal position $\mathbf{x}_k = \mathbf{x}_f$ is visited with a lower uncertainty ϵ_f defined by the user. The path is finally reconstructed by connecting the successive predecessor from $\mathbf{q}_f$ to $\mathbf{q}_0$.

Cost map version of Gonzalez

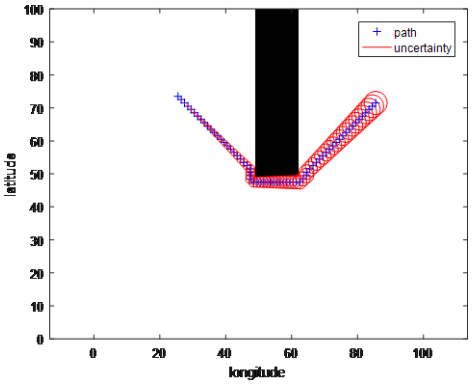
In the version of Gonzalez [109], a 2D cost map is assumed for a driving robot. The cost from one configuration $\mathbf{q}_k$ to another $\mathbf{q}_{k+1}$ is based on the sum of the cost values in the map weighted according to the Gaussian distribution for all states inside the 2σ (ϵ) uncertainty. In the proposed A* algorithm, an Euclidean heuristic is used to reach more quickly the goal location (f function). An example of the planner proposed by Gonzalez is depicted in Figure 3.10(a). An high cost region (may be an obstacle), depicted in black, has a cost defined at 100 and the free space is 5. In Figure 3.10(b), the path in the extended state space is shown in green where the visited nodes (CLOSED set) are represented by red crosses and the OPEN set by blue crosses. The final uncertainty, depending obviously on the discretization along the uncertainty vector, is $5.85m$.



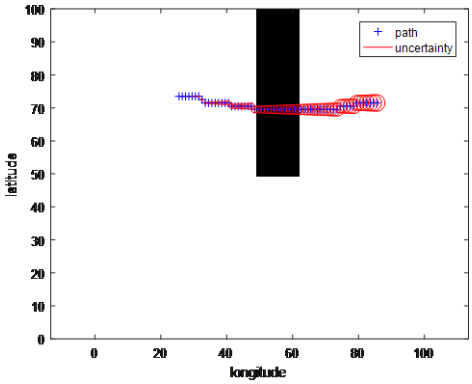
(a) Path found with uncertainty propagation. (b) OPEN and CLOSED sets in the extended state space.

Figure 3.10: Path search in extended state space where $\epsilon_0 = 1.5m$ and $\epsilon_f = 6m$. The high cost region in black has a cost at 100 and in free space it is 5. The parameter α is fixed at 10% of the distance travelled.

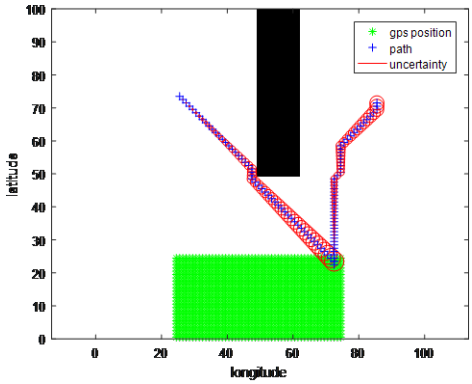
If the uncertainty at the end is not respected by getting around the high cost region, the path goes through it as depicted in Figure 3.11. Indeed, assuming an increasing uncertainty of $\alpha = 4\%$ of the distance travelled, the uncertainty at the goal location is $3.48m$ with a cost of the path at 151 by getting around the high cost region represented in Figure 3.11(a). If $\epsilon_f < 3.48m$, for example $\epsilon_f = 3m$, the robot goes through the high cost area resulting in a path cost of 552 and a final uncertainty at $2.52m$ respecting the final uncertainty condition as shown in Figure 3.11(b).



(a) Path getting around the high cost region where the final uncertainty asked was $\epsilon_f < 5m$. The final uncertainty is actually $3.48m$ and the cost path 151.



(b) Path going through the high cost region in black to respect the final uncertainty condition fixed at $\epsilon_f < 3m$. The path cost is 552 and the final uncertainty is $2.52m$.



(c) Path going through the relocation area (GPS for example) shown in green. The final uncertainty is $2.14m$ respecting the final uncertainty condition $\epsilon_f < 3m$.

Figure 3.11: Path finding in extended state space where $\epsilon_0 = 0m$. The high cost region in black has a cost at 100 and in free space it is 5. The drift parameter α is fixed at 4%.

When a state $\mathbf{x}_k$ with the associated 2σ uncertainty (ϵ_k) is totally included in a landmark region, the uncertainty ϵ_k is reduced to a fixed value $\epsilon_{landmark}$. It enables to avoid the high cost region by going to these relocation areas and then having a lower cost path as depicted in Figure 3.11(c) where ϵ_f is defined at $3m$ resulting in a final uncertainty at the goal location of $2.14m$ and a path cost of 230.

The concept of a cost map may be interesting in the context of underwater environment. Indeed an high cost region could an area where the density of landmarks is close to 0, it could area where some obstacles are present such as wrecks or it could an area where the current is too high.

Proposed version with relocation areas and sensing

In the version proposed as example in this thesis, the cost function, known as the g function in the A* algorithm, is simply the length of the path, meaning that the transition cost between two neighbours is equal to the Euclidean distance between these two configurations in the 2D

plane.

$$cost(\mathbf{x}_k, \mathbf{x}_{k+1}) = \begin{cases} \text{resolution grid} & \text{the neighbour is in the vertical or the horizontal line} \\ \text{resolution grid} \cdot \sqrt{2} & \text{the neighbour is in the diagonal line} \end{cases} \tag{3.19}$$

It will be assumed an Euclidean heuristic too.

On the same principle, it will be assumed that if an uncertain state q_k, with mean state $\mathbf{x}_k$ and a 2σ circle of uncertainty (ϵ_k), is totally included in a relocation area, the uncertainty will be reduced to a fixed value. That means all the possible states $\mathbf{x}_k$ in the uncertain position are able to detect the landmark considered.

Assuming a set of punctual landmarks $\mathbb{M} = \{\mathbf{m}_1, \mathbf{m}_2, ...\mathbf{m}_n\}$ where $\mathbf{m}_i$ is the location of a landmark. Contrary to the *unique detection region* proposed in the initial version [109] where only one landmark could belong to this region to avoid any ambiguities and the problem of data association, it will assumed here that some regions may intersect and that some process are able to make the associations. When considering a sensor with a 360° field of view and a range R, the detection region are circles centered at the landmark location with a radius equal to the range R.

Firstly, in our problem, it will assumed that the sensor can detect between a minimal and a maximal range (R_{min} and R_{max}) as the AUV is navigating above the seabed at the certain altitude, and the sensor swath is tilted.

An uncertain extended state $\mathbf{q}_k = (x_k, y_k, \epsilon_k)$ is defined as:

$$Circle(\mathbf{q}_k) = \{p \in \mathbb{R}^2 | \sqrt{(p(1) - x_k)^2 + (p(2) - y_k)^2} \leq \epsilon_k\} \tag{3.20}$$

that represents the set of the robot configuration inside the 2σ uncertainty circle.

A landmark i is then simply detected at the extended state $\mathbf{q}_k$ if:

$$\forall \mathbf{p} \in Circle(\mathbf{q}_k) \quad \sqrt{(m_i(1) - p(1))^2 + (m_i(2) - p(2))^2} \in [R_{min}, R_{max}] \tag{3.21}$$

Secondly, it will be assumed a sensor, such as a Forward Looking Sonar (FLS), that can only sense in front of the robot in a defined angular aperture $[-\beta, \beta]$ and in a interval of ranges $[R_{min}, R_{max}]$. Under the assumption that the robot does not stop its motion to turn on itself at the extended state $\mathbf{q}_k$ (it depends on nonholonomic constraints), a part of the path has to be taken into account. For a motion from the extended state $\mathbf{q}_k$ to a neighbouring extended state $\mathbf{q}_{k+1}$ where the new uncertainty is computed similarly as previously (Equation 3.18), a landmark i will detected at the extended state $\mathbf{q}_{k+1}$ if:

$$\forall \mathbf{p} \in Circle(\mathbf{q}_{k+1}) \begin{cases} \sqrt{(m_i(1) - p(1))^2 + (m_i(2) - p(2))^2} \in [R_{min}, R_{max}] \\ atan2(m_i(2) - p(2), m_i(1) - p(1)) - atan2(y_{k+1} - y_k, x_{k+1} - x_k) \in [-\beta, \beta] \end{cases} \tag{3.22}$$

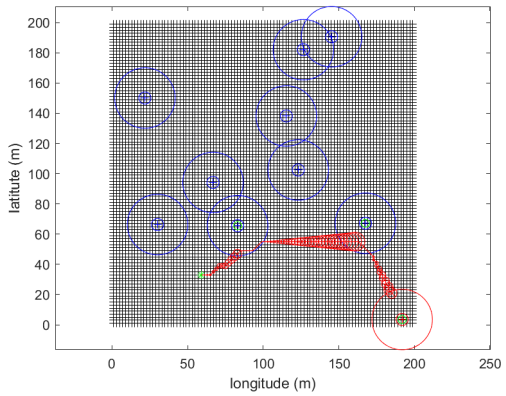
The first test checks for the range detection and the second test for the angular detection. In the second test, $atan2(y_{k+1} - y_k, x_{k+1} - x_k)$ corresponds to the heading of the robot along the transition path between the two configurations.

Contrary to the classical path planning algorithm and the planner proposed by Gonzalez to reach a unique final state (x, y) with possibly an uncertainty condition ϵ_f, a set of final states is possible for the relocation of the landmark. Indeed, instead of reaching a final state, if an

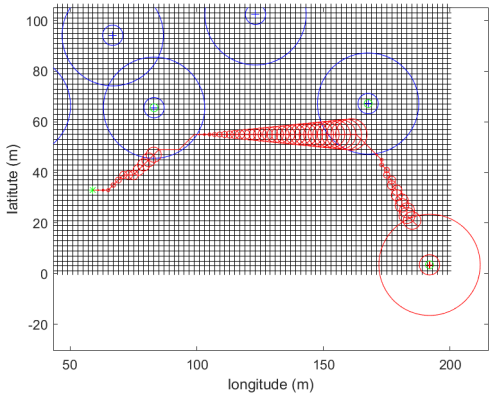
extended state checks the detection tests presented above for the target (assumed punctual), the algorithm should stop and the path is reconstructed by taking the successive predecessors of this "winning" extended state. The target behaves as a landmark.

Example 1

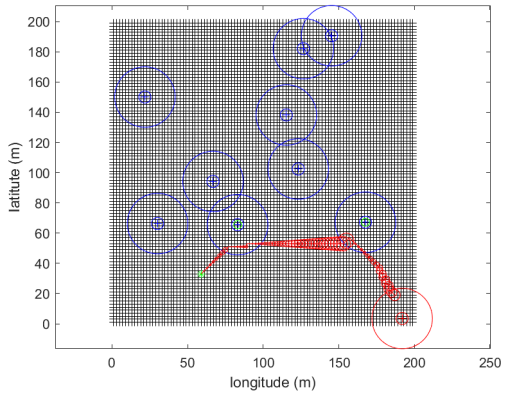
Ten landmarks are generated randomly. One is selected as the target. The robot can be equipped by a 360° field of view sensor with a range of detection between 4 and 20 meters or by a FLS with a range of detection between 4 and 20 meters and an aperture angle of 60° (whole aperture angle). The parameter α is fixed at 10% for the propagation of the uncertainty. The regular grid has a resolution of 2 meters in the (x, y) plane and a resolution of $0.02m$ along the uncertainty axis. This latter resolution should be choiced according the parameter α and the resolution of the grid. The initial state $\mathbf{x}_0 = (x_0, y_0)$ is generated randomly with an null uncertainty at the beginning. The uncertainty is reduced at 0 when a landmark is detected. Figure 3.12 depicts the comparison of the path found between the two sensors. Figure 3.12(a) shows the path found with the isotropic sensor. It can be seen in the zoom in Figure 3.12(b) where the robot stays on the boarder of the relocation area when it detects the first landmark. Then it goes straightforward to the second landmark to finally reach the area of the target to detect it. The path is slightly different with the FLS as depicted in Figure 3.12(c). Indeed, in the zoom in Figure 3.12(d) when the robot detects the first landmark, it goes to the second landmark. As the landmark is not anymore in the field of view, the uncertainty grows even if the state $\mathbf{x} = (x, y)$ belongs to the 2D relocation area based on ranges (blue circles). Before leaving the area of influence of the first landmark, the robot changes its direction to redetect the first landmark and relocate. Then the robot goes to the second landmark area and finally to the target area to guarantee the detection.



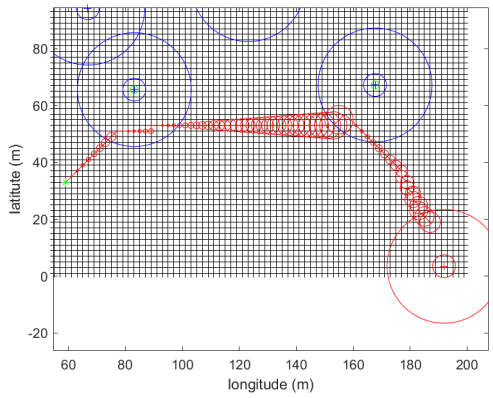
(a) Path generated with the 360° sensor and the limited range.



(b) Zoom of the path (a).



(c) Path generated with the FLS.



(d) Zoom of the path (c).

Figure 3.12: Comparison between the paths found with the two sensors. The black crosses show the grid (the states). The blue points are the landmarks with the associated range detection drawn by blue circles. It is drawn in red similarly for the target. The green asterisk is the initial configuration. The path is in red with the uncertainty circles at 2σ in red too.

Example 2

The second example comes from real data. A survey mission mission in a *boustrophédon* pattern has been conducted by an A9 from ECA Group in a training environment. Some landmarks (cubes) are present in the environment and are detected with the integral image detector presented in Chapter 2. The environment is mainly composed of sand with small rocks and the cubes. A false manta mine is present for detection purpose. These landmarks are considered as *punctual* due to their reduced size ($\approx 1m \times 1m$). Figure 3.13 depicts the different position of the landmark. The position of the false manta mine is represented by the red cross. The smallest landmarks are removed and only the landmarks with the strongest backscattered signal are kept to be detectable by the revisit system. This latter is supposed to be equipped with a compass that has a low uncertainty and a Forward Looking Sonar (FLS) with a range detection between 4 and $30m$ enabling a good image at an altitude of $3m - 4m$ above the seabed. The aperture angle is 60°. The values for the FLS are defined according to some products.

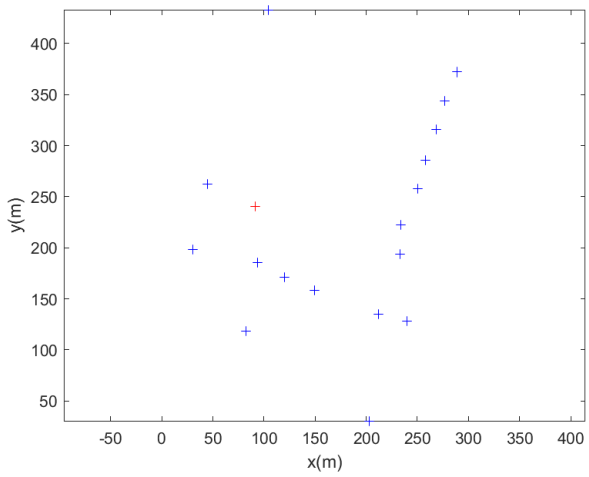


Figure 3.13: Real environment with punctual landmarks in blue and a false manta mine in red.

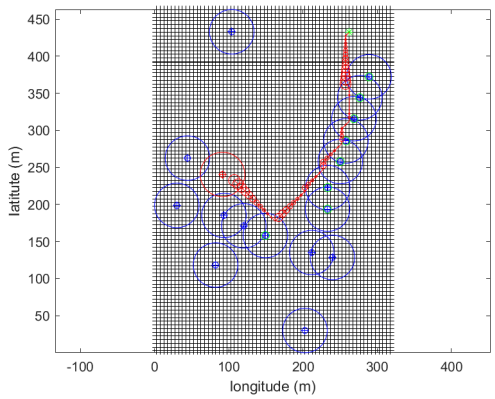
The initial position of the robot is assumed to be known but is generated on purpose far from the target. The resolution of the grid is $5m$ allowing a fast computation of the algorithm. The parameter α is fixed at 10%. The path found with the algorithm is proposed in Figure 3.14 with some zooms. The path consists in following the line of landmarks and then switching to the target. On the last zoom in Figure 3.14(d), a sudden changes of the direction is present to reduce the uncertainty. The path needs to be smoothed and can be computed with a higher resolution to avoid this sudden changes.

Conclusion

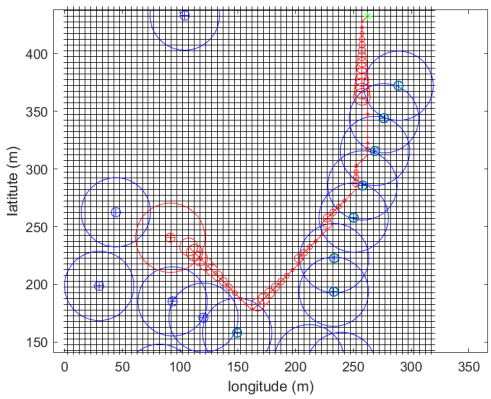
This first algorithm is quite simple but shows a first implementation of the planning method under uncertainty adapted to the problem of the revisit with an exteroceptive sensor. Based on a simple discretization of robot configuration in a regular grid, an extended (augmented) state space is created to include the uncertainty as a state variable with a single parameter that increases linearly with the distance travelled at a certain rate. In many underwater applications, the uncertainty on the robot position is expressed per distance travelled. As it can be seen in the different examples, the discretization in a regular grid implies many robot configurations outside regions of influence of the landmarks. Planning methods based on strategic sampling may appear as an interesting alternative solution that will be proposed in the next subsection. Moreover planning on grids is not really adapted to propose plans for vehicle under nonholonomic constraint.

3.5.3 Lambert-Gruyer planner

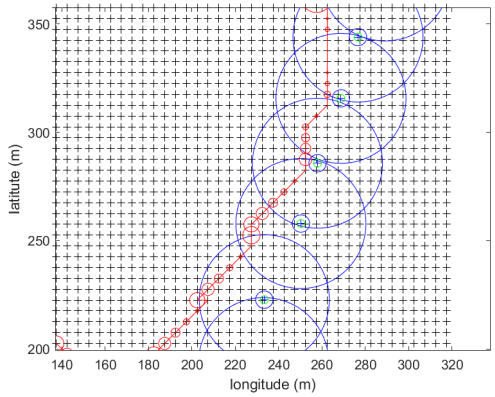
This second planner is inspired by the work proposed in [173] where a Probabilistic Path Planner (PPP) is used to create the graph. This PPP is based on the work in [102] to propose a planning method for car-like robots with uncertainty in motion. Once the graph is generated (vertices = samples and edges = connected paths), an A* algorithm is used to find the shortest path based on the Dubins [83] distance between the samples. It introduces the notion of tower of uncertainties to keep at the same node different path costs and uncertainties. It consists then to plan in an uncertainty-configuration space as the extended configuration space on the grid presented with the Gonzalez planner. It is necessary to plan in this uncertainty-configuration space since the A* search tries to find the smallest path. Indeed in the original version of the A* algorithm a previously explored node (in the CLOSED set) cannot be explored again because the cost will be bigger than the previous path. If the algorithm has found the smallest



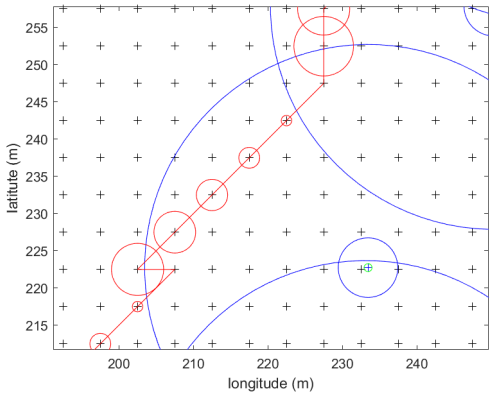
(a) Path generated with the FLS.



(b) Zoom of the path.



(c) Zoom on a part of the path.



(d) Zoom on the change in direction for relocation.

Figure 3.14: Path found for the revisit with the FLS. The black crosses show the grid (the states). The blue points are the landmarks with the associated range detection drawn by blue circles. It is drawn in red similarly for the target. The green asterisk is the initial configuration. The path is in red with the uncertainty circles at 2σ in red too.

path to go to a specific configuration (node), it keeps only the smallest path and does not look for a longer path with potentially a lowest uncertainty. Each tower has then a state (x, y, θ) and levels are added to this tower that specify the uncertainty. When a node has not been explored, a tower is created with a first level. A new level is created at this tower according to a pruning strategy as in the Gonzalez planner. If the path computed is greater than all the previous ones at a specific tower, a new level is created only if the uncertainty is better than all the previous uncertainties. The uncertainties are ellipsoids generated based on the Kalman filter. If the path has a lower cost than all the previous ones, a new level is added by checking the upper levels of the tower corresponding to higher cost of the g function in the A* algorithm. The algorithm stops when a tower of uncertainty is expanded in the goal area.

The planner is proposed in a bounded error context. Assuming a robot described by the same dynamics as in Equation 6.3, the Dubins path will be considered to connect the samples. The Dubins path will provide a minimum length path between two configurations. Only the paths RSR, LSL, RSL and LSR will be considered. "L" corresponds to a left turn, "S" to a straight path and "R" to a right turn. The Dubins paths propose then a sequence of three commands to join the next samples. These three commands are defined by three durations. As a compass will be assumed embedded in the vehicle, the heading of the robot θ will be known

with an uncertainty. That means the heading belongs to an interval around the value given by the compass: $\theta \in [-\epsilon_{comp}, \epsilon_{comp}]$ ($\epsilon_{comp} = 3°$).

The other uncertainty will be on the linear speed v. It will be supposed that the angular rate w is perfect. Assuming a constant but uncertain linear speed, the minimum turning radius is given by:

$$[r] = \frac{[v]}{w} \tag{3.23}$$

where v is an interval around the desired speed. This speed interval is obtained as follows:

$$[v] = v_{desired} \cdot \alpha \cdot [-1, 1] \tag{3.24}$$

where α is a user specified value and $v_{desired}$ is the motion speed of the vehicle assumed constant along the path. This speed is fixed at $1m/s$ which is equivalent to 2 knots approximately. Generally when conducting a survey mission, the AUVs are navigating at a speed of $3-4$ knots. The parameter α is choiced at 10% meaning that the speed is uncertain at 10%. The angular rate is then defined to respect the turning radius of the vehicle navigating at the specified speed.

The Dubins paths provide a set of three commands $\mathbb{U} = \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ with associated durations $\mathbb{T} = \{t_1, t_2, t_3\}$ that are computed based on the desired speed and the angular rate. The command $\mathbf{u}_i = ([v], w)$ is composed of an uncertain linear speed and a perfect angular rate.

The initial pose of the robot $\mathbf{x}_0$ belongs to a set $\mathbb{X}_0$ represented by a box which can be reduced to a singleton when the position is perfectly known. A simple integration model as in [245] is used along a path where intervals are manipulated to generate the new state $\mathbf{x}_{k+1} = (x, y, \theta)$ coming from the state $\mathbf{x}_k = (x, y, \theta)$ after a Dubins command. Guaranteed integration with tubes may be used [275][274]. The next Chapter will introduce the notion of intervals. Due to the presence of a compass, the uncertainty on the heading does not have to be propagated, only the growing uncertainty on the (x, y) position of the AUV is important.

It will be assumed again a Forward Looking Sonar (FLS) as exteroceptive sensor and an environment composed of a set of punctual landmarks. Compared to the previous planner, the notion of point of view introduced in Chapter 2 is proposed. Indeed this notion is relevant in underwater environment due to the different backscattered signal according to the point of view. To provide a guaranteed plan, it is meaningful to revisit the different landmark and the target at the approximate same points of view. If the angle of detection of a landmark during the survey mission was $\theta_{detection}$ and assuming an angular flexibility γ, the landmark is then detectable with a point of view as follows:

$$[\theta_{revisit}] = \theta_{detection} + [-\gamma, \gamma] \tag{3.25}$$

That means the landmark can be revisited with a point of view of detection lying in the interval $[\theta_{revisit}]$. This can be expressed in robot position:

$$\mathbb{X}_{revisit} = \{(x, y) \in \mathbb{R}^2 | atan2(y - m_i(2), x - m_i(1)) \in [-\gamma, \gamma]\} \tag{3.26}$$

where m_i is the location of the landmark considered. These notions will be more detailed in the Chapter dealing with registration maps.

Now the samples $\mathbf{x} = (x, y, \theta)$ can be generated in the field of influence of a landmark as in [103] known as localization aware sampling in [257][258] or information sampling in [127]. Figure 3.15 depicts the environment where the landmarks are marked by blue dots, the initial

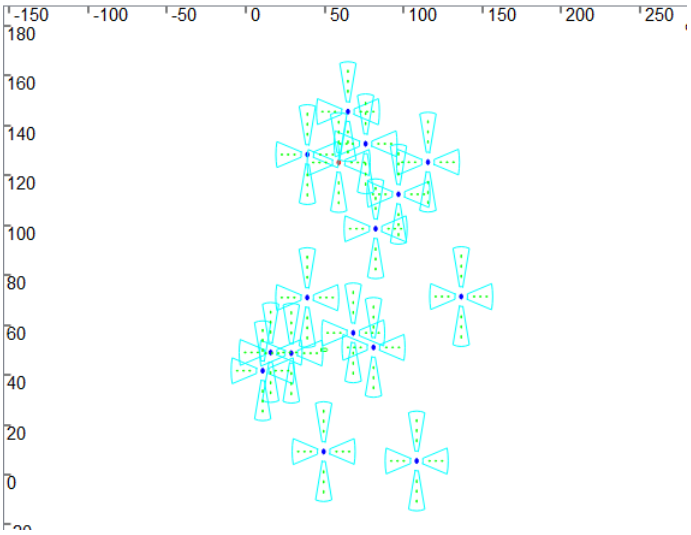


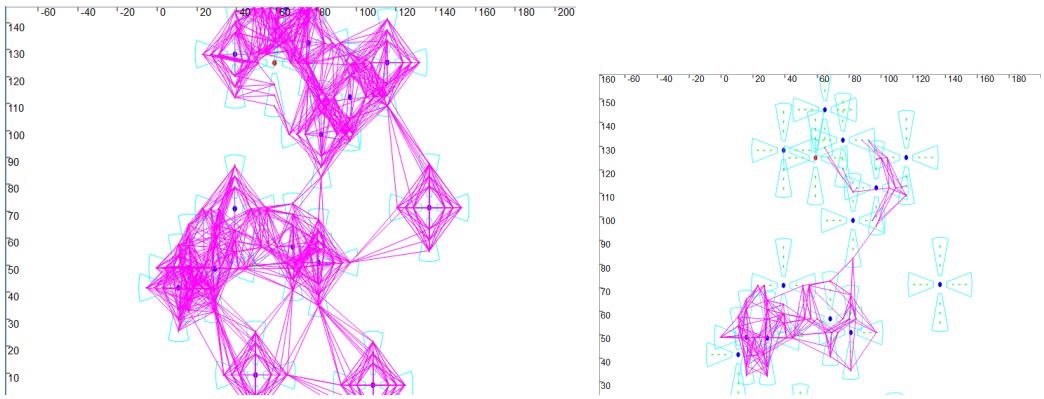
Figure 3.15: Environment with punctual landmarks.

pose is represented by a big green AUV and the samples by little green AUVs. The initial location (x, y) of the AUV is perfectly known but not the heading θ that belongs to an interval. The target is depicted by a red dot. Notice that the points of view have been represented by pies where the constraint of the limited range of detection has been added.

The samples are firstly connected based on the Dubins paths connecting them and the K-nearest (here $K = 10$) neighbours strategy. This graph is depicted in Figure 3.16(a). It gives only the potential neighbours for each samples but this path are not yet guaranteed to provide a localization procedure. The modified A* algorithm tries to find a path in the precomputed graph by extending the state with the uncertainty. Contrary to the concept of tower of uncertainty, that can be implemented too, the algorithm proposed here is only looking for a feasible path and not for the optimal one. Even if it uses a cost function based on the Dubins distance between the samples, it only looks for possibilities, meaning that a visited node cannot be revisited. During the expansion, an edge is valid only if the uncertainty propagated from the state $\mathbf{x}_k$ given by a box $\mathbb{X}_k$ (uncertain state) to the state $\mathbf{x}_{k+1}$ (given by $\mathbb{X}_{k+1}$) enables the detection of the landmark considered at the state $\mathbf{x}_{k+1} = (x, y, \theta)$ despite the uncertainty accumulated. The following tests are checked to guarantee the detection of the landmark i located at $(m_i(1), m_i(2))$ at this expanded state:

$$\forall \mathbf{p} \in \mathbb{X}_{k+1} \begin{cases} (m_i(1) - p(1))^2 + (m_i(2) - p(2))^2 \in [R_{min}, R_{max}] \\ atan2(m_i(2) - p(2), m_i(1) - p(1)) - p(3) \in [-\beta, \beta] \\ atan2(p(2) - m_i(2), p(1) - m_i(1)) \in [-\gamma, \gamma] \end{cases} \tag{3.27}$$

Set-membership tools can check this test quickly as it will be presented in the next Chapter. If all these tests are successful, an edge is then created as shown in Figure 3.16(b) where the tree expansion is depicted. The uncertainty at this state (node) is then reset to a fixed value as in [186] or in the Gonzalez planner. In this problem it will be null. Once a sample lies in the relocation area of the target which enables its detection despite the uncertainty, the A* expansion stops and the path is reconstructed as shown in Figure 3.17. The uncertainty along the path is represented by yellow boxes. The algorithm provides a path that guarantees the detection of the different landmark along the path but suffers from the relocation around the mean state which corresponds to samples generated during the first step of the Probabilistic Path Planner.



(a) Potential edges of the graph before the modified A* search.

(b) Modified A* search that guarantees the detection of the landmarks.

Figure 3.16: Edges of the graph and modified A* graph search by propagating the uncertainty.

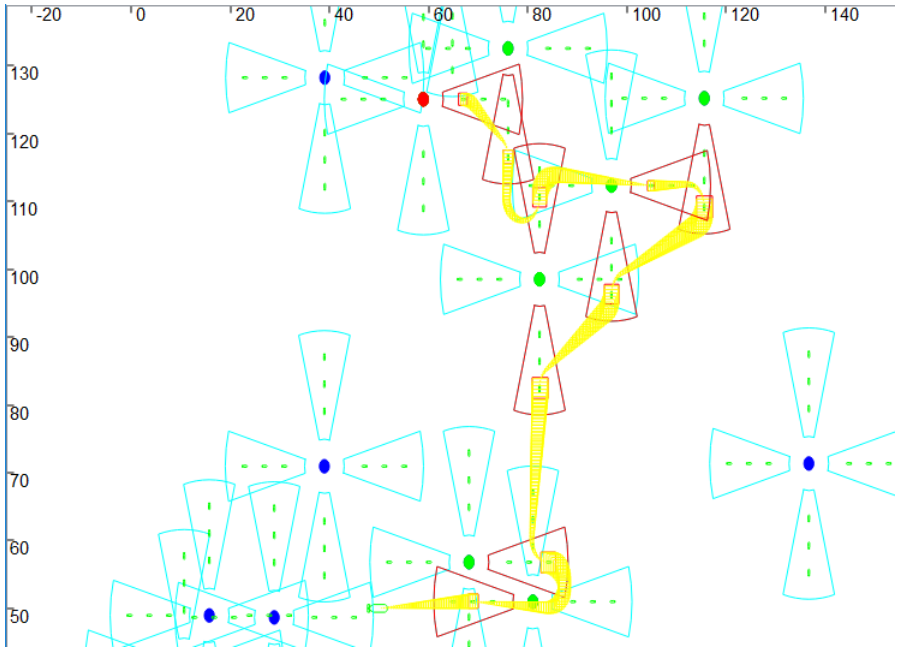


Figure 3.17: Path found with uncertainty propagation along the path.

Considering the uncertain state $\mathbf{x}_k$ belonging to the box $\mathbb{X}_k$, an engineering solution would be to expand the 8 states corresponding to the 8 corners of the box, as the state is represented by 3 variables (x, y, θ), to the new samples $\mathbf{x}_{k+1}$. This process is known as differential command [245]. It consists in giving a different command to the robot when it is at the state $\mathbf{x}_k \in \mathbb{X}_k$ depending on the position of the robot inside the box $\mathbb{X}_k$. When a state $\mathbf{x}_{k-1}$ is expanded to the state $\mathbf{x}_k$, it guarantees the detection of the landmark despite the uncertainty along the path, but the robot may be anywhere in the box $\mathbb{X}_k$. Due to the localization process, the state of the robot is finally known. Consequently, a different motion command can be applied to join the next state (sample) in the graph. The idea proposed in [245] would be to divide the state in many boxes and to prove that all lies in a reduced final box after applying a particular motion command to all of them.

The engineering solution would be to expand the 8 corners of the state $\mathbb{X}_k$ to reach the samples $\mathbf{x}_{k+1}$ and then take the union of the resulting boxes to give the global enclosure of the uncertainty $\mathbb{X}_k$. The result of such concept is presented in Figure 3.18 where a monte carlo

simulation guarantees at 100% the revisit of the target in red dot according to the different points of view. The landmark painted in green are the landmark revisited before reaching the target.

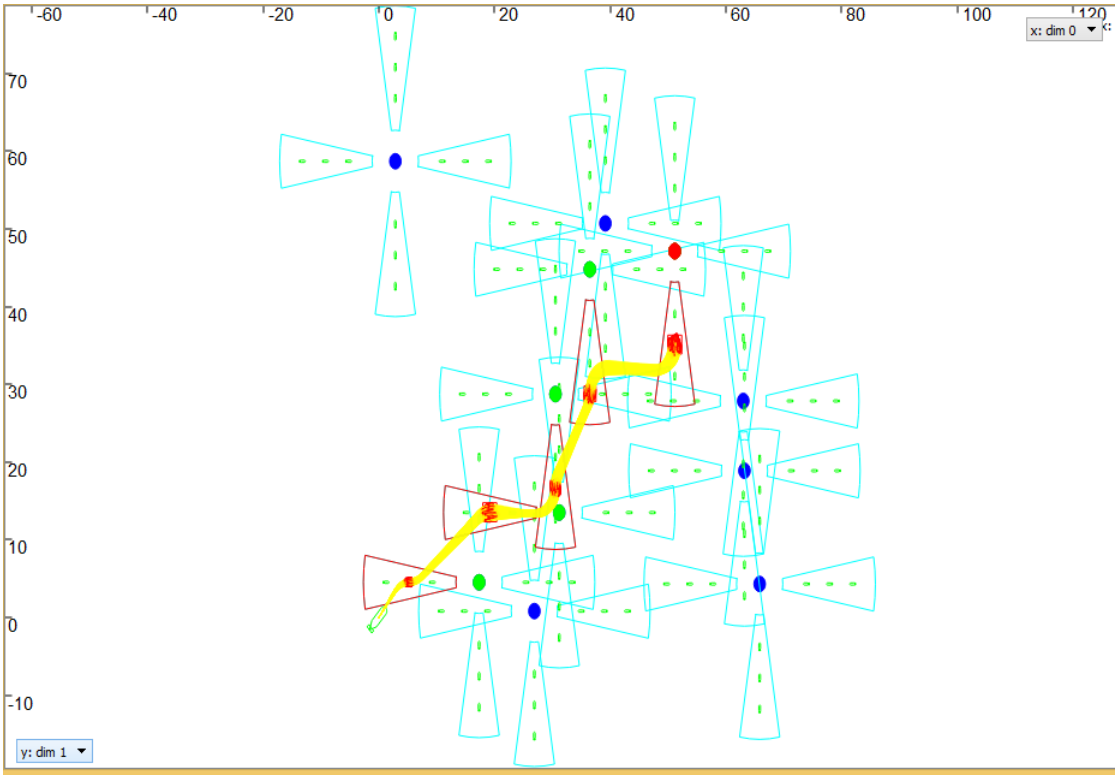


Figure 3.18: Engineering solution by expanding the 8 corners of a state.

Conclusion

In this second planner, the planning method takes into account the nonholonomic constraint of the vehicle, expressed as minimum turning radius, to propose a plan that reaches the target. Only punctual landmark were considered again and the notion of point of view has been introduced. In the context of this thesis, the advantage of such planner is the possibility to sample in the information space where a relocation process is possible. Indeed, contrary to grid planners, the time outside landmark field of influence is avoided. However, the strategies are limited and require to follow a predefined path. The differential command may be an alternative to relax this constraint of following a nominal path.

3.6 Conclusion

In this Chapter, most of the state-of-the-art algorithms in motion planning are presented. Methods dealing without uncertainty are firstly introduced to understand the concept and the basis of the motion planning. Due to the lack of absolute positioning system in underwater environment, a low cost robot that cannot rely on expensive inertial system has to deal with uncertainty. Consequently a plan that does not consider the propagation of the uncertainty will likely fail. The main motion planning methods dealing with uncertainty are then introduced to present the complexity and the difficulty to find a reliable and robust path. The notion of localization when dealing with motion planning is handled mainly around the mean state estimate of the robot and the plan generated is only possible if the controller is able to drive

the robot back to the desired path. Two planners based on a grid and the Probabilistic Path Planner are proposed in the context of the revisit of a target with a Forward Looking Sonar. In both cases, an extended (augmented) state space is considered, i.e. considering the uncertainty as a state variable, where the optimality of the path can be obtained. Due to the reduced ability of detection of the exteroceptive sensor in a low cost robot, i.e. lower resolution image, and the sparsity of landmarks present in most of underwater environment, a more complex strategy has to be investigated. Indeed, the main methods dealing with uncertainty and landmark relocation are based on the fact that a plan going to a landmark region will be detected if the uncertain robot state belongs to the field of influence of the landmark. However, due the weak density of landmarks, considering a set of closed landmarks may be interesting to extend the guaranteed area of exploration of the robot. Indeed, when considering a set of closed landmarks, the plan may not guarantee the revisit of a single landmark considered alone but can provide the guarantee to revisit one of them if considered together. In this case, the position of the robot will remain unknown when it enables to reach longer ranges. A disambiguation strategy has then to be realized to determine the position relatively to the set of landmarks. This solution has been adopted in this thesis and will be presented in Chapter 6. Due to the necessity of guarantee for operational customers, a set-membership approach has been prioritized. The next Chapter will introduce the basic knowledge of set-membership tools based on interval analysis to understand the planner proposed.

Chapter 4

Interval analysis

4.1 Introduction

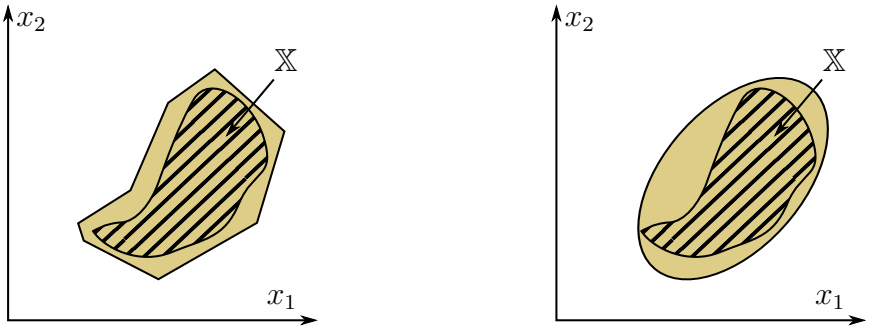
In this thesis, a motion plan based on set-membership methods has been preferred to classical bayesian planners to provide a guaranteed strategy if the errors remain in their bounds. These methods are highly used in localization problems where the position of the robot is determined by a set defined by non-linear constraints [138][142]. In the context of the set-membership approach, an approximation of this set needs to be determined to be handled by computers. Several approximations can be found in the litterature where some of them are presented in Figure 4.1 such as zonotopes or polyhedral enclosures [59], ellipsoids [169][276], intervals or subpavings [144]. These approximations corresponds to over-approximations. Interval analysis will be the focus in this thesis.

Interval analysis enables to compute a set of $\mathbb{R}^n$ defined by constraints. For example the set could be the parameters that are consistent with some intervals measurements [106], parameters vector of a controller in order to have a closed loop system stable [76], calibration parameters [264] or even attractors of dynamical systems [314].

Generally, a mathematical problem can be expressed with a *Constraint Network (CN)* where a set of *variables* belonging to some *domains* have to satisfy simulteneously a number of elementary rules called *constraints* [205].

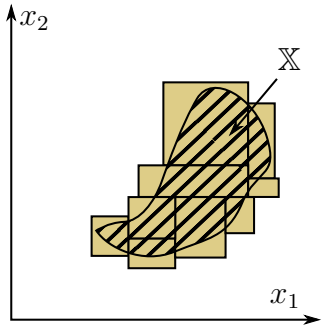
A Constraint Network is defined as a triple $\langle \mathbf{X}, \mathbb{D}, \mathcal{L} \rangle$ where $\mathbf{X} = \{x_1, ..., x_n\}$ is a set of variables, $\mathbb{D} = \{\mathbb{X}_1, ..., \mathbb{X}_n\}$ is a set of domains and $\mathcal{L} = \{\mathcal{L}_1, ..., \mathcal{L}_m\}$ is as set of constraints.

The problem consists in finding the smallest sub-domains of the $\mathbb{X}_i$ for the variables x_i consistent will all constraints. The variables are real numbers, vectors of $\mathbb{R}^n$ or even shapes, and the domains are intervals, boxes, sets or shape intervals in case of shape variable.



(a) Polytope approximation.

(b) Ellipsoid approximation.



(c) Subpavings approximation.

Figure 4.1: Examples of the approximation of the set $\mathbb{X}$

4.2 Set theory

According to the definition given by Georg Cantor, one of the founders of the set theory, in [38]:

> "A set is a gathering together into a whole of definite, distinct objects of our perception or of our thought, which are called elements of the set."

The objects may be numbers, people, graphs, letters of the alphabet, etc... The set of real numbers is denoted as $\mathbb{R}$.

4.2.1 Operations

The set theory is based on a binary relation that determines if an element x belongs or not to a set $\mathbb{X}$, denoted by $x \in \mathbb{X}$. When considering real numbers, the set of positive real numbers for example, noted $\mathbb{R}^+$, is defined according to:

$$\mathbb{R}^+ = \{x \in \mathbb{R} | x \geq 0\} \tag{4.1}$$

The empty set, denoted as $\emptyset$, contains no element and the universe Ω contains all elements. Some basic operations on sets can be defined.

The complement of a set $\mathbb{X}$, denoted as $\overline{\mathbb{X}}$ is:

$$\overline{\mathbb{X}} = \{x \in \Omega | x \notin \mathbb{X}\} \tag{4.2}$$

The inclusion between two sets $\mathbb{X}$ and $\mathbb{Y}$ is defined as follows:

$$\mathbb{X} \subset \mathbb{Y} \Leftrightarrow \forall x \in \mathbb{X}, x \in \mathbb{Y} \tag{4.3}$$

The equality between two sets $\mathbb{X}$ and $\mathbb{Y}$ is given as follows:

$$\mathbb{X} = \mathbb{Y} \Leftrightarrow (\mathbb{X} \subset \mathbb{Y} \wedge \mathbb{X} \subset \mathbb{Y}) \tag{4.4}$$

where the operator $\wedge$ is the logical *and* operator. $\vee$ denotes the *or* operator.

The intersection is defined as follows:

$$\mathbb{X} \cap \mathbb{Y} = \{x | x \in \mathbb{X} \wedge x \in \mathbb{Y}\} \tag{4.5}$$

and represented in Figure 4.2(a).

The union is defined as follows:

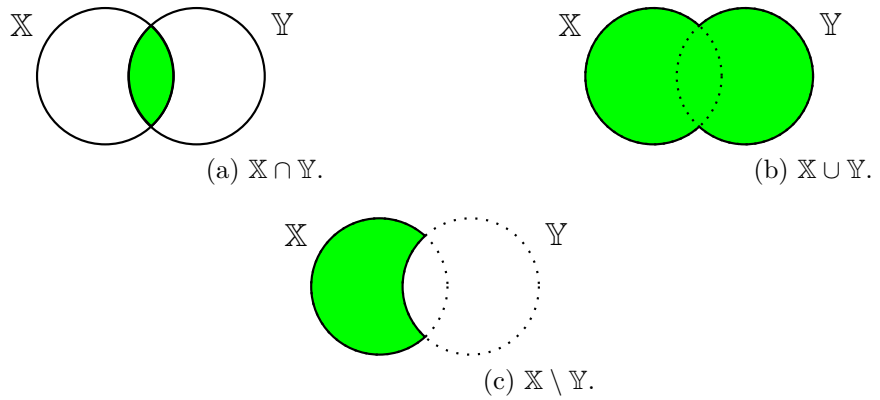
$$\mathbb{X} \cup \mathbb{Y} = \{x | x \in \mathbb{X} \vee x \in \mathbb{Y}\} \tag{4.6}$$

and represented in Figure 4.2(b).

The difference is defined as follows:

$$\mathbb{X} \setminus \mathbb{Y} = \{x | x \in \mathbb{X} \vee x \notin \mathbb{Y}\} = \mathbb{X} \cap \overline{\mathbb{Y}} \tag{4.7}$$

and represented in Figure 4.2(c).



(a) $\mathbb{X} \cap \mathbb{Y}$. (b) $\mathbb{X} \cup \mathbb{Y}$. (c) $\mathbb{X} \setminus \mathbb{Y}$.

Figure 4.2: Main operators on sets.

The Cartesian product is defined as follows:

$$\mathbb{X} \times \mathbb{Y} = \{(x, y) | x \in \mathbb{X} \wedge y \in \mathbb{Y}\} \tag{4.8}$$

Two laws have been stated by De Morgan:

$$\overline{(\mathbb{X} \cup \mathbb{Y})} = \overline{\mathbb{X}} \cap \overline{\mathbb{Y}} \tag{4.9}$$

$$\overline{(\mathbb{X} \cap \mathbb{Y})} = \overline{\mathbb{X}} \cup \overline{\mathbb{Y}} \tag{4.10}$$

The q-relaxed-intersection has been defined in [136] where it enables solving inconsistent Constraint Network by relaxing a small number q of constraints. In the context of localization or parameters estimation, it enables to be robust to a given number of outliers (erroneous measurements). For N sets $\mathbb{X}$, ...,$\mathbb{X}_N$ of $\mathbb{R}^n$, the q-relaxed-intersection, denoted by $\overset{\{q\}}{\cap}\mathbb{X}_i$, is the set of all $x \in \mathbb{R}^N$ which belong to all $\mathbb{X}_i$'s except q at most. The relaxed intersection can be expressed as unions of intersection. Considering three sets $\mathbb{X}_1$, $\mathbb{X}_2$, $\mathbb{X}_3$, the 1-relaxed-intersection is defined as follows:

$$\overset{\{q\}}{\cap}\mathbb{X}_i = (\mathbb{X}_1 \cap \mathbb{X}_2) \cup (\mathbb{X}_1 \cap \mathbb{X}_3) \cup (\mathbb{X}_2 \cap \mathbb{X}_3) \tag{4.11}$$

4.2.2 Set image

If a function $f : \mathbb{X} \to \mathbb{Y}$ and $\mathbb{X}_1 \subset \mathbb{X}$, the direct image by f of $\mathbb{X}_1$ is:

$$f(\mathbb{X}_1) = \{f(x) \subset \mathbb{Y} | x \in \mathbb{X}_1\} \tag{4.12}$$

The reciprocal image f of $\mathbb{Y}_1 \subset \mathbb{Y}$ is:

$$f^{-1}(\mathbb{Y}_1) = \{x \in \mathbb{X} | f(x) \in \mathbb{Y}_1\} \tag{4.13}$$

Interval analysis has been proved to be efficient in the Set Inversion Problem. Given a function $\mathbf{f}$ from $\mathbb{R}^n$ to $\mathbb{R}^m$ that may be non-linear, and $\mathbb{Y}$ a subset of $\mathbb{R}^m$, the Set Inversion Problem aims at computing the reciprocal image of $\mathbb{Y}$ by $\mathbf{f}$:

$$\mathbb{X} = \{x \in \mathbb{R}^n | f(x) \in \mathbb{Y}\} = \mathbb{Y} \tag{4.14}$$

The sets can be defined by constraints and combined according to some operators to build more complex ones. The next subsection introduces the interval analysis which is an efficient manner to deal with such sets defined by constraints.

4.3 Interval analysis

At the early stage, Interval Analysis was developed to quantify the error on numerical computations [225]. Indeed, real numbers are represented by float numbers with a limited number of digits. This limitation can lead to a small error that can be propagated and increased after several operations. Generally it refers to 32-bits floating point precision and others...

Intervals are bounds to represent a value of interest for computers where the limits are floating point numbers with finite precision ensuring a guaranteed numerical representation of a real number such as π. However, it introduces pessimism due to the fact it handles a range of possiblites instead of a unique value. Interval methods can then be extended to physical uncertainties highly present in robotics applications where errors exist on measurements for example. In this thesis, the focus will not be on the low level uncertainty of numbers but the concept of intervals will be used to manipulate the uncertainty on input parameters. For instance, a range sensor provides a distance with an error that can be represented as an interval. This subsection provides the basic knownledge of interval arithmetics. The reader may refer to [141] for more detailed information.

4.3.1 Interval arithmetics

4.3.1.1 Intervals

An *interval*, denoted by $[x]$, is a closed and connected subset of $\mathbb{R}$ defined by a lower bound x^- and an upper bound x^+:

$$[x] = [x^-, x^+] = \{x \in \mathbb{R} | x^- \leq x \leq x^+\} \tag{4.15}$$

An interval can be infinite.

An interval containing only one element is called degenerate or singleton, it is noted $\{x\}$. The width w of an interval is computed by $w([x]) = x^+ - x^-$. The mid is given by $mid([x]) = \frac{x^- + x^+}{2}$.

Considering two intervals $[x]$ and $[y]$ of $\mathbb{R}^n$ and an operator $\diamond \in \{+, -, \cdot, /\}$, $[x] \diamond [y]$ corresponds to the smallest interval containing all feasible values for $x \diamond y$ when $x \in [x]$ and $y \in [y]$:

$$[x] \diamond [y] = [\{x \diamond y | x \in [x], y \in [y]\}] \tag{4.16}$$

where [.] is the convex hull.

The intersection of two intervals is an interval:

$$[x] \cap [y] = \{a \in \mathbb{R} | a \in [x] \quad and \quad a \in [y]\} \tag{4.17}$$

However, the union of intervals may not be an interval:

$$[x] \cup [y] = \{a \in \mathbb{R} | a \in [x] \quad or \quad a \in [y]\} \tag{4.18}$$

The interval union corresponds to the interval hull of $[x] \cup [y]$ in order to have a connected subset of $\mathbb{R}$. This union is usually denoted $[x] \sqcup [y]$ and is defined as:

$$[x] \sqcup [y] = [[x] \cup [y]] \tag{4.19}$$

Examples

$$[-1, 2] + [3, 4] = [2, 6] \tag{4.20}$$
$$[-1, 2] - [3, 4] = [-5, -1] \tag{4.21}$$
$$[-1, 2] \cap [3, 4] = \emptyset \tag{4.22}$$
$$[-1, 4] \cap [2, 5] = [2, 4] \tag{4.23}$$
$$[-1, 3] \cup [4, 8] = [-1, 8] \tag{4.24}$$
$$1/[1, 2] = [0.5, 1] \tag{4.25}$$
$$1/[-1, 2] = [-\infty, \infty] \tag{4.26}$$
$$1/[0, 2] = [0.5, \infty] \tag{4.27}$$
$$w([-1, 2]) = 3 \tag{4.28}$$
$$mid([4, 8]) = 6 \tag{4.29}$$

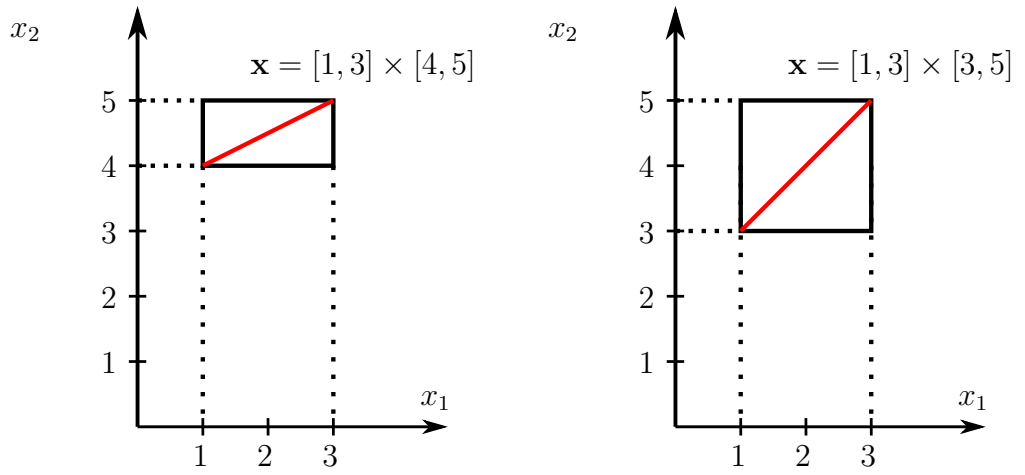
The notion of intervals can be extended to functions, shapes [75], booleans, graphs [138] or trajectories [274] for example.

4.3.1.2 Boxes

A *box* is an interval vector $[\mathbf{x}]$ of $\mathbb{R}^n$ which corresponds to a Cartesian product of n intervals. The set of all boxes of $\mathbb{R}^n$ is denoted by $\mathbb{IR}^n$.

$$[\mathbf{x}] = [x_1] \times [x_2] \times ... \times [x_n] \tag{4.30}$$

The i^{th} component $[x_i]$ is the projection of $[x]$ onto the i^{th} axis.



(a) Box a: $w_\infty([\mathbf{x}]) = 2$ and $w([\mathbf{x}]) = \sqrt{5}$. (b) Box b: $w_\infty([\mathbf{x}]) = 2$ and $w([\mathbf{x}]) = \sqrt{8}$.

Figure 4.3: Boxes representation and comparison. The two boxes have the same width w_∞ although the area is different.

Some boxes are represented in Figure 4.3 where two different metrics are used to compute the width of a box $[x] \in \mathbb{IR}^n$.

The classical definition of the width w_∞ corresponds to the width of the largest interval along all dimensions:

$$w_\infty([\mathbf{x}]) = \max_{1 \leq i \leq n} w([x_i]) \tag{4.31}$$

The second definition is given by:

$$w([\mathbf{x}]) = \sqrt{\sum_{i=1}^{n} w([x_i])^2} \tag{4.32}$$

Some basic operations on real numbers and vectors can still be computed on interval vectors by considering each component and to vector such as dot or cross product.

Examples

Considering $[\mathbf{x}] = [1,3] \times [4,5]$ and $[\mathbf{y}] = [-1,1] \times [0,3]$:

$$[\mathbf{x}] + [\mathbf{y}] = [0,4] \times [4,8] \tag{4.33}$$
$$[\mathbf{x}] \cdot [\mathbf{y}] = [-1,45] \tag{4.34}$$
$$w([\mathbf{x}]) = \sqrt{5} \tag{4.35}$$
$$w_\infty([\mathbf{x}]) = 2 \tag{4.36}$$

where $\cdot$ is the dot product.

4.3.1.3 Inclusion function

Definition

By considering $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ and a subset $\mathbb{X} \subset \mathbb{R}^n$, the image set of $\mathbf{f}$ is defined as follows:

$$\mathbf{f}(\mathbb{X}) = \{\mathbf{y} \in \mathbb{R}^m | \exists \mathbf{x} \in \mathbb{X}, \mathbf{y} = \mathbf{f}(\mathbf{x})\} \tag{4.37}$$

Considering a n-dimensional box $[\mathbf{x}]$ as input of a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ it may result in a set that can have any shape, be non convex, composed of holes and even disconnected as depicted in Figure 4.4.

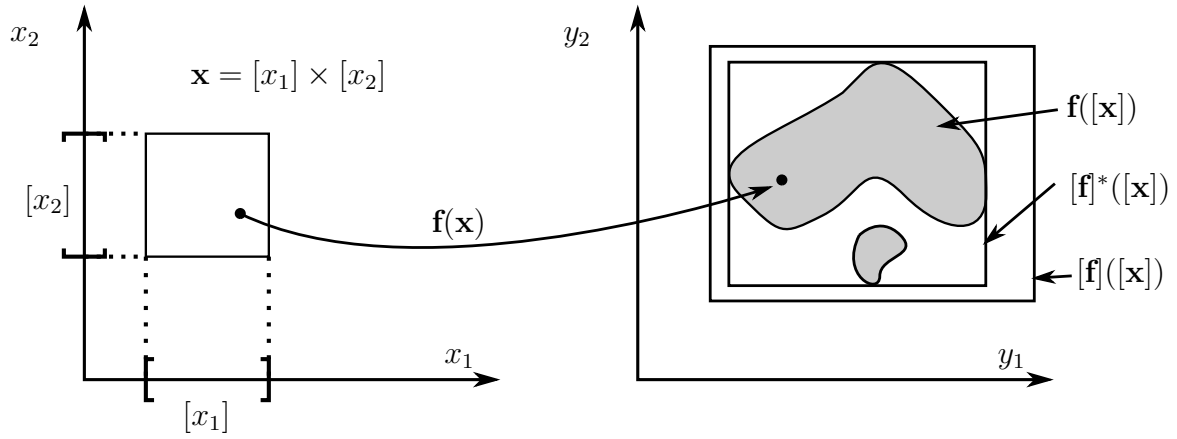


Figure 4.4: The image of a box $[\mathbf{x}]$ by a function $\mathbf{f}$ is depicted by the gray area. It may not be a box. The dotted area shows the pessimism introduced by the inclusion function $[\mathbf{f}]$ compared to the minimal inclusion function $[\mathbf{f}]^*$.

Computing an accurate representation of the image set can be complicated with a high computational burden. The notion of inclusion is then introduced.

An inclusion function $[\mathbf{f}] : \mathbb{IR}^n \to \mathbb{IR}^m$ of $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is a function that satisfies:

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}([\mathbf{x}])] \tag{4.38}$$

The inclusion function $[\mathbf{f}]$ of $\mathbf{f}$ enables to compute a box $[\mathbf{f}]([x])$ that guarantees to contain $\mathbf{f}([\mathbf{x}])$.

Properties

An inclusion function $[\mathbf{f}]$ is said to be *inclusion monotonic* if:

$$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow \mathbf{f}([\mathbf{x}]) \subset \mathbf{f}([\mathbf{y}]) \tag{4.39}$$

An infinity of inclusion functions exist for a given $\mathbf{f}$ but only one will be *minimal*, it is denoted $[\mathbf{f}]^*$. An inclusion function is minimal if for any $[\mathbf{x}]$, $[\mathbf{f}]([\mathbf{x}])$ is the smallest box containing $\mathbf{f}([\mathbf{x}])$. An inclusion function that is not minimal is said *pessimistic*.

Pessimism can come from the multiple occurrences of variables in the expression of $\mathbf{f}$ and the *wrapping effect*. This latter comes from the fact that intervals and boxes are axis-aligned. Therefore any set which is not a box made of boundaries aligned with axes will suffer from a pessimistic enclosure. Some methods are used to reduce this effect but will require longer computation times and more memory space.

An inclusion function $[\mathbf{f}]$ is said to be *thin* if the image for any degenerate interval vector $[\mathbf{x}] = \{\mathbf{x}\}$ is also punctual: $[\mathbf{f}]([\mathbf{x}]) = \{\mathbf{f}([\mathbf{x}])\}$ meaning that the image of a singleton is a singleton. This is not always the case when *thick* functions are used [75].

Natural inclusion function

A *natural inclusion function* can be built for any function that is obtained from the composition of elementary operators such as $+, -, \cdot, /, \cos, \sin, \exp$,... by replacing these operators by their interval counterpart $[cos], [sin]$,*etc.*

If $\mathbf{f}$ is continuous, the natural inclusion function is monotonic, convergent and thin but is generally not minimal due to the causes of pessimism presented earlier.

Examples

The natural inclusion function $[\mathbf{f}]$ of $\mathbf{f}(x_1, x_2) = x_1 \cdot \sin(x_2)$ is:

$$[\mathbf{f}]([x_1], [x_2]) = [x_1] \cdot [sin]([x_2]) \tag{4.40}$$

Considering $[\mathbf{x}] = [x_1] \times [x_2] = [-3, 1] \times [\frac{\pi}{3}, \frac{\pi}{2}]$, it gives:

$$\begin{aligned} [\mathbf{f}]([-3,1],[\frac{\pi}{4},\frac{\pi}{3}]) &= [-3,1] \cdot [sin]([\frac{\pi}{4},\frac{\pi}{3}]) \\ &= [-3,1] \cdot [\frac{\sqrt{2}}{2},\frac{\sqrt{3}}{2}] \\ &= [-\frac{3\sqrt{3}}{2},\frac{\sqrt{3}}{2}] \end{aligned} \tag{4.41}$$

4.4 Contractors and separators

In interval analysis, operators called contractors and separators have been created to characterize more efficiently and more rapidly the domain of the variables in a Constraint Network. They enable to remove parts of a box, i.e. interval vector, that do no satisfy the constraints. These operators are always associated to one or several constraints. This subsection introduces these operators.

4.4.1 Contractors

This operator reduces, or *contracts*, an initial box by removing parts that do not satisfy the constraint.

4.4.1.1 Definition

A contractor $\mathcal{C}$ is an operator $\mathbb{IR}^n \mapsto \mathbb{R}^n$[47] such that:

$$\mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}] \qquad \text{(contractance)} \tag{4.42}$$

$$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow \mathcal{C}([\mathbf{x}]) \subset \mathcal{C}([\mathbf{y}]) \qquad \text{(monotonicity)} \tag{4.43}$$

A set $\mathbb{X}$ is consistent with the contractor $\mathcal{C}$, it is written $\mathbb{X} \sim \mathcal{C}$ if for all $[\mathbf{x}]$:

$$\mathcal{C}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X} \tag{4.44}$$

A contractor $\mathcal{C}$ associated to a set $\mathbb{X}$ contracts a box of $\mathbb{R}^n$ without loosing a single point of the subset $\mathbb{X}$ of $\mathbb{R}^n$. This concept is illustrated in Figure 4.5.

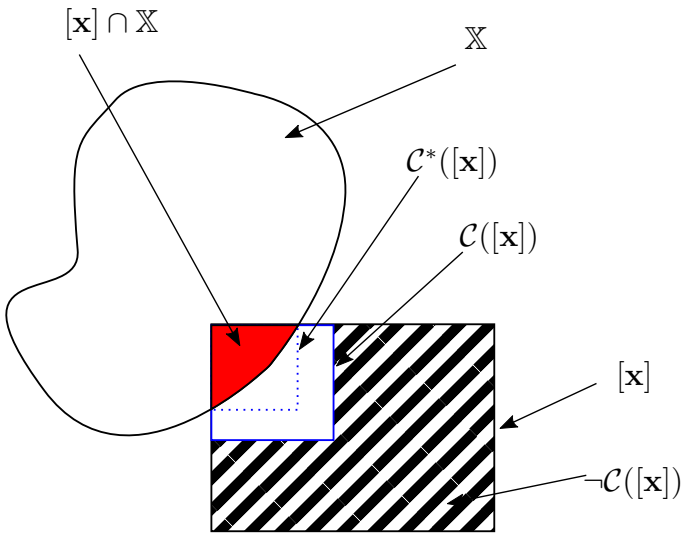


Figure 4.5: Contractor consistent with the set $\mathbb{X}$. The dashed area is removed by the contractor. The minimal contractor $\mathcal{C}^*$ returns the smallest box enclosing the solution set of $[\mathbf{x}] \cap \mathbb{X}$ in red.

The inclusion between two contractors $\mathcal{C}_1$ and $\mathcal{C}_2$ is defined as follows:

$$\mathcal{C}_1 \subset \mathcal{C}_2 \Leftrightarrow \forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}_1([\mathbf{x}]) \subset \mathcal{C}_2([\mathbf{x}]) \tag{4.45}$$

Two contractors $\mathcal{C}_1$ and $\mathcal{C}_2$ are equivalent ($\mathcal{C}_1 \sim \mathcal{C}_2$) if:

$$\mathbb{X} \sim \mathcal{C}_1 \Leftrightarrow \mathbb{X} \sim \mathcal{C}_2 \tag{4.46}$$

A contractor $\mathcal{C}$ is *minimal* if for any other contractor $\mathcal{C}_1$, the following implication is true:

$$\mathcal{C} \sim \mathcal{C}_1 \Rightarrow \mathcal{C} \subset \mathcal{C}_1 \tag{4.47}$$

If $\mathcal{C}$ is a minimal contractor consistent with $\mathbb{X}$, then $\mathcal{C}([\mathbf{x}])$ corresponds exactly to the smallest box that can be obtained after the contraction of $[\mathbf{x}]$ without loosing any point of $\mathbb{X}$. Therefore a unique minimal contractor exists and is denoted as $\mathcal{C}^*$.

The negation of a contractor $\mathcal{C}$, denoted as $\neg\mathcal{C}$, corresponds to:

$$\neg\mathcal{C}([\mathbf{x}]) = \{\mathbf{x} \in [\mathbf{x}] | \mathbf{x} \notin \mathcal{C}([\mathbf{x}])\} \tag{4.48}$$

Figure 4.5 shows the contractions of a contractor $\mathcal{C}$ and the minimal contractor $\mathcal{C}^*$ for the box $[\mathbf{x}]$ considered. The negation of the contractor $\mathcal{C}$ is not a box in general but an union of boxes represented by the dashed area. Therefore the negation of a contractor is not a contractor. This has been the motivation for the creation of the separators.

4.4.1.2 How to build a contractor?

When constraints are defined by equations or inequalities, such that $\mathbf{f}(\mathbf{x}) \in [\mathbf{y}]$, then a contractor based on the inclusion function $[\mathbf{f}]$ of $\mathbf{f}$ can be built. For a given box $[\mathbf{x}]$ it is defined as follows:

$$\mathcal{C}([\mathbf{x}]) = \begin{cases} \emptyset & \text{if} \quad [\mathbf{f}]([\mathbf{x}]) \cap [\mathbf{y}] = \emptyset \\ [\mathbf{x}] & \text{otherwise} \end{cases} \tag{4.49}$$

This has been improved with a forward/backward contractor called *HC4-revise* [22]. It consists in evaluating $\mathbf{f}(\mathbf{x})$ using interval arithmetic (forward step) and instead of only considering

the result of the intersection with $[\mathbf{y}]$, it retro-propagates the information to the initial domains (backward step). The following examples show this forward-backward concept.

Examples

Considering three variables $x \in [x]$, $y \in [y]$ and $z \in [z]$, the contractor for the constraint $z = x - y$ is given as follows:

$$\mathcal{C}_{-} : \begin{pmatrix} [z] \\ [x] \\ [y] \end{pmatrix} = \begin{pmatrix} [z] \cap ([x] - [y]) \\ [x] \cap ([z] + [y]) \\ [y] \cap ([x] + [z]) \end{pmatrix} \tag{4.50}$$

The first equation corresponds to the forward step and the last two equations are the backward steps.

For example, $\mathcal{C}_{-}([1,4],[2,5],[0,3]) = ([2,4],[2,3],[0,2])$.

4.4.1.3 Contractors algebra

It is possible to combine contractors to solve systems of constraints. Considering two contractors $\mathcal{C}_1$ and $\mathcal{C}_2$, the following operations are defined [47]:

$$(\mathcal{C}_1 \cap \mathcal{C}_2)([\mathbf{x}]) = \mathcal{C}_1([\mathbf{x}]) \cap \mathcal{C}_2([\mathbf{x}]) \tag{4.51}$$
$$(\mathcal{C}_1 \sqcup \mathcal{C}_2)([\mathbf{x}]) = \mathcal{C}_1([\mathbf{x}]) \sqcup \mathcal{C}_2([\mathbf{x}]) \tag{4.52}$$
$$(\mathcal{C}_1 \circ \mathcal{C}_2)([\mathbf{x}]) = \mathcal{C}_1(\mathcal{C}_2([\mathbf{x}])) \tag{4.53}$$
$$\mathcal{C}_1^{\infty}([\mathbf{x}]) = \mathcal{C}_1 \circ \mathcal{C}_1 \circ ... \circ \mathcal{C}_1([\mathbf{x}]) \tag{4.54}$$

where $\sqcup$ is the union hull defined as follows:

$$[\mathbf{x}] \sqcup [\mathbf{y}] = [\![[\mathbf{x}] \cup [\mathbf{y}]]\!] \tag{4.55}$$

Consequently, complex contractors can be built based on primitive contractors.

Example

Consider the range-only localization problem, where a robot described by its position $\mathbf{x} = (x_1, x_2)$ measures with an exteroceptive sensor some distances to a set of three landmarks (or beacons) M_i with $i \in \{1,2,3\}$ defined by their position $\mathbf{m}_i = (m_x^i, m_y^i)$. Figure 4.6(a) depicts the situation where the position of the robot is drawn by a yellow AUV. The landmarks are represented by red dots. This leads to the following observation function on the landmark i:

$$d_i = \sqrt{(x_1 - m_x^i)^2 + (x_2 - m_y^i)^2} \tag{4.56}$$

where d_i is the distance measured. This latter is not known accurately and lies in an interval: $d_i \in [d_i]$. The errors on the measurements is $30cm$. Table 4.1 indicates the position of the landmarks, the true distances d_i^* and the bounded measurements (distances).

	$\mathbf{m}_i$	d_i^*	$[d_i]$
M_1	$(15,4)$	10.05	$[9.75, 10.35]$
M_2	$(0,6)$	5.1	$[4.8, 5.4]$
M_3	$(8,9)$	5	$[4.7, 5.3]$

Table 4.1: Landmark's positions and measurements.

The set of feasible positions of the robot according to the measurement on the i^{th} landmark is defined by:

$$\mathbb{X}_i = \{\mathbf{x} \in \mathbb{R}^2 | \exists d_i \in [d_i], (x_1 - m_x^i)^2 + (x_2 - m_y^i)^2 = d_i\} \tag{4.57}$$

These sets for each landmark measurements correspond to rings around the position of the landmarks as it can be seen in Figure 4.6(a).

A contractor can be built based on this observation function considering the contractors $\mathcal{C}_-$, $\mathcal{C}_+$, $\mathcal{C}_{(.)^2}$ and $\mathcal{C}_{\sqrt{.}}$ which are elementary constraints. Using a forward/backward propagation algorithm, a contractor $\mathcal{C}_{d_i}$ consistent with $\mathbb{X}_i$ is then built by combining these primitive contractors for each landmark. Therefore, three contractors are built.

The position of the robot is given by the intersection of the three sets $\mathbb{X}_i$ and then is given by:

$$\mathbb{X} = \mathbb{X}_1 \cap \mathbb{X}_2 \cap \mathbb{X}_3 \tag{4.58}$$

The contractor $\mathcal{C}_{\mathbb{X}}$ consistent with the solution set $\mathbb{X}$ is given by:

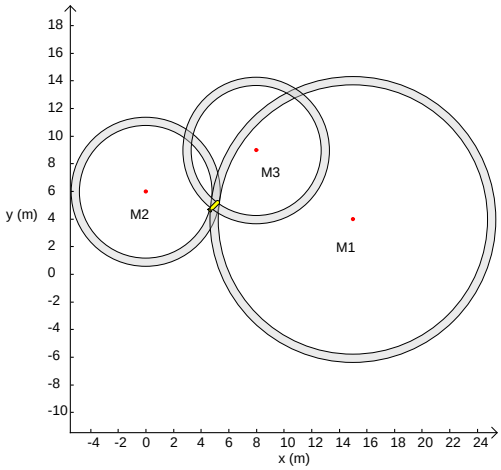
$$\mathcal{C}_{\mathbb{X}} = \mathcal{C}_{d_3} \circ \mathcal{C}_{d_2} \circ \mathcal{C}_{d_1} \tag{4.59}$$

the composition of the three contractors.

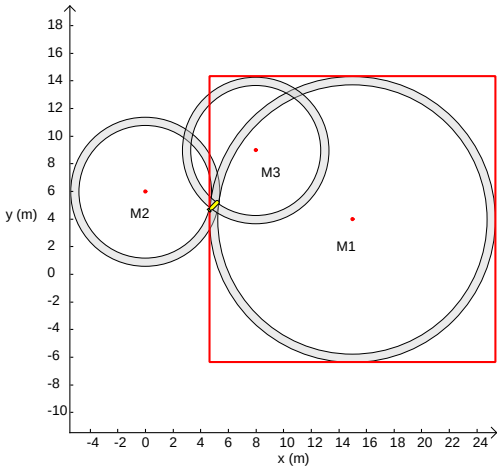
When using contractors, the idea is to get the smallest box enclosing the solution set. The contractor $\mathcal{C}_{\mathbb{X}}$ is then iteratively composed, denoted as $\mathcal{C}^{\infty} = \mathcal{C}_{\mathbb{X}} \circ \mathcal{C}_{\mathbb{X}} \circ ... \circ \mathcal{C}_{\mathbb{X}}$ until a fixed point is reached. Due to the monotonicity of the contractors, it can be proved that the iterated $\mathcal{C}^{\infty}$ will always converge to the smallest box, regardless the order of the contractors in the composition [222].

The initial position of the robot is assumed unknown $[\mathbf{x}]_0 = [-\infty, \infty]^2$.

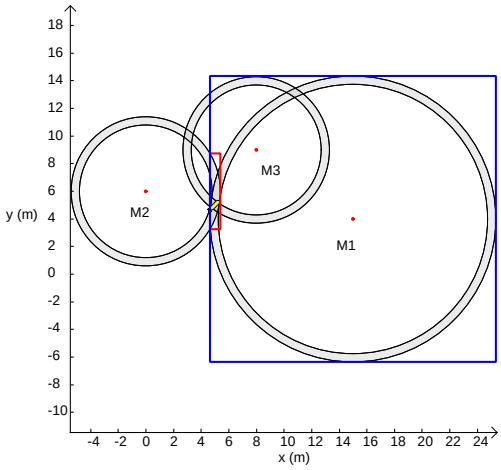
Figure depicts 4.6, step by step, the propagation process. Firstly, the position of the robot is unknown as no measurements are given. Any box can enclose the position as shown in Figure 4.6(a). The first contraction $\mathcal{C}_{d_1}([\mathbf{x}]_0)$ based on the bounded range measurement coming from M_1 gives a box that encloses the ring around M_1 and is shown by a red box in Figure 4.6(b). Based on the second measurement, the second contraction $\mathcal{C}_{d_2}$ on the box obtained (from the first contraction), represented in blue in Figure 4.6(c), gives a new red box, that is consistent with the constraint on the second measurement associated to the second landmark. Finally, similarly, the third measurement coming from the third landmark reduces the box size in Figure 4.6(d). Until now, one iteration step $\mathcal{C}_{\mathbb{X}}$ has been proposed. This process is iterated until no more contraction is possible as depicted in Figure 4.6(e) after 10 iterations. A zoom is provided in Figure 4.6(f). The result gives an over-approximation (outer-approximation) of the solution. To get a better approximation of the solution and to reduce the pessimism of the solution, the idea is to bisect (split into two parts) the final box obtained after the 10 iterations and then apply again the contractor $\mathcal{C}_{\mathbb{X}}$ on each part. This process can be recursively applied until the width of the resulting subboxes is smaller than a given threshold. This concept will be presented by the SIVIA algorithm.



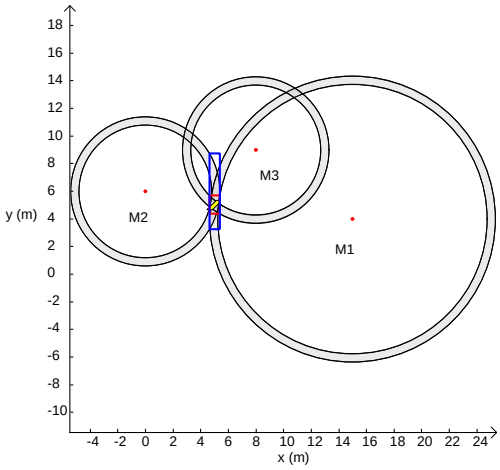
(a) Initial situation: the position of the robot is unknown.



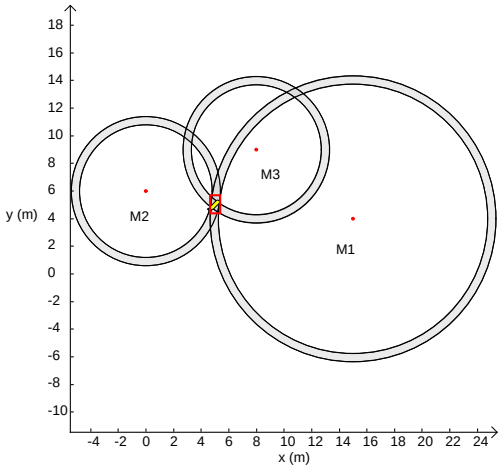
(b) Contraction from M1's measurement. The red box depicts the contraction.



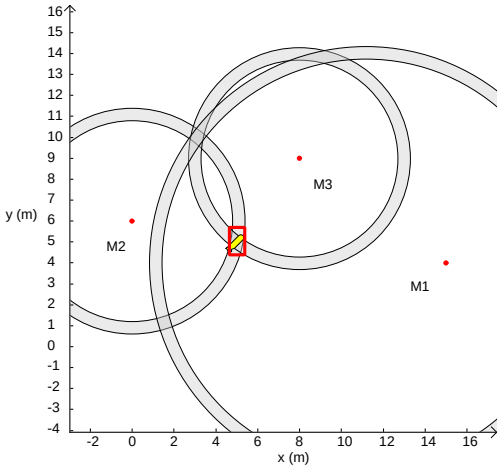
(c) Contraction from M2's measurement. The blue box represents the box before the contraction.



(d) Contraction from M3's measurement.



(e) Fixed point result after 10 iterations.



(f) Zoom on the fixed point result.

Figure 4.6: Localization with contractors based on range-only measurements. The landmarks are depicted by red dots and the true position of the robot is drawn by a yellow AUV. It shows the successive contractions coming from the range bounded measurements.

4.4.2 Separators

The contractor consistent with a given set $\mathbb{X}$ only provides an outer approximation of the solution set. If the set $\mathbb{X}$ has not an empty volume (points exist in $\mathbb{X}$ that do not are on the border), it can be interesting to prove that a part of the initial space belongs to this solution set. This can be done by considering the complementary contractor consistent with $\overline{\mathbb{X}}$. An inner and an outer approximation of the solution can be characterized with a separator [140].

4.4.2.1 Definition

A separator $\mathcal{S}$ associated to a set $\mathbb{X}$ is defined as follows:

$$\begin{aligned} \mathcal{S}\colon \mathbb{IR}^n &\to \mathbb{IR}^n \times \mathbb{IR}^n \\ [\mathbf{x}] &\mapsto ([\mathbf{x}_{in}], [\mathbf{x}_{out}]) \end{aligned} \tag{4.60}$$

with the properties:

$$\begin{aligned} [\mathbf{x}] &= [\mathbf{x}_{in}] \cup [\mathbf{x}_{out}] \\ [\mathbf{x}_{out}] \cap \mathbb{X} &= [\mathbf{x}] \cap \mathbb{X} \\ [\mathbf{x}_{in}] \cap \overline{\mathbb{X}} &= [\mathbf{x}] \cap \overline{\mathbb{X}} \end{aligned} \tag{4.61}$$

A separator can be seen as a pair of contractors $\{\mathcal{S}^{in}, \mathcal{S}^{out}\}$ and so for all $[\mathbf{x}] \in \mathbb{IR}^n$:

$$\mathcal{S}^{in}([\mathbf{x}]) \cup \mathcal{S}^{out}([\mathbf{x}]) = [\mathbf{x}] \quad \text{(complementary)} \tag{4.62}$$

A set $\mathbb{X}$ is consistent with the separator $\mathcal{S}$ ($\mathbb{X} \sim \mathcal{S}$) if:

$$\mathbb{X} \sim \mathcal{S}^{out} \quad \text{and} \quad \overline{\mathbb{X}} \sim \mathcal{S}^{in} \tag{4.63}$$

where $\overline{\mathbb{X}} = \{\mathbf{x} | \mathbf{x} \notin \mathbb{X}\}$. The separator can then be rewritten as the pair of contractors $\{\mathcal{C}, \overline{\mathcal{C}}\}$ where $\mathcal{C}$ is consistent with $\mathbb{X}$ and $\overline{\mathcal{C}}$ is consistent with $\overline{\mathbb{X}}$ ($\mathcal{C} \sim \mathcal{S}^{out}$ and $\overline{\mathcal{C}} \sim \mathcal{S}^{in}$). The concept is depicted in Figure 4.7.

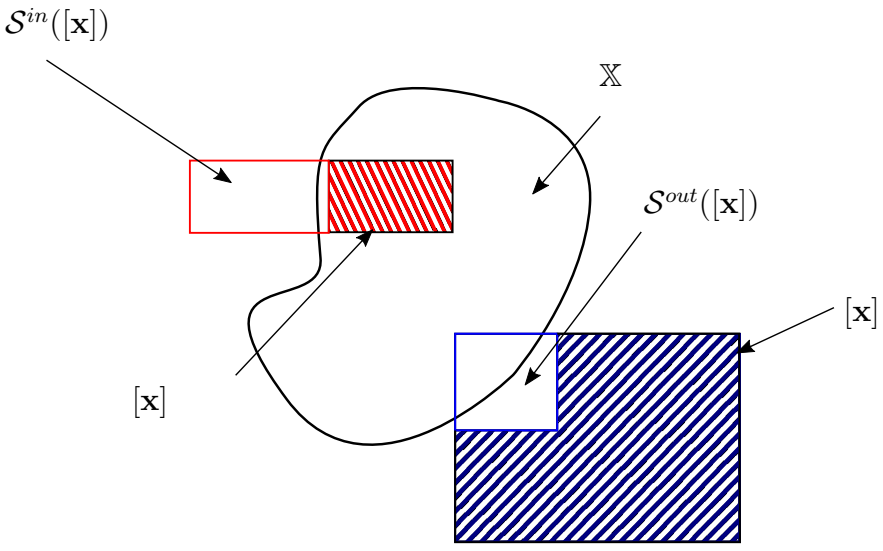


Figure 4.7: Separator applied on two boxes. The outer contractor removes the blue dashed area and the inner contractor the red dashed area.

The inclusion between two separators $\mathcal{S}_1$ and $\mathcal{S}_2$ can be defined as follows:

$$\mathcal{S}_1 \subset \mathcal{S}_2 \Leftrightarrow \mathcal{S}_1^{in} \subset \mathcal{S}_2^{in} \text{ and } \mathcal{S}_1^{out} \subset \mathcal{S}_2^{out} \tag{4.64}$$

A separator $\mathcal{S}$ is *minimal* if:

$$\mathcal{S}_1 \subset \mathcal{S} \Rightarrow \mathcal{S}_1 = \mathcal{S} \tag{4.65}$$

If $\mathcal{S}$ is minimal, then the two contractors $\mathcal{S}^{in}$ and $\mathcal{S}^{out}$ are both minimal.

4.4.2.2 Separators algebra

The separator algebra is an extension of the contractor algebra [47].

Considering a separator $\mathcal{S} = \{\mathcal{S}^{in}, \mathcal{S}^{out}\}$, the complement is defined as:

$$\overline{\mathcal{S}} = \{\mathcal{S}^{out}, \mathcal{S}^{in}\} \tag{4.66}$$

Considering two separators $\mathcal{S}_1 = \{\mathcal{S}_1^{in}, \mathcal{S}_1^{out}\}$ and $\mathcal{S}_2 = \{\mathcal{S}_2^{in}, \mathcal{S}_2^{out}\}$, some operations can be defined:

$$\begin{aligned}
\mathcal{S}_1 \cap \mathcal{S}_2 &= \{\mathcal{S}_1^{in} \cup \mathcal{S}_2^{in}, \mathcal{S}_1^{out} \cap \mathcal{S}_2^{out}\} && (4.67)\\
\mathcal{S}_1 \cup \mathcal{S}_2 &= \{\mathcal{S}_1^{in} \cap \mathcal{S}_2^{in}, \mathcal{S}_1^{out} \cup \mathcal{S}_2^{out}\} && (4.68)\\
\mathcal{S}_1 \setminus \mathcal{S}_2 &= \mathcal{S}_1 \cap \overline{\mathcal{S}_2} && (4.69)\\
& && (4.70)
\end{aligned}$$

For more details about separators, the reader may refer to [140].
Some examples will be provided in the next subsection when paver will be introduced.

4.4.3 Paver

The pessimism of some solution sets is highly important when the solution is made of holes or non-connected subsets for example. The previous example on the range only localization problem shows that the result was enclosed by a box at the intersection of the rings which was an over approximation. The enclosure will only provide a solution with a strong pessimism. As mentioned earlier, a solution to overcome this issue would be to divide the resulting box and continue the contraction. This is the concept of the pavers. It results finally in a subpaving which is the union of non-overlapping boxes $[\mathbf{x}]^i$ included in $[\mathbf{x}]$. A subpaving that completely covers $[\mathbf{x}]$ is called a paving of $[\mathbf{x}]$. A thinner approximation of a set $\mathbb{X}$ can be done by characterizing two subpavings $\mathbb{X}^-$ and $\mathbb{X}^+$ such that:

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+ \tag{4.71}$$

$\mathbb{X}^-$ corresponds to the inner approximation of the set $\mathbb{X}$ and $\mathbb{X}^+$ to the outer approximation as presented in Figure 4.8.

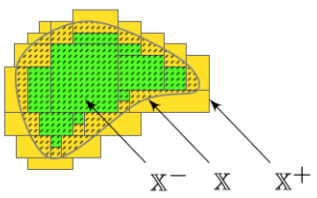


Figure 4.8: Inner and outer approximation of the set $\mathbb{X}$. $\mathbb{X}^-$ and $\mathbb{X}^+$ are two subpavings enclosing the solution set. The boundary of the set belongs to the outer approximation and not the inner one. Picture extracted from [274].

Initially, the paver, called SIVIA[1] [144], was developed to compute the reciprocal image $\mathbb{X} \subset \mathbb{R}^n$ such that $\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$ where $\mathbb{Y} \subset \mathbb{R}^m$ is the image of the set $\mathbb{X}$ by a possibly non-linear function $\mathbf{f}$: $\mathbb{R}^n \to \mathbb{R}^m$, known as the set inversion problem, formalized as follows:

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}) \tag{4.72}$$

By considering the inclusion function $[\mathbf{f}]: \mathbb{IR}^n \to \mathbb{IR}^m$ of the function $\mathbf{f}$, this algorithm is able to provide the approximation of the set $\mathbb{X}$ between the two subpavings $\mathbb{X}^-$ and $\mathbb{X}^+$. A recursive version of the SIVIA algorithm is proposed in Algorithm 7. The parameter ϵ is the precision required.

Algorithm 7 SIVIA(**in** : $[\mathbf{f}]$,$[\mathbf{x}]$,$\mathbb{Y}$,ϵ **inout**: $\mathbb{X}^-$,$\mathbb{X}^+$)

```
1:  if [f]([x]) ∩ Y ≠ ∅ then
2:      if [f]([x]) ⊂ Y then
3:          X+ ← X+ ∪ [x]                                   ▷ outer set
4:          X− ← X− ∪ [x]                                   ▷ inner set
5:      else if w∞([x]) < ε then
6:          X+ ← X+ ∪ [x]                                   ▷ outer set only ([x] on the border)
7:      else
8:          bisect([x]) into [x]1 and [x]2
9:          SIVIA([f],[x]1,Y,ε,X−,X+)
10:         SIVIA([f],[x]2,Y,ε,X−,X+)
```

Starting from an initial defined box $[\mathbf{x}]_0 \in \mathbb{IR}^n$, it consists in deciding if the box belongs to $\mathbb{X}^+$, both $\mathbb{X}^+$ and $\mathbb{X}^-$ or none with some inclusion tests. If it cannot decide, as mentioned earlier, it bisects the box and apply again the algorithm to test the two resulting boxes. Different tests are checked in Algorithm 7:

- $[\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset$: $[\mathbf{x}]$ does not belong to $\mathbb{X}$ (line 1).
- $[\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y}$: any vector in $[\mathbf{x}]$ is solution ($[\mathbf{x}] \subset \mathbb{X}$), therefore $[\mathbf{x}]$ is stored in both $\mathbb{X}^-$ and $\mathbb{X}^+$ (lines $2, 3, 4$).
- $[\mathbf{f}]([\mathbf{x}])$ has a non empty intersection with $\mathbb{Y}$ but it is not a subset of $\mathbb{Y}$. This case is said to be undetermined.

In that last case, two options are possible:

- if the width of the box $[\mathbf{x}]$ is smaller than the criterion ϵ (line 5), then the box is considered too small with respect to the precision. This box is stored in $\mathbb{X}^+$ (line 6). It probably contains the border of the set as depicted in Figure 4.8.
- otherwise (line 7), the box is bisected along the largest dimension (line 8) for example, and the algorithm is applied again on these resulting subboxes (lines 9 and 10).

In this SIVIA algorithm, it only tests boxes with the inclusion function and it does deal with contractors and separators. The precision of the approximation is handled by the precision parameter ϵ that defines the width of the interval $[\mathbb{X}^-, \mathbb{X}^+]$. The thinner the parameter is, the better the approximation is.

The contractors and the separators have been developed to characterize more efficiently and more rapidly the domain of the variables in a constraint network. This SIVIA algorithm has

[1] Set-Inversion via Interval Analysis (SIVIA)

then been adapted to deal with contractors and separators. It results in a reduced number of bisections needed to get the approximation. Algorithm 8 proposes the version adapted to the separator as it can provide an inner ($\mathbb{X}^-$) and an outer approximation ($\mathbb{X}^+$) of a set $\mathbb{X}$, contrary to the contractors that can only provide an outer approximation. A separator $\mathcal{S}$ is built according to a constraint network defining the set that is aimed to be characterized.

Firstly the algorithm initializes a list $\mathcal{L}$ (line 1) containing all the initial boxes. A single box $[\mathbf{x}]_0$ is usually stored in the list, but it is possible to partition the space in non-overlapping boxes before. Secondly it pulls a box $[\mathbf{x}]$ from the list (line 3) and uses the separator $\mathcal{S}$ to contract the box into two boxes $[\mathbf{x}_{in}]$ and $[\mathbf{x}_{out}]$ (line 4). Then it stores the part $(\neg\mathcal{S}^{in}([\mathbf{x}]))$ proved to be inside $\mathbb{X}$ into $\mathbb{X}^+$ and $\mathbb{X}^-$. Then it computes the border $\partial\mathcal{S}([\mathbf{x}])$ by intersecting the two resulting boxes $[\mathbf{x}_{in}]$ and $[\mathbf{x}_{out}]$. If this box is too small compared to the precision ϵ, it is stored inside $\mathbb{X}^+$. Otherwise it is bisected along its largest dimension and the two resulting boxes are stored in the list $\mathcal{L}$. The algorithm stops when no more box is inside the list. The final result is the enclosure $\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+$ with the two subpavings $\mathbb{X}^-$ and $\mathbb{X}^+$.

Algorithm 8 Separator paver(**in**: $[\mathbf{x}]$,$\mathcal{S}$,ϵ **out**: $\mathbb{X}^-$,$\mathbb{X}^+$)

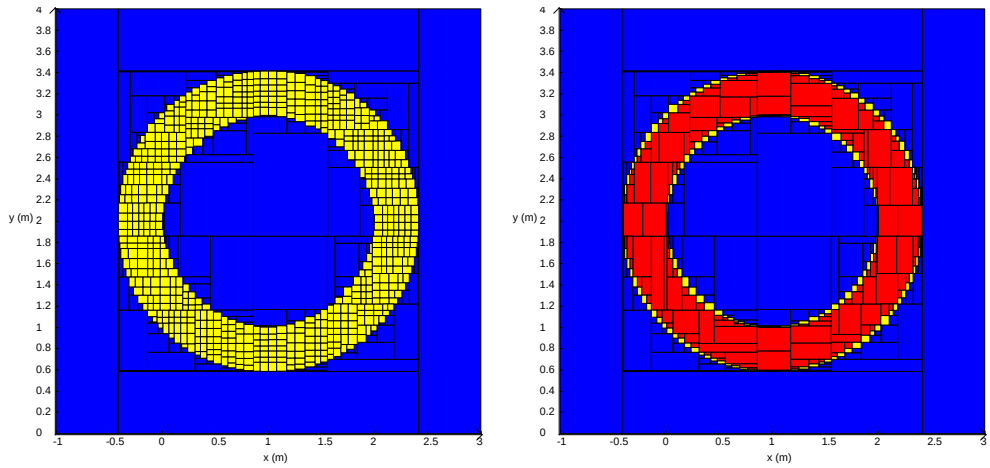
1: $\mathcal{L}$: $= \{[\mathbf{x}]\}$ ▷ initialize the list
2: **while** $\mathcal{L} \neq \emptyset$ **do**
3: Pull $[\mathbf{x}]$ from the list $\mathcal{L}$
4: $\{[\mathbf{x}_{in}], [\mathbf{x}_{out}]\} = \mathcal{S}([\mathbf{x}])$
5: Store $[\mathbf{x}] \setminus [\mathbf{x}_{in}]$ into $\mathbb{X}^-$ and into $\mathbb{X}^+$
6: $[\mathbf{x}] = [\mathbf{x}_{in}] \cap [\mathbf{x}_{out}]$
7: **if** $w_\infty([\mathbf{x}]) < \epsilon$ **then**
8: Store $[\mathbf{x}]$ into $\mathbb{X}^+$
9: **else**
10: Bisect $[\mathbf{x}]$ and push the two resulting boxes into $\mathcal{L}$

Example 1

Consider the set:

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^2, (x_1 - 1)^2 + (x_2 - 2)^2 \in [1, 2]\} \tag{4.73}$$

which is a ring centered at $(1, 2)$ with a thickness of 1. The minimal contractor $\mathcal{C}_{\mathbb{X}}$ consistent with $\mathbb{X}$ is built using the classical forward/backward propagation. A paver adapted to the contractor (not given here) provides an outer approximation of the solution set depicted in Figure 4.9(a). It only removes the part of the space outside $\mathbb{X}$ (drawn in blue). Due to the consistency (Equation 4.44), the contractor $\mathcal{C}_{\mathbb{X}}$ cannot contract boxes already included in $\mathbb{X}$. Consequently it bisects the boxes until it reaches the threshold value ϵ.



(a) Paving from the contractor $\mathcal{C}_{\mathbb{X}}$. Blue boxes are outside the solution set and no conclusion can be made for yellow boxes.

(b) Paving from the separator $\mathcal{S}_{\mathbb{X}}$. Blue boxes are outside the solution set, red ones are inside and no conclusion can be made for yellow boxes.

Figure 4.9: Approximations of the set $\mathbb{X}$.

By considering the complementary $\overline{\mathbb{X}}$:

$$\overline{\mathbb{X}} = \{\mathbf{x} \in \mathbb{R}^2, (x_1 - 1)^2 + (x_2 - 2)^2 \notin [1, 2]\} \tag{4.74}$$

the separator $\mathcal{S}_{\mathbb{X}}$ consistent with $\mathbb{X}$ can be built based on the pair of the contractors $\{\mathcal{C}_{\mathbb{X}}, \mathcal{C}_{\overline{\mathbb{X}}}\}$. Using the paver given in Algorithm 8, the result of the separator is depicted in Figure 4.9(b). Now it is possible to get an inner approximation $\mathbb{X}^-$ painted in red. The blue boxes are outside $\mathbb{X}$. The yellow boxes are on the frontier and belong, as mentioned, to the outer approximation $\mathbb{X}^+$ with the red ones. The parameter ϵ handles the precision of the paving ($\epsilon = 0.01$ here).

Example 2

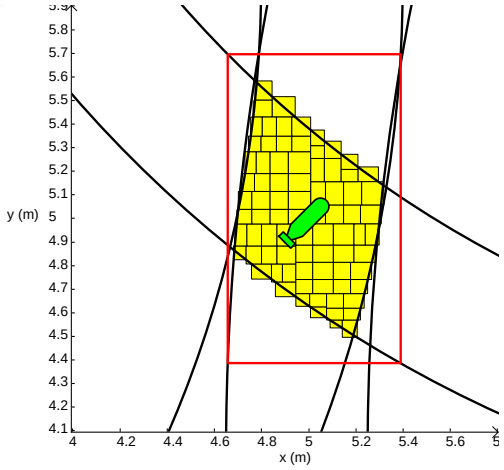
Come back to the previous example of the range only localization problem. The contractors were built for the three measurements on the three landmarks based on the bounded distances. The global contractor consistent with the set $\mathbb{X}$ that corresponds to the position of the robot based on the three bounded measurements and the position of the landmarks can be built as follows:

$$\mathcal{C}_{\mathbb{X}} = \mathcal{C}_{d_1} \cap \mathcal{C}_{d_2} \cap \mathcal{C}_{d_3} \tag{4.75}$$

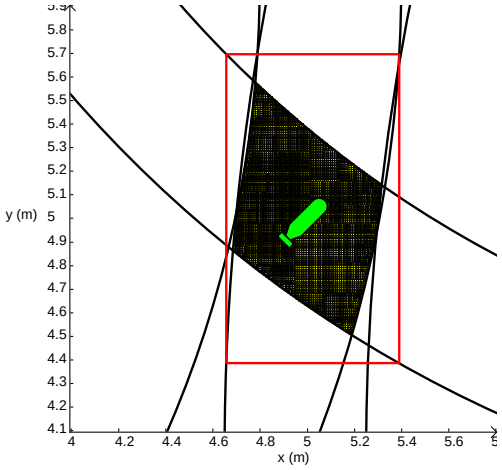
Similarly the separator $\mathcal{S}_{\mathbb{X}}$ can be built by considering the complementary sets of the three constraints.

The results using the contractor and the separator are depicted in Figure 4.10 where the box outside the set $\mathbb{X}$ are not represented for ease of interpretation. The box obtained by the iterative contractions at the fixed point is represented in red (enclosing box). Different values of ϵ are used to show the accuracy of the solution set. The figures on the left have a precision value of 0.1 and the figures on the right 0.01. The figures on the top are the pavings obtained with the contractor. When the precision is high (smallest ϵ) there are too many boxes such that only black contours are visible. Similarly it comes from the consistency property. At the bottom, the figures show the results of the separator. It is possible to characterize the inner solution of the problem. Notice that the results obtained from both are better approximations

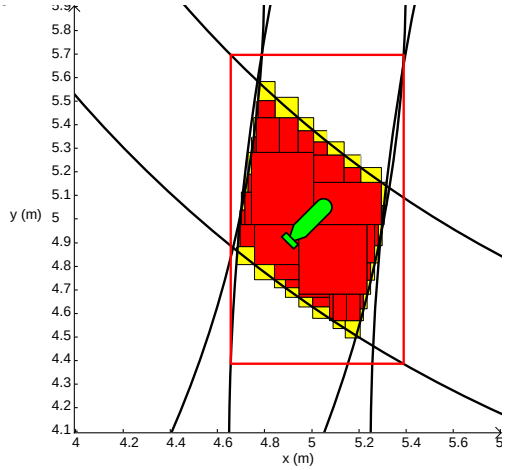
of the solution set than the enclosing box from the iterative contractions (red box). This comes from the bisections in the paver.



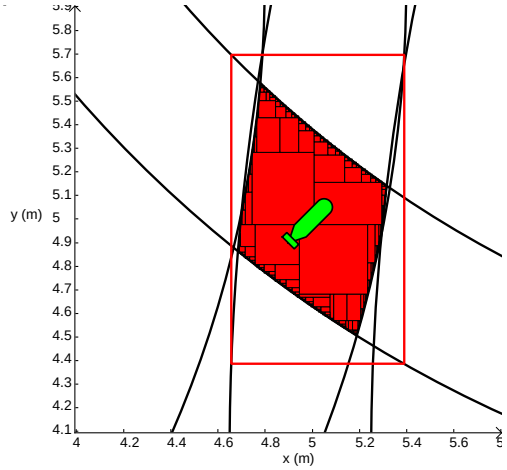
(a) Paving obtained using the contractor with $\epsilon = 0.1$.



(b) Paving obtained using the contractor with $\epsilon = 0.01$. The boxes are too small, the contour of the boxes in black is predominant.



(c) Paving obtained using the separator with $\epsilon = 0.1$.



(d) Paving obtained using the separator with $\epsilon = 0.01$.

Figure 4.10: Approximations of the solution set for the range only localization problems. Boxes outside (generally blue) of the set $\mathbb{X}$ are not represented for ease of interpretation. A zoom on the solution set is depicted. The true position of the AUV is shown in green. The big red box was the box obtained after the successive contractions at the fixed point.

4.4.4 Inversion of separators

A set can be represented by a separator. The inversion of a separator [140] enable to get the inverse image of a set by a function as in the set inversion problem defined in the Equation 4.72 and remind here.

The inverse of a set $\mathbb{Y} \subset \mathbb{R}^m$ by a function $\mathbf{f}$: $\mathbb{R}^n \to \mathbb{R}^m$ is expressed as follows:

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}) \tag{4.76}$$

If a contractor $\mathcal{C}_{\mathbb{Y}}$ consistent with $\mathbb{Y}$ is available, a contractor $\mathcal{C}_{\mathbb{X}}$ for $\mathbb{X}$ can be defined as shown in [46]. The contractor $\mathcal{C}_{\mathbb{X}}$ is called the inverse of $\mathcal{C}_{\mathbb{Y}}$ by $\mathbf{f}$. Similarly the inverse of a separator [140] $\mathcal{S}_{\mathbb{Y}}$ associated to a set $\mathbb{Y}$ can be defined:

$$\mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}) = \{\mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{in}), \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{out})\} \tag{4.77}$$

Theorem 4.4.1 *The separator $\mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}})$ is a separator consistent with the set $\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$:*

$$\mathbf{f}^{-1}(\mathbb{Y}) \sim \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}) \tag{4.78}$$

Proof

$$\begin{array}{rc} & \mathbb{Y} \sim \{\mathcal{S}_{\mathbb{Y}}^{in}, \mathcal{S}_{\mathbb{Y}}^{out}\} \\ \Leftrightarrow & \mathbb{Y} \sim \mathcal{S}_{\mathbb{Y}}^{out}, \overline{\mathbb{Y}} \sim \mathcal{S}_{\mathbb{Y}}^{in} \\ \Rightarrow & \mathbf{f}^{-1}(\mathbb{Y}) \sim \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{out}), \mathbf{f}^{-1}(\overline{\mathbb{Y}}) \sim \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{in}) \\ \Leftrightarrow & \mathbf{f}^{-1}(\mathbb{Y}) \sim \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{out}), \overline{\mathbf{f}^{-1}(\mathbb{Y})} \sim \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{in}) \\ \Leftrightarrow & \mathbf{f}^{-1}(\mathbb{Y}) \sim \{\mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{in}), \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{out})\} \\ \Leftrightarrow & \mathbf{f}^{-1}(\mathbb{Y}) \sim \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}) \end{array} \tag{4.79}$$

Definition

If $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a bijective function, then the image by $\mathbf{f}$ of a separator $\mathcal{S}_{\mathbb{X}}$ is defined as follows:

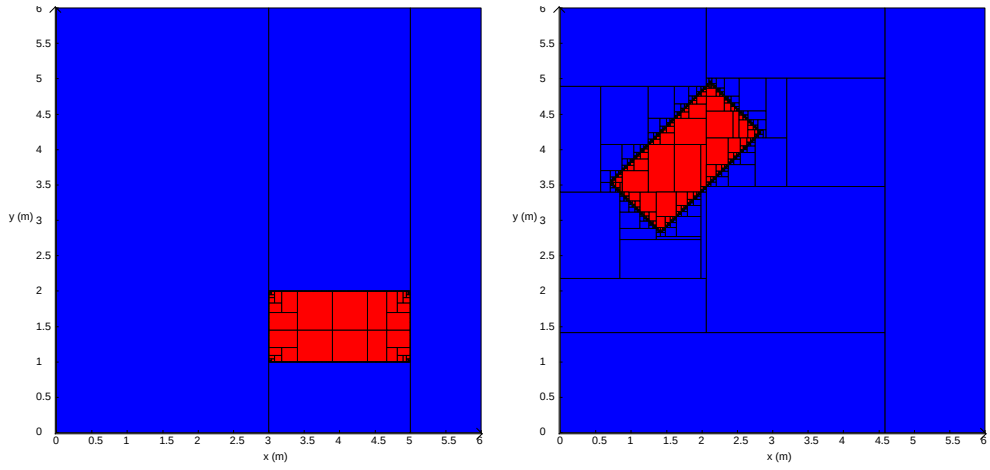
$$\mathbf{f}(\mathcal{S}_{\mathbb{X}}) = \mathbf{f} \circ \mathcal{S}_{\mathbb{X}} \circ \mathbf{f}^{-1} \tag{4.80}$$

Example

Consider the set defined by:

$$\mathbb{Y} = \{\mathbf{y} \in \mathbb{R}^2 | y_1 \in [3,5] \quad \text{and} \quad y_2 \in [1,2]\} \tag{4.81}$$

and the rotation of angle $\frac{\pi}{4}$ as function $\mathbf{f}$ denoted $\mathcal{R}_{\frac{\pi}{4}}$. This is a bijective function meaning that the inverse function can be computed $\mathbf{f}^{-1}$ which corresponds to $\mathcal{R}_{-\frac{\pi}{4}}$. A separator $\mathcal{S}_{\mathbb{Y}}$ is built for $\mathbb{Y}$ and the result of the rotation $\mathcal{R}_{\frac{\pi}{4}}$ using the definition and the theorem is depicted in Figure 4.11.



(a) Initial set $\mathbb{Y}$ (b) Rotated set.

Figure 4.11: Inversion of a separator by a rotation of angle $\frac{\pi}{4}$.

4.5 Specific contrators and separators

In the previous subsections, interval analysis tools have been presented such as contractors and separators. Coupled with a paver, they enable to compute an inner and an outer approximation of a set defined by constraints. In this subsection some contractors and separators will be introduced that deal with the problem of observation. Indeed an exteroceptive sensor provides information that can be described by an equation, an inequality, a shape, an image, etc... Contractors and separators can describe these sets by means of constraints. Some operators on these sets have to be handled before any processing. It will be assumed a 2D environment.

4.5.1 Separators on images

In the context of imagery, such as the underwater sonar imagery, the object manipulated is a color or grayscale image for example and not a set. To create sets, it is necessary to extract information from the image and create a binary image of the specific object. Generally some image processing techniques are used to binarize the images, that were discussed in Chapter 2 in sonar imagery. It will be assumed that a technique provides a binary image that represents an unstructured dataset $\mathbb{X}$. This binary image is an occupancy grid where the pixels at 1 correspond to the set $\mathbb{X}$ and the pixels at 0 to the complementary. In Chapter 2, a detector based on the integral image was introduced. This integral image will be used again to propose an image contractor.

4.5.1.1 Image contractor

The image contractor was proposed in [295] where it used the integral image introduced in the subsection 2.4.2 (Chapter 2).

Based on the computation of the integral image, called I, and according to Equation 2.15, it is quite easy to compute the number of 1-valued pixels of the binary image in any rectangle. Denote as ϕ the function that gives the number of occupied cells in a given box $[\mathbf{x}] = [x_1^-, x_1^+] \times [x_2^-, x_2^+]$ of $\mathbb{IR}^2$ overlaid on the image (grid). The four corners of this box are $A = (x_1^-, x_2^-)$, $B = (x_1^+, x_2^-)$, $C = (x_1^-, x_2^+)$ and $D = (x_1^+, x_2^+)$ according to the convention in Figure 2.18. The function ϕ gives then according to Equation 2.15:

$$\begin{array}{rcl} \mathbb{IR}^2 & \rightarrow & \mathbb{N} \\ \phi([\mathbf{x}]) & \mapsto & I(A) + I(C) - I(B) - I(D) \end{array} \tag{4.82}$$

From an initial box $[\mathbf{x}_0]$, the image contractor consists in finding the smallest box $[\mathbf{x}]$ included in $[\mathbf{x}_0]$ which contains exactly the same number of 1-pixels ($\phi([\mathbf{x}_0]) = \phi([\mathbf{x}])$).

World coordinates have to be taken into account to build a contractor on a georeferenced image. If $\mathbf{p}_0$ are the coordinate of the top-left corner and $\epsilon = (\epsilon_x, \epsilon_y)$ the horizontal and vertical pixel sizes, the change between the image coordinate $[\mathbf{x}]$ to the workspace coordinates $[\mathbf{x}_{world}]$ is:

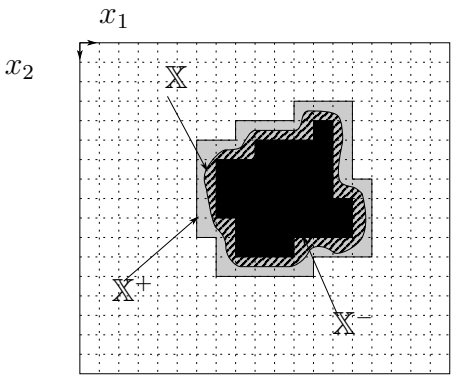
$$[\mathbf{x}_{world}] = \mathbf{p}_0 + [\mathbf{x}] \cdot (\{\mathbf{0}\} \cup \{\epsilon\}) \tag{4.83}$$

Due to the image convention where the vertical axis is oriented downward, the value of ϵ_y is negative and $(\{\mathbf{0}\} \cup \{\epsilon\})$ is equal to $[0, \epsilon_x] \times [\epsilon_y, 0]$.

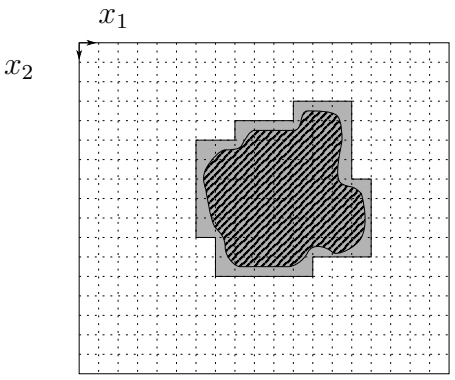
When a set $\mathbb{X}$ is represented by a binary image, this latter has to be enlarged by one pixel to describe an over approximation of the set $\mathbb{X}$ to keep the consistency of the contractor. It will then be minimal according to the image and to the set.

4.5.1.2 Image separator

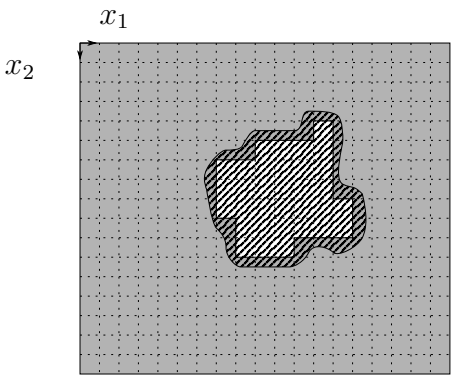
A separator, as previously introduced, can be built by using two complementary contractors. Considering two images representing an upper and lower approximations of a set $\mathbb{X}$, a separator can be built with contractors on each approximated images [71].



(a) Black pixels are inside the set $\mathbb{X}$ (lower approximation $\mathbb{X}^-$), the white ones outside and the gray ones are on the border.



(b) Image used by the outer contractor ($\mathcal{C}_{\mathbb{X}} \sim \mathcal{S}^{out}_{\mathbb{X}}$). Gray pixels are at 1 and white ones are null.

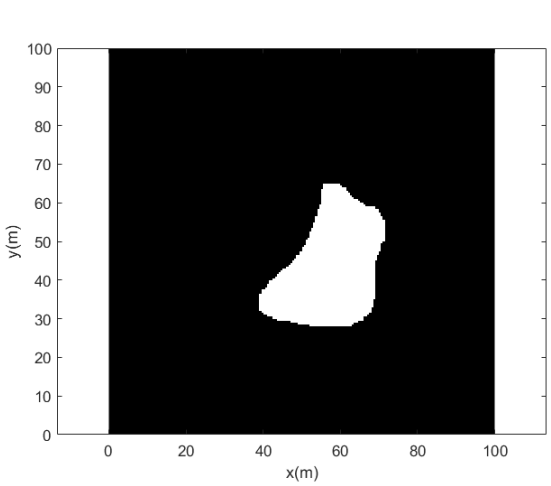


(c) Image used by the inner contractor ($\mathcal{C}_{\overline{\mathbb{X}}} \sim \mathcal{S}^{in}_{\mathbb{X}}$). Gray pixels are at 1 and white ones are null.

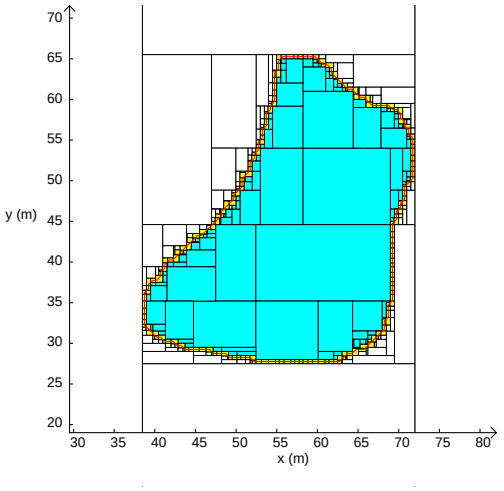
Figure 4.12: Image separator for the set $\mathbb{X}$ depicted by the black dashed area.

Example

A binary image is randomly drawn by hand in Figure 4.13(a). The paving for this separator associated to this shape is represented in Figure 4.13(b) where the contour of the shape is depicted by the red line.



(a) Random shape. White pixels represent the lower approximation.



(b) Separator for the associated image. White boxes are outside, cyan pixels are inside and yellow ones no conclusion can be made. The contour of the shape is represented in red.

Figure 4.13: Image separator. The scale is not the same in the two images.

4.5.2 Polar separator

Many sensors in robotics applications provide distance (range) and goniometric (bearing) measurements such as the Forward Looking Sonar (FLS). It is then important to handle correctly the change between the polar coordinates system to the Cartesian ones when bounded distance and bearing are provided. The polar equations refer generally to the system where the continuity on θ has to be handled:

$$x = \rho \cos\theta \tag{4.84}$$

$$y = \rho \sin\theta \tag{4.85}$$

$$\rho = \sqrt{x^2 + y^2} \tag{4.86}$$

$$\theta = \text{atan}(\frac{y}{x}) = atan2(y, x) \tag{4.87}$$

The conversion from polar coordinates to Cartesian is defined as the polar set:

$$\mathbb{X}_{polar} = \{(x, y) \in \mathbb{R}^2 | (x, y) = \mathbf{f}_{polar}(\rho, \theta)\} \tag{4.88}$$

On the domains $(x, y) \in \mathbb{R}^+ \times \mathbb{R}^+$ and $(\rho, \theta) \in \mathbb{R}^+ \times [0, \frac{\pi}{4}]$ the polar function $\mathbf{f}_{polar}$ is:

$$\mathbf{f}_{polar} \begin{pmatrix} \rho \\ \theta \end{pmatrix} = \begin{pmatrix} \rho \cos\theta \\ \rho \sin\theta \end{pmatrix} \tag{4.89}$$

The inverse of the polar function is:

$$\begin{pmatrix} \rho \\ \theta \end{pmatrix} = \mathbf{f}^{-1}_{polar} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \sqrt{x^2 + y^2} \\ \text{atan}(\frac{y}{x}) \end{pmatrix} \tag{4.90}$$

By using the symmetries and composing transformation functions, the initial domains for $(x, y) \in \mathbb{R}^+ \times \mathbb{R}^+$ and for $(\rho, \theta) \in \mathbb{R}^+ \times [0, \frac{\pi}{4}]$ can be extended to $\mathbb{R}^2$ and $\mathbb{R}^2$ according to [27][289][71].

Given a bounded range $[\rho]$ and bearing $[\theta]$ measurement, the polar set can be rewritten as:

$$\begin{aligned}\mathbb{X}_{polar} &= \{(x,y) \in \mathbb{R}^2 | \exists \rho \in [\rho], \exists \theta \in [\theta], \sqrt{x^2+y^2} = \rho \quad \text{and} \quad \text{atan2}(y,x) = \theta\} & (4.91)\\ &= \{(x,y) \in \mathbb{R}^2 | \exists \rho \in [\rho], \exists \theta \in [\theta], (x,y) = \mathbf{f}_{polar}(\rho,\theta)\} & (4.92)\\ &= \{(x,y) \in \mathbb{R}^2 | (x,y) = \mathbf{f}_{polar}([\rho],[\theta])\} & (4.93)\end{aligned}$$

A minimal polar separator has been proposed for Equation 4.93 in [72] using the concept of projections. This function is called *SepPolarXY* in *pyIbex*. Given a measurement $[\mathbf{y}] = [\rho] \times [\theta]$, the minimal separator for the polar set will be denoted as: $\mathcal{S}^{[\mathbf{y}]}_{\mathbf{f}^{-1}_{polar}}$.

A separator for the polar constraint can be built considering two manners:

- Using a classical forward-backward propagation in Equation 4.91.
- Using the minimal polar separator $\mathcal{S}^{[\mathbf{y}]}_{\mathbf{f}^{-1}_{polar}}$ that has been defined.

Based on the measurement $[\mathbf{y}] = [\rho] \times [\theta]$ (range and bearing measurements), the separator for the polar constraint using one of the two approaches will be denoted $\mathcal{S}^{[\mathbf{y}]}_{pol}$ in the following of this thesis.

Example 1

Consider a robot located at $\mathbf{x} = (x_1, x_2)$ and a punctual landmark at a known position $\mathbf{m} = (m_1, m_2) = (5, 6)$. The robot measures the distance d and the direction θ in the local reference frame with a bounded error, leading to two intervals $[d]$ and $[\theta]$ respectively. In this example, $[\mathbf{y}] = [d] \times [\theta] = [3, 4] \times [\frac{\pi}{4}, \frac{\pi}{3}]$. The set $\mathbb{X}$ of feasible robot location is defined as follows:

$$\begin{aligned}\mathbb{X} &= \{\mathbf{x} \in \mathbb{R}^2 | \exists d \in [d], \exists \theta \in [\theta], \mathbf{x} + \begin{pmatrix} d\cos\theta \\ d\sin\theta \end{pmatrix} = \mathbf{m}\} & (4.94)\\ &= \{\mathbf{x} \in \mathbb{R}^2 | \exists d \in [d], \exists \theta \in [\theta], \mathbf{m} - \mathbf{x} = \begin{pmatrix} d\cos\theta \\ d\sin\theta \end{pmatrix}\} & (4.95)\end{aligned}$$

It gives for the set $\mathbb{X}$:

$$\begin{aligned}\mathbb{X} &= \{\mathbf{x} \in \mathbb{R}^2 | \exists \mathbf{y} \in [\mathbf{y}], \mathbf{m} - \mathbf{x} = \mathbf{f}_{polar}(\mathbf{y})\} & (4.96)\\ &= \{\mathbf{x} \in \mathbb{R}^2 | \mathbf{m} - \mathbf{x} = \mathbf{f}_{polar}([\mathbf{y}])\} & (4.97)\\ &= \mathbf{g}_m \circ \mathbf{f}_{polar}([\mathbf{y}]) & (4.98)\end{aligned}$$

where the function $\mathbf{f}_{polar}$ has been defined in Equation 4.89 and the function $\mathbf{g_m}$ is:

$$\mathbf{g_m}(\mathbf{x}) = \mathbf{m} - \mathbf{x} \tag{4.99}$$

This latter function corresponding to a translation is box-conservative.

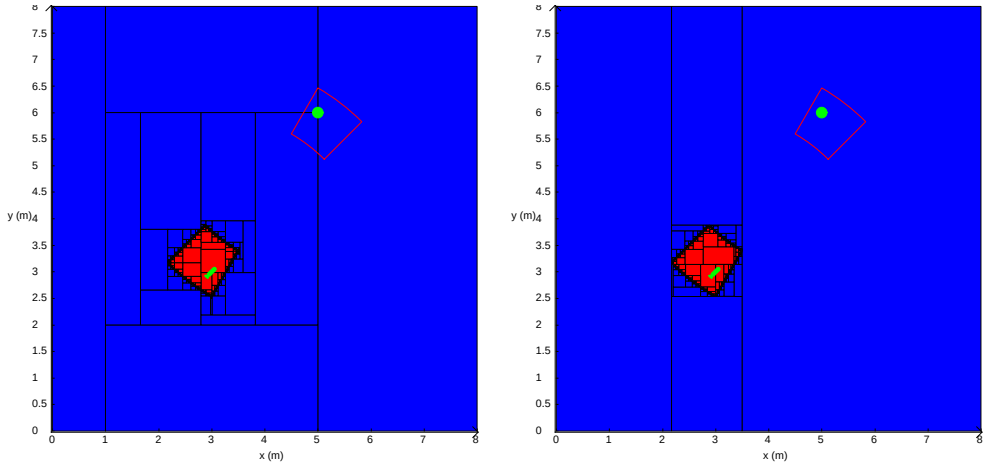
Firstly, a separator for the polar constraint has to be built according one of the two manners.

Secondly, the final separator consistent with $\mathbb{X}$ is given by:

$$\mathcal{S}_{\mathbb{X}} = \mathbf{g_m} \circ \mathcal{S}^{[\mathbf{y}]}_{pol} \tag{4.100}$$

Finally by using the concept of inversion of separators and due to the fact that $\mathbf{g_m}$ is bijective, the set $\mathbb{X}$ can be obtained.

The results from the two methods are depicted in Figure 4.14 where the set $\mathbb{X}$ using the classical forward-backward separator is illustrated in Figure 4.14(a) and using the minimal separator is proposed in 4.14(b). The difference can be seen in the manner it contracts on the set $\mathbb{X}$. More bisections are needed for the classical forward-backward separator.



(a) Classical forward-backward separator.

(b) Minimal polar separator.

Figure 4.14: Approximation of $\mathbb{X}$ where the initial domain is $\mathbb{X}_0 = [0, 8]^2$. The landmark position is drawn by a green disk, one position of the AUV is drawn by a green AUV with the associated distance and bearing measurements depicted by a red pie.

Example 2

Come back again on the problem of the range only localization but consider now that bearing is also measured. Assume that bearing is measured with a bounded uncertainty of 5°, meaning that the true bearing remains in an interval of ±5° around this angle. Table 4.2 sums up the different positions and measurements.

	$\mathbf{m}_i$	$d_i^*(m)$	$[d_i]$ (m)	θ_i^* (deg)	$[\theta_i](deg)$
M_1	$(15, 4)$	10.05	$[9.75, 10.35]$	-5.7	$[-10.7, -0.7]$
M_2	$(0, 6)$	5.1	$[4.8, 5.4]$	168.7	$[163.7, 173.7]$
M_3	$(8, 9)$	5	$[4.7, 5.3]$	53.1	$[48.1, 58.1]$

Table 4.2: Landmark's positions and measurements.

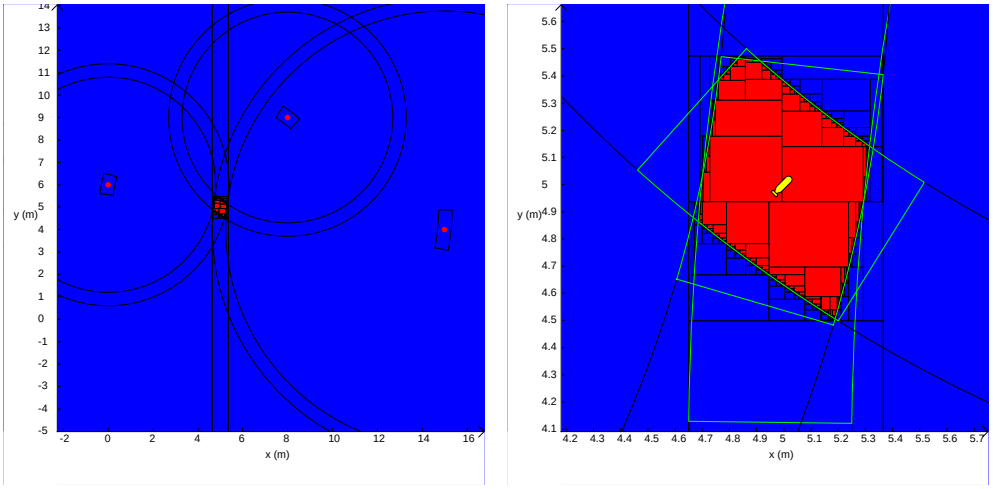
The feasible robot position $\mathbb{X}$ is given by:

$$\mathbb{X} = \bigcap_i \mathbf{g}_{\mathbf{m}_i} \circ \mathbf{f}_{polar}([\mathbf{y}_i]) \tag{4.101}$$

Using the polar separator $\mathcal{S}_{pol}^{[\mathbf{y}_i]}$ introduced where $[\mathbf{y}_i] = [d_i] \times [\theta_i]$, the associated separator $\mathcal{S}_{\mathbb{X}}$ is then:

$$\mathcal{S}_{\mathbb{X}} = \bigcap_i \mathbf{g}_{\mathbf{m}_i} \circ \mathcal{S}_{pol}^{[\mathbf{y}_i]} \tag{4.102}$$

The result is given in Figure 4.15 where black pies are projected on the landmark position based on the uncertain measurements according to the true robot location in Figure 4.15(a). A zoom is provided in Figure4.15(b) where the green pies represent the projection of the uncertain measurements on the robot location according to the landmark position.



(a) Approximation of the set with black pies representing the uncertain measurements projected on the landmarks according to the true location of the robot.

(b) Zoom of the solution with green pies representing the projection on the robot location, represented by the yellow AUV, of the uncertain measurements according to the landmark's positions.

Figure 4.15: Approximation of the $\mathbb{X}$.

Example 3

Another example of the range-bearing localization with values given in Table 4.3 considering three landmarks or beacons.

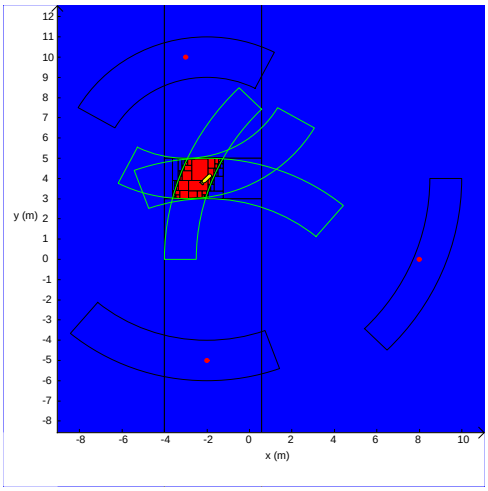
	$\mathbf{m}_i$	$[d_i]$ (m)	$[\theta_i](deg)$
M_1	$(8, 0)$	$[10.5, 12]$	$[-45, 0]$
M_2	$(-2, -5)$	$[8, 10]$	$[-130, -70]$
M_3	$(-3, 10)$	$[5, 7]$	$[63, 150]$

Table 4.3: Landmark's positions and measurements.

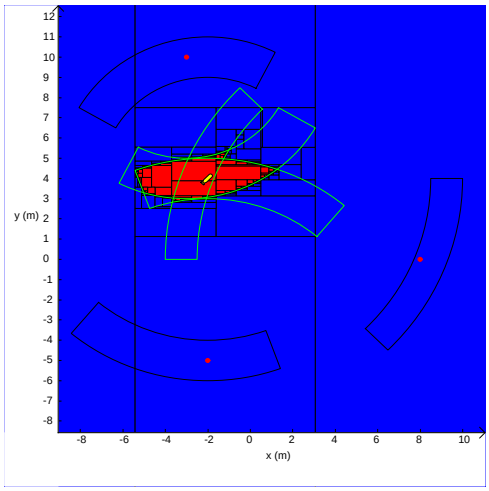
Consider now that outliers are present among the measurements (false detection for example). The solution of the set $\mathbb{X}$ is now given by the q-relaxed intersection [145][41]. The associated separator is then:

$$\mathcal{S}_{\mathbb{X}} = \bigcap_{i\in\{1,2,3\}}^{\{q\}} \mathbf{g}_{\mathbf{m}_i} \circ \mathcal{S}_{pol}^{[\mathbf{y}_i]} \tag{4.103}$$

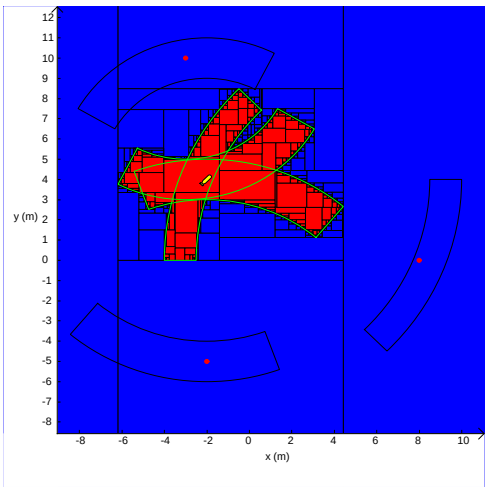
The *0*-relaxed intersection is the classical intersection where the result is depicted in Figure 4.16(a). The *1*-relaxed intersection corresponds to the set considering one false measurement shown in Figure 4.16(b). It is given by the union of each pair of intersection (considering two measurements each time). Finally the *2*-relaxed intersection considers that two outliers are among the measurements and then the robot's position could be anywhere in the union of the corresponding set for each landmark as shown in Figure 4.16(c).



(a) *0*-relaxed intersection.



(b) *1*-relaxed intersection.



(c) *2*-relaxed intersection.

Figure 4.16: Approximation of the $\mathbb{X}$ considering the q-relaxed intersection. Green pies depict the projection of the range and bearing measurements according to the landmark's positions onto the robot location.

4.5.3 Projection of separators

The notion of projection associated to separators [74] is now introduced as it will be the main operator to solve some problems in this thesis.

Definition 1

Given two sets $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{Y} \subset \mathbb{R}^p$. Considering the set $\mathbb{Z} = \mathbb{X} \times \mathbb{Y}$, the projection of a subset $\mathbb{Z}_1$ of $\mathbb{Z}$ onto $\mathbb{X}$ (with respect to $\mathbb{Y}$) is defined as follows:

$$proj_{\mathbb{X}}(\mathbb{Z}_1) = \{\mathbf{x} \in \mathbb{X} | \exists \mathbf{y} \in \mathbb{Y}, (\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_1\} \tag{4.104}$$

Figure 4.17 depicts the projection in red of the $\mathbb{Z}_1 \subset \mathbb{Z}$ drawn in blue onto the set $\mathbb{X}$ (here the bottom plane).

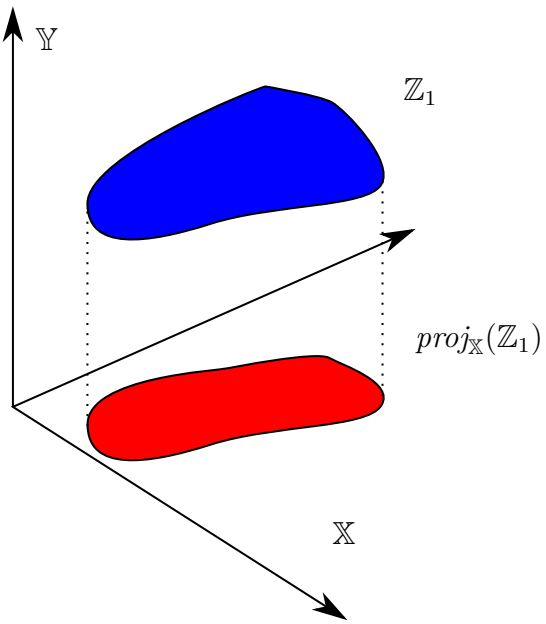


Figure 4.17

If a separator is available for $\mathbb{Z}_1$, a separator consistent with $proj_{\mathbb{X}}(\mathbb{Z}_1)$ can be built.

Definition 2

Considering the box $[\mathbf{x}] \times [\mathbf{y}]$ and a contractor $\mathcal{C}$. The contractor $\mathcal{C}([\mathbf{x}], [\mathbf{y}])$ describes the contraction on this box. The partial contractor with respect to $\mathbf{x}$ is defined as the projection of the box after the contraction $\mathcal{C}([\mathbf{x}], [\mathbf{y}])$ onto $\mathbf{x}$:

$$\partial_{\mathbf{x}}\mathcal{C}([\mathbf{x}], [\mathbf{y}]) = [\mathbf{a}] \quad \textbf{and} \quad \partial_{\mathbf{y}}\mathcal{C}([\mathbf{x}], [\mathbf{y}]) = [\mathbf{b}] \tag{4.105}$$

where $([\mathbf{a}], [\mathbf{b}]) = \mathcal{C}([\mathbf{x}], [\mathbf{y}])$ the contraction of the box $[\mathbf{x}] \times [\mathbf{y}]$ according to $\mathcal{C}$.

Definition 3

Now, consider the separator $\mathcal{S}([\mathbf{x}], [\mathbf{y}]) = \{\mathcal{S}^{in}([\mathbf{x}], [\mathbf{y}]), \mathcal{S}^{out}([\mathbf{x}], [\mathbf{y}])\}$ on the box $[\mathbf{x}] \times [\mathbf{y}]$. The projection of $\mathcal{S}$ is defined as follows:

$$proj_{\mathbf{x}}(\mathcal{S})([\mathbf{x}]) = \left\{ \bigcap_{\mathbf{y} \in [\mathbf{y}]} \partial_{\mathbf{x}}\mathcal{S}^{in}([\mathbf{x}], [\mathbf{y}]), \bigcup_{\mathbf{y} \in [\mathbf{y}]} \partial_{\mathbf{x}}\mathcal{S}^{out}([\mathbf{x}], [\mathbf{y}]) \right\} \tag{4.106}$$

where $\partial_{\mathbf{x}}\mathcal{S}^{in}$ and $\partial_{\mathbf{x}}\mathcal{S}^{out}$ denote the partial contractors for the inner and the outer contractor respectively, introduced in **Definition 2**.

Theorem 4.5.1 *If a separator $\mathcal{S}$ is consistent with the set $\mathbb{Z}$ then:*

$$proj_{\mathbf{x}}(\mathbb{Z}) \sim proj_{\mathbf{x}}(\mathcal{S}) \tag{4.107}$$

Proof: It is sufficient to prove that the outer contractor is consistent with $proj_{\mathbf{x}}(\mathbb{Z})$ and the inner contractor is consistent with the complementary $\overline{proj_{\mathbf{x}}(\mathbb{Z})}$.

Firstly:

$$
\begin{aligned}
proj_{\mathbf{x}}(\mathbb{Z}) &= \{\mathbf{x}|\exists \mathbf{y} \in [\mathbf{y}], (\mathbf{x},\mathbf{y}) \in \mathbb{Z}\} && (4.108)\\
&= \bigcup_{\mathbf{y}\in[\mathbf{y}]} \{\mathbf{x}, (\mathbf{x},\mathbf{y}) \in \mathbb{Z}\} && (4.109)\\
&\sim \bigcup_{\mathbf{y}\in[\mathbf{y}]} \partial_{\mathbf{x}}\mathcal{S}^{out} && (4.110)
\end{aligned}
$$

Secondly, since $\mathbb{Z} \sim \mathcal{S}^{in}$:

$$
\begin{aligned}
\overline{proj_{\mathbf{x}}(\mathbb{Z})} &= \overline{\{\mathbf{x}|\exists \mathbf{y} \in [\mathbf{y}], (\mathbf{x},\mathbf{y}) \in \mathbb{Z}\}} && (4.111)\\
&= \{\mathbf{x}|\forall \mathbf{y} \in [\mathbf{y}], (\mathbf{x},\mathbf{y}) \notin \mathbb{Z}\} && (4.112)\\
&= \{\mathbf{x}|\forall \mathbf{y} \in [\mathbf{y}], (\mathbf{x},\mathbf{y}) \in \overline{\mathbb{Z}}\} && (4.113)\\
&= \bigcap_{\mathbf{y}\in[\mathbf{y}]} \{\mathbf{x}, (\mathbf{x},\mathbf{y}) \in \overline{\mathbb{Z}}\} && (4.114)\\
&\sim \bigcap_{\mathbf{y}\in[\mathbf{y}]} \partial_{\mathbf{x}}\mathcal{S}^{in} && (4.115)
\end{aligned}
$$

In *pyIbex* library, the implementation of the projection of separators is based on *Proj-Union* and *Proj-intersection* described in [47]. It consists in splitting the initial domain $[\mathbf{y}]$ into a list of small intervals $[\mathbf{y}_i]$. The outer contraction is realized on each sub-subdomain $[\mathbf{x}] \times [\mathbf{y}_i]$ and the hull of the results is returned. Points $\mathbf{y}_i$ are sampled in $[\mathbf{y}_i]$ and the inner contraction is applied on each $[\mathbf{x}] \times \{\mathbf{y}_i\}$. The intersection is then returned. A parameter ϵ has to be defined as the size of the split. The projection assumes that the set $\mathbb{Z}$ does not have an empty volume to define an inner contractor.

Example

Consider the sets $\mathbb{A} \subset \mathbb{R}$, $\mathbb{B} \subset \mathbb{R}$, $\mathbb{T} \subset \mathbb{R}$ and define the set $\mathbb{X} = \mathbb{A} \times \mathbb{B}$. The set $\mathbb{Z} \subset \mathbb{X}$ is defined as follows:

$$\mathbb{Z} = \{(a,b) \in \mathbb{A} \times \mathbb{B} | \exists t \in \mathbb{T}, \quad at^2 + bt + 1 \leq 0\} \tag{4.116}$$

This set $\mathbb{Z}$ corresponds to the values $(a,b) \in \mathbb{A} \times \mathbb{B}$ such that the equation $at^2 + bt + 1$ is lower than 0 at a time t. This latter may be different depending on the box $[a] \times [b]$. This set corresponds to the projection of the set $\mathbb{X} \times \mathbb{T}$ onto $\mathbb{X}$ with respect to the time $\mathbb{T}$. A separator $\mathcal{S}$ is firstly built according to the constraint $at^2 + bt + 1 \leq 0$ expressed in Equation 4.116. Three parameters define the constraint. A separator $\mathcal{S}_{\mathbb{Z}} \sim proj_{\mathbb{X}}(\mathcal{S})$ can then be built for the projection according to Theorem 4.5.1.

With $\mathbb{A} = [-5,5]$, $\mathbb{B} = [-5,5]$ and $\mathbb{T} = [0,20]$, the paving obtained for the set $\mathbb{Z} \subset \mathbb{A} \times \mathbb{B}$ is depicted in Figure 4.18.

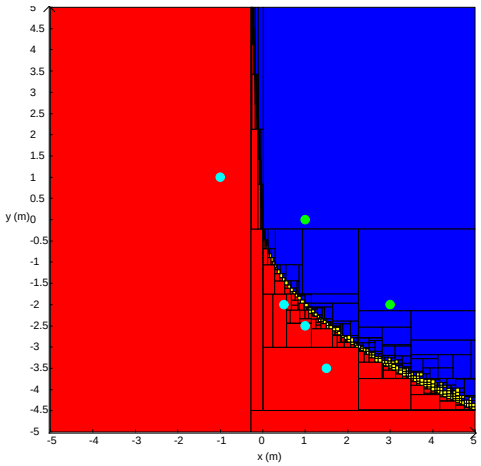
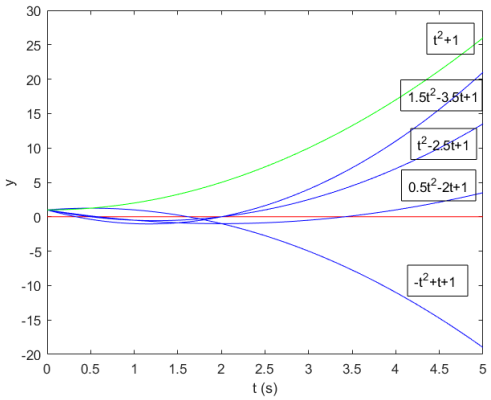
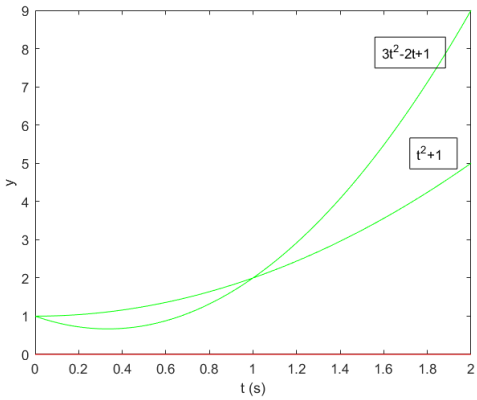


Figure 4.18: Inner and outer approximation of the set $\mathbb{Z}$. Blue boxes are outside the set, red ones are inside and yellow ones no conclusion can be made.

The temporal curves according to the equation $y = at^2 + bt + 1$ are depicted in Figure 4.19 for some points (a, b) shown in the result in Figure 4.18 for the set $\mathbb{Z}$. Cyan points in Figure 4.18 are temporally represented in blue in Figure 4.19(a). It is proved that at a certain time t the value $y = at^2 + bt + 1$ is lower than 0. If a is below than 0, the inequality expressed in Equation 4.116 is always true for $t \geq 0$. The green dots depicted in Figure 4.18 do not cross the x-axis, it is always upper than 0 for $t \geq 0$ as represented in Figure 4.19(b).



(a) Blue curves cross the x-axis at a certain time contrary to the green curve that is always upper than 0.



(b) Zoom for the two green curves associated to the two green dots.

Figure 4.19: Some temporal curves according to points depicted in Figure 4.18. Cyan dots correspond to blue curves and green dots to green curves. The red curve depicts the x-axis.

4.6 Conclusion

In this Chapter, the interval analysis theory has been introduced with some operators on sets such as unions, intersections, projections, inversions, etc... When dealing with sets defined by constraints, contractors have been developed to characterize more efficiently and more rapidly the sets based on a forward-backward propagation algorithm. However this propagation can only provide an over approximation of the sets. Coupled with a *paver*, a contractor overcomes this issue by providing a better approximation of the sets, called an outer approximation, with

only one parameter to set up: ϵ. However, when dealing with sets where the volume is not empty, the paver with the contractor only provides an outer approximation of the sets leading to many bisections for boxes that are actually inside the sets. An inner approximation is not available. By considering the complementary set, a separator overcomes this problem by providing at the same time the inner and the outer approximations of the sets coupled with a *paver* again. Set-membership methods have proven to be well-suited to solve engineering problems when equations are non-linear and non-convex due to their efficiency to handle and propagate the uncertainties, and to manipulate sets that are defined by constraints. However, the performance of paving methods rely on the accuracy of the contractors, meaning the minimality of the contractors. Some operators such the image separator and the polar separator have been introduced that can deal with images and measurements that are frequently provided by sonar sensors such as the Forward Looking Sonar (FLS). The notion of projections of separators shows many interests to deal with many variables and will be at the heart of the resolution of many problems in the next Chapters. In many robotics applications, the notion of registration is important to relocate the robot and to reduce its uncertainty. The registration will be the topic of the next Chapter in this set-membership context.

Chapter 5

Registration maps

5.1 Introduction

In Chapter 2, many detectors are presented to extract landmarks from an underwater environment using sonars. Indeed acoustics is able to provide information in dark environment compared to cameras that require some lights. Many landmarks may be detected in this challenging environment such as rocks, man-made objects, textures of seabed, etc... All these landmarks can be detected based on the parameters of the sonar and the characteristics of the vehicle that is carrying the sensors. These landmarks are usually well geolocalized thanks to a reliable navigation of the platform, i.e. surface vehicle or Unmanned Underwater Vehicle (UUV), and performant exteroceptive sensors such as sidescan sonars or Synthetic Aperture Sonar (SAS). From these landmarks, when a revisit mission is required, a registration map can be computed based on the exteroceptive abilities of the revisit vehicle. Indeed, in the context of Mine Counter Measure (MCM) missions, revisiting a previously suspicious mapped object is important to identify and/or neutralize it. This registration map indicates locations where the robot could reduce its uncertainty by detecting again the landmarks. The concept of registration maps is not new. In Chapter 3, the Sensory Uncertainty Field (SUF) [304] was introduced that indicates at different poses of the robot in a geometrical *a priori* known world the ability to relocate at these poses with laser measurements. Based on the Field Of View (FOV) of the sensor, a localization space is computed according to the placement of landmark.

In the previous Chapter, the set-membership tools have been introduced where sets are defined by constraints. Separators, coupled with a *paver*, enable to provide an inner and an outer approximation of a set when its volume is not empty. These tools are particularly well-suited when dealing with observations such as shapes from images or range-bearing measurements.

In this Chapter, it will be assumed a patch exploration revisit sensor with limited abilities. It means that the sensor is able to detect in a restricted FOV between some ranges and/or aperture angle. These sensors refer to cameras or Forward Looking Sonar (FLS). This restricted visibility of the sensor can be defined by constraints. It will be assumed that the map $\mathbb{M}$ is composed of any shaped landmarks and the visibility of the sensor (measurements) are subsets of $\mathbb{R}^q$ where q is the dimension of the map (2 or 3 generally). For simplicity reasons, only the 2D case will be developed but the method could obviously be extended to 3D. The map and the visibility of the sensor are called shapes.

5.2 Registration map

The concept of registration map will be firstly presented under the assumption that the sensor measurements and the map are perfect. No uncertainty is present.

5.2.1 Visibility area

As mentionned in the introduction, it will be assumed in this Chapter an exteroceptive sensor with limited sensing abilities. Dealing with ranging and imaging sensors, known as patch exploration [73], it will be assumed that the robot can observe the environment between a minimal (could be 0) and a maximal range, and potentially with a limited aperture angle. These refer to a limited FOV of the sensor. These limited sensing abilities can be defined as constraints as introduced in the previous Chapter. These sensors may refer to cameras, FLS or 3D FLS for example. A sensor that can sense with a 360° aperture angle is said to detect isotropically. Assuming a 2D environment for ease of proving the concept, the visibility area $\mathbb{V} \subset \mathbb{R}^2$ corresponds to the area on the seabed that is visible by the sensor depending on the robot pose $\mathbf{x}$. The visibility area is also called the sensor field of view in [260].

The problem of the visibility area aims at finding the set $\mathbb{V}(\mathbf{x})$ such that:

$$\mathbb{V}(\mathbf{x}) = \{\mathbf{z} \in \mathbb{R}^2 | \mathbf{f}(\mathbf{x}, \mathbf{z}) \in [\mathbf{y}]\} \tag{5.1}$$

where $\mathbf{f}$ denotes the range and bearing equations according to the pose of the robot $\mathbf{x}$ and $[\mathbf{y}]$ the intervals of sensing abilities. For example, $[\mathbf{y}] = [0, 10] \times [0, 360](m \times deg)$ corresponds to a sensor that can detect up to $10m$ with a 360° aperture angle (isotropically). The visibility area $\mathbb{V}(\mathbf{x})$ is a subset of $\mathbb{R}^2$.

The two different sensors will be developed in the following subsections.

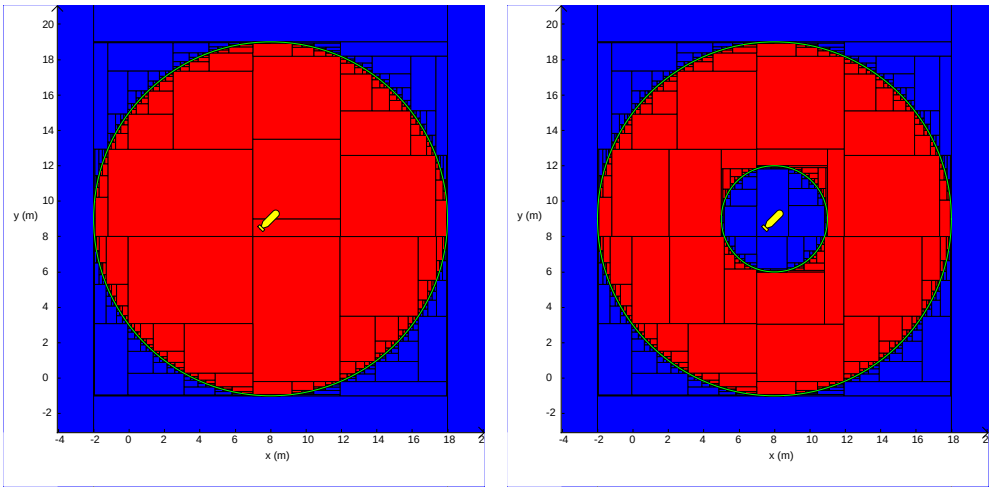
5.2.1.1 Limited range with 360° aperture angle

The sensor can detect between a minimal R_{min} and a maximal R_{max} range. This sensor is then able to detect in an interval $[R_{min}, R_{max}]$. As it can detect isotropically, this will be the single constraint on the visibility area which is defined as follows:

$$\mathbb{V}(\mathbf{x}) = \{\mathbf{z} \in \mathbb{R}^2 | \sqrt{(z_1 - x_1)^2 + (z_2 - x_2)^2} \in [R_{min}, R_{max}]\} \tag{5.2}$$

Due to the isotropic ability of sensing, the robot pose $\mathbf{x}$ can only be described by its location $(x_1, x_2) = (x, y)$ where x and y are the coordinates of the robot. It is not dependant on the heading θ. This constraint can be defined as presented by an associated separator $\mathcal{S}_{\mathbb{X}}$.

This set corresponds finally to a ring as depicted in Figure 5.1(b) or a disk, as shown in Figure 5.1(a), if R_{min} equals to 0 around the robot position $\mathbf{x} = (8, 9)$.



(a) Range of detection in the interval $[0, 10]m$.

(b) Range of detection in the interval $[3, 10]m$.

Figure 5.1: Visibility area for an isotropic sensor with limited ranges. The position of the robot is represented by the yellow AUV. The heading is not important. The green line shows the contour.

5.2.1.2 Limited range and aperture angle

Generally, cameras or FLS can detect between a minimal R_{min} and a maximal R_{max} range and have a limited aperture angle. This is defined as an additional constraint to the previous range-only sensor. The robot can then detect in a cone with an opening angle $2\theta_3$, referring to the aperture angle of a sonar. The heading of the robot has to be considered now. Depending on how the sensor is mounted on the AUV, it may be side-looking or forward-looking, the visibility area may be different. Assuming for example a forward looking sensor where the main direction of sensing coincides with the heading of the robot, the visibility area is defined as follows:

$$\begin{aligned} \mathbb{V}(\mathbf{x}) &= \{\mathbf{z} \in \mathbb{R}^2 | \sqrt{(z_1 - x_1)^2 + (z_2 - x_2)^2} \in [R_{min}, R_{max}] \\ &\quad \textbf{and } \operatorname{atan2}(z_2 - x_2, z_1 - x_1) - x_3 \in [-\theta_3, \theta_3]\} \end{aligned} \tag{5.3}$$

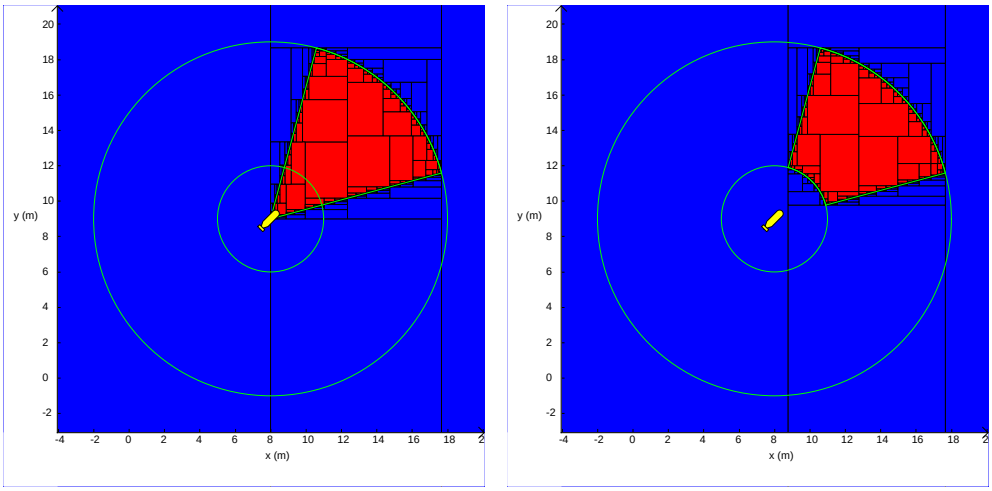
where $\mathbf{x} = (x_1, x_2, x_3) = (x, y, \theta)$ and θ_3 corresponds to the half aperture angle.

This set corresponds to a similar set expressed in Equation 4.95. It can be rewritten as:

$$\mathbb{V}(\mathbf{x}) = \{\mathbf{z} \in \mathbb{R}^2 | \exists r \in [R_{min}, R_{max}], \exists \beta \in [-\theta_3, \theta_3], \mathbf{z} - \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} r \cdot \cos(x_3 + \beta) \\ r \cdot \sin(x_3 + \beta) \end{pmatrix}\} \tag{5.4}$$

It corresponds to the composition of a translation $\mathbf{g}_{\mathbf{x}}(\mathbf{z}) = \mathbf{z} - \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and a polar constraint as presented in the previous Chapter. Using the separator $\mathcal{S}_{pol}^{[\mathbf{y}]}$ for the polar constraint, a separator can be obtained for the set $\mathbb{V}(\mathbf{x})$ based on the same concept as in Equation 4.100.

This set corresponds finally to a cone with a range between 0 and R_{max} as depicted in Figure 5.2(a) for $\theta = 45°$ the heading of the vehicle located at $(8, 9)$. It represents a pie, i.e. a cone with cutting ranges, with a limited distance of detection between 3 and $10m$ in Figure 5.2(b).



(a) With a limited maximal range at $10m$ and a whole aperture angle of 60°.

(b) With a limited range in $[3, 10]m$ and a whole aperture angle of 60°.

Figure 5.2: Visiblity area for a sensor with limited FOV in range and aperture angle. The pose of the robot $\mathbf{x} = (8, 9, 45°)$ is represented by the yellow AUV.

5.2.2 Definition of the problem

Given a visibility area $\mathbb{A} \subset \mathbb{R}^2$ of a sensor, introduced in the previous section, and a landmark described by $\mathbb{B} \subset \mathbb{R}^2$ linked by a constraint $\mathcal{L}$ that depends on a parameter $\mathbf{p} \in \mathbb{R}^p$, the registration map problem is intended to find the set $\mathbb{P}$ of the parameters vector consistent with $\mathcal{L}$. This constraint is called the registration map.

The registration map problem aims at finding the set $\mathbb{P}$ that corresponds to the robot configurations able to detect a part or entirely a landmark $\mathbb{B}$ based on the visibility area $\mathbb{A}$ of the sensor. It consists then in finding the parameters vector $\mathbf{p} \in \mathbb{P}$ of a possibly non linear transformation $\mathbf{f}$ such that $\mathbf{f}(\mathbf{p}, \mathbb{A}) \cap \mathbb{B} \neq \emptyset$ as depicted in Figure 5.3. $\mathbf{p}_1$ belongs to the set $\mathbb{P}$ meaning that after transformation by $\mathbf{f}$, the visibility area $\mathbb{A}$ intersects $\mathbb{B}$ represented by the blue area. On the contrary, $\mathbf{p}_2$ does not belong to $\mathbb{P}$ which leads to an empty intersection between $\mathbb{B}$ and the transformed set $\mathbf{f}(\mathbf{p}_2, \mathbb{A})$. This defines the map registration constraint. The function $\mathbf{f}$ may correspond to a translation, a rotation or the composition of both for example. This constraint is justified by the fact that detecting an element of a landmark enables to relocate the robot with respect to this landmark. The set $\mathbb{B}$ may be reduced to a singleton which is called a punctual landmark.

This concept has similarities with the shape-registration problem where it is looking for the set $\mathbb{P}$ such that $\mathbf{f}(\mathbf{p}, \mathbb{A}) \subset \mathbb{B}$ presented in [71] and discussed in a next subsection.

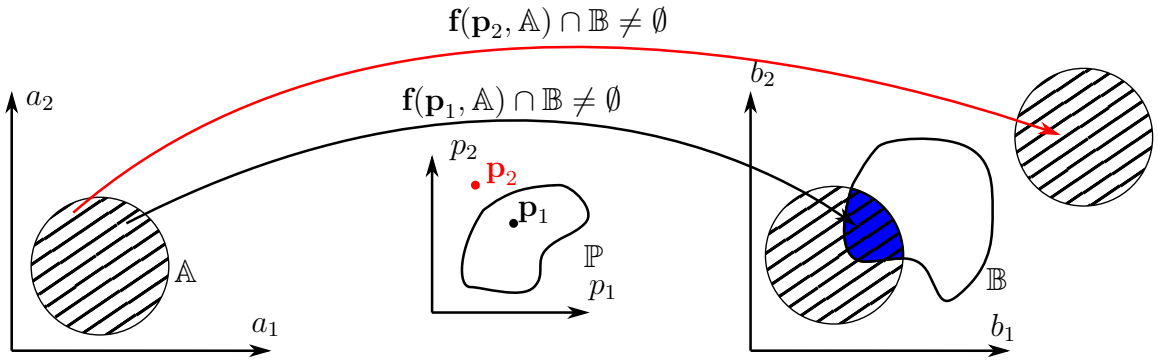


Figure 5.3: Example of the registration map concept associated to the constraint $\mathbf{f}(\mathbf{p}, \mathbb{A}) \cap \mathbb{B} \neq \emptyset$. The set $\mathbb{P}$ corresponds to the parameter of the function $\mathbf{f}$ such as the transformation of the shape $\mathbb{A}$ intersects $\mathbb{B}$. A parameter vector $\mathbf{p}_2$ outside the solution set does not enable the intersection between $\mathbb{B}$ and the set $\mathbb{A}$ after transformation.

The registration map problem is formulated as a set projection. Consider a function:

$$\mathbf{f} : \begin{cases} \mathbb{R}^2 \times \mathbb{R}^p & \rightarrow & \mathbb{R}^2 \\ (\mathbf{a}, \mathbf{p}) & \rightarrow & \mathbf{f}(\mathbf{a}, \mathbf{p}) \end{cases} \tag{5.5}$$

With $\mathbf{p} \in \mathbb{R}^P$, $\mathbb{A} \subset \mathbb{R}^2$, $\mathbb{B} \subset \mathbb{R}^2$ and $\mathbb{Z} \subset \mathbb{R}^2 \times \mathbb{R}^p$, the following notations are used:

$$\mathbf{f}(\mathbb{A}, \mathbf{p}) = \{\mathbf{b} | \exists \mathbf{a} \in \mathbb{A}, \mathbf{b} = \mathbf{f}(\mathbf{a}, \mathbf{p})\} \tag{5.6}$$
$$\mathbf{f}^{-1}(\mathbb{B}) = \{\mathbf{z} = (\mathbf{a}, \mathbf{p}) | \exists \mathbf{b} \in \mathbb{B}, \mathbf{b} = \mathbf{f}(\mathbf{a}, \mathbf{p})\} \tag{5.7}$$
$$proj_{\mathbf{p}}(\mathbb{Z}) = \{\mathbf{p} | \exists \mathbf{a}, (\mathbf{a}, \mathbf{p}) \in \mathbb{Z}\} \tag{5.8}$$

where the operator $proj()$ has been introduced in the previous Chapter.

By considering the set:

$$\mathbb{P} = \{\mathbf{p} \in \mathbb{R}^p | \mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} \neq \emptyset\} \tag{5.9}$$

The vector $\mathbf{p}$ is associated to a transformation $\mathbf{f}$. A transformation vector $\mathbf{p}$ is consistent if after the transformation of $\mathbb{A}$, it intersects the set $\mathbb{B}$. This leads to:

$$\mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} \neq \emptyset \tag{5.10}$$
$$\Leftrightarrow \quad \exists \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B} \tag{5.11}$$
$$\Leftrightarrow \quad \exists \mathbf{a} \in \mathbb{A}, (\mathbf{a}, \mathbf{p}) \in \mathbf{f}^{-1}(\mathbb{B}) \tag{5.12}$$
$$\Leftrightarrow \quad \exists \mathbf{a}, (\mathbf{a}, \mathbf{p}) \in \mathbb{R}^2 \times \mathbb{R}^p \wedge (\mathbf{a}, \mathbf{p}) \in \mathbf{f}^{-1}(\mathbb{B}) \tag{5.13}$$

According to the notation in Equation 5.8, it results then:

$$\mathbb{P} = proj_{\mathbf{p}}((\mathbb{A} \times \mathbb{R}^p) \cap \mathbf{f}^{-1}(\mathbb{B})) \tag{5.14}$$

If separators $\mathcal{S}_{\mathbb{A}}$ for $\mathbb{A}$ and $\mathcal{S}_{\mathbb{B}}$ for $\mathbb{B}$ are available, then a separator $\mathcal{S}_{\mathbb{P}}$ for $\mathbb{P}$ can be computed using the separator algebra presented. The separator $\mathcal{S}_{\mathbb{P}}$ consistent with $\mathbb{P}$ is then given by:

$$\mathcal{S}_{\mathbb{P}} = proj_{\mathbf{p}}((\mathcal{S}_{\mathbb{A}} \times \mathcal{S}_{\mathbb{R}^p}) \cap \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{B}})) \tag{5.15}$$

Coupled with a *paver*, this separator enables to provide an inner and an outer approximation of the set $\mathbb{P}$.

As defined, the registration map corresponds to the robot configurations able to detect the landmark considered $\mathbb{B}$ according to the visibility area $\mathbb{A}$. If the visibility area is computed

at the particular location $(0, 0, 0°)$ for the robot pose, then the set $\mathbb{P}$ will correspond to the robot configurations able to detect the landmark considering a function $\mathbf{f}$ that may be either a translation, or a rotation or a composition of both. The function $\mathbf{f}$ depends on the FOV of the sensor. For each of these transformations, the function $\mathbf{f}$ is bijective and the inverse function $\mathbf{f}^{-1}$ can be computed.

Considering the general case where the function $\mathbf{f}$ corresponds to the composition of a rotation and a translation, the parameter vector $\mathbf{p}$ is then $\mathbf{p} = (p_1, p_2, p_3) \in \mathbb{R}^3$ with (p_1, p_2) the parameters of the 2D translation and p_3 the angle of the rotation in the 2D plane. The final function can be written as follows:

$$\mathbf{f} : \begin{cases} \mathbb{R}^2 \times \mathbb{R}^3 & \rightarrow & \mathbb{R}^2 \\ (\mathbf{x}, \mathbf{p}) & \rightarrow & \mathcal{R}(p_3) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \end{cases} \tag{5.16}$$

where $\mathcal{R}$ corresponds to the 2D rotation of angle p_3:

$$\mathcal{R}(p_3) = \begin{pmatrix} \cos(p_3) & -\sin(p_3) \\ \sin(p_3) & \cos(p_3) \end{pmatrix} \tag{5.17}$$

This general case enables to handle the sensors with a limited aperture angle that depends on the way it's mounted in the vehicle. If the visibility area is a disk or a ring, i.e. only range limited sensor, the parameter p_3 does not need to be found. The set $\mathbb{P}$ may then be a subset of $\mathbb{R}^2$ or $\mathbb{R}^3$. In both case, an inner and an outer approximation of the set $\mathbb{P}$ can be computed according to Equation 5.15 if separators for the visibility area $\mathbb{A}$ and the landmark $\mathbb{B}$ are available.

Due to the bijection property of such transformations, the problem could be solved considering the inverse function $\mathbf{f}^{-1}$. Indeed, looking for the set $\mathbb{P}$ expressed in Equation 5.9, the problem may be rewritten directly as:

$$\mathcal{S}_{\mathbb{P}} = proj_{\mathbf{p}}(\mathbf{f}(\mathcal{S}_{\mathbb{A}}) \cap (\mathcal{S}_{\mathbb{B}} \times \mathcal{S}_{\mathbb{R}^p})) \tag{5.18}$$

with the inverse function $\mathbf{f}^{-1}$ defined as follows:

$$\mathbf{f}^{-1} : \begin{cases} \mathbb{R}^3 \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^2 \\ (\mathbf{y}, \mathbf{p}) & \rightarrow & \mathcal{R}^T(p_3) \begin{pmatrix} y_1 - p_1 \\ y_2 - p_2 \end{pmatrix} \end{cases} \tag{5.19}$$

where $\mathcal{R}^T$ is the transpose of $\mathcal{R}$. It consists in finding the set $\mathbb{P}$ by moving the visibility area $\mathbb{A}$ in order to intersect the landmark $\mathbb{B}$.

In the following examples, only the case of a sensor with a limited range and aperture angle is presented. Indeed, with this kind of sensor, the heading of the robot influences the visibility area and so the registration map will be a 3D set.

Example 1

Consider an ellipse as a landmark where the set of points inside the ellipse are generated as follows:

$$\mathbb{E} = \{\mathbf{x} \in \mathbb{R}^2 | (\frac{x_1}{a})^2 + (\frac{x_2}{b})^2 \leq 1\} \tag{5.20}$$

where a and b are the semi-axis of the ellipse aligned with the reference system. This ellipse is centered at $(0, 0)$. A separator $\mathcal{S}_{\mathbb{E}}$ can be built consistent with the set $\mathbb{E}$. A translation (t_x, t_y)

and rotation of angle α can be applied to the separator $\mathcal{S}_{\mathbb{E}}$ to move and orientate the ellipse by the function:

$$\mathbf{g}: \begin{cases} \mathbb{R}^3 \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^2 \\ ((t_x, t_y, \alpha), \mathbf{x}) & \rightarrow & \mathcal{R}(\alpha) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \end{cases} \tag{5.21}$$

The separator $\mathcal{S}_{\mathbb{B}}$ is then given by:

$$\mathcal{S}_{\mathbb{B}} = \mathbf{g} \circ \mathcal{S}_{\mathbb{E}} \tag{5.22}$$

As the function $\mathbf{g}$ is invertible, to compute the set $\mathbb{B}$, the inverse function $\mathbf{g}^{-1}$ has to be considered which corresponds to:

$$\mathbf{g}^{-1}: \begin{cases} \mathbb{R}^3 \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^2 \\ ((t_x, t_y, \alpha), \mathbf{x}) & \rightarrow & \mathcal{R}^T(\alpha) \begin{pmatrix} x_1 - t_x \\ x_2 - t_y \end{pmatrix} \end{cases} \tag{5.23}$$

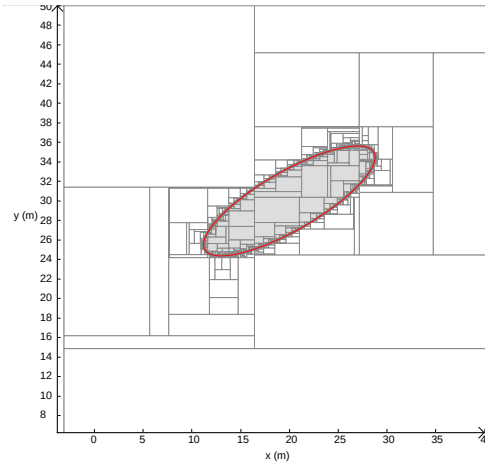
According to Theorem 4.4.1 or the definition 4.80 it is then possible to compute the set $\mathbb{B}$.

With $a = 10$, $b = 3$, $\alpha = 30°$, $t_x = 20m$ and $t_y = 30m$, it gives $\mathcal{S}_{\mathbb{B}}$ as depicted in Figure 5.4(a). Assume that the robot is equipped with a Forward Looking Sonar (FLS) that can only perceive the area forward the AUV. A separator is then built at the particular location $(0, 0, 0°)$ with a range of detection between 4 and $20m$ and a whole aperture angle of $60°$ as depicted in Figure 5.4(b) using the polar separator. The set of robot configurations that can detect a part of the landmark is computed according to Equation 5.15. The set $\mathbb{P}$ corresponds to a 3D shape that indicates the robot locations (x, y) and the heading θ that allow the detection of the landmark, i.e. $\mathbf{p} = (x, y, \theta)$. For ease of interpretation, only slices of this 3D set are provided to get a 2D representation.

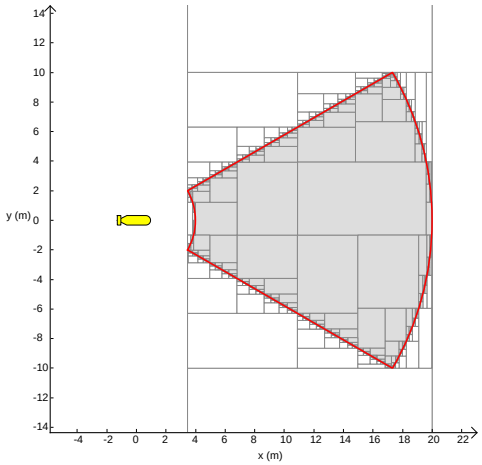
Firstly, some slices at different heading values are provided in Figure 5.4(c)(d)(e) where some AUV poses are presented in yellow with their visibility area shown by a blue pie. The contour of the ellipse is represented by a red contour.

Secondly, some slices of the 3D set $\mathbb{P}$ at different x or y values fixed to get a $(y-\theta)$ or $(x-\theta)$ representation of the solution sets are provided in Figure 5.5. For the different solution sets, some AUV poses are choiced, depicted by a red dot in the solution sets and the fixed value, and are represented in the 2D world $((x-y)$ coordinates) in the joint subfigure. In this latter, the union of all the intersections between the visibility area at these different poses and the landmark (ellipse) is provided.

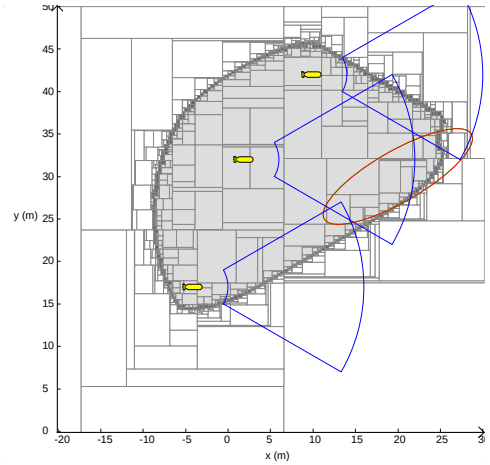
Notice that in each case, the solution sets provide the solution of the problem even in 3D.



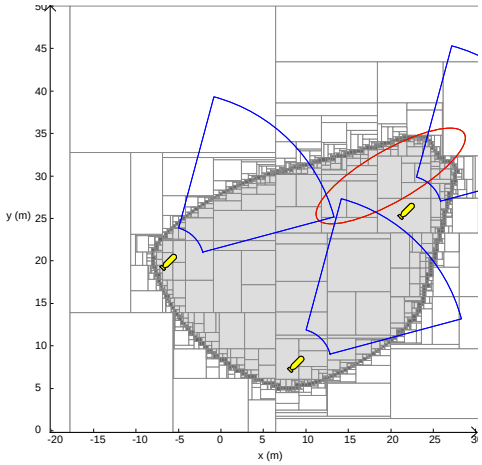
(a) Landmark set $\mathbb{B}$.



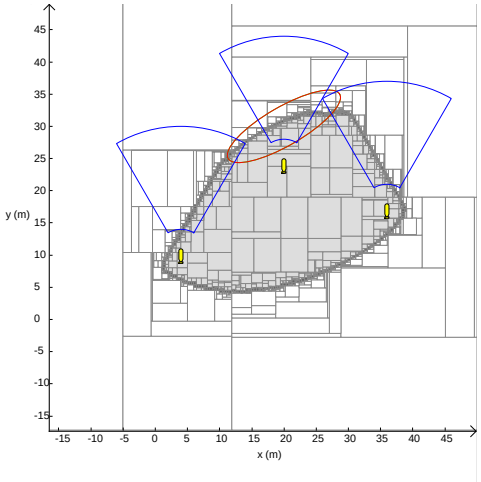
(b) Visibility of the sensor at the particular pose $(0, 0, 0°)$ drawn by the AUV.



(c) Registration map for a heading of the AUV at 0°.

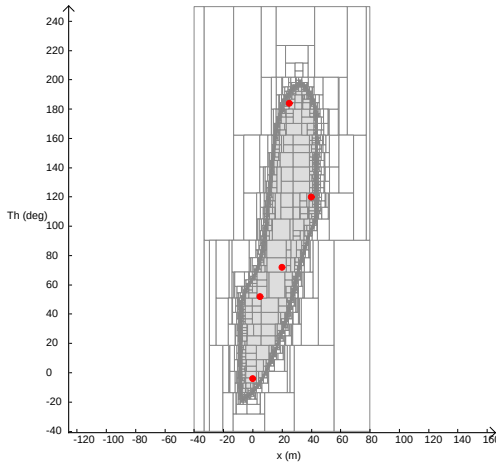


(d) Registration map for a heading of the AUV at 45°.

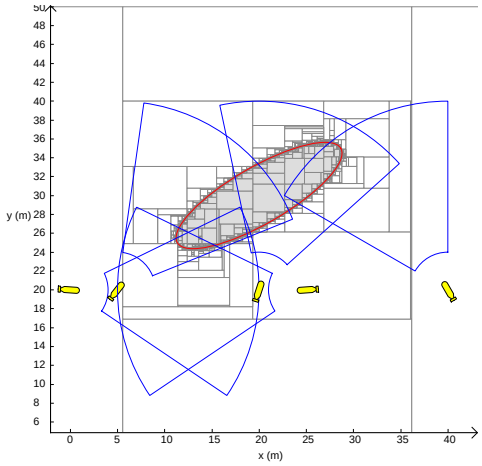


(e) Registration map for a heading of the AUV at 90°.

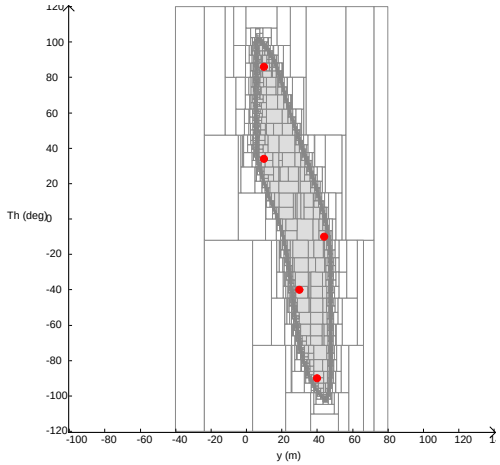
Figure 5.4: Registration map for different headings of the AUV considering an ellipse as a landmark and a forward looking sensor with limited range and aperture angle. Gray boxes are inside the set, white ones are outside and dark gray ones no conclusion can be made. Some poses of the AUV are depicted in yellow with the visible area drawn by blue pies.



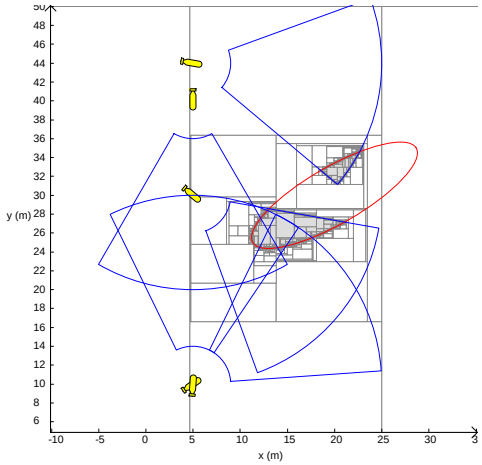
(a) Registration map with $y = 20m$ fixed. The representation is in the plane $(x - \theta)$.



(b) Representation in the $(x - y)$ plane of the AUV poses depicted by red dots in the figure (a).



(c) Registration map with $x = 5m$ fixed. The representation is in the $(y - \theta)$ plane.



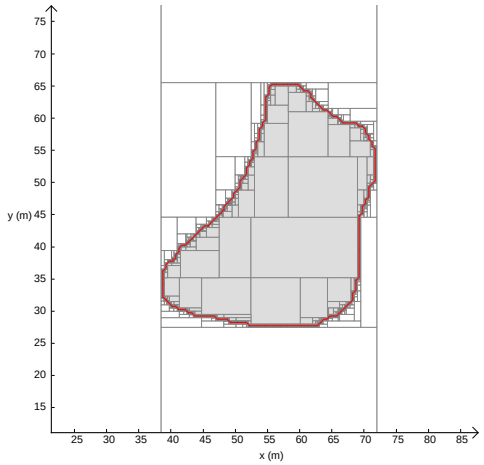
(d) Representation in the $(x - y)$ plane of the AUV poses depicted by red dots in the figure (c).

Figure 5.5: Registration map for x or y values fixed. Gray boxes are inside the set, white ones are outside and dark gray ones no conclusion can be made. Some poses of the AUV are depicted on the solution sets by red dots which are then represented in the $(x - y)$ plane in yellow with the visible area drawn by blue pies.

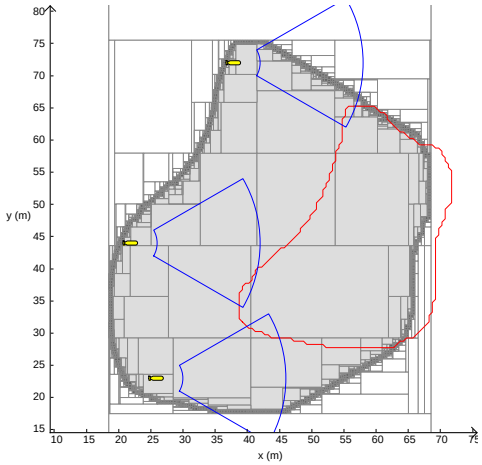
Example 2

Consider now a random shaped landmark $\mathbb{B}$ from an image separator as depicted in Figure 5.6(a). Using the same sensor as previously, the registration map can still be computed. Due to the 3D shape of the solution sets, the results assuming a fixed heading are firstly provided in Figure 5.6(b)(c)(d).

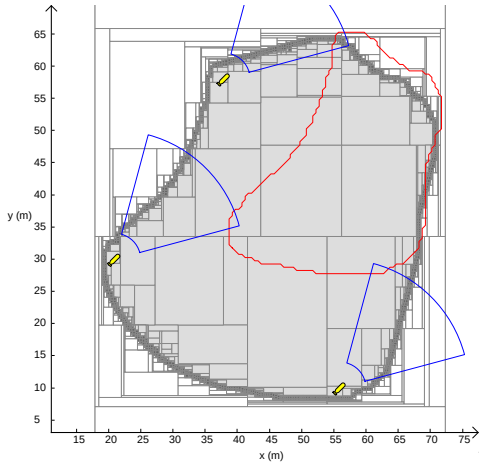
The results assuming a fixed x or y value to have the $(y-\theta)$ or $(x-\theta)$ solution are represented in Figure 5.7 where some AUV poses depicted as red dots in the solution sets according to the fixed value are represented in a joint subfigure for a representation in the 2D world ($(x - y)$ coordinates). Again the union of the intersection between the visibility area at these different poses and the landmark is provided.



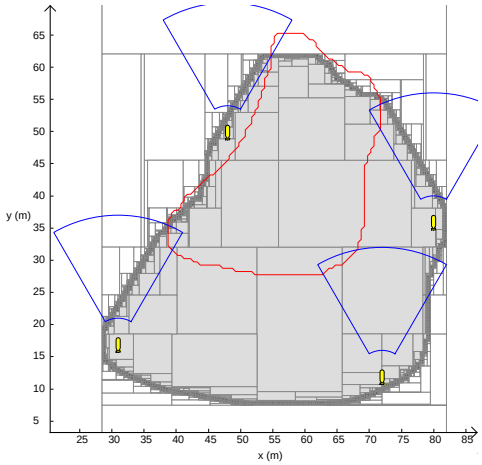
(a) Random shape landmark set $\mathbb{B}$ built with image separator.



(b) Registration map for a heading of the AUV at 0°.

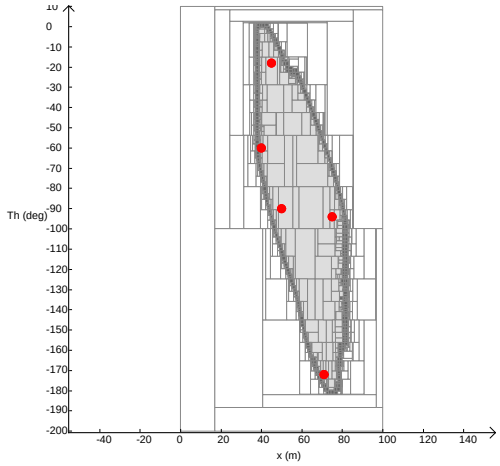


(c) Registration map for a heading of the AUV at 45°.

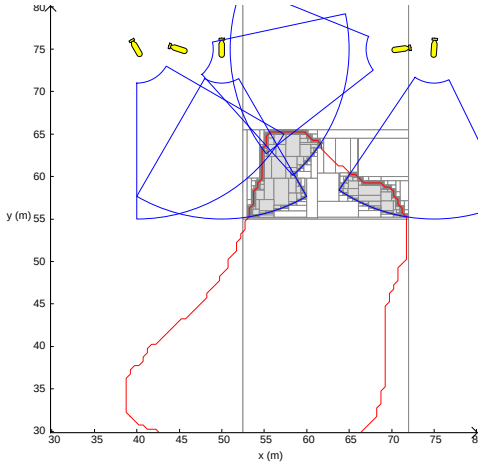


(d) Registration map for a heading of the AUV at 90°.

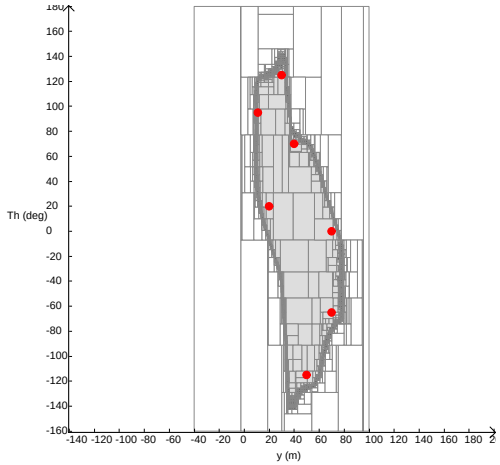
Figure 5.6: Registration map for different headings of the AUV considering a random shape as a landmark and a forward looking sensor with limited range and aperture angle. Gray boxes are inside the set, white ones are outside and dark gray ones no conclusion can be made. Some poses of the AUV are depicted in yellow with the visible area drawn by blue pies.



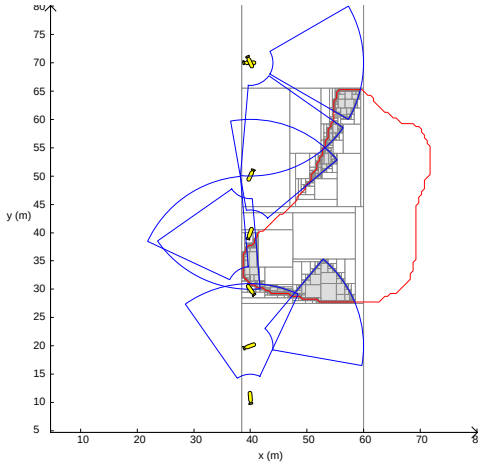
(a) Registration map with $y = 75m$ fixed. The representation is in the plane $(x - \theta)$.



(b) Representation in the $(x - y)$ plane of the AUV poses depicted by red dots in the figure (a).



(c) Registration map with $x = 40m$ fixed. The representation is in the plane $(y - \theta)$.



(d) Representation in the $(x - y)$ plane of the AUV poses depicted by red dots in the figure (c).

Figure 5.7: Registration map for x or y values fixed. Gray boxes are inside the set, white ones are outside and dark gray ones no conclusion can be made. Some poses of the AUV are depicted on the solution sets by red dots which are then represented in the $(x - y)$ plane in yellow with the visible area drawn by blue pies.

Punctual landmark

As mentioned in Chapter 2, sometimes the landmarks are considered as punctual due to their small size or for ease of interpretation. In this case, the set $\mathbb{B}$ is reduced to a singleton $\mathbb{B} = \{\mathbf{b}\}$. The registration map is still computable. Considering the landmark position $\mathbf{b} = (b_1, b_2)$ where (b_1, b_2) are the coordinates of the landmark, and the same transformation function $\mathbf{f}$ expressed in Equation 5.16, it leads then the following equation:

$$\mathbf{f}(\mathbf{x}, \mathbf{p}) = \mathbf{b} \tag{5.24}$$

By considering the inverse function, the registration map for a punctual landmark is then

defined by:

$$\mathbf{f}_{\mathbf{b}}^{-1} : \begin{cases} \mathbb{R}^3 & \rightarrow & \mathbb{R}^2 \\ \mathbf{p} & \rightarrow & \mathcal{R}^T(p_3) \begin{pmatrix} b_1 - p_1 \\ b_2 - p_2 \end{pmatrix} \end{cases} \tag{5.25}$$

where $\mathbf{b}$ are the coordinates of the punctual landmark.

Given a visibility area $\mathbb{A} \subset \mathbb{R}^2$ (at the particular pose $(0, 0, 0°)$) and the associated separator $\mathcal{S}_{\mathbb{A}}$, the registration map for the punctual landmark is defined as follows:

$$\mathbf{f}_{\mathbf{b}}^{-1}(\mathbb{P}) = \mathbb{A} \tag{5.26}$$

which is a set inversion problem.

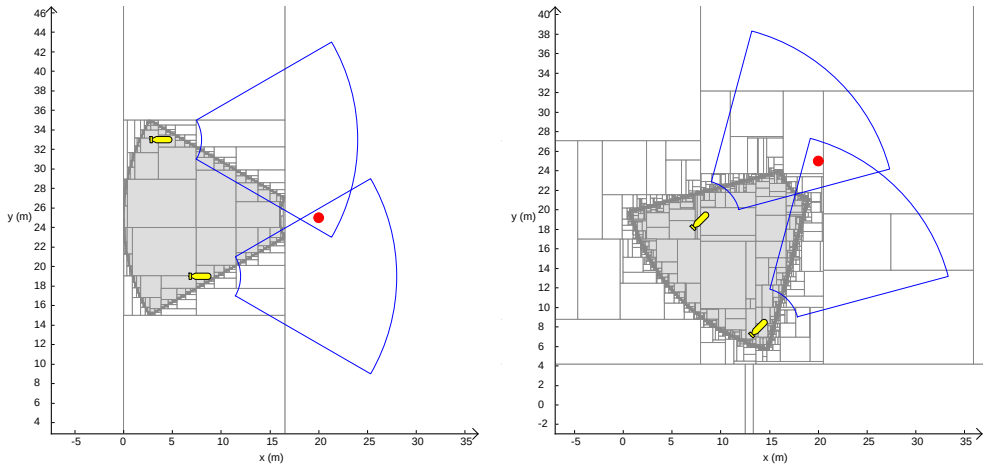
A separator for $\mathbb{P}$ can then be given by:

$$\mathcal{S}_{\mathbb{P}} = \mathbf{f}_{\mathbf{b}}(\mathcal{S}_{\mathbb{A}}) \tag{5.27}$$

where $\mathcal{S}_{\mathbb{A}}$ is the separator of the visibility area defined at the particular pose $(0, 0, 0°)$.

Example

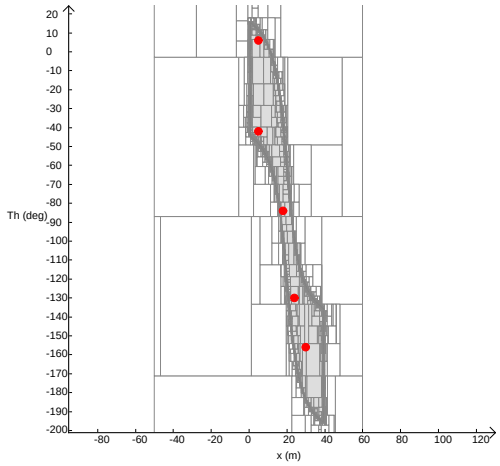
Consider again the same visibility area generated at the particular pose $(0, 0, 0°)$ and a punctual landmark located at $\mathbf{b} = (20, 25)$. The resulting sets $\mathbb{P}$ for different heading values are represented in Figure 5.8. The resulting sets for different x or y values are represented in Figure 5.9. Contrary to the other landmarks, the intersection of the visibility areas is depicted for the different AUV poses represented by red dots in the solution set considering a x or y value fixed. The punctual landmark has to be visible for all the AUV poses that are solution, it results in a non empty intersection of the visibility areas as depicted in the joint subfigure in Figure 5.9.



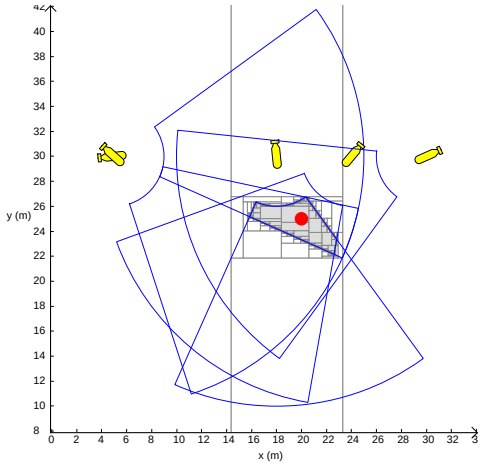
(a) Registration map for a heading of the AUV at 0°.

(b) Registration map for a heading of the AUV at 45°.

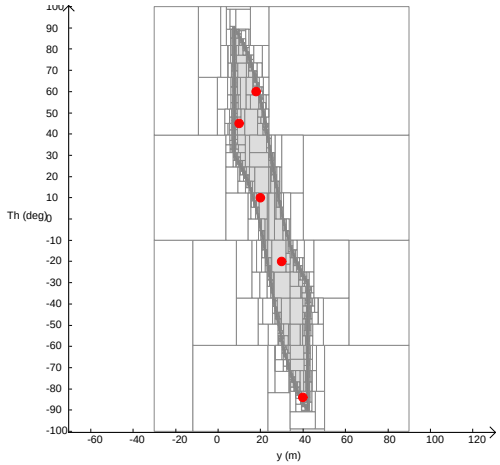
Figure 5.8: Registration map for different headings of the AUV considering a punctual landmark drawn by a red dot and assuming a forward looking sensor with limited range and aperture angle. Gray boxes are inside the set, white ones are outside and dark gray ones no conclusion can be made. Some poses of the AUV are depicted in yellow with the visible area drawn by blue pies.



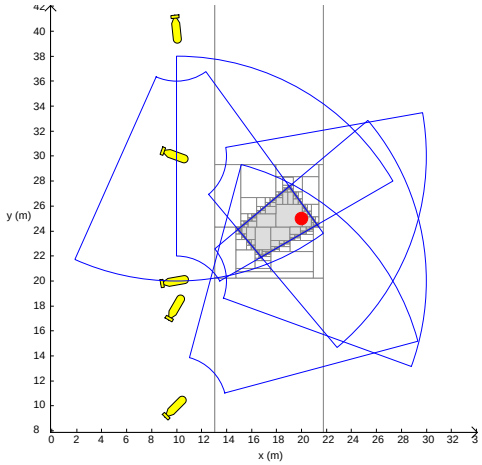
(a) Registration map with $y = 30m$ fixed. The representation is in the plane $(x - \theta)$.



(b) Representation in the $(x - y)$ plane of the AUV poses depicted by red dots in the solution set associated.



(c) Registration map with $x = 20m$ fixed. The representation is in the plane $(y - \theta)$.



(d) Representation in the $(x - y)$ plane of the AUV poses depicted by red dots in the solution set associated.

Figure 5.9: Registration map for x or y values fixed. Gray boxes are inside the set, white ones are outside and dark gray ones no conclusion can be made. Some poses of the AUV are depicted on the solution sets by red dots which are then represented in the $(x - y)$ plane in yellow with the visible area drawn by blue pies.

Registration map with heading defined

Note that if the heading p_3 of the AUV is defined (fixed) as depicted in the corresponding figures, the registration map is now a subset of $\mathbb{R}^2$ in the $(x - y)$ plane and only the translation parameters have to be found to compute the registration map according to the landmark. Indeed the angle of the rotation is defined, and therefore the initial visibility area at the particular pose $(0, 0, 0°)$ can be rotated of that angle p_3. The initial visibility area $\mathbb{A}$ is then a rotated set of angle p_3, it will be called $\mathbb{A}_{p_3}$. It comes from the fact that the order of the composition between the translation and the rotation does not matter. The visibility area can

then be rewritten as:

$$\mathbb{V}(\mathbf{p}) = \{\mathbf{z} \in \mathbb{R}^2 | \exists r \in [R_{min}, R_{max}], \exists \beta \in [-\theta_3, \theta_3], \mathbf{z} - \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} rcos(p_3 + \beta) \\ rsin(p_3 + \beta) \end{pmatrix}\} \quad (5.28)$$

$$= \{\mathbf{z} \in \mathbb{R}^2 | \exists r \in [R_{min}, R_{max}], \exists \gamma \in [p_3 - \theta_3, p_3 + \theta_3], \mathbf{z} - \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} rcos(\gamma) \\ rsin(\gamma) \end{pmatrix}\} \quad (5.29)$$

where p_3 corresponding to the heading of the AUV is defined. This set represents a polar constraint.

Minimal separator for a punctual landmark

In the case of a punctual landmark $\mathbf{m}$, the registration map is then given by the set $\mathbb{P}$:

$$\mathbb{P} = \{\mathbf{p} \in \mathbb{R}^2 | \exists r \in [R_{min}, R_{max}], \exists \gamma \in [p_3 - \theta_3, p_3 + \theta_3], \mathbf{m} - \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} rcos(\gamma) \\ rsin(\gamma) \end{pmatrix}\} \quad (5.30)$$

where $\mathbf{m}$ corresponds to the landmark coordinates.

Using the polar separator, the registration map for the punctual landmark can then be found easily as in Equation 4.100 and the previous examples with punctual landmarks.

What about the other landmarks?

When the heading p_3 of the vehicle is defined, the registration map is computed only based on the translation parameters. The function $\mathbf{f}$ in Equation 5.9 becomes:

$$\mathbf{f} : \begin{cases} \mathbb{R}^2 \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^2 \\ (\mathbf{p}, \mathbf{a}) & \rightarrow & \mathbf{a} + \mathbf{p} \end{cases} \quad (5.31)$$

Therefore, the set $\mathbb{P}$ becomes according to Equation 5.9:

$$\mathbb{P} = \{\mathbf{p} \in \mathbb{R}^2 | (\mathbb{A}_{p_3} + \mathbf{p}) \cap \mathbb{B} \neq \emptyset\} \quad (5.32)$$

where the set $\mathbb{A}_{p_3}$ can be both the rotated initial visibility area of angle p_3 or the polar separator at the particular pose $(0, 0, p_3)$ of the AUV expressed by the set:

$$\mathbb{V}(0, 0, p_3) = \{\mathbf{z} \in \mathbb{R}^2 | \exists r \in [R_{min}, R_{max}], \exists \beta \in [-\theta_3, \theta_3], \mathbf{z} = \begin{pmatrix} rcos(p_3 + \beta) \\ rsin(p_3 + \beta) \end{pmatrix}\} \quad (5.33)$$

$$= \{\mathbf{z} \in \mathbb{R}^2 | \exists r \in [R_{min}, R_{max}], \exists \gamma \in [p_3 - \theta_3, p_3 + \theta_3], \mathbf{z} = \begin{pmatrix} rcos(\gamma) \\ rsin(\gamma) \end{pmatrix}\} \quad (5.34)$$

which corresponds to a polar constraint.

Equation 5.32 leads to:

$$\mathbb{P} = \{\mathbf{p} \in \mathbb{R}^2 | (\mathbb{A}_{p_3} + \mathbf{p}) \cap \mathbb{B} \neq \emptyset\} \quad (5.35)$$

$$= \overline{\{\mathbf{p} \in \mathbb{R}^2 | (\mathbb{A}_{p_3} + \mathbf{p}) \cap \mathbb{B} = \emptyset\}} \quad (5.36)$$

$$= \overline{\{\mathbf{p} \in \mathbb{R}^2 | (\mathbb{A}_{p_3} + \mathbf{p}) \subset \overline{\mathbb{B}}\}} \quad (5.37)$$

The set $\overline{\mathbb{P}}$ corresponds then to:

$$\overline{\mathbb{P}} = \{\mathbf{p} \in \mathbb{R}^2 | (\mathbb{A}_{p_3} + \mathbf{p}) \subset \overline{\mathbb{B}}\} \quad (5.38)$$

By defining $\overline{\mathbb{B}} = \mathbb{C}$ and replacing $\mathbb{A}_{p_3}$ by $\mathbb{A}$ (just for ease of reading), the set $\overline{\mathbb{P}}$ is then given by:

$$\overline{\mathbb{P}} = \{\mathbf{p} \in \mathbb{R}^2 | (\mathbb{A} + \mathbf{p}) \subset \mathbb{C}\} \tag{5.39}$$

which corresponds to the Minkowski difference. When dealing with translation between sets, the Minkowski sum and difference are interesting tools and will be detailed in a further subsection. As presented in [74], the Minkowski operators can be seen as a shape registration where the set $\mathbb{P}$ corresponds to:

$$\mathbb{P} = \{\mathbf{p} | \mathbf{f}(\mathbb{A}, \mathbf{p}) \subset \mathbb{B}\} \tag{5.40}$$

This problem has been presented in [71] and a brief overview is given in the next subsection before introducing the Minkowski operators.

5.2.3 Shape registration

The problem of registration has been widely discussed in the image processing community [348] for example, where points between two overlapping images are tried to be matched. Surface, lines, etc... may be tried to be matched. In a bathymetric survey, when using a MultiBeam EchoSounder (MBES) for high coverage, the points given an (x, y, z) information are post-processed to match all the surfaces of the seabed from the different straight line paths of the platform. A famous technique is the registration of points cloud with Iterative Closest Points [25].

According to the same notations as in the definition of the registration map problem (Equations (5.5)(5.6)(5.7)(5.8)), the shape registration problem corresponds to the set:

$$\mathbb{P} = \{\mathbf{p} | \mathbf{f}(\mathbb{A}, \mathbf{p}) \subset \mathbb{B}\} \tag{5.41}$$

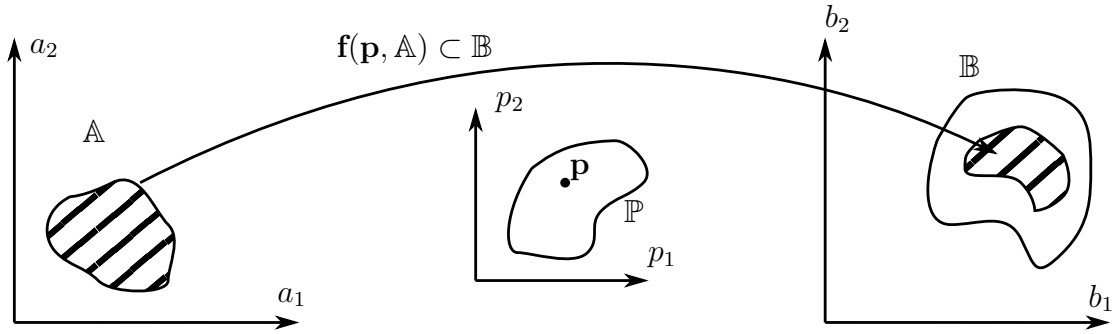


Figure 5.10: Example of the shape registration concept associated to the constraint $\mathbf{f}(\mathbf{p}, \mathbb{A}) \subset \mathbb{B}$. The set $\mathbb{P}$ corresponds to the parameters of the function $\mathbf{f}$ such as the transformation of the shape $\mathbb{A}$ is included in $\mathbb{B}$.

A transformation vector $\mathbf{p}$ is consistent if after transformation of $\mathbb{A}$ it is included in $\mathbb{B}$ as depicted in Figure 5.10. This leads to:

$$\begin{aligned}
& \mathbf{f}(\mathbb{A}, \mathbf{p}) \subset \mathbb{B} && (5.42)\\
\Leftrightarrow \quad & \forall \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B} && (5.43)\\
\Leftrightarrow \quad & \neg\exists \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \overline{\mathbb{B}} && (5.44)\\
\Leftrightarrow \quad & \neg\exists \mathbf{a} \in \mathbb{A}, (\mathbf{a}, \mathbf{p}) \in \mathbf{f}^{-1}(\overline{\mathbb{B}}) && (5.45)\\
\Leftrightarrow \quad & \neg\exists \mathbf{a}, (\mathbf{a}, \mathbf{p}) \in \mathbb{A} \times \mathbb{R}^p \wedge (\mathbf{a}, \mathbf{p}) \in \mathbf{f}^{-1}(\overline{\mathbb{B}}) && (5.46)
\end{aligned}$$

Similarly as in the previous subsection,

$$\mathbb{P} = \overline{proj_{\mathbf{p}}((\mathbb{A} \times \mathbb{R}^p) \cap \mathbf{f}^{-1}(\overline{\mathbb{B}}))} \tag{5.47}$$

If separators $\mathcal{S}_{\mathbb{A}}$ for $\mathbb{A}$ and $\mathcal{S}_{\mathbb{B}}$ for $\mathbb{B}$ are available, then a separator $\mathcal{S}_{\mathbb{P}}$ for $\mathbb{P}$ can be computed using the separator algebra presented. The separator $\mathcal{S}_{\mathbb{P}}$ consistent with $\mathbb{P}$ is then given by:

$$\mathcal{S}_{\mathbb{P}} = \overline{proj_{\mathbf{p}}((\mathcal{S}_{\mathbb{A}} \times \mathcal{S}_{\mathbb{R}^p}) \cap \mathbf{f}^{-1}(\overline{\mathcal{S}_{\mathbb{B}}}))} \tag{5.48}$$

Note that the registration map can be seen as a shape registration problem where the following separators were built in [71]:

$$\begin{aligned} \mathbb{P}_{=} &= \{\mathbf{p} \in \mathbb{R}^p | \mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} = \emptyset\} && (5.49)\\ &= \{\mathbf{p} \in \mathbb{R}^p | \mathbf{f}(\mathbb{A}, \mathbf{p}) \subset \overline{\mathbb{B}}\} && (5.50) \end{aligned}$$

and

$$\begin{aligned} \mathbb{P}_{\neq} &= \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} \neq \emptyset\} && (5.51)\\ &= \overline{\mathbb{P}_{=}} && (5.52) \end{aligned}$$

The registration map would correspond to $\mathbb{P}_{\neq}$.

Since the shape registration problem has been introduced, now focus on the Minkowski operators.

5.2.4 Minkowski Sum and Difference

When the heading p_3 of the AUV is defined and if the registration map is constituted by a composition of translation and rotation, the problem corresponds to a Minkowski operation between the set representing the landmark $\mathbb{B}$ and the visibility area $\mathbb{V}$ of the sensor at the particular pose $(0, 0, p_3)$.
Minkowski operations are widely used in morphological mathematics to perform dilation or erosion of sets. The Minkowski operations have already been performed with subpavings in [292]. The Minkowski operators can also be seen as a shape registration problem as proposed in [74] and will be detailed here.

5.2.4.1 Minkowski Difference

Definition

Given two sets $\mathbb{A} \subset \mathbb{R}^n$ and $\mathbb{B} \subset \mathbb{R}^n$, the Minkowski difference [230], denoted as $\ominus$, is defined as follows:

$$\mathbb{B} \ominus \mathbb{A} = \{\mathbf{p} \in \mathbb{R}^n | \mathbb{A} + \mathbf{p} \subset \mathbb{B}\} \tag{5.53}$$

Proposition

If separators $\mathcal{S}_{\mathbb{A}}$ for $\mathbb{A}$ and $\mathcal{S}_{\mathbb{B}}$ for $\mathbb{B}$ are available then the Minkowski difference [74] is defined as follows:

$$\mathcal{S}_{\mathbb{B}} \ominus \mathcal{S}_{\mathbb{A}} = \overline{proj_{\mathbf{p}}((\mathcal{S}_{\mathbb{A}} \times \mathcal{S}_{\mathbb{R}^p}) \cap \mathbf{f}^{-1}(\overline{\mathcal{S}_{\mathbb{B}}}))} \tag{5.54}$$

where $\mathbf{f}(\mathbf{p}, \mathbf{a}) = \mathbf{p} + \mathbf{a}$.

Proof

The Minkowski difference corresponds to a shape registration where $\mathbf{f}(\mathbf{p}, \mathbf{a}) = \mathbf{p} + \mathbf{a}$, which is a translation of vector $\mathbf{p}$.

Example

Consider a triangle as the initial set $\mathbb{B}$ that is defined by 3 inequalities:

$$\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^2 | x_2 + x_1 - 2 \leq 0 \text{ and } x_2 - x_1 - 2 \leq 0 \text{ and } x_2 + 1 \geq 0\} \tag{5.55}$$

which is represented in Figure 5.11(b).

Consider a circle of radius 0.5 as the set $\mathbb{A}$ depicted in Figure 5.11(a). In image processing, it is called the structuring element.

The Minkowski difference is shown in Figure 5.11(c). The green triangle shows the initial triangle and the green circle corresponds to the structuring element on the border of the initial triangle. Notice how the dilation is given by the Minkowski difference.

5.2.4.2 Minkowski Sum

Definition

Given two sets $\mathbb{A} \subset \mathbb{R}^n$ and $\mathbb{B} \subset \mathbb{R}^n$, the Minkowski sum [230], denoted as $\oplus$, is defined as follows:

$$\mathbb{A} \oplus \mathbb{B} = \{\mathbf{a} + \mathbf{b}, \mathbf{a} \in \mathbb{A}, \mathbf{b} \in \mathbb{B}\} \tag{5.56}$$

Proposition

If separators $\mathcal{S}_\mathbb{A}$ for $\mathbb{A}$ and $\mathcal{S}_\mathbb{B}$ for $\mathbb{B}$ are available then the Minkowski sum [74] is defined as follows:

$$\mathcal{S}_\mathbb{A} \oplus \mathcal{S}_\mathbb{B} = \overline{\overline{\mathcal{S}_\mathbb{B}} \ominus -\mathcal{S}_\mathbb{A}} \tag{5.57}$$

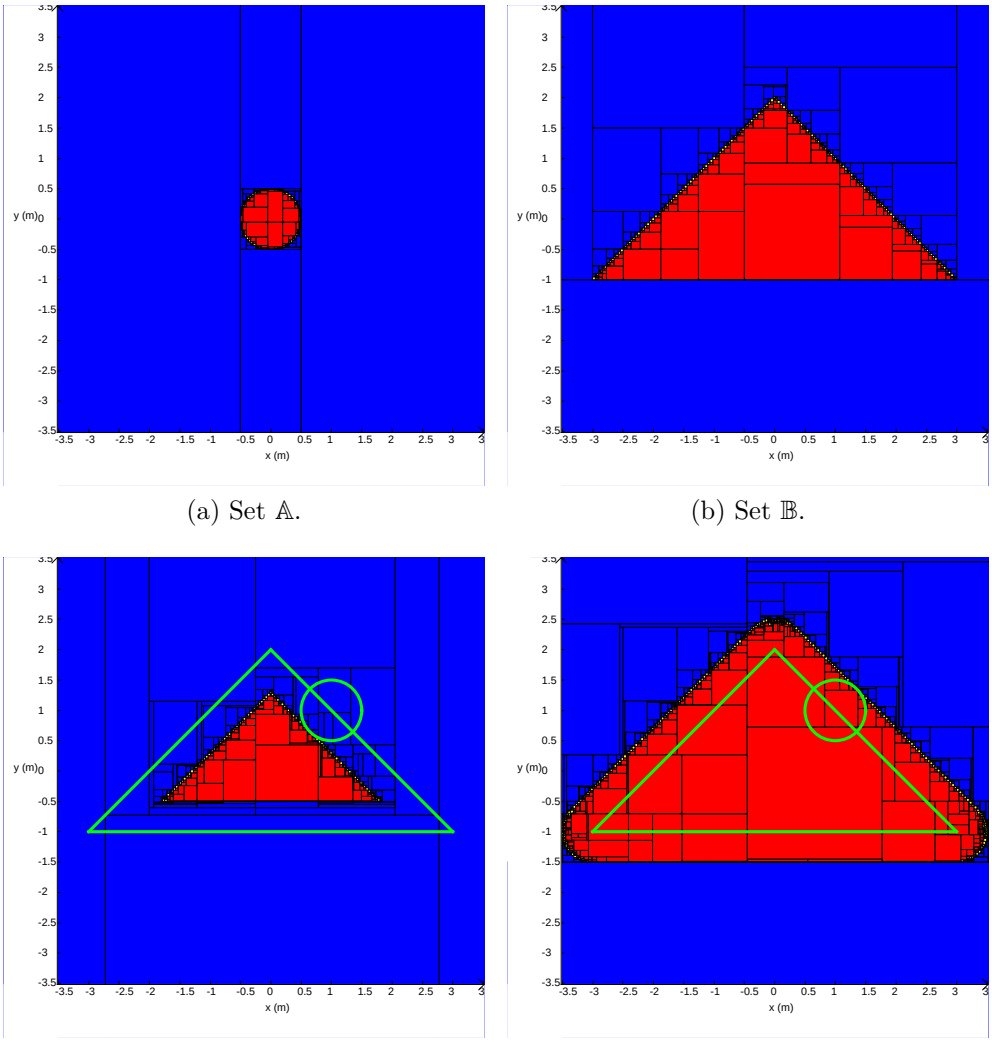
Proof

$$\begin{aligned}
\mathbb{A} \oplus \mathbb{B} &= \{\mathbf{p} | \exists \mathbf{a} \in \mathbb{A}, \exists \mathbf{b} \in \mathbb{B}, \mathbf{p} = \mathbf{a} + \mathbf{b}\} && (5.58)\\
&= \{\mathbf{p} | \exists \mathbf{a} \in \mathbb{A}, \exists \mathbf{b} \in \mathbb{B}, \mathbf{p} - \mathbf{a} = \mathbf{b}\} && (5.59)\\
&= \{\mathbf{p} | (\mathbf{p} - \mathbb{A}) \cap \mathbb{B} \neq \emptyset\} && (5.60)\\
&= \overline{\{\mathbf{p} | (\mathbf{p} - \mathbb{A}) \cap \mathbb{B} = \emptyset\}} && (5.61)\\
&= \overline{\{\mathbf{p} | (\mathbf{p} + (-\mathbb{A})) \subset \overline{\mathbb{B}}\}} && (5.62)\\
&= \overline{\overline{\mathbb{B}} \ominus -\mathbb{A}} && (5.63)
\end{aligned}$$

Therefore the separator for $\mathbb{A} \oplus \mathbb{B}$ is given by $\overline{\overline{\mathcal{S}_\mathbb{B}} \ominus -\mathcal{S}_\mathbb{A}}$.

Example

Consider the same sets as for the difference. The result of the Minkowski sum is given in Figure 5.11(d). Notice how the triangle is eroded by the disk.



(a) Set $\mathbb{A}$.

(b) Set $\mathbb{B}$.

(c) Minkowsky difference $\mathbb{B} \ominus \mathbb{A}$.

(d) Minkowsky sum $\mathbb{A} \oplus \mathbb{B}$.

Figure 5.11: Minkowski operations. The green lines depict the initial triangle and a circle on the contour of the initial triangle with the radius of the set $\mathbb{A}$.

5.2.4.3 Minkowski operators for the registration map

As mentioned earlier, the registration map corresponds almost to a Minkowski difference when the heading p_3 of the AUV is fixed. Considering only the translation with a given rotated visibility area $\mathbb{A}_{p_3}$ according to the desired heading p_3 (or given by the polar separator) at the location $(0, 0)$, the registration map is given for a landmark $\mathbb{B}$ by the set in Equation 5.37 which leads to:

$$\mathbb{P} = \overline{\{\mathbf{p} \in \mathbb{R}^2 | (\mathbb{A}_{p_3} + \mathbf{p}) \subset \overline{\mathbb{B}}\}} \tag{5.64}$$

$$= \overline{\overline{B} \ominus \mathbb{A}_{p_3}} \tag{5.65}$$

Considering a separator $\mathcal{S}_{\mathbb{B}}$ for $\mathbb{B}$ and a separator $\mathcal{S}_{\mathbb{A}}$ for $\mathbb{A}_{p_3}$, the set consistent with $\mathbb{P}$ at

the desired heading p_3 is given by:

$$\begin{aligned} \mathcal{S}_{\mathbb{P}} &= \overline{\overline{\mathcal{S}_{\mathbb{B}}} \ominus \mathcal{S}_{\mathbb{A}}} && (5.66) \\ &= \overline{\overline{proj_{\mathbf{p}}((\mathcal{S}_{\mathbb{A}} \times \mathcal{S}_{\mathbb{R}^p}) \cap \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{B}}))}} && (5.67) \\ &= proj_{\mathbf{p}}((\mathcal{S}_{\mathbb{A}} \times \mathcal{S}_{\mathbb{R}^p}) \cap \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{B}})) && (5.68) \end{aligned}$$

which corresponds finally to the definition of the registration map based on projection in Equation 5.15 as expected. It finally corresponds to the definition of the registration map in Equation 5.14 which here is explained as a Minkowsky difference between a landmark set $\mathbb{B}$ and a visibility area $\mathbb{A}$ (at a given heading p_3).

The resulting sets for the different examples on the landmarks are the same due to the fact that the computation is based on projections. It slightly differs in the number of bisections if the minimal polar separator is used at a given heading.

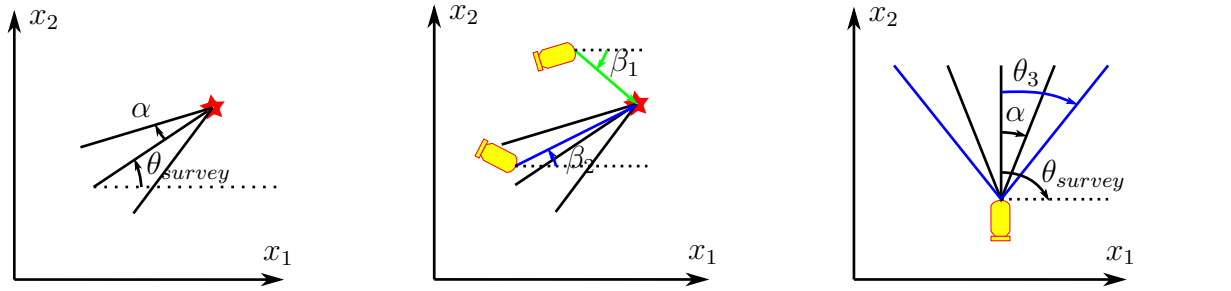
5.2.5 Point of view of a landmark

In Chapter 2, the notion of point of view was introduced in the underwater environment. Indeed, seeing a landmark at a specific point of view may not be detectable at a different point of view. For example, at another point of view, the landmarks may be covered by sand or do not have the same backscattered property due to the geometric aspect of the landmark. Moreover the sand ripples for instance, as explained, are not detectable at every aspects. During a survey mission, the landmarks are detected at specific angles of view. In Equation 2.21, an angular flexibility was introduced as it is quite impossible to revisit (to see again) a landmark at the exact same point of view. Denoting as θ_{survey} the angle of detection of a landmark during the survey mission, this landmark is then detectable in a interval:

$$[\theta_{detection}] = \theta_{survey} + [-\alpha, \alpha] \tag{5.69}$$

where α is the angular flexibility ($\approx 10 - 15°$). For example, if the orientation of the sand ripples is known, the parameter α can be adapted according to the orientation and the angular abilities to detect them. The concept of point of view is illustrated in Figure 5.12 where a punctual landmark is considered. The point of view with the angular flexibility indicates a set of robot position (x, y) that may enable the sensor to detect it. In Figure 5.12(b) this constraint is illustrated with two AUV positions where the green measurement is not compatible with the point of view contrary to the blue one. Moreover, the point of view reduces the set of possible AUV headings according to the aperture angle (θ_3 which is the half aperture angle) of the sensor as depicted in Figure 5.12(c) where the interval of headings θ^{AUV} can be guessed:

$$\begin{aligned} \theta^{AUV}_{max} &= \theta_{survey} + \alpha + \theta_3 && (5.70) \\ \theta^{AUV}_{min} &= \theta_{survey} - \alpha - \theta_3 && (5.71) \end{aligned}$$



(a) Point of view at an angle θ_{survey} with an angular flexibility α.

(b) Constraint on the AUV position where two AUV positions are drawn. The green measurement (direction) cannot detect the landmark according to the point of view contrary to the blue one.

(c) Constraint on the AUV heading. In the figure, the heading is aligned with the angle θ_{survey}.

Figure 5.12: Principle of the point of view with the constraints.

The point of view is then an additional constraint on the pose of the robot that can be decomposed in two constraints:

A first constraint, depicted in Figure 5.12(c), on the heading of the AUV according to the aperture angle of the sensor:

$$\begin{aligned} \mathbb{P}^{\theta_3}_{view}([\theta_{detection}]) &= \{\mathbf{p} \in \mathbb{R}^3 | \exists \gamma \in [-\theta_3, \theta_3], p_3 + \gamma \in [\theta_{detection}]\} && (5.72) \\ &= \{\mathbf{p} \in \mathbb{R}^3 | p_3 \in [\theta_{detection}] - [-\theta_3, \theta_3]\} && (5.73) \end{aligned}$$

where θ_3 corresponds to the half aperture angle of the sensor.

A second constraint, depicted in Figure 5.12(b), on the position of the AUV according to the landmark $\mathbb{B}$:

$$\mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}]) = \{\mathbf{p} \in \mathbb{R}^3 | \exists \mathbf{b} \in \mathbb{B}, \text{atan2}(b_2 - p_2, b_1 - p_1) \in [\theta_{detection}]\} \quad (5.74)$$

These constraints are separated for ease of reading.

The second constraint can be reformulated as follows:

$$\begin{aligned} \mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}]) &= \{\mathbf{p} \in \mathbb{R}^3 | \exists \mathbf{b} \in \mathbb{B}, \exists \theta \in [\theta_{detection}], \exists r \in \mathbb{R}^+, \begin{pmatrix} b_1 - p_1 \\ b_2 - p_2 \end{pmatrix} = \begin{pmatrix} r \cdot \cos\theta \\ r \cdot \sin\theta \end{pmatrix}\} && (5.75) \\ &= \{\mathbf{p} \in \mathbb{R}^3 | \exists \mathbf{b} \in \mathbb{B}, \exists \theta \in [\theta_{detection}], \exists r \in \mathbb{R}^+, \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = -\begin{pmatrix} r \cdot \cos\theta \\ r \cdot \sin\theta \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}\} \end{aligned}$$

This second constraint is independent of the heading p_3. The first term on the right hand side of the set is the polar constraint defined by the set:

$$\mathbb{X}_{polar} = \{\mathbf{x} \in \mathbb{R}^2 | \exists r \in \mathbb{R}^+, \exists \theta \in [\theta_{detection}], \mathbf{x} = \begin{pmatrix} r \cdot \cos\theta \\ r \cdot \sin\theta \end{pmatrix}\} \quad (5.76)$$

Finally, this second constraint corresponds to a 2D Minkowski sum of the polar constraint (with the sign $-$) and the landmark $\mathbb{B}$ expressed as follows:

$$\mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}]) = (-\mathbb{X}_{polar}) \oplus \mathbb{B} \quad (5.77)$$

The Minkowski sum is only realized on the 2D plane as it only contracts the (x, y) position of the vehicle according to the landmark $\mathbb{B}$ and the point of view $[\theta_{detection}]$.

The detection of a landmark at a desired point of view has to consider the aperture angle of the sensor to be sure the visible points belonging to the landmark are viewed with an angle corresponding to the angle of detection $[\theta_{detection}]$. This concept is illustrated in the following example.

Example

Consider the shaped landmark $\mathbb{B}$ (where a separator $\mathcal{S}_{\mathbb{B}}$ is built) previously presented and the same forward looking sensor, i.e. same range and aperture angle. The robot pose is defined at $\mathbf{x} = (58, 25, 60°)$. The visibility area $\mathbb{V}(\mathbf{x})$ of the sensor at this pose can be computed considering the set in Equation 5.4 by using a composition of translation and the polar constraint. The set of points visible $\mathbb{X}$ from the landmark considering this sensor and this robot pose $\mathbf{x}$ are given by:

$$\mathbb{X}_{\mathbf{x}} = \mathbb{B} \cap \mathbb{V}(\mathbf{x}) \tag{5.78}$$

The result is depicted in Figure 5.13(a). However the point of view of these points depend on their locations and the position of the robot. Now consider that the landmark is only visible at a particular point of view as defined in Equation 5.69. The set of points from the landmark that are visible by the sensor at this pose at the particular point of view $[\theta_{detection}]$ is given by:

$$\mathbb{X}_{\mathbf{x}}([\theta_{detection}]) = \{\mathbf{z} \in \mathbb{B} \cap \mathbb{V}(\mathbf{x}) | \operatorname{atan2}(z_2 - x_2, z_1 - x_1) \in [\theta_{detection}]\} \tag{5.79}$$

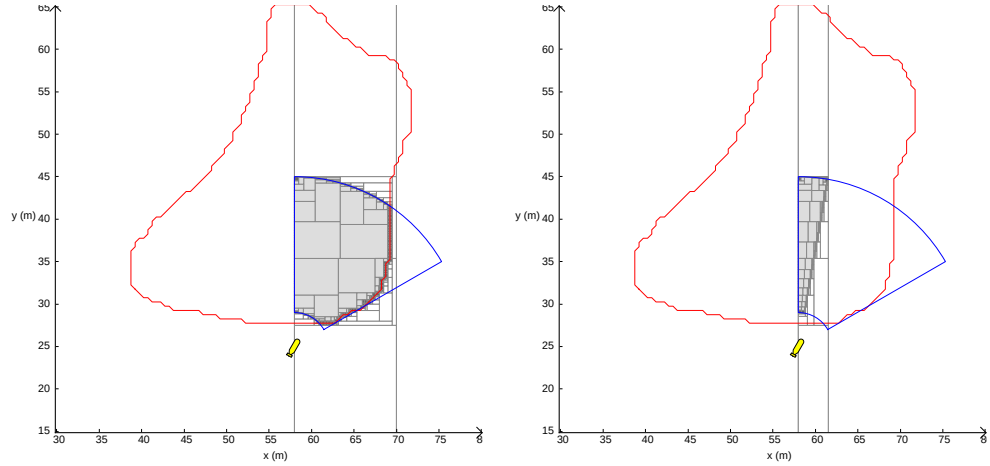
where the point of view is an additional constraint on the visible point defined by the set:

$$\mathbb{P}_{view}([\theta_{detection}]) = \{\mathbf{z} \in \mathbb{R}^2 | \operatorname{atan2}(z_2 - x_2, z_1 - x_1) \in [\theta_{detection}]\} \tag{5.80}$$

The result is finally given by:

$$\mathbb{X}_{\mathbf{x}}([\theta_{detection}]) = \mathbb{B} \cap \mathbb{V}(\mathbf{x}) \cap \mathbb{P}_{view}([\theta_{detection}]) \tag{5.81}$$

Consider an angle of detection at 90° and an angular flexibility at 10° leading to $[\theta_{detection}] = [80, 100](deg)$, this set is depicted in Figure 5.13(b). Notice that the set is smaller than the previous one and only a part of the landmark is actually visible despite a relatively high coverage area.



(a) Set of visible points from the landmark by the sensor at the robot pose $(58, 25, 60°)$.

(b) Set of visible points from the landmark by the sensor at the robot pose $(58, 25, 60°)$ and respecting the point of view $[\theta_{detection}]$.

Figure 5.13: Difference between visible points from a landmark when considering or not the point of view of the landmark.

The set in Equation 5.73 can be simplified by using the interval arithmetics for computing the difference $[\theta_{detection}] - [-\theta_3, \theta_3]$. Considering the registration map $\mathbb{P}^{\mathbb{B}}$ for a landmark $\mathbb{B}$ (remind that $\mathbb{P}^{\mathbb{B}} \subset \mathbb{R}^3$), the set of robot configurations able to detect the landmark at a particular point of view $[\theta_{detection}]$ is given by:

$$\mathbb{P}^{\mathbb{B}}([\theta_{detection}]) = \mathbb{P}^{\mathbb{B}} \cap \mathbb{P}^{\theta_3}_{view}([\theta_{detection}]) \cap \mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}]) \tag{5.82}$$

where $\mathbb{P}^{\theta_3}_{view}([\theta_{detection}])$ and $\mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}])$ are computed according to Equation 5.73 and Equation 5.74 or 5.77 respectively. Due to the fact that the constraint on the point of view from the aperture angle is only along the $\theta = p_3$ dimension and the constraint according to the landmark is on the $(x, y) = (p_1, p_2)$ plane, the resulting separator giving the set in Equation 5.82 can be defined as follows:

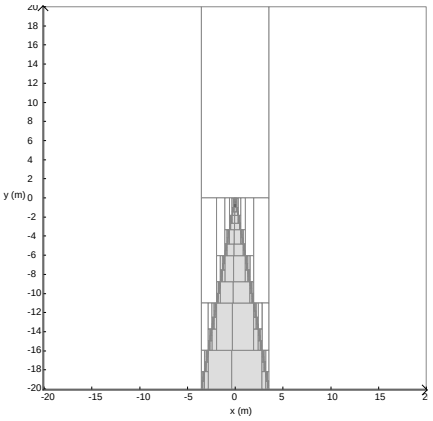
$$\mathcal{S}_{\mathbb{P}^{\mathbb{B}}([\theta_{detection}])} = \mathcal{S}_{\mathbb{P}^{\mathbb{B}}} \cap (\mathcal{S}_{\mathbb{R}^2} \times \mathcal{S}_{\mathbb{P}^{\theta_3}_{view}([\theta_{detection}])}) \cap (\mathcal{S}_{\mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}])} \times \mathcal{S}_{\mathbb{R}}) \tag{5.83}$$

where $\mathcal{S}_{\mathbb{R}^2}$ and $\mathcal{S}_{\mathbb{R}}$ are used to adapt the dimension.

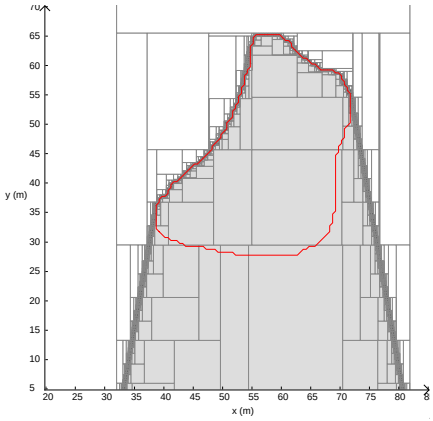
Example 1

Consider again the same shaped landmark $\mathbb{B}$, the same sensor with an half aperture angle θ_3 defined at 30° and the same point of view $[\theta_{detection}] = [80, 100](deg)$. Due to the 3D aspect of the resulting set, some slices at different headings will be shown. The first constraint on the point of view according to the aperture angle can be checked directly when headings are choiced. Indeed the possible headings belong to the interval $[\theta_{detection}] - [-\theta_3, \theta_3] = [50, 130](deg)$. Using the polar separator, the set corresponding to $-\mathbb{X}_{polar}$ expressed in Equation 5.76 (by making the transformation for the sign $-$) is represented in Figure 5.14(a). The Minkowski sum defined in Equation 5.77 is depicted in Figure 5.14(b) to represent the set $\mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}])$. The initial registration map $\mathbb{P}^{\mathbb{B}}$ taken at heading $\theta = 60°$ is represented in Figure 5.14(c). This heading $\theta = 60° \in [\theta_{detection}]$ belongs to the possible heading values, so the last constraint $\mathbb{P}^{\theta_3}_{view}([\theta_{detection}])$ is satisfied. Finally the registration map $\mathbb{P}^{\mathbb{B}}([\theta_{detection}])$ at that given heading is proposed in Figure 5.14(d) where some AUV poses are presented with their visibility area

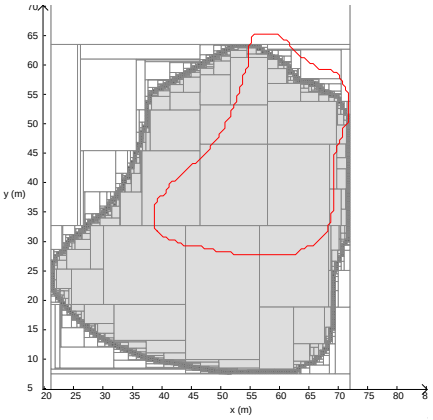
drawn by blue pies. Notice how the set of (x, y) positions is reduced to satisfy the constraint $\mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}])$. Consider now the registration map at the heading $\theta = 0°$. The computation of the set $\mathbb{P}^{\mathbb{B}} \cap \mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}])$ corresponding to the intersection of the registration map at that heading and the constraint of the point of view according to the landmark (Minkowski sum) is represented in Figure 5.14(e). Notice that the intersection is not empty. However if the last constraint of the point of view according to the aperture angle is taken into account, the result is presented in Figure 5.14(f) which is empty. Indeed the heading $\theta = 0°$ does not belong to the possible headings according the aperture angle. The visible points of the landmark seen by the sensor at the two AUV poses (considering a heading at $\theta = 60°$) in Figure 5.14(d) that respect the constraint on the point of view are represented in Figure 5.15.



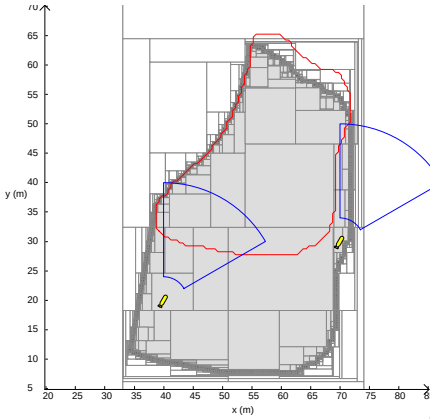
(a) Polar constraint on (x, y) robot position for the point of view. It represents the set $-\mathbb{X}_{polar}$. The angle of detection is 90° and the angular flexibility is 10°.



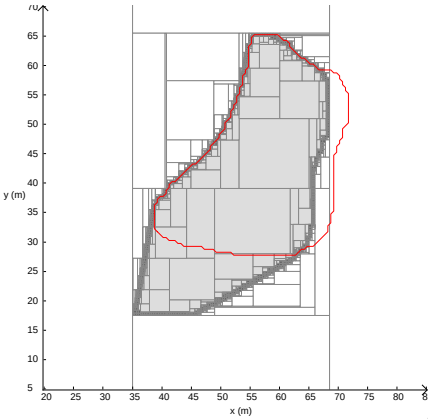
(b) Minkowski sum to compute the (x, y) robot position consistent with the point of view on the shape landmark corresponding to the set $\mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}])$.



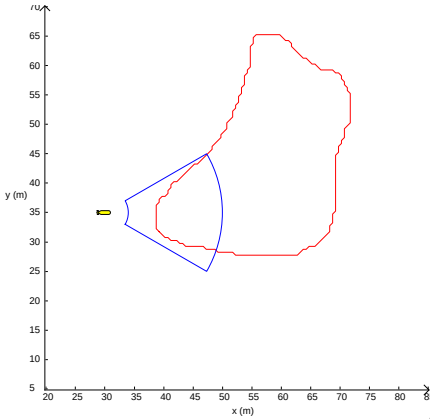
(c) Registration map $\mathbb{P}^{\mathbb{B}}$ at heading $\theta = 60°$.



(d) Registration map at heading $\theta = 60°$ taking into account the constraint on the point of view at 90°. It represents the slice of $\mathbb{P}^{\mathbb{B}}([\theta_{detection}])$ at 60°.



(e) Intersection of the registration map at heading $\theta = 0°$ and the polar constraint at 90°.



(f) Real registration map at heading $\theta = 0°$ taking into account the constraint on the point of view at 90° and the test on the intersection.

Figure 5.14: Registration map with a point of view at 90° on a shape landmark represented by the red contour and an angular flexibility of 10°. Gray boxes are inside the set, white ones outside and dark gray ones no conclusion can be made.

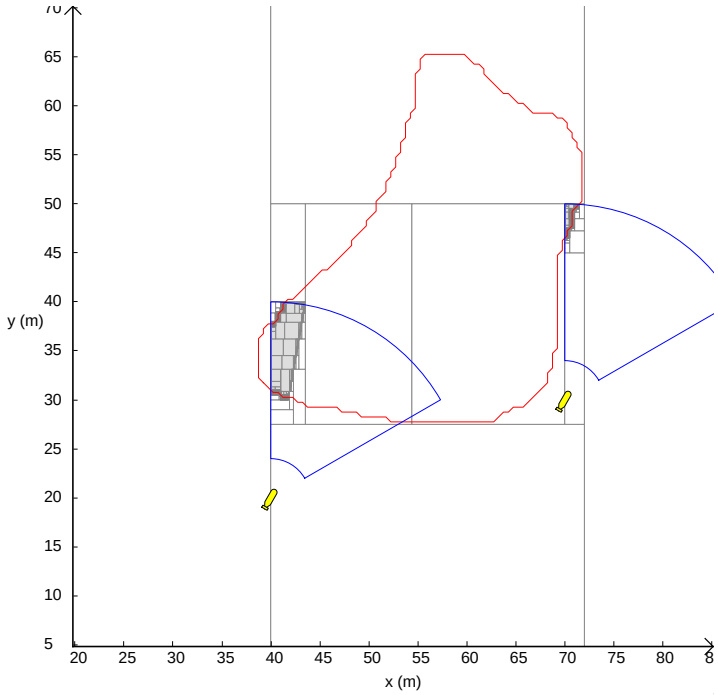


Figure 5.15: Set of visible points from the landmark by the sensor respecting the point of view at the two robots poses in Figure 5.14(d).

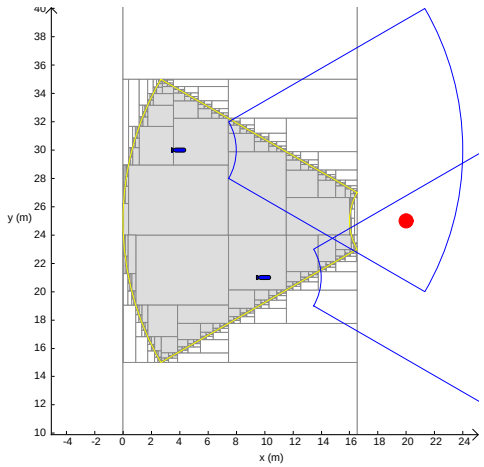
Example 2

Consider now again the punctual landmark represented by a red dot in Figure 5.16 and a point of view at 0° with an angular flexibility of 15°.

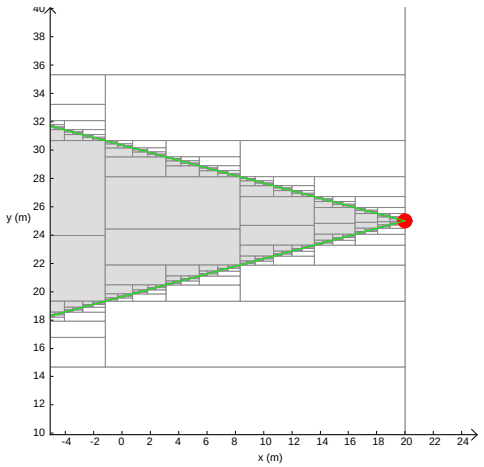
The slice at heading $\theta = 0°$ of the registration map $\mathbb{P}^{\mathbb{B}}$ is represented in Figure 5.16(a). Two AUV poses are depicted with their viability area represented by a blue pie. The landmark set $\mathbb{B}$ is reduced to a singleton $\mathbb{B} = \{\mathbf{b}\}$. The constraint of the point of view according to the landmark can be rewritten as:

$$\mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}]) = \mathbb{P}^{\mathbf{b}}_{view}([\theta_{detection}]) = \{\mathbf{p} \in \mathbb{R}^3 | \exists \theta \in [\theta_{detection}], \exists r \in \mathbb{R}^+, \mathbf{b} - \mathbf{p} = \begin{pmatrix} r \cdot \cos\theta \\ r \cdot \sin\theta \end{pmatrix}\} \tag{5.84}$$

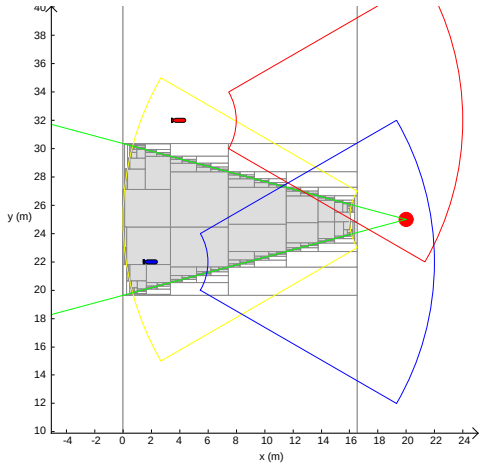
which corresponds to the same constraint as in Equation 4.95 where $\mathbf{m} = \mathbf{b}$. The examples were provided in the previous Chapter. Figure 5.16(b) shows this constraint on (x, y) position according the point of view and the landmark. Green lines are represented to highlight the direction according to the point of view. Finally the registration map $\mathbb{P}^{\mathbb{B}}([\theta_{detection}])$ at heading $\theta = 0°$ is given in Figure 5.16(c) where two AUV poses are represented with their associated visibility area (same colors). The blue AUV pose is able to detect the landmark since all the constraints are satisfied contrary to the red one that does not satisfy $\mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}])$ even if it satisfies $\mathbb{P}^{\theta_3}_{view}([\theta_{detection}])$. The registration map at heading $\theta = 35°$ is represented in Figure 5.16(d) where only a reduced set of (x, y) position are able to detect the landmark at that heading.



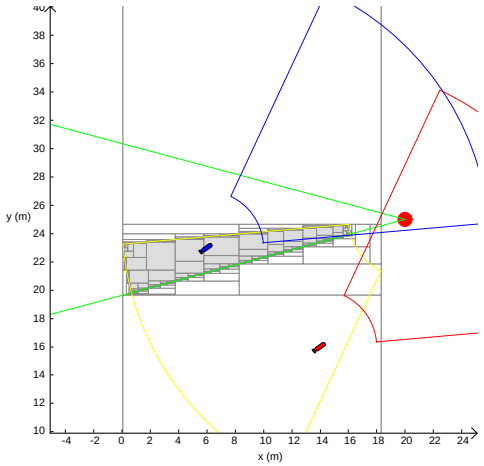
(a) Registration map $\mathbb{P}^{\mathbb{B}}$ at heading $p_3 = \theta = 0°$.



(b) Constraint on (x, y) robot position for the point of view on the landmark corresponding to the set $\mathbb{P}^{\mathbb{B}}_{view}([\theta_{detection}])$. The angle of detection is 0° and the angular flexibility 15°.



(c) Registration map $\mathbb{P}^{\mathbb{B}}([\theta_{detection}])$ at heading $\theta = 0°$ taking into account the constraint on the point of view at 0°. Blue AUV can detect the landmark contrary to the red one.



(d) Registration map $\mathbb{P}^{\mathbb{B}}([\theta_{detection}])$ at heading $\theta = 35°$ taking into account the constraint on the point of view at 0°. Blue AUV can detect the landmark contrary to the red one.

Figure 5.16: Registration map with a point of view at 0° on a punctual landmark represented by a red dot and an angular flexibility of 15°. The green lines depict the point of view and the yellow pies show the set corresponding to the registration map at the defined heading. Gray boxes are inside the set, white ones outside and dark gray ones no conclusion can be made.

In the context of the thesis, it will be assumed that the angle of detection and the angular flexibility is constant, however a landmark may seen at different angles of detection. It would be possible to compute an angular flexibility that depends on the angle of detection of the landmark .

5.2.6 Conclusion

In this subsection, the problem of the registration map is handled with the notion of projection. From any shaped landmark where a separator is available, a separator consistent with the

set of robot configuration able to detect the landmark considering a patch visibility sensor and a transformation function $\mathbf{f}$ can be computed. The transformation function proposed is the composition of a rotation and a translation. This set is a subset of $\mathbb{R}^3$ corresponding to the position (x, y) and heading θ values of the AUV in a 2D environment. If the heading value is defined, the problem of determining the translation parameters may be seen as a Minkowski difference. Due to the notion of aspects or point of view in underwater environment, an additional constraint composed of two constraints is proposed to detect the landmark at a given angle of detection considering an angular flexibility. In this section, the position and/or the shape of the landmark were known accurately. Moreover, the abilities of the sensor were also perfect. In the next section, the uncertainties will be taken into account.

5.3 Registration map with uncertainties

In the previous subsection, the registration map was built assuming no uncertainty on the sensing and on the map. This was illustrated by *Thin Sets* for the map (landmark) and the sensor. In this subsection, the uncertainty on the map and the sensor will be treated as interval of sets, called *Thick Set* or *Interval Shape* [75]. Indeed, an uncertainty on the contour of a landmark is highly possible when using a segmentation algorithm to separate two textures for example. Moreover, if the data are collected by an AUV, the position of these textures may be uncertain due to uncertainty accumulated by the robot.

Firstly, the notion of thick set will be introduced. Then the representation of the uncertainty on the sensing (visibility area) and on the map (landmark) will be presented. Finally the computation of the registration map taking both uncertainties will be provided.

5.3.1 Thick Set

The Thick Set $[\mathbb{X}]$ of $\mathbb{R}^n$ is an extension of the interval analysis [75] such that there exists two subsets [342] of $\mathbb{R}^n$ called the lower bound $\mathbb{X}^-$ and the upper bound $\mathbb{X}^+$ and defined as follows:

$$[\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+] \tag{5.85}$$

$$= \{\mathbb{X} \in \mathcal{P}(\mathbb{R}^n) | \mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+\} \tag{5.86}$$

where $\mathcal{P}(\mathbb{R}^n)$ is a complete lattice with respect to $\subset$ [75]. $(\mathcal{P}(\mathbb{R}^n), \subset)$ denotes the powerset of $\mathbb{R}^n$ with the inclusion $\subset$ as an order of relation. Intervals are now represented in the space of shapes.

A thick set $[\mathbb{X}]$ is represented in Figure 5.17 where $\mathbb{R}^n$ is partitioned into three zones: a clear zone $\mathbb{X}^-$ (red), the penumbra $\mathbb{X}^+ \setminus \mathbb{X}^-$ (orange) and the dark zone $\mathbb{R}^n \setminus \mathbb{X}^+$ (white).

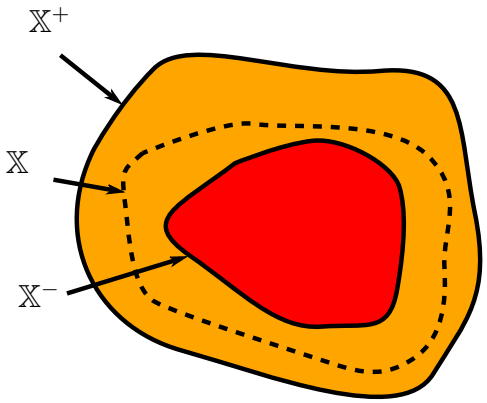


Figure 5.17: Thick set representation.

A thick set $[\mathbb{X}]$ is a sub-lattice of $\mathcal{P}(\mathbb{R}^n, \subset)$ meaning that if $\mathbb{A} \in [\mathbb{X}]$ and $\mathbb{B} \in [\mathbb{X}]$ then $\mathbb{A} \cap \mathbb{B} \in [\mathbb{X}]$ and $\mathbb{A} \cup \mathbb{B} \in [\mathbb{X}]$. As for classical intervals, a set of thick sets of $\mathbb{R}^n$ is denoted $\mathbb{I}\mathcal{P}(\mathbb{R}^n)$. If for a given thick set $[\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+]$, $\mathbb{X}^- = \mathbb{X}^+$ then the thick set $[\mathbb{X}]$ is said to be *thin* and it corresponds to the classical subset of $\mathbb{R}^n$ presented earlier and a singleton in $\mathcal{P}(\mathbb{R}^n)$.

5.3.1.1 Some operators

Extension of classical set operations on the elements of a thick set and operations on intervals of sets can be defined [137] such as:

$$[\mathbb{A}] \cap [\mathbb{B}] = [\mathbb{A}^- \cap \mathbb{B}^-, \mathbb{A}^+ \cap \mathbb{B}^+] \tag{5.87}$$
$$[\mathbb{A}] \sqcap [\mathbb{B}] = [\mathbb{A}^- \cup \mathbb{B}^-, \mathbb{A}^+ \cap \mathbb{B}^+] \tag{5.88}$$
$$[\mathbb{A}] \cup [\mathbb{B}] = [\mathbb{A}^- \cup \mathbb{B}^-, \mathbb{A}^+ \cup \mathbb{B}^+] \tag{5.89}$$
$$[\mathbb{A}] \sqcup [\mathbb{B}] = [\mathbb{A}^- \cap \mathbb{B}^-, \mathbb{A}^+ \cup \mathbb{B}^+] \tag{5.90}$$
$$[\mathbb{A}] \times [\mathbb{B}] = [\mathbb{A}^- \times \mathbb{B}^-, \mathbb{A}^+ \times \mathbb{B}^+] \tag{5.91}$$
$$[\mathbb{A}] \setminus [\mathbb{B}] = [\mathbb{A}^- \setminus \mathbb{B}^+, \mathbb{A}^+ \setminus \mathbb{B}^-] \tag{5.92}$$
$$[\mathbb{A}] \oplus [\mathbb{B}] = [\mathbb{A}^- \oplus \mathbb{B}^-, \mathbb{A}^+ \oplus \mathbb{B}^+] \tag{5.93}$$

where $[\mathbb{A}] = [\mathbb{A}^-, \mathbb{A}^+]$ and $[\mathbb{B}] = [\mathbb{B}^-, \mathbb{B}^+]$.

If $\mathbf{f}$ is a function from $\mathbb{R}^n$ to $\mathbb{R}^m$, the image of a thick set $\mathbb{A}$ by $\mathbf{f}$ is:

$$\mathbf{f}([\mathbb{A}]) = \left[\mathbf{f}(\mathbb{A}^-), \mathbf{f}(\mathbb{A}^+)\right] \tag{5.94}$$

Some examples will be provided when thick separators will be introduced.

5.3.1.2 Thick separator

Previously separators were associated to a paver to characterize a thin set. Now, the sets are uncertainly defined showing the presence of a penumbra which has a non-empty volume. To avoid many bisections in the penumbra, a thick separator has been developed inspired by the construction of a separator based on a pair of two complementary contractors.

Definition

A *thick separator* $[\mathcal{S}]$ for a thick set $[\mathbb{X}]$ is now a 3−uple of contractors $\{\mathcal{S}^{in}, \mathcal{S}^{?}, \mathcal{S}^{out}\}$, as depicted in Figure 5.18 such that for all $[\mathbf{x}] \in \mathcal{I}\mathbb{R}^n$:

$$\mathcal{S}^{in}([\mathbf{x}]) \cap \mathbb{X}^{in} = [\mathbf{x}] \cap \mathbb{X}^{in} \tag{5.95}$$
$$\mathcal{S}^{?}([\mathbf{x}]) \cap \mathbb{X}^{?} = [\mathbf{x}] \cap \mathbb{X}^{?} \tag{5.96}$$
$$\mathcal{S}^{out}([\mathbf{x}]) \cap \mathbb{X}^{out} = [\mathbf{x}] \cap \mathbb{X}^{out} \tag{5.97}$$

Separators were built from a pair of two complementary contractors $\{\mathcal{C}, \overline{\mathcal{C}}\}$, thick separators are built from a pair of separators. Considering a separator $\mathcal{S}^{-} = \{\mathcal{S}^{-}_{in}, \mathcal{S}^{-}_{out}\}$ consistent with $\mathbb{X}^{-}$ and $\mathcal{S}^{+} = \{\mathcal{S}^{+}_{in}, \mathcal{S}^{+}_{out}\}$ consistent with $\mathbb{X}^{+}$, the thick separator $[\mathcal{S}]$ consistent with $[\mathbb{X}]$ is defined as follows:

$$[\mathcal{S}] = \{\mathcal{S}^{-}_{out}, \mathcal{S}^{-}_{in} \cap \mathcal{S}^{+}_{out}, \mathcal{S}^{+}_{in}\} \tag{5.98}$$

Proof

This comes directly from the definition of $\mathcal{S}^{-}$ and $\mathcal{S}^{+}$:

$$\mathcal{S}^{-}_{out} \sim \mathbb{X}^{-} = \mathbb{X}^{in} \tag{5.99}$$
$$\mathcal{S}^{+}_{in} \sim \overline{\mathbb{X}^{+}} = \mathbb{X}^{out} \tag{5.100}$$
$$\mathcal{S}^{-}_{in} \cap \mathcal{S}^{+}_{out} \sim \overline{\mathbb{X}^{-}} \cap \mathbb{X}^{+} = \mathbb{X}^{?} \tag{5.101}$$

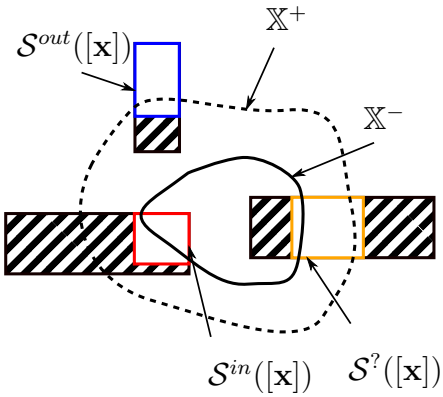


Figure 5.18: Thick separator. Contractions for three boxes $[\mathbf{x}]$.

Algebra for thick separators [75] can be defined similarly as for contractors [47] and separators [140].

Moreover, an image thick separator can be defined by a ternary image using the image separator introduced and presented in subsection 5.3.3 which deals with map uncertainties.

Example

Consider a thick set $[\mathbb{A}]$ defined by $\mathbb{A}^{-}$ and $\mathbb{A}^{+}$ such that:

$$\mathbb{A}^{-} = \{(x,y) \in \mathbb{R}^2 | y - 2x - 1 \leq 0, y - 0.5x + 1 \geq 0, y + 0.5x - 1 \leq 0\} \tag{5.102}$$
$$\mathbb{A}^{+} = \{(x,y) \in \mathbb{R}^2 | y - 2x - 4 \leq 0, y - 0.5x + 2 \geq 0, y + 0.5x - 2 \leq 0\} \tag{5.103}$$

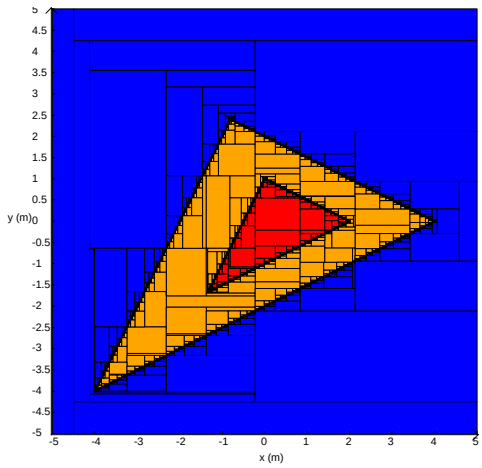
where separators are built for each sets. The result of the thick separator using a paving is depicted in Figure 5.19(a). Similarly consider a second thick set $[\mathbb{B}]$ defined by $\mathbb{B}^-$ and $\mathbb{B}^+$ such that:

$$\mathbb{B}^- = \{(x, y) \in \mathbb{R}^2 | \sqrt{x^2 + y^2} - 1 \leq 0\} \tag{5.104}$$

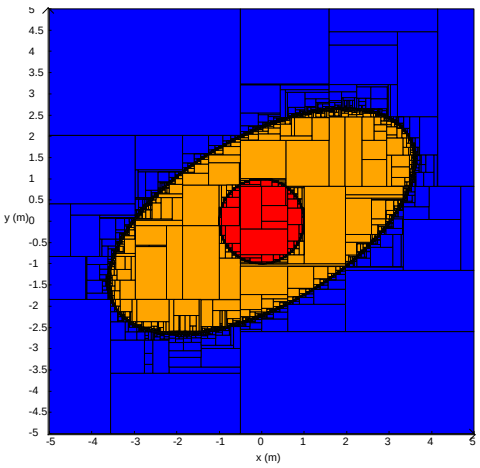
and an ellipse set corresponding to:

$$\mathbb{E} = \{(x, y) \in \mathbb{R}^2 | (\frac{x}{4})^2 + (\frac{y}{2})^2 - 1 \leq 0\} \tag{5.105}$$

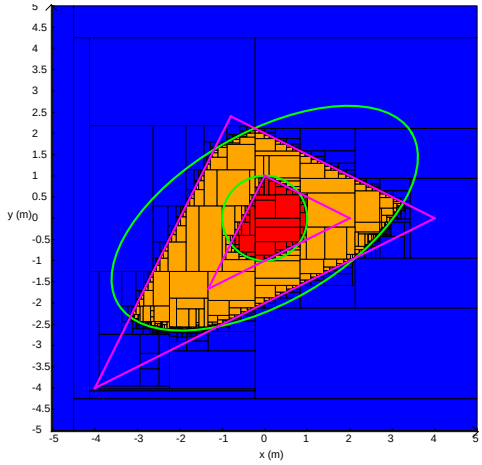
The set $\mathbb{B}^+$ corresponds to the rotation of 30° of the set $\mathbb{E}$. The result of this thick set $[\mathbb{B}]$ is depicted in Figure 5.19(b). The different unions and intersections are shown in the following subfigures where the contours of the initial sets are painted in magenta for the thick set $[\mathbb{A}]$ and in green for $[\mathbb{B}]$.



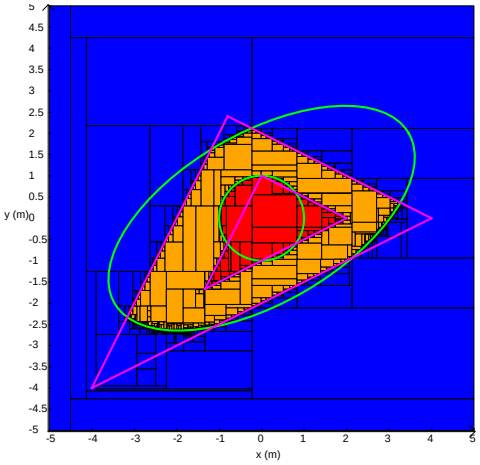
(a) Thick set $[\mathbb{A}]$.



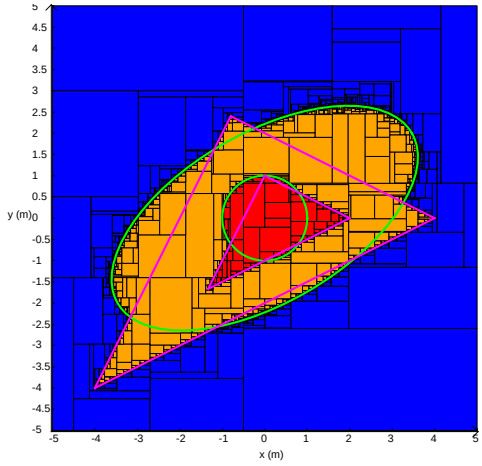
(b) Thick set $[\mathbb{B}]$.



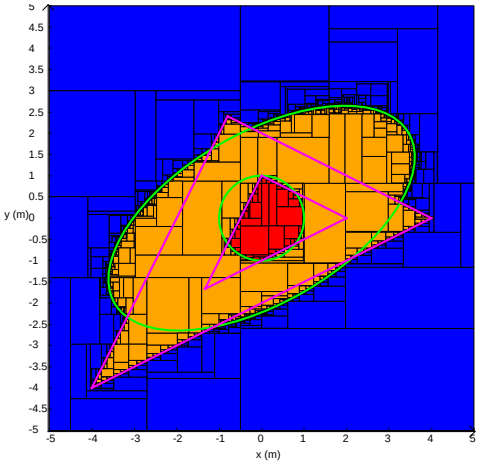
(c) $[\mathbb{A}] \cap [\mathbb{B}]$.



(d) $[\mathbb{A}] \sqcap [\mathbb{B}]$.



(e) $[\mathbb{A}] \cup [\mathbb{B}]$.



(f) $[\mathbb{A}] \sqcup [\mathbb{B}]$.

Figure 5.19: Both intersection and union on thick sets. Blue boxes are outside, orange boxes are in the penumbra, red ones are inside and yellow ones no conclusion can be made. Magenta and green lines depict the contours of $[\mathbb{A}]$ and $[\mathbb{B}]$ respectively.

5.3.1.3 Uncertain set inversion

For a given function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ and a set $\mathbb{Y} \subset \mathbb{R}^m$, the classical set inversion aims to bracket from inside and outside the set $\mathbb{X}$ such that:

$$\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y}) \tag{5.106}$$

It was used presented in Chapter 4.

However, when the function $\mathbf{f}$ and the set $\mathbb{Y}$ are uncertain, the problem cannot be solved directly. The set inversion problem can be written as [71]

$$\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y}) \text{ with } \mathbf{f} \in \mathbb{F} \text{ and } \mathbb{Y} \in [\mathbb{Y}] \tag{5.107}$$

where $[\mathbb{Y}]$ is a thick set and $\mathbb{F}$ is a set of functions.

A set $\mathbb{X}$ is said to be a *feasible solution* if:

$$\exists \mathbf{f} \in \mathbb{F}, \exists \mathbb{Y} \in [\mathbb{Y}], \mathbb{X} = \mathbf{f}^{-1}(\mathbb{X}) \tag{5.108}$$

However the set of all feasible solutions is not generally a thick set.

Set inversion theorem

The theorem is given in [71] and is reminded here:

Theorem 5.3.1 *Given a set of function $\mathbb{F} \subset \mathcal{F}(\mathbb{R}^n, \mathbb{R}^m)$ and a thick set $[\mathbb{Y}] = [\mathbb{Y}^-, \mathbb{Y}^+]$, the smallest thick set that encloses all sets $\mathbb{X}$ such that:*

$$\exists \mathbf{f} \in \mathbb{F}, \exists \mathbb{Y} \in [\mathbb{Y}], \mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y}) \tag{5.109}$$

is the thick set $[\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+]$ with:

$$\mathbb{X}^- = \bigcap_{\mathbf{f} \in \mathbb{F}} \mathbf{f}^{-1}(\mathbb{Y}^-) \tag{5.110}$$

$$\mathbb{X}^+ = \bigcup_{\mathbf{f} \in \mathbb{F}} \mathbf{f}^{-1}(\mathbb{Y}^+) \tag{5.111}$$

The proof can be found in [71].

This theorem provides the exact formulation of the thick set inversion problem by enclosing the set $\mathbb{X}$ between $\mathbb{X}^-$ and $\mathbb{X}^+$. However, the difficulty is to get an inner approximation of the penumbra and existing interval methods accumulate inside the penumbra and spend most of the computation time to test small box that are actually inside this penumbra.

When the set of functions $\mathbb{F}$ is composed of a single one function $\mathbf{f}$, the smallest thick set according to Equation 5.94 is:

$$[\mathbb{X}] = \left[\mathbf{f}^{-1}(\mathbb{Y}^-), \mathbf{f}^{-1}\mathbb{Y}^+\right] \tag{5.112}$$

where classical set inversion algorithms can be applied for $\mathbb{X}^-$ and $\mathbb{X}^+$. Some examples are provided in the following subsections dealing with the registration maps.

When the set of functions $\mathbb{F}$ depends on a parameter, the projection algorithm presented in Chapter 4 can be used to remove the quantified parameters. However, bissections are needed in the parameter space which can be costly when the dimension of the parameters is high. If the set of functions can be described by an interval of functions, a *Thick Function* can be used. The reader may refer to [75][71] for more details on the thick functions.

Set inversion with parametric function

The function $\mathbf{f}$ depends now on a parameter $\mathbf{p} \in [\mathbf{p}] \subset \mathbb{R}^p$ and is written $\mathbf{f}(\mathbf{x}, \mathbf{p})$. It will be denoted as $\mathbf{f_p}(\mathbf{x})$ for simplicity. The set inversion problem is formulated as follows:

$$[\mathbb{X}] = \mathbf{f}^{-1}_{[\mathbf{p}]}([\mathbb{Y}]) \tag{5.113}$$

From **Theorem** 5.3.1, the smallest thick set $[\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+]$ is defined by [71]:

$$\mathbb{X}^- = \bigcap_{\mathbf{p}\in[\mathbf{p}]} \mathbf{f}_\mathbf{p}^{-1}(\mathbb{Y}^-) = \{\mathbf{x} | \forall \mathbf{p} \in [\mathbf{p}], \mathbf{f_p}(\mathbf{x}) \in \mathbb{Y}^-\} \tag{5.114}$$

$$\mathbb{X}^+ = \bigcup_{\mathbf{p}\in[\mathbf{p}]} \mathbf{f}_\mathbf{p}^{-1}(\mathbb{Y}^+) = \{\mathbf{x} | \exists \mathbf{p} \in [\mathbf{p}], \mathbf{f_p}(\mathbf{x}) \in \mathbb{Y}^+\} \tag{5.115}$$

Using the projection algorithm presented in Chapter 4, it is possible to give an expression of $\mathbb{X}^-$ and $\mathbb{X}^+$ as a projection. Indeed, according to the definition of the projection in Chapter 4, the set $\mathbb{X}^+$ is given by:

$$\mathbb{X}^+ = proj_{\mathbb{D}_\mathbb{X}}(\mathbf{f}_\mathbf{p}^{-1}(\mathbb{Y}^+)) \tag{5.116}$$

where $\mathbb{D}_\mathbb{X}$ denotes the domain of $\mathbb{X}$. It will written for ease of reading $\mathbb{X}$ directly that gives:

$$\mathbb{X}^+ = proj_{\mathbb{X}}(\mathbf{f}_\mathbf{p}^{-1}(\mathbb{Y}^+)) \tag{5.117}$$

The definition of $\mathbb{X}^-$ is given by:

$$\mathbb{X}^- = \{\mathbf{x} | \forall \mathbf{p} \in [\mathbf{p}], \mathbf{f_p}(\mathbf{x}) \in \mathbb{Y}^-\} \tag{5.118}$$

$$= \overline{\{\mathbf{x} | \exists \mathbf{p} \in [\mathbf{p}], \mathbf{f_p}(\mathbf{x}) \notin \mathbb{Y}^-\}} \tag{5.119}$$

$$= \overline{\{\mathbf{x} | \exists \mathbf{p} \in [\mathbf{p}], \mathbf{f_p}(\mathbf{x}) \in \overline{\mathbb{Y}^-}\}} \tag{5.120}$$

$$= \overline{proj_{\mathbb{X}}(\mathbf{f}_\mathbf{p}^{-1}(\overline{\mathbb{Y}^-}))} \tag{5.121}$$

If separators for $\mathbb{Y}^-$ and $\mathbb{Y}^+$ are available, then a separator $\mathcal{S}^-$ consistent with $\mathbb{X}^-$ and a separator $\mathcal{S}^+$ consistent with $\mathbb{X}^+$ can be built to get a thick representation of the solution set using these projections. This will be particularly used in the next Chapter for the motion planning.

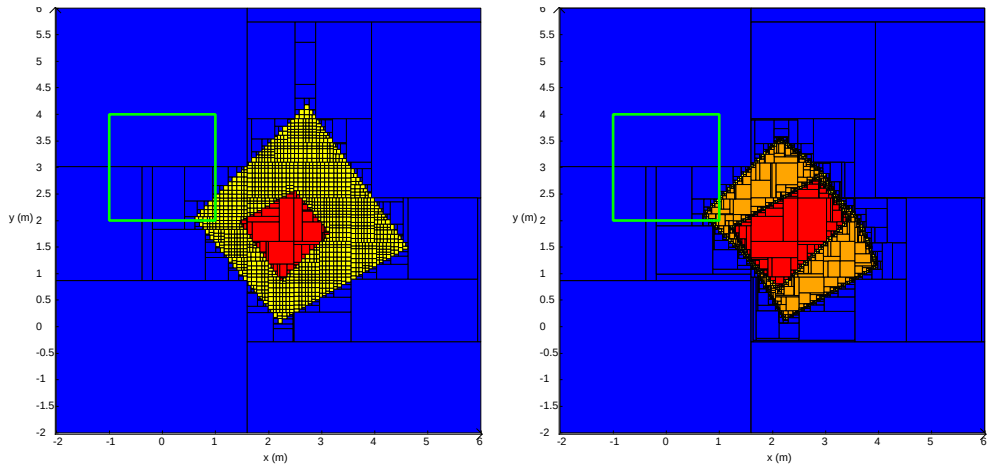
Example inspired in [71]

Consider the set $\mathbb{Y}$ defined by $\mathbb{Y}^- = \mathbb{Y}^+ = [\mathbf{y}] = [-1, 1] \times [2, 4]$ depicted by the green square in Figure 5.20 and the parametric function as the rotation of an angle $\theta \in [\frac{\pi}{4}, \frac{\pi}{3}]$. The inversion problem is expressed as follows:

$$[\mathbb{X}] = \mathbf{f}_{[\theta]}([\mathbf{y}]) \tag{5.122}$$

The results are depicted in Figure 5.20. The classical set inversion 5.20(a) accumulates in the penumbra with small yellow boxes since the algorithm cannot determine if after the uncertain rotation it will inside the green square. The set inversion with the projection algorithm giving $\mathbb{X}^-$ and $\mathbb{X}^+$ according to Equation 5.114 and Equation 5.115 respectively provides the thick set

represented in Figure 5.20(b). The red boxes are proved to be in the green square after the uncertain rotation. Notice that it does not accumulate in the penumbra (orange).



(a) Classical set inversion. Blue boxes are outside, red ones are inside and yellow ones no conclusion can be made.

(b) Set inversion with projection. Blue boxes are outside, orange ones are in the penumbra, red ones are inside and yellow ones no conclusion can be made.

Figure 5.20: Set inversion to determine the set $\mathbb{X}$ of points that ends in the green square after an uncertain rotation of angle $\theta \in [\theta] = [\frac{\pi}{4}, \frac{\pi}{3}]$.

5.3.2 Visibility area with uncertainties

The parameters of the sensor are not known accurately such as the minimal and the maximal range. As presented in section 5.2, the visibility of the sensor is assumed to be a patch with a limited FOV. The visibility area defined in Equation 5.1 is now ill-defined and depends on a thick interval vector $[\mathbf{y}]$ [48]. In the general case where the sensor has a limited range and aperture angle, assuming an uncertainty δR on the range and $\delta\theta_3$ on the bearing, the thick set is expressed as follows:

$$[\mathbf{y}] = [R^-, R^+] \times [\theta_3^-, \theta_3^+] \tag{5.123}$$

where $R^- = [R_{min} - \delta R, R_{min} + \delta R]$, $R^+ = [R_{max} - \delta R, R_{max} + \delta R]$, $\theta_3^- = [-\theta_3 - \delta\theta_3, -\theta_3 + \delta\theta_3]$ and $\theta_3^+ = [\theta_3 - \delta\theta_3, \theta_3 + \delta\theta_3]$ are intervals containing respectively the lower bound of the range, the upper bound of the range, the lower bound of the aperture angle and the upper bound of the aperture angle.

The visibility area $\mathbb{V}(\mathbf{x})$ at the pose $\mathbf{x}$ is now a thick set $[\mathbb{V}(\mathbf{x})]$ which will be denoted $\mathbb{V}(\mathbf{x})$. It can be defined by two sets $\mathbb{V}^-(\mathbf{x})$ and $\mathbb{V}^+(\mathbf{x})$ such that $\mathbb{V}^-(\mathbf{x}) \subset \mathbb{V}(\mathbf{x}) \subset \mathbb{V}^+(\mathbf{x})$ with:

$$\mathbb{V}^-(\mathbf{x}) = \{\mathbf{z} \in \mathbb{R}^2 | \exists r \in [ub(R^-), lb(R^+)], \exists \beta \in [ub(\theta_3^-), lb(\theta_3^+)], \mathbf{z} - \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} r \cdot \cos(x_3 + \beta) \\ r \cdot \sin(x_3 + \beta) \end{pmatrix}\} \tag{5.124}$$

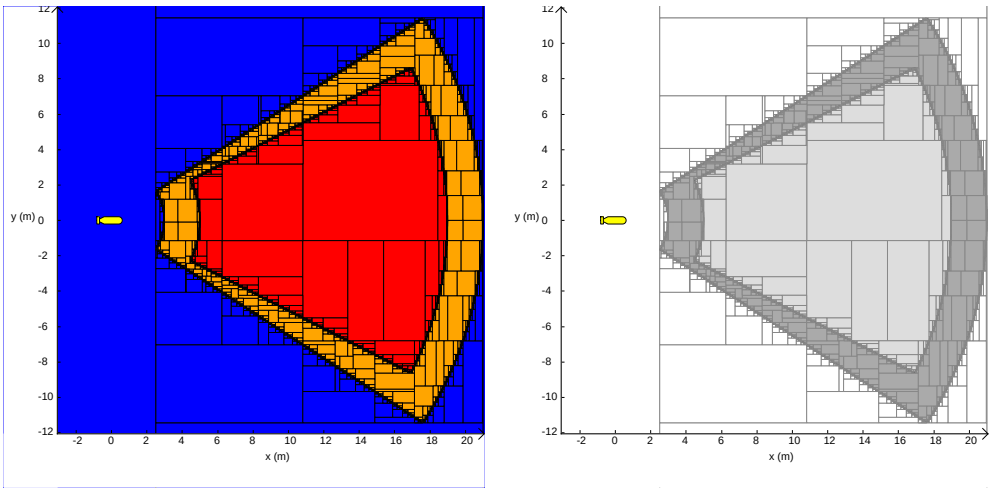
and

$$\mathbb{V}^+(\mathbf{x}) = \{\mathbf{z} \in \mathbb{R}^2 | \exists r \in [lb(R^-), ub(R^+)], \exists \beta \in [lb(\theta_3^-), ub(\theta_3^+)], \mathbf{z} - \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} r \cdot \cos(x_3 + \beta) \\ r \cdot \sin(x_3 + \beta) \end{pmatrix}\} \tag{5.125}$$

where $lb()$ and $ub()$ are the lower and upper bounds of the corresponding intervals. In the proposed sets, it is assumed that $R_{min} + \delta R < R_{max} - \delta R$ and $-\theta_3 + \delta\theta_3 < \theta_3 - \delta\theta_3$. In other terms, it is assumed that $2\delta R < (R_{max} - R_{min})$ and $\delta\theta_3 < \theta_3$.

Example

Consider again the forward looking sensor with $R_{min} = 4m$, $R_{max} = 20m$ and $\theta_3 = 30°$ and assume that $\delta R = 1m$ and $\delta\theta_3 = 3°$. Values are overrated on purpose to see the penumbra. Building separators for $\mathbb{V}^-(\mathbf{x})$ and $\mathbb{V}^+(\mathbf{x})$, a thick separator can provide the set $\mathbb{V}(\mathbf{x})$ with a paver. The visibility area is represented in Figure 5.21 at the particular pose $(0, 0, 0°)$.



(a) Color representation. Blue boxes are outside, orange ones are in the penumbra, red ones are inside and yellow ones no conclusion can be made.

(b) Grayscale representation. White boxes are outside, dark gray ones are in the penumbra, soft gray ones are inside and black ones no conclusion can be made (they are very little).

Figure 5.21: Thick set representation of the visibility area taking into account the uncertainties in range and bearing at the particular pose $\mathbf{x} = (0, 0, 0°)$. Different color representations.

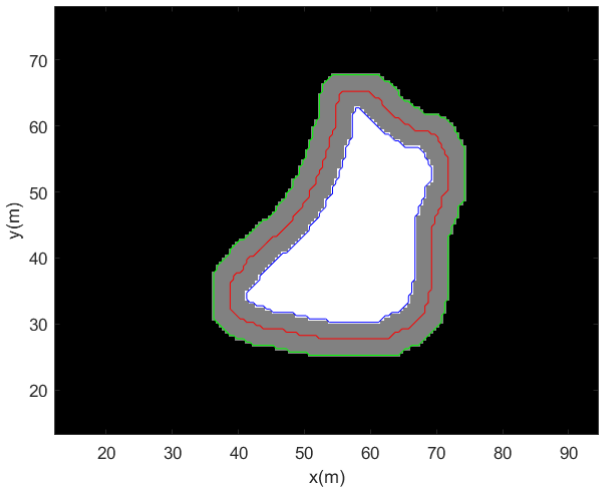
5.3.3 Map with uncertainties

Uncertainties on the map refer to an ill-defined map [168] coming from the uncertainties of the survey platform, i.e. AUV, from the sensing measurements and/or from the detection process (automatically or manually). The contour of a region landmark may be difficult to determine or the position of a punctual landmark may be uncertain and contained in a box or an uncertain ellipse [309].

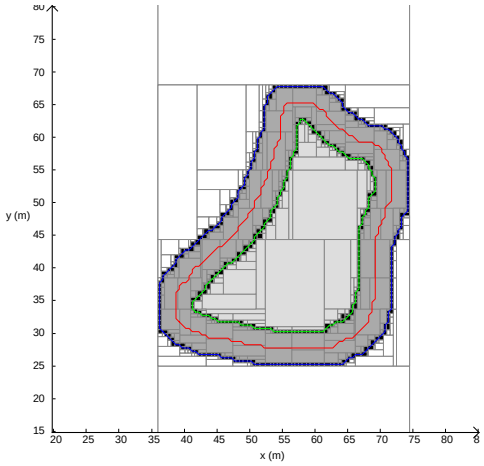
Example

Consider again the shape landmark presented in Figure 4.13(a) and used for the registration map in Figures 5.6 and 5.7. This corresponds to the true landmark. Due to uncertainties in navigation, processing, etc... the contour of this landmark is not known accurately. This landmark is then a thick set $[\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+]$. Consider a disk of radius $3m$ as structuring element for the computation of the erosion and dilation of the initial image. The dilation provides the set $\mathbb{X}^+$ depicted by the green contour in Figure 5.22(a) and the erosion the set $\mathbb{X}^-$ depicted by the blue contour in Figure 5.22(a). Using the thick separator associated to a paver, the enclosure

of $\mathbb{X}$ is provided in Figure 5.22(b) where the grayscale convention is used. The dark gray region corresponds to the penumbra (thick contour). The true contour of the landmark lays in the penumbra. Due to the uncertain position of AUV when capturing images underwater, the thick set representation enables to provide an uncertain representation of the shape observed such as sand ripples region.



(a) Grayscale image where the white part corresponds to $\mathbb{X}^-$ and the (white+gray) part corresponds to $\mathbb{X}^+$. Red lines depicts the contour of the shape presented in Figure 4.13(a). Green lines is the contour of the dilation with a disk of radius $3m$ and the blue lines is the erosion with the same element.



(b) Thick set representation in grayscale convention. Contours are drawn.

Figure 5.22: Thick set $[\mathbb{X}]$ from a grayscale image.

5.3.4 Registration map with uncertainties

The constraint for the registration map was defined in Equation 5.9 and is reminded here:

$$\mathbb{P} = \{\mathbf{p} \in \mathbb{R}^p | \mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} \neq \emptyset\} \tag{5.126}$$

where $\mathbb{A}$ and $\mathbb{B}$ correspond respectively to the visibility area $\mathbb{V}$ and the landmark. Earlier these sets were clearly defined and represented by *thin* sets. Now these sets are uncertain (thick set) and the idea is to develop a thick separator to characterize the set of feasible parameters $\mathbf{p}$ associated to this constraint. A thick set $\mathbb{A}$ which was initially written as $[\mathbb{A}]$ is denoted $\mathbb{A}$ for simplicity reasons.

Proposition

Given a function $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$, $\mathbb{A} \in [\mathbb{A}]$ and $\mathbb{B} \in [\mathbb{B}]$, the set $\mathbb{P}$ corresponding to the parameters $\mathbf{p}$ can be enclosed between two sets $\mathbb{P}^-$ and $\mathbb{P}^+$ such that $\mathbb{P} \in [\mathbb{P}^-, \mathbb{P}^+]$ with:

$$\mathbb{P}^- = \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}^-, \mathbf{p}) \cap \mathbb{B}^- \neq \emptyset\} \tag{5.127}$$
$$\mathbb{P}^+ = \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}^+, \mathbf{p}) \cap \mathbb{B}^+ \neq \emptyset\} \tag{5.128}$$

Proof:

$$\{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} \neq \emptyset\} = \{\mathbf{p}, \exists \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}\} \quad (5.129)$$

$$\overset{\mathbb{B} \subset \mathbb{B}^+}{\subset} \{\mathbf{p}, \exists \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}^+\} \quad (5.130)$$

$$\overset{\mathbb{A} \subset \mathbb{A}^+}{\subset} \{\mathbf{p}, \exists \mathbf{a} \in \mathbb{A}^+, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}^+\} \quad (5.131)$$

and

$$\{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} \neq \emptyset\} = \{\mathbf{p}, \exists \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}\} \quad (5.132)$$

$$\overset{\mathbb{B}^- \subset \mathbb{B}}{\supset} \{\mathbf{p}, \exists \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}^-\} \quad (5.133)$$

$$\overset{\mathbb{A}^- \subset \mathbb{A}}{\supset} \{\mathbf{p}, \exists \mathbf{a} \in \mathbb{A}^-, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}^-\} \quad (5.134)$$

Therefore the enclosure of $\mathbb{P}$ is:

$$\mathbb{P}^- = \{\mathbf{p}, \exists \mathbf{a} \in \mathbb{A}^-, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}^-\} = \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}^-, \mathbf{p}) \cap \mathbb{B}^- \neq \emptyset\} \quad (5.135)$$

$$\mathbb{P}^+ = \{\mathbf{p}, \exists \mathbf{a} \in \mathbb{A}^+, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}^+\} = \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}^+, \mathbf{p}) \cap \mathbb{B}^+ \neq \emptyset\} \quad (5.136)$$

According to the definition of the registration map in Equation 5.14, this leads to the enclosure:

$$\mathbb{P}^- = proj_{\mathbf{p}}((\mathbb{A}^- \times \mathbb{R}^p) \cap \mathbf{f}^{-1}(\mathbb{B}^-)) \quad (5.137)$$

$$\mathbb{P}^+ = proj_{\mathbf{p}}((\mathbb{A}^+ \times \mathbb{R}^p) \cap \mathbf{f}^{-1}(\mathbb{B}^+)) \quad (5.138)$$

which corresponds to two registration map problems using classical set inversion algorithms in order to have a thick set representation.

Example

Consider the thick set defined by a circle and an ellipse in the **Example** of the subsection 5.3.1.2 presenting some operators on thick set. This thick set will be the landmark $\mathbb{B} = [\mathbb{B}^-, \mathbb{B}^+]$ and is presented in Figure 5.19(b). Consider again the sensor presented in the example of the subsection 5.3.2 dealing with sensor uncertainties on the visibility area depicted in Figure 5.21(a). Some slices of the registration map according to the definition in Equations 5.127 and 5.128 are depicted in Figure 5.23. The thick landmark is represented by a circle for $\mathbb{B}^-$ in green and an ellipse for $\mathbb{B}^+$ in magenta. Some slices at defined heading, x or y values are proposed with the 2D corresponding representation. The colored AUVs or dots depicted in the solution set on the left corresponding to some AUV poses are represented on the right in the (x, y) plane with the associated color. Moreover the thick visibility area defined by $\mathbb{V}^-$ and $\mathbb{V}^+$ are represented by pies with the same color of the AUV pose. Blue corresponds to poses that can detect the landmark, red corresponds to poses that may detect the landmark and green refers to poses that are not able to detect the landmark.

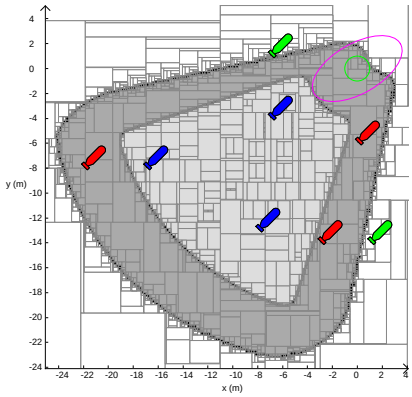
Firstly, the slice of the solution set is presented in 5.23(a) with the heading of the AUV defined at 45°. Notice that the green poses that are outside the solution set are indeed never intersecting the thick landmark $\mathbb{B}$. The red AUVs inside the penumbra of the solution set correspond to poses that guarantee $\mathbb{V}^+$ intersects $\mathbb{B}^+$ but do not guarantee that $\mathbb{V}^-$ will intersect $\mathbb{B}^-$ and therefore detect the landmark. The landmark may be detected at these poses. On the contrary, the blue AUVs correspond to robot poses that can detect for sure the landmark

despite the uncertainty on the landmark and on the visibility area due to the fact that $\mathbb{V}^-$ intersects $\mathbb{B}^-$. This reasoning is the same for all the slices and 2D representation. The contours of the boxes are kept in this first solution set, however in the two following representation, the contours have been removed for ease of visualization.

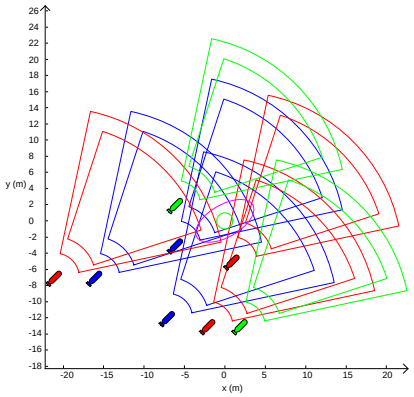
Secondly, the slice in the $(y - \theta)$ plane with $x = -10m$ is presented in Figure 5.23(c) with the associated 2D representation in Figure 5.23(d).

Finally, the slice in the $(x - \theta)$ plane with $y = -15m$ is presented in Figure 5.23(e) with the associated 2D representation in Figure 5.23(f).

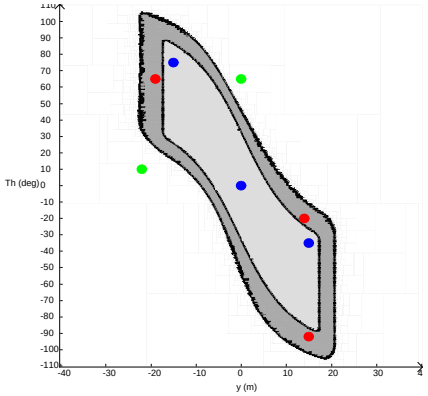
Notice that the dimension of the axis are not the same in the solution set.



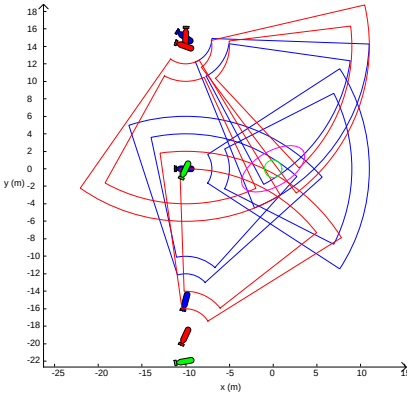
(a) Registration map at heading 45°. The contours of the thick landmark are represented in green for $\mathbb{B}^-$ and in magenta for $\mathbb{B}^+$. Some AUV poses are choiced for a 2D representation.



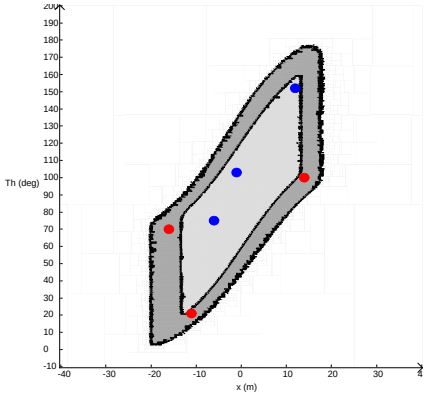
(b) 2D representation for the associated colored poses in the figure (a).



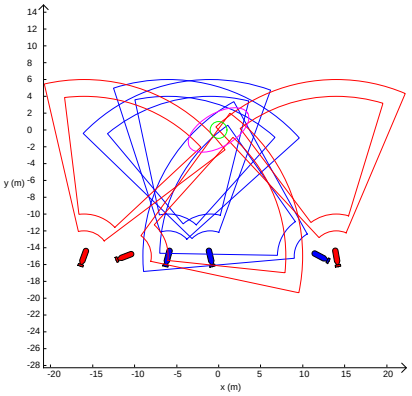
(c) Registration map in the $(y-\theta)$ plane with $x=-10m$. Some poses are defined by dots.



(d) 2D representation of the selected poses with the color associated in the figure (c).



(e) Registration map in the $(x-\theta)$ plane with $y=-15m$.



(f) Same definition for the 2D representation as in (d) where it corresponds to poses in the figure (e).

Figure 5.23: Registration maps at different slices and 2D associated representation. Grayscale convention for the solution set. The contours of the boxes are very thin for ease of interpretation in (c) and (e).

Registration map for an ill-geolocalized punctual landmark

When dealing with an uncertain position of a punctual landmark located in a circle, an ellipse or any shape, it is still possible to compute the registration map. The landmark $\mathbb{B}$ is defined by $\mathbb{B}^- = \{\mathbf{b}\}$ (a singleton) and $\mathbb{B}^+$ that may correspond to the circle, the ellipse or the shape containing for sure the punctual landmark (or at a certain level such as 90%) where $\mathbf{b}$ corresponds to the most likely position of the landmark. The registration map is then computable as before according to Equations 5.127 and 5.128 where $\mathbb{B}^- = \{\mathbf{b}\}$. The first equation giving $\mathbb{P}^-$ can be computed according to the method presented for the punctual landmark in the subsection 5.2.2. The set $\mathbb{P}^+$ can still be computed according to $\mathbb{B}^+$. However the true position of the landmark may be anywhere in the shape containing it. The Proposition defining the enclosure of the set $\mathbb{P}$ has to be modified. Indeed the position of the punctual landmark $\{\mathbf{b}\} = \mathbf{b}$ may be anywhere in $\mathbb{B}^+$, this leads to a new definition of $\mathbb{P}^-$:

$$\mathbb{P}^- = \{\mathbf{p} \in \mathbb{R}^p | \forall \mathbf{b} \in \mathbb{B}^+, \mathbf{b} \in \mathbf{f}(\mathbb{A}, \mathbf{p})\} \tag{5.139}$$

which can be rewritten as:

$$\mathbb{P}^- = \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}, \mathbf{p}) \supset \mathbb{B}^+\} \tag{5.140}$$

The set $\mathbb{P}^-$ could be represented as a thick set where the two set $\mathbb{P}^-_m$ and $\mathbb{P}^-_M$ are defined as follows:

$$\mathbb{P}^-_m = \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}^-, \mathbf{p}) \supset \mathbb{B}^+\} \tag{5.141}$$
$$\mathbb{P}^-_M = \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}^+, \mathbf{p}) \supset \mathbb{B}^+\} \tag{5.142}$$

with $\mathbb{P}^- \in [\mathbb{P}^-_m, \mathbb{P}^-_M]$. This comes from the fact that $\mathbb{A}^- \subset \mathbb{A} \subset \mathbb{A}^+$ and a similar proof can be made as earlier. However if you consider $\mathbb{P}^-_M$ it is possible that the true landmark position lays in the penumbra of the visibility area for some particular poses $\mathbf{p}$ and then the sensor won't guarantee the detection. Therefore the set $\mathbb{P}^-$ is defined as follows:

$$\mathbb{P}^- = \mathbb{P}^-_m = \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}^-, \mathbf{p}) \supset \mathbb{B}^+\} \tag{5.143}$$

which is a thin set. Finally the set $\mathbb{P}$ can be enclosed by $\mathbb{P}^-$ given by Equation 5.143 and by $\mathbb{P}^+$ given by Equation 5.128. The set $\mathbb{P}^-$ can be interpreted as the robot poses where the whole uncertain position of the landmark defined by $\mathbb{B} = [\mathbb{B}^-, \mathbb{B}^+]$ will be in the certainly visible FOV $\mathbb{A}^- = \mathbb{V}^-$ of the sensor. It is possible that $\mathbb{P}^-$ is reduced to an empty set when the uncertain position of the landmark is too big compared to the visibility area of the sensor.

The set $\mathbb{P}^-$ can be rewritten as:

$$\begin{aligned}
\mathbb{P}^- &= \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}^-, \mathbf{p}) \supset \mathbb{B}^+\} && (5.144)\\
&= \{\mathbf{p} \in \mathbb{R}^p | \mathbb{B}^+ \subset \mathbf{f}(\mathbb{A}^-, \mathbf{p})\} && (5.145)\\
&= \{\mathbf{p} \in \mathbb{R}^p | \forall \mathbf{b} \in \mathbb{B}^+, \exists \mathbf{a} \in \mathbb{A}^-, \mathbf{b} = \mathbf{f}(\mathbf{a}, \mathbf{p})\} && (5.146)\\
&= \{\mathbf{p} \in \mathbb{R}^p | \forall \mathbf{b} \in \mathbb{B}^+, \exists \mathbf{a} \in \mathbb{A}^-, \mathbf{f}^{-1}(\mathbf{b}) = (\mathbf{a}, \mathbf{p})\} && (5.147)
\end{aligned}$$

Considering the function $\mathbf{f}$ as a composition of a translation and a rotation where $\mathbf{f} : \mathbb{R}^2 \times \mathbb{R}^3 \to \mathbb{R}^2$, given in Equation 5.16 and reminded here:

$$\mathbf{f} : \begin{cases} \mathbb{R}^2 \times \mathbb{R}^3 & \to \mathbb{R}^2 \\ (\mathbf{a}, \mathbf{p}) & \to \mathcal{R}(p_3) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \end{cases} \tag{5.148}$$

the reciprocal function already expressed in Equation 5.19 is given by:

$$\mathbf{g} = \mathbf{f}^{-1} : \begin{cases} \mathbb{R}^2 \times \mathbb{R}^3 & \rightarrow & \mathbb{R}^2 \\ (\mathbf{x}, \mathbf{p}) & \rightarrow & \mathcal{R}^T(p_3) \begin{pmatrix} x_1 - p_1 \\ x_2 - p_2 \end{pmatrix} \end{cases} \tag{5.149}$$

Consequently Equation 5.147 can be rewritten as:

$$\begin{aligned} \mathbb{P}^- &= \{\mathbf{p} \in \mathbb{R}^p | \forall \mathbf{b} \in \mathbb{B}^+, \exists \mathbf{a} \in \mathbb{A}^-, \mathbf{f}^{-1}(\mathbf{b}) = (\mathbf{a}, \mathbf{p})\} && (5.150) \\ &= \{\mathbf{p} \in \mathbb{R}^p | \forall \mathbf{b} \in \mathbb{B}^+, \exists \mathbf{a} \in \mathbb{A}^-, \mathbf{g}(\mathbf{b}, \mathbf{p}) = \mathbf{a}\} && (5.151) \\ &= \{\mathbf{p} \in \mathbb{R}^p | \forall \mathbf{b} \in \mathbb{B}^+, \exists \mathbf{a} \in \mathbb{A}^-, \mathbf{f}^{-1}(\mathbf{b}, \mathbf{p}) = \mathbf{a}\} && (5.152) \\ &= \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}^{-1}(\mathbb{B}^+, \mathbf{p}) \subset \mathbb{A}^-\} && (5.153) \end{aligned}$$

which corresponds to a registration problem as in Equation 5.41. According to Equation 5.47, the set $\mathbb{P}^-$ is given by:

$$\mathbb{P}^- = \overline{proj_{\mathbf{p}}((\mathbb{B}^+ \times \mathbb{R}^p) \cap \mathbf{f}(\overline{\mathbb{A}^+}))} \tag{5.154}$$

Computing the set $\mathbb{P}^-$ in Equation 5.154 needs the reciprocal function of $\mathbf{f}$ given in Equation 5.19.

Example with uncertain punctual landmark

Consider an uncertain punctual landmark located in a circle $\mathbb{B}^+$ centered at $(0, 0)$ with a radius equal to $4m$. This circle is depicted in green in Figure 5.24(a) and in black in Figures 5.24(b)(c)(e)(f). The set $\mathbb{P}$ is computed according to the definitions of $\mathbb{P}^-$ and $\mathbb{P}^+$. Some slices are again given at defined heading and x values of the thick registration map. Similarly the colored AUVs or dots in the solution sets (Figures on the left) are represented in the (x, y) plane (Figures on the right) to better understand the different poses with the associated color. Moreover the thick visibility is also represented with the same color as the poses.

Firstly the slice at 45° of the thick registration map is given in Figure 5.24(a). It can be noticed that the green (thick) visibility area at the green poses never intersect the black circle. The whole black circle will be detected at the blue poses where this disk is totally included in the set $\mathbb{V}^-$ for each blue poses. The red AUVs correspond to robot poses that may detect a part of the black circle but do not guarantee the detection of the punctual landmark that lays somewhere in the black circle.

Secondly, the slice at $x = -10m$ in the $(y - \theta)$ plane is represented where the major part of the solution set is the penumbra (dark gray).

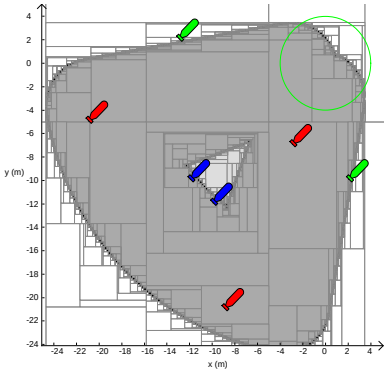
In each 2D representation, the blue poses are able to detect the whole disk $\mathbb{B}^+$ containing the uncertain punctual landmark guaranteeing the detection. Therefore, the disk $\mathbb{B}^+$ will be always visible at any blue poses $\mathbf{p}$, meaning that it is included in the certainly visible area $\mathbb{V}^-(\mathbf{p})$. Consequently it is completely included in the lower bound of the classical thick set intersection (defined in Equation 5.87) of the thick visibility areas at the blue poses defined as:

$$\bigcap_{\mathbf{p} \in \mathbb{P}^-} \mathbb{V}(\mathbf{p}) = [\bigcap_{\mathbf{p} \in \mathbb{P}^-} \mathbb{V}^-(\mathbf{p}), \bigcap_{\mathbf{p} \in \mathbb{P}^-} \mathbb{V}^+(\mathbf{p})] \tag{5.155}$$

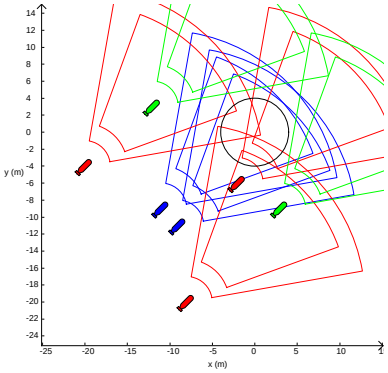
where $\mathbb{V}(\mathbf{p})$ is the thick visibility area defined at the pose $\mathbf{p}$. Consequently:

$$\mathbb{B}^+ \subset \bigcap_{\mathbf{p} \in \mathbb{P}^-} \mathbb{V}^-(\mathbf{p}) \tag{5.156}$$

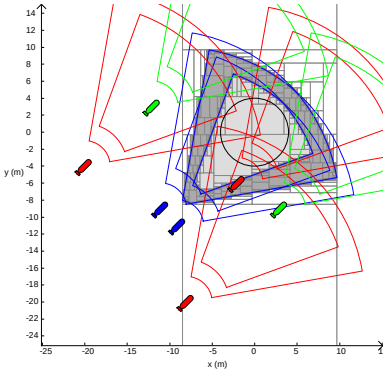
This is illustrated in Figures 5.24(c) and 5.24(f) where the thick intersection of the visibility areas at the three blue poses is depicted. It can be noticed that the black circle is inside the soft gray region corresponding to the lower bound of the thick intersection.



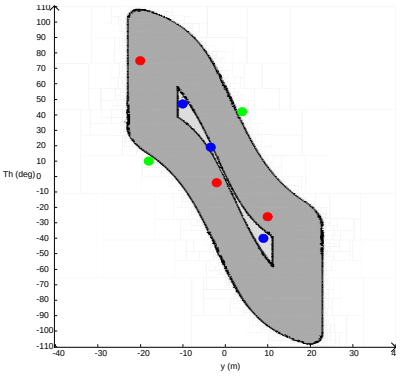
(a) Registration map at heading 45°. $\mathbb{B}^+$ is depicted by a green circle with a radius equal to $4m$.



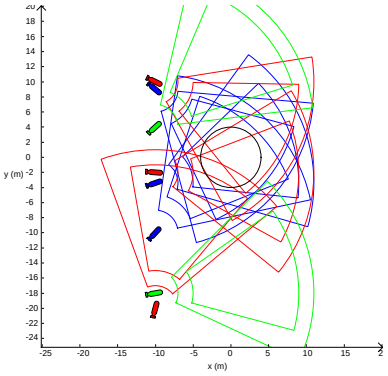
(b) 2D representation for the associated colored poses in the figure (a). $\mathbb{B}^+$ is depicted by the black circle.



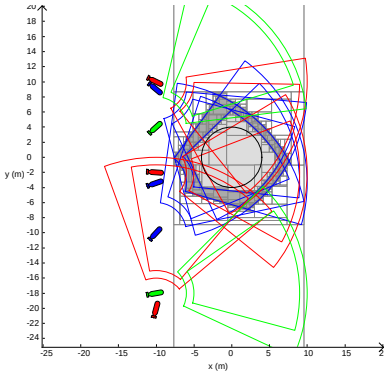
(c) Thick intersection of the thick visibility areas at the 3 blue poses.



(d) Registration map in the $(y-\theta)$ plane with $x=-10m$. Some poses are defined by dots.



(e) 2D representation for the associated colored poses in the figure (d).



(f) Same definition for the 2D representation as in (c).

Figure 5.24: Registration maps at different slices and 2D associated representation. Grayscale convention for thick set representation.

5.4 What about other sensors ?

Until now, only sensors providing directly images such as cameras or Forward Looking Sonar (FLS) were considered. This was called patch exploration [73] meaning that the visible set (or visibility area) had the dimension of the space considered. In this thesis the space is assumed 2D for simplicity reasons, i.e. $q=2$. In this case an inner and outer approximation of the registration map were available, even considering uncertainty in the visibility area and/or the landmark definition.

However, when considering a sweep exploration[73], the registration map cannot be computed due to the fact that the visibility area has a dimension $q-1=1$ ($q=2$ in this thesis). This type of exploration refers to Sidescan Sonar (SSS), SAS or MBES for example that need a motion of the platform (AUV) to collect data even if there is a very small aperture angle in

azimuth. When using these sensors, the motion has to be straight to gather exploitable data. The trajectory of the robot has then to be taken into account to prove that a landmark has been detected or not when post processing data [73][244] or at the planning phase of a mission. Similarly when considering a pen exploration [73], the registration map cannot be computed directly due to the fact that the dimension of the visible area is $q - 2 = 0$ which means that only points are observed. This is the case when a laser range-finder or an altimeter is used that takes only one range measurement at a given frequency. However, when using these types of sensor it is still possible to compute the registration map under some assumptions. Indeed it can be assumed that at each measurement a point of the landmark can be detected. Considering then a range and/or an aperture angle of detection of the sensor, the registration map is again computable. However outliers have to be handled. Usually in underwater environment, AUVs are equipped with a Doppler Velocity Log (DVL) that provides 4 range measurements to estimate the velocity of the vehicle with respect to the ground. Contrary to the altimeter that only provides one measurement, these 4 measurements can provide an information about the local surface of the ground or it can be simply used to provide the altitude above the seabed.

5.5 Conclusion

In this Chapter, the notion of registration map has been presented in a set-membership manner. Assuming a 2D environment, it indicates the robot poses that are able to detect any shape landmark depending on the visibility area of the sensor. This latter depends on how the sensor is mounted in the platform, it could at the head or on the side for example. It was assumed a sensor that provides images such as cameras or Forward Looking Sonar (FLS) in underwater environment. The problem of the registration map has been solved using the projection introduced in Chapter 4. Considering the transformation function $\mathbf{f}$ as a composition of translation and rotation, an inner and an outer approximation of the solution set is available. Some examples considering punctual, ellipse or random shape landmark are provided. Due to the 3D dimension of the solution set (x, y, θ) referring to the position (x, y) and the heading θ of the AUV, some slices at different x, y or θ values are provided to give an idea of the solution set and for ease of interpretation based on interval analysis. When the registration map at a given heading θ is required, the problem consists in estimating the translation parameters to detect a part of the landmark. This problem can almost be seen as a Minkowski difference. The Minkowski operations, referring to erosion or dilation in image processing, are presented as a shape registration based again on the projection.

The notion of point of view of a landmark, introduced in Chapter 2, has been seen as an additional constraint on the registration map. This is particularly interesting in the underwater environment when using sonars because a landmark that has been detected at a specific point of view during a previous survey mission may not be detectable at other points of view. Indeed it is possible that the landmark may be covered by sand, the backscattered signal may not be strong enough compared to noise or the geometry of the landmark may reflect the sound waves in other directions for example.

In a first part, the landmark and the visibility area of the sensor were considered as *thin* sets assuming no uncertainty. The landmark is reduced to a singleton when dealing with a punctual landmark. Then in a second part some uncertainties in the landmark and in the visibility area have been introduced leading to an ill-defined contour of the landmark and the visibility area. This problem was handled with *thick* sets that are an extension of intervals. A thick set $[\mathbb{X}]$ can

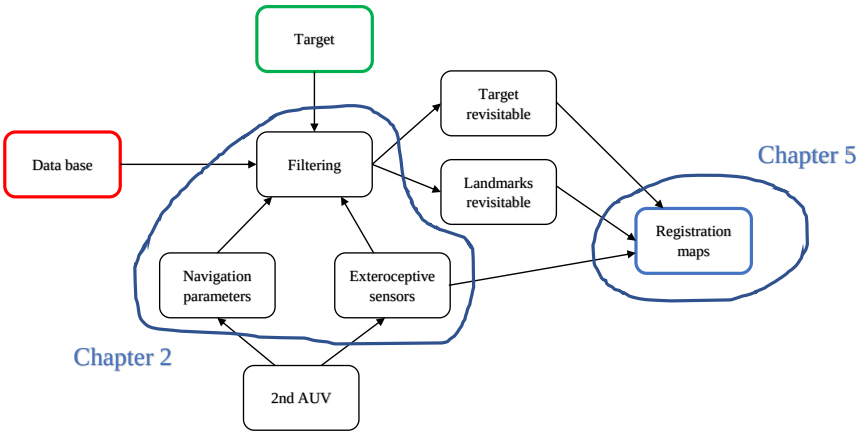


Figure 5.25: Diagram of the adaptation to the revisit system

be enclosed between two sets $\mathbb{X}^-$ and $\mathbb{X}^+$ such that $[\mathbb{X}] \in [\mathbb{X}^-, \mathbb{X}^+]$. It introduces a penumbra. If the classical approach was used, the computation of the registration map will spend most of the time in this penumbra to determine if some robot poses can detect the landmark. The computation of the registration map dealing with both uncertainties (in map and in sensing) leads to a thick representation with some examples provided at defined x, y or θ values for a 2D representation. The penumbra has been rigorously handled to avoid spending time in this penumbra and focus on real robot poses that are able to detect the landmark.

Based on a *a priori* known environment, possibly with ill-defined location, the registration map is computed based on the landmark and the visibility area as depicted in Figure 1.19 which is drawn again here in Figure 5.25. These registration maps are the inputs of the motion planner as it indicates the robot poses where the uncertainty accumulated by the robot can be reduced. The next Chapter proposes a planner based on the registration maps to navigate certainly despite the uncertainties in motion between these regions where the uncertainty can be reset as in [186].

Chapter 6

Motion planner

6.1 Introduction

In Chapter 2, some algorithms enabling the detection of underwater landmarks were presented. The detections depend on the landmark obviously and on the sensor. Due to the poor visibility in underwater environment and the performance of acoustics, the focus was on sonar imagery. A simple adaptation based on the sonar resolution was proposed to keep only a set of landmarks initially detected by an high resolution sonar during a survey mission. An object was classified as a Mine Like Contact (MILCO) and a revisit was required. Some state-of-the-art motion planners were proposed in Chapter 3 that take into account the pose uncertainty due to low cost design of revisit vehicles that may self detonate to remove the object if it is identified as a mine. In Chapter 5, the registration map was presented based on theoretical tools introduced in Chapter 4 on interval analysis. These registration maps were built according to any shape landmark and based on the visibility area of the sensor. These maps indicate robot poses that are able to detect the landmark considered. The proposed approach in this Chapter is to provide a high level strategy for the robot based on the registration maps as depicted in Figure 1.20 in Chapter 1 and reminded here in Figure 6.1. Based on the parameters of the revisit vehicle, i.e. sensor used, the registration maps in the red box were computed. Similarly the registration map concerning the target in the green box was computed. Based on all these registration maps, the algorithm proposed here provides a high level strategy taking into account the uncertainty in the motion and the constraint on the manoeuvrability of the vehicle. The algorithm is based on set-membership theoretical tools presented in Chapter 4 and 5 and on graph search strategy presented in Chapter 3. However some assumptions have to be made to solve the problem and will be presented along this Chapter.

In a first section, the problem of the motion planning is explicited. Then the robot motion models used to solve the motion planning problem are presented. It poposes the use of Dubins paths [83] to find strategy for vehicle with non-holonomic constraints and linear path based on uncertain heading direction for vehicle that can move in any direction. In a third section, the problem of the preimage of a set is introduced as a backward reach set. Moreover, the notion of forward reach set is also presented. In a fourth section, the motion planner is proposed for a simplified environment. Then the motion planner based on the registration maps is proposed.

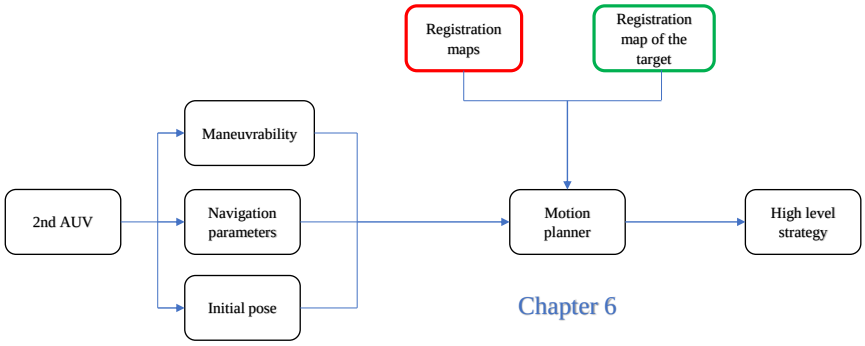


Figure 6.1: Diagram of the high level strategy finding

6.2 Problem statement

The problem of the motion planning proposed in this thesis consists in finding a sequence π of open-loop strategies μ_i to join a goal region. The aim of the planner is the revisit of an object identified as potentially dangerous where the absolute or approximate coordinates are provided by a survey mission. This object was designated as suspicious according to an operator or an automatic algorithm. This goal region is consequently defined as the registration map of the target developped in the last Chapter based on the visibility area of the sensor embedded in the revisit vehicle. This motion planning problem is inspired by the concept of preimage backchaining [203] introduced in Chapter 3 where a landmark based planner was proposed in [186][187]. This first algorithm relied on circular relocation areas, known as islands of perfection or landmark areas, where the control and position sensing were assumed perfect. A feature such as a corner may be existing in these relocation areas. As many motion planners, it also assumed the presence of obstacles designed as circular areas too. All these disks could have different radius. The planning problem provides a set of motion commands which guarantee that the robot will move into the goal and stop in it. Starting from the goal region, it solves the problem by iteratively building a growing set of landmark areas from where the robot can reliably reach the goal. A new landmark area is added to the set when it can reach from this one a landmark area already in the set by executing a single move defined as a motion command. The algorithm terminates when the robot can reach this growing set from the initial region, otherwise the planner returns failure. The single moves are computed based on omnidirectional backprojections defined as the disjoint union of directional backprojections over all possible directions of motion. As the uncertainty is assumed on the direction d, which could come from an embedded compass, it results in a cone propagation around d of the motion of the robot outside the relocation areas as depicted in Figure 3.6(a) in Chapter 3. This cone propagation defines one directional backprojection in the direction d. The planner is complete and its complexity is polynomial in the number of landmark areas and obstacles. Finally, the

planner computed a guaranteed plan under given uncertainty bounds by backchaining omnidirectional backprojections of the goal (the goal is successively extended with some relocation areas) until a backprojection fully contains the initial set of robot positions. When the motion was guaranteed to reach a relocation area, the uncertainty on the position of the robot was reset to 0 (or an ϵ value is possible too). Only a discrete set of directional backprojections were needed to compute the omnidirectional backprojection at specific directions defined as critical directions. An illustration of such plan was proposed in Chapter 3 in Figure 3.6(c).

The formalization of the motion planner problem in [186] is solved in this thesis in a set-membership manner using the concept of preimage as backward reach set where the landmark areas represent the registration maps computed according to the previous Chapter. It extends then the work to unstructured shape environment and takes into account the sensing abilities of the sensor by using the visibility area.

The high-level open-loop plan π can be expressed as follows:

$$\pi = (\mu_1, \mu_2, ..., \mu_k) \tag{6.1}$$

which is a sequence of k motion commands. These motion commands μ_i may refer for example to different high-level strategies:

- Go to the North,...
- Follow the wall.
- Follow the contour of the shape defining sand ripples.
- Move the robot to a certain area using vision sensor in a relocation area.

These high-level commands depend on measurements based on exteroceptive sensors such as sonars or compass. In this thesis, it is assumed that a compass is embedded in the vehicle providing the heading information. However, this latter is not accurate and it will be assumed that it lays in an bounded interval. It will the be assumed that the robot is able to follow an uncertain heading as in [186]. Moreover, the commands "*Follow the contour of the shape*" or "*Move the robot to a certain area*" rely on the detection of a landmark and a registration to relocalize the robot. It will be assumed in this thesis that a low-level controller exists and is able to perform the visually guided motion. This open-loop plan is actually composed of closed-loop motion commands.

Contrary to a predefined path as in many motion planning algorithms, it deals here with an high level strategy to follow involving relocation process. Following a predefined path for an AUV is not an easy task when the dynamic model is non linear and requires the development of robust controller [176][175].

The generation of these motion commands depend on the preimage of a set defined in this thesis by a registration map and a motion function. This latter will be discussed in Section 6.3. Based on the motion function, the preimage will then be computed as a backward reach set. This will be discussed in Section 6.4. Finally, the preimage backchaining will be discussed in Section 6.5 to provide the high level graph for the strategy that is optimized under a graph search with a Dijkstra or an A* algorithm. Then the planner will be applied to the registration maps in Section 6.6. Finally, a spiral exploration will be proposed to disambiguate the true robot position when the robot has to revisit a cluster of indistinguishable landmarks.

6.3 Robot motion model

A robot is described by an evolution equation:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{6.2}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state of the robot, $\mathbf{u} \in \mathbb{R}^m$ is the control applied on the state $\mathbf{x}$ of the robot and $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the evolution function.

In this thesis, the environment is assumed 2D then $\mathbf{x} = (x, y, \theta) \in \mathbb{R}^3$. It will be assumed a simple dynamics model known as the Dubins car [83] which is described by:

$$\begin{cases} \dot{x} = v \cos\theta \\ \dot{y} = v \sin\theta \\ \dot{\theta} = w \end{cases} \tag{6.3}$$

with $\mathbf{u} = (v, w)$ the command of the system. v corresponds to the linear speed and w to the angular rate (turn rate).

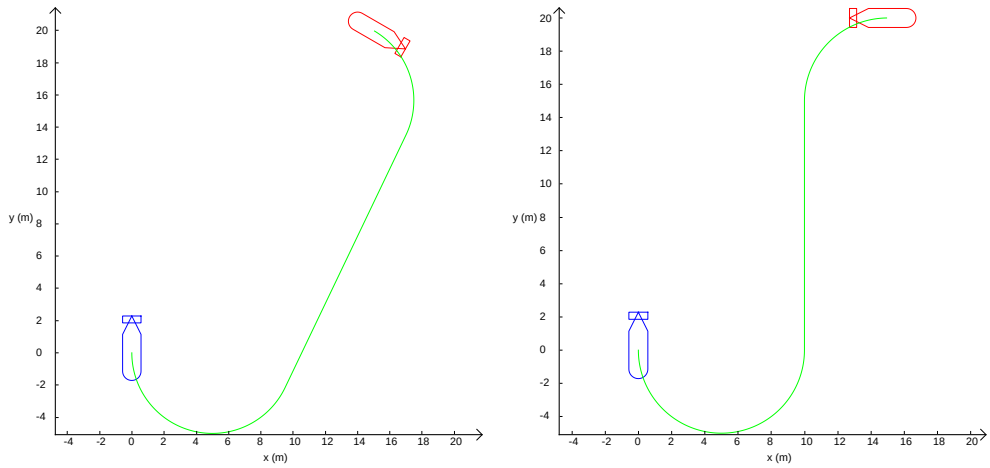
6.3.1 Dubins paths

Considering a constant linear speed and a constant turning rate, the so-called Dubins paths, known as curves, can connect [83] any two configurations $\mathbf{x}_A$ and $\mathbf{x}_B$ in the plane by the composition of turns and straight motions as depicted in Figure 6.2 where $\mathbf{x}_A$ and $\mathbf{x}_B$ are drawn respectively by the blue and the red AUV. The four paths, represented in green, are called: LSL, LSR, RSL and RSR with L describing a "*left turn*", R is a "*right turn*" and S corresponds to the command driving or going "*straight*". For example, the path LSR indicates the following steps:

- firstly "turn left".
- secondly "go straight".
- finally "turn right".

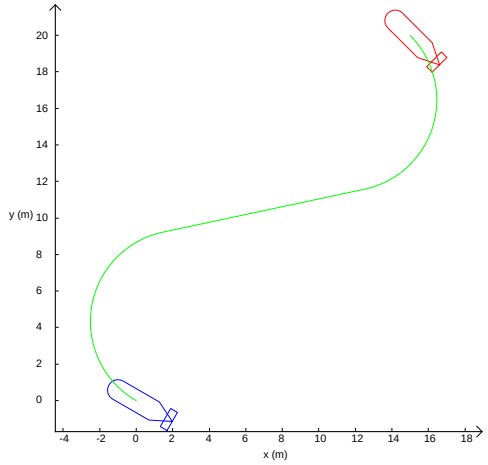
The paths provided are optimal in term of distance travelled, or in other words it corresponds to the shortest Dubins path between the two configurations since the angular rate w is bounded. Indeed, considering a bounded turning rate w leads to a maximum turning rate and finally to the so-called minimum turning radius. Considering the minimum turning radius will enhance the shortest path between any two configurations. The link between the angular rate and the turning radius will be highlighted juts below. The Dubins paths are an interesting way for motion planning under non-holonomic constraints for a vehicle.

In Figure 6.2, the linear speed is fixed at $1m/s$ and the angular rate at $0.3rad/s$ which lead approximately to a turning radius at $3m$. The method to compute such paths is not provided here.

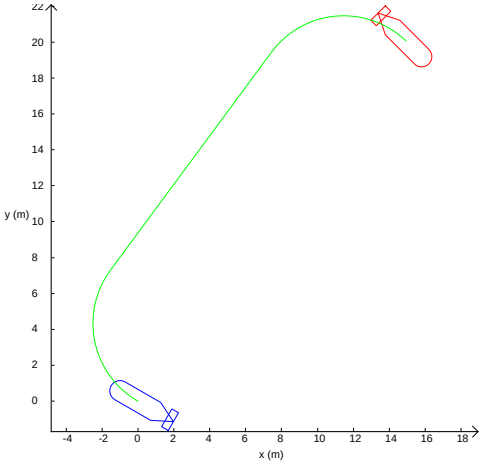


(a) LSL (Left-Straight-Left) path.

(b) LSR (Left-Straight-Right) path.



(c) RSL (Right-Straight-Left) path.



(d) RSR (Right-Straight-Right) path.

Figure 6.2: Shortest Dubins paths between the initial configuration in blue and the final configuration in red.

6.3.2 Parametric Dubins paths

These paths are described by the composition of turns and straight motions which can be described by a parametric equation taking into the duration t. Indeed, turning can be described temporally by integrating Equation 6.3 with a constant linear speed v and a constant angular rate $w \neq 0$:

$$\mathbf{f}_c : \begin{cases} \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ (\mathbf{x}_0, t, v, w) & \rightarrow & \begin{pmatrix} x_0 + \frac{v}{w}(\sin(wt + \theta_0) - \sin(\theta_0) \\ y_0 + \frac{v}{w}(\cos(\theta_0) - \cos(wt + \theta_0) \\ \theta_0 + wt \end{pmatrix} \end{cases} \tag{6.4}$$

where $\mathbf{x}_0 = \mathbf{x}(t = 0) = (x_0, y_0, \theta_0)$. The function is called $\mathbf{f}_c$ with "c" for circular and gives the pose of the robot at any time t. If $w > 0$, it corresponds to a turn on the left in the time and if $w < 0$ it is a turn on the right. It can be directly seen that if $w > 0$, θ will increase with time t and turning left. Notice that the term $\frac{v}{w}$ appears when integrating, it corresponds to the so-called turning radius. For example, in Figure 6.2, with $v = 1m/s$ and $w = 0.3rad/s$, the turning radius is almost $3m$.

Similarly, the straight motion is given by the integration of Equation 6.3 with an angular rate $w = 0$:

$$\mathbf{f}_s : \begin{cases} \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ (\mathbf{x}_0, t, v, w) & \rightarrow & \begin{pmatrix} x_0 \\ y_0 \\ \theta_0 \end{pmatrix} + \begin{pmatrix} v\cos(\theta_0)t \\ v\sin(\theta_0)t \\ 0 \end{pmatrix} \end{cases} \tag{6.5}$$

This function is called $\mathbf{f}_s$ with "s" for straight.

The Dubins paths considered are the four possible combinations expressed before. It would have been obviously possible to consider the two last combinations RLR and LRL for optimality. More combinations are possible but it wouldn't be optimal. They are consequently described firstly by a turn with a duration called t_{c_1}, a straight motion with a duration t_s and finally a last turn with a duration t_{c_2}. These paths can then be described by the composition of the functions defined in Equation 6.4 for the turning motion and in Equation 6.5 for the straight motion which leads to the following function:

$$\mathbf{f}_{Dubins} : \begin{cases} \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ (\mathbf{x}_0, \mathbf{t}, v, \mathbf{w}) & \rightarrow & \mathbf{f}_c(\mathbf{f}_s(\mathbf{f}_c(\mathbf{x}_0, t_{c_1}, v, w_{c_1}), t_s, v, 0), t_{c_2}, v, w_{c_2}) \end{cases} \tag{6.6}$$

with $\mathbf{t} = (t_{c_1}, t_s, t_{c_2})$ and $\mathbf{w} = (w_{c_1}, w_{c_2})$. In this model, it is assumed that the linear speed v is the same in the three portions of the motion. w_{c_1} and w_{c_2} correspond to the angular rates of the first turn and the second turn respectively. The angular rate w is assumed to be the same in the two turns which means that $w_{c_1} = \pm w$ and $w_{c_1} = \pm w$. The sign depends on the chosen direction for turning.

The function expressed in Equation 6.6 gives the pose of the robot $\mathbf{x} = (x, y, \theta)$ at time $\mathbf{t} = (t_{c_1}, t_s, t_{c_2})$ which is expressed as follows:

$$\mathbf{x}(\mathbf{t}) = \mathbf{f}_{Dubins}(\mathbf{x}_0, \mathbf{t}, v, \mathbf{w}) \tag{6.7}$$

Finally, the pose of the robot can be given at any time $t \in \mathbb{R}^+$ according to a Dubins path defined by $\mathbf{t}_{path} = (t_{c_1}, t_s, t_{c_2})$ by considering the following $\mathbf{t}$ in Equation 6.7:

$$\begin{cases} \mathbf{t} = (t, 0, 0) & \text{if } t \in [0, t_{c_1}] \\ \mathbf{t} = (t_{c_1}, t, 0) & \text{if } t \in [t_{c_1}, t_{c_1} + t_s] \\ \mathbf{t} = (t_{c_1}, t_s, t) & \text{if } t \in [t_{c_1} + t_s, t_{c_2}] \end{cases} \tag{6.8}$$

This function provides a parametric description of the Dubins paths.

6.3.3 Uncertain parametric Dubins paths

It is assumed that errors exist in robot control, that means the robot does not execute perfectly the motion commands required. These errors will enhance the robot to drift from the desired trajectory. Compared to probabilistic method [308], it will be assumed bounded errors to remain in a set-membership context. It will be assumed that the uncertainties are on the initial heading θ_0 due to the presence of a compass and on the linear speed v. $\delta\theta$ is introduced to model the error on θ_0 and to reflect the error on the compass for example. As it is assumed that the robot is equipped with a compass that provides the heading information with a bounded error in the interval $[-\alpha_\theta, \alpha_\theta]$, this enables the robot to follow the proposed trajectories with a perfect angular rate w but an uncertain initial heading θ_0. The true heading at the initial pose

is then inside an interval $[\theta_0 - \alpha_\theta, \theta_0 + \alpha_\theta]$ with θ_0 provided by the compass. In general, the error on the compass α_θ is few degrees ($\approx 2-3°$). An error rate α_v (usually between $3\% - 10\%$) is introduced for the linear speed. That means the linear speed $v \in v_d + v_d \cdot [-\alpha_v, \alpha_v]$ with v_d the desired speed. The linear speed is constant but uncertain on this interval.

Considering these uncertain parameters, a slight change on the function expressed in Equations 6.4, 6.5, 6.6 and 6.7. As the angular rate is a fixed parameter and certainly defined, it does not need to be included in the definition of the function. However, the value of w has to be defined according to desired turn (left or right).

The definition of the circular trajectory becomes:

$$\mathbf{g}_c^w : \begin{cases} \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} & \rightarrow & \mathbb{R}^3 \\ (\mathbf{x}_0, t, v, \delta\theta) & \rightarrow & \begin{pmatrix} x_0 + \frac{v}{w}(\sin(wt + \theta_0 + \delta\theta) - \sin(\theta_0 + \delta\theta) \\ y_0 + \frac{v}{w}(\cos(\theta_0 + \delta\theta) - \cos(wt + \theta_0 + \delta\theta) \\ \theta_0 + \delta\theta + wt \end{pmatrix} \end{cases} \tag{6.9}$$

with $\delta\theta$ the parameter for the initial heading error. Notice that the angular rate w is added to the function to distinguish the right and the left turn.

The definition of the straight motion becomes:

$$\mathbf{g}_s : \begin{cases} \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} & \rightarrow & \mathbb{R}^3 \\ (\mathbf{x}_0, t, v, \delta\theta) & \rightarrow & \begin{pmatrix} x_0 \\ y_0 \\ \theta_0 \end{pmatrix} + \begin{pmatrix} v\cos(\theta_0 + \delta\theta)t \\ v\sin(\theta_0 + \delta\theta)t \\ \delta\theta \end{pmatrix} \end{cases} \tag{6.10}$$

Considering as $\mathbf{p} = (v, \delta\theta)$ the uncertain parameters vector in the motion, the global function giving the Dubins paths is defined as follows:

$$\mathbf{g}_{Dubins} : \begin{cases} \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ (\mathbf{x}_0, \mathbf{t}, \mathbf{p}) & \rightarrow & \mathbf{g}_c^{w_{c_2}}(\mathbf{g}_s(\mathbf{g}_c^{w_{c_1}}(\mathbf{x}_0, t_{c_1}, v, \delta\theta), t_s, v, 0), t_{c_2}, v, 0) \end{cases} \tag{6.11}$$

Notice that the error $\delta\theta$ on the initial heading θ_0 is only applied on the first motion (the turn with angular rate w_{c_1}). This error will be propagated in two last portions of the motion. The values of w_{c_1} and w_{c_2} are defined according to the desired Dubins paths.

The function expressed in Equation 6.7 can be modified as follows:

$$\mathbf{x}(\mathbf{t}) = \mathbf{g}_{Dubins}(\mathbf{x}_0, \mathbf{t}, \mathbf{p}) \tag{6.12}$$

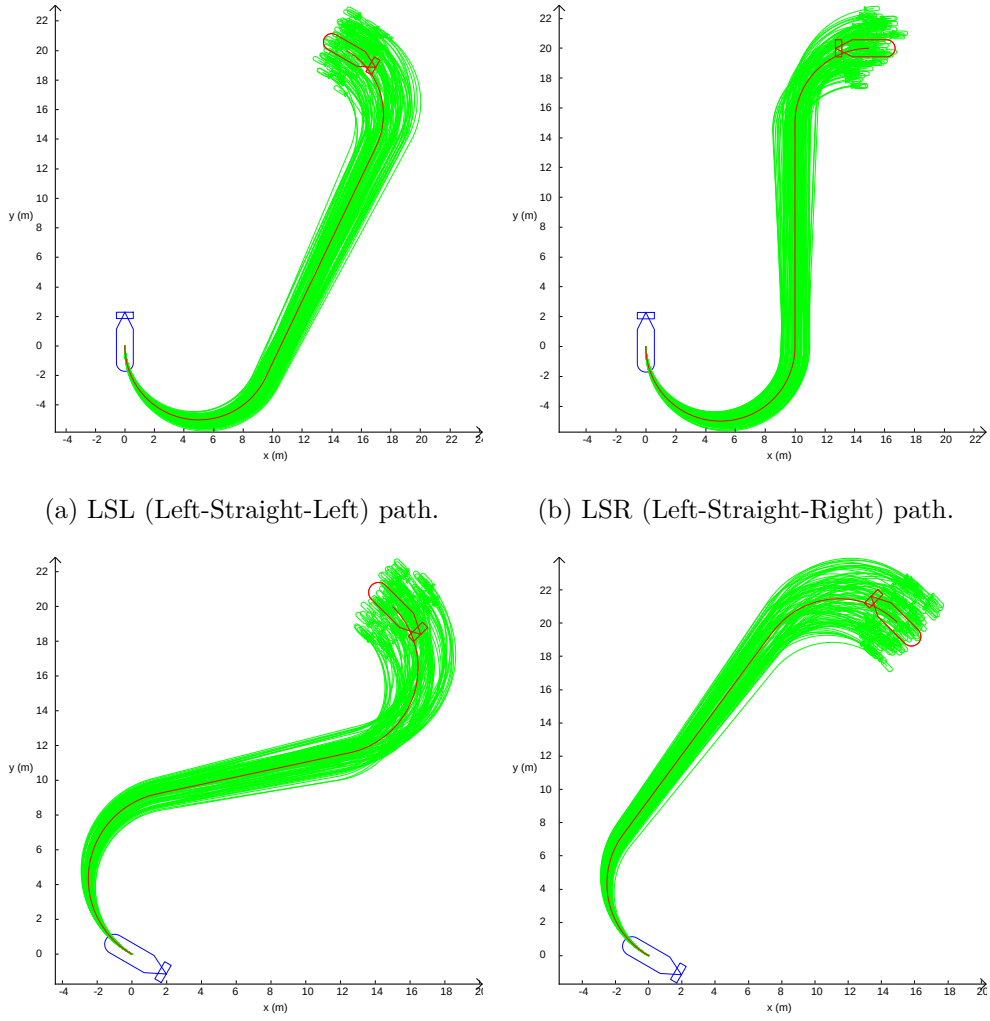
with $\mathbf{p} = (v, \delta\theta)$ the uncertain parameters vector to model the drift.

Similarly, the pose of the robot can be given at any time $t \in \mathbb{R}^+$ according to the different cases in Equation 6.8.

The uncertain parameters were defined as follows: $v \in v_d + v_d \cdot [-\alpha_v, \alpha_v]$ and $\delta\theta \in [-\alpha_\theta, \alpha_\theta]$ which leads to $[\mathbf{p}] = [v_d(1 - \alpha_v), v_d(1 + \alpha_v)] \times [-\alpha_\theta, \alpha_\theta]$.

Some uncertain trajectories are depicted in Figure 6.3 where the motion commands generated are the same as in Figure 6.2. These motion commands correspond to the vector $\mathbf{t} = (t_{c_1}, t_s, t_{c_2})$ according to the defined linear speed and angular rate. It can be seen by the red trajectory.

The uncertain parameters used for the different figures are $\alpha_v = 10\%$ and $\alpha_\theta = 3°$. The linear speed was fixed at $1m/s$ and the angular rate at $0.3rad/s$. It results then in the uncertain parameters vector $[\mathbf{p}] = [0.9, 1.1] \times [-3, 3](\frac{m}{s} \times deg)$. Random samples have been chosen in the two intervals to compute the green trajectories. It could have been possible to compute the set of trajectories using tubes defined in [274].



(a) LSL (Left-Straight-Left) path.

(b) LSR (Left-Straight-Right) path.

(c) RSL (Right-Straight-Left) path.

(d) RSR (Right-Straight-Right) path.

Figure 6.3: Uncertain Dubins paths in green generated by the command for the shortest Dubins path in red between the initial configuration in blue and the final configuration in red. Uncertainties: 3° error on initial heading and 10% error on the linear speed.

6.3.4 Two motion models

From this uncertain Dubins path model, two motion models will be considered in this thesis. To follow the robotics convention about the transition function, the function **g** will be denoted **f** instead. The name **g** was given only to make the difference with paths that do not consider uncertain parameters.

Firstly, when considering an AUV with non-holonomic constraint and the presence of a minimum turning radius, the model used will be the one in Equation 6.12. The evolution

function will be called $\mathbf{f}_{NH}$ and is defined as follows:

$$\mathbf{f}_{NH} : \begin{cases} \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ (\mathbf{x}_0, \mathbf{t}, \mathbf{p}) & \rightarrow & \mathbf{f}_c^{w_{c2}}(\mathbf{f}_s(\mathbf{f}_c^{w_{c1}}(\mathbf{x}_0, t_{c_1}, v, \delta\theta), t_s, v, 0), t_{c_2}, v, 0) \end{cases} \tag{6.13}$$

with $\mathbf{p} = (v, \delta\theta)$ where $\mathbf{p} \in [\mathbf{p}] = [v_d(1-\alpha_v), v_d(1+\alpha_v)] \times [-\alpha_\theta, \alpha_\theta]$. v_d is the desired speed along the complete motion.

Secondly, when considering an holonomic AUV, it will be assumed that the AUV is able to move in any direction by rotating on itself and following straight trajectories with uncertain initial heading and speed. The model used is then expressed in Equation 6.10 and is expressed as follows:

$$\mathbf{f}_H : \begin{cases} \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ (\mathbf{x}_0, t, \mathbf{p}) & \rightarrow & \begin{pmatrix} x_0 \\ y_0 \\ \theta_0 \end{pmatrix} + \begin{pmatrix} v\cos(\theta_0 + \delta\theta)t \\ v\sin(\theta_0 + \delta\theta)t \\ \delta\theta \end{pmatrix} \end{cases} \tag{6.14}$$

with $\mathbf{p}$ defined as previously. v_d is the desired speed along the straight motion.

As many goal sets in the different examples will be only in $\mathbb{R}^2$, the 2D version is defined as follows:

$$\mathbf{f}_{H2} : \begin{cases} \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^2 \\ (\mathbf{x}_0, t, \mathbf{p}) & \rightarrow & \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} v\cos(\theta_0 + \delta\theta)t \\ v\sin(\theta_0 + \delta\theta)t \end{pmatrix} \end{cases} \tag{6.15}$$

When dealing with 2D goal sets, in the $(x-y)$ plane, the model for the holonomic robot is named $\mathbf{f}_{H2}$. It is the same model as in Equation 6.14 but the third dimension that provides the heading is removed.

In both models, it will be assumed that the vehicle can reach rapidly the desired speed v_d to remain in the interval of uncertainty according to the linear speed. This assumption is valid when navigating at low speed (few meters/second) such as AUVs.

A general description giving the pose $\mathbf{x}$ at any time $\mathbf{t}$ for both models can be described as follows:

$$\mathbf{x}(\mathbf{t}) = \mathbf{f}(\mathbf{x}_0, \mathbf{t}, \mathbf{p}) \tag{6.16}$$

where $\mathbf{t}$ is defined according to Equation 6.8 to give the position at any time $t \in \mathbb{R}^+$ when the first model is used, and $\mathbf{t} = t$ when the second model is used.

6.4 Reachability problem

The robot motion model has been defined as a parametric function with uncertain parameters to model the drift of the robot. The open-loop strategies μ_i of the motion plan are based on this motion model. The robot has to follow these strategies to navigate between relocation areas defined as registration maps in this thesis. To guarantee that the robot will safely navigate between two areas without being lost, the notion of preimage is introduced and was used to solve the motion planning problem in [186]. Before building a graph and providing the high level strategy, this notion has to be clearly defined.

A strategy μ_i is defined by a particular motion command m_i and a termination condition denoted as $\mathbf{TC}_i$ in [186]. When the robot executes the motion command m_i, it follows the defined motion and stops when the termination condition $\mathbf{TC}_i$ is true. This termination condition could be the contact with a wall, the detection of the landmark or a defined time for example. A strategy μ_i could be to follow a Dubins path command or to follow a particular direction for instance.

Definition

A preimage of a goal region for a given strategy corresponds to the set of robot configurations such that if the robot starts the strategy from this set, it is guaranteed to reach the goal region and stop in it according to the termination condition.

The set of robot configurations corresponds to a backward reach set or a backprojection [183] associated to a termination condition. The robot has to reach the goal (goal reachability) despite the motion uncertainty and stop in it (goal recognizability) despite the sensing uncertainty. In [89], Erdmann suggested that the goal reachability and the goal recognizability should be treated separately. In this thesis, it will be generally assumed that the goal recognizability (termination condition) is based on the detection of the landmark revisited and then depends on an automatic or manual detection algorithm in sonar images. However, the goal reachability and recognizability will be computed at the same time when dealing with point of view of a landmark assuming the holonomic robot motion model. Therefore, in this thesis, the main work is focused on the goal reachability and so on the registration map reachability.

6.4.1 Backward reach set

It is important to guarantee that the goal is reached despite the uncertainty on the motion. It refers to the *strong backprojection* (**SB**) in [183] compared to the *weak backprojection* (**WB**) for a given strategy μ.

Consider now an evolution function $\mathbf{f}$ as defined in Equation 6.16 and reminded here:

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{x}_0, t, \mathbf{p}) \tag{6.17}$$

with $\mathbf{p}$ the uncertain parameters leading to a drift from the desired trajectory. The pose of the robot $\mathbf{x}$ belongs to $\mathbb{R}^3$. A strategy μ is defined according to a particular motion and is attached to $\mathbf{f}$. It will be denoted as $\mathbf{f}_\mu$.

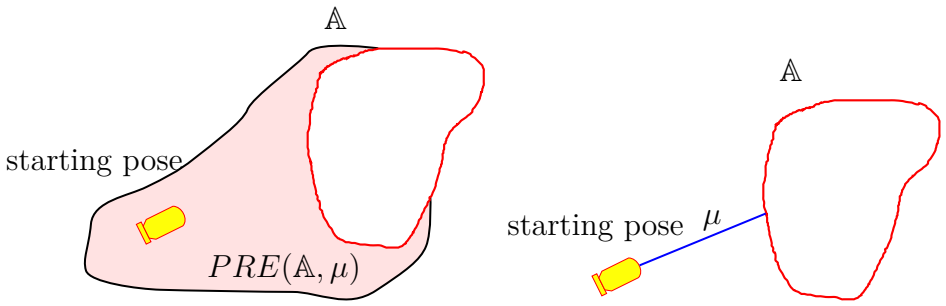
Definition 1

The backward reach set (strong backprojection) of a set $\mathbb{A} \subset \mathbb{R}^3$ under a strategy μ corresponds to:

$$BACK(\mathbb{A}, \mu) = \{\mathbf{x} \in \mathbb{R}^3 | \forall \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, \mathbf{f}_\mu(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \tag{6.18}$$

The set $\mathbb{A}$ may not be restricted to a connected set and it will proved in the following of the document the computation of the backward reach set of the union of disconnected or connected sets. The backward reach set $BACK(\mathbb{A}, \mu)$ is also denoted as $PRE(\mathbb{A}, \mu)$ to use the name preimage. An example is given in Figure 6.4 where the soft pink area corresponds to the preimage of the set $\mathbb{A}$ under the strategy μ. As the pose of the AUV, represented by the yellow

AUV, is inside this preimage, then it will reach for sure the set $\mathbb{A}$ despite the uncertainty on the motion. A blue link is then created between this robot pose and the set $\mathbb{A}$.



(a) Preimage with strategy μ (b) Link created.

Figure 6.4: Preimage as a backprojection under strategy μ.

The backward reach set can be seen as the inversion of a parametric function as in Equation 5.114 where $\mathbb{Y}^- = \mathbb{A}$:

$$
\begin{aligned}
BACK(\mathbb{A}, \mu) &= \bigcap_{\mathbf{p}\in[\mathbf{p}]} \{\mathbf{x} \in \mathbb{R}^3 | \exists t \in \mathbb{R}^+, \mathbf{f}_\mu(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} && (6.19)\\
&= \bigcap_{\mathbf{p}\in[\mathbf{p}]} \bigcup_{t\in\mathbb{R}^+} \{\mathbf{x} \in \mathbb{R}^3 | \mathbf{f}_\mu(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} && (6.20)\\
&= \bigcap_{\mathbf{p}\in[\mathbf{p}]} \bigcup_{t\in\mathbb{R}^+} \mathbf{f}_\mu^{-1}(\mathbb{A}) && (6.21)\\
& && (6.22)
\end{aligned}
$$

which can be rewritten with projection as:

$$BACK(\mathbb{A}, \mu) = \overline{proj_{\mathbb{X}}^{[\mathbf{p}]}(\overline{proj_{\mathbb{X}\times\mathbb{P}}^{\mathbb{T}}(\mathbf{f}_\mu^{-1}(\mathbb{A}))})} \tag{6.23}$$

according to Equations 5.117 and 5.121 where $\mathbb{P}$ is the domain of $\mathbf{p}$, $\mathbb{X}$ is the domain of $\mathbf{X}$ and $\mathbb{T} = \mathbb{R}^+$ is the domain of t. The definition of projection was given in Equation 4.104 in Chapter 4. However, a notation is added to the projection to indicate the domain with respect to the projection is executed. The new notation is then given as follows:

New notation

Given two sets $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{Y} \subset \mathbb{R}^p$. Considering the set $\mathbb{Z} = \mathbb{X} \times \mathbb{Y}$, the projection of a subset $\mathbb{Z}_1$ of $\mathbb{Z}$ onto $\mathbb{X}$, with respect to $\mathbb{Y}$, is defined as follows:

$$proj_{\mathbb{X}}^{\mathbb{Y}}(\mathbb{Z}_1) = \{\mathbf{x} \in \mathbb{X} | \exists \mathbf{y} \in \mathbb{Y}, (\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_1\} \tag{6.24}$$

It leads to the notation in Equation 6.23.

The backward reach set defined in Equation 6.18 can be formulated as follows: it indicates the robot poses such that for all uncertain (bounded) parameters $\mathbf{p}$, a time t exists such that the motion of the robot crosses the set $\mathbb{A}$ under the strategy μ. The robot may not cross the set $\mathbb{A}$ at the same time t depending on the uncertain parameter $\mathbf{p}$.

From the definition of the back reach set, the following proposition can be defined when considering several connected or disconnected goal sets.

Proposition

Consider two sets $\mathbb{A} \subset \mathbb{R}^3$ and $\mathbb{B} \subset \mathbb{R}^3$ that may be disconnected then:

$$BACK(\mathbb{A}, \mu) \cup BACK(\mathbb{B}, \mu) \subset BACK(\mathbb{A} \cup \mathbb{B}, \mu) \tag{6.25}$$

Proof

Choose $\mathbf{x} \in \mathbb{R}^3$ such that:

$$\begin{aligned}
& \mathbf{x} \in BACK(\mathbb{A}, \mu) \cup BACK(\mathbb{B}, \mu) && (6.26)\\
\Leftrightarrow\ & (\forall \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, \mathbf{f}_\mu(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}) \vee (\forall \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, \mathbf{f}_\mu(\mathbf{x}, t, \mathbf{p},) \in \mathbb{B}) && (6.27)\\
\Leftrightarrow\ & \forall \mathbf{p} \in [\mathbf{p}], (\exists t \in \mathbb{R}^+, \mathbf{f}_\mu(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}) \vee (\exists t \in \mathbb{R}^+, \mathbf{f}_\mu(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B}) && (6.28)\\
\Rightarrow\ & \forall \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, (\mathbf{f}_\mu(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}) \vee (\mathbf{f}_\mu(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B}) && (6.29)\\
\Leftrightarrow\ & \forall \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, \mathbf{f}_\mu(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A} \cup \mathbb{B} && (6.30)\\
\Leftrightarrow\ & \mathbf{x} \in BACK(\mathbb{A} \cup \mathbb{B}, \mu) && (6.31)
\end{aligned}$$

that ends the proof.

Definition 2

The weak backward reach set (weak backprojection) of a set $\mathbb{A} \subset \mathbb{R}^3$ under strategy μ corresponds to:

$$WBACK(\mathbb{A}, \mu) = \{\mathbf{x} \in \mathbb{R}^3 | \exists \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, \mathbf{f}_\mu(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \tag{6.32}$$

The difference with the previous definition of the backward reach set is the quantifier $\exists \mathbf{p}$ instead of $\forall \mathbf{p}$. That means some trajectories may reach the set $\mathbb{A}$ but not necessary all the trajectories according to the uncertain parameter $\mathbf{p}$. This notion will be used when dealing with obstacles in particular.

It can also be rewritten as projection:

$$WBACK(\mathbb{A}, \mu) = proj_{\mathbb{X}}^{[\mathbf{p}]\times\mathbb{T}}(\mathbf{f}_\mu^{-1}(\mathbb{A})) \tag{6.33}$$

Some illustrations will be given in the following of the document considering the proposed motion models.

From the two motion models defined in Equations 6.13 and 6.14 in the last subsection, some strategies can be defined. Firstly the directional backprojection as proposed in[186] is developed where the propagation of the uncertainty on the robot position is along a cone assuming directions with bounded uncertainty. The holonomic robot motion model will be used. Secondly, the backprojection dealing with the parametric Dubins paths will be considered to propose the reachability of the set $\mathbb{A}$ for robots with a minimum turning radius. The nonholonomic motion model will be used in this case.

6.4.2 Directional and omnidirectional backprojection

A directional backprojection corresponds to a backprojection of goal set at a defined direction. The strategy μ consists then to follow this direction despite the uncertainties to reach the goal

set, i.e. $\mu = \theta_d$ where θ_d is the desired direction. This strategy can be applied in the case of the holonomic motion model where the robot turns on itself to be in the direction θ_d before executing the motion for example. The motion model will be $\mathbf{f}_H$.

The directional backprojection of a set $\mathbb{A} \subset \mathbb{R}^3$ at a defined direction $\mu = \theta_d$ is defined as follows:

$$BACK(\mathbb{A}, \theta_d) = \{\mathbf{x} \in \mathbb{R}^3 | \forall \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, x_3 = \theta_d \text{ and } \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \tag{6.34}$$

Notice that the backward reach set is actually a subset of $\mathbb{R}^2$ as the direction (heading=x_3) is imposed at θ_d. The directional backprojection consists in finding the set:

$$BACK(\mathbb{A}, \theta_d) = \{(x_1, x_2) \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, x_3 = \theta_d \text{ and } \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \tag{6.35}$$

The omnidirectional backprojection corresponds to the 3D set:

$$BACK_{od}(\mathbb{A}) = \{\mathbf{x} \in \mathbb{R}^3 | \forall \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \tag{6.36}$$

where '*od*' means omnidirectional.

It is defined as the disjoint union of all directional backprojection for all possible values of θ_d in [187]. A directional backprojection is then a slice as in [186] at a given direction (heading). According to this representation it can be rewritten as follows:

$$BACK_{od}(\mathbb{A}) = \{(\theta_d, BACK(\mathbb{A}, \theta_d)) \in \mathbb{R} \times \mathbb{R}^2\} \tag{6.37}$$

The nondirectional backprojection [187][80] corresponds to the classical union of all backprojections for all direction θ_d:

$$BACK_{nd}(\mathbb{A}) = \bigcup_{\theta_d} BACK(\mathbb{A}, \theta_d) \tag{6.38}$$

where '*nd*' means nondirectional. Notice that the nondirectional backprojection is a subset of $\mathbb{R}^2$. It can be rewritten as:

$$\begin{aligned} BACK_{nd}(\mathbb{A}) &= \bigcup_{\theta_d} BACK(\mathbb{A}, \theta_d) && (6.39)\\ &= \{(x_1, x_2) \in \mathbb{X}_1 \times \mathbb{X}_2 | \exists x_3 \in \mathbb{X}_3,\ (x_1, x_2) \in BACK(\mathbb{A}, x_3)\} && (6.40)\\ &= proj^{\mathbb{X}_3}_{\mathbb{X}_1 \times \mathbb{X}_2}(BACK_{od}(\mathbb{A})) && (6.41) \end{aligned}$$

where $\mathbb{X}_1$, $\mathbb{X}_2$ and $\mathbb{X}_3$ denotes the domains for x_1, x_2 and x_3.

Similarly, the directional weak backprojection can be defined according to Equation 6.32:

$$WBACK(\mathbb{A}, \theta_d) = \{(x_1, x_2) \in \mathbb{R}^2 | \exists \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, x_3 = \theta_d \text{ and } \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \tag{6.42}$$

and the omnidirectional weak backprojection:

$$WBACK_{od}(\mathbb{A}) = \{\mathbf{x} \in \mathbb{R}^3 | \exists \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \tag{6.43}$$

Firstly, the backprojection of a single connected set will be developed where the projection on the uncertain parameter can be removed by considering the bounds of the error on the direction to save time computation. A simple intersection between two sets is performed. Secondly the backprojection of several sets connected or not will be explained where again the second projection can be removed.

6.4.2.1 Backprojection of a single connected set

The uncertain parameter $\mathbf{p}$ is defined as:

$$\mathbf{p} \in [\mathbf{p}] = [v_d(1-\alpha_v), v_d(1+\alpha_v)] \times [-\alpha_\theta, \alpha_\theta] \tag{6.44}$$

where v_d is the desired speed, α_v (expressed in %) corresponds to the rate of error on the speed and α_θ is the uncertainty on the direction that can come from the compass.

The motion model $\mathbf{f}_H$ corresponds to a translated polar parametric equation where:

$$\rho = vt \tag{6.45}$$

Since the backward reach set is defined when the existence of a time t is proved, the speed only influences when the goal can be reached but not on its reachability. It can then only have an impact on the goal recognizability and the termination condition.

Therefore, an uncertainty on the speed is not needed for the reachability of the goal set. The uncertain parameter vector $\mathbf{p}$ will then be defined by:

$$\mathbf{p} = (v, \delta\theta) \in [\mathbf{p}] = \{v_d\} \times [-\alpha_\theta, \alpha_\theta] \tag{6.46}$$

where $\delta\theta$ is the single uncertain parameter.

The set consistent with the omnidirectional backprojection of a closed and connected set $\mathbb{A}$ can be computed according to Equation 6.36 and Equation 6.23 with the use of double projection. The result of a slice at $\theta_d = 45°$ is given in Figure 6.5(a) for the reachability of a disk defined in the following example. It corresponds to $BACK(\mathbb{A}, 45°)$.

However, since the set $\mathbb{A}$ is a connected and closed set, the projection with respect to the uncertain parameter $\mathbf{p}$ on the domain $[\mathbf{p}]$ may be avoided by considering only the bounds of the single uncertain parameter $\delta\theta$ and finding the following set:

$$\begin{aligned} BACK_{od}(\mathbb{A}) &= \{\mathbf{x} \in \mathbb{R}^3 | \forall \delta_\theta \in \{-\alpha_\theta, \alpha_\theta\}, \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} && (6.47)\\ &= \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} \{\mathbf{x} \in \mathbb{R}^3 | \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} && (6.48)\\ &= \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj_{\mathbb{X}}^{\mathbb{T}}(\mathbf{f}_H^{-1}(\mathbb{A})) && (6.49) \end{aligned}$$

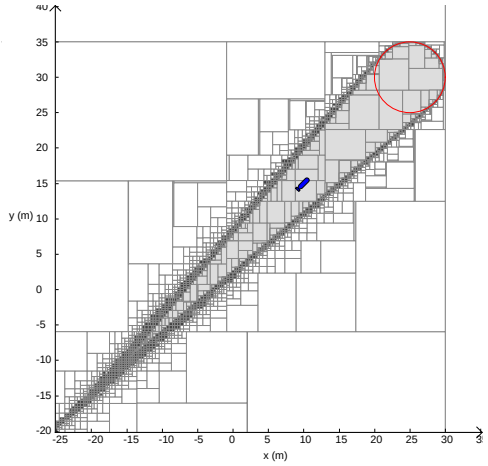
where $\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2 \times \mathbb{X}_3$ is the domain of (x, y, θ) and $\mathbb{T}$ is the domain of t. Indeed, all the possible trajectories are inside the ones defined at the bounds $\theta_d - \alpha_\theta$ and $\theta_d + \alpha_\theta$ of the uncertain direction $\theta_d = x_3$. The backward reach set is then simply obtained by taking the intersection between the two sets defined at the bounds of α_θ.

When the direction θ_d is defined, the 2D backward reach set in the $(x - y)$ plane is then expressed as follows:

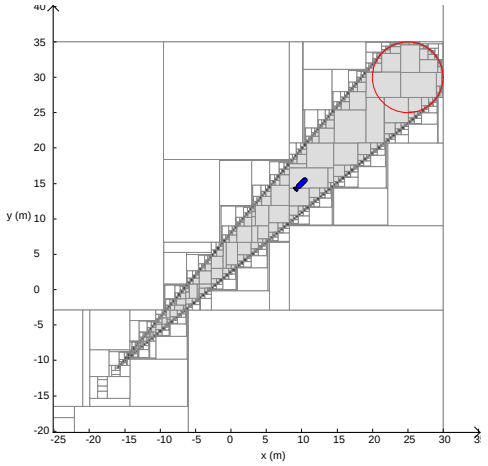
$$\begin{aligned} BACK(\mathbb{A}, \theta_d) &= \{\mathbf{x} \in \mathbb{R}^2 | \forall \delta_\theta \in \{-\alpha_\theta, \alpha_\theta\}, \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} && (6.50)\\ &= \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} \{\mathbf{x} \in \mathbb{R}^2 | \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} && (6.51)\\ &= \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj_{\mathbb{X}}^{\mathbb{T}}(\mathbf{f}_H^{-1}(\mathbb{A})) && (6.52) \end{aligned}$$

where $\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2$ is the domain of $(x-y)$, $\mathbb{T}$ is the domain of t and the direction θ_d is defined in the function $\mathbf{f}_H$.

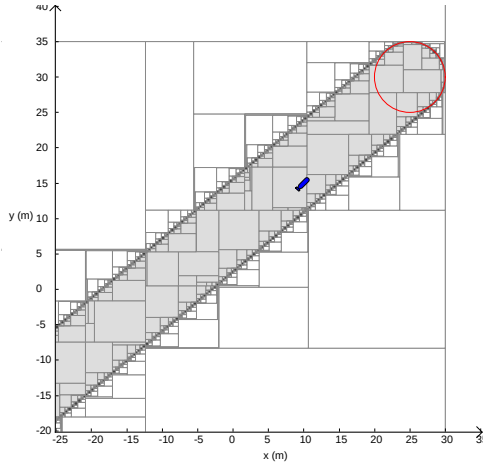
Avoiding the projection with respect to the uncertain parameter $\delta\theta$ means avoiding slicing in the parameter space and saving computation time as presented in Figure 6.5(b) at the same slice $\theta_d = 45°$. Notice how the number of unclassified boxes is reduced by working on the bounds. This figure is obtained by taking the intersection of the sets defined at the bounds of the direction error and depicted in Figure 6.5(c) for $\alpha_\theta = -5°$ and in Figure 6.5 for $\alpha_\theta = 5°$. The time computation is reduced by not slicing in the uncertain parameter and not loosing time to test small intervals along the uncertain parameter space. Any poses in the soft gray solution with a heading $x_3 = \theta_d = 45°$ will reach for sure the red disk despite the uncertainty on the direction. A sensor for example will indicate that it has reached this disk since the time t it reached may be different depending on the true error in the interval.



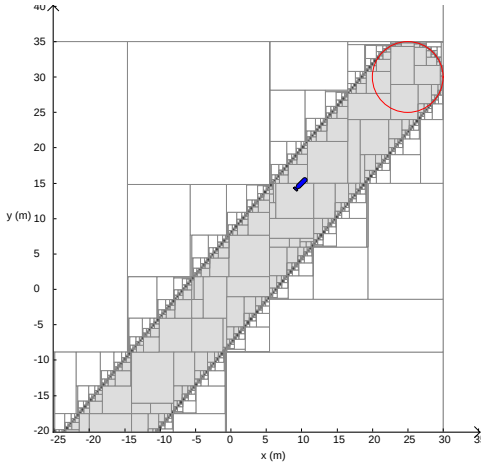
(a) Projection on $\delta\theta$ and on t (double projection).



(b) Simple intersection of the sets defined at the bounds of $\delta\theta$. Single projection on t.



(c) Backward reach set at $\theta_d = 45°$ and $\delta\theta = -\alpha_\theta$ (lower bound).



(d) Backward reach set at $\theta_d = 45°$ and $\delta\theta = \alpha_\theta$ (upper bound).

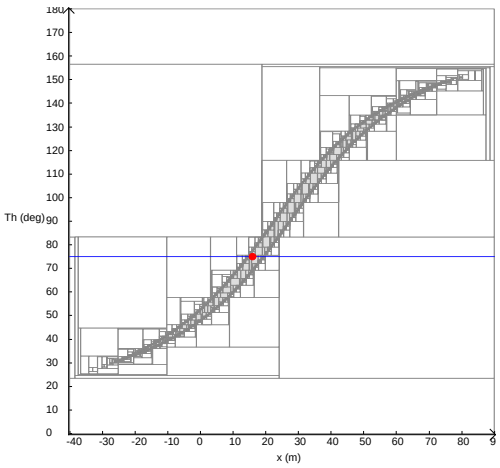
Figure 6.5: Comparison between double projection and single projection by working on the bounds of $\delta\theta$. Grayscale convention for classical thin sets. The direction is depicted by a blue AUV.

Example 1

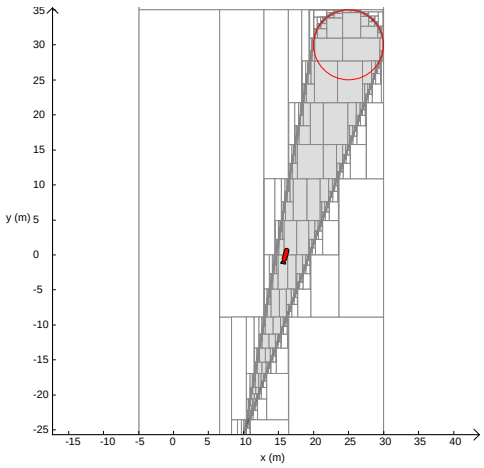
Consider only an uncertainty on the compass $\alpha_\theta = 5°$ and no uncertainty on the speed $\alpha_v = 0\%$. The desired speed v_d is fixed at $1m/s$. Therefore $[\mathbf{p}] = \{1\} \times [-5, 5](\frac{m}{s} \times deg)$ Define a set $\mathbb{A}$ as follows:

$$\mathbb{A} = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 25)^2 + (x_2 - 30)^2} \leq 5\} \tag{6.53}$$

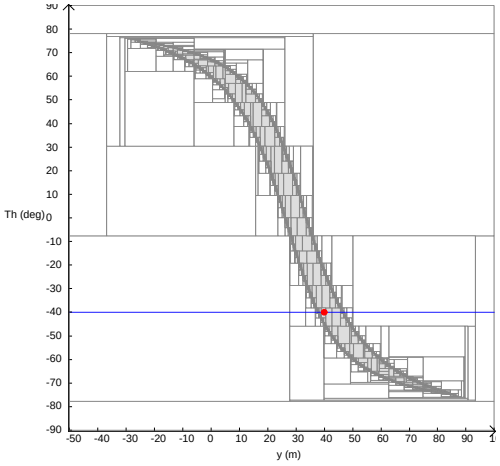
which corresponds to a disk centered at $(25, 30)$ with a radius at $5m$. As the set $\mathbb{A}$ is a subset of $\mathbb{R}^2$, the second model $\mathbf{f}_{H2}$ of the holonomic robot will be used. The angle of arrival in the relocation area $\mathbb{A}$ has no influence on the backward reach set in this case. Due to the 3D representation of the backward reach set, some slices are represented at different fixed values in Figure 6.6. The slice at $y = 0m$ is presented in Figure 6.6(a) where a blue line depicts the slice given in Figure 6.6(b) at $x_3 = \theta_d = 45°$. This latter is the representation of $BACK(\mathbb{A}, 45°)$ The red AUV corresponds to the red dot on the blue line in Figure 6.6(a). Similarly the result is present in Figure 6.6(c) and (d) with the slice at $x = 10m$ and the slice at $\theta_d = -40°$ giving $BACK(\mathbb{A}, -40°)$.



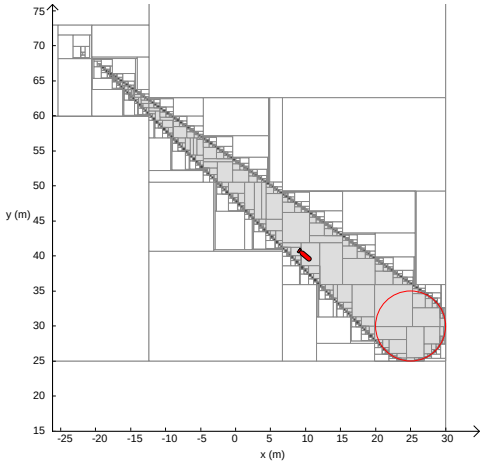
(a) (x, θ_d) representation for the slice at $y = 0m$.



(b) (x, y) (2D) representation for the slice at $\theta_d = 75°$ depicted by the blue line in the figure (a). A red AUV indicates the pose at the red dot in the figure (a).



(c) (y, θ_d) representation for the slice at $x = 10m$.



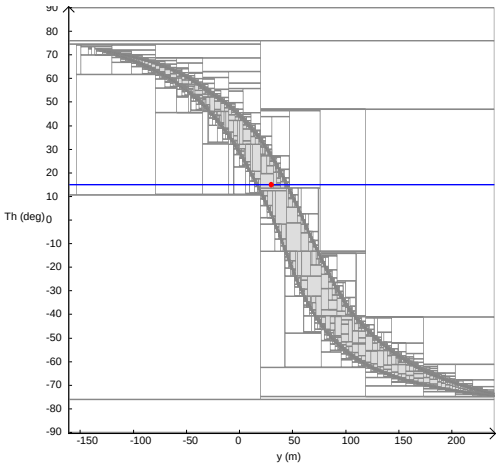
(d) (x, y) (2D) representation for the slice at $\theta_d = -40°$ depicted by the blue line in the figure (c). A red AUV indicates the pose at the red dot in the figure (c).

Figure 6.6: Backward reach set of the set $\mathbb{A}$ defined by a 2D disk (in red).

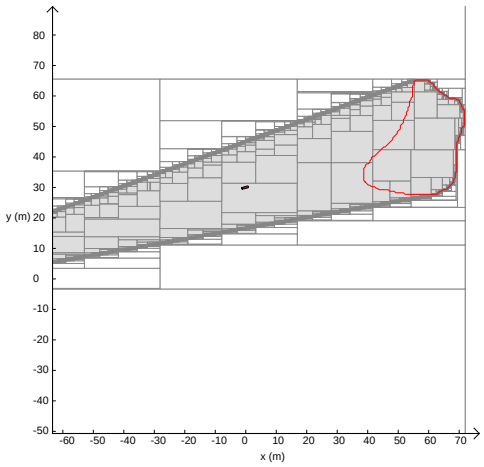
Example 2

Consider again the shape landmark presented in the document (the thin set and not the thick set). The omnidirectional backward reach set can obviously be computed according to this unstructured relocation area since any equations needs to be computed to describe the shape. Some slices of the omnidirectional backward reach set are presented in Figure 6.7. Firstly, the slice at $x = 0m$ is presented where the blue line and the red dot depict respectively the slice and the red AUV poses given in Figure 6.7(b). This solution set looks similar to the previous ones. In Figure 6.7(c), the slice at $y = 50m$ is provided and it can be noticed a continuity in the heading at some x coordinates and the fact that the landmark is reachable at any heading. Indeed, the slice at $y = 50m$ crosses the relocation area defined by the shape and this particularity corresponds to the poses that are actually inside the shape. Obviously when the position of the AUV is already inside the landmark, any heading ($x_3 = \theta_d$) is possible since

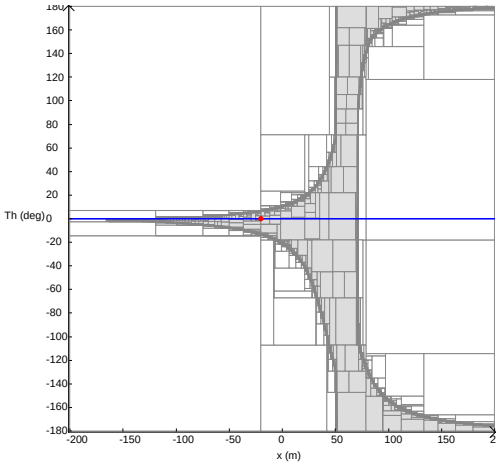
the shape defines only a constraint on (x, y).



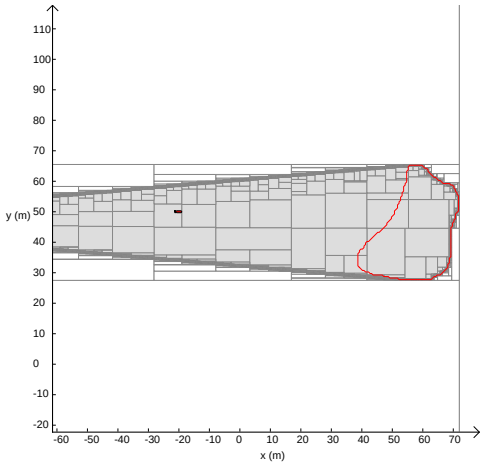
(a) (y, θ_d) representation for the slice at $x = 0m$.



(b) (x, y) (2D) representation for the slice at $\theta_d = 15°$ depicted by the blue line in the figure (a). A red AUV indicates the pose at the red dot in the figure (a).



(c) (x, θ_d) representation for the slice at $y = 50m$.



(d) (x, y) (2D) representation for the slice at $\theta_d = 0°$ depicted by the blue line in the figure (c). A red AUV indicates the pose at the red dot in the figure (c).

Figure 6.7: Backward reach set of the shape $\mathbb{A}$ defined by a red contour.

Remark

In the previous work [187][186], the exact backprojection is computed based on disk goal sets. In this thesis, the work is extended to any shape goal sets based on the backward reach set using interval analysis.

6.4.2.2 Backprojection of several sets

Consider now that the goal set is defined by two sets $\mathbb{A} \subset \mathbb{R}^3$ and $\mathbb{B} \subset \mathbb{R}^3$. They can be only subsets of $\mathbb{R}^2$ as in the previous example, then consider the model $\mathbf{f}_{H2}$ instead of $\mathbf{f}_H$. The theory is the same in both case. Indeed if the set $\mathbb{A}$ is a subset of $\mathbb{R}^2$ then consider $\mathbb{A} \times \mathbb{R}$ instead of

just $\mathbb{A}$, then use again the model $\mathbf{f}_H$. For ease of notation, $\mathbf{f}_{H2}$ was introduced. The angle of arrival in this case has no influence on the backprojection. Each set is connected but the union may not be connected. The backprojection can be given by Equation 6.23 using the double projection even if the two sets are not connected, i.e. the intersection $\mathbb{A} \cap \mathbb{B}$ is empty. However, due to the reasons explained before, computing this double projection can be time consuming. Therefore considering again the bounds of the error on the direction could improve the computation of the backprojection, but it has to be handled carefully. This will be discussed just below.

Consider now only the backprojection using the bounds as shown before.

If the two sets are connected, i.e. the intersection $\mathbb{A} \cap \mathbb{B}$ is not empty, then the backprojection of the two sets is simply given by taking the backprojection of the union of the two sets. That means $\mathbb{A}$ in Equation 6.49 becomes here $\mathbb{A} \cup \mathbb{B}$ which leads to:

$$BACK_{od}(\mathbb{A} \cup \mathbb{B}) = \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj_{\mathbb{X}}^{\mathbb{T}}(\mathbf{f}_H^{-1}(\mathbb{A} \cup \mathbb{B})) \tag{6.54}$$

An example is provided in **Example 1**.

If the two sets are not connected, i.e. the intersection $\mathbb{A} \cap \mathbb{B}$ is empty, the problem is a little bit more complicated.

Consider the following set:

$$\mathbb{X}_{\mathbb{A} \to \mathbb{B}} = \{\mathbf{x} \in \mathbb{R}^3 | BACK_{od}(\mathbb{B}) \cap \mathbb{A} \neq \emptyset\} \tag{6.55}$$

It indicates the set of robot poses from where it is possible to leave $\mathbb{A}$ and reach $\mathbb{B}$ for sure by following a particular direction. The definition of this set will be highly used in the next section. It can be rewritten as follows:

$$\mathbb{X}_{\mathbb{A} \to \mathbb{B}} = \{\mathbf{x} \in \mathbb{A} | \forall \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B}\} \tag{6.56}$$

Similarly consider:

$$\mathbb{X}_{\mathbb{B} \to \mathbb{A}} = \{\mathbf{x} \in \mathbb{R}^3 | BACK_{od}(\mathbb{A}) \cap \mathbb{B} \neq \emptyset\} \tag{6.57}$$

Finally consider the global set that links $\mathbb{A}$ and $\mathbb{B}$:

$$\begin{aligned} \mathbb{X}_{\mathbb{A} \leftrightarrow \mathbb{B}} &= \mathbb{X}_{\mathbb{B} \to \mathbb{A}} \cup \mathbb{X}_{\mathbb{B} \to \mathbb{A}} && (6.58) \\ &= \{\mathbf{x} \in \mathbb{R}^3 | BACK_{od}(\mathbb{B}) \cap \mathbb{A} \neq \emptyset \vee BACK_{od}(\mathbb{A}) \cap \mathbb{B} \neq \emptyset\} && (6.59) \end{aligned}$$

Considering this motion model, the backprojection of $\mathbb{A} \cup \mathbb{B}$ can be computed according to two cases:
Firstly if $\mathbb{X}_{\mathbb{A} \leftrightarrow \mathbb{B}} \neq \emptyset$:

$$BACK_{od}(\mathbb{A} \cup \mathbb{B}) = \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj_{\mathbb{X}}^{\mathbb{T}}(\mathbf{f}_H^{-1}(\mathbb{A} \cup \mathbb{B})) \tag{6.60}$$

where $\mathbb{X} \subset \mathbb{R}^3$ corresponds to the domain of $(x, y, \theta) = (x_1, x_2, x_3)$ and $\mathbb{T}$ to the temporal domain.
Secondly if $\mathbb{X}_{\mathbb{A} \leftrightarrow \mathbb{B}} = \emptyset$:

$$\begin{aligned} BACK_{od}(\mathbb{A} \cup \mathbb{B}) &= \Big(\bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj_{\mathbb{X}}^{\mathbb{T}}(\mathbf{f}_H^{-1}(\mathbb{A}))\Big) \bigcup \Big(\bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj_{\mathbb{X}}^{\mathbb{T}}(\mathbf{f}_H^{-1}(\mathbb{B}))\Big) && (6.61) \\ &= \bigcup_{\mathbb{Y} \in \{\mathbb{A}, \mathbb{B}\}} proj_{\mathbb{X}}^{\mathbb{T}}(\mathbf{f}_H^{-1}(\mathbb{Y})) && (6.62) \\ &= BACK_{od}(\mathbb{A}) \cup BACK_{od}(\mathbb{B}) && (6.63) \end{aligned}$$

This second case means that no link exists between $\mathbb{A}$ and $\mathbb{B}$, and it corresponds simply to the union of the backprojection. In the first case, due to the existence of some directions between $\mathbb{A}$ and $\mathbb{B}$, the backprojection can be computed directly as previously.

Proving that $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} \subset \mathbb{R}^3$ ($\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$ or $\Theta_{\mathbb{B}\leftrightarrow\mathbb{A}}$) is empty may be time consuming, and it is hard to have an idea of the 2D backward reach set in the $(x-y)$ plane according to the set of directions. Due to the 3D representation, a simpler version is proposed now, again on two different cases.

Denotes as $\Theta_{\mathbb{A}\to\mathbb{B}}$ the set of directions such that it is possible to leave $\mathbb{A}$ and reach $\mathbb{B}$. This set is defined as follows:

$$\Theta_{\mathbb{A}\to\mathbb{B}} = \{\theta_d \in \mathbb{R} | BACK(\mathbb{B}, \theta_d) \cap \mathbb{A} \neq \emptyset\} \tag{6.64}$$

where $BACK(\mathbb{B}, \theta_d)$ was defined in Equation 6.52. This is similar to Equation 6.54 where it only deals with the directions.

This equation corresponds to:

$$\begin{aligned}
\Theta_{\mathbb{A}\to\mathbb{B}} &= \{x_3 | \exists (x_1, x_2) \in \mathbb{X}_1 \times \mathbb{X}_2, \forall \delta_\theta \in \{-\alpha_\theta, \alpha_\theta\}, \exists t \in \mathbb{R}^+, \\
&\quad \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B} \text{ and } \mathbf{x} \in \mathbb{A}\} && (6.65)\\
&= proj_{\mathbb{X}_3}^{\mathbb{X}_1\times\mathbb{X}_2}\Big(\big(\bigcap_{\delta\theta\in\{-\alpha_\theta,\alpha_\theta\}} proj_{\mathbb{X}}^{\mathbb{T}}(\mathbf{f}_H^{-1}(\mathbb{B})) \big) \cap \mathbb{A} \Big) && (6.66)\\
&= proj_{\mathbb{X}_3}^{\mathbb{X}_1\times\mathbb{X}_2}\Big(\bigcap_{\delta\theta\in\{-\alpha_\theta,\alpha_\theta\}} proj_{\mathbb{X}}^{\mathbb{T}}(\mathbf{f}_H^{-1}(\mathbb{B})) \cap \mathbb{A} \Big) && (6.67)
\end{aligned}$$

where $\mathbb{X}_1$, $\mathbb{X}_2$ and $\mathbb{X}_3$ represent respectively the domains of x_1, x_2 and $x_3 = \theta_d$. This comes from the definition given in Equation 6.49. Notice that it has to handle again a double projection, but it is necessary to compute 1D or 2D sets to have visual results and better understand the approach since 3D sets are manipulated. Moreover the 2D backward reach set (in the $(x-y)$ plane) defined on bounds depends on the set of directions that links the two sets.

According to Equation 6.49, the set defined in Equation 6.67 refers also to:

$$\begin{aligned}
\Theta_{\mathbb{A}\to\mathbb{B}} &= proj_{\mathbb{X}_3}^{\mathbb{X}_1\times\mathbb{X}_2}\Big(BACK_{od}(\mathbb{B}) \cap \mathbb{A} \Big) && (6.68)\\
&= proj_{\mathbb{X}_3}^{\mathbb{X}_1\times\mathbb{X}_2}\Big(\mathbb{X}_{\mathbb{A}\to\mathbb{B}} \Big) && (6.69)
\end{aligned}$$

Similarly, it can be defined the following set:

$$\Theta_{\mathbb{B}\to\mathbb{A}} = \{\theta_d | BACK(\mathbb{A}, \theta_d) \cap \mathbb{B} \neq \emptyset\} \tag{6.70}$$

where a computation is possible as in Equation 6.67 by inverting $\mathbb{A}$ and $\mathbb{B}$.

Finally, the global set of directions to transit between $\mathbb{A}$ and $\mathbb{B}$ is given as follows:

$$\begin{aligned}
\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} &= \Theta_{\mathbb{B}\to\mathbb{A}} \cup \Theta_{\mathbb{B}\to\mathbb{A}} && (6.71)\\
&= \{\theta_d | BACK(\mathbb{B}, \theta_d) \cap \mathbb{A} \neq \emptyset \vee BACK(\mathbb{A}, \theta_d) \cap \mathbb{B} \neq \emptyset\} && (6.72)
\end{aligned}$$

This is similar to Equation 6.59, but it only handles the direction to transit between $\mathbb{A}$ and $\mathbb{B}$. It also corresponds to:

$$\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} = proj_{\mathbb{X}_3}^{\mathbb{X}_1\times\mathbb{X}_2}(\mathbb{X}_{\mathbb{A}\leftrightarrow\mathbb{B}}) \tag{6.73}$$

The backprojection of $\mathbb{A} \cup \mathbb{B}$ in the (x, y) plane (2D backprojection) can be computed according to two cases:
Firstly if $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} = \emptyset$, the backprojection in the $(x - y)$ plane is given as follows:

$$BACK(\mathbb{A} \cup \mathbb{B}, \theta_d) = BACK(\mathbb{A}, \theta_d) \cup BACK(\mathbb{B}, \theta_d) \tag{6.74}$$

No link exists between $\mathbb{A}$ and $\mathbb{B}$.

Secondly, if $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} \neq \emptyset$, the backprojection in the (x, y) plane can be computed according to two subcases.

If $\theta_d \in \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$:

$$BACK(\mathbb{A} \cup \mathbb{B}, \theta_d) = \bigcap_{\delta\theta\in\{-\alpha_\theta,\alpha_\theta\}} proj^{\mathbb{T}}_{\mathbb{X}_1\times\mathbb{X}_2}(\mathbf{f}_H^{-1}(\mathbb{A} \cup \mathbb{B})) \tag{6.75}$$

with x_3 defined at θ_d, $\mathbb{X}_1$ the domain of x_1 and $\mathbb{X}_2$ the domain of x_2. Since the existence of some directions between $\mathbb{A}$ and $\mathbb{B}$, the computation is similar to Equation 6.54 with the difference that $\mathbb{X}$ is only along (x, y) dimension now as the direction is defined.

If $\theta_d \notin \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$:

$$BACK(\mathbb{A} \cup \mathbb{B}, \theta_d) = BACK(\mathbb{A}, \theta_d) \cup BACK(\mathbb{B}, \theta_d) \tag{6.76}$$

The second subcase of the second case can be grouped with the first case because if $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$ is empty then for all θ_d it does not belong to the set $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$. Finally, the two cases are:

If $\theta_d \in \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$:

$$BACK(\mathbb{A} \cup \mathbb{B}, \theta_d) = \bigcap_{\delta\theta\in\{-\alpha_\theta,\alpha_\theta\}} proj^{\mathbb{T}}_{\mathbb{X}_1\times\mathbb{X}_2}(\mathbf{f}_H^{-1}(\mathbb{A} \cup \mathbb{B})) \tag{6.77}$$

If $\theta_d \notin \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$:

$$BACK(\mathbb{A} \cup \mathbb{B}, \theta_d) = BACK(\mathbb{A}, \theta_d) \cup BACK(\mathbb{B}, \theta_d) \tag{6.78}$$

This second case includes the fact that $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$ may be empty.

As it can be noticed, the computation of $\Theta_{\mathbb{A}\to\mathbb{B}}$ (or $\Theta_{\mathbb{B}\to\mathbb{A}}$) in Equation 6.67 requires a second projection to get the set of directions with respect to the $(x - y)$ plane. The dimension of the 3D backward reach set is then reduced to a single dimension along the space of directions. If the 3D backward reach set was computed with the bisections in the parameter space, then a triple projection would have been needed to get the set of directions and to prove that some links exist between $\mathbb{A}$ and $\mathbb{B}$. Working on the bounds even with this second projection with respect to the $(x - y)$ plane enables to compute the set of directions and to prove that there exists some links between $\mathbb{A}$ and $\mathbb{B}$.

Example 1

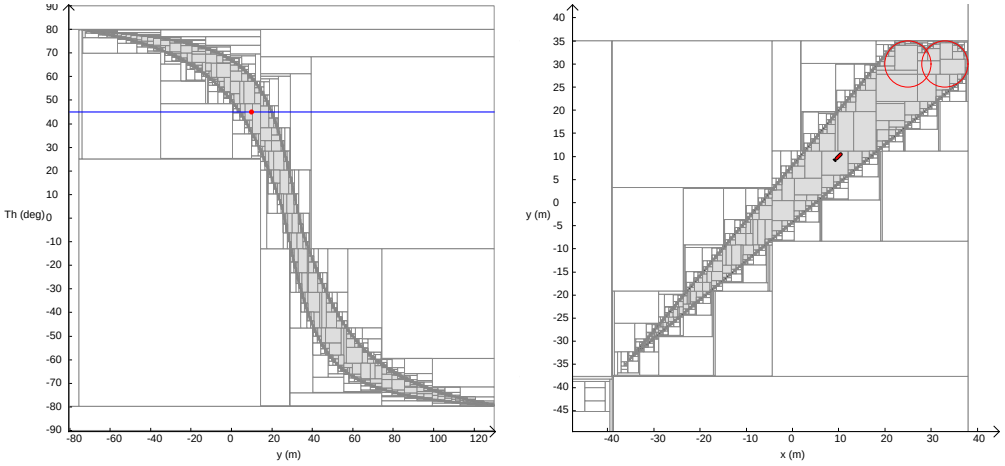
Consider again an uncertainty on the compass $\alpha_\theta = 5°$ and no uncertainty on the speed. The desired speed is fixed at $1m/s$. Consider again the set $\mathbb{A}$ defined by the disk:

$$\mathbb{A} = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 25)^2 + (x_2 - 30)^2} \leq 5\} \tag{6.79}$$

and now consider another disk as a set $\mathbb{B}$:

$$\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 33)^2 + (x_2 - 30)^2} \leq 5\} \tag{6.80}$$

As the sets $\mathbb{A}$ and $\mathbb{B}$ are subsets of $\mathbb{R}^2$, the model used will be $\mathbf{f}_{H2}$. The intersection of the two disks $\mathbb{A}$ and $\mathbb{B}$ is not empty, so they are connected as shown in Figure 6.8(b) by the two red circles. The backprojection of the union of the sets is therefore given by Equation 6.54 as depicted in Figure 6.8



(a) (y, θ_d) representation for the slice at $x = 10m$.

(b) (x, y) (2D) representation for the slice at $\theta_d = 45°$ depicted by the blue line in the figure (a). A red AUV indicates the pose at the red dot in the figure (a).

Figure 6.8: Backward reach set of connected sets $\mathbb{A}$ and $\mathbb{B}$ defined by red contours. Grayscale convention for thin sets.

Example 2

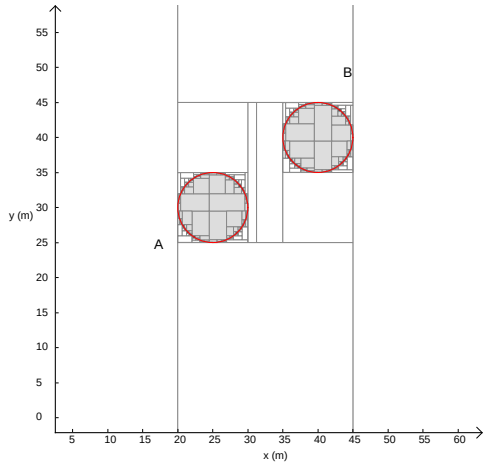
Consider the same example, but modify the set $\mathbb{B}$ as follows:

$$\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 40)^2 + (x_2 - 40)^2} \leq 5\} \tag{6.81}$$

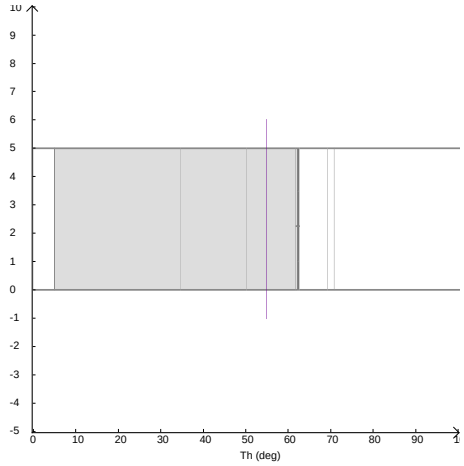
The two sets are depicted in Figure 6.9(a) where it can be noticed that now the intersection between the two disks is empty, so they are not connected. Firstly the set of directions linking $\mathbb{A}$ to $\mathbb{B}$ expressed in Equation 6.64 and computed according to Equation 6.67 is shown in Figure 6.9(b) in a 2D representation. It can be noticed that $\Theta_{\mathbb{A}\to\mathbb{B}}$ is not empty. Due to the 1D dimension of the set of directions, a 2D representation is proposed to have a visual aspect of the solution set meaning that the values along the $y-$axis do not matter, only the $x-$values of the boxes are important. The soft gray boxes are inside the solution set, or in other words, for this set of directions there exists some locations in the set $\mathbb{A}$ that enable to go for sure in $\mathbb{B}$ with a direction inside this set. For example, one direction is selected at $x_3 = \theta_d = 55°$ depicted by the blue line in Figure 6.9(b). The backward reach set in the $(x - y)$ plane at this direction is proposed in Figure 6.9(c). Since $\theta_d \in \Theta_{\mathbb{A}\to\mathbb{B}} \subset \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} \neq \emptyset$, then the 2D backward reach set is computed according to Equation 6.77.

If the two sets were considered separately, the union of the backward reach set at this direction would have given the set in Figure 6.9(d). The set is smaller than the previous one. Notice that the blue AUV in Figure 6.9(d) at this specified direction ($\theta_d = 55°$) does not seem to be able to join the two disks for sure at it lays outside the solution set. However, according to the result in Figure 6.9(c) for the same pose it is proved that it will reach at a certain time one of the two disks according to the theory developed above. Indeed considering the two goal

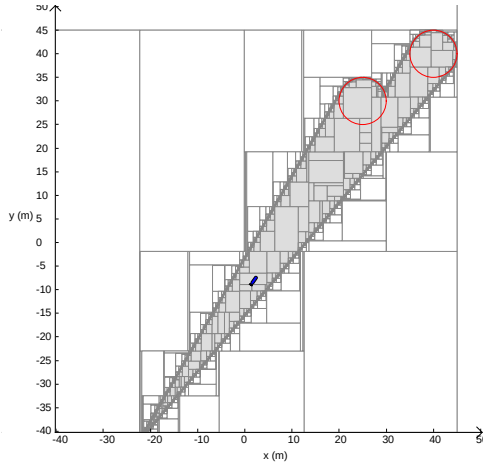
sets as a global goal set enables to travel further. One direction outside the set of directions is depicted by a red line in Figure 6.9(e) where the 2D representation is proposed in Figure 6.9(f). Since $\theta_d = 2^\circ \notin \Theta_{\mathbb{A}\to\mathbb{B}} \neq \emptyset$, the 2D backward reach set is computed according to Equation 6.78. Indeed, at some locations, it is possible that some directions to follow may go between the two disks and miss the goal set.



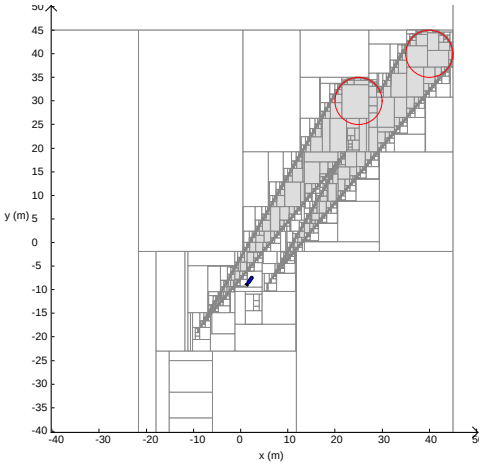
(a) Goal sets $\mathbb{A}$ and $\mathbb{B}$.



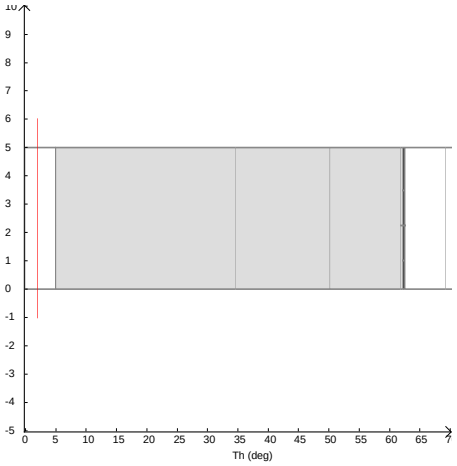
(b) Solution $\Theta_{\mathbb{A}\to\mathbb{B}}$. Only the $x-$axis is important, it indicates the set of directions.



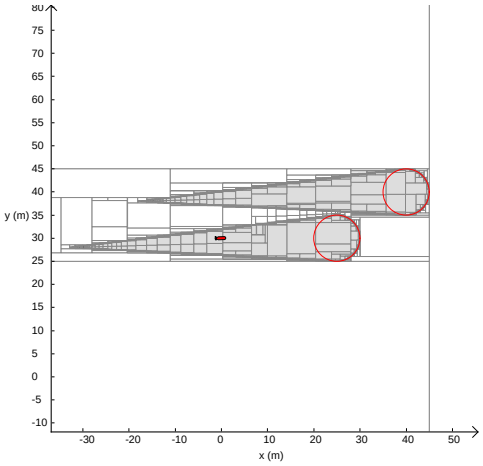
(c) Backward reach set in the (x, y) plane at $\theta_d = 55°$. This direction is depicted by the blue line in the figure (b).



(d) Union of the backward reach taken separately at this same direction. The blue AUV does not belong to the backward reach set contrary to the one in the figure (c) at the same pose.



(e) Solution $\Theta_{\mathbb{A}\to\mathbb{B}}$ with a wrong direction selected in red. Only the $x-$values are relevant.

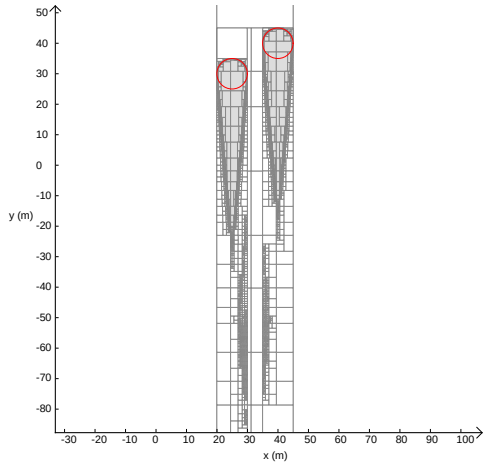


(f) 2D backward reach set for the wrong (red) selected direction.

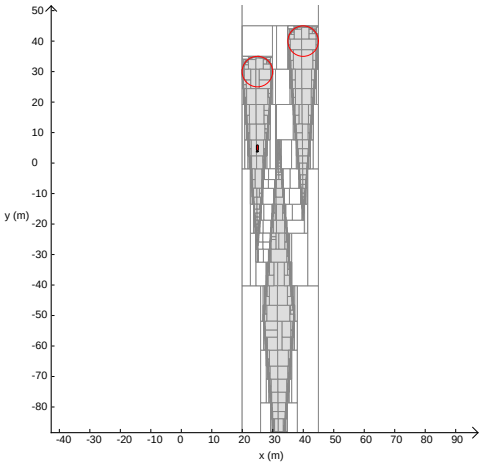
Figure 6.9: Backward reach set of disconnected sets $\mathbb{A}$ and $\mathbb{B}$ defined by red contours. Grayscale convention for thin sets.

Remark 1

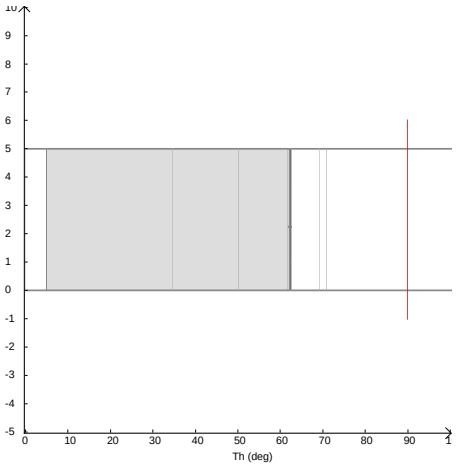
As explained before, considering the double projection with bisections in the parameter space, the 2D backward reach set in this example would have lead the result depicted in Figure 6.10(a) for a direction defined at 90°. At this direction, it can be guessed that the backward reach set corresponds to the union of the backward reach set since the disks are slightly separated along the $x-$axis. The projection with the bisections in the parameter space seems to have difficulties to determine this 2D backward reach set. Considering the goal set as the union of the disks and computing directly the backward reach set as proposed in Equation 6.54 by working only on the bounds gives the result in Figure 6.10(b). Notice that a wrong set is added below the backward reach set. This comes from the fact that the goal set was defined by the two disks and due to the computation only at the bounds of the error on the directions will lead to this artefact. This latter was not present when using the double projection. However, when looking at the set of directions linking $\mathbb{A}$ to $\mathbb{B}$ (same solution set as in the example), this direction $\theta_d = 90°$, depicted by the red line in Figure 6.10(c), is indeed outside the solution ($90° \notin \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} \neq \emptyset$) proving that the backward reach set of this goal set (defined by the two disks) is actually the union of the two backward reach sets depicted in Figure 6.10(d). Notice that the result of the 2D backward reach set is well defined since the direction is valid. Working on the bounds improves the speed of the computation of the 2D backward reach set but it requires firstly to compute the set of directions, which depends on a double projection too.



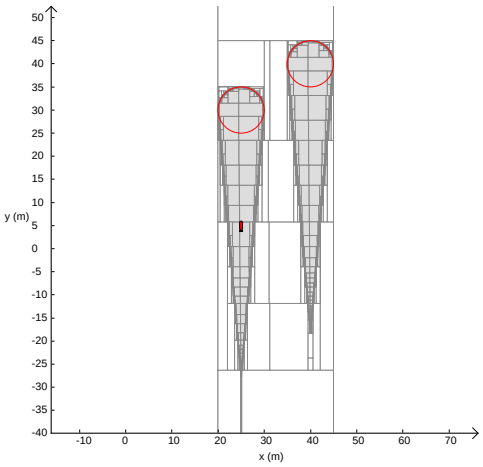
(a) Backward reach set in the (x, y) plane with bisections in the parameter space at the direction $\theta_d = 90°$.



(b) Backward reach set of the goal set defined by the two disks at the same direction and working on the bounds, generated according to Equation 6.54. An artefact has appeared.



(c) Solution $\Theta_{\mathbb{A}\to\mathbb{B}}$ with the proof that this direction is not valid. The red line is outside the solution set.



(d) 2D backward reach set given by the union of the backward reach set since $\theta_d \notin \Theta_{\mathbb{A}\to\mathbb{B}}$.

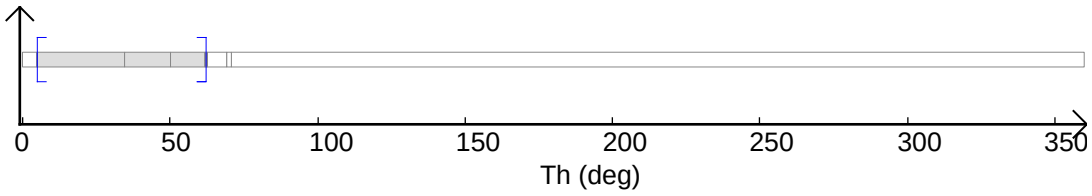
Figure 6.10: Backward reach set of disconnected sets $\mathbb{A}$ and $\mathbb{B}$ defined by red contours at the direction defined at 90°. Grayscale convention for thin sets.

Remark 2

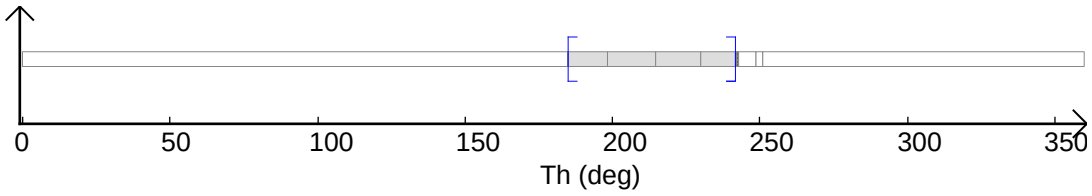
The set of directions linking $\mathbb{A}$ and $\mathbb{B}$ can be interesting to know to have an idea about the directions to impose to the robot to navigate safely between these two relocation areas. As the set computation is based on separators, it provides an inner and outer approximation. Taking the inner approximation guarantees the inter-reachability as shown in Figure 6.11. Figure 6.11(a) depicts the set of directions $\Theta_{\mathbb{A}\to\mathbb{B}}$ such that it is possible to leave $\mathbb{A}$ and reach $\mathbb{B}$ for sure. It can be enclosed in several intervals called $[\Theta_{\mathbb{A}\to\mathbb{B}}]_i$. Here only a single interval is sufficient to enclose the solution that comes from the inner approximation as shown by the blue enclosure. This latter depends on the ϵ (*epsilon*) parameter that is used for the **SIVIA** computation. The initial interval for $[\Theta_{\mathbb{A}\to\mathbb{B}}]$ was $[0, 360](deg)$. Similarly the set $\Theta_{\mathbb{B}\to\mathbb{A}}$ in Figure 6.11(b) indicates the directions to leave $\mathbb{B}$ and reach $\mathbb{A}$. Finally the set of directions $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$ that links $\mathbb{A}$ and $\mathbb{B}$ according to Equation 6.71 is proposed in Figure 6.11(c) where the solution

is decomposed into two intervals represented in blue. Notice that both sets of directions, i.e. $\Theta_{\mathbb{A}\to\mathbb{B}}$ and $\Theta_{\mathbb{B}\to\mathbb{A}}$ are not empty.

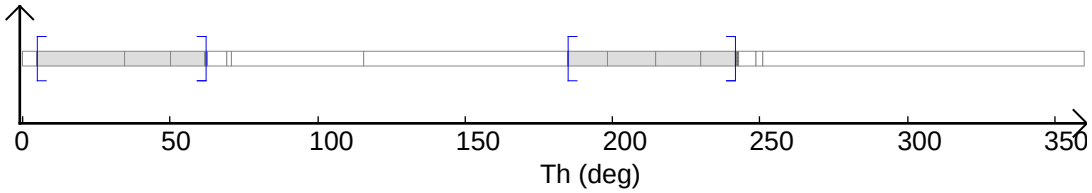
The set of directions to leave $\mathbb{A}$ and reach $\mathbb{B}$ is enclosed in the interval $[\Theta_{\mathbb{A}\to\mathbb{B}}] = [\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}]_1 = [5.03, 62.26](deg)$. The set of directions to leave $\mathbb{B}$ and reach $\mathbb{A}$ is enclosed in the interval $[\Theta_{\mathbb{B}\to\mathbb{A}}] = [\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}]_2 = [185.02, 241.74](deg)$. Both are only described by a single interval. Finally the directions linking $\mathbb{A}$ and $\mathbb{B}$ for sure belongs to one of the two intervals $[\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}]_1$ or $[\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}]_2$. The directions linking $\mathbb{A}$ and $\mathbb{B}$ are then enclosed in $[\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}]_i \in \{[\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}]_1, [\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}]_2\}$. It was initially only defined by a single one $[0, 360](deg)$. Whatever the direction inside these two intervals, the computation of the 2D backward reach set (in the (x, y) plane) can be computed according to Equation 6.77. Outside these intervals, Equation 6.78 would provide the backward reach set which corresponds in this case to the union of the backward reach set.



(a) Solution of $\Theta_{\mathbb{A}\to\mathbb{B}}$.



(b) Solution of $\Theta_{\mathbb{B}\to\mathbb{A}}$.



(c) Solution of $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} = \Theta_{\mathbb{A}\to\mathbb{B}} \cup \Theta_{\mathbb{B}\to\mathbb{A}}$.

Figure 6.11: Different sets of direction linking $\mathbb{A}$ and $\mathbb{B}$. Only the values on the $x-$axis are relevant. The blue intervals indicate the inner approximation. Grayscale convention for thin sets.

What about more sets ?

The goal set was firstly defined by a single connected set, then it has been extended to two sets that may be not connected. When dealing with two sets, the computation of the backward reach set when the sets were connected was straightforward by considering the union of the sets as the new set. However when the sets are not connected, the problem was a little bit more difficult. Working on the bounds of the uncertain parameter (error on the direction), the set of directions linking two sets $\mathbb{A}$ and $\mathbb{B}$ has been computed based on the intersection of the backward reach set of $\mathbb{A}$ on the set $\mathbb{B}$ and vice versa. These sets could be seen as intervals on which it is guaranteed to navigate between $\mathbb{A}$ and $\mathbb{B}$, as depicted by blue intervals in Figure 6.11 where it indicates the inner approximation. Outside these intervals, the backward reach set

is defined as the union of the backward reach set (it includes the case when the set $\Theta_{\mathbb{AB}}$ is empty).

Consider now a third set $\mathbb{C}$.

If the union of the sets $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$ is connected then the backward reach set is straightforward and can be computed by considering the union of the sets.

If the union of the sets $\mathbb{A}$ and $\mathbb{B}$ is connected but $\mathbb{C}$ is not connected to it then the backward reach set is similar to the previous case where it manipulated only two sets.

If the sets $\mathbb{A}$ and $\mathbb{B}$ are not connected, but the set $\mathbb{C}$ is connected either to $\mathbb{A}$ or $\mathbb{B}$ then the problem is straightforward by considering a global set with the one that intersects the set $\mathbb{C}$. In other words, if the intersection $\mathbb{C} \cap \mathbb{A}$ is not empty, then consider $\mathbb{A} \cup \mathbb{C}$ instead of $\mathbb{A}$.

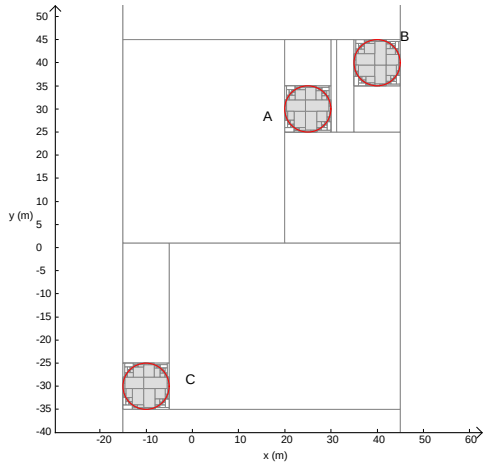
If all the sets are disconnected to each other, the problem is again a little bit more complicated as before.

Consider the previous sets $\mathbb{A}$ and $\mathbb{B}$ defined by the two disks and a new set $\mathbb{C}$ defined as follows:

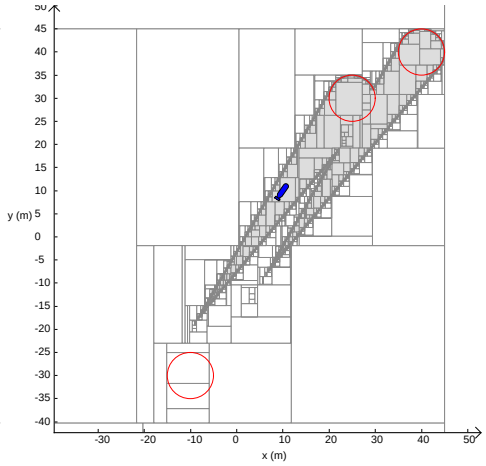
$$\mathbb{C} = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - (-10))^2 + (x_2 - (-30))^2} \leq 5\} \tag{6.82}$$

This environment is represented in Figure 6.12(a). Notice that it is indeed disconnected. Since the set of directions linking $\mathbb{A}$ and $\mathbb{B}$ was already computed $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$ and enclosed in $[\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}]_i \in \{[\Theta_{\mathbb{A}\rightarrow\mathbb{B}}], [\Theta_{\mathbb{B}\rightarrow\mathbb{A}}]\}$, the problem consists now to determine if it is possible to connect directly the three sets $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$ by working on these intervals that link $\mathbb{A}$ and $\mathbb{B}$. In other words, is it possible to determine a bigger backward reach set such that if the robot starts from these locations at some defined directions it will for sure end in either $\mathbb{A}$, $\mathbb{B}$ or $\mathbb{C}$? Is it possible to leave $\mathbb{C}$ and reach $\mathbb{A}$ or $\mathbb{B}$ for sure?

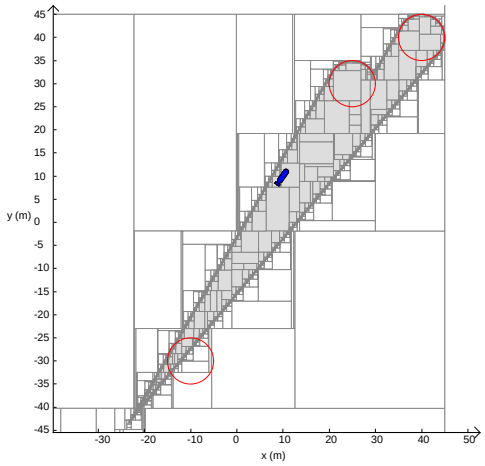
It can be guessed that at the direction $\theta_d = 55°$ ($\theta_d \in \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$), it is possible to reach $\mathbb{A}$ or $\mathbb{B}$ starting from some locations inside $\mathbb{C}$ as depicted in Figure 6.12(c). However, if the goal set composed of $\mathbb{A}$ and $\mathbb{B}$ is considered separately, meaning that $\mathbb{A}$ and $\mathbb{B}$ are considered separately, then the backward reach set, corresponding to the union of the backward reach set, does not enable to start from $\mathbb{C}$ and reach $\mathbb{A}$ or $\mathbb{B}$ as shown in Figure 6.12(b). Considering the goal set as the union of some sets on some intervals of direction that are proved to have some links can extend the set of initial robot locations.



(a) Goal set defined by three non connected sets $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$.



(b) Union of the backward reach set of $\mathbb{A}$ and $\mathbb{B}$ at the direction 55°.



(c) Backward reach set of $\mathbb{A} \cup \mathbb{B}$ at this same direction since $55° \in \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$.

Figure 6.12: Comparison of the backward reach of $\mathbb{A}$ and $\mathbb{B}$ onto $\mathbb{C}$. Grayscale convention for thin sets.

If $\theta_d \in [\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}]_i$, the 2D backward reach set of $\mathbb{A} \cup \mathbb{B}$ is defined according to Equation 6.77.

The set of directions to transit from $\mathbb{C}$ to a goal set defined by $\mathbb{A}$ and $\mathbb{B}$ is determined similarly as in Equation 6.64:

$$\Theta_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} = \{\theta_d \in \mathbb{R} | BACK(\mathbb{A} \cup \mathbb{B}, \theta_d) \cap \mathbb{C} \neq \emptyset\} \tag{6.83}$$

From the potential link between $\mathbb{A}$ and $\mathbb{B}$ this set can be decomposed as follows:

$$\begin{aligned}
\Theta_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} &= \{\theta_d \in \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} | BACK(\mathbb{A} \cup \mathbb{B}, \theta_d) \cap \mathbb{C} \neq \emptyset\} \\
&\cup \{\theta_d \notin \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} | BACK(\mathbb{A} \cup \mathbb{B}, \theta_d) \cap \mathbb{C} \neq \emptyset\} \qquad (6.84) \\
&= \{\theta_d \in \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} | BACK(\mathbb{A} \cup \mathbb{B}, \theta_d) \cap \mathbb{C} \neq \emptyset\} \\
&\cup \{\theta_d \in \overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}} | BACK(\mathbb{A} \cup \mathbb{B}, \theta_d) \cap \mathbb{C} \neq \emptyset\} \qquad (6.85) \\
&= \Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} \cup \Theta^{\overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} \qquad (6.86)
\end{aligned}$$

From the second case explained above, according to Equation 6.78 and the definition in

Equation 6.52, the second term $\Theta^{\overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}}$ in Equation 6.86 can be decomposed as follows:

$$\begin{aligned}
\Theta^{\overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} &= \{\theta_d \in \overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}} | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset\} && (6.87)\\
&= \{\theta_d \in \overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}} | (BACK(\mathbb{A},\theta_d)\cup BACK(\mathbb{B},\theta_d))\cap\mathbb{C}\neq\emptyset\} && (6.88)\\
&= \{\theta_d \in \overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}} | (BACK(\mathbb{A},\theta_d)\cap\mathbb{C}\neq\emptyset)\vee(BACK(\mathbb{B},\theta_d))\cap\mathbb{C}\neq\emptyset)\} && (6.89)\\
&= \{\theta_d \in \overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}} | BACK(\mathbb{A},\theta_d)\cap\mathbb{C}\neq\emptyset\}\\
&\cup \{\theta_d \in \overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}} | BACK(\mathbb{B},\theta_d))\cap\mathbb{C}\neq\emptyset\} && (6.90)\\
&= (\Theta_{\mathbb{C}\rightarrow\mathbb{A}}\cap\overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}})\cup(\Theta_{\mathbb{C}\rightarrow\mathbb{B}}\cap\overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}) && (6.91)\\
&= (\Theta_{\mathbb{C}\rightarrow\mathbb{A}}\cup\Theta_{\mathbb{C}\rightarrow\mathbb{B}})\cap\overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}} && (6.92)
\end{aligned}$$

Considering the set of directions that does not link $\mathbb{A}$ to $\mathbb{B}$, finding the set of directions to leave $\mathbb{C}$ and reach $\mathbb{A}$ or $\mathbb{B}$ consists indeed to find separately the set of directions to leave $\mathbb{C}$ and reach $\mathbb{A}$ ($\Theta_{\mathbb{C}\rightarrow\mathbb{A}}$) and the set of directions to leave $\mathbb{C}$ and reach $\mathbb{B}$ ($\Theta_{\mathbb{C}\rightarrow\mathbb{B}}$). In others terms, it looks for connection with the other sets in the goal set. The sets $\Theta_{\mathbb{C}\rightarrow\mathbb{A}}$ and $\Theta_{\mathbb{C}\rightarrow\mathbb{B}}$ can be computed according to Equation 6.67.

The first term $\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}}$ in Equation 6.86 is defined as follows:

$$\begin{aligned}
\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} &= \{\theta_d \in \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset\} && (6.93)\\
&= \bigcup_{\Theta\in\{\Theta_{\mathbb{A}\rightarrow\mathbb{B}},\Theta_{\mathbb{B}\rightarrow\mathbb{A}}\}} \{\theta_d \in \Theta | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset\} && (6.94)\\
&= \Theta_{\mathbb{C}\rightarrow(\mathbb{A}\rightarrow\mathbb{B})}\cup\Theta_{\mathbb{C}\rightarrow(\mathbb{B}\rightarrow\mathbb{A})} && (6.95)
\end{aligned}$$

where $BACK(\mathbb{A}\cup\mathbb{B},\theta_d)$ can be computed according to Equation 6.77 since $\theta_d \in \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$.

Moreover the set $\Theta_{\mathbb{C}\rightarrow(\mathbb{A}\rightarrow\mathbb{B})}$ is defined as follows:

$$\Theta_{\mathbb{C}\rightarrow(\mathbb{A}\rightarrow\mathbb{B})} = \{\theta_d \in \Theta_{\mathbb{A}\rightarrow\mathbb{B}} | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset\} \quad (6.96)$$

As $\Theta_{\mathbb{A}\rightarrow\mathbb{B}}$ was defined according to Equation 6.64, it finally gives:

$$\Theta_{\mathbb{C}\rightarrow(\mathbb{A}\rightarrow\mathbb{B})} = \{\theta_d \in \mathbb{R} | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset \wedge BACK(\mathbb{B},\theta_d)\cap\mathbb{A}\neq\emptyset\} \quad (6.97)$$

Similarly $\Theta_{\mathbb{C}\rightarrow(\mathbb{B}\rightarrow\mathbb{A})}$ can be defined by inverting $\mathbb{A}$ and $\mathbb{B}$.

The set of directions to leave $\mathbb{C}$ and reach $\mathbb{A}$ or $\mathbb{B}$ when $\theta_d \in \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}$ according to Equation 6.95 and Equation 6.97 is then given as follows:

$$\begin{aligned}
\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} &= \Theta_{\mathbb{C}\rightarrow(\mathbb{A}\rightarrow\mathbb{B})}\cup\Theta_{\mathbb{C}\rightarrow(\mathbb{B}\rightarrow\mathbb{A})} && (6.98)\\
&= \{\theta_d \in \mathbb{R} | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset \wedge BACK(\mathbb{B},\theta_d)\cap\mathbb{A}\neq\emptyset\}\\
&\cup \{\theta_d \in \mathbb{R} | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset \wedge BACK(\mathbb{A},\theta_d)\cap\mathbb{B}\neq\emptyset\} && (6.99)\\
&= \{\theta_d \in \mathbb{R} | (BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset \wedge BACK(\mathbb{B},\theta_d)\cap\mathbb{A}\neq\emptyset)\\
&\vee (BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset \wedge BACK(\mathbb{A},\theta_d)\cap\mathbb{B}\neq\emptyset)\} && (6.100)\\
&= \{\theta_d \in \mathbb{R} | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset\\
&\wedge (BACK(\mathbb{B},\theta_d)\cap\mathbb{A}\neq\emptyset \vee BACK(\mathbb{A},\theta_d)\cap\mathbb{B}\neq\emptyset)\} && (6.101)\\
&= \{\theta_d \in \mathbb{R} | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset\}\\
&\cap \{\theta_d \in \mathbb{R} | BACK(\mathbb{B},\theta_d)\cap\mathbb{A}\neq\emptyset \vee BACK(\mathbb{A},\theta_d)\cap\mathbb{B}\neq\emptyset\} && (6.102)\\
&= \{\theta_d \in \mathbb{R} | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset\}\\
&\cap \Big(\{\theta_d \in \mathbb{R} | BACK(\mathbb{B},\theta_d)\cap\mathbb{A}\neq\emptyset\}\cup\{\theta_d \in \mathbb{R} | BACK(\mathbb{A},\theta_d)\cap\mathbb{B}\neq\emptyset\}\Big) && (6.103)\\
&= \{\theta_d \in \mathbb{R} | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset\}\cap(\Theta_{\mathbb{A}\rightarrow\mathbb{B}}\cup\Theta_{\mathbb{B}\rightarrow\mathbb{A}}) && (6.104)\\
&= \{\theta_d \in \mathbb{R} | BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cap\mathbb{C}\neq\emptyset\}\cap\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} && (6.105)
\end{aligned}$$

Notice that Equation 6.105 corresponds indeed to:

$$\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} = \{\theta_d \in \mathbb{R} | BACK(\mathbb{A}\cup\mathbb{B}, \theta_d) \cap \mathbb{C} \neq \emptyset\} \cap \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} \quad (6.106)$$

$$= \Theta_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} \cap \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} \quad (6.107)$$

However the set $BACK(\mathbb{A}\cup\mathbb{B}, \theta_d)$ is computed here according to Equation 6.77 which gives:

$$\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} = \{\theta_d \in \mathbb{R} | \bigcap_{\delta\theta\in\{-\alpha_\theta,\alpha_\theta\}} proj^{\mathbb{T}}_{\mathbb{X}_1\times\mathbb{X}_2}(\mathbf{f}_H^{-1}(\mathbb{A}\cup\mathbb{B})) \cap \mathbb{C}\} \cap \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} \quad (6.108)$$

where again $\mathbb{X}_1$, $\mathbb{X}_2$, and $\mathbb{T}$ are the domains to x, y and t respectively.

Finally the global set of directions to leave $\mathbb{C}$ and reach $\mathbb{A}$ or $\mathbb{B}$ is given as follows:

$$\Theta_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} = \Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} \cup \Theta^{\overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} \quad (6.109)$$

$$= \Big(\{\theta_d \in \mathbb{R} | \bigcap_{\delta\theta\in\{-\alpha_\theta,\alpha_\theta\}} proj^{\mathbb{T}}_{\mathbb{X}_1\times\mathbb{X}_2}(\mathbf{f}_H^{-1}(\mathbb{A}\cup\mathbb{B})) \cap \mathbb{C}\} \cap \Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}\Big)$$

$$\cup \Big((\Theta_{\mathbb{C}\rightarrow\mathbb{A}} \cup \Theta_{\mathbb{C}\rightarrow\mathbb{B}}) \cap \overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}\Big) \quad (6.110)$$

From this equation, several particular cases can be established:

- If the intersection $\mathbb{A}\cap\mathbb{B}$ is not empty, then $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} = \mathbb{R}$ since some points in $\mathbb{A}$ are at the same time in the backward reach set of $\mathbb{B}$ and in $\mathbb{B}$, and vice versa. Consequently, $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} = \emptyset$ and finally it leads to:

  $$\Theta_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} = \{\theta_d \in \mathbb{R} | \bigcap_{\delta\theta\in\{-\alpha_\theta,\alpha_\theta\}} proj^{\mathbb{T}}_{\mathbb{X}_1\times\mathbb{X}_2}(\mathbf{f}_H^{-1}(\mathbb{A}\cup\mathbb{B})) \cap \mathbb{C}\} \quad (6.111)$$

  which corresponds to Equation 6.67 where $\mathbb{B}$ is now $\mathbb{A}\cup\mathbb{B}$ and $\mathbb{A}$ is $\mathbb{C}$. It confirms what was stated before.

- If no links exist between $\mathbb{A}$ and $\mathbb{B}$, i.e. $\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}} = \emptyset$ or equivalently $\overline{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}} = \mathbb{R}$, then it simply looks for direct connection with $\mathbb{A}$ and $\mathbb{B}$ separately:

  $$\Theta_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}} = \Theta_{\mathbb{C}\rightarrow\mathbb{A}} \cup \Theta_{\mathbb{C}\rightarrow\mathbb{B}} \quad (6.112)$$

The explanation of the different cases for the computation of the 2D backward reach set that depends on the values of θ_d according to the different sets computed is straightforward. The explanation was provided when two sets $\mathbb{A}$ and $\mathbb{B}$ were considered. Some cases will be presented in the following examples. The computation of the global set $\Theta_{\mathbb{C}\leftrightarrow\mathbb{A}\cup\mathbb{B}}$ needs to consider the different backward reachability of $\mathbb{C}$ onto $\mathbb{A}$ and $\mathbb{B}$.

Important note

According to Equation 6.92, when no link exists between $\mathbb{A}$ and $\mathbb{B}$, considering a third set $\mathbb{C}$ in the goal set, it looks for connection between $\mathbb{C}$ and $\mathbb{A}$ or $\mathbb{C}$ and $\mathbb{B}$ (separately). The problem can be seen as a graph building where the first set $\mathbb{A}$ is tried to be connected to other sets such as $\mathbb{B}$ or $\mathbb{C}$. When a link exists between $\mathbb{A}$ and $\mathbb{B}$ for example, it extends the space of the backward reach set according to Equation 6.105 where a possible third set $\mathbb{C}$ can be connected. This graph creation will be the topic of the next section and is at the heart of the motion planning problem.

Example

Consider the example with the three sets $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$ provided in this subsection where an illustration is depicted in Figure 6.12(a). Consider the same robot with the same uncertainties. The set of directions linking $\mathbb{A}$ and $\mathbb{B}$ was already computed and the results can be found in Figure 6.11 where the inner approximation was defined by blue intervals. Some results between the three sets is provided in Figure 6.13. Adding this third set $\mathbb{C}$, the computation of the set $\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}}$ is provided in Figure 6.13(a). Since $\Theta_{\mathbb{C}\rightarrow(\mathbb{B}\rightarrow\mathbb{A})}$ is empty, and according to Equation 6.95, the set $\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}}$ is then described only by $\Theta_{\mathbb{C}\rightarrow(\mathbb{A}\rightarrow\mathbb{B})}$. The inner approximation of the set $\Theta_{\mathbb{A}\rightarrow\mathbb{B}}$ is depicted by the red interval. The new inner approximation of the set considering $\mathbb{C}$ is depicted by the blue interval. Notice that indeed $\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}}$ is a subset of $\Theta_{\mathbb{A}\rightarrow\mathbb{B}}$. A direction shown by the green line is selected at $\theta_d = 60°$ in the solution set. Since θ_d belongs to $\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}}$, meaning that it also belongs $\Theta_{\mathbb{A}\rightarrow\mathbb{B}}$, then the 2D backward reach set is given as follows:

$$BACK(\mathbb{A}\cup\mathbb{B}\cup\mathbb{C}) = \bigcap_{\delta\theta\in\{-\alpha_\theta,\alpha_\theta\}} proj^{\mathbb{T}}_{\mathbb{X}_1\times\mathbb{X}_2}(\mathbf{f}_H^{-1}(\mathbb{A}\cup\mathbb{B}\cup\mathbb{C})) \tag{6.113}$$

which is similar to Equation 6.77 with the set $\mathbb{C}$ added to the goal set.

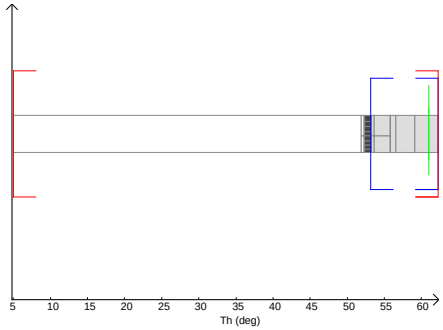
The backward reach set in the $(x-y)$ for this direction is shown in Figure 6.13(b). Starting from any of these soft gray boxes and having a direction defined at $\theta_d = 60°$ will lead the robot to one of the three sets despite the uncertainty on the direction in single motion commands.

If a direction $\theta_d = 10°$ is selected outside $\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}}$ but inside $\Theta_{\mathbb{A}\rightarrow\mathbb{B}}$ as depicted by the green line in Figure 6.13(c), the corresponding 2D backward reach set is provided in Figure 6.13(d). Indeed, since θ_d does not belong to $\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\rightarrow\mathbb{A}\cup\mathbb{B}}$, the backward reach set of $\mathbb{C}$ cannot be considered with $\mathbb{A}$ and $\mathbb{B}$ but separately as in the case in Equation 6.78. Moreover, θ_d belongs to $\Theta_{\mathbb{A}\rightarrow\mathbb{B}}$ then the backward reach of $\mathbb{A}$ and $\mathbb{B}$ can be computed as in Equation 6.77. The global backward reach is then defined as follows:

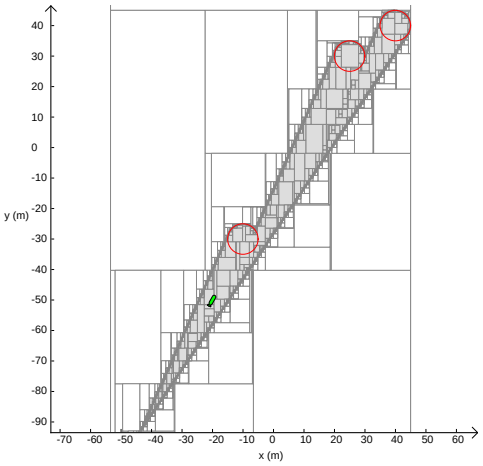
$$BACK(\mathbb{A}\cup\mathbb{B}\cup\mathbb{C},\theta_d) = BACK(\mathbb{A}\cup\mathbb{B},\theta_d)\cup BACK(\mathbb{C},\theta_d) \tag{6.114}$$

where $BACK(\mathbb{A}\cup\mathbb{B},\theta_d)$ is computed according to Equation 6.77 and $BACK(\mathbb{C},\theta_d)$ according to Equation 6.52.

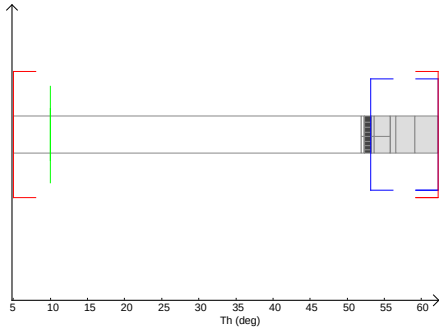
The selected direction provides the robot location able to reach $\mathbb{A}$ or $\mathbb{B}$ (together) and $\mathbb{C}$ separately since the three sets are not connected directly as in Figure 6.13(a) and (b).



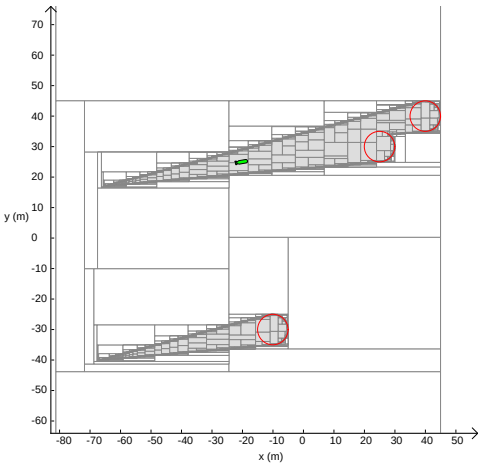
(a) Solution of $\Theta_{\mathbb{C}\to(\mathbb{A}\to\mathbb{B})}$. Red interval indicates the initial interval $\Theta_{\mathbb{A}\to\mathbb{B}}$ and the blue interval is the inner approximation of the solution set.



(b) 2D backward reach set at $\theta_d = 61° \in \Theta_{\mathbb{C}\to(\mathbb{A}\to\mathbb{B})}$ represented by the green line in the figure (a). It links $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$.



(c) $\theta_d = 10°$ outside the solution set represented by the green line.



(d) 2D backward reach set at $\theta_d = 10°$.

Figure 6.13: Different sets of direction linking $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$. The blue intervals indicate the inner approximation of $\Theta^{\Theta_{\mathbb{A}\leftrightarrow\mathbb{B}}}_{\mathbb{C}\to\mathbb{A}\cup\mathbb{B}}$ and the red brackets are the intervals (set of directions) that link $\mathbb{A}$ to $\mathbb{B}$. Grayscale convention for thin sets.

In the following of the document, $\mathbb{A} \to \mathbb{B}$ will be denoted as $\mathbb{AB}$ and $\mathbb{B} \to \mathbb{A}$ as $\mathbb{BA}$ for ease of reading. The place of the sets indicates the direction. $\mathbb{AB}$ means that it starts from $\mathbb{A}$ and reach $\mathbb{B}$.

6.4.2.3 What about the departure positions ?

Until now, the algorithm looked for some directions that could linked in a guaranteed manner different sets, $\mathbb{A}$ and $\mathbb{B}$ for example. It is based on a projection on θ with respect to the $(x-y)$ plane by checking if some points in a backward reach set of a set $\mathbb{B}$ are inside a set $\mathbb{A}$. This was defined in Equation 6.64 where the computation based on projection was provided in Equation 6.67. Therefore, the set of directions corresponded to the projection of the intersection of the set $\mathbb{A}$ and the omnidirectional backprojection of the set $\mathbb{B}$ onto the space of directions θ with respect to the $(x-y)$ plane as mentioned in Equation 6.69. The computation of the starting

point in $\mathbb{A}$ to reach $\mathbb{B}$ is then the projection of the intersection of $\mathbb{A}$ and the omnidirectional backprojection of $\mathbb{B}$ onto the $(x-y)$ plane with respect to the space of directions:

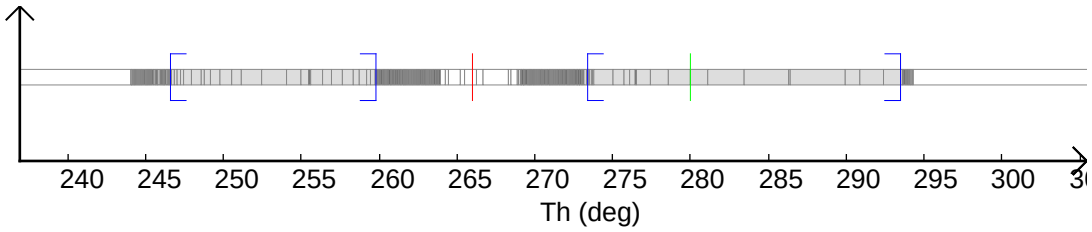
$$\begin{aligned}
\mathbb{P}_{\mathbb{A}\to\mathbb{B}} &= \{(x_1,x_2)\in\mathbb{X}_1\times\mathbb{X}_2 | \exists x_3\in\mathbb{X}_3, \mathbf{x}\in BACK_{od}(\mathbb{B}) \text{ and } \mathbf{x}\in\mathbb{A}\} && (6.115)\\
&= \{(x_1,x_2)\in\mathbb{X}_1\times\mathbb{X}_2 | \exists x_3\in\mathbb{X}_3, \mathbf{x}\in BACK_{od}(\mathbb{B})\cap\mathbb{A}\neq\emptyset\} && (6.116)\\
&= proj^{\mathbb{X}_3}_{\mathbb{X}_1\times\mathbb{X}_2}\Big(\bigcap_{\delta\theta\in\{-\alpha_\theta,\alpha_\theta\}} proj^{\mathbb{T}}_{\mathbb{X}}(\mathbf{f}_H^{-1}(\mathbb{B}))\cap\mathbb{A}\Big) && (6.117)\\
&= proj^{\mathbb{X}_3}_{\mathbb{X}_1\times\mathbb{X}_2}\Big(\mathbb{X}_{\mathbb{A}\to\mathbb{B}}\Big) && (6.118)
\end{aligned}$$

where $\mathbb{X}_1$, $\mathbb{X}_2$ and $\mathbb{X}_3$ represent respectively the domains of x_1, x_2 and $x_3=\theta_d$.

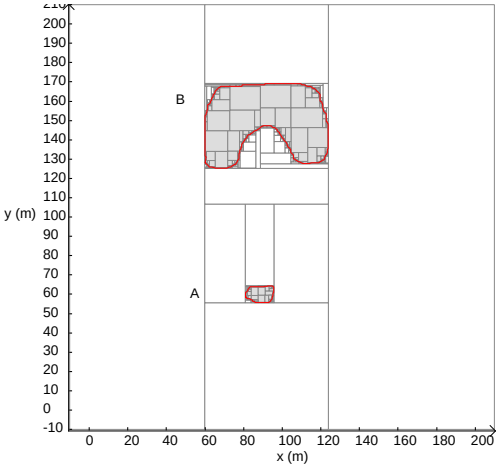
Example

Consider the sets depicted in Figure 6.14(b) where the upper set is $\mathbb{A}$ and the lower set is $\mathbb{B}$. The set of directions $\Theta_{\mathbb{A}\to\mathbb{B}}=\Theta_{\mathbb{A}\mathbb{B}}$ that links $\mathbb{A}$ to $\mathbb{B}$ is represented in Figure 6.14(a) where again only the values along the $x-$dimension are important. Notice that now the set of directions could be approximately separated in two blue intervals due to the shape of the $\mathbb{A}$. Indeed, the 2D backward reach set of $\mathbb{B}$ at the direction $\theta_d=280°$ (green line) enables to leave $\mathbb{A}$ and reach $\mathbb{B}$ as depicted in Figure 6.14(c). However, at the direction $\theta_d=266°$, as it can be seen in the 2D backward reach in Figure 6.14(d) it is not possible to start from $\mathbb{A}$ and reach $\mathbb{B}$ with this motion command. A zoom is provided in Figure 6.14(e) to illustrate more precisely.

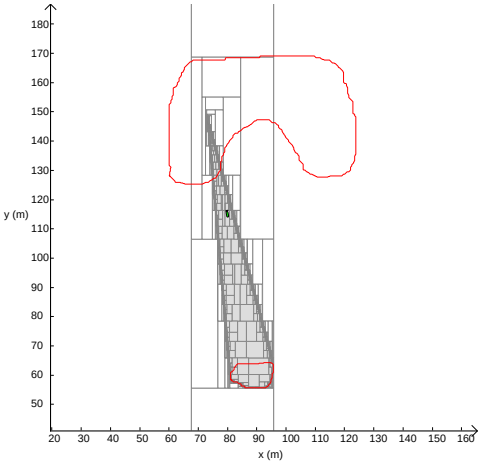
What are then the possible locations in $\mathbb{A}$ such that a motion command (a direction) exists to leave $\mathbb{A}$ and reach $\mathbb{B}$? The solution corresponding to Equation 6.118 is provided in Figure 6.15. A zoom is provided in Figure 6.15(b). It indicates the $(x-y)$ positions the robot has to be in, such that there exists at least one direction that leads the robot in $\mathbb{B}$ for sure. Notice that many little unclassified boxes are present when it is far from the set $\mathbb{B}$. Indeed, the further the position is, the more difficult it is to guarantee the reachability and the smaller the solution boxes will be. Moreover the resolution of the initial image was large. The idea beyond this computation is just to prove that there exists some location inside $\mathbb{A}$ to reach $\mathbb{B}$ for sure. The results are complementary to the computation of $\Theta_{\mathbb{A}\mathbb{B}}$. For example, at the position $(70,130)$ represented by a blue dot in Figure 6.15(a), the set $\Theta_{\mathbb{A}\mathbb{B}}$ from this initial location is given in Figure 6.15(c). At this location, with a direction inside the set $\Theta_{\mathbb{A}\mathbb{B}}$, it guarantees the reachability of $\mathbb{B}$ despite the uncertainty on the direction. For example, at the direction $\theta_d=285°$, indicates by the green line in Figure 6.15(c), some trajectories from this initial pose $(70,130,285)(m,m,deg)$ are depicted in Figure 6.15(d) in blue with the error on the direction selected uniformly in the interval $[-5,5](deg)$. Notice that all trajectories cross at a certain time the set $\mathbb{B}$, guaranteeing the reachability of $\mathbb{B}$.



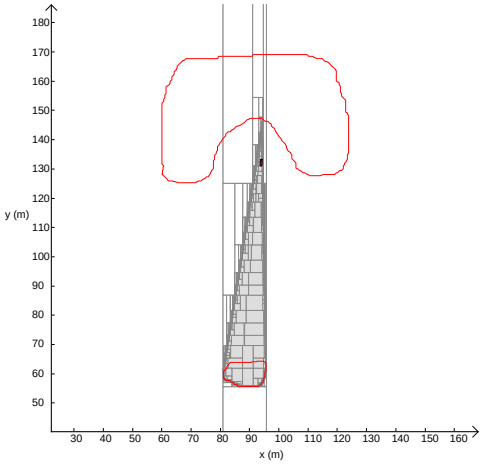
(a) $\Theta_{\mathbb{A}\to\mathbb{B}} = \Theta_{\mathbb{A}\mathbb{B}}$. Blue intervals are approximative.



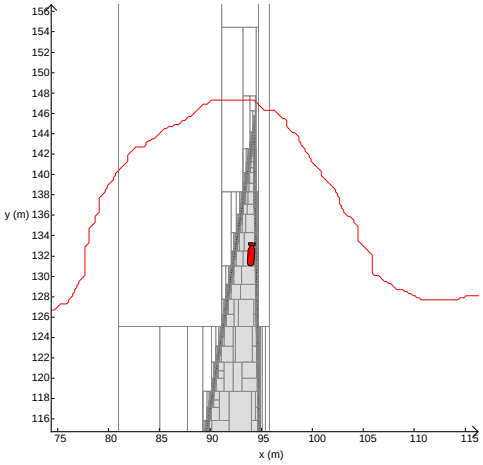
(b) Environment with complex shapes.



(c) 2D backward reach at $\theta_d = 280°$ corresponding to the green line in the figure (a).

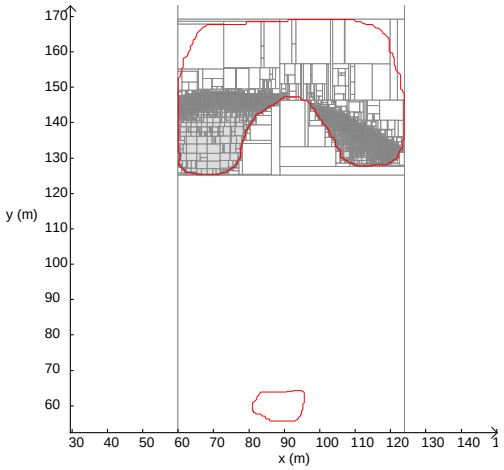


(d) 2D backward reach at $\theta_d = 266°$ corresponding to the red line in the figure (a).

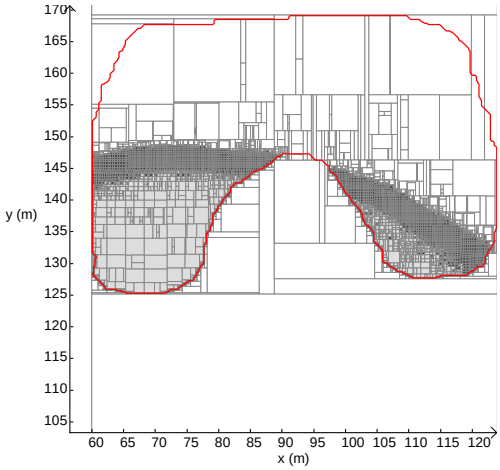


(e) Zoom of the figure (d). No location inside $\mathbb{A}$ can reach $\mathbb{B}$ with this direction $\theta_d = 266°$.

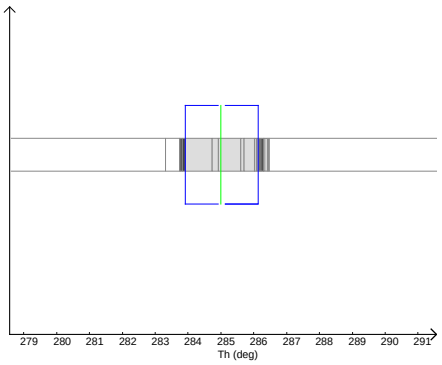
Figure 6.14: Link between two shapes. Grayscale convention for thin sets.



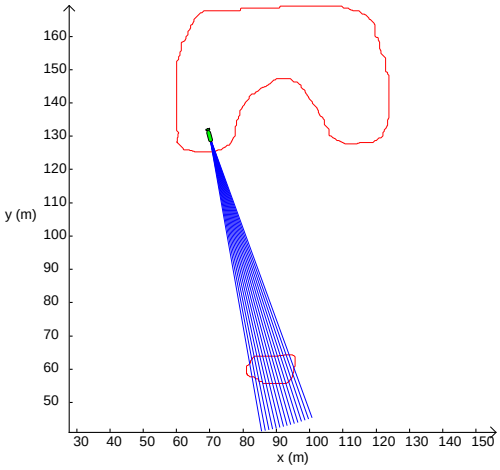
(a) Initial locations $\mathbb{P}_{\mathbb{A}\to\mathbb{B}}$ in $\mathbb{A}$ where at least one direction exists that can lead the robot for sure in $\mathbb{B}$.



(b) Zoom of $\mathbb{P}_{\mathbb{A}\to\mathbb{B}}$.



(c) $\Theta_{\mathbb{A}\to\mathbb{B}}$ computed at the position $(70, 130)$ represented by the blue dot in the figure (a) .



(d) Some trajectories in blue at the direction $\theta_d = 285°$ represented by the green line in the figure (c) from the initial pose in green with an error uniformly generated in the interval $[-5, 5](deg)$.

Figure 6.15: Computation of $\mathbb{P}_{\mathbb{A}\to\mathbb{B}}$. Many little boxes due to the resolution of the initial image and the size of the boxes that are getting smaller with the distance to $\mathbb{B}$.

6.4.2.4 Backprojection with forbidden areas

It was assumed that any obstacle was present in the underwater environment but the method can obviously incorporate obstacles or forbidden areas. These areas can be defined by sets from an image or from inequalities. By considering N connected forbidden areas $\mathbb{O}_i \subset \mathbb{R}^3$, the set of forbidden areas is given as follows:

$$\mathbb{O} = \bigcup_{i\in\{0,..,N\}} \mathbb{O}_i \tag{6.119}$$

Usually the forbidden areas as obstacles are only subsets of $\mathbb{R}^2$ but it can be extended to $\mathbb{R}^3$ to restrict the heading. A forbidden area can be a region where no landmarks are available or too many which can lead to wrong data association.

The omnidirectional backward reachability of a set $\mathbb{A} \subset \mathbb{R}^3$ considering forbidden areas is given as follows:

$$\begin{aligned} BACK_{od}^{\mathbb{O}}(\mathbb{A}) &= \{\mathbf{x} \in \mathbb{R}^3 | \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A} \text{ and} \\ &\quad \forall \delta_\theta \in [\alpha_\theta, \alpha_\theta], \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{O}\} \quad (6.120) \\ &= BACK_{od}(\mathbb{A}) \cap \{\mathbf{x} \in \mathbb{R}^3 | \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \forall t \in \mathbb{R}^+, \\ &\quad \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{O}\} \quad (6.121) \\ &= BACK_{od}(\mathbb{A}) \cap \overline{\{\mathbf{x} \in \mathbb{R}^3 | \exists \delta_\theta, \exists t, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{O}\}} \quad (6.122) \\ &= BACK_{od}(\mathbb{A}) \cap \overline{WBACK_{od}(\mathbb{O})} \quad (6.123) \end{aligned}$$

where $WBACK$ refers to the weak backprojection introduced in Equation 6.32 and defined according to the directional/omnidirectional backprojection in Equation 6.42 and in Equation 6.43.

The backward reach set can also be decomposed in two parts, one related to the goal set $\mathbb{A}$ and one related to the forbidden areas $\mathbb{O}$. The classic backward reach set $BACK_{od}$ provides the set of robot configurations guaranteeing to reach the goal set $\mathbb{A}$ contrary to the weak backward reach set $WBACK_{od}$ that provides the robot configurations that may cross at a certain time an obstacle $\mathbb{O}$ according to the uncertain parameter $\mathbf{p}$ which is here only $\delta\theta$.

Any trajectories that could cross a forbidden area may lead to the destruction of the robot if they are actually obstacles.

The weak directional backprojection can be computed with projection as follows:

$$WBACK_{od}(\mathbb{O}) = proj_{\mathbb{X}}^{[-\alpha_\theta, \alpha_\theta] \times \mathbb{R}^+}(\mathbf{f}_H^{-1}(\mathbb{O})) \quad (6.124)$$

where $[-\alpha_\theta, \alpha_\theta]$ is the error on the direction and $\mathbb{R}^+$ is the domain of t.

Finally the omnidirectional backward reachability can be computed as follows:

$$BACK_{od}^{\mathbb{O}}(\mathbb{A}) = BACK_{od}(\mathbb{A}) \cap \overline{WBACK_{od}(\mathbb{O})} \quad (6.125)$$

If the direction θ_d is defined, the direction backprojection including obstacles can be defined as follows:

$$BACK^{\mathbb{O}}(\mathbb{A}, \theta_d) = BACK(\mathbb{A}, \theta_d) \cap \overline{WBACK(\mathbb{O}, \theta_d)} \quad (6.126)$$

Notice that if the direction θ_d is defined, the 2D weak direction backprojection (in the $(x-y)$ plane) of a set $\mathbb{B}$ ($\mathbb{B}$ is preferred to $\mathbb{O}$ for notation) can be expressed as a Minkowski sum. Consider a set $\mathbb{B} \subset \mathbb{R}^2$ and then the function $\mathbf{f}_H = \mathbf{f}_{H2}$.

$$\begin{aligned} WBACK(\mathbb{B}, \theta_d) &= \{\mathbf{x} \in \mathbb{R}^2 | \exists \delta_\theta \in [\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B}\} \\ &= \{\mathbf{x} \in \mathbb{R}^2 | \exists \delta_\theta \in [\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{x} + v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{B}\} \\ &= \{\mathbf{x} \in \mathbb{R}^2 | \exists \delta_\theta \in [\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \exists \mathbf{b} \in \mathbb{B}, \mathbf{x} + v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} = \mathbf{b}\} \\ &= \{\mathbf{x} \in \mathbb{R}^2 | \exists \delta_\theta \in [\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \exists \mathbf{b} \in \mathbb{B}, \mathbf{x} = \mathbf{b} - v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix}\} \end{aligned}$$

The second term corresponds to a polar set defined as follows:

$$\mathbb{X}_{polar}(\theta_d) = \{\mathbf{x} \in \mathbb{R}^2 | \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{x} = v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix}\} \quad (6.127)$$

where $\rho = v \cdot t$ and $\theta = \theta_d + \delta_\theta$.

Finally, the weak backprojection is given as follows:

$$WBACK(\mathbb{B}, \theta_d) = \mathbb{B} \oplus (-\mathbb{X}_{polar}(\theta_d)) \tag{6.128}$$

It could have been noticed that:

$$-v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} = v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \pi + \delta_\theta) \\ \sin(\theta_d + \pi + \delta_\theta) \end{pmatrix} \tag{6.129}$$

The proposition given in Equation 6.126 is not always true and in particular is false when the obstacle is behind the relocation area. This will be explained in **Example 2**.

Example 1

Consider an uncertainty on the compass $\alpha_\theta = 5°$ and no uncertainty on the speed $\alpha_v = 0\%$. The desired speed v_d is fixed at $1m/s$. Therefore $[\mathbf{p}] = \{1\} \times [-5, 5](\frac{m}{s} \times deg)$ Define the same goal set $\mathbb{A}$ as in Figure 6.6 which was described as follows:

$$\mathbb{A} = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 25)^2 + (x_2 - 30)^2} \leq 5\} \tag{6.130}$$

Define two forbidden areas $\mathbb{O}_1$ and $\mathbb{O}_2$ as follows:

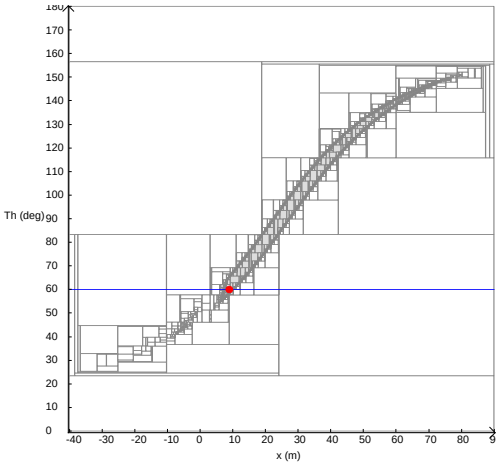
$$\mathbb{O}_1 = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 10)^2 + (x_2 - 10)^2} \leq 2\} \tag{6.131}$$

and

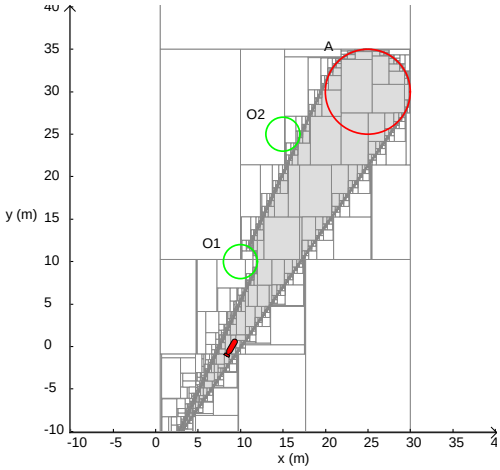
$$\mathbb{O}_2 = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 15)^2 + (x_2 - 25)^2} \leq 2\} \tag{6.132}$$

such that $\mathbb{O} = \mathbb{O}_1 \cup \mathbb{O}_2$. This environment is depicted in Figure 6.16(b) and (c) where the goal set is in red and the forbidden areas are in green. The slice at $y = 0m$ of the backward reach set of $\mathbb{A}$ is given in Figure 6.16(a) with a (x, θ_d) representation. The 2D backward reach set at the slice $\theta_d = 60°$ corresponding to the blue line is depicted in Figure 6.16(b) where the red AUV pose corresponds to the red dot in the figure (a). Similarly the slice at $x = 10m$ and the associated 2D backward reach set the blue line are given in Figure 6.16(c) and (d). The red AUV pose corresponds again to the red dot on the associated solution set. The 2D backward reach set passes between the two forbidden areas. Notice the difference at the slice $y = 0m$ of the solution set without the forbidden areas in Figure 6.6(a) and with the forbidden areas in Figure 6.16(a). A comparison can also be made between the solution at the slice $x = 10m$ between Figure 6.6(c) and 6.16(c). Some parts of the solution set without forbidden areas are removed.

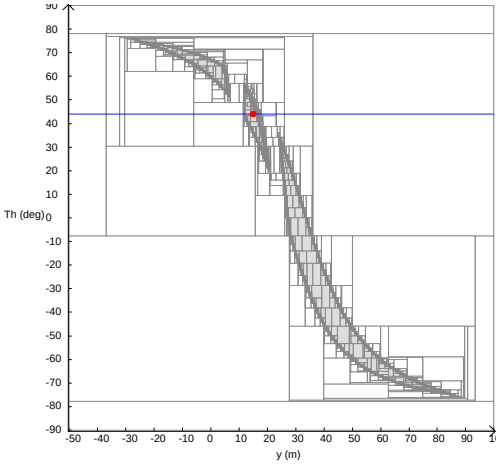
Notice how the 2D backward reach takes into account the forbidden areas to provide a safe backward reach set.



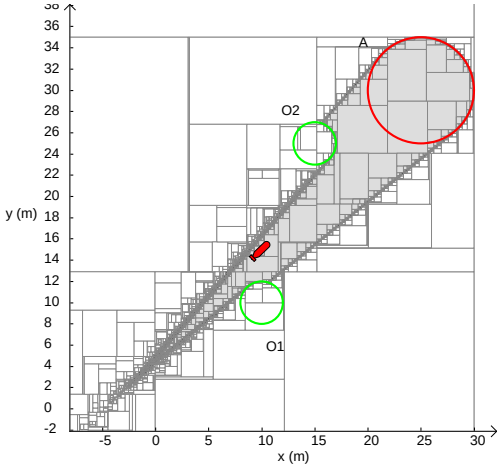
(a) (x, θ_d) representation for the slice at $y = 0m$.



(b) (x, y) (2D) representation for the slice at $\theta_d = 60°$ depicted by the blue line in the figure (a). A red AUV indicates the pose at the red dot in the figure (a).



(c) (y, θ_d) representation for the slice at $x = 10m$.



(d) (x, y) (2D) representation for the slice at $\theta_d = 44°$ depicted by the blue line in the figure (c). A red AUV indicates the pose at the red dot in the figure (c).

Figure 6.16: Backward reach set of the set $\mathbb{A}$ defined by a 2D disk (in red) in presence of forbidden areas (in green).

Example 2

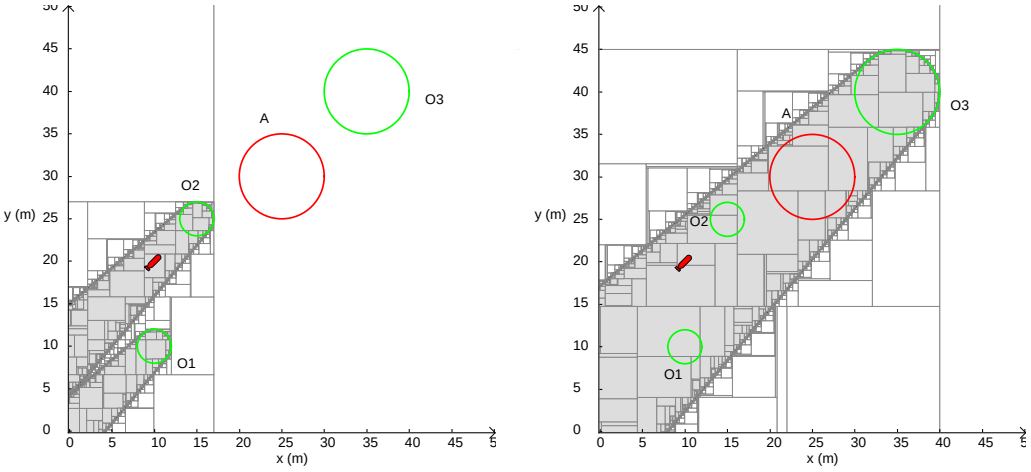
Consider the same example as in **Example 1** where a third obstacle $\mathbb{O}_3$ defined as follows is added:

$$\mathbb{O}_3 = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 35)^2 + (x_2 - 40)^2} \leq 5\} \tag{6.133}$$

The environment is depicted in Figure 6.17 where the obstacles are represented in green and the goal area in red. The direction defined at $\theta_d = 45°$ is indicated by the red AUV. If only $\mathbb{O}_1$ and $\mathbb{O}_2$ are considered as obstacles, the 2D weak backprojection $WBACK(\mathbb{O}_1 \cup \mathbb{O}_2, 45°)$ is represented in Figure 6.17(a). If $\mathbb{O}_3$ is added to the set of obstacles, the 2D weak backprojection $WBACK(\mathbb{O}_1 \cup \mathbb{O}_2 \cup \mathbb{O}_3, 45°)$ is given in Figure 6.17(b) where as it can be noticed the goal area $\mathbb{A}$ is hidden by the 2D weak backprojection of $\mathbb{O}_3$ at $\theta_d = 45°$. When computing the backward reach set of $\mathbb{A}$ according to Equation 6.125 considering these three obstacles, it would return no solution at $\theta_d = 45°$ or it has been shown in Figure 6.16(d) at almost the same direction that

the 2D backward reach set actually exists. Therefore $\mathbb{O}_3$ must not be considered as an obstacle when computing the 2D backward reach set of $\mathbb{A}$ at this direction, only $\mathbb{O}_1$ and $\mathbb{O}_2$ may be considered at this direction. This is explained by the fact that $\theta_d = 45° \notin \Theta_{\mathbb{O}_3\mathbb{A}}$ contrary to $\theta_d = 45° \in \Theta_{\mathbb{O}_1\mathbb{A}}$ and $\theta_d = 45° \in \Theta_{\mathbb{O}_2\mathbb{A}}$.

Consequently, Equation 6.125 is true under some conditions. This will be particularly important when dealing with indistinguishable relocation areas to avoid any ambiguity at the moment of the graph building.



(a) 2D weak backprojection $WBACK(\mathbb{O}_1 \cup \mathbb{O}_2, 45°)$.

(b) 2D weak backprojection $WBACK(\mathbb{O}_1 \cup \mathbb{O}_2 \cup \mathbb{O}_3, 45°)$.

Figure 6.17: Weak backprojection of obstacles at the defined direction $\theta_d = 45°$.

Considering several sets in the goal set, it is still possible to compute the safe backward reach set according to Equation 6.125 and the theory developed above on several sets with directions. It will not be discussed more.

6.4.2.5 Goal recognizability

Until now it was assumed that a sensor was able to indicate that the robot has reached a goal set that may be defined by several sets. The termination condition [183][88] was then based on this available exteroceptive information. Only the uncertainty on the direction was taken into account since the speed would only influences when the robot will reach the goal set. Consider now the uncertainty on the speed is not null anymore, i.e. $\alpha_v > 0\%$. The time t has then to be considered. The reachability was defined as the ability to reach the goal and the recognizability was the ability to stop in it. Contrary to what stated in [89] about the separated computation of the reachability and the recognizability, it is possible to compute both at the same time when dealing with directional backprojection.

The omnidirectional backprojection providing the reachability and the recognizability of a set $\mathbb{A} \subset \mathbb{R}^3$:

$$BACK_{od}^{rec}(\mathbb{A}) = \{\mathbf{x} \in \mathbb{R}^3 | \exists t \in \mathbb{R}^+, \forall \mathbf{p} \in [\mathbf{p}], \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \tag{6.134}$$

where "rec" is added to indicate the reachability and the recognizability at the same time. Notice the place difference of $\exists t$ with Equation 6.36.

In general, $\exists$ and $\forall$ do not commute. The set defined in Equation 6.134 looks for the initial poses such that at a certain time all the poses of the robot are inside the goal set $\mathbb{A}$. This is different from Equation 6.36 where it looks for the initial poses such that whatever the uncertainties are the robot will reach the goal set $\mathbb{A}$ at a certain time.

The set defined in Equation 6.134 is more restrictive and it can be noticed that:

$$BACK_{od}^{rec}(\mathbb{A}) \subset BACK_{od}(\mathbb{A}) \tag{6.135}$$

Contrary to the theory developed on the reachability with the work on the bounds, in this case it is not possible to consider the bounds since it has to be proved that all the poses are inside the goal set whatever the uncertainties are. Considering only the bounds would lead to an over approximation of the solution set.

The set expressed in Equation 6.134 can be computed with projection:

$$BACK_{od}^{rec}(\mathbb{A}) = proj_{\mathbb{X}}^{\mathbb{T}}\Big(\overline{proj_{\mathbb{X}\times\mathbb{P}}^{[\mathbf{p}]}(\overline{\mathbf{f}_H^{-1}(\mathbb{A})})}\Big) \tag{6.136}$$

where $\mathbb{P}$ is the domain of the uncertain parameter $\mathbf{p}$ ($\mathbb{P} = [\mathbf{p}]$) and $\mathbb{X}$ is domain of $\mathbf{x} = (x, y, \theta)$ which can be $\mathbb{R}^3$.

At a defined initial robot location $\mathbf{q}_0 = (x_0, y_0)$, since the robot can move in any direction, it is possible to compute the set of motion commands (θ_d, t) such that it guarantees the robot has reached the goal set depending on the direction to follow θ_d and end in it depending on the time t. It is called the termination condition (TC).

The set of motion commands, or termination condition (TC), at the defined initial robot location $\mathbf{q}_0 = (x_0, y_0)$ is expressed as follows:

$$TC(\mathbf{q}_0) = \{(\theta_d, t) \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], \mathbf{f}_H(\mathbf{x}_0, t, \mathbf{p}) \in \mathbb{A}\} \tag{6.137}$$

where $\mathbf{x}_0 = (\mathbf{q}_0, \theta_d) = (q_0(1), q_0(2), \theta_d) = (x_0, y_0, \theta_d)$.

This set can be expressed with projection:

$$TC(\mathbf{q}_0) = \overline{proj_{\mathbb{X}_3\times\mathbb{T}}^{[\mathbf{p}]}(\overline{\mathbf{f}_H^{-1}(\mathbb{A})})} \tag{6.138}$$

where $\mathbb{X}_3$ is the domain of $\theta_d = x_3$.

The set expressed in Equation 6.137 can be rewritten as follows:

$$\begin{aligned} TC(\mathbf{q}_0) &= \{(\theta_d, t) \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], \mathbf{f}_H(\mathbf{x}_0, t, \mathbf{p}) \in \mathbb{A}\} &(6.139)\\ &= \bigcap_{\mathbf{p}\in\mathbb{P}} \{(\theta_d, t) \in \mathbb{R}^2 | \mathbf{f}_H(\mathbf{x}_0, t, \mathbf{p}) \in \mathbb{A}\} &(6.140)\end{aligned}$$

where $\mathbb{P} = [\mathbf{p}] = [v] \times [\delta_\theta] = [v_d(1-\alpha_v), v_d(1+\alpha_v)] \times [-\alpha_\theta, \alpha_\theta]$.

As the set $\mathbb{P}$ corresponds to an interval vector, it can be discretized in grid for example (uniformly along each dimension). The set $\mathbb{P}$ then corresponds to:

$$\mathbb{P}_{discret} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N\} \tag{6.141}$$

Finally the termination condition according to this discretization is then given as follows:

$$\begin{aligned} TC(\mathbf{q}_0) &= \bigcap_{\mathbf{p}\in\mathbb{P}_{discret}} \{(\theta_d, t) \in \mathbb{R}^2 | \mathbf{f}_H(\mathbf{x}_0, t, \mathbf{p}) \in \mathbb{A}\} &(6.142)\\ &= \bigcap_{\mathbf{p}\in\mathbb{P}_{discret}} \mathbf{f}_H^{-1}(\mathbb{A}) &(6.143)\end{aligned}$$

This discretization enables to avoid the projection but the result does not provide the complete guarantee. If the number of samples approaches infinity, then the result approaches to the complete guarantee. As in motion planning, it is probabilistically complete.

Similarly, this approximation can be made for the computation of the backward reach set in Equation 6.137 to save computation time.

Goal recognizability as a shape registration

The problem expressed here can be seen as a shape registration introduced in Chapter 5 in section 5.2.3. Indeed the propagation of the robot positions from an initial pose is presented in Figure 6.19 which corresponds to a polar set. The problem consists then in checking after a motion command (θ_d, t) that this set is inside the goal set as it is defined in the shape registration problem. The proof is provided just below.

It will be considered a goal set $\mathbb{A} \subset \mathbb{R}^2$ for ease of writing and as it is the most spread case. Due to this 2D goal set, the 2D motion model in Equation 6.15 will be used for holonomic robot as assumed in this subsection. According to the definition of the function $\mathbf{f}_{H2}$, the uncertain parameter vector is $\mathbf{p} = (v, \delta\theta)$ and the initial pose $\mathbf{x}_0 = (x_0, y_0, \theta_0)$ where $\theta_0 = \theta_d$ when the robot has to follow the direction θ_d. The 2D motion model in Equation 6.15 can be rewritten as follows:

$$\begin{aligned}
\mathbf{f}_{H2}(\mathbf{x}_0, t, \mathbf{p}) &= \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + vt \begin{pmatrix} \cos(\theta_0 + \delta\theta) \\ \sin(\theta_0 + \delta\theta) \end{pmatrix} && (6.144)\\
&= \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + vt \begin{pmatrix} \cos(\theta_0)\cos(\delta\theta) - \sin(\theta_0)\sin(\delta\theta) \\ \sin(\theta_0)\cos(\delta\theta) + \cos(\theta_0)\sin(\delta\theta) \end{pmatrix} && (6.145)\\
&= \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + vt \begin{pmatrix} \cos(\theta_0) & -\sin(\theta_0) \\ \sin(\theta_0) & \cos(\theta_0) \end{pmatrix} \begin{pmatrix} \cos(\delta\theta) \\ \sin(\delta\theta) \end{pmatrix} && (6.146)\\
&= \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + vt \cdot \mathcal{R}(\theta_0) \begin{pmatrix} \cos(\delta\theta) \\ \sin(\delta\theta) \end{pmatrix} && (6.147)
\end{aligned}$$

where $\mathcal{R}(\theta_0)$ corresponds to the 2D rotation matrix.

The set of termination condition according to a defined initial location $\mathbf{q}_0$ in Equation 6.137 and since $\theta_d = \theta_0$ can then be rewritten as follows:

$$\begin{aligned}
TC(\mathbf{q}_0) &= \{(\theta_0, t) \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], \mathbf{f}_{H2}(\mathbf{x}_0, t, \mathbf{p}) \in \mathbb{A}\} && (6.148)\\
&= \{(\theta_0, t) \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], \exists \mathbf{a} \in \mathbb{A}, \mathbf{f}_{H2}(\mathbf{x}_0, t, \mathbf{p}) = \mathbf{a}\} && (6.149)\\
&= \{(\theta_0, t) \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], \exists \mathbf{a} \in \mathbb{A}, \mathbf{q}_0 + vt \cdot \mathcal{R}(\theta_0) \begin{pmatrix} \cos(\delta\theta) \\ \sin(\delta\theta) \end{pmatrix} = \mathbf{a}\} && (6.150)\\
&= \{(\theta_0, t) \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], \exists \mathbf{a} \in \mathbb{A}, vt \cdot \mathcal{R}(\theta_0) \begin{pmatrix} \cos(\delta\theta) \\ \sin(\delta\theta) \end{pmatrix} = \mathbf{a} - \mathbf{q}_0\} && (6.151)\\
& && (6.152)
\end{aligned}$$

where $\mathbf{x}_0 = (x_0, y_0, \theta_0)$.

Let's define the translated set $\mathbb{B}$ from the set $\mathbb{A}$ according to the function:

$$\mathbf{f}_{\mathbf{q}_0}(\mathbf{x}) = \mathbf{x} - \mathbf{q}_0 \tag{6.153}$$

The set $\mathbb{B}$ is then given by: $\mathbb{B} = \mathbf{f}_{\mathbf{q}_0}(\mathbb{A})$.

Now define the polar set $\mathbb{D}$ given by:

$$\mathbb{D} = \{\mathbf{x} \in \mathbb{R}^2 | \exists \mathbf{p} \in [\mathbf{p}], \mathbf{x} = \begin{pmatrix} v\cos(\delta\theta) \\ v\sin(\delta\theta) \end{pmatrix}\} \tag{6.154}$$

$$= \{\mathbf{x} \in \mathbb{R}^2 | \exists \mathbf{p} \in [\mathbf{p}], \mathbf{x} = \begin{pmatrix} p_1\cos(p_2) \\ p_1\sin(p_2) \end{pmatrix}\} \tag{6.155}$$

Finally define the scaled rotation function $\mathbf{f}_{sr} : \mathbb{R}^4 \to \mathbb{R}^2$ as follows:

$$\mathbf{f}_{sr} : \begin{cases} \mathbb{R}^2 \times \mathbb{R}^2 & \to \mathbb{R}^2 \\ (\mathbf{x}, \mathbf{m}) & \to m_2 \cdot \mathcal{R}(m_1) \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \end{cases} \tag{6.156}$$

where $\mathcal{R}$ is the 2D rotation matrix.

By defining $\mathbf{m} = (\theta_0, t)$, the set of termination condition $TC(\mathbf{q}_0)$ in Equation 6.152 is then given as follows:

$$TC(\mathbf{q}_0) = \{\mathbf{m} \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], \exists \mathbf{a} \in \mathbb{A}, vt \cdot \mathcal{R}(\theta_0) \begin{pmatrix} \cos(\delta\theta) \\ \sin(\delta\theta) \end{pmatrix} = \mathbf{a} - \mathbf{q}_0\} \tag{6.157}$$

$$= \{\mathbf{m} \in \mathbb{R}^2 | \forall \mathbf{d} \in \mathbb{D}, \exists \mathbf{a} \in \mathbb{A}, \mathbf{f}_{sr}(\mathbf{d}, \mathbf{m}) = \mathbf{f}_{\mathbf{q}_0}(\mathbf{a})\} \tag{6.158}$$

$$= \{\mathbf{m} \in \mathbb{R}^2 | \mathbf{f}_{sr}(\mathbb{D}, \mathbf{m}) \subset \mathbb{B}\} \tag{6.159}$$

which is indeed a shape registration problem as defined in Equation 5.41 where $\mathbb{A} = \mathbb{D}$ and $\mathbf{p} = \mathbf{m}$ defined in Chapter 5 in section 5.2.3.

Similarly it could be proved that the set giving the backprojection in Equation 6.134 can almost be seen as a shape registration.

Example

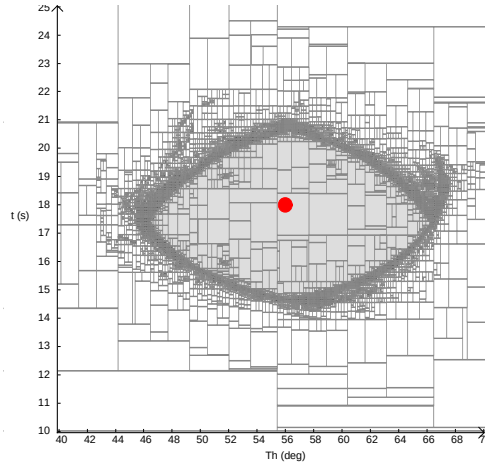
Consider again an uncertainty on the compass $\alpha_\theta = 5°$ and now an uncertainty on the speed $\alpha_v = 10\%$. The desired speed is fixed at $1m/s$. The uncertain parameter vector is $[\mathbf{p}] = [0.9, 1.1] \times [-5, 5](\frac{m}{s} \times deg)$. Define again the same goal set $\mathbb{A}$ as in Figure 6.6 which is depicted by a red circle on the right in Figure 6.18. Consider an initial robot location at $\mathbf{q}_0 = (x_0, y_0) = (15, 15)$. The sets of motion commands that provides the reachability and the recognizability according to Equation 6.137 from this initial location are given in Figure 6.18 with associated sampled trajectories. In Figure 6.18(a), the set of motion commands is computed with projection according to Equation 6.138. Notice that many boxes are unclassified due to the projection and the difficulty to prove that it is actually inside the set $\mathbb{A}$. A motion command is selected at $(\theta_d, t) = (56°, 18s)$ shown by the red disk. According to this motion command, randomly sampled trajectories are represented in blue in Figure 6.18(b) where the last position (at time $t = 18s$) is represented by a yellow AUV. The last poses of the robot at the bounds of the uncertain parameter vector $\mathbf{p}$ are shown by red AUVs.

Using the discretization strategy, the set of motion commands is represented in Figure 6.18(c) where each interval of the interval vector $\mathbf{p}$ was split in 5, which leads to a 5×5 polar grid as depicted in Figure 6.18(d). The trajectories were generated according to the same motion command as earlier with this splitting way. Considering a split with 30 samples, the set of motion commands is represented in Figure 6.18(e) where the polar grid can be viewed in Figure

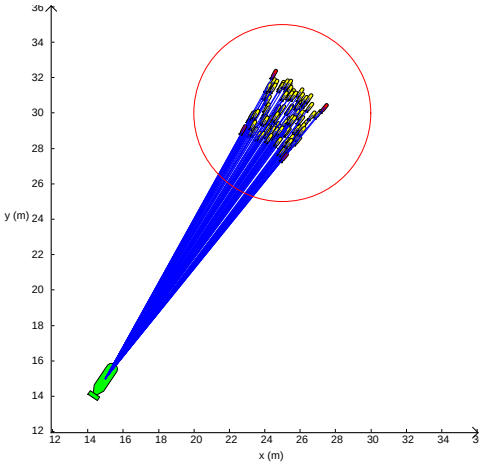
6.18(f). Notice that it does not really improve the result in terms of set computation. However, in this case, the goal set was simple but when dealing with shapes and holes for example, more samples will be needed or the method with projection.

The set of motion commands computed from the shape registration in Equation 6.159 gives a similar result as in Figure 6.18(a) as it relies on projection.

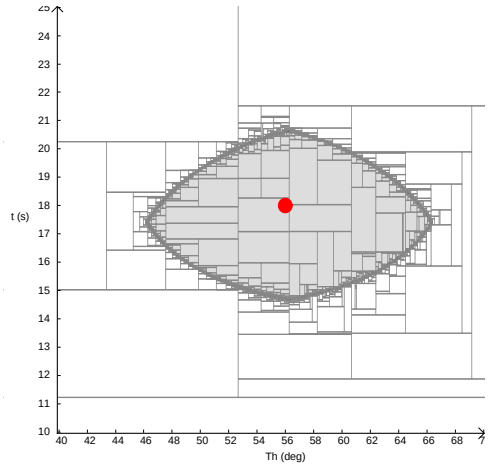
The set of positions from the initial location $(15, 15)$ after the motion command $(\theta_d, t) = (56°, 18s)$ is given in Figure 6.19. Notice that this corresponds to a polar set. A red pie enclosed the solution set. At any time and at any direction, it is possible to represent the set of positions with a pie since it corresponds to a polar set.



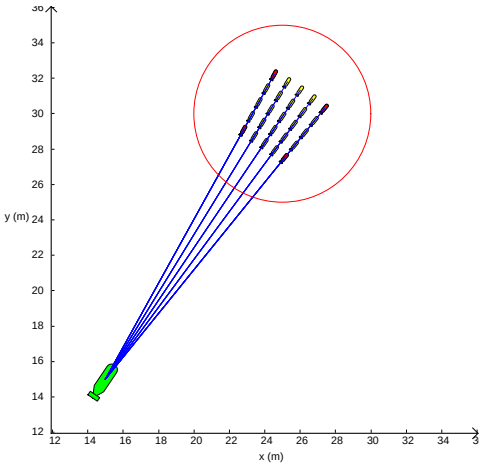
(a) $TC((15, 15))$ computed with projection.



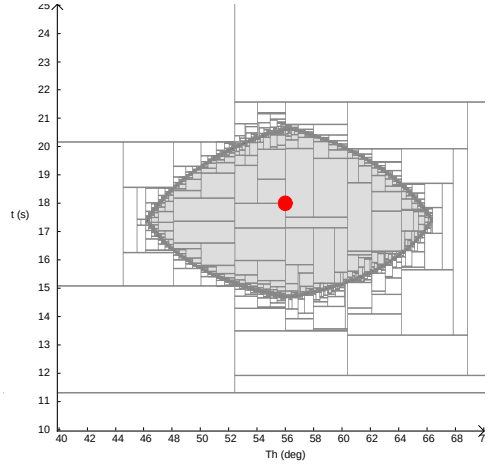
(b) 100 trajectories in blue from the initial location at the motion command represented by the red dot in the figure (a).



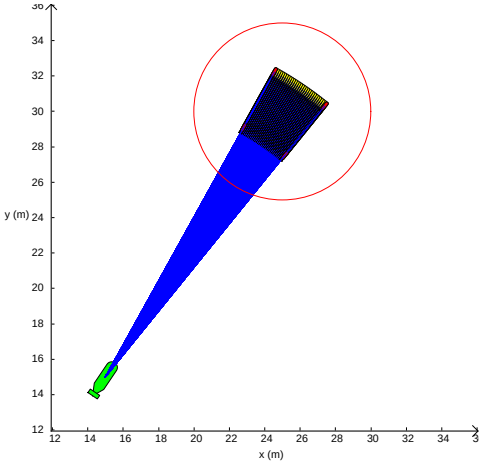
(c) $TC((15, 15))$ computed with sampling defined at 5 along each dimension of $\mathbf{p}$.



(d) Trajectories associated to the sampling (5×5) and the same motion command.



(e) $TC((15, 15))$ computed with sampling defined at 30 along each dimension of $\mathbf{p}$.



(f) Trajectories associated to the sampling (30×30) and to the same motion command.

Figure 6.18: Set of motion commands (θ_d, t) (termination condition) to reach and stop in $\mathbb{A}$ from the initial location $(15, 15)$ and simulated trajectories.

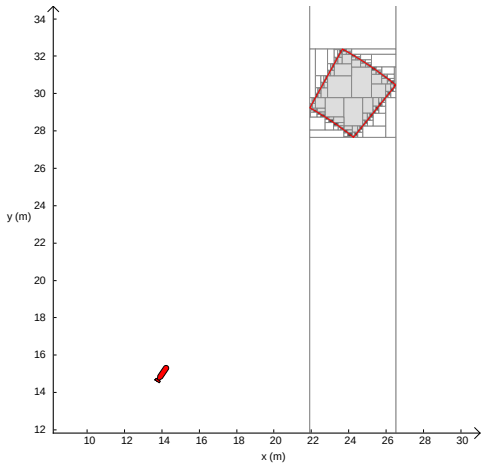


Figure 6.19: Set of positions after a motion command $(\theta_d, t) = (56°, 18s)$ starting from the initial pose $(15, 15, 56°)$ represented by the red AUV.

What if the initial position is uncertainly known?

If the initial position $\mathbf{q}_0$ is uncertainly known but it belongs to an initial set $\mathbb{Q}_0$, it is still possible to provide the set of motion commands. Moreover according to the uncertain initial location, a thick set representation is also possible.

Consider that the initial location $\mathbf{q}_0 \in \mathbb{Q}_0 \subset \mathbb{R}^2$, the set of motion commands whatever the initial location $\mathbf{q}_0$ is then expressed as follows:

$$TC(\mathbb{Q}_0)^- = \{(\theta_d, t) \in \mathbb{R}^2 | \forall \mathbf{q}_0 \in \mathbb{Q}_0, \forall \mathbf{p} \in [\mathbf{p}], \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \tag{6.160}$$

where $\mathbf{x} = (\mathbf{q}_0, \theta_d)$.

Consider now the set:

$$TC(\mathbb{Q}_0)^+ = \{(\theta_d, t) \in \mathbb{R}^2 | \exists \mathbf{q}_0 \in \mathbb{Q}_0, \forall \mathbf{p} \in [\mathbf{p}], \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \tag{6.161}$$

The set of motion commands $TC(\mathbf{q}_0)$ with $\mathbf{q}_0 \in \mathbb{Q}_0$ is enclosed between $TC(\mathbb{Q}_0)^-$ (strong termination condition) and $TC(\mathbb{Q}_0)^+$ (weak termination condition). This gives finally:

$$TC(\mathbf{q}_0) \in [TC(\mathbb{Q}_0)^-, TC(\mathbb{Q}_0)^+] \tag{6.162}$$

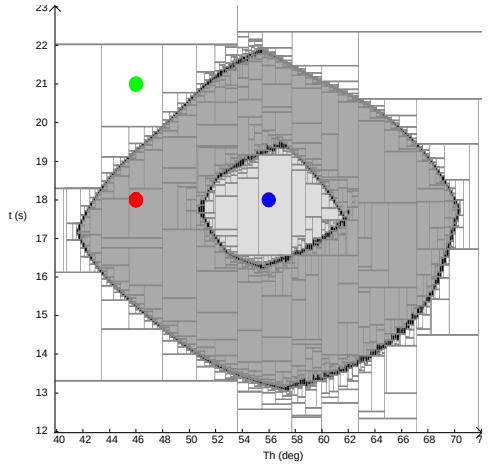
Example

Consider the same example as in Figure 6.18. The initial position of the robot is assumed to belong to a box:

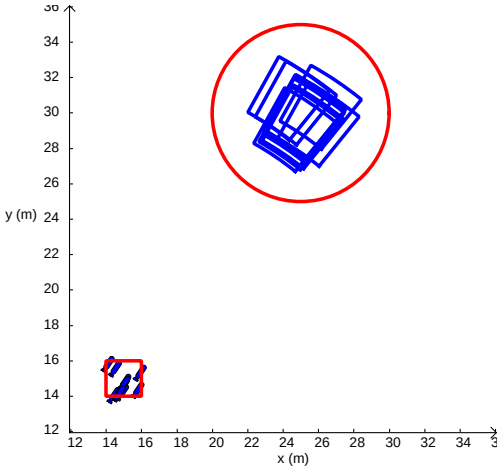
$$\mathbf{q}_0 \in \mathbb{Q}_0 = [14, 16]^2 \tag{6.163}$$

This corresponds to the previous certain position with an inflation of $1m$ (uncertainty defined at $1m$). The results are presented in Figure 6.20 where the initial red box can be seen in Figure 6.20(b),(c) or (d). The solution set $TC(\mathbf{q}_0) \in [TC(\mathbb{Q}_0)^-, TC(\mathbb{Q}_0)^+]$ is depicted by a thick representation in Figure 6.20 where the discretization strategy was used. Notice how the penumbra (dark gray) is large compared to the inner approximation due to the large initial location. Some motion commands represented by dots are selected according to the different parts of the thick set. In Figure 6.20(b)(c) and (d), blue AUVs correspond to correct initial location which guarantee that after the selected motion command all the robot positions, represented then by blue pies, end in the goal set (red circle). Green AUVs do not provide this

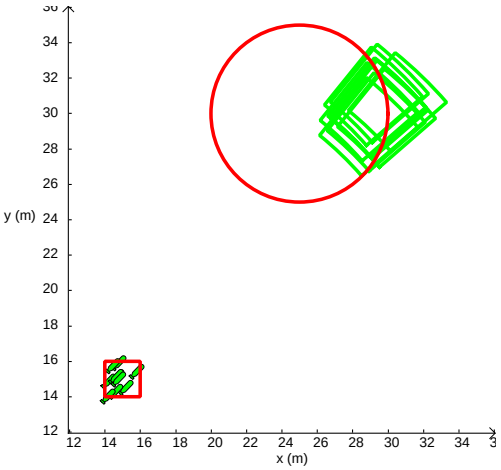
guarantee as some positions are outside the goal set. A random sampling has been performed to select 10 initial robot locations in the red box. In Figure 6.20(a), it presents the result for the motion command $(56°, 18s)$ depicted by the blue dot which is inside the solution set. As it can be noticed this motion command guarantees indeed that all the robot positions end in the goal set. The goal set is then reached and recognized. In Figure 6.20(c), the motion command $(48°, 21s)$ (green dot) is selected outside the solution set. All the robot positions after this motion command for each initial positions are not totally in the goal set but only partially (in green). The goal has been reached but not recognized temporally. Finally in Figure 6.20(d), the motion command $(48°, 18s)$ is selected inside the penumbra (red dot). All the poses from some initial poses are inside the goal set, represented in blue, and some are only partially, represented in green. Indeed, this motion command only guarantees that some initial robot locations in the red box will lead all the robot positions after executing this motion in the goal set.



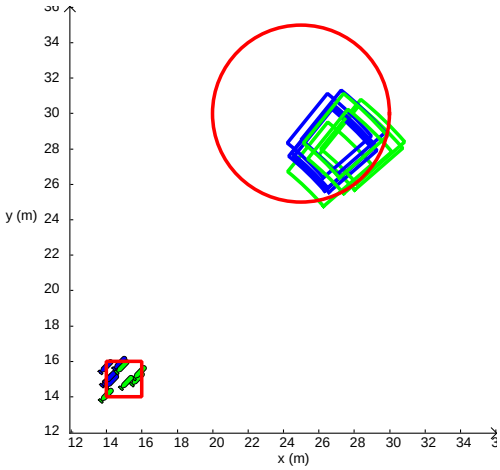
(a) Thick solution of the set $TC(\mathbf{q}_0)$. Grayscale convention for thick sets.



(b) Motion command $(56°, 18s)$ corresponding to the blue dot inside the solution set in the figure (a). All the robot poses end in the goal set after this motion command.



(c) Motion command $(48°, 21s)$ corresponding to the green dot outside the solution set in the figure (a). All the robot poses do not end in the goal set after this motion command.



(d) Motion command $(48°, 18s)$ corresponding to the red dot in the penumbra in the figure (a). Some of the robot poses (blue) end in the goal set and some (green) do not after this motion command (only partially).

Figure 6.20: Thick representation of the motion command due to uncertain initial location of the robot in the red box. The propagation of the robot positions is represented by pie as in Figure 6.19. Blue AUVs correspond to correct initial poses with the associated blue pies that are inside the goal set (red circle). Green ones are wrong.

Motion commands with uncertain initial position and Minkowski operators

Finding the motion commands has been solved previously considering the projection (shape registration or directly) or the discretization even when the position was uncertainly known.

Consider an initial position $\mathbf{q} \in \mathbb{Q}_0 \subset \mathbb{R}^2$ and a goal set $\mathbb{A} \subset \mathbb{R}^2$ as in the previous proof about the goal recognizability as a shape registration.

Introduce again $\mathbf{m} = (\theta_d, t) = (\theta_0, t)$.

The strong termination condition given in Equation 6.160 can be rewritten as follows:

$$TC(\mathbb{Q}_0)^- = \{\mathbf{m} \in \mathbb{R}^2 | \forall \mathbf{q}_0 \in \mathbb{Q}_0, \forall \mathbf{p} \in [\mathbf{p}], \mathbf{f}_{H2}(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \quad (6.164)$$

$$= \{\mathbf{m} \in \mathbb{R}^2 | \forall \mathbf{q}_0 \in \mathbb{Q}_0, \forall \mathbf{p} \in [\mathbf{p}], \mathbf{q}_0 + v \cdot m_2 \cdot \begin{pmatrix} \cos(m_1 + \delta\theta) \\ \sin(m_1 + \delta\theta) \end{pmatrix} \in \mathbb{A}\} \quad (6.165)$$

$$= \{\mathbf{m} \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], \mathbb{Q}_0 + v \cdot m_2 \cdot \begin{pmatrix} \cos(m_1 + \delta\theta) \\ \sin(m_1 + \delta\theta) \end{pmatrix} \subset \mathbb{A}\} \quad (6.166)$$

which corresponds almost to a Minkowski difference defined in Equation 5.53 where the parameter vector $\mathbf{p}$ corresponds here to $v \cdot m_2 \cdot \begin{pmatrix} \cos(m_1 + \delta\theta) \\ \sin(m_1 + \delta\theta) \end{pmatrix}$.

Define then as $\mathbb{H}(\mathbb{Q}_0)^-$ the set corresponding to:

$$\mathbb{H}(\mathbb{Q}_0)^- = \{\mathbf{h} \in \mathbb{R}^2 | \mathbb{Q}_0 + \mathbf{h} \subset \mathbb{A}\} \quad (6.167)$$

$$= \mathbb{A} \ominus \mathbb{Q}_0 \quad (6.168)$$

Finally,

$$TC(\mathbb{Q}_0)^- = \{\mathbf{m} \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], v \cdot m_2 \cdot \begin{pmatrix} \cos(m_1 + \delta\theta) \\ \sin(m_1 + \delta\theta) \end{pmatrix} \in \mathbb{H}(\mathbb{Q}_0)^-\} \quad (6.169)$$

$$= \{\mathbf{m} \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], v \cdot m_2 \cdot \begin{pmatrix} \cos(m_1 + \delta\theta) \\ \sin(m_1 + \delta\theta) \end{pmatrix} \in \mathbb{A} \ominus \mathbb{Q}_0\} \quad (6.170)$$

The weak termination condition $TC(\mathbb{Q}_0)^+$ Equation 6.161 can be rewritten as follows:

$$TC(\mathbb{Q}_0)^+ = \{\mathbf{m} \in \mathbb{R}^2 | \exists \mathbf{q}_0 \in \mathbb{Q}_0, \forall \mathbf{p} \in [\mathbf{p}], \mathbf{f}_{H2}(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \quad (6.171)$$

$$= \{\mathbf{m} \in \mathbb{R}^2 | \exists \mathbf{q}_0 \in \mathbb{Q}_0, \forall \mathbf{p} \in [\mathbf{p}], \exists \mathbf{a} \in \mathbb{A}, \mathbf{f}_{H2}(\mathbf{x}, t, \mathbf{p}) = \mathbf{a}\} \quad (6.172)$$

$$= \{\mathbf{m} \in \mathbb{R}^2 | \exists \mathbf{q}_0 \in \mathbb{Q}_0, \forall \mathbf{p} \in [\mathbf{p}], \exists \mathbf{a} \in \mathbb{A}, \mathbf{q}_0 + v \cdot m_2 \cdot \begin{pmatrix} \cos(m_1 + \delta\theta) \\ \sin(m_1 + \delta\theta) \end{pmatrix} = \mathbf{a}\} \quad (6.173)$$

$$= \{\mathbf{m} \in \mathbb{R}^2 | \exists \mathbf{q}_0 \in \mathbb{Q}_0, \forall \mathbf{p} \in [\mathbf{p}], \exists \mathbf{a} \in \mathbb{A}, \mathbf{a} + (-\mathbf{q}_0) = v \cdot m_2 \cdot \begin{pmatrix} \cos(m_1 + \delta\theta) \\ \sin(m_1 + \delta\theta) \end{pmatrix}\} \quad (6.174)$$

which corresponds almost to a Minkowski sum.

Define then as $\mathbb{H}(\mathbb{Q}_0)^+$ the set corresponding to:

$$\mathbb{H}(\mathbb{Q}_0)^+ = \{\mathbf{h} \in \mathbb{R}^2 | \exists \mathbf{q}_0 \in \mathbb{Q}_0, \exists \mathbf{a} \in \mathbb{A}, \mathbf{h} = \mathbf{a} + (-\mathbf{q}_0)\} \quad (6.175)$$

$$= \mathbb{A} \oplus (-\mathbb{Q}_0) \quad (6.176)$$

Finally,

$$TC(\mathbb{Q}_0)^+ = \{\mathbf{m} \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], v \cdot m_2 \cdot \begin{pmatrix} \cos(m_1 + \delta\theta) \\ \sin(m_1 + \delta\theta) \end{pmatrix} \in \mathbb{H}(\mathbb{Q}_0)^+\} \quad (6.177)$$

$$= \{\mathbf{m} \in \mathbb{R}^2 | \forall \mathbf{p} \in [\mathbf{p}], v \cdot m_2 \cdot \begin{pmatrix} \cos(m_1 + \delta\theta) \\ \sin(m_1 + \delta\theta) \end{pmatrix} \in \mathbb{A} \oplus (-\mathbb{Q}_0)\} \quad (6.178)$$

Since usually $\mathbb{Q}_0$ is represented by a box, a circle or an ellipse which are simple sets, and the set $\mathbb{A}$ may be very complex (random shape), it will be preferred to take $-(\mathbb{Q}_0 \oplus (-\mathbb{A}))$ which is equal to $\mathbb{A} \oplus (-\mathbb{Q}_0)$.

The names $\mathbb{H}(\mathbb{Q}_0)^-$ and $\mathbb{H}(\mathbb{Q}_0)^+$ do not correspond to any thick set representation and dot have any link, they were simply given as intermediate variables.

Finally the thick set representation of the set of motion commands $TC(\mathbf{q}_0)$ with $\mathbf{q}_0 \in \mathbb{Q}_0$ can be computed according to a Minkowski difference and a Minkowski sum to provide the enclosure given in Equation 6.162 and reminded here:

$$TC(\mathbf{q}_0) \in [TC(\mathbb{Q}_0)^-, TC(\mathbb{Q}_0)^+] \tag{6.179}$$

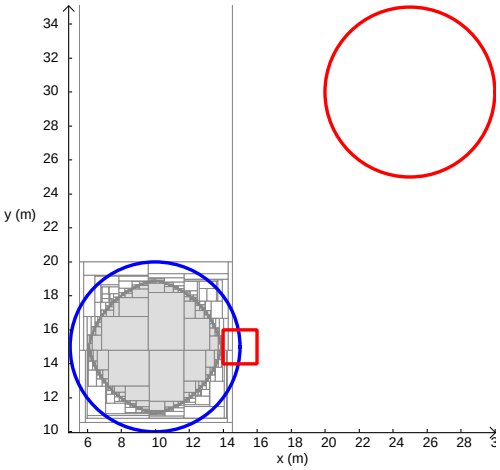
Example

Consider the same example to be able to compare the results. The set $\mathbb{A}$ is represented by a red circle and the uncertain initial pose $\mathbb{Q}_0$ by a red box in Figure 6.21(a) and (c).

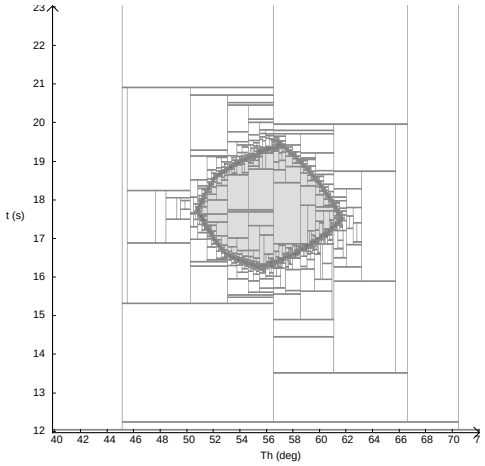
Firstly the computation of the Minkowski difference $\mathbb{A} \ominus \mathbb{Q}_0$ from Equation 6.168 giving the intermediate set $\mathbb{H}(\mathbb{Q}_0)^-$ is provided in Figure 6.21(a) where the blue circle, corresponding to a translation of the red circle, has been drawn to highlight the Minkowski difference known also as erosion in image processing. Finally the set of motion commands $TC(\mathbb{Q}_0)^-$, defined as strong motion commands, is represented in Figure 6.21(b). Notice that this set is the same as in the lower bound of the thick representation represented by the soft gray region in Figure 6.20(a) (the scales of the figure are the same).

Secondly, the Minkowski sum $\mathbb{A} \oplus (-\mathbb{Q}_0)$ from Equation 6.176 giving the intermediate set $\mathbb{H}(\mathbb{Q}_0)^-$ is provided in Figure 6.21(c) where the blue circle has also been added to highlight the Minkowski sum known as dilation in image processing. Finally the set of motion commands $TC(\mathbb{Q}_0)^+$, defined as weak motion commands, is represented in Figure 6.21(d). Notice again that the set corresponds to the upper bound of the thick set in Figure 6.20(a).

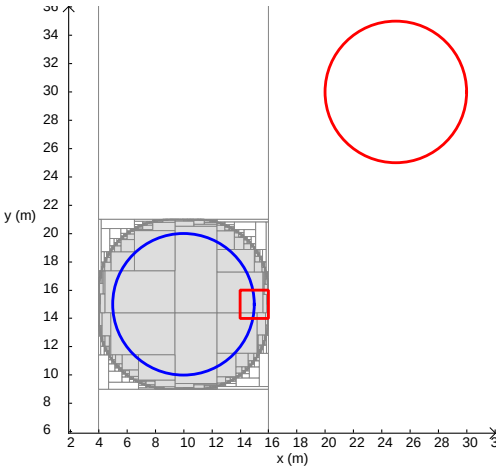
Finally the thick set representation $[TC(\mathbb{Q}_0)^-, TC(\mathbb{Q}_0)^+]$ given by Equation 6.162 based on the two sets is provided in Figure 6.21(e) which is the same thick solution as in Figure 6.20(a).



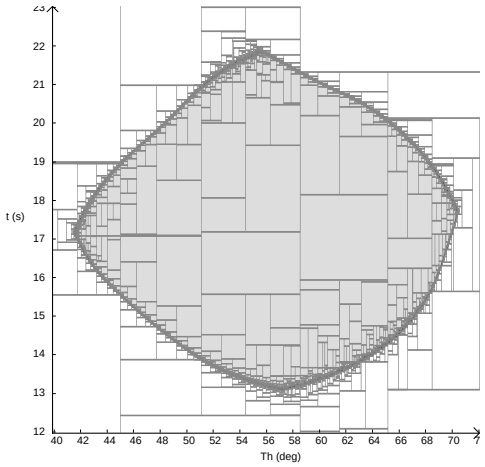
(a) Minkowski difference $\mathbb{A} \ominus \mathbb{Q}_0 = \mathbb{H}(\mathbb{Q}_0)^-$. $\mathbb{A}$ is represented by the red circle and $\mathbb{Q}_0$ by the red box. The blue circle is the translated circle of $\mathbb{A}$ to highlight the erosion.



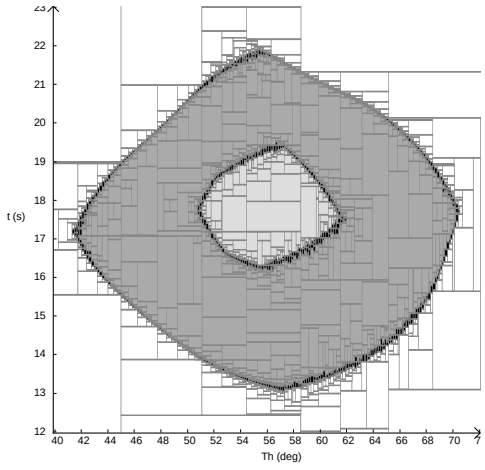
(b) Strong termination condition $TC(\mathbb{Q}_0)^-$.



(c) Minkowski sum $\mathbb{A} \oplus (-\mathbb{Q}_0) = \mathbb{H}(\mathbb{Q}_0)^+$.



(d) Weak termination condition $TC(\mathbb{Q}_0)^+$.



(e) Thick representation of $TC(\mathbf{q}_0) = [TC(\mathbb{Q}_0)^-, TC(\mathbb{Q}_0)^+]$ with $\mathbf{q}_0 \in \mathbb{Q}_0$.

Figure 6.21: Computation of the motion command with an uncertain initial location $\mathbf{q}_0$ of the robot in the red box $\mathbb{Q}_0$ using Minkowski operators.

6.4.3 Dubins path backprojection

When dealing with nonholonomic robot, the Dubins paths were considered to take into account a potential non null minimum turning radius. The model of the motion $\mathbf{f}_{NH}$ was proposed in Equation 6.13 in the previous section that provides a parametric Dubins paths with uncertain parameters. Since the motion is composed of several elementary motions such as circular or linear displacements, a time vector $\mathbf{t}$ (or duration) was introduced to describe all these portions. This vector corresponded to $\mathbf{t} = \{t_{c_1}, t_s, t_{c_2}\}$ where t_{c_1}, t_s and t_{c_2} were respectively the duration of the first circular motion, then the linear portion and finally the circular portion. Several Dubins paths were considered: RSL, LSL, RSR and LSR. A strategy μ corresponding to one of these Dubins paths is defined as follows: $\mu \in \{RSL, LSL, RSR, LSR\}$. If only the RSR motion is considered, then $\mu = RSR$. It will be considered in the following that the final portion of the motion, i.e. a circular portion, has to enter in the goal set for each of the possible Dubins motions. According to Equation 6.18 based on the general motion model in Equation 6.16, the Dubins backward reach set of a set $\mathbb{A} \subset \mathbb{R}^3$ is then defined as follows:

$$BACK(\mathbb{A}, \mu) = \{\mathbf{x} \in \mathbb{R}^3 | \exists (t_{c_1}, t_s) \in \mathbb{T}_{c_1} \times \mathbb{T}_s, \forall \mathbf{p} \in [\mathbf{p}], \exists t_{c_2} \in \mathbb{T}_{c_2}, \mathbf{f}_\mu(\mathbf{x}, \mathbf{t}, \mathbf{p}) \in \mathbb{A}\} \quad (6.180)$$

where $\mu \in \{RSR, RSL, LSL, LSR\}$, $\mathbf{t} = (t_{c_1}, t_s, t_{c_2}) \in \mathbb{T}_{c_1} \times \mathbb{T}_s \times \mathbb{T}_{c_2}$ and $\mathbf{p}$ the uncertain parameter vector. $\mathbb{T}_{c_1}$, $\mathbb{T}_s$ and $\mathbb{T}_{c_2}$ are intervals and $\mathbf{p} \in [\mathbf{p}]$ (interval vector). The notation $\mathbf{f}_\mu$ will be kept but it refers to the function $\mathbf{f}_{NH}$ with the right angular velocities defined according to the desired motion.

Notice that indeed, only in the last portion at time t_{c_2}, the pose of the robot has to belong to the goal set $\mathbb{A}$. The reachability of the Dubins paths is the modelled according to the first two portions of the motion, i.e. according to t_{c_1} and t_s. This can be noticed by the place of the quantifiers which appear before the uncertain parameters. Indeed, the reachability has to be guaranteed on the last portion of the motion despite the uncertainties, therefore it needs the existence of durations for the first two portions.

Since the angular speed w is assumed fixed, the domain of the duration $\mathbb{T}_{c_1}$ and $\mathbb{T}_{c_2}$ can be reduced to:

$$\mathbb{T}_{c_1} = \mathbb{T}_{c_2} = [0, \frac{2\pi}{w}] \quad (6.181)$$

Indeed, according to w and due to the fact that the motion is circular, it will be assumed only a single round.

Considering all the possible Dubins paths, the general backward reach set is then defined as follows:

$$BACK(\mathbb{A}) = \bigcup_{\mu \in \{RSR, RSL, LSL, LSR\}} BACK(\mathbb{A}, \mu) \quad (6.182)$$

which gives the whole backward reach set according to the four possible paths. It will then indicates that at least one Dubins paths is possible.

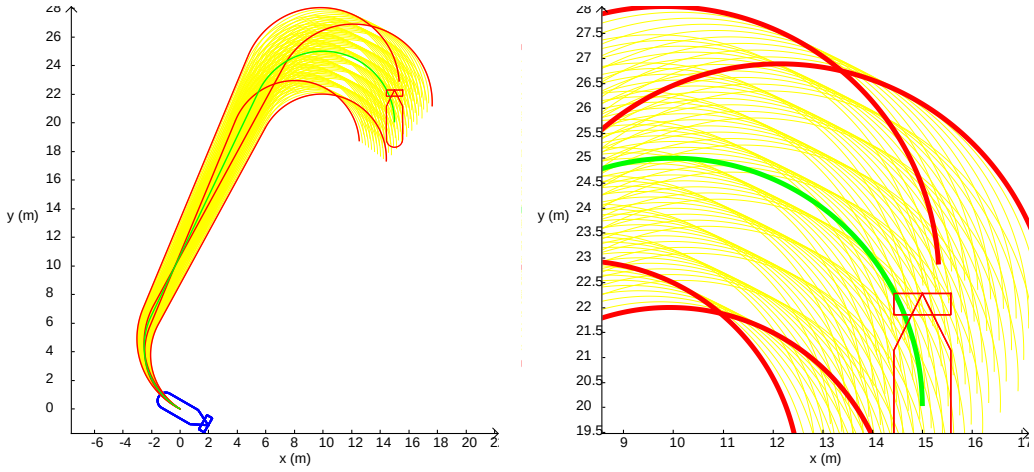
According to Equation 6.180 and similarly to Equation 6.23, the backward reach set for a strategy $\mu \in \{RSR, RSL, LSL, LSR\}$ of Dubins path can be defined with projections as follows:

$$BACK(\mathbb{A}, \mu) = proj_{\mathbb{X}}^{\mathbb{T}_{c_1} \times \mathbb{T}_s}(\overline{proj_{\mathbb{X} \times \mathbb{T}_{c_1} \times \mathbb{T}_s}^{[\mathbf{p}]}(\overline{proj_{\mathbb{X} \times \mathbb{T}_{c_1} \times \mathbb{T}_s \times \mathbb{P}}^{\mathbb{T}_{c_2}}(\mathbf{f}_\mu^{-1}(\mathbb{A}))})}) \quad (6.183)$$

where $\mathbb{P}$ corresponds to the domain of the uncertain parameter vector.

Contrary to the main work on directional backprojection, now the uncertainty on the speed is considered which means that $\mathbf{p} \in [\mathbf{p}] = [v_d(1-\alpha_v), v_d(1+\alpha_v)] \times [-\alpha_\theta, \alpha_\theta]$ where $\alpha_v \geq 0$ and $\alpha_\theta \geq 0$.

Notice that the set defined in Equation 6.183 is composed of three levels of projection. Consider now only the trajectories at the bounds of the uncertain parameter vector $\mathbf{p}$, i.e. $v \in \{v_d(1-\alpha_v), v_d(1+\alpha_v)\} = \{v_{min}, v_{max}\}$ and $\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}$. According to a defined time vector $\mathbf{t} = (t_{c_1}, t_s, t_{c_2})$, it will be assumed that all the trajectories are enclosed between these trajectories on the bounds as depicted in Figure 6.22(a). The time command $\mathbf{t}$ between an initial configuration in blue and a final configuration in red is computed and the corresponding path without uncertainty is represented in green. Uniformly sampled trajectories in the interval errors are shown in yellow and the trajectories on the bounds are represented in red. However this enclosure is under approximative as it can be seen on the second turn where some trajectories as depicted in Figure 6.22(b) at the top go outside this enclosure. This will be considered insignificant since these trajectories come back in the enclosure and could reach the goal set.



(a) Uncertain trajectories represented in yellow where a uniform sampling in the error intervals. Red trajectories are at the bounds of the error intervals.

(b) Second turn where few trajectories go outside the enclosure.

Figure 6.22: Uncertain Dubins path according to the motion command linking the initial blue pose and the final red pose represented by the green nominal trajectory.

Under this approximation, the backward reach set firstly defined in Equation 6.180 can be reformulated as follows:

$$
\begin{aligned}
BACK(\mathbb{A}, \mu) &= \{\mathbf{x} \in \mathbb{R}^3 | \exists (t_{c_1}, t_s) \in \mathbb{T}_{c_1} \times \mathbb{T}_s, \forall v \in \{v_{min}, v_{max}\}, \forall \delta\theta \in \{-\alpha_\theta, \alpha_\theta\}, \exists t_{c_2} \in \mathbb{T}_{c_2} \\
&\quad \mathbf{f}_\mu(\mathbf{x}, \mathbf{t}, \mathbf{p}) \in \mathbb{A}\} \qquad (6.184) \\
&= proj_{\mathbb{X}}^{\mathbb{T}_{c_1} \times \mathbb{T}_s} \Big(\bigcap_{v \in \{v_{min}, v_{max}\}} \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj_{\mathbb{X} \times \mathbb{T}_{c_1} \times \mathbb{T}_s}^{\mathbb{T}_{c_2}} (\mathbf{f}_\mu^{-1}(\mathbb{A})) \Big) \qquad (6.185)
\end{aligned}
$$

where $v_{min} = v_d(1-\alpha_v)$, $v_{max} = v_d(1+\alpha_v)$ with v_d the desired speed. $\mathbb{X} \subset \mathbb{R}^3$ is the domain of $\mathbf{x} = (x, y, \theta)$. As it can be noticed, only two levels of projection are needed. Therefore the computation time is reduced. This set indicates the initial robot poses such that all the trajectories despite the uncertain parameter $\mathbf{p}$ it enters a set $\mathbb{A}$ according to a Dubins path defined by the strategy μ.

If the initial pose of the robot is known $\mathbf{x}(0) = \mathbf{x}_0$, the time parameter (t_{c_1}, t_s), defined as the time reachability, to guarantee the existence of a time t_{c_2} such that all trajectories cross the set $\mathbb{A}$ can be determined as follows:

$$\mathbb{T}_{reach}(\mathbf{x}_0) = \{(t_{c_1}, t_s) \in \mathbb{T}_{c_1} \times \mathbb{T}_s | \forall \mathbf{p} \in [\mathbf{p}], \exists t_{c_2} \in \mathbb{T}_{c_2}, \mathbf{f}_\mu(\mathbf{x}_0, \mathbf{t}, \mathbf{p}) \in \mathbb{A}\} \tag{6.186}$$

which can be reformulated with projection according to the trajectories on the bounds:

$$\mathbb{T}_{reach}(\mathbf{x}_0) = \bigcap_{v \in \{v_{min}, v_{max}\}} \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj^{\mathbb{T}_{c_2}}_{\mathbb{T}_{c_1} \times \mathbb{T}_s}(\mathbf{f}_\mu^{-1}(\mathbb{A})) \tag{6.187}$$

where $\mathbf{x} = \mathbf{x}_0$ is imposed in the function $\mathbf{f}_\mu$.

When the initial position $(x - y)$ is uncertain (there is already an uncertain parameter for the heading $\delta\theta$), i.e. $\mathbf{x}_0$ belongs to an interval vector $\mathbf{x}_0 \in [\mathbf{x}_0]$ or to set $\mathbf{x}_0 \in \mathbb{X}_0$, a similar computation as in the previous subsection can be performed by using directly the projection or with Minkowski operators.

Example

Consider the following errors: $\alpha_v = 10\%$ and $\alpha_\theta = 5°$.

Consider the goal set defined by a translated ellipse as follows:

$$\mathbb{A} = \{\mathbf{x} \in \mathbb{R}^3 | (\frac{x_1 - 50}{5})^2 + (\frac{x_2 - 50}{1})^2 - 1 \leq 0\} \tag{6.188}$$

Notice that the third dimension has no influence meaning that any directions of the robot is valid. This set is illustrated in Figure 6.23(a).

Consider a departure set $\mathbb{B}$ defined as follows:

$$\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^3 | \sqrt{(x_1 - 35)^2 + (x_2 - 35)^2} \leq 8 \text{ and } x_3 = 0°\} \tag{6.189}$$

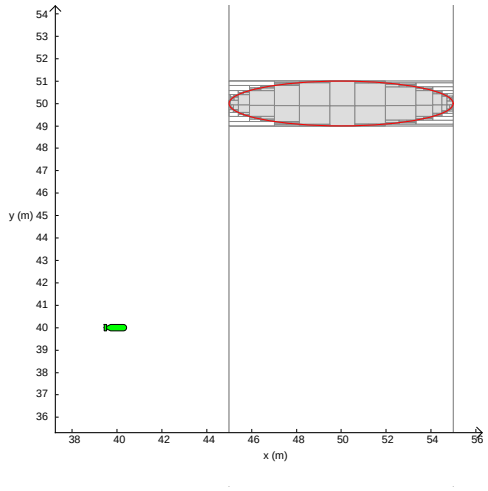
This corresponds to a disk at $\theta_d = 0°$.

Considering the strategy $\mu = LSL$, i.e. $w_{c_1} = w_{c_2} = w$ with w defined at $0.3rad/s$, the set of robot position coming from $\mathbb{B}$ and reaching $\mathbb{A}$ is given by:

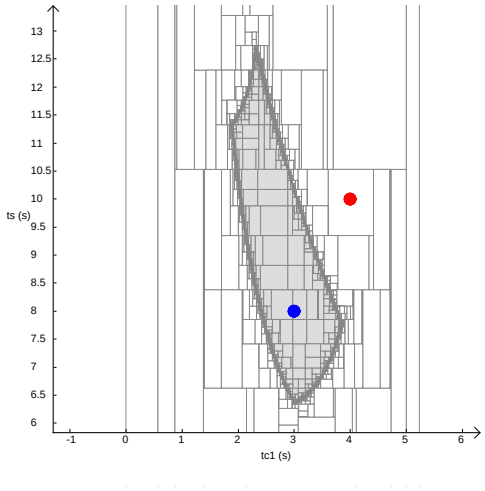
$$\mathbb{P}_{LSL} = \{\mathbf{x} \in \mathbb{R}^3 | BACK(\mathbb{A}, LSL) \cap \mathbb{B} \neq \emptyset\} \tag{6.190}$$

where $BACK(\mathbb{A}, LSL)$ is given by Equation 6.185 with $\mu = LSL$.

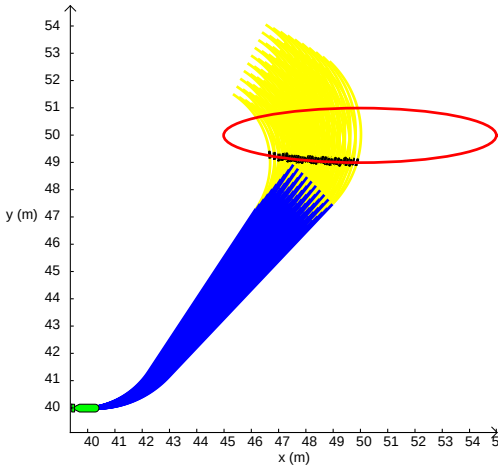
Consider now the initial pose $\mathbf{x}_0 = (40, 40, 0°)$, the time reachability given in Equation 6.187 is represented in Figure 6.23(b) where two time commands (t_{c_1}, t_s) are shown by dots. The blue dot corresponds to a time command that guarantees the reachability of the ellipse as depicted in Figure 6.23(c) where the yellow trajectories are the last portions of the motion (left turn) and the blue trajectories correspond to the two first portions. The black AUVs are the positions when they enter the set $\mathbb{A}$ in this last portion. The trajectories corresponding to the red dot are depicted in Figure 6.23(d). Notice that some last portions of some trajectories begin after the ellipse or even pass by the ellipse.



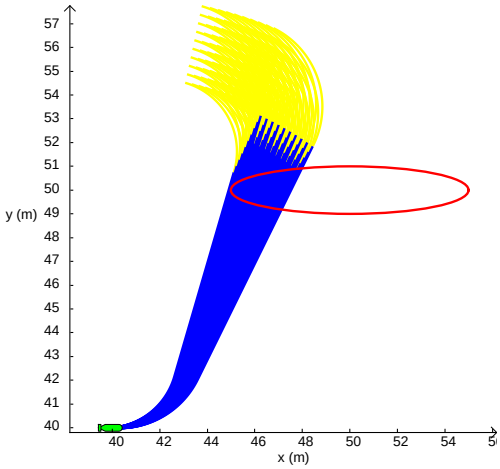
(a) Initial pose represented by the green AUV and a goal set defined as an ellipse.



(b) $\mathbb{T}_{reach}$ at this initial pose with a LSL motion.



(c) Trajectories according to the motion command represented by the blue dot (inside the set) in the figure (b). Black AUVs correspond to the robot poses when it reaches the ellipse on the last portion (left turn) represented by yellow curves.



(d) Trajectories according to the motion command represented by the red dot(outside the set) in the figure (b). Some trajectories do not go through the ellipse and some last portion never cross the ellipse.

Figure 6.23: Time reachability of the ellipse a the initial pose $(40, 40, 0°)$ with a LSL motion. It has to reach the ellipse only in the last left turn.

As you may imagine, since the angle of arrival has no influence on the backward reachability, it is possible to consider only a LS motion instead of LSL (or RS). This can be given with $t_{c_2} = 0s$. The backward reachability consists now in proving that there exists some time t_s such that it reaches $\mathbb{A}$, it is given as follows:

$$BACK(\mathbb{A}, \mu) = \{\mathbf{x} \in \mathbb{R}^3 | \exists t_{c_1} \in \mathbb{T}_{c_1}, \forall \mathbf{p} \in [\mathbf{p}], \exists t_s \in \mathbb{T}_s, \mathbf{f}_\mu(\mathbf{x}, \mathbf{t}, \mathbf{p}) \in \mathbb{A}\} \tag{6.191}$$

where $\mathbf{t} = (t_{c_1}, t_s, 0)$.

Similarly the backward reach set by working on the bounds can be given with projection:

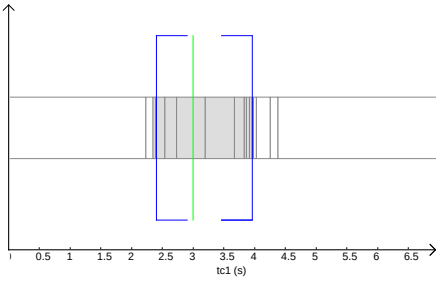
$$BACK(\mathbb{A}, \mu) = proj_{\mathbb{X}}^{\mathbb{T}_{c_1}} \Big(\bigcap_{v \in \{v_{min}, v_{max}\}} \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj_{\mathbb{X} \times \mathbb{T}_{c_1}}^{\mathbb{T}_s} (\mathbf{f}_\mu^{-1}(\mathbb{A})) \Big) \tag{6.192}$$

and finally the time reachability at a given pose $\mathbf{x}_0$ is given by:

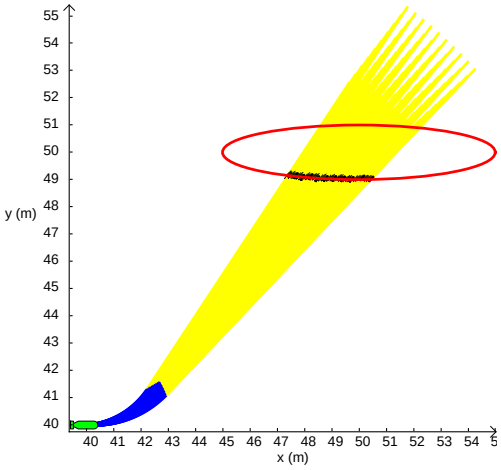
$$\mathbb{T}_{reach}(\mathbf{x}_0) = \{t_{c_1} \in \mathbb{T}_{c_1} | \forall \mathbf{p} \in [\mathbf{p}], \exists t_s \in \mathbb{T}_s, \mathbf{f}_\mu(\mathbf{x}_0, \mathbf{t}, \mathbf{p}) \in \mathbb{A}\} \quad (6.193)$$

$$= \bigcap_{v \in \{v_{min}, v_{max}\}} \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj^{\mathbb{T}_s}_{\mathbb{T}_{c_1}}(\mathbf{f}_\mu^{-1}(\mathbb{A})) \quad (6.194)$$

At the same initial pose, the time reachability is given in Figure 6.24(a) where the blue interval shows the inner approximation. Only the values along the $x-$axis are relevant. The trajectories at the motion command t_{c_1} indicates by the green are represented in Figure 6.24(b) where the blue trajectories are the first portion of the motion and the yellow trajectories the second and last portion of the motion. The black AUV indicate when the AUV enter the ellipse. In this example, only a motion LS is sufficient to reach this 2D ellipse. However, if the angle of arrival was defined, it would perhaps have led to an empty solution and therefore the complete Dubins path would be necessary.



(a) $\mathbb{T}_{reach}$ at the initial pose $(40, 40, 0°)$ with a LS motion. Blue interval is the inner approximation.



(b) Trajectories according to the motion command defined by t_{c_1} represented by the green line in the figure (a). Yellow lines are the second (last) portion of the motion (straight). Black AUVs are the poses when the robot enters the ellipse.

Figure 6.24: Reachability of the ellipse at the initial pose $(40, 40, 0°)$ with a LS motion.

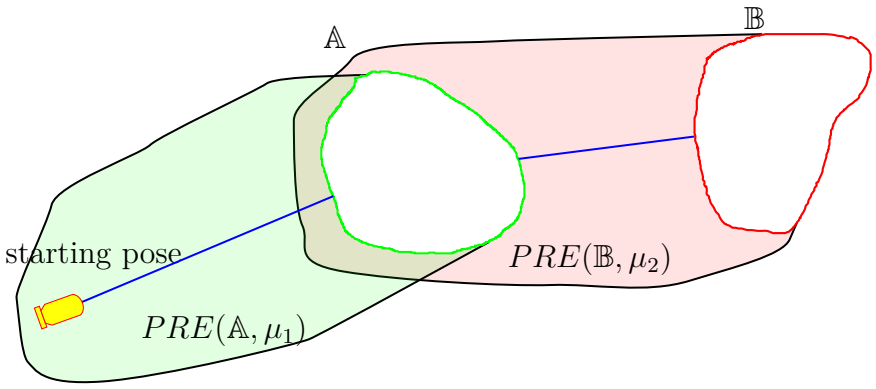
6.5 Graph building

In the previous section, the main tools to compute preimages as backprojection under a strategy μ from a goal set that may be defined by random shapes have been presented according to the two motion models used to describe the behaviour of the robot. The motion of the robot is subject to drift which has been introduced with uncertain parameters defined between bounds to remain in a set membership context. The backprojection has been computed using separators and projection introduced in Chapter 4. This backprojection under a strategy μ such as "follow a direction" enables to indicate all the robot poses from there if it follows this strategy, the robot will for sure reach the desired set. It will recognize the desired set based on exteroceptive measurements for example or by computing the motion command (θ_d, t) which proves that all the robot poses from an initial location will reach the desired goal set at a defined direction and after a particular duration of motion when the robot is considered holonomic. As presented in

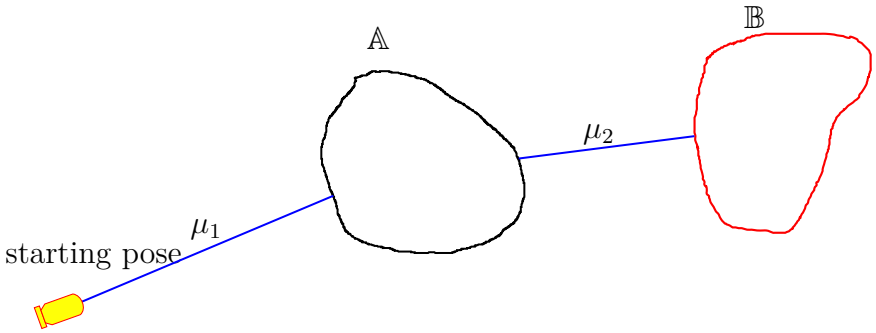
the **Important note** in the previous section, the problem of the backprojection of sets can be seen as a graph construction.

6.5.1 Definition

The initial problem of the motion planning was introduced in Section 6.2 where it consists in finding a sequence of open loop strategies μ_i to join a goal set based on backprojection of sets. These sets, defined as relocation area, correspond to robot poses where an exteroceptive information is available such as a GPS measurement or the detection of geolocalized landmark for example. Starting from the goal set and taking the successive backprojection, it is possible to find a motion plan when the initial position is totally included in a backprojection. Figure 6.25 shows a guaranteed motion plan strategy, defined as a high level strategy, to reach a goal set $\mathbb{B}$ by passing through an intermediate set $\mathbb{A}$. Since $\mathbb{A} \subset BACK(\mathbb{B}, \mu_2)$ meaning that all the poses in $\mathbb{A}$ are able to reach under a defined strategy μ_2 despite the uncertainty on the motion the goal set $\mathbb{B}$. A blue is then created between $\mathbb{A}$ and $\mathbb{B}$ as shown in Figure 6.25(b). Then taking the backprojection of the intermediate set $\mathbb{A}$ under another defined strategy μ_1 terminates the graph construction since the initial pose $\mathbb{X}_0$ (starting pose) of the AUV is totally included in this backprojection ($\mathbb{X}_0 \subset BACK(\mathbb{A}, \mu_1)$). A link is again created between the initial pose and the set $\mathbb{A}$. The final graph is depicted in Figure 6.25(b). This graph provides the high level plan that could be described as follows. From the initial pose of the robot, follow the strategy μ_1 until it detects through exteroceptive measurements or other means that the AUV has reached the set $\mathbb{A}$. When it is inside $\mathbb{A}$, follow the strategy μ_2 to reach the goal set $\mathbb{B}$ which will be detected again based on exteroceptive measurements or other means. The detection may be based on optical images, sonar images, GPS availability, etc...



(a) Preimage of $\mathbb{B}$ under strategy μ_2 and preimage of $\mathbb{A}$ under strategy μ_1



(b) Links creation.

Figure 6.25: Graph construction with preimages.

In most of the cases, the intermediate set $\mathbb{A}$ is not totally included in the backward reach set, or the backprojection, of the set $\mathbb{B}$. This is illustrated in Figure 6.26. Only a reduced set,

defined as $\mathbb{A}_0 \subset \mathbb{A}$, of robot poses inside the intermediate set $\mathbb{A}$ are able to reach for sure the goal set $\mathbb{B}$. This reduced set is defined as follows:

$$\mathbb{A}_0 = \mathbb{A} \cap BACK(\mathbb{B}, \mu_2) \tag{6.195}$$

where a strategy μ_2 has been defined. A blue link is then created between $\mathbb{A}_0$ and $\mathbb{B}$.

Since an exteroceptive information is available in $\mathbb{A}$, the robot is able to localize itself inside $\mathbb{A}$. Under the assumption that the robot can join the set $\mathbb{A}_0$ from anywhere in $\mathbb{A}$, $\mathbb{A}_0$ is then linked directly to $\mathbb{A}$. This process can be called auto-relocation. It could be defined as follows:

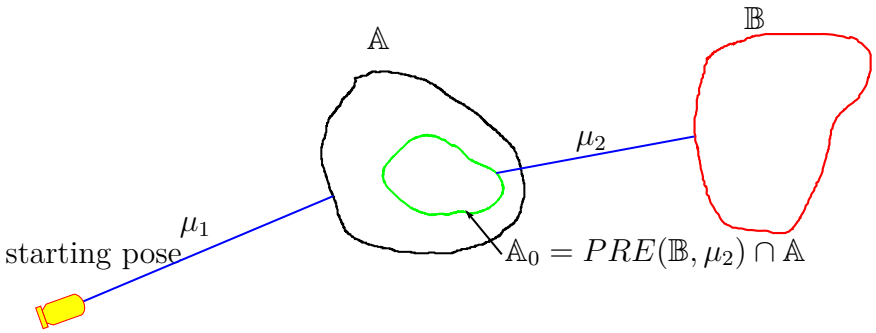
$$\forall \mathbf{a} \in \mathbb{A}, \exists \mu_{\mathbb{A}}, \mathbf{f}_{\mu_{\mathbb{A}}}(\mathbf{a}) \in \mathbb{A}_0 \tag{6.196}$$

where $\mathbf{f}_{\mu_{\mathbb{A}}}$ is a motion function depending on a strategy $\mu_{\mathbb{A}}$. This strategy $\mu_{\mathbb{A}}$ may depend on the pose $\mathbf{a}$ in $\mathbb{A}$. It can be said that there exists some strategies μ_a such that $\mathbb{A} = BACK(\mathbb{A}_0, \mu_{\mathbb{A}})$. This relocation process may be visually guided motion as in [287] where the automatic landing of an unmanned aerial vehicle (UAV) is based on the detection of the platform where it should land and adapt its behaviour to land correctly. It is then based on the estimated pose of the robot relatively to the landmark according to the exteroceptive measurements. It is called visually guided navigation or visual servoing as in [165]. In [284], it finds the shortest path under non-holonomic constraints for a vehicle while maintaining an object in sight of camera with a limited Field Of View (FOV). The visual guided motion is more and more used nowadays in autonomous driving by detecting obstacles and the lines of the road for example to adapt its motion relatively to these elements. If the landmark is geolocalized, it can be located in the reference frame.

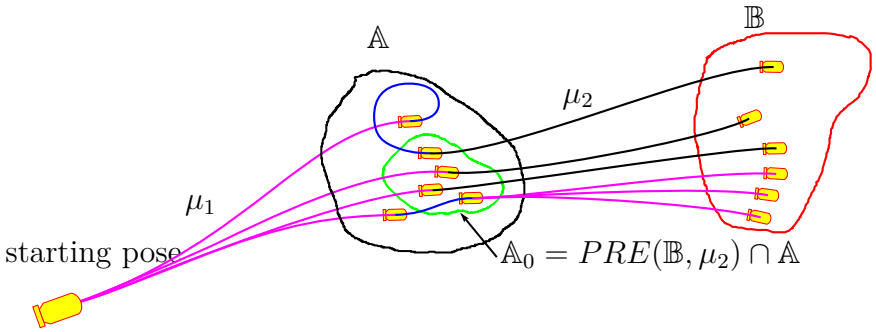
If the initial pose belongs to the backprojection of the set $\mathbb{A}$ under strategy μ_1 as depicted by the blue link in Figure 6.26(a), then the complete graph can be built in Figure 6.26(c) where $\mathbb{A}$ is linked directly to $\mathbb{B}$ since it is possible to join $\mathbb{A}_0$ from $\mathbb{A}$ anywhere in $\mathbb{A}$. The high level strategy can be described as follows:

- Follow the strategy μ_1 from the initial pose until it reaches $\mathbb{A}$.
- If the pose is already in $\mathbb{A}_0$, wait for the next motion command, otherwise execute the auto-relocation to join $\mathbb{A}_0$.
- Follow the strategy μ_2 to reach the goal set $\mathbb{B}$.

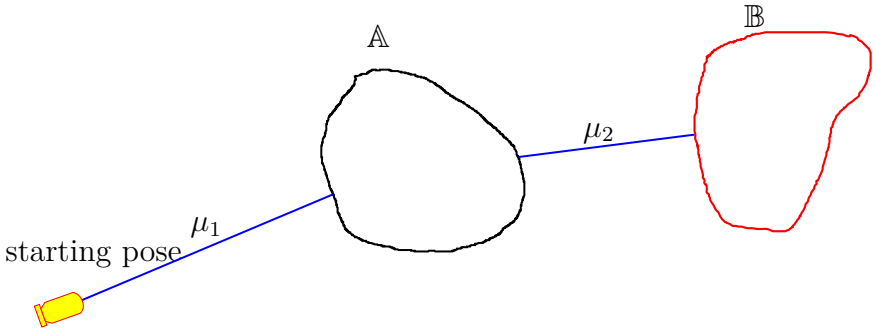
The execution of this strategy is illustrated in Figure 6.26(b). From the initial pose of the AUV, the robot follows the strategy μ_1 to reach for sure the set $\mathbb{A}$. Many trajectories depicted in magenta are possible due to the drift from the nominal trajectory. Some AUV poses end directly in $\mathbb{A}_0$ represented in green. But others poses need an auto-relocation to join $\mathbb{A}_0$, this is represented by blue trajectories using a visually guided motion for example. The motion constraints of the vehicle have to be handled carefully. When the pose of the AUV is in $\mathbb{A}_0$, then the robot follows the strategy μ_2 to reach the goal set $\mathbb{B}$. Again many trajectories are possible, but it has been proved that all the possible trajectories will reach at a certain time the goal set $\mathbb{B}$ by starting from $\mathbb{A}_0$.



(a) Graph with links and associated strategies.



(b) Path execution. Magenta paths depict the potential paths due to the drift of the robot at different poses. Some paths already end in $\mathbb{A}_0$ and two others need a relocation represented by the blue paths.



(c) Simplified graph.

Figure 6.26: Relocation inside the set $\mathbb{A}$ to reach $\mathbb{A}_0 = PRE(\mathbb{B}, \mu_2) \cap \mathbb{A}$. It is assumed the existence of μ_A such that $\mathbb{A} = PRE(\mathbb{A}_0, \mu_{\mathbb{A}})$. $\mu_{\mathbb{A}}$ may depend on the pose in $\mathbb{A}$.

It is possible that μ_2 is composed of several strategies that depend on the pose of the AUV in $\mathbb{A}_0$. But it is proved that at least one strategy exists according to the pose (possibly uncertain) that guarantees the reachability of $\mathbb{B}$.

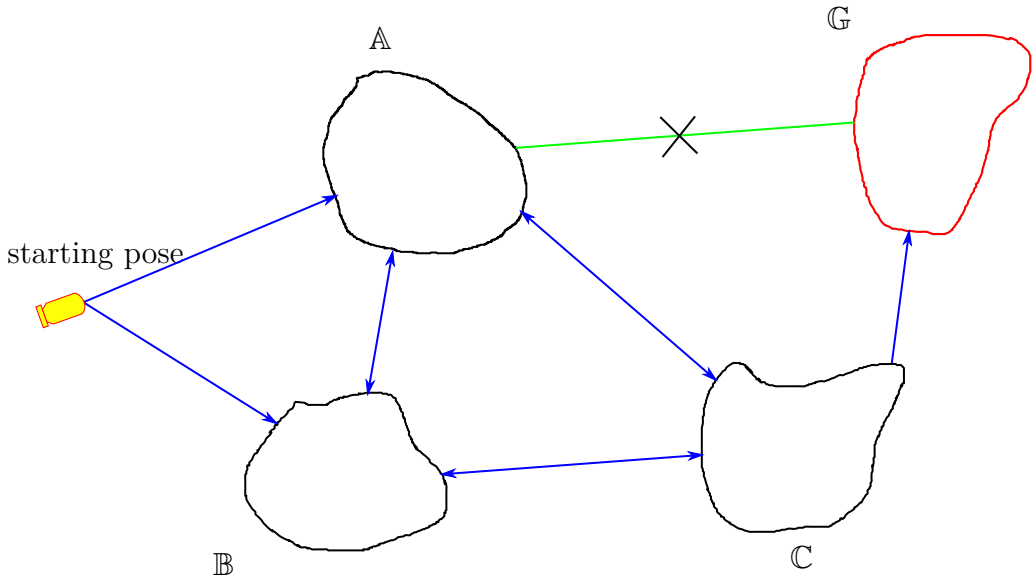
The name $\mathbb{A}_0$ was given as the first level of relocation process in $\mathbb{A}$. Due to possibly strong non-holonomic constraints, the AUV needs several levels of relocation before having a larger set $\mathbb{A}_N$ ($\mathbb{A}_N \subseteq \mathbb{A}$) that is reachable from another $\mathbb{C}$ for example by taking the backprojection of $\mathbb{A}_N$ onto $\mathbb{C}$.

Based on the existence of some strategies μ_i that links the different sets, a complete graph can be built where a simplified version is provided in Figure 6.27(a) to join a goal set defined by $\mathbb{G}$. Based on auto-relocation or direct reachability, the sets are connected as shown by blue links. It is called an hyper-graph. The different sets are the nodes of the graph and the links correspond to the edges.

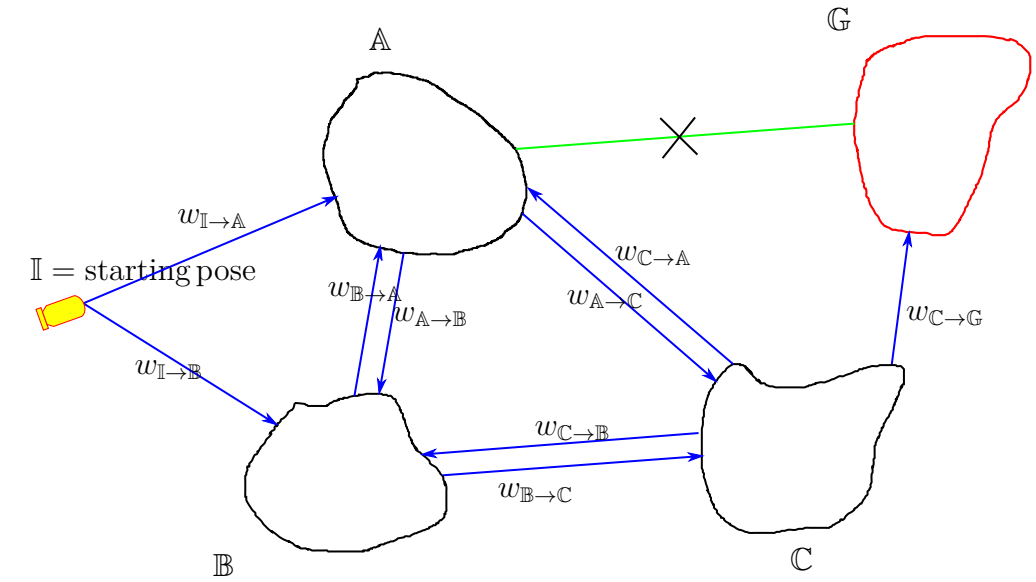
In Figure 6.27 the green link is not possible, meaning that it is not possible to leave $\mathbb{A}$ and

reach $\mathbb{G}$ for sure. Different weights linking the sets can be computed based on the distance between the landmarks for example (or the centroids). A graph search such as the Dijkstra or the A* algorithm presented in Chapter 3 will finally provide the high level strategy that optimizes a cost function.

The complete graph is not necessary to find a path. The strategy in this thesis as in [186] is to start from the goal set, and then to perform backprojections until the initial pose is totally included in a backprojection (or preimage). The graph is therefore built in a backward manner and it is possible that no path exists. It will return failure in this case. It can then be said that it is a complete motion planner, however the computation of the backprojection based on interval analysis depends on an epsilon (ϵ) parameter (precision). It is consequently almost a complete planner. At the beginning, only the nodes (sets) are present and no link exist between the nodes. The links are built iteratively based on different backprojection. An optimized path can be obtained by using a Dijkstra or an A* algorithm according to a cost function.



(a) Graph with links. Blue arrows depict the different reachabilities. The green link is not possible.



(b) Graph with weighted links.

Figure 6.27: Graph search based on weights.

The definition of the different steps of the motion planner proposed in this thesis has been provided. This concept will be applied using the tools developed in the different Chapters. Firstly, the graph building and search will be performed on defined shapes as shown in the last section using the motion model presented. The sets that constitute the nodes of the graph will be directly defined to understand the graph creation. Secondly, the problem will be applied to the registration maps in Section 6.6 to get back to the context of this thesis and the use of sensors with limited Field Of View (FOV).

6.5.2 Graph building with directional backprojection

The concept of the motion planning will be presented using the directional (omnidirectional) backprojections and the motion model in Equation 6.14 or the 2D model in Equation 6.15 when the heading has no influence for the reachability. This is similarly as the concept on the

preimages presented in [186][187] where the problem is solved here with interval analysis and enables to propose random shape relocation areas and/or obstacles contrary to the work in [186] which was reduced only to disks.

The theory developed in Section 6.4.2 will be applied here to build the graph, so the reader may refer to it to understand the tools that will be used.

Assumptions

- The problem will be solved in the 2D plane.
- Similarly as in [186], it is assumed that the robot is able to navigate certainly in a relocation area and does not have any constraint on its motion meaning that it has an holonomic behaviour. The motion control is then accurate.
- It is assumed that the robot is equipped with a compass that provides the heading information. However this information is not certainly known but lies within a bounded uncertainty interval that is defined by α_θ.
- Moreover, it will be assumed that the AUV moves at a speed $v_d = 1m/s$ which is close to the speed of most AUVs nowadays. However an uncertainty on the speed is present and is handled by α_v and is also represented by a bounded uncertainty interval.
- It is assumed that the robot is equipped with some exteroceptive sensors indicating it is inside a relocation area (detect a landmark for example). However it is necessarily able to indicate in which relocation area it is.

Since the uncertain parameters lie in bounded intervals, the set membership context is well suited to solve the problem.

The uncertain parameter vector $\mathbf{p}$ of the motion model in Equation 6.14 is then described by $(v, \delta\theta)$ where $\mathbf{p} \in [\mathbf{p}] = [v_d(1-\alpha_v), v_d(1+\alpha_v)] \times [-\alpha_\theta, \alpha_\theta]$ as presented in Section 6.4.2 on the work of the directional backprojection with sets.

It will be assumed only relocation areas in $\mathbb{R}^2$ meaning that the heading of arrival has no influence. The 2D motion model $\mathbf{f}_{H2}$ given in Equation 6.15 will then be used but will be denoted as $\mathbf{f}_H$. This model is proposed to illustrate simply the graph building and the computation of the backprojection as proposed in the previous section.

Consider a map $\mathbb{M}$ composed by a set of relocation areas $\mathbb{A}_i$ in $\mathbb{R}^2$ such that:

$$\mathbb{M} = \{\mathbb{A}_1, \mathbb{A}_2, ..., \mathbb{A}_n\} \tag{6.197}$$

If the intersection between some relocation areas $\mathbb{A}_i$ is not empty, a cluster is created. The biggest cluster is obtained iteratively by adding to an actual cluster any relocation areas $\mathbb{A}_i$ that intersects this actual cluster. Several clusters can be present in the map $\mathbb{M}$.

Finally the map $\mathbb{M}$ is composed of a set of relocation areas and clusters of relocation areas. The map $\mathbb{M}$ is therefore firstly filtered to find and create clusters before applying any backprojection. This comes from the fact that the backprojection can be computed directly when the intersection between sets is not empty. For ease of understanding, the clusters will be called relocation areas too.

Based on these relocation areas, two different motion planner algorithms will be proposed. Firstly, it will be assumed that the different relocation areas (sets) are distinguishable. Secondly, the problem will be solved under the assumption that the relocation areas are indistinguishable. The motion planners will be illustrated through examples. The difference is depicted in Figure 6.28 and will be more explained in the following of the document.

The main difference can be expressed as follows. The 2D backward reachability of two disconnected relocation areas $\mathbb{A}$ and $\mathbb{B}$ shown by green circles at the direction $\theta_d = 60°$ is depicted in Figure 6.28. It has been computed according to Equation 6.77 since $\theta_d \in \Theta_{\mathbb{AB}}$. From the initial robot pose represented by the red AUV, belonging to the 2D backward reach set with the heading oriented at 60°, some trajectories are depicted in blue with random values in the interval of the direction error $[\delta\theta] = [-5, 5](deg)$. As it can be noticed some trajectories ended in the relocation area $\mathbb{A}$ and some in $\mathbb{B}$. Revisiting a cluster of disconnected sets as proposed by this strategy is possible for the motion planner considering distinguishable relocation areas since a sensor will indicate in which relocation area it is and/or depending on the nature of the landmark. However this strategy wouldn't be possible when considering indistinguishable relocation areas due to the ambiguity of the robot location when reaching the relocation areas. Strategies combining disconnected relocation areas will then be possible only for the first motion planner.

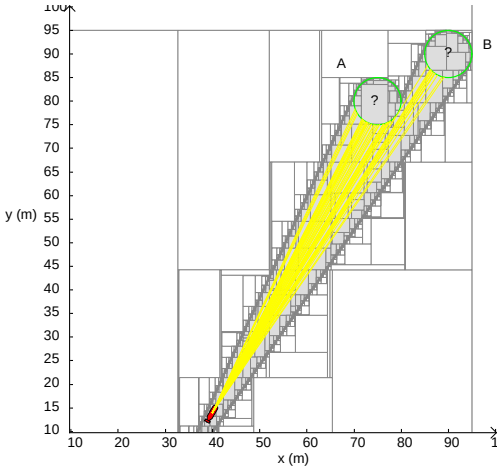


Figure 6.28: Difference between motion planners considering distinguishable or not relocation areas.

6.5.2.1 Distinguishable relocation areas

If the robot has reached a relocation area, it is able to indicate in which relocation area it is among all the relocation areas. This is particularly the case when the strategy is to reach a cluster of disconnected relocation areas as introduced in Figure 6.28 and presented in Figure 6.30(d). This assumption could be true in reality if all the landmarks are different from each other and no ambiguity is present.

The uncertain parameters are defined at $\alpha_\theta = 5°$ and $\alpha_v = 10\%$.
Remind that the uncertainty on the speed had no influence in the backward reach set but was used to compute the goal recognizability.

Certainly located in relocation areas

Consider the robot is certainly located in a relocation area, meaning that the true position is available. As assumed, the AUV can move anywhere in a relocation area without any constraint. The robot starts from an initial configuration $\mathbf{x}_0$. This initial configuration may be uncertain and then located inside a set $\mathbf{x}_0 \in \mathbb{X}_0$. The environment $\mathbb{M}$ is composed by a set of disconnected relocation areas as explained before. A preprocessing step may have created clusters if some were connected. A goal set $\mathbb{G}$ is defined. If a relocation area is connected to the goal set $\mathbb{G}$, then this relocation area is added to the goal set, and the set $\mathbb{G}$ is replaced by the union of the sets. The motion planner is proposed in Algorithm 9 where "PP" corresponds to "Path Planner".

Algorithm 9 Distinguish-PP (**in** : $\mathbb{X}_0$, $\mathbb{G}$, $\mathbb{M}$, **out** : path)

$OPEN \leftarrow \{\mathbb{G}\}$
$CLOSED \leftarrow \emptyset$
$end \leftarrow 0$
$pathFound \leftarrow 0$
$graph.init()$
$graph.createVertex(\mathbb{G})$
while $end \neq 1$ **do**
    $\mathbb{U} \leftarrow selectNode(OPEN)$
    $OPEN \leftarrow OPEN \backslash \{\mathbb{U}\}$
    $CLOSED \leftarrow CLOSED \cup \{\mathbb{U}\}$
    if $connect(\mathbb{X}_0, \mathbb{U})$ **then**
        $graph.createVertex(\mathbb{X}_0)$
        $graph.createEdge(\mathbb{X}_0, \mathbb{U})$
        $end \leftarrow 1$
        $pathFound \leftarrow 1$
    else
        for all $\mathbb{V} \in$ neighbourhood($\mathbb{U}$) in $\mathbb{M}$ and $\mathbb{V} \neq \mathbb{U}$ **do**
            if $\mathbb{V} \in CLOSED$ **then**
                continue
            if $\mathbb{X}_{\mathbb{V} \to \mathbb{U}} \neq \emptyset$ **then**
                if $\mathbb{V} \notin OPEN$ **then**
                    $OPEN \leftarrow OPEN \cup \{\mathbb{V}\}$
                    $graph.createVertex(\mathbb{V})$
                    $graph.createEdge(\mathbb{V}, \mathbb{U})$
                if $\{\mathbb{V}, \mathbb{U}\} \notin OPEN$ and $\{\mathbb{V}, \mathbb{U}\} \notin CLOSED$ **then**
                    $OPEN \leftarrow OPEN \cup \{\mathbb{V}, \mathbb{U}\}$
                    $graph.createVertex(\{\mathbb{V}, \mathbb{U}\})$
                    $graph.createEdge(\{\mathbb{V}, \mathbb{U}\}, \mathbb{U})$
    if $OPEN = \emptyset$ **then**
        $end \leftarrow 1$
if $pathFound = 1$ **then**
    $path \leftarrow reconstructPath(graph, \mathbb{X}_0, \mathbb{G})$
else
    $path \leftarrow \emptyset$

The explanation of the motion planner will be proposed through an example.

Consider a simple environment composed of disks as depicted in Figure 6.29(a) where names are given to the different relocation areas. The blue AUV, denoted as I in the bottom left corner,

indicates the initial pose and the red circle is the goal area $\mathbb{G}$. At the beginning, the graph is only composed by a single node $\mathbb{G}$. Moreover only the goal set $\mathbb{G}$ is in the $OPEN$ list, then the set $\mathbb{U} = \mathbb{G}$. It firstly looks (line 11) if a connection can be made with the initial pose based on the termination condition:

$$TC(\mathbf{q}_0) = \{\theta_d \in \mathbb{R} | \forall \mathbf{p} \in [\mathbf{p}], \exists t, \mathbf{f}_H(\mathbf{x}_0, t, \mathbf{p}) \in \mathbb{U}\} \tag{6.198}$$

where $\mathbf{x}_0 = (\mathbf{q}_0, \theta_d) = (x_0, y_0, \theta_d)$ when the initial pose is certainly known. If the position is uncertainly known, the termination condition corresponds to:

$$TC(\mathbb{Q}_0)^- = \{\theta_d \in \mathbb{R} | \forall \mathbf{q}_0 \in \mathbb{Q}_0, \forall \mathbf{p} \in [\mathbf{p}], \exists t, \mathbf{f}_H(\mathbf{x}_0, t, \mathbf{p}) \in \mathbb{U}\} \tag{6.199}$$

where a method based on the Minkowsky difference was proposed in the previous section.

If $TC(\mathbf{q}_0)$ (in the certain initial position case) or $TC(\mathbb{Q}_0)^-$(in the uncertain initial position case) is not empty then a connection can be made and the algorithm terminates. The computation is based on the SIVIA algorithm as used all along this thesis.

If the termination condition is empty, it looks for connection with the other relocation areas in the map $\mathbb{M}$ (line 17 and 18). These connections are computed based on Equation 6.55. For example, consider the set $\mathbb{C}$. If $\mathbb{X}_{\mathbb{C}\to\mathbb{G}} = \mathbb{X}_{\mathbb{C}\mathbb{G}}$ is not empty (line 18) then a vertex $\mathbb{C}$ (node) is created (line 21) in the graph and an edge between $\mathbb{C}$ and $\mathbb{G}$ is also created (line 22) if the considered set $\mathbb{C}$ is neither in the $OPEN$ nor in the $CLOSED$ list. Moreover it is added to the $OPEN$ list. Since complex strategies involving disconnected sets as explained in Figure 6.28 on the distinguishable property, the global set $\mathbb{C} \to \mathbb{G} = \mathbb{C}\mathbb{G}$ is also added to the graph (line 25) with a connection to $\mathbb{G}$ (line 26), and added to the $OPEN$ list. If the $OPEN$ list is empty, the algorithm terminates informing that no path exists.

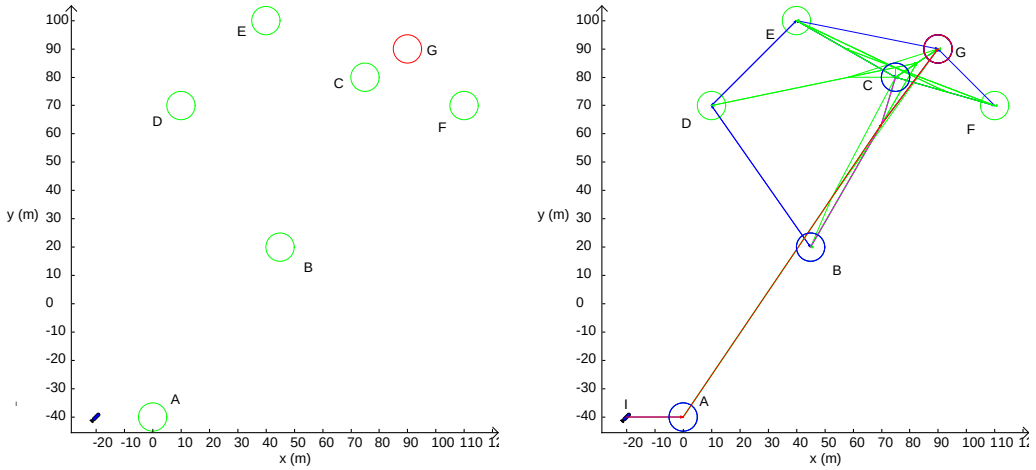
The selection of the tested node (line 8) in the $OPEN$ list can be made according to some strategies such as the closest one to the initial configuration for example, or by considering only single node at first.

If the tested node $\mathbb{U}$ is for example $\mathbb{C}\mathbb{G}$ and the possible neighbour $\mathbb{B}$, then it has to be proved that $\mathbb{X}_{\mathbb{B}\to(\mathbb{C}\mathbb{G})} \neq \emptyset$ to create a connection between $\mathbb{B}$ and $\mathbb{C}$ or $\mathbb{G}$. This will be more explained in the following when describing the strategy found by the algorithm.

When a path is found, a simple graph search in the created graph can rebuild the path.

The complete graph built is proposed in Figure 6.29(b) where the blue links are direct links between two disconnected relocation areas and the green links are high level links that connect at least three relocation areas. In this latter case, the centroid of the involved relocation areas is used as reference and the green lines are drawn from the centroid to each center (centroid) of the different relocation areas.

The path found in a backward manner is depicted in red. As it can be noticed (a little bit difficult to see), the path is composed of high level strategies. Due to the difficulty to understand this graph, the different strategies are detailed just below.



(a) Environment with disk relocation areas in green and goal area in red. The initial pose is represented by the blue AUV.

(b) Graph built with the path found in red. Blue links are direct links between two relocation areas, and green links are high level links between a set of disconnected relocation areas (at least 3).

Figure 6.29: Environment and high level strategy found in a backward manner.

The first strategy μ_1, as it can be seen in the graph created in Figure 6.29(b), is to join the relocation area $\mathbb{A}$. From the certainly initial pose $\mathbb{I}$ of the robot at the location $\mathbf{q}_0 = (x_0, y_0)$, the set of directions $\Theta_{\mathbb{I}\to\mathbb{A}} = \Theta_{\mathbb{I}\mathbb{A}}$ is computed according to Equation 6.198 where $\mathbb{U} = \mathbb{I}$. Since it corresponds to the termination condition according to the set $\mathbb{A}$, it could be denoted as $TC_{\mathbb{A}}(\mathbf{q}_0)$. If the initial location was uncertainly known, the set of direction is computed according to Equation 6.199.

Computing the set of directions $\Theta_{\mathbb{I}\mathbb{A}} = TC_{\mathbb{A}}(\mathbf{q}_0)$ is similar to Equation 6.137 that was provided in the goal recognizability subsection, but here only the set of directions is important and not the time, since it is assumed that the robot is able to indicate that it has reached the relocation area with some exteroceptive sensors.

Due to the assumption of a connected set $\mathbb{A}$, the problem was reduced to a simple intersection at the bounds of the error on the direction. The set can be computed as follows:

$$\Theta_{\mathbb{I}\mathbb{A}} = \{\theta_d | \forall \delta_\theta \in \{-\alpha_\theta, \alpha_\theta\}, \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \quad (6.200)$$

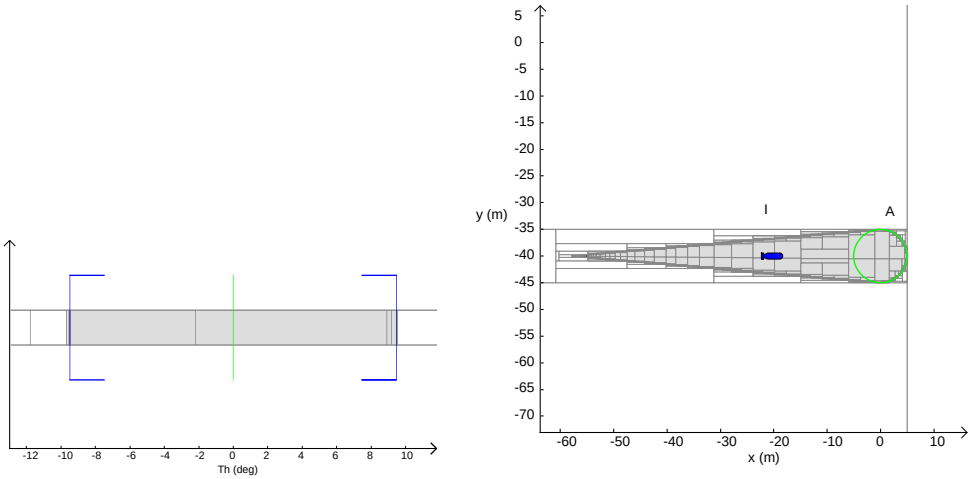
$$= \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} \{\theta_d | \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \quad (6.201)$$

$$= \bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj^{\mathbb{T}}_{\mathbb{X}_3}(\mathbf{f}_H^{-1}(\mathbb{A})) \quad (6.202)$$

where $\mathbf{x} = (x_0, y_0, \theta_d)$ in the function $\mathbf{f}_H$ and $\mathbb{X}_3$ is the initial domain of θ_d.

The set of directions to join $\mathbb{A}$ is given in Figure 6.30(a) where the initial domain was $\mathbb{X}_3 = [-180, 180](deg)$. Since disks are regular shapes and due to the conic shape of the backprojection, the set of directions can be enclosed by a blue interval that depicts the inner approximation. One direction $\theta_d = 0°$ has been selected in green in the solution and the corresponding 2D backward reach set has been provided in Figure 6.30(b). Notice that the initial location is indeed in the 2D backward reach set at this defined direction. Therefore, the robot can reach for sure the relocation area $\mathbb{A}$ from this initial location by following for example the direction $\theta_d = 0°$. At the beginning of the mission, it requires the robot to turn on itself

and to orient in this direction with the measured direction provided by the compass, which is uncertain. This ends the first strategy μ_1 to reach $\mathbb{A}$ from the initial position.



(a) Solution $\Theta_{\mathbb{IA}}$.

(b) 2D backward reach set at the direction $\theta_d = 0°$ defined at the green line in the figure (a).

Figure 6.30: Strategy μ_1 to reach the first relocation area $\mathbb{A}$.

The second strategy μ_2 as depicted in Figure 6.29(b) consists in a high level link, meaning that at least three disconnected relocation areas are involved. The idea is to reach in a guaranteed manner a set of disconnected relocation areas. As it can be noticed the goal area is included in this set. The steps describing this high level link will be described just below.

When the graph was built backwardly from the goal area $\mathbb{G}$, it firstly looks for a link with other sets based on Equation 6.55 and in particular with the set $\mathbb{C}$:

$$\mathbb{X}_{\mathbb{CG}} = \{\mathbf{x} \in \mathbb{R}^3 | BACK_{od}(\mathbb{G}) \cap \mathbb{C} \neq \emptyset\} \tag{6.203}$$

The problem was proposed only along the dimension of directions due to the 3D representation which can be written as:

$$\Theta_{\mathbb{CG}} = \{\theta_d | BACK(\mathbb{G}, \theta_d) \cap \mathbb{C} \neq \emptyset\} \tag{6.204}$$

since $\Theta_{\mathbb{CG}}$ is linked to $\mathbb{X}_{\mathbb{CG}}$ according to Equation 6.69, proving that $\mathbb{X}_{\mathbb{CG}}$ is not empty is equivalent to prove that $\Theta_{\mathbb{CG}}$ is not empty.

This set is represented in Figure 6.31(a) which is not empty. The two relocation areas $\mathbb{C}$ and $\mathbb{G}$ are then linked. It is possible to leave $\mathbb{C}$ and reach $\mathbb{G}$ for sure with the direction indicated at the green line $\theta_d = 40°$ for example. The 2D backward reach set ($BACK(\mathbb{G}, 40°)$) is presented in Figure 6.31(b). If the AUV is located at the position shown the blue AUV in $\mathbb{C}$ it can reach for sure $\mathbb{G}$ with this direction.

According to the graph building, the two sets $\mathbb{C}$ and $\mathbb{G}$ are then expanded where the computation considering the two sets at the same time, in terms of union, as in Equation 6.77 can be performed. This computation is true only if $\theta_d \in \Theta_{\mathbb{CG}}$. The graph expansion then tried to connect the set $\mathbb{B}$ by proving that the set expressed in Equation 6.96 and reformulated

equivalently in Equation 6.97 is not empty. This is reminded here with the right notation:

$$\begin{aligned}\Theta_{\mathbb{B}\rightarrow(\mathbb{C}\mathbb{G})} &= \Theta_{\mathbb{B}\mathbb{C}\mathbb{G}} && (6.205)\\ &= \{\theta_d \in \Theta_{\mathbb{C}\mathbb{G}} | BACK(\mathbb{C}\cup\mathbb{G}, \theta_d) \cap \mathbb{B} \neq \emptyset\} && (6.206)\\ &= \{\theta_d \in \mathbb{R} | BACK(\mathbb{C}\cup\mathbb{G}, \theta_d) \cap \mathbb{B} \neq \emptyset\} \cap \Theta_{\mathbb{C}\mathbb{G}} && (6.207)\end{aligned}$$

This set is presented in Figure 6.31(c) and is not empty. For example, the 2D backward reach set at the direction $\theta_d = 60°$ indicated by the green line is provided in Figure 6.31(d). If the robot is located at the blue AUV position in $\mathbb{B}$, by following this direction it will reach for sure $\mathbb{C}$ or $\mathbb{G}$.

As previously, the three sets $\mathbb{B}$, $\mathbb{C}$ and $\mathbb{G}$ can then be expanded in terms of union. Consider now the relocation area $\mathbb{A}$ and the set $\Theta_{\mathbb{A}\rightarrow(\mathbb{B}\mathbb{C}\mathbb{G})}$ that could link $\mathbb{A}$ to $\mathbb{B}$, $\mathbb{C}$ and $\mathbb{G}$ expressed as follows:

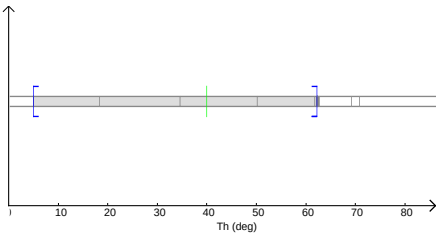
$$\begin{aligned}\Theta_{\mathbb{A}\rightarrow(\mathbb{B}\mathbb{C}\mathbb{G})} &= \Theta_{\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{G}} && (6.208)\\ &= \{\theta_d \in \Theta_{\mathbb{B}\mathbb{C}\mathbb{G}} | BACK(\mathbb{B}\cup\mathbb{C}\cup\mathbb{G}, \theta_d) \cap \mathbb{A} \neq \emptyset\} && (6.209)\\ &= \{\theta_d \in \mathbb{R} | BACK(\mathbb{B}\cup\mathbb{C}\cup\mathbb{G}, \theta_d) \cap \mathbb{A} \neq \emptyset\} \cap \Theta_{\mathbb{B}\mathbb{C}\mathbb{G}} && (6.210)\end{aligned}$$

where $\Theta_{\mathbb{B}\mathbb{C}\mathbb{G}}$ was computed in the previous equation.

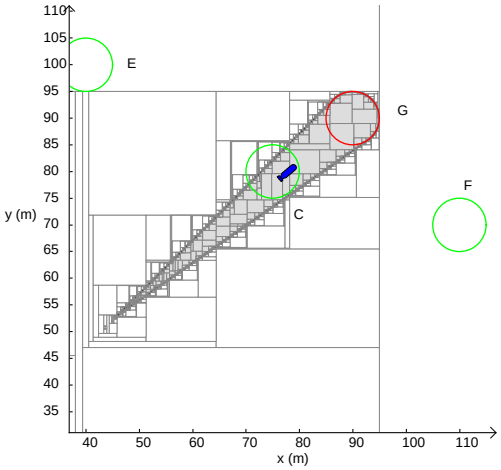
As it can be noticed, each times a set is tried to be connected to a cluster of disconnected relocation areas, it propagates the links (set of directions) that connected the cluster of disconnected relocation areas.

The result of the set $\Theta_{\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{G}}$ is provided in Figure 6.31(e) where only few directions can link the four relocation areas. This set is again not empty. The precision of the computation was 0.1° so it explains the presence of unclassified solution (dark gray). The 2D backward reach at the direction defined at $\theta_d = 55.7°$ (inside the solution set) shown by the green line is depicted in Figure 6.31(f). The same interpretation can be made as before. Since the AUV can move anywhere in $\mathbb{A}$ and is certainly located, it can go to the area of departure and orient its direction at $\theta_d = 55.7°$ to reach $\mathbb{B}$, $\mathbb{C}$ or $\mathbb{G}$. As it can be noticed, considering at the same time the three sets enables to start from $\mathbb{A}$. Considering them separately would not lead to a solution and would have returned no solution. This ends the second strategy μ_2 to reach $\mathbb{G}$ from $\mathbb{A}$ by passing possibly by intermediate relocation areas $\mathbb{B}$ or $\mathbb{C}$.

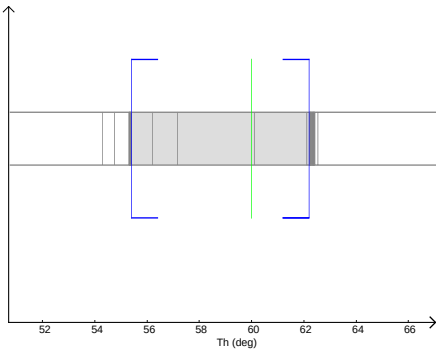
Finally the global solution with the two strategies μ_1 and μ_2 is represented in Figure 6.32 where the 2D backward reach set is proposed at the two defined direction $\theta_d = 0°$ for μ_1 and $\theta_d = 55.7°$ for μ_2. Obviously other directions in the different solution sets could have been chosen.



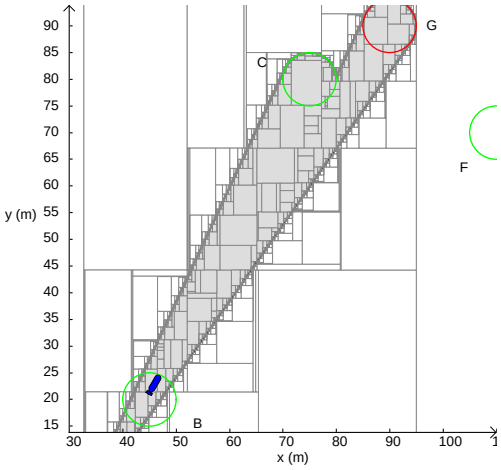
(a) Solution $\Theta_{\mathbb{C}\mathbb{G}}$.



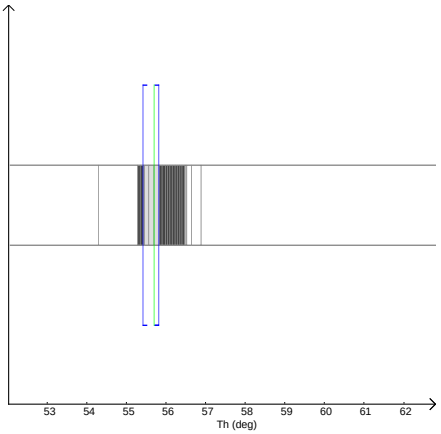
(b) 2D backward reach set at the direction $\theta_d = 40°$ defined at the green line in the figure (a).



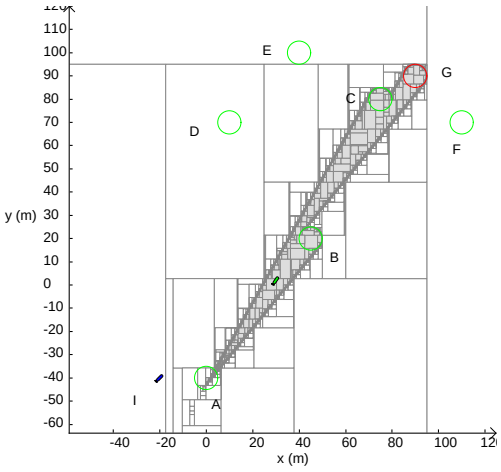
(c) Solution $\Theta_{\mathbb{B}\mathbb{C}\mathbb{G}}$.



(d) 2D backward reach set at the direction $\theta_d = 60°$ defined at the green line in the figure (c).



(e) Solution $\Theta_{\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{G}}$.



(f) 2D backward reach set at the direction $\theta_d = 55.7°$ defined at the green line in the figure (e).

Figure 6.31: Decomposition of the solution of the high level link for the strategy μ_2.

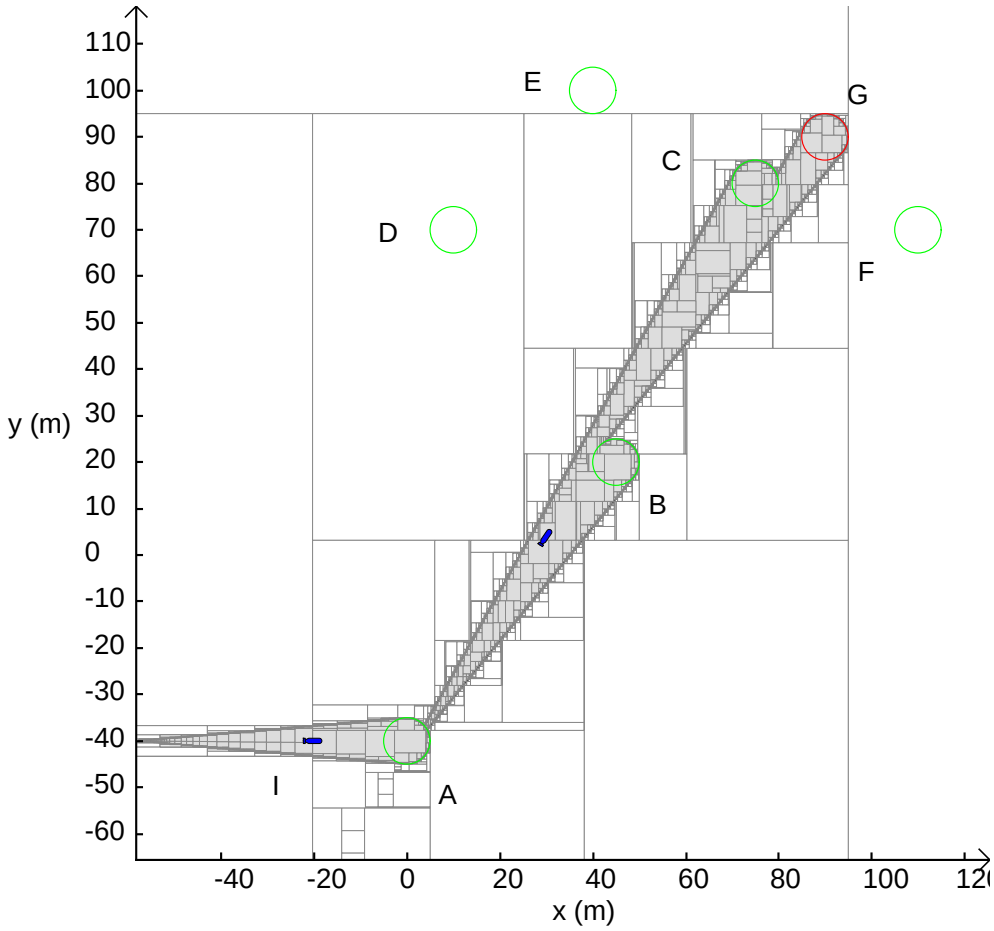


Figure 6.32: Global solution represented by the 2D backward reach set with $\theta_d = 0°$ for μ_1 and $\theta_d = 55.7°$ for μ_2.

The complete path execution is presented in Figure 6.33 where the green AUVs are the departure poses (inside the departure areas) of the different relocation areas. The magenta lines are the relocation process to join the departure poses that were choiced according to the backward reachability of the different sets. Blue lines correspond to the navigation outside the relocation areas. It is supposed that a controller is able to stabilize perfectly the direction defined which explains the representation with straight lines. The different cases of the high level strategy is described below.

When the robot executes this high level strategy $\pi = \{\mu_1, \mu_2\}$, it will firstly go to the relocation area $\mathbb{A}$ by following one direction, for example $\theta_d = 0°$ as shown in Figure 6.33. The set of directions to reach $\mathbb{A}$ from the initial position was presented in Figure 6.30. An exteroceptive sensor indicates that the robot has reached $\mathbb{A}$. The robot then goes to the departure area in $\mathbb{A}$ to reach $\mathbb{B}$, $\mathbb{C}$ or directly $\mathbb{G}$ as it has been proved and shown by the magenta lines. One departure pose is represented by a green AUV in Figure 6.33 in $\mathbb{A}$. Again it follows one direction, $\theta_d = 55.7°$ for example in Figure 6.33, that is proved to reach perhaps directly the goal area $\mathbb{G}$. However due to the uncertainty on the direction, the robot may reach $\mathbb{B}$. Since the position is accurately known, it joins the departure pose in $\mathbb{B}$, represented by the magenta lines, to go to $\mathbb{C}$ or $\mathbb{G}$ as was explained in Figure 6.31(c) and (d) with the direction $\theta_d = 60°$ (for example). No direct link exists between $\mathbb{B}$ and $\mathbb{G}$. The robot may have reached $\mathbb{C}$, then again it joins the departure pose to go directly to $\mathbb{G}$ with the direction $\theta_d = 40°$ (for example) since they are connected as depicted in Figure 6.31(a) and (b). This ends the strategy execution. The complete path execution was presented in Figure 6.33 with the different cases

of reachability and all the trajectories end in the goal area despite the uncertainty on the direction if the error remains in its bounds. The union of all the 2D backward reach sets, called $\mathbb{P}$, presented in Figure 6.31(b)(d)(f) and in Figure 6.30(b) according to the different selected direction is proposed in Figure 6.33 which can be described as follows:

$$\mathbb{P} = BACK(\mathbb{G}, 40°) \cup BACK(\mathbb{C} \cup \mathbb{G}, 60°) \cup BACK(\mathbb{B} \cup \mathbb{C} \cup \mathbb{G}, 55.7°) \cup BACK(\mathbb{A}, 0°) \quad (6.211)$$

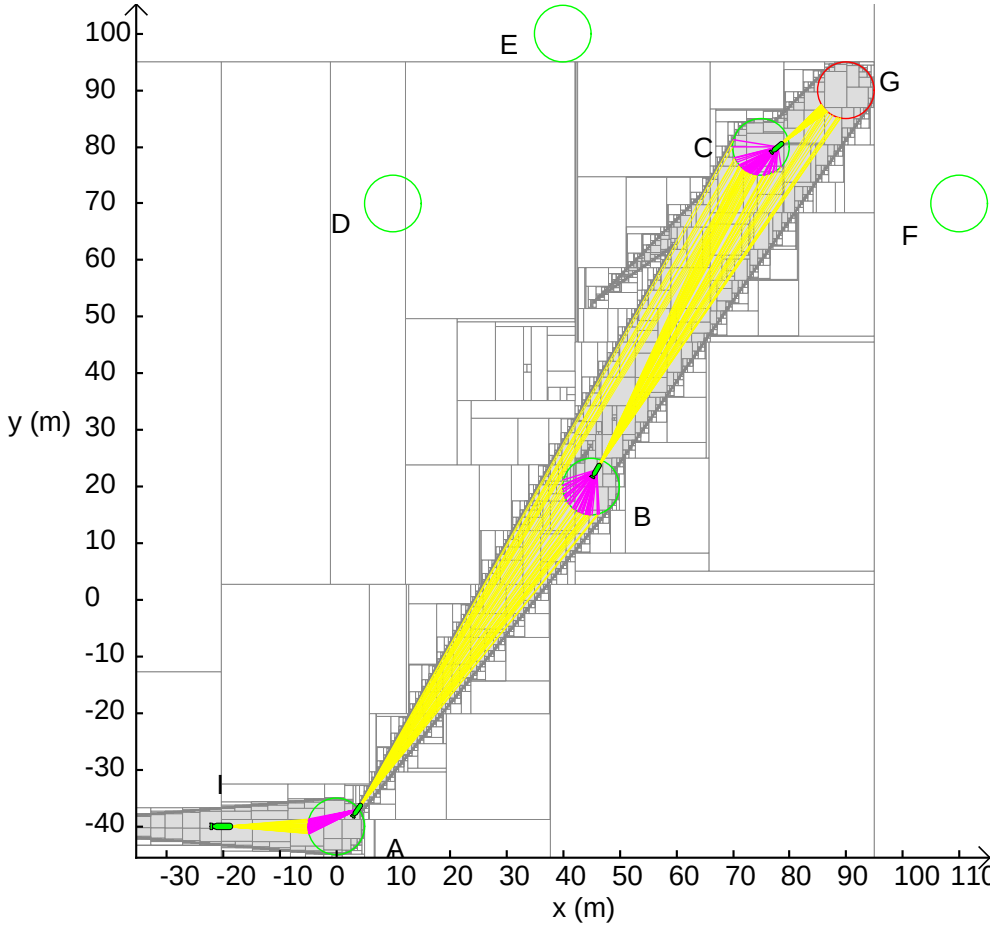


Figure 6.33: Path execution with relocation process in magenta to join the departure point in the different relocation areas. The different backward reach set according to the defined direction is also presented. Random samples are generated in the error intervals.

Conclusion

Assuming distinguishable relocation areas, the graph building is able to find a complex strategy that may connect several disconnected relocation areas to extend the possibility of exploration. The problem has been presented with a set-membership description of the relocation areas enabling the graph building with complex shapes as presented earlier. The example based on simple shapes such as disks was provided to ease the understanding. However, this method requires a real distinction between the different relocation areas defined as the uniqueness. In reality, these relocation areas may be areas between buildings by taking into account obstacles as presented in the previous section. In the underwater context, this assumption could even be possible since some landmarks may be very different such as ropes, rocks or wrecks. Uncertainty

in the relocation areas was not presented here, but will be presented in the next subsection when dealing with indistinguishable relocation areas that are more adapted to the underwater context since the difference between rocks may be difficult.

6.5.2.2 Indistinguishable relocation areas

When the robot has to reach a cluster of relocation areas, it is not able anymore to say in which it is as shown in Figure 6.28 where the robot could have joined $\mathbb{A}$ or $\mathbb{B}$. Therefore the strategy of joining a cluster of disconnected relocation areas as presented in the previous example is not anymore possible. The graph building can then expand only isolated relocation areas as shown from line 19 to 22 in Algorithm 9. The global expansion from line 23 to 26 are prohibited. Remind that relocation areas could have been formed by the connected union of some relocation areas, defined as clusters. When the robot has to reach a cluster of relocation areas (connected union), a strategy has to be established in order to remove the ambiguity. Therefore, reaching a cluster of connected relocation areas is still possible and particularly interesting to travel longer distance. Moreover when the AUV starts from a relocation area $\mathbb{A}$ to reach another relocation area $\mathbb{B}$ where the link has been proved, if some trajectories due to the uncertain direction may intersect another relocation area $\mathbb{C}$, it can lead to a wrong data association and then a wrong relocation process. The robot would think it has reached $\mathbb{B}$ but actually it is inside $\mathbb{C}$. Consequently, Algorithm 9 could then be used by removing the lines from 23 to 26 but it requires some changes to take into account the possible wrong data association with another relocation areas.

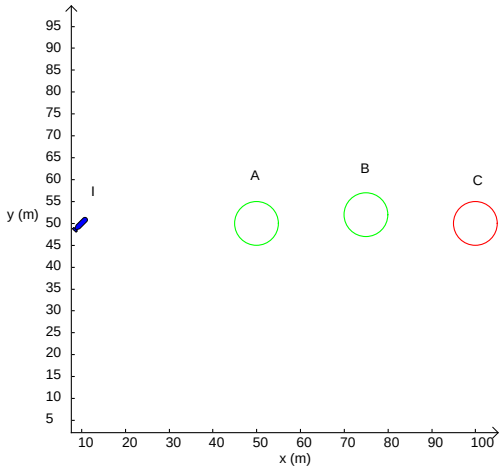
In this part, it will be explained how to connect the initial position of the robot that may be uncertain, and how to connect two relocation areas to avoid possible ambiguities when the robot is certainly and uncertainly located in the departure relocation area. It will be notified when the different cases are explained. Obstacles are also added to the environment as $\mathbb{O}$ and have to be avoided also.

If the robot is certainly located in a relocation area, considering the environment in Figure 6.29(a), the new algorithm that will be presented in Algorithm 10 returns failure since it is not possible to start from $\mathbb{B}$ and reach $\mathbb{C}$.

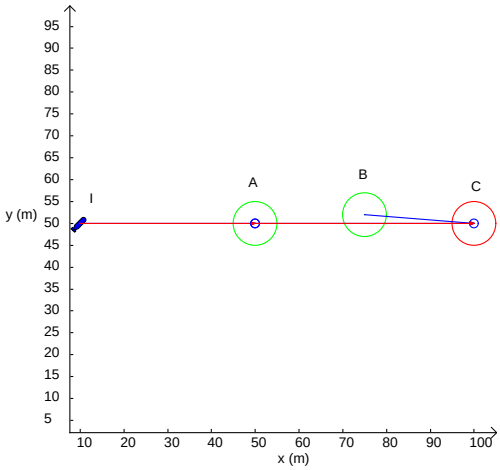
Consider the same uncertain parameters $\alpha_\theta = 5^\circ$ and $\alpha_v = 10\%$.

Consider the simple environment in Figure 6.34(a) where a goal area $\mathbb{G}$ has to be reached. Two relocation areas $\mathbb{A}$ and $\mathbb{B}$ are present in the map $\mathbb{M}$. The initial position of the robot is presented by a blue AUV on the left. Using Algorithm 9 by removing the lines 23 to 26 (global expansion) leads to the graph built in Figure 6.34(b). A path has been found and is presented in red. The global 2D backward reach set of the possible direction solution is presented Figure 6.34(c). As it can be noticed, the 2D backward reach passes through the relocation area $\mathbb{B}$ since it ignored it when it built the graph. Due to the indistinguishability property of the relocation areas, when executing this strategy, the robot would think it has reached $\mathbb{G}$ but it is actually in $\mathbb{B}$. This is unacceptable. The set $\mathbb{B}$ has then to be taken into account when testing a connection between $\mathbb{A}$ and $\mathbb{G}$. Moreover, even if the set $\mathbb{B}$ was considered as an obstacle, it would be impossible to leave $\mathbb{A}$ and reach $\mathbb{G}$ without going through $\mathbb{B}$ at a certain time. Algorithm 9 has then to be modified to take into account other relocation areas to avoid any wrong data association as represented in the graph built in Figure 6.34(d) where the 2D backward reach set of the path found is depicted in Figure 6.34(e). Notice that the new solution is now to pass by

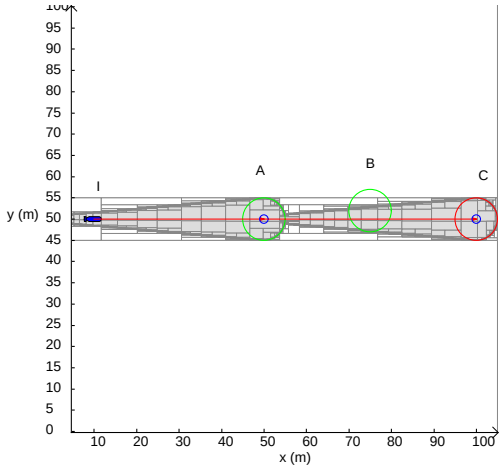
$\mathbb{B}$ before reaching $\mathbb{G}$. The new algorithm is proposed in Algorithm 10 where "PP" corresponds to "Path Planner".



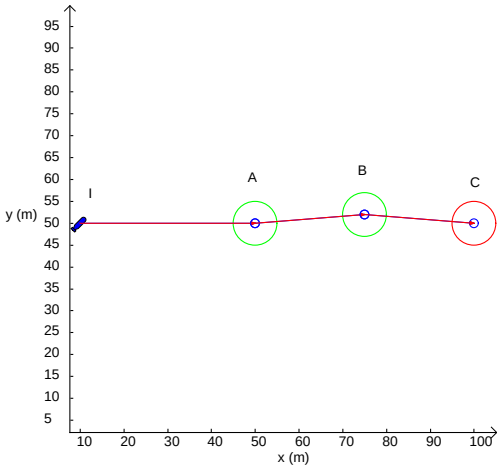
(a) Simple environment with a disk goal area $\mathbb{G}$.



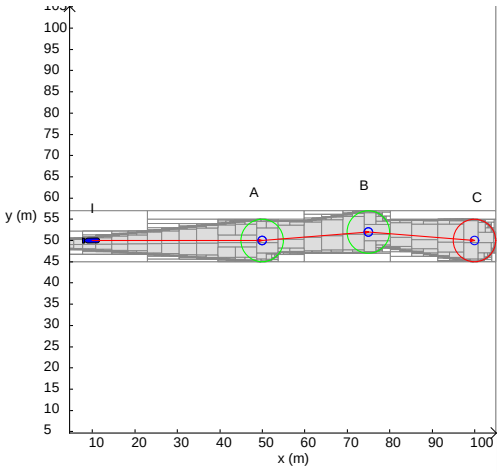
(b) Graph built with Algorithm 9 and the removed lines.



(c) 2D backward reach set of the path found. The direction are defined according to the different reachabilities. The solution goes through $\mathbb{B}$ leading to a wrong data association with $\mathbb{G}$.



(d) New graph built taken into account other relocation areas.



(e) 2D backward reach set of the path found. No wrong data association.

Figure 6.34: Difference with Algorithm 9 and the new algorithm taking into account the other relocation areas.

Algorithm 10 has been named "Indistinguish-PP" for motion planning with relocation areas that are indistinguishable and possible obstacles mentionned by the set $\mathbb{O}$ which to the union of the possible obstacles $\mathbb{O}_i$. Algorithm 10 is similar to Algorithm 9 but now a cost function has been added to find an optimal path with an A* search [124] or Dijkstra search [77] when the heuristic h is null. At the beginning of the algorithm, only the goal relocation area is present and is added to the $OPEN$ list (line 1). Since no complex strategies involving disconnected relocation areas in $\mathbb{M}$ are possible now due to the indistiguishability property, all the relocation areas in $\mathbb{M}$ are nodes of the hyper-graph which is then initialized in line 5. Only the edges are not present at the beginning. The nodes $\mathbb{U}$ are selected according to a function f (line 12) that takes into account a possible heuristic to find an optimal path by finding here the minimum path according to the cost function. When a node $\mathbb{U}$ is selected, a variable *AmbiguousArea* is created which stores all the obstacles and the relocation areas in the map $\mathbb{M}$ excepted the selected relocation area $\mathbb{U}$. It has been added to avoid the problem presented before. Firstly, it tries to connect the initial node $\mathbb{X}_0$ that may be certainly defined $\mathbb{X}_0 = \{\mathbf{x}_0\}$ or uncertainly defined by a set $\mathbb{X}_0$ by taking into account these ambiguous relocation areas (line 16). This initial connection despite the ambiguous relocation areas present will be detailed after the explanation of the complete algorithm. If it succeeds to connect to the initial node without ambiguity, the algorithm terminates and the path is provided, otherwise it will test other relocation areas in the neighbourhood of $\mathbb{U}$ in the map $\mathbb{M}$ (line 22). The neighbourhood of $\mathbb{U}$ may be for example the K-nearest neighbour ($K - NN$) where K is defined by the user to reduce the complexity of the graph and not to test all the relocation areas. If a connection between a neighbour node $\mathbb{V}$ and the selected node $\mathbb{U}$ is possible, it has to check if no ambiguity is present between these two relocation areas. The tested relocation area $\mathbb{V}$ is removed from the list *AmbiguousArea* and only the local ambiguous relocation areas between $\mathbb{U}$ and $\mathbb{V}$ are selected for some reasons that will be explained. If it succeeds, a *testScore* is evaluated with the cost between $\mathbb{V}$ and $\mathbb{U}$ and the similar principle as the A* or the Dijkstra algorithm is executed.

Connection between the initial set $\mathbb{X}_0$ and a relocation area

In this part, the explanation about the *connect* function at line 17 of Algorithm 10 between the initial set and a relocation area from the map $\mathbb{M}$ is provided even if the initial position is uncertain.

In reality only the position $\mathbf{q}_0$ is defined since the robot can change its orientation to be in the right direction θ_d.

When the initial position $\mathbb{Q}_0 = \{\mathbf{q}_0\}$ is certainly defined, the set of direction linking this position to a node $\mathbb{A}$ for example was defined in Equation 6.198 and computed according to Equation 6.202 where a simple intersection of projections was used.

However, as depicted in Figure 6.35, the presence of other relocation areas between $\mathbf{q}_0 = (18, 18)$ (red AUV) and the goal set $\mathbb{A}$ may lead to wrong data association. Indeed the AUV may reach the relocation area $\mathbb{B}$ by following the direction $\theta_d = 45°$ that guarantees to reach the desired relocation area $\mathbb{A}$ shown in Figure 6.35(a). Fortunately from this location, it is still possible to follow some directions to reach $\mathbb{A}$ without being bored by $\mathbb{B}$.

The two sets are defined as follows:

$$\mathbb{A} = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 40)^2 + (x_2 - 40)^2} \leq 5\} \tag{6.212}$$

Algorithm 10 Indistinguish-PP (**in** : $\mathbb{X}_0$, $\mathbb{G}$, $\mathbb{M}$, $\mathbb{O}$, **out** : path)

$OPEN \leftarrow \{\mathbb{G}\}$
$CLOSED \leftarrow \emptyset$
$end \leftarrow 0$
$pathFound \leftarrow 0$
$graph.init()$
$prev \leftarrow$ empty map
$g \leftarrow$ map with ∞ value
$f \leftarrow$ map with ∞ value
$g[\mathbb{G}] \leftarrow 0$
$f[\mathbb{G}] \leftarrow h(\mathbb{G}, \mathbb{X}_0)$
while $end \neq 1$ **do**
    $\mathbb{U} \leftarrow argmin_f\{OPEN\}$
    $OPEN \leftarrow OPEN \backslash \{\mathbb{U}\}$
    $CLOSED \leftarrow CLOSED \cup \{\mathbb{U}\}$
    $AmbiguousArea \leftarrow \mathbb{O} \cup (\mathbb{M} \backslash \{\mathbb{U}\})$
    $localAmbiguousArea \leftarrow select(AmbiguousArea, \mathbb{V}, \mathbb{U})$
    if $connect(\mathbb{X}_0, \mathbb{U}, localAmbiguousArea)$ **then**
        $graph.createEdge(\mathbb{X}_0, \mathbb{U})$
        $prev[\mathbb{X}_0] = \mathbb{U}$
        $end \leftarrow 1$
        $pathFound \leftarrow 1$
    else
        for all $\mathbb{V} \in$ neighbourhood($\mathbb{U}$) in $\mathbb{M}$ and $\mathbb{V} \neq \mathbb{U}$ **do**
            if $\mathbb{V} \in CLOSED$ **then**
                continue
            if $\mathbb{X}_{\mathbb{V} \rightarrow \mathbb{U}} \neq \emptyset$ **then**
                $AmbiguousAreaReloc \leftarrow AmbiguousArea \backslash \{\mathbb{V}\}$
                $localAmbiguousArea \leftarrow select(AmbiguousAreaReloc, \mathbb{V}, \mathbb{U})$
                if $connect(\mathbb{V}, \mathbb{U}, localAmbiguousArea)$ **then**
                    $testScore = g[\mathbb{U}] + cost(\mathbb{V}, \mathbb{U})$
                    if $testScore < g[\mathbb{V}]$ **then**
                        $graph.createEdge(\mathbb{V}, \mathbb{U})$
                        $g[\mathbb{V}] = testScore$
                        $f[\mathbb{V}] = testScore + h(\mathbb{V}, \mathbb{X}_0)$
                        $prev[\mathbb{V}] = \mathbb{U}$
                    if $\mathbb{V} \notin OPEN$ **then**
                        $OPEN \leftarrow OPEN \cup \{\mathbb{V}\}$
    if $OPEN = \emptyset$ **then**
        $end \leftarrow 1$
if $pathFound = 1$ **then**
    $path \leftarrow reconstructPath(prev, \mathbb{X}_0)$
else
    $path \leftarrow \emptyset$

and

$$\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 30)^2 + (x_2 - 34)^2} \leq 2\} \tag{6.213}$$

Consider now the following set:

$$W\Theta_{\mathbb{IB}} = \{\theta_d | \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B}\} \tag{6.214}$$

where $\mathbf{x} = (\mathbf{q}_0, \theta_d) = (x_0, y_0, \theta_d)$ and $\mathbb{I}$ refers to the initial location.

This set can be computed using projection as follows:

$$W\Theta_{\mathbb{IB}} = proj_{\mathbb{X}_3}^{[\mathbf{p}]\times\mathbb{T}}(\mathbf{f}_H^{-1}(\mathbb{B})) \tag{6.215}$$

where the initial position $\mathbf{q}_0$ is directly added in the function $\mathbf{f}_H$, $\mathbb{X}_3$ is the domain of θ_d, $[\mathbf{p}] = [-\alpha_\theta, \alpha_\theta] \times \{v_d\}$ and $\mathbb{T}$ is the domain of t. Remind that the speed v_d is defined and is not considered here for the reachability.

This set corresponds to the weak backprojection ($WBACK$) of $\mathbb{B}$ where the initial position $\mathbf{q}_0$ is defined. W has been added before to make the difference between the notation $\Theta_{\mathbb{IB}}$ that guarantees to reach $\mathbb{B}$. Here $\mathbb{B}$ is considered as an obstacle and no trajectories have to cross $\mathbb{B}$, as mentioned in the last section about the weak backprojection. It indicates the set of directions to follow that may reach $\mathbb{B}$ but not guarantee to reach $\mathbb{B}$ for sure.

Finally the set of directions to reach $\mathbb{A}$ and avoid $\mathbb{B}$ is given by:

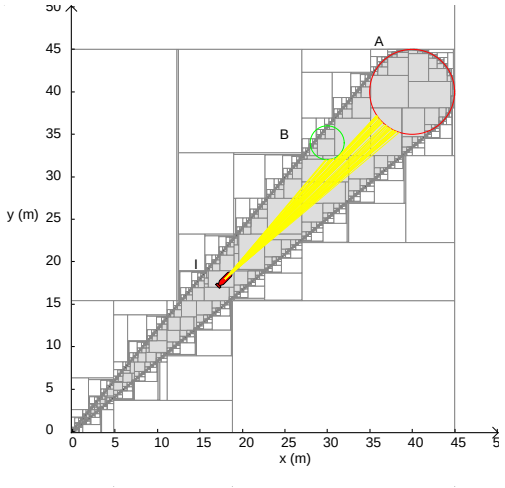
$$\begin{aligned}
\Theta_{\mathbb{IA}}^{\mathbb{B}} &= \{\theta_d | \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A} \\
&\quad \text{and } \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{B}\} && (6.216) \\
&= \{\theta_d | \forall \delta_\theta \in \{-\alpha_\theta, \alpha_\theta\}, \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \\
&\quad \cap \{\theta_d | \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{B}\} && (6.217) \\
&= \Theta_{\mathbb{IA}} \cap \overline{\{\theta_d | \exists \delta_\theta \in \{-\alpha_\theta, \alpha_\theta\}, \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B}\}} && (6.218) \\
&= \Theta_{\mathbb{IA}} \cap \overline{W\Theta_{\mathbb{IB}}} && (6.219)
\end{aligned}$$

where $\mathbf{x} = (\mathbf{q}_0, \theta_d) = (x_0, y_0, \theta_d)$.

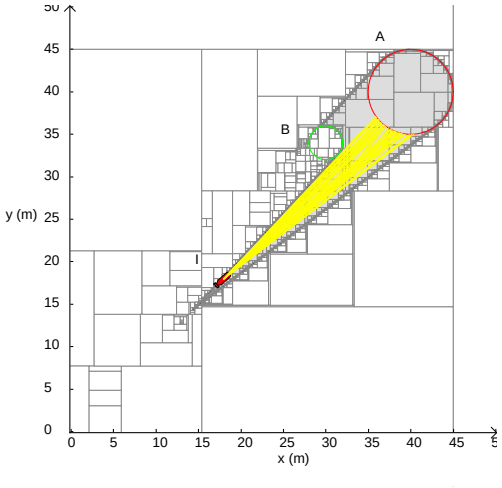
It corresponds also to what was named a termination condition TC where the set to reach is $\mathbb{A}$ by avoiding $\mathbb{B}$. It could be written $TC_{\mathbb{A}}^{\mathbb{B}}(\mathbf{q}_0)$.

The result of this set is given in Figure 6.35(d) where it can be noticed that the direction $\theta_d = 45°$ (red line) cannot reach $\mathbb{A}$ without maybe passing through $\mathbb{B}$. However, if $\theta_d = 42°$ (green line), the robot can execute its mission without being bored by $\mathbb{B}$ as depicted in Figure 6.35(b) where random sampled trajectories are generated according to the uncertain parameter δ_θ on the direction.

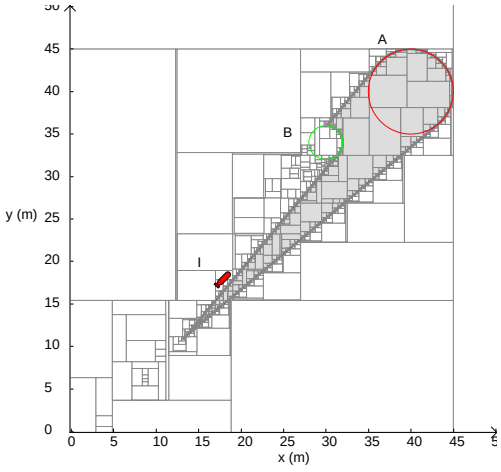
By considering $\mathbb{B}$ as an obstacle, the 2D backward reach set $BACK^{\mathbb{B}}(\mathbb{A}, 45°)$ according to Equation 6.126 is given in Figure 6.35(c) where it can indeed be noticed that the initial location does not belong to the backward reach set.



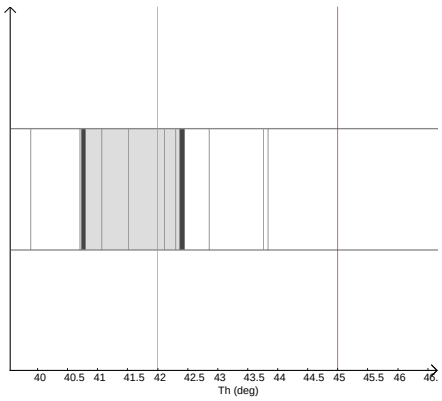
(a) Wrong relocation where some trajectories pass through $\mathbb{B}$ at the direction $\theta_d = 45°$.



(b) Guaranteed reachability with the direction $\theta_d = 42°$.



(c) 2D backward reach set at the direction $\theta_d = 45°$ taking into account $\mathbb{B}$. The initial position does not indeed belong to the set.



(d) Set of direction guaranteeing the reachability of $\mathbb{A}$ from the initial location despite the presence of $\mathbb{B}$. The green line corresponds to the figure (b) and the red line to the figure (c).

Figure 6.35: Reachability of a set $\mathbb{A}$ from the initial location despite the presence of another relocation area $\mathbb{B}$.

When the initial position is uncertainly defined, $\mathbf{q}_0 \in \mathbb{Q}_0$ a method based on Minkowski operators was proposed in the subsection 6.4.2.5 about the goal recognizability to reach a set $\mathbb{A}$ in direction θ_d and in time t. Now the robot has to avoid relocation areas or obstacles. The set of directions to guarantee the reachability of $\mathbb{A}$ despite the presence of $\mathbb{B}$ is also a termination condition, and can be denoted as $TC^{\mathbb{B}}_{\mathbb{A}}(\mathbb{Q}_0)$. A thick set representation was provided in the subsection 6.4.2.5 and can still be defined here according to the uncertain initial position $\mathbf{Q}_0$.

The strong termination condition is defined as follows:

$$\begin{aligned} TC^{\mathbb{B}}_{\mathbb{A}}(\mathbb{Q}_0)^- \quad = \quad & \{\theta_d | \forall \mathbf{q}_0 \in \mathbb{Q}_0, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A} \\ & \text{and } \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{B}\} \end{aligned} \tag{6.220}$$

and the weak termination condition:

$$\begin{aligned} TC_{\mathbb{A}}^{\mathbb{B}}(\mathbb{Q}_0)^{+} &= \{\theta_d | \exists \mathbf{q}_0 \in \mathbb{Q}_0, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A} \\ &\quad \text{and } \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{B}\} \end{aligned} \tag{6.221}$$

where $\mathbf{x} = (\mathbf{q}_0, \theta_d) = (x_0, y_0, \theta_d)$.

Only the strong termination condition is important in this thesis since it guarantees for all initial position in $\mathbb{Q}_0$ that by following a particular direction θ_d, all the robot position won't cross $\mathbb{B}$ and reach $\mathbb{A}$.

A similar proof as in subsection 6.4.2.5 about the thick set representation using Minkowski operators can be made and is proposed below. The reader may refer to the previous proof to understand some steps.

Firstly, the strong termination condition was defined in Equation 6.220 and can be split as in the certain initial position in Equation 6.219. Remind that $\mathbf{x} = (\mathbf{q}_0, \theta_d)$ and it is assumed here that $\mathbf{f}_H$ is $\mathbf{f}_{H2}$ where $\mathbb{A}$ and $\mathbb{B}$ are subsets or $\mathbb{R}^2$. Only the position is important.

$$\begin{aligned} TC_{\mathbb{A}}^{\mathbb{B}}(\mathbb{Q}_0)^{-} &= \{\theta_d | \forall \mathbf{q}_0 \in \mathbb{Q}_0, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \\ &\quad \bigcap \{\theta_d | \forall \mathbf{q}_0 \in \mathbb{Q}_0, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{B}\} \\ &= \{\theta_d | \forall \mathbf{q}_0 \in \mathbb{Q}_0, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A}\} \\ &\quad \bigcap \overline{\{\theta_d | \exists \mathbf{q}_0 \in \mathbb{Q}_0, \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B}\}} \\ &= \{\theta_d | \forall \mathbf{q}_0 \in \mathbb{Q}_0, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{q}_0 + v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{A}\} \\ &\quad \overline{\bigcap \{\theta_d | \exists \mathbf{q}_0 \in \mathbb{Q}_0, \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{q}_0 + v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{B}\}} \\ &= \{\theta_d | \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbb{Q}_0 + v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \subset \mathbb{A}\} \\ &\quad \overline{\bigcap \{\theta_d | \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{B} \oplus (-\mathbb{Q}_0)\}} \\ &= \{\theta_d | \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{A} \ominus \mathbb{Q}_0\} \\ &\quad \overline{\bigcap \{\theta_d | \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{B} \oplus (-\mathbb{Q}_0)\}} \end{aligned}$$

The reader may refer to Equation 6.176 to understand why a Minkowski sum appears.

Since $\mathbb{A}$ and $\mathbb{Q}_0$ are connected sets, then $\mathbb{A} \ominus \mathbb{Q}_0$ is also connected. Consequently, it leads to:

$$\begin{aligned} TC_{\mathbb{A}}^{\mathbb{B}}(\mathbb{Q}_0)^{-} &= \{\theta_d | \forall \delta_\theta \in \{-\alpha_\theta, \alpha_\theta\}, \exists t \in \mathbb{R}^+, v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{A} \ominus \mathbb{Q}_0\} \\ &\quad \overline{\bigcap \{\theta_d | \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{B} \oplus (-\mathbb{Q}_0)\}} \end{aligned} \tag{6.222}$$

Define the following function $\mathbf{h}$:

$$\mathbf{h}: \begin{cases} \mathbb{R} \times [-\alpha_\theta, \alpha_\theta] \times \mathbb{T} & \rightarrow & \mathbb{R}^2 \\ (\theta_d, \delta_\theta, t) & \rightarrow & v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \end{cases} \tag{6.223}$$

where $v = v_d$ the defined speed.

Finally, the strong termination condition can be computed with projection as follows:

$$TC_{\mathbb{A}}^{\mathbb{B}}(\mathbb{Q}_0)^- = \frac{\Big(\bigcap_{\delta_\theta \in \{\alpha_\theta, \alpha_\theta\}} proj_{\mathbb{X}_3}^{\mathbb{T}}(\mathbf{h}^{-1}(\mathbb{A} \ominus \mathbb{Q}_0)) \Big) \bigcap}{proj_{\mathbb{X}_3}^{[\delta_\theta] \times \mathbb{T}}(\mathbf{h}^{-1}(\mathbb{B} \oplus (-\mathbb{Q}_0)))} \tag{6.224}$$

where $[\delta_\theta] = [-\alpha_\theta, \alpha_\theta]$ and $\mathbb{X}_3$ is the domain of θ_d.

Secondly and similarly it can be proved that the weak termination condition is given by:

$$TC_{\mathbb{A}}^{\mathbb{B}}(\mathbb{Q}_0)^+ = \frac{\{\theta_d | \forall \delta_\theta \in \{-\alpha_\theta, \alpha_\theta\}, \exists t \in \mathbb{R}^+, v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{A} \oplus (-\mathbb{Q}_0)\}}{\bigcap \{\theta_d | \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{B} \ominus \mathbb{Q}_0\}} \tag{6.225}$$

Using the same function $\mathbf{h}$, it can be rewritten with projection:

$$TC_{\mathbb{A}}^{\mathbb{B}}(\mathbb{Q}_0)^+ = \frac{\Big(\bigcap_{\delta_\theta \in \{\alpha_\theta, \alpha_\theta\}} proj_{\mathbb{X}_3}^{\mathbb{T}}(\mathbf{h}^{-1}(\mathbb{A} \oplus (-\mathbb{Q}_0))) \Big) \bigcap}{proj_{\mathbb{X}_3}^{[\delta_\theta] \times \mathbb{T}}(\mathbf{h}^{-1}(\mathbb{B} \ominus \mathbb{Q}_0))} \tag{6.226}$$

where $[\delta_\theta] = [-\alpha_\theta, \alpha_\theta]$ and $\mathbb{X}_3$ is the domain of θ_d.

The thick set about the termination condition considering an obstacle is then given by:

$$TC_{\mathbb{A}}^{\mathbb{B}}(\mathbb{Q}_0) = [TC_{\mathbb{A}}^{\mathbb{B}}(\mathbb{Q}_0)^-, TC_{\mathbb{A}}^{\mathbb{B}}(\mathbb{Q}_0)^+] \tag{6.227}$$

which indicates the set of direction for the reachability of $\mathbb{A}$ despite the presence of another relocation area or obstacle $\mathbb{B}$ between $\mathbb{Q}_0$ and $\mathbb{A}$.

Generally, an uncertain position is defined by a most likely position $\mathbf{q}_0$ which corresponds to the middle of a box, the center of a circle, or even the center of an ellipse. This center is defined by $\mathbf{q}_0$. Then the set of position is defined as follows:

$$\mathbb{Q}_0 = \{\mathbf{x} \in \mathbb{R}^2 | \forall \epsilon_{xy} \in \mathbb{P}_0, \mathbf{x} = \mathbf{q}_0 + \epsilon_{xy}\} \tag{6.228}$$

where $\mathbb{P}_0$ corresponds to the uncertain set of position centered at $(0, 0)$. The set $\mathbb{Q}_0$ is then simply a translation of $\mathbb{P}_0$ at the most likely position $\mathbf{q}_0$.

According to this definition with the most likely position in Equation 6.228, the strong termination condition $TC(\mathbb{Q}_0)^-$ expressed in Equation 6.220 can be reformulated as follows:

$$\begin{aligned}
TC^{\mathbb{B}}_{\mathbb{A}}(\mathbb{Q}_0)^- &= \{\theta_d | \forall \mathbf{q}_0 \in \mathbb{Q}_0, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A} \\
&\quad \text{and } \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{B}\} \qquad (6.229) \\
&= \{\theta_d | \forall \epsilon_{xy} \in \mathbb{P}_0, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x} + \epsilon_{xy}, t, \mathbf{p}) \in \mathbb{A} \\
&\quad \text{and } \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x} + \epsilon_{xy}, t, \mathbf{p}) \notin \mathbb{B}\} \qquad (6.230) \\
&= \{\theta_d | \forall \epsilon_{xy} \in \mathbb{P}_0, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{q}_0 + \epsilon_{xy} + v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{A}\} \\
&\quad \overline{\bigcap \{\theta_d | \exists \epsilon_{xy} \in \mathbb{P}_0, \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{q}_0 + \epsilon_{xy} + v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix} \in \mathbb{B}\}} \\
&= \{\theta_d | \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) + \mathbb{P}_0 \subset \mathbb{A}\} \\
&\quad \bigcap \overline{\{\theta_d | \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B} \oplus (-\mathbb{P}_0)\}} \\
&= \{\theta_d | \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A} \ominus \mathbb{P}_0\} \\
&\quad \bigcap \overline{\{\theta_d | \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B} \oplus (-\mathbb{P}_0)\}} \qquad (6.231)
\end{aligned}$$

where $\mathbf{x} = (\mathbf{q}_0, \theta_d) = (x_0, y_0, \theta_d)$, $\mathbf{p} = (\delta_\theta, v)$ and $\mathbb{P}_0$ is the set of uncertain position centered at $(0, 0)$. The reader may refer to the previous demonstration to understand the different steps.

And the weak termination condition can be defined as follows:

$$\begin{aligned}
TC^{\mathbb{B}}_{\mathbb{A}}(\mathbb{Q}_0)^+ &= \{\theta_d | \forall \delta_\theta \in \{-\alpha_\theta, \alpha_\theta\}, \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{A} \oplus (-\mathbb{P}_0)\} \\
&\quad \bigcap \overline{\{\theta_d | \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B} \ominus \mathbb{P}_0\}} \qquad (6.232)
\end{aligned}$$

Similarly, these sets can be defined with projection.

Consider the same sets $\mathbb{A}$ and $\mathbb{B}$, but now the initial location is defined as follows:

$$\mathbb{Q}_0 = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 28)^2 + (x_2 - 27)^2} \leq 0.5\} \qquad (6.233)$$

where the initial position $\mathbf{q}_0$ belongs a disk centered in $(28, 27)$ with a radius of $0.5m$.

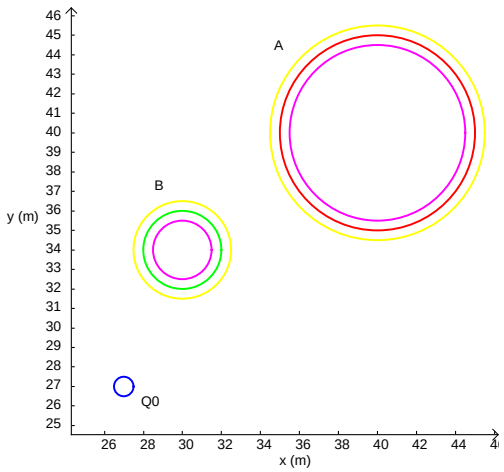
This disk refers to the set $\mathbb{P}_0$ defined then as follows:

$$\mathbb{P}_0 = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{x_1^2 + x_2^2} \leq 0.5\} \qquad (6.234)$$

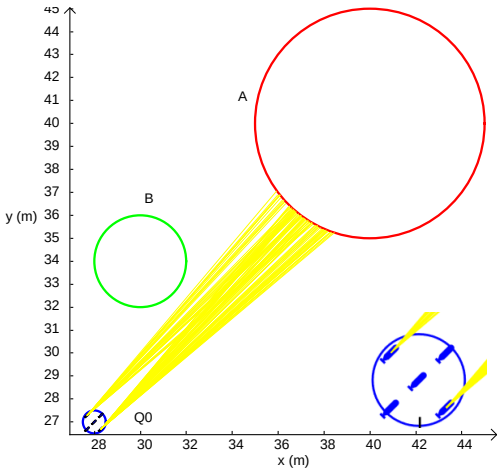
where the most likely position $\mathbf{q}_0 = (28, 27)$.

The results are presented in Figure 6.36 where the environment is depicted in the figure (a). The erosion and the dilation according to $\mathbb{P}_0$ are shown in magenta and in yellow respectively for both sets $\mathbb{A}$ and $\mathbb{B}$. Due to the definition of a disk centered in $(0, 0)$ for $\mathbb{P}_0$, $\mathbb{P}_0 = -\mathbb{P}_0$. The uncertain initial position $\mathbb{Q}_0$ is represented by the blue circle. The thick solution set is depicted in Figure 6.36(e) in a grayscale convention for thick set. Three directions are selected. Firstly, with $\theta_d = 45°$ inside the solution set, all the initial position (shown by blue AUVs) in Figure 6.36(a) reach $\mathbb{A}$ without any ambiguity with $\mathbb{B}$ as it can be noticed that any yellow trajectories cross $\mathbb{B}$. Secondly, with $\theta_d = 56°$ inside the penumbra, only one sampled position (in blue AUV) in Figure 6.36(c) guarantees the reachability (blue trajectories) of $\mathbb{A}$ despite the presence of $\mathbb{B}$. For one sampled position (green AUV), which does not provide the guarantee, the trajectories are represented in yellow and it can be obviously noticed that it crosses at some

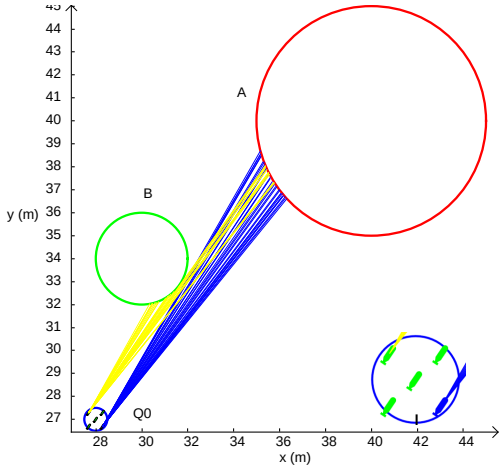
time the set $\mathbb{B}$. Finally, at the direction $\theta_d = 60^\circ$ outside the solution set, any initial position guarantees the reachability of $\mathbb{A}$ without the ambiguity of $\mathbb{B}$ or simply not all the reachability of $\mathbb{A}$ for all the possible trajectories according to the uncertain parameter δ_θ on the direction.



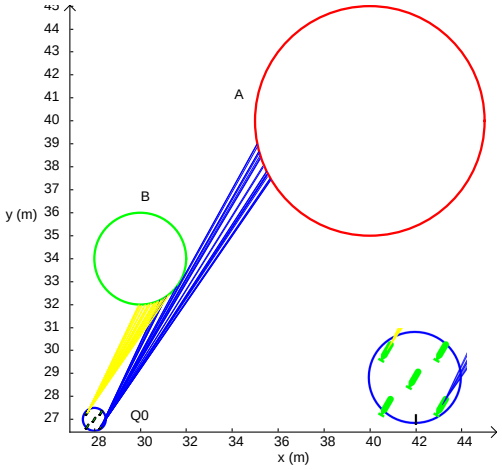
(a) Environment with set initial pose $\mathbb{Q}_0$ in the blue circle, $\mathbb{A}$ is represented by the red circle and $\mathbb{B}$ by the blue circle. Magenta and yellow circle are the erosion ($\mathbb{A} \ominus \mathbb{P}_0$ and $\mathbb{B} \ominus \mathbb{P}_0$) and the dilation ($\mathbb{A} \oplus \mathbb{P}_0$ and $\mathbb{B} \oplus \mathbb{P}_0$) of the sets $\mathbb{A}$ and $\mathbb{B}$.



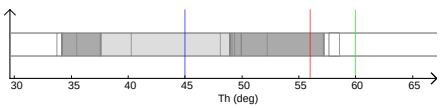
(b) $\theta_d = 45°$ inside the solution set (blue line in the figure (e)), where all the sampled position in blue AUV can reach $\mathbb{A}$. A zoom of the initial poses in $\mathbb{Q}_0$ is provided in the bottom right corner.



(c) $\theta_d = 56°$ inside the penumbra (red line in the figure (e)), where some position guarantees to reach $\mathbb{A}$ (position in blue AUV and blue trajectories) and others (green AUV and yellow trajectories) with trajectories that cross $\mathbb{B}$. Similarly a zoom is provided.



(d) $\theta_d = 60°$ outside the solution set (green line in the figure (e)), where no initial position (green AUV) can reach for sure $\mathbb{A}$ without ambiguity with $\mathbb{B}$ or even reachability of $\mathbb{A}$. Similarly a zoom is provided.



(e) Direction reachability $TC_{\mathbb{A}}^{\mathbb{B}}(\mathbb{Q}_0)$ in a thick representation. Grayscale convention.

Figure 6.36: Reachability of a disk $\mathbb{A}$ with another relocation area $\mathbb{B}$ and uncertain initial position $\mathbb{Q}_0$.

As mentioned in the section 6.4.2.4 about including forbidden areas, a local function will be provided to remove the relocation areas that may lie behind the desired relocation areas since the computation relies on the weak backprojection of the obstacles or ambiguous relocation

areas.

Connection between relocation areas

In this part, it will be explained how to connect two relocation areas mentioned in the *connect* function at line 29 in Algorithm 10 when the robot is certainly and uncertainly located in the departure relocation area.

Firstly, it will be explained when the robot is certainly located in the departure relocation area.

The connection between two relocation areas could be performed according to Equation 6.55 by proving that the set was not empty. Due to the presence of another relocation areas that may lead to ambiguities, a method was provided in the section 6.4.2.4 about the forbidden areas under the condition that any another relocation areas lie behind the desired relocation area. Consequently, leaving a relocation $\mathbb{A}$ and reaching a relocation $\mathbb{B}$ with the presence of another relocation $\mathbb{C}$ between $\mathbb{A}$ and $\mathbb{B}$ can be defined according to Equation 6.125:

$$\mathbb{X}_{\mathbb{A}\mathbb{B}}^{\mathbb{C}} = \{\mathbf{x} \in \mathbb{R}^3 | BACK_{od}^{\mathbb{C}}(\mathbb{B}) \cap \mathbb{A} \neq \emptyset\} \tag{6.235}$$

which was decomposed into:

$$\mathbb{X}_{\mathbb{A}\mathbb{B}}^{\mathbb{C}} = \{\mathbf{x} \in \mathbb{R}^3 | BACK_{od}(\mathbb{B}) \cap \overline{WBACK_{od}(\mathbb{C})} \cap \mathbb{A} \neq \emptyset\} \tag{6.236}$$

where $\mathbb{A}$ can be defined as $\mathbb{A} = \mathbb{A} \times \mathbb{R}$ to have a 3D set if $\mathbb{A}$ was initially defined in 2D.

As it may be noticed, it is similar to the previous set giving the connection between two sets in Equation 6.55 where an additional constraint has been added about the ambiguity with other relocation areas.

Due to the 3D aspect of the set, it was reduced to a single dimension along the set of directions using projection.

$$\Theta_{\mathbb{A}\mathbb{B}}^{\mathbb{C}} = \{\theta_d \in \mathbb{R} | BACK^{\mathbb{C}}(\mathbb{B}, \theta_d) \cap \mathbb{A} \neq \emptyset\} \tag{6.237}$$

which can be also decomposed into:

$$\Theta_{\mathbb{A}\mathbb{B}}^{\mathbb{C}} = \{\theta_d \in \mathbb{R} | BACK(\mathbb{B}, \theta_d) \cap \overline{WBACK(\mathbb{C}, \theta_d)} \cap \mathbb{A} \neq \emptyset\} \tag{6.238}$$

Coming back to the true definition in Equation 6.237 leads to:

$$\begin{aligned}
\Theta_{\mathbb{A}\mathbb{B}}^{\mathbb{C}} &= \{\theta_d \in \mathbb{R} | BACK^{\mathbb{C}}(\mathbb{B}, \theta_d) \cap \mathbb{A} \neq \emptyset\} && (6.239)\\
&= \{\theta_d \in \mathbb{R} | BACK(\mathbb{B}, \theta_d) \cap \overline{WBACK(\mathbb{C}, \theta_d)} \cap \mathbb{A} \neq \emptyset\} && (6.240)\\
&= \{\theta_d \in \mathbb{R} | \exists \mathbf{a} \in \mathbb{A}, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B} \\
&\quad \text{and } \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{C}\} && (6.241)
\end{aligned}$$

where $\mathbf{x} = (\mathbf{a}, \theta_d)$.

It may be noticed that finding the set of direction linking $\mathbb{A}$ to $\mathbb{B}$ by avoiding $\mathbb{C}$ is the same as $TC_{\mathbb{A}}^{\mathbb{B}}(\mathbb{Q}_0)^+$ defined in Equation 6.221 where $\mathbb{Q}_0$ is $\mathbb{A}$, $\mathbb{A}$ is $\mathbb{B}$ and $\mathbb{B}$ is $\mathbb{C}$ in the new equation.

Consequently, it results the following proposition:

$$\Theta^{\mathbb{C}}_{\mathbb{A}\mathbb{B}} = TC^{\mathbb{C}}_{\mathbb{B}}(\mathbb{A})^{+} \tag{6.242}$$

with the same system of notation.

It leads then to the following proposition with the right notation associated to the problem:

$$\begin{aligned}\Theta^{\mathbb{C}}_{\mathbb{A}\mathbb{B}} &= \{\theta_d | \forall \delta_\theta \in \{-\alpha_\theta, \alpha_\theta\}, \exists t \in \mathbb{R}^+, v \cdot t \cdot \begin{pmatrix}\cos(\theta_d + \delta_\theta)\\ \sin(\theta_d + \delta_\theta)\end{pmatrix} \in \mathbb{B} \oplus (-\mathbb{A})\} \\ &\overline{\bigcap \{\theta_d | \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, v \cdot t \cdot \begin{pmatrix}\cos(\theta_d + \delta_\theta)\\ \sin(\theta_d + \delta_\theta)\end{pmatrix} \in \mathbb{C} \ominus \mathbb{A}\}}\end{aligned} \tag{6.243}$$

However, it may be possible that $\mathbb{C} \ominus \mathbb{A}$ becomes empty.

The set of directions linking $\mathbb{A}$ to $\mathbb{B}$ by avoiding $\mathbb{C}$ will be computed according to Equation 6.237 or as the decomposition in Equation 6.238 which can be computed with projection according to Equation 6.67 as follows:

$$\Theta^{\mathbb{C}}_{\mathbb{A}\mathbb{B}} = proj^{\mathbb{X}_1 \times \mathbb{X}_2}_{\mathbb{X}_3}\Big(\big(\bigcap_{\delta\theta \in \{-\alpha_\theta, \alpha_\theta\}} proj^{\mathbb{T}}_{\mathbb{X}}(\mathbf{f}_H^{-1}(\mathbb{B})) \big) \bigcap \overline{proj^{[-\alpha_\theta, \alpha_\theta] \times \mathbb{T}}_{\mathbb{X}}(\mathbf{f}_H^{-1}(\mathbb{C}))} \bigcap \mathbb{A} \Big) \tag{6.244}$$

where $\mathbb{X}_3$ is the domain of θ_d (direction), $\mathbb{X}_1$ corresponds to the domain of x and $\mathbb{X}_2$ is the domain of y, and finally $\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2 \times \mathbb{X}_3$. The set $\mathbb{A} = \mathbb{A}_{2D} \times \mathbb{R}$ since $\mathbb{A}$ is usually defined in the $(x - y)$ plane.

Notice that $BACK_{od}(\mathbb{B})$ corresponds indeed to Equation 6.49 and $WBACK_{od}(\mathbb{C})$ to Equation 6.124.

Remind that the set of directions corresponds to:

$$\Theta^{\mathbb{C}}_{\mathbb{A}\mathbb{B}} = proj^{\mathbb{X}_1 \times \mathbb{X}_2}_{\mathbb{X}_3}(\mathbb{X}^{\mathbb{C}}_{\mathbb{A}\mathbb{B}}) \tag{6.245}$$

but the solution was only provided along the dimension of the directions to have a visual aspect.

Consider the three following sets as example and represented in Figure 6.37:

$$\mathbb{A} = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 15)^2 + (x_2 - 15)^2} \leq 5\} \tag{6.246}$$

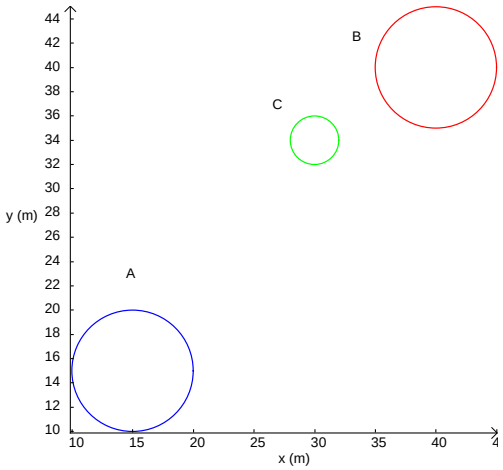
$$\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 40)^2 + (x_2 - 40)^2} \leq 5\} \tag{6.247}$$

and

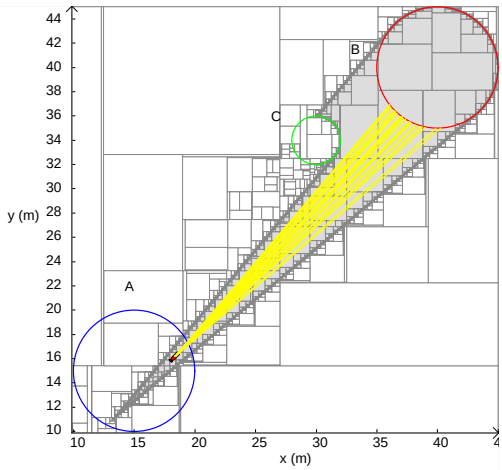
$$\mathbb{C} = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - 30)^2 + (x_2 - 34)^2} \leq 2\} \tag{6.248}$$

where $\mathbb{A}$ is the departure set (in blue), $\mathbb{B}$ is the goal area (in red) and $\mathbb{C}$ is another relocation area or an obstacle (in green) that has to be avoided for all possible trajectories. The set of directions $\Theta^{\mathbb{C}}_{\mathbb{A}\mathbb{B}}$ according to Equation 6.237 and computed according to Equation 6.244 is represented in Figure 6.37(e). The 2d backward reach set at the defined direction $\theta_d = 45°$ (represented by a green line in Figure 6.37(e)) is shown in Figure 6.37(b). An initial location of a possible position of the AUV inside $\mathbb{A}$ is represented by a red AUV. Random trajectories due to the uncertain parameter on the direction $\alpha_\theta = 5°$ are drawn in yellow. The same representation is provided in Figure 6.37(c) at the direction $\theta_d = 39°$. Notice that in both situation,

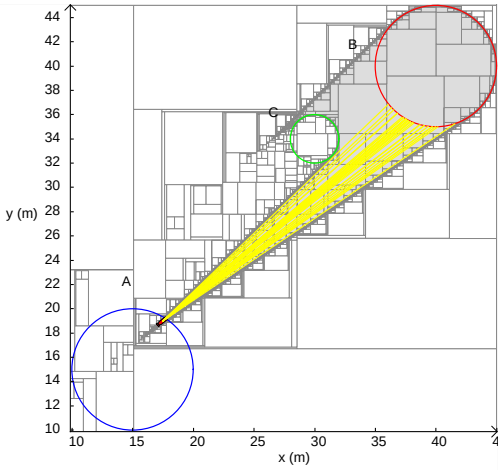
which are proved to reach $\mathbb{B}$ from $\mathbb{A}$ without any ambiguity (green line in the solution set), all the yellow trajectories do not cross $\mathbb{C}$ at any time. The 2D backward reach set at the direction $\theta_d = 36°$ selected outside the solution at the red line in Figure 6.37(e) is represented in Figure 6.37(d). Indeed, it is not possible to leave $\mathbb{A}$ and reach $\mathbb{B}$ with the presence of $\mathbb{C}$ at this direction since the 2D backward reach set does not intersect $\mathbb{A}$.



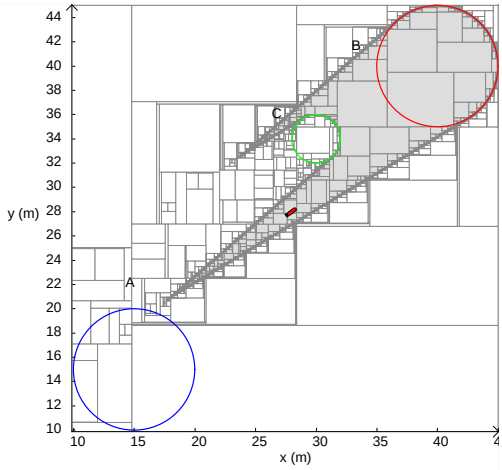
(a) Environment with a goal area $\mathbb{B}$ from a relocation area $\mathbb{A}$ with an obstacle or another relocation area $\mathbb{C}$.



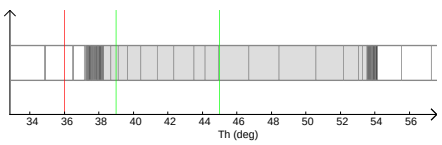
(b) 2D backward reach set at $\theta_d = 45°$ (green line in the figure (e)) where some trajectories in yellow are represented from a possible initial location at the red AUV.



(c) 2D backward reach set at $\theta_d = 39°$ (green line in the figure (e)) where some trajectories in yellow are represented from a possible initial location at the red AUV.



(d) 2D backward reach set at $\theta_d = 36°$ (red line in the figure (e)), no possible link at this direction.



(e) Set of directions $\Theta^{\mathbb{C}}_{\mathbb{AB}}$.

Figure 6.37: Reachability of a disk $\mathbb{B}$ from a relocation area $\mathbb{B}$ with another relocation area $\mathbb{C}$ between them.

Secondly, if the robot is uncertainly located in the departure relocation area, it can be de-

scribed by the most likely position $\mathbf{q}_0$ (the heading is provided by the compass) and an error on this position by a set $\mathbb{P}_0$ that may be an ellipse, a box or a disk for example. This is similar to the problem of the connection between the initial uncertain position and a relocation area.

For example, consider an uncertainty defined by a disk $\mathbb{P}_0$ as follows:

$$\mathbb{P}_0 = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{x_1^2 + x_2^2} \leq 0.5\} \tag{6.249}$$

which means that the position lies in a circle with a radius $0.5m$ around $\mathbf{q}_0$.

The set of directions is then given, inspired by Equation 6.237, by the following set:

$$\begin{aligned} \Theta_{\mathbb{A}\mathbb{B}}^{\mathbb{C}}(\mathbb{P}_0) &= \{\theta_d \in \mathbb{R} | \exists \mathbf{a} \in \mathbb{A}, \forall \epsilon_{xy} \in \mathbb{P}_0, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B} \\ &\quad \text{and } \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{C} \text{ and } \mathbf{a} + \epsilon_{xy} \in \mathbb{A}\} \end{aligned} \tag{6.250}$$

where $\mathbf{x} = (\mathbf{a}, \theta_d) = (a_1, a_2, \theta_d)$.

Since the departure position $\mathbf{a}$ is uncertainly located inside $\mathbb{A}$, it has to be checked for all possible positions in the uncertain set $\mathbb{P}_0$ centered in $\mathbf{a}$ that one direction θ_d exists such that all the possible positions will reach $\mathbb{B}$ without being bored by $\mathbb{C}$.

This can be reformulated as follows:

$$\begin{aligned} \Theta_{\mathbb{A}\mathbb{B}}^{\mathbb{C}}(\mathbb{P}_0) &= \{\theta_d \in \mathbb{R} | \exists \mathbf{a} \in \mathbb{A}, \forall \epsilon_{xy} \in \mathbb{P}_0, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B} \\ &\quad \text{and } \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{C} \text{ and } \mathbf{a} + \epsilon_{xy} \in \mathbb{A}\} \end{aligned} \tag{6.251}$$

$$\begin{aligned} &= \{\theta_d \in \mathbb{R} | \exists \mathbf{a} \in \mathbb{A}, \forall \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \in \mathbb{B} \\ &\quad \text{and } \forall t \in \mathbb{R}^+, \mathbf{f}_H(\mathbf{x}, t, \mathbf{p}) \notin \mathbb{C} \text{ and } \mathbf{a} \in \mathbb{A} \ominus \mathbb{P}_0\} \end{aligned} \tag{6.252}$$

which can be given by:

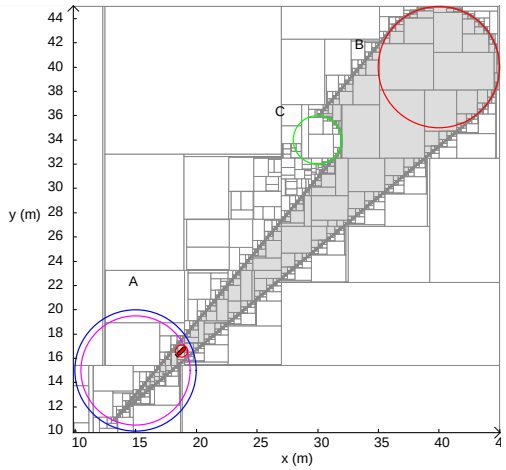
$$\Theta_{\mathbb{A}\mathbb{B}}^{\mathbb{C}}(\mathbb{P}_0) = \{\theta_d \in \mathbb{R} | BACK^{\mathbb{C}}(\mathbb{B}, \theta_d) \cap (\mathbb{A} \ominus \mathbb{P}_0) \neq \emptyset\} \tag{6.253}$$

which can be also decomposed into:

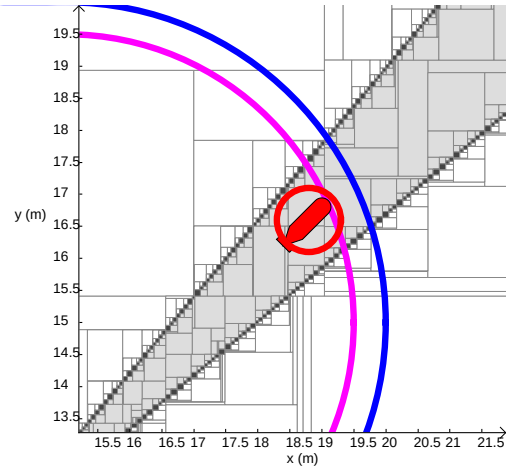
$$\Theta_{\mathbb{A}\mathbb{B}}^{\mathbb{C}}(\mathbb{P}_0) = \{\theta_d \in \mathbb{R} | BACK(\mathbb{B}, \theta_d) \cap \overline{WBACK(\mathbb{C}, \theta_d)} \cap (\mathbb{A} \ominus \mathbb{P}_0) \neq \emptyset\} \tag{6.254}$$

When the robot is uncertainly located in the departure relocation area $\mathbb{A}$ in a set $\mathbb{P}_0$ around the most likely position, then consider the erosion of $\mathbb{A}$ by $\mathbb{P}_0$ instead of $\mathbb{A}$ in Equation 6.237.

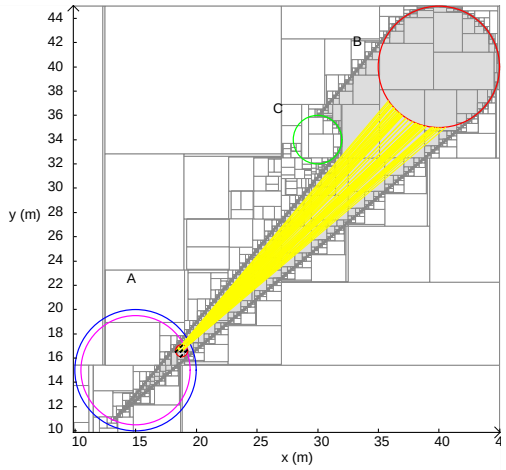
Consider the three previous sets $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$. Consider an uncertain location in $\mathbb{A}$ defined by the disk $\mathbb{P}_0$ (disk with a radius equal to $0.5m$).



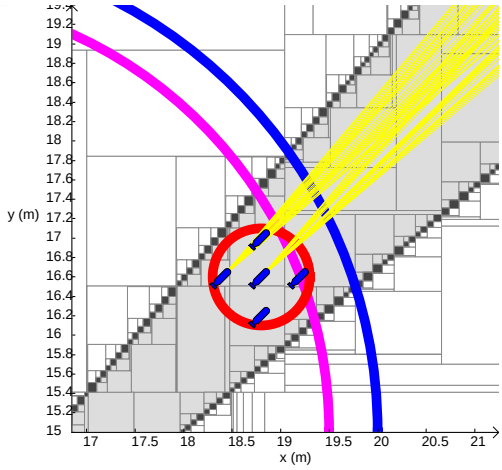
(a) 2D backward reach set of $\mathbb{B}$ at $\theta_d = 45°$ in presence of $\mathbb{C}$. The magenta circle is the erosion $\mathbb{A} \ominus \mathbb{P}_0$.



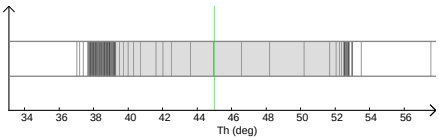
(b) Zoom of the figure (a).



(c) Valid initial positions (blue AUVs) with the direction $\theta_d = 45°$ with yellow trajectories for two initial poses.



(d) Zoom of the figure (c).



(e) Set of directions $\Theta^{\mathbb{C}}_{\mathbb{A}\mathbb{B}}(\mathbb{P}_0)$.

Figure 6.38: Reachability of a disk $\mathbb{B}$ with another relocation area $\mathbb{C}$ from a relocation area $\mathbb{A}$ with an initial uncertain position.

The set of directions $\Theta^{\mathbb{C}}_{\mathbb{A}\mathbb{B}}(\mathbb{P}_0)$ is depicted in Figure 6.38(e) and looks reduced compared to the previous set $\Theta^{\mathbb{C}}_{\mathbb{A}\mathbb{B}}$ depicted in Figure 6.37 without uncertainty in $\mathbb{A}$. The 2D backward reach set at $\theta_d = 45°$ is shown in Figure 6.38(a) where the magenta circle shows the erosion of $\mathbb{A}$ by $\mathbb{P}_0$. From the initial uncertain location at the red AUV, it is possible to reach for sure $\mathbb{B}$ without ambiguity. The 2D backward reach has to intersect the magenta disk to conclude the existence of a position guaranteeing the reachability despite the uncertain location. A zoom is provided in Figure 6.38(b) where the red disk indicates the uncertain location centered at the red AUV. Notice that this disk is included in the 2D backward reach set at $\theta_d = 45°$. Some positions, represented by blue AUV, are sampled in this uncertain position, where it concluded

that all the trajectories reach $\mathbb{B}$ without crossing $\mathbb{C}$. A zoom is given in Figure 6.38(d) where sampled trajectories are represented in yellow from two initial locations.

Similarly when linking the initial position to a relocation area, connecting two relocation areas without any ambiguity needs a selection step to remove the sets that are behind the desired relocation area. This will be proposed in the following.

Selecting local ambiguous area

As pointed out in the two previous parts about the connection, a *select* function is present in Algorithm 10 at line 16 and 28 to remove the relocation areas or obstacles that may be lie behind the desired relocation area to reach since the weak projection of these sets may hide the backward reach set of the desired relocation areas. This was explained in section 6.4.2.4 about the forbidden areas in **Example 2**.

In Algorithm 10, at line 15, a variable *AmbiguousArea* stores all the obstacles in $\mathbb{O}$ and all the relocation areas excepted the one that is tested. According to the initial position, some of them are removed since they could hide the backward reach set of the tested relocation area $\mathbb{U}$. This is realized by the *select* function. Similarly at line 28, the *select* function removes the relocation area behind the desired one $\mathbb{U}$ on a reduced set *AmbiguousAreaReloc* which corresponds to the previous *AmbiguousArea* where the tested node $\mathbb{V}$ has been removed too.

Consider two relocation areas $\mathbb{A}$ and $\mathbb{B}$, where $\mathbb{A}$ corresponds to $\mathbb{V}$ and $\mathbb{B}$ to $\mathbb{U}$ in Algorithm 10 when the connection with the initial location has not been proved. $\mathbb{B}$ is then the relocation area to reach and $\mathbb{A}$ is the departure area. The set of directions $\Theta_{\mathbb{AB}}$ to leave $\mathbb{A}$ and reach $\mathbb{B}$ was given in Equation 6.64. Consider now a third set $\mathbb{C}$ to test if it may lie behind the desired relocation area $\mathbb{B}$ to avoid any ambiguity when computing the reachability as previously presented.

The set of directions $\Theta_{\mathbb{CB}}$ linking $\mathbb{C}$ to $\mathbb{B}$ is also given by Equation 6.64. Since the set of directions linking $\mathbb{A}$ to $\mathbb{B}$ is given by $\Theta_{\mathbb{AB}}$, then consider the following set:

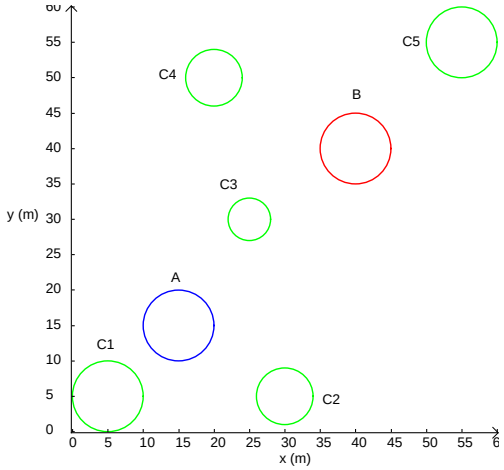
$$\Theta^{amb}_{\mathbb{AB}}(\mathbb{C}) = \{\theta_d \in \mathbb{R} | \Theta_{\mathbb{AB}} \cap \Theta_{\mathbb{CB}}\} \tag{6.255}$$

This set indicates the set of directions that are common between $\mathbb{A}$ and $\mathbb{C}$ to reach $\mathbb{B}$.

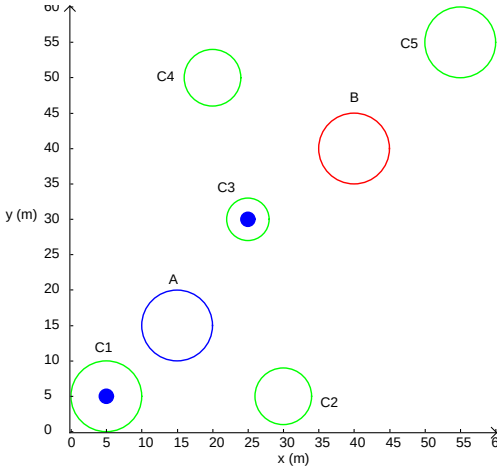
For the considered relocation area or obstacle $\mathbb{C}$, if the set $\Theta^{amb}_{\mathbb{AB}}(\mathbb{C})$ is empty, then the tested relocation area will not be considered as ambiguous since it is possible to travel easily from $\mathbb{A}$ to $\mathbb{B}$. However, if it is not empty, $\mathbb{C}$ is added to the *localAmbiguousArea* variable as some directions may lead the robot to cross the relocation area $\mathbb{C}$ due to the uncertain direction instead of $\mathbb{B}$ by leaving $\mathbb{A}$. The reachability of an undesired relocation area will lead to a wrong data association and then a wrong strategy following.

The *select* function consists then in testing all the relocation areas or obstacles, or at least in the local environment of $\mathbb{A}$ and $\mathbb{B}$ as depicted for example by the red box in Figure 6.39(c), if one is present between $\mathbb{A}$ and $\mathbb{B}$. An environment composed of a goal area $\mathbb{B}$ in red, a starting area $\mathbb{A}$ and another possible ambiguous relocation areas $\mathbb{C}_i$ in green are represented in Figure 6.39(a). For all $\mathbb{C}_i$, $\Theta^{amb}_{\mathbb{AB}}(\mathbb{C}_i)$ is computed according to Equation 6.255. If $\Theta^{amb}_{\mathbb{AB}}(\mathbb{C}_i) \neq \emptyset$, the set $\mathbb{C}_i$ is added to the *localAmbiguousArea* as for the set $\mathbb{C}_1$ and $\mathbb{C}_3$ highlighted by a blue disk at their center as shown in Figure 6.39(b). All the other areas $\mathbb{C}_i$ are not taken into account. As it may be noticed, the set $\mathbb{C}_1$ does not lie between $\mathbb{A}$ and $\mathbb{B}$, it could be removed by considering the 2D forward reach set of $\mathbb{A}$ defined at the possible directions $\Theta_{\mathbb{AB}}$ between $\mathbb{A}$ and $\mathbb{B}$ as shown

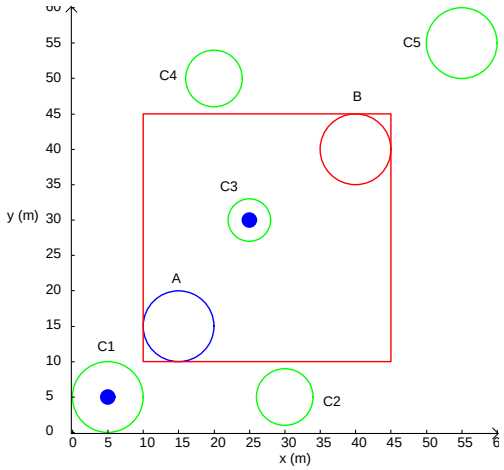
in Figure 6.39(c) at the direction $\theta_d = 45°$. The forward reach set will be explained just below. It could also have been removed since it does not belong to the local area represented by the red box in Figure 6.39(c). The red box englobing the sets $\mathbb{A}$ and $\mathbb{B}$ may be used as a preprocessing step to keep only the possible relocation areas or obstacles that intersect this box to improve the speed of the algorithm. Then the test based on the direction in Equation 6.255 could be used to remove any possible relocation areas or obstacles that could lie behind the goal area $\mathbb{B}$ when $\mathbb{B}$ is a complex shape or are outside the possible direction. Finally the 2D forward reach set of $\mathbb{A}$ on the set of directions linking $\mathbb{A}$ to $\mathbb{B}$ could remove the sets that are before $\mathbb{A}$.



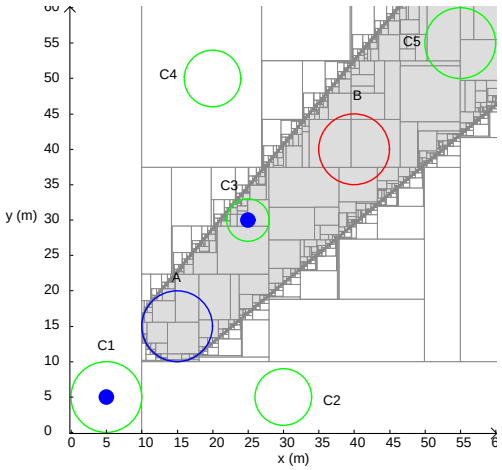
(a) Environment with a goal area $\mathbb{B}$ in red, a starting area $\mathbb{A}$ in blue and some relocation areas or obstacles $\mathbb{C}_i$ in green.



(b) Ambiguous relocation areas with a blue disk based on $\Theta^{amb}_{\mathbb{AB}}(\mathbb{C}_i)$.



(c) Local environment between $\mathbb{A}$ and $\mathbb{B}$ by a red box.



(d) 2D forward reach set of $\mathbb{A}$ at the direction $\theta_d = 45°$ to remove the area outside the transit between $\mathbb{A}$ and $\mathbb{B}$.

Figure 6.39: Ambiguous relocation areas selection.

Removing $\mathbb{C}_1$ is not necessary since the computation of the set of directions $\Theta^{\mathbb{C}}_{\mathbb{AB}}$ linking $\mathbb{A}$ to $\mathbb{B}$ despite the presence of some relocation areas or obstacles $\mathbb{C}$ according to Equation 6.237 would not change the result. Indeed, the weak backprojection of $\mathbb{C}_1$ in this case according to the set of directions between $\mathbb{A}$ and $\mathbb{B}$ would be oriented toward the bottom left corner and wouldn't even cross $\mathbb{A}$. Therefore the relocation areas that are placed before $\mathbb{A}$ do not influence the result.

In some cases, the shapes $\mathbb{A}$ and $\mathbb{B}$ may be very complex and $\mathbb{C}$ cannot be removed by the red box as depicted in Figure 6.40. The set $\mathbb{C}$ will be considered in the local area (preprocessing step) and the method based on the forward reach set would easily remove $\mathbb{C}$ by considering the set of directions $\Theta_{\mathbb{AB}}$.

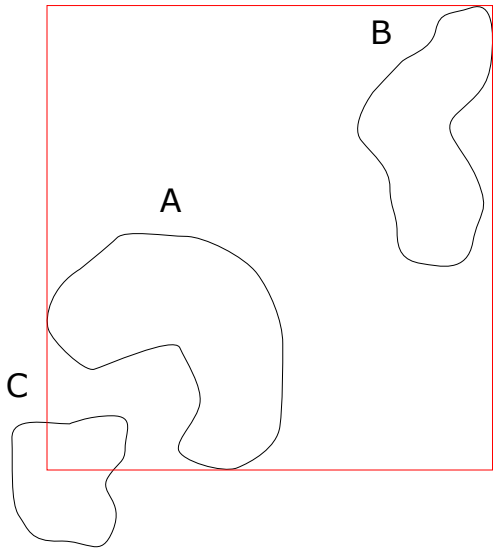


Figure 6.40: Complex situation for area removal.

The forward reach set of a set $\mathbb{A} \subset \mathbb{R}^2$ is defined as follows:

$$FORW(\mathbb{A}) = \{(x_1, x_2, \theta_d) | \exists \mathbf{a} \in \mathbb{A}, \exists \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \mathbf{f}_{H2}(\mathbf{x}_0, t, \mathbf{p})\} \tag{6.256}$$

where $\mathbf{x}_0 = (\mathbf{a}, \theta_d) = (a_1, a_2, \theta_d)$ only the function $\mathbf{f}_{H2}$ in the 2D space is used. If $\mathbb{A}$ is a subset of $\mathbb{R}^3$, then consider the projection in the $(x - y)$ plane.

The 2D forward reach set can then be defined as follows:

$$FORW(\mathbb{A}, \theta_d) = \{(x_1, x_2) | \exists \mathbf{a} \in \mathbb{A}, \exists \mathbf{p} \in [\mathbf{p}], \exists t \in \mathbb{R}^+, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \mathbf{f}_{H2}(\mathbf{x}_0, t, \mathbf{p})\} \tag{6.257}$$

where $\mathbf{x}_0 = (\mathbf{a}, \theta_d) = (a_1, a_2, \theta_d)$.

In this part, the uncertain parameter $\mathbf{p}$ is only described by an uncertainty δ_θ on the direction. The 2D forward reach set may reformulated as follows:

$$\begin{aligned} FORW(\mathbb{A}, \theta_d) &= \{(x_1, x_2) \in \mathbb{R}^2 | \exists \mathbf{a} \in \mathbb{A}, \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \mathbf{f}_{H2}(\mathbf{x}_0, t, \mathbf{p})\} \\ &= \{(x_1, x_2) \in \mathbb{R}^2 | \exists \mathbf{a} \in \mathbb{A}, \exists \delta_\theta \in [-\alpha_\theta, \alpha_\theta], \exists t \in \mathbb{R}^+, \\ &\quad \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + v \cdot t \cdot \begin{pmatrix} \cos(\theta_d + \delta_\theta) \\ \sin(\theta_d + \delta_\theta) \end{pmatrix}\} \qquad (6.258) \\ &= \mathbb{A} \oplus \mathbb{X}_{polar}(\theta_d) \qquad (6.259) \end{aligned}$$

where $\mathbb{X}_{polar}(\theta_d)$ was defined in Equation 6.127 when the weak backprojection was explained to take into account forbidden areas in the section 6.4.2.4.

In the provided example in Figure 6.39, and considering the set $\mathbb{C}_1$ which is actually not between $\mathbb{A}$ and $\mathbb{B}$ can be removed by finding the following set:

$$\mathbb{P}^{amb}_{\mathbb{AB}}(\mathbb{C}) = \{\mathbf{x} \in \mathbb{R}^2 | \bigcap_{\theta_d \in \Theta_{\mathbb{AB}}} (FORW(\mathbb{A}, \theta_d) \cap \mathbb{C})\} \tag{6.260}$$

where $\mathbb{C}$ is $\mathbb{C}_1$. $\mathbb{C}$ is assumed to be a subset of $\mathbb{R}^2$. If it is not the case, then consider the projection in the $(x-y)$ plane of $\mathbb{C}$.

If $\mathbb{P}^{amb}_{\mathbb{AB}}(\mathbb{C}) \neq \emptyset$, the set $\mathbb{C}$ intersects at least one forward reach set of $\mathbb{A}$ defined at a direction in the set of directions linking $\mathbb{A}$ to $\mathbb{B}$, and then may be ambiguous. It can be potentially used at the end as it is not necessary to remove the sets that are before $\mathbb{A}$ and not between $\mathbb{A}$ and $\mathbb{B}$. The final result is given in Figure 6.41 where the set $\mathbb{C}_1$ has been removed from the ambiguous areas. Finally, the variable *localAmbiguousArea* would only contain $\mathbb{C}_3$.

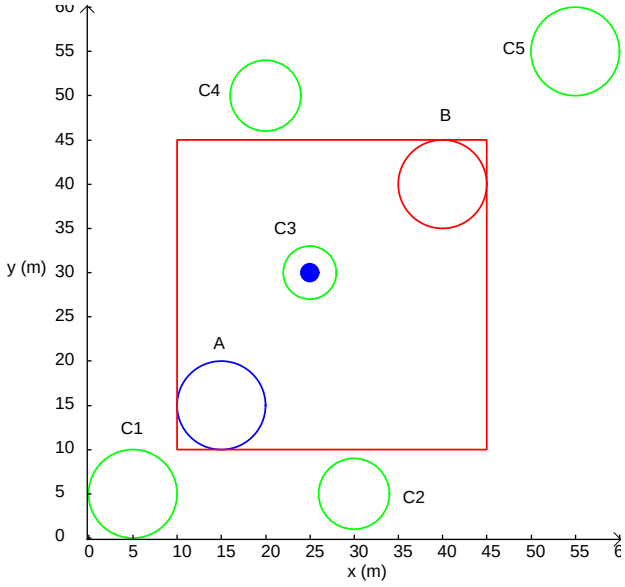


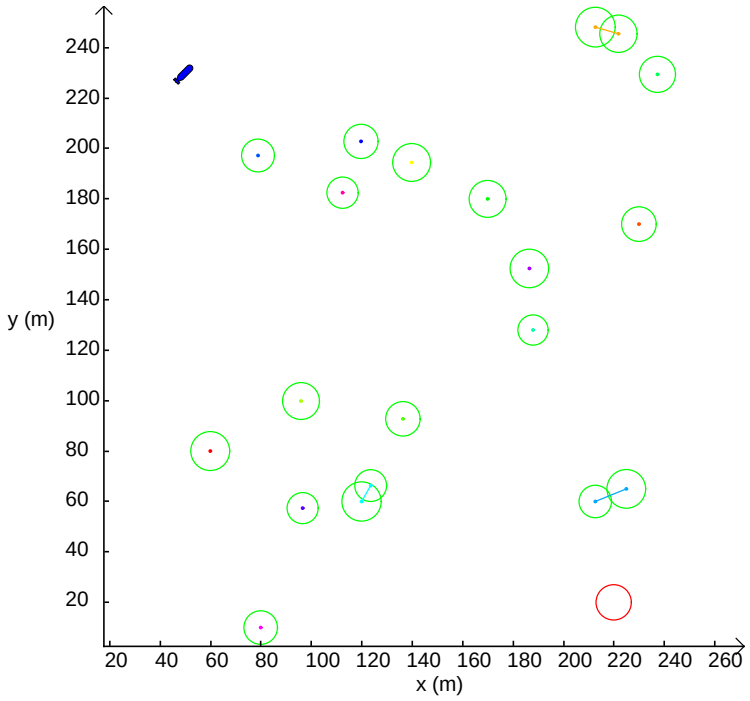
Figure 6.41: Final selection of ambiguous area with the forward reach set.

Example of graph building

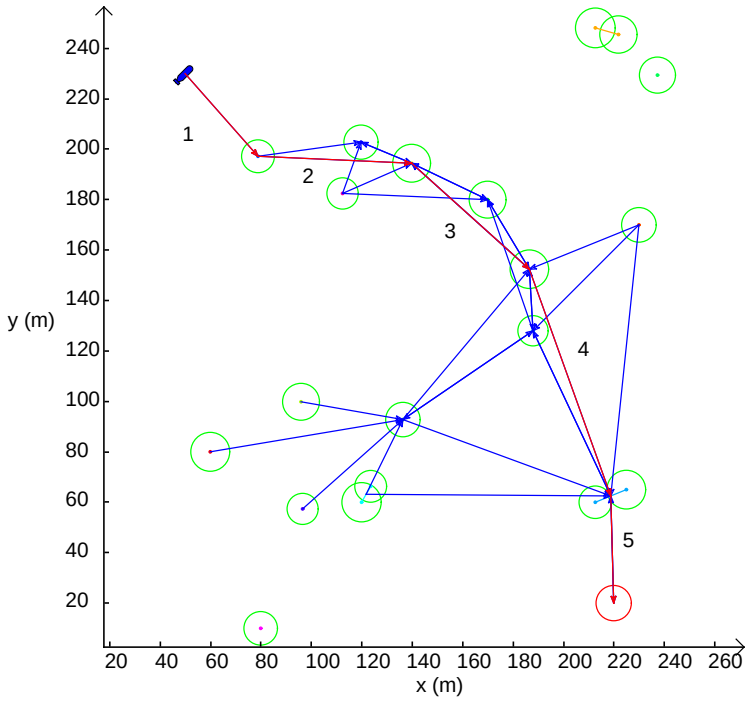
Consider the environment depicted in Figure 6.42(a) where the goal area is represented by a red circle and the initial position of the AUV by the blue AUV in the top right corner. 20 relocation areas are present in green and represented by disks with a radius varying between 6 and $8m$. The motion planning is proposed with disks since it is quite easy to generate random environment. As it may be noticed, some relocation areas intersect another relocation areas: cluster are then formed. 3 clusters of 2 relocation areas are represented where a link between the relocation areas is shown. Since clusters are seen as relocation areas, consequently only 17 relocation areas are present including 3 clusters. To reduce the complexity of the graph, only the 5 nearest neighbours in terms of distance between the centers or the centroids are considered. Therefore, the set of possible nodes $\mathbb{V}$ connected to $\mathbb{U}$ is only composed of 5 relocation areas.

The graph expansion is represented in Figure 6.42(b) where blue arrows are guaranteed links despite the presence of other relocation areas, the origin of an arrow indicates the departure relocation area. A blue arrow confirms then the possibility to navigate without ambiguity between two relocation areas. The path found by Algorirthm 10 is represented by the red path

composed of arrows that is optimized in distance. A number has been given for each part of the motion and some of them will be more detailed in Figure 6.43.



(a) Environment with a goal area in red, a starting position shown by the blue AUV and some relocation areas in green. Some relocation areas forms clusters as the one just above the goal area.



(b) Links between relocation areas without ambiguity in blue and the path found in red.

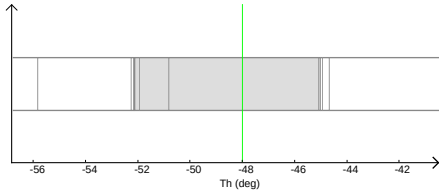
Figure 6.42: Environment and the path found with Algorithm 10.

The proof of the path 1 is provided in Figure 6.43(a) where $\Theta_{\mathbb{IU}}$ is computed since any another relocation areas are present between them. The 2D backward reach at $\theta_d = -48°$, indicated by the green line, is given in Figure 6.43(b). Indeed the initial position of the AUV belongs to the 2D backward reach set at this direction.

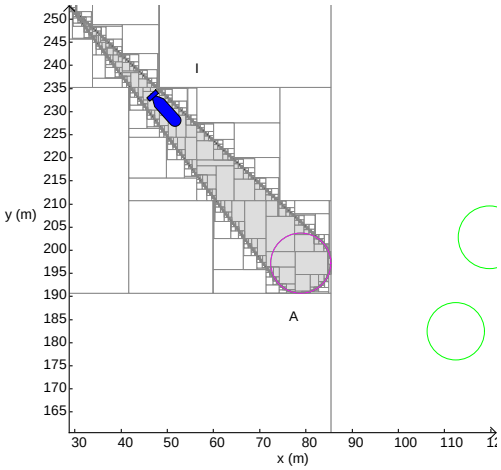
The proof of the path 2 is given in Figure 6.43(c) where the set of directions between the departure relocation area $\mathbb{A}$ in blue and the intermediate goal area $\mathbb{B}$ in blue is computed by taking into account the two local relocation areas $\mathbb{C}_1$ and $\mathbb{C}_2$ between them. Please see Figure 6.43(d) for the names. The corresponding 2D backward reach set taking into these other relocation areas at the direction $\theta_d = 0°$ (at the green line) is given in Figure 6.43(d) where a possible departure position is represented by the red AUV. Notice that indeed the weak backprojection of an obstacle reduces the departure area at this direction.

The proof of the path 4 is given in Figure 6.43(e) where the robot has to reach a cluster $\mathbb{B}$ in magenta despite the presence of one another relocation area $\mathbb{C}$ in black. Please see Figure 6.43(f) for the names. The corresponding 2D backward reach set at the direction $\theta_d = -72°$ is given in Figure 6.43(f). From the possible departure position represented by the red AUV, it could reach one of the two relocation areas in the cluster. Since the AUV cannot distinguish between the two relocation areas, a local strategy to disambiguate the true location of the robot has to be performed before going to the departure area of the last portion of the high-level strategy.

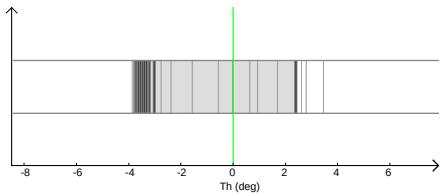
The union of the 2D backward reach sets taking into account possible another relocation areas of the strategy found is proposed in Figure 6.44(a). It includes the three parts of the strategy proposed in Figure 6.43. One direction has then been selected for each part of the strategy according to the computed sets to propose this complete 2D backward reach set. A possible path execution is proposed in Figure 6.44(b) where the yellow lines correspond to the uncertain trajectories according to the uncertain parameter $\delta_\theta \in [-\alpha_\theta, \alpha_\theta]$ on the direction selected. The different red AUVs corresponds to possible departure positions according to the different directions selected in the strategy. Magenta paths are the relocation process inside a relocation area since the robot is able to relocate and move perfectly. As it can be noticed the AUV can reach the goal area in red in a guaranteed manner.



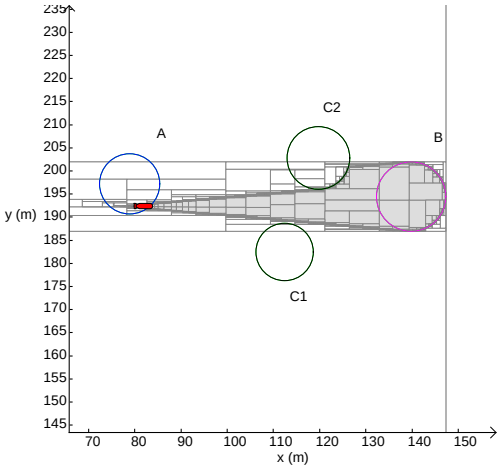
(a) Set of direction $\Theta_{\mathbb{IA}}$ (see figure (b)).



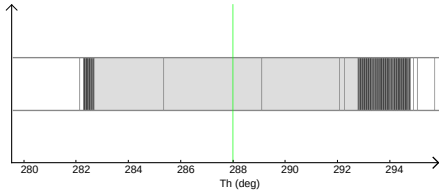
(b) 2D corresponding backward reach set at $\theta_d = -48°$.



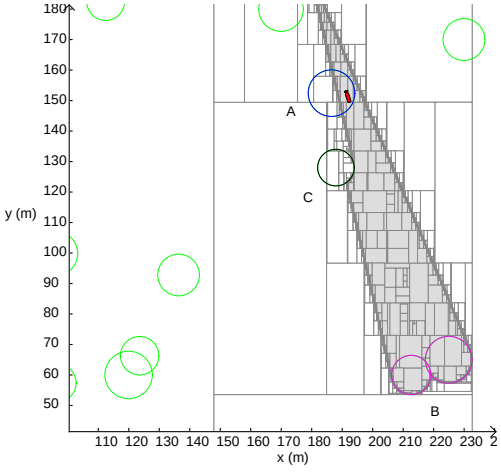
(c) Set of direction $\Theta_{\mathbb{AB}}^{\mathbb{C}_1 \cup \mathbb{C}_2}$ (see figure (d)).



(d) 2D corresponding backward reach set at $\theta_d = 0°$ taking into account $\mathbb{C}_1$ and $\mathbb{C}_2$.

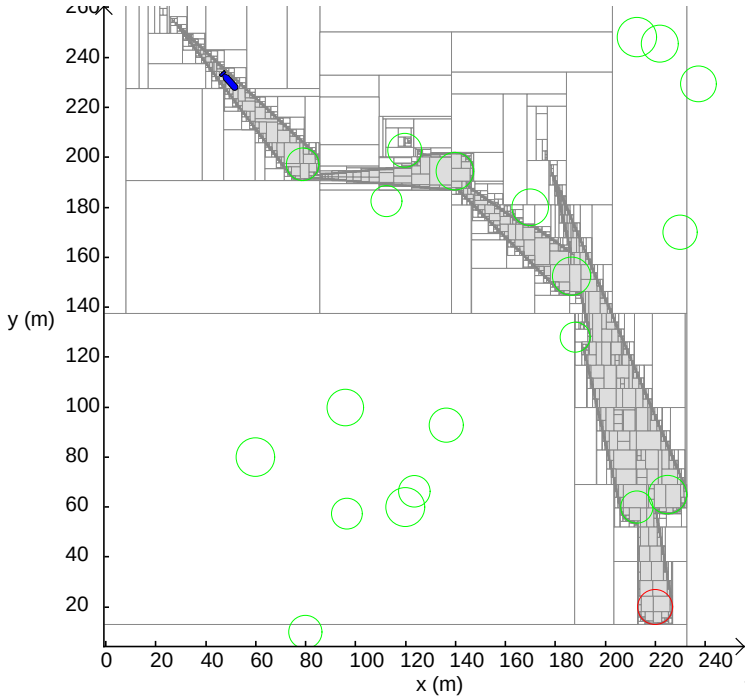


(e) Set of direction $\Theta_{\mathbb{AB}}^{\mathbb{C}}$ (see figure (f)).

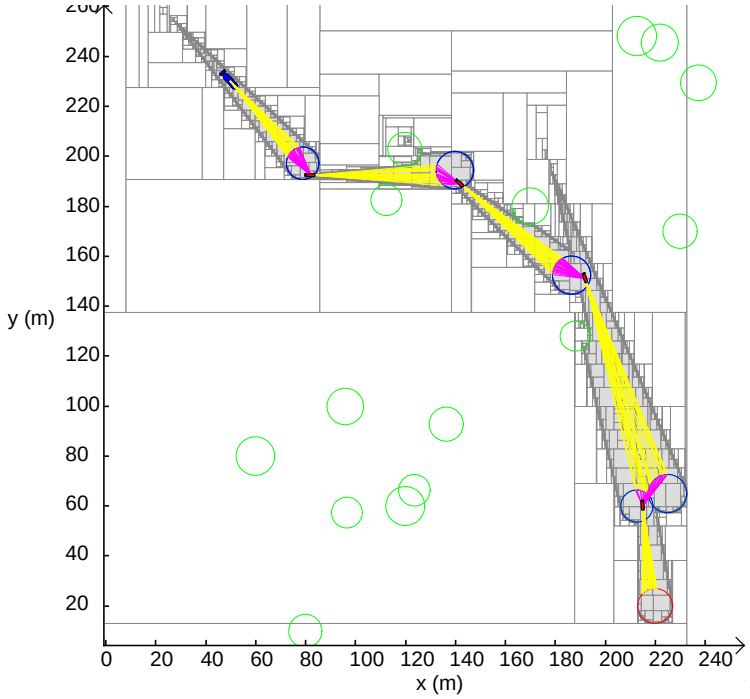


(f) 2D corresponding backward reach set at $\theta_d = -72°$ taking into account $\mathbb{C}$.

Figure 6.43: Intermediate reachability between the blue departure area and the intermediate goal area in magenta taking into account potential another relocation areas in black.



(a) Union of the 2D backward reach sets at selected directions θ_d to navigate safely between the relocation areas in the strategy.



(b) Path execution with uncertain trajectories in yellow and relocation process in magenta inside relocation areas with perfect moves and positioning.

Figure 6.44: Environment and the path found with Algorithm 10.

Simplified version for fast computation

Since it may be time consuming to prove the existence of at least one direction using the **SIVIA** algorithm with projection between two relocation areas despite the presence of another possible relocation areas, a simplified version based on a discrete set of possible directions

taking into account obviously the possible another relocation areas to avoid any ambiguity is proposed briefly.

The *connect* function at line 29 in Algorithm 10 between a departure relocation area $\mathbb{A}$ and an intermediate goal area $\mathbb{B}$ does not consist anymore to prove that $\Theta^{\mathbb{C}}_{\mathbb{AB}}$ is not empty despite the presence of local obstacles or another relocation areas. A discrete set of possible directions denoted as $\{\Theta_{\mathbb{AB}}\}$ between $\mathbb{A}$ and $\mathbb{B}$ are computed. Consider the following set at a possible direction θ_i inside $\{\Theta_{\mathbb{AB}}\}$:

$$\mathbb{P}^{\mathbb{C}}_{\mathbb{AB}}(\theta_i) = \{\mathbf{x} \in \mathbb{R}^2 | BACK(\mathbb{B}, \theta_i) \cap \overline{WBACK(\mathbb{C}, \theta_i)} \cap \mathbb{A}\} \tag{6.261}$$

where $\mathbb{A}$ is a subset of $\mathbb{R}^2$. If it is not the case, then consider the projection in the $(x-y)$ plane. $\mathbb{C}$ is the local ambiguous relocation areas or obstacles according to the method proposed above in the paragraph **Selecting local ambiguous area**.

Then consider the global set:

$$\mathbb{P}^{\mathbb{C}}_{\mathbb{AB}}(\{\Theta_{\mathbb{AB}}\}) = \bigcup_{\theta_i \in \{\Theta_{\mathbb{AB}}\}} \mathbb{P}^{\mathbb{C}}_{\mathbb{AB}}(\theta_i) \tag{6.262}$$

If $\mathbb{P}^{\mathbb{C}}_{\mathbb{AB}}(\{\Theta_{\mathbb{AB}}\})$ is not empty, then a least one direction exists such that it is possible to leave $\mathbb{A}$ and reach $\mathbb{B}$ without ambiguity.

The connection between two relocation areas depends then on the discretization of the possible directions. The advantage is the fast computation of a potential graph. However it is not anymore a complete planner.

The planner proposed based on the **SIVIA** computation depends on a epsilon ϵ parameter then cannot be considered as a complete planner but as almost a complete planner. Moreover, it relies on projection where a parameter has to be defined too. However it proves the guarantee of the solution.

6.6 Planning with registration maps

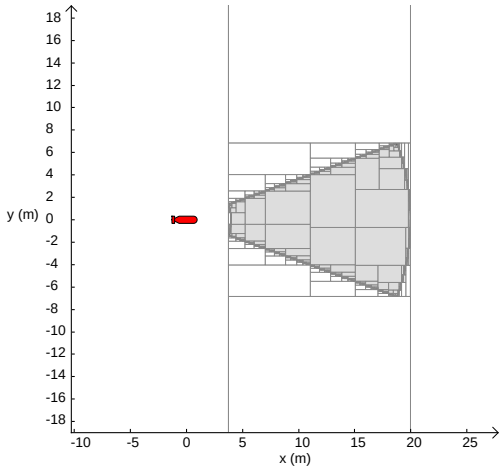
The motion planning problem was until now proposed only on disks or shapes that are subsets of $\mathbb{R}^2$ but the method can be extended to 3D. Two motion planners were proposed to deal with distinguishable and indistinguishable relocation areas. In this part, the strategy finding is proposed with the registration maps developed in Chapter 5. The planners are then proposed only for the holonomic vehicle using directional/omnidirectional backprojection. It will be assumed that the robot is perfectly located in a relocation area and can move anywhere without constraints on motion in a relocation area. It has been shown how to deal with uncertain location in relocation.

6.6.1 Planning without point of view

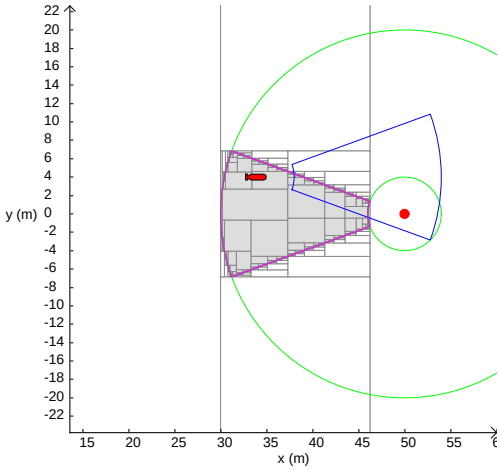
In this first subsection, it is assumed that the landmarks are detectable at any point of view. Remind that the uncertain parameters were $\alpha_\theta = 5^\circ$ and $\alpha_v = 10\%$. Illustrations will be based mainly on punctual landmarks.

Backward reach set with a single landmark

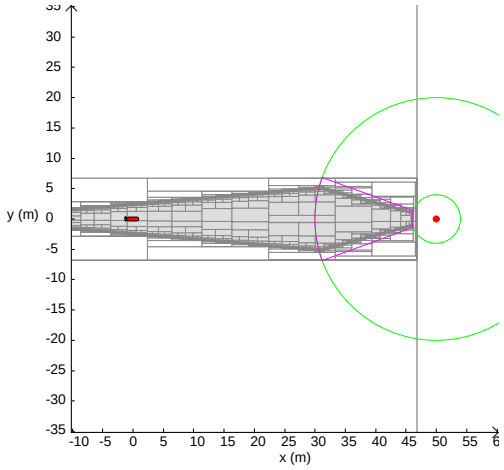
Consider a forward looking sensor as depicted in Figure 6.45(a) with a range of detection between 4 and $20m$ and a whole aperture angle of 40° ($\theta_3 = 20^\circ$). Consider a punctual landmark located at $(50, 0)$ where the registration map at the heading 0° is shown in Figure 6.45(b) computed according to Equation 5.14 (see section **Punctual landmark**). Magenta pie encloses the solution set (polar set). The 3D registration map (x, y, θ) is called $\mathbb{A} \subset \mathbb{R}^3$. Using the motion model $\mathbf{f}_H$ in Equation 6.14 for a directional robot including the direction and not $\mathbf{f}_{H2}$, the 2D backward reach set at the direction $\theta_d = 0^\circ$ ($BACK(\mathbb{A}, 0^\circ)$) is given in Figure 6.45(c). Notice that the 2D backward reach set at $\theta_d = 0^\circ$ does not correspond to the 2D backward reach set of the 2D registration map at the heading 0° since the direction of the AUV is uncertain ($\theta_{AUV} \in [\theta_d - \alpha_\theta, \theta_d + \alpha_\theta] = [-5, 5](deg)$). The magenta pie is larger the 2D backward reach set in the field of detection of the landmark. Some trajectories from an initial location at $(0, 0)$ at the command direction $\theta_d = 0^\circ$ are drawn in yellow and stop when it detects the punctual landmark at the blue AUV pose with the associated visibility area. Notice that indeed all the trajectories are able to detect with this sensor the landmark from this initial pose.



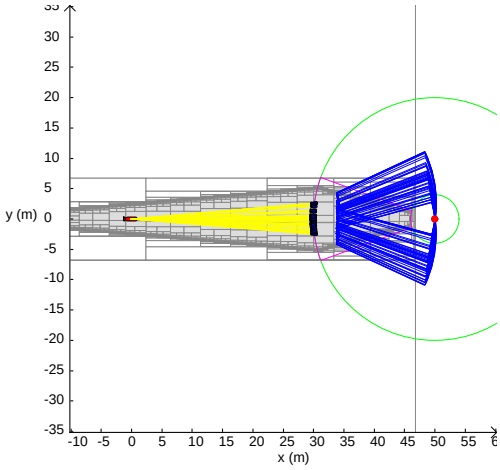
(a) Visibility area of the sensor at the pose $(0, 0, 0°)$.



(b) Registration map at the direction $\theta = 0°$ of the landmark. Magenta pie encloses the solution set.



(c) 2D backward reach at $\theta_d = 0°$ set that guarantees the detection of the punctual landmark.



(d) Some trajectories from the initial location $(0, 0)$ at the desired direction $\theta_d = 0°$. Blue AUVs indicate the pose when it detects the landmark. The visibility area is associated to the detection pose.

Figure 6.45: Reachability of a punctual landmark with a forward looking sensor.

When registration maps are considered as connected sets ?

In the motion planners proposed, the notion of connected sets was mainly proposed when 2D shape intersects. Now it has to be adapted to the registration maps which are subsets of $\mathbb{R}^3$. For ease of simplication and understanding, only punctual landmarks will be considered. The idea is to prove when two registration maps can be considered as connected.

Consider a first punctual landmark "a", called reference landmark, located at $\mathbf{a} = (50, 0)$ and a second punctual landmark "b" located at $\mathbf{b} = (50, y_b)$ which is at the vertical of the reference landmark. A parameter y_b is defined to move this second landmark according to the reference landmark. By considering the sensor with the visibility area presented in Figure 6.45(a), the two registration maps $\mathbb{A}$ and $\mathbb{B}$ are computed according respectively to the two landmark positions $\mathbf{a}$ and $\mathbf{b}$ (a value for y_b is then defined). The reader may refer to Chapter 5 how to compute the registration map for a punctual landmark. They are subsets of $\mathbb{R}^3$ since the (x, y) location

and the heading of the AUV θ is important due to the limited aperture angle and limited range.

Consider again the intersection of the registration maps $\mathbb{A}$ and $\mathbb{B}$ which is given by the set:

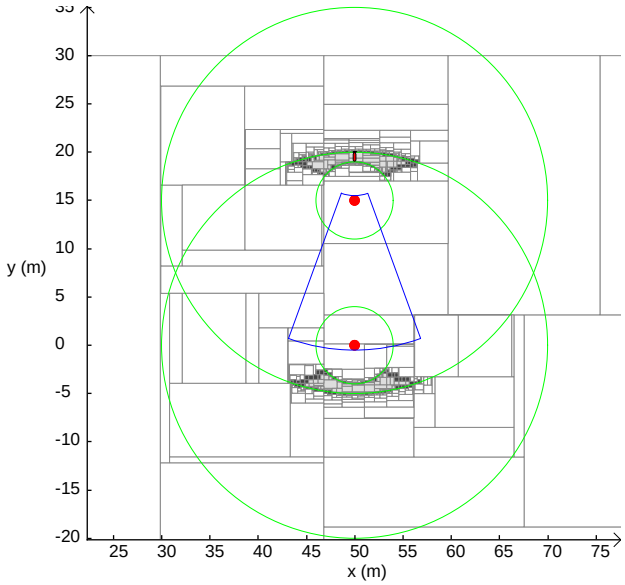
$$\mathbb{P}_{\mathbb{AB}} = \{\mathbf{x} \in \mathbb{R}^3 | \mathbb{A} \cap \mathbb{B}\} \tag{6.263}$$

Since it is a subset of $\mathbb{R}^3$, a 2D representation (in the (x, y) plane) can be given as follows:

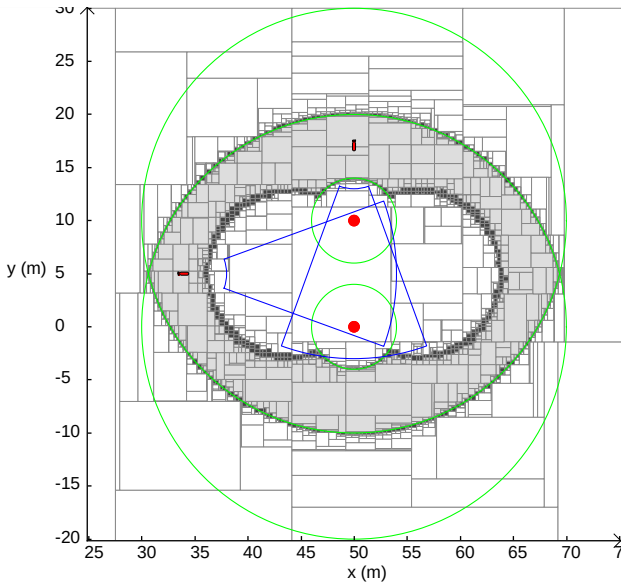
$$\mathbb{P}^{2D}_{\mathbb{AB}} = proj^{\mathbb{X}_3}_{\mathbb{X}_1 \times \mathbb{X}_2}(\mathbb{P}_{\mathbb{AB}}) \tag{6.264}$$

where $\mathbb{X}_1$ is the domain of x, $\mathbb{X}_2$ is the domain of y and $\mathbb{X}_3$ is the domain of θ (heading).

Considering the two registration maps $\mathbb{A}$ and $\mathbb{B}$, Figure 6.46 shows the results of Equation 6.264 where $\mathbb{X}_3$ is defined by the interval $[-180, 180](deg)$. The position of the landmarks are drawn for the two values of y_b by red dots. The green circles are the minimal and maximal ranges of detection of detection according to the landmark positions. Some poses are represented red AUVs with the associated visibility area of the sensor. In Figure 6.46(a), y_b is defined at $15m$ and it can be noticed that the AUV is not able to detect at every poses around the two landmarks at the same time. In Figure 6.46(b) where $y_b = 10m$, the robot is able all around the cluster of landmarks according to its visibility area to detect both at the same time.



(a) $\mathbb{P}^{2D}_{\mathbb{AB}}$ with $y_b = 15m$. The two registration maps are not connected.

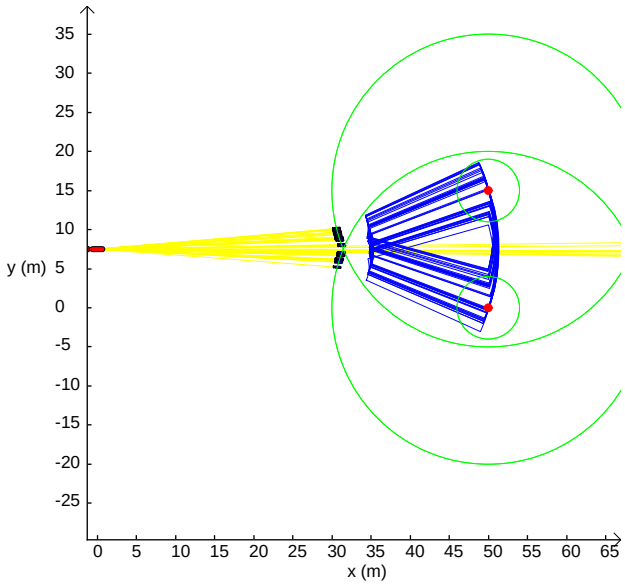


(b) $\mathbb{P}^{2D}_{\mathbb{AB}}$ with $y_b = 10m$. The two registration maps are connected.

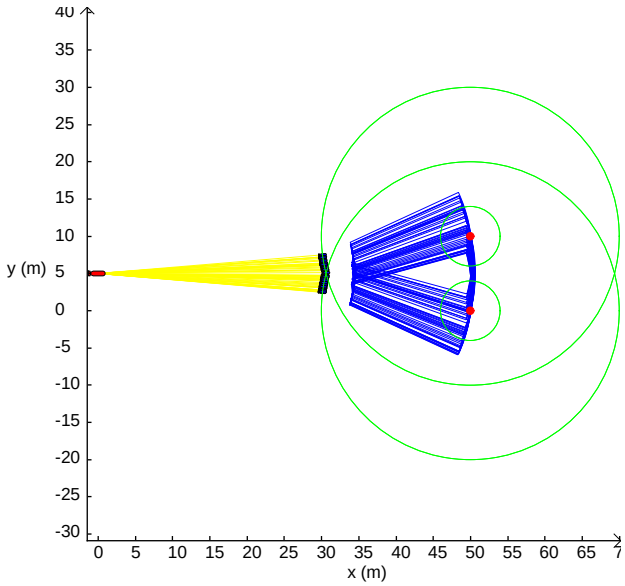
Figure 6.46: Intersection of the registration maps projected on the ground according to Equation 6.264. Some poses are represented by red AUV poses whit the associated visibility area in blue, the punctual landmarks are shown by red dots, the minimum and maximum ranges for each landmark are represented by green circles.

In the first case ($y_b = 15m$), the two registration maps cannot be considered as connected sets contrary to the second case ($y_b = 10m$) where it exists at least one heading all around the cluster to detect both at the same time. This is illustrated in Figure 6.47 where the reachability of both landmarks is tested from an initial pose of the robot defined at the worst case considering the two landmarks. This worst case is defined as follows: the direction of the motion is perpendicularly to the line between the two landmarks, i.e. $\theta_d = 0°$ (in reality $\theta_d = k\pi$ with $k \in \mathbb{Z}$) and y_0 is defined at the middle of the two punctual landmarks ($\frac{y_b}{2}$). It corresponds then to the line between the two landmarks where the initial x_0 was defined on this line at $0m$ for example to have the heading $\theta_d = 0°$ of the motion.Consequently the initial pose is $\mathbf{x}_0 = (0, 7.5, 0°)$

and the direction to follow is $\theta_d = 0°$. In Figure 6.47(a) with $y_b = 15m$, trajectories from the initial pose are drawn in yellow according to the uncertain direction ($\delta_\theta \in [-\alpha_\theta, \alpha_\theta]$). When the pose of a robot is able to detect one of the two landmark, i.e. the pose is in the union of the registration maps, an AUV is drawn is blue with the associated visibility area in blue, and the path stops as a landmark is detected. As it may be noticed some trajectories go through the field of influence of the two landmarks without detecting any of them (drawn by the long yellow line). On the contrary, considering $y_b = 10m$ in Figure 6.47(b) and then $\mathbf{x}_0 = (0, 5, 0°)$ with the same direction $\theta_d = 0°$ (worst case), all the trajectories in yellow are able to detect at least one landmark.



(a) Reachability with $y_b = 15m$. Some trajectories cannot detect any landmark.



(b) Reachability with $y_b = 10m$. All the trajectories can detect at least one landmark.

Figure 6.47: Reachability of a two registration maps associated to the two punctual landmarks with different y_b. Yellow lines are trajectories, the red AUV is the initial pose, the motion direction is $\theta_d = 0°$, little blue AUVs with their associated visibility area (blue pie) are the poses that can detect at least one landmark.

It is possible to determine y_b, i.e the set of distances between the landmarks, to prove that the registration maps are connected or not. This is illustrated in Figure 6.48 where "a" is the reference landmark and "b" can move according to the parameter d.

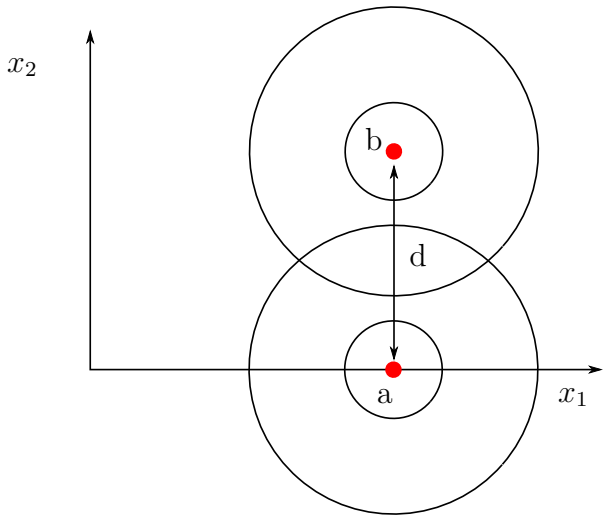


Figure 6.48: Distance determination to consider connected registration maps. d is the parameter to determine, "a" is the reference landmark and "b" the moving landmark.

Since the critical direction of the motion in this situation is $\theta_d = 0°$, consider then the two registration maps at the defined heading $\theta = \theta_d = 0°$.

Consider the following function (please refer to Chapter 5):

$$\mathbf{f_b} : \begin{cases} \mathbb{R}^2 \times \mathbf{x} & \rightarrow & \mathbb{R}^2 \\ (p_1, p_2) & \rightarrow & \mathcal{R}^T(p_3) \begin{pmatrix} b_1 - p_1 \\ b_2 - p_2 \end{pmatrix} \end{cases} \tag{6.265}$$

where $\mathbf{p}$ corresponds to the registration map for the punctual landmark $\mathbf{b}$. This function was denoted as $\mathbf{f_b^{-1}}$ in Chapter 5. p_3 is defined at the heading $\theta = 0°$ and $\mathbf{b}$ is the position of the second landmark. The visibility area $\mathbb{V}$ is defined at the particular pose $(0, 0, 0°)$ which leads to the registration map:

$$\mathbb{P}_{\mathbf{b}}^{2D} = \mathbf{f_b^{-1}}(\mathbb{V}) \tag{6.266}$$

Since it depends on the reference landmark, then consider the function:

$$\mathbf{f_a} : \begin{cases} \mathbb{R} \times \mathbb{R}^2 & \rightarrow & \mathbb{R}^2 \\ (d, p_1, p_2) & \rightarrow & \mathcal{R}^T(p_3) \begin{pmatrix} a_1 - p_1 \\ a_2 - p_2 + d \end{pmatrix} \end{cases} \tag{6.267}$$

where d is actually y_b since $a_2 = 0$.

Considering the visibility area $\mathbb{V}$ defined at the particular pose $(0, 0, 0°)$ where the heading is defined ($p_3 = 0°$), the set of robot position that can detect the landmark including the parameter d is given by:

$$\mathbb{P}(d) = \mathbf{f_a^{-1}}(\mathbb{V}) \tag{6.268}$$

Finally the set of distances is given by:

$$\mathbb{D} = \{d \in \mathbb{R} | \mathbb{P}(d) \cap (\mathbb{R} \times \mathbb{A}(0°))\} \tag{6.269}$$

where $\mathbb{A}(0°)$ is the registration map of the reference landmark "a" at the defined heading $\theta = 0°$. $\mathbb{R}$ has been added since the dimension of $\mathbb{P}(d)$ is 3 and $\mathbb{A}(0°)$ is only 2.

The solution of this set is given in Figure 6.49 where it can indeed be noticed that the distance $d = y_b = 15m$ is outside (red line) and then the two registration map cannot be considered as connected contrary the distance $d = y_b = 10m$ (green line). These distances were used in Figure 6.47 but now the set of distances is valid. By taking the inner approximation, it can be concluded according to the parameters of the projection that two registration maps involving punctual landmarks are connected if the distance d is below than $13.67m$ (approximately), otherwise some trajectories may be "lost". d_{max} is then defined at $13.67m$.

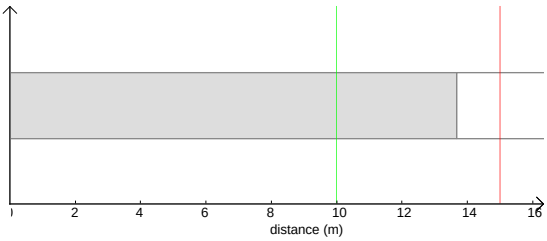


Figure 6.49: Set of distance $\mathbb{D}$ between the two landmarks to consider the registration maps as connected. The green line corresponds to situation in Figure 6.47(b) and the red line to the situation in Figure 6.47(a).

The problem could be solved if the position of the punctual landmarks was uncertain too. Since it is assumed now that the environment is certainly located, if two punctual landmarks are closer than $13.67m$ then the associated registration maps are considered as connected sets.

The value of d_{max} can be analytically obtained considering the sensor and the parameters that define this sensor as depicted in Figure 6.50. The maximum distance can then be obtained as follows:

$$d_{max} = 2 \cdot R_{max} \cdot \sin(\theta_3) \tag{6.270}$$

which gives with the defined parameters $d_{max} = 13.68m$ which is the same value as obtained before.

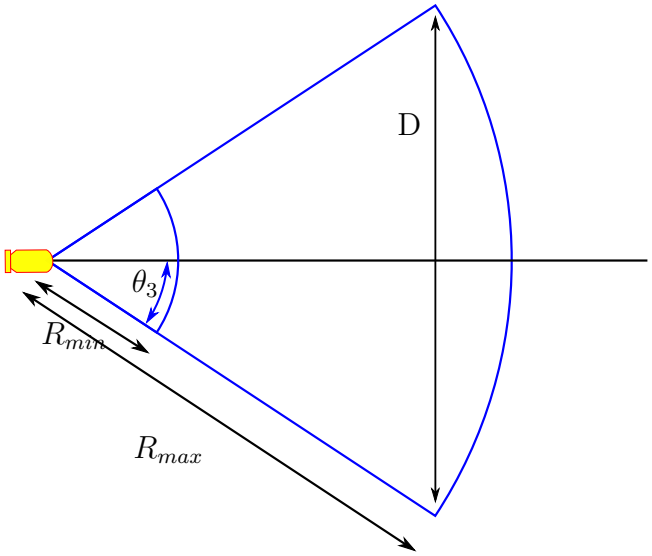


Figure 6.50: Simplified determination of an under approximation of d_{max}.

This method can be extended to any landmark by dilating one landmark by d_{max} and looking for intersection with another landmark. If the intersection is not empty, then the registration

maps of the two considered landmarks (the one dilated and the classic one) can be considered connected since it is impossible for the robot to navigate between them without detecting at least one of them. If the point of view was included, this proposition may not be true.

Connection between registration maps

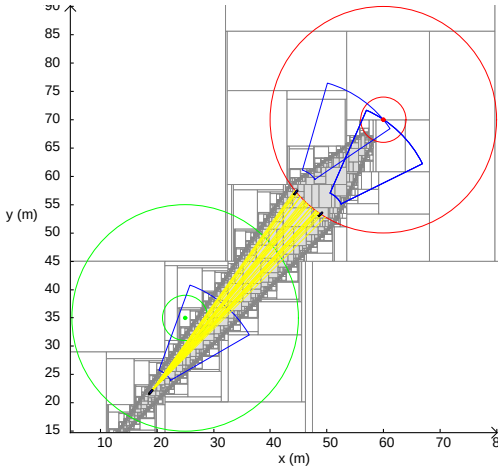
It is assumed no uncertainty in position in a relocation area when detecting the landmark. See the previous section to take into account possible uncertainty which is the same every where in the landmark area. This is discussed in the section 7.2(Perspectives).

Proving a connection is still provided by Equation 6.55 where $\mathbb{B}$ is now the registration map to reach and $\mathbb{A}$ is the registration map from where the robot starts its motion.

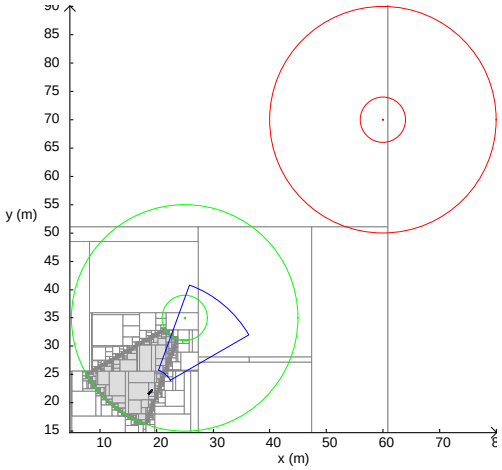
Two methods can be envisaged and will be explained through an example.

Consider for example a punctual landmark to reach located at $(60, 70)$ (red dot) and a landmark from where the robot starts its motion located at $(25, 35)$ (green dot) as depicted in Figure 6.51. Consider for a example the direction $\theta_d = 50°$. The 2D backward reachability of the goal registration map is given in Figure 6.51(a) where some trajectories are plotted from an initial position that is able to detect the departure landmark (green). All the trajectories are able to detect the goal landmark.

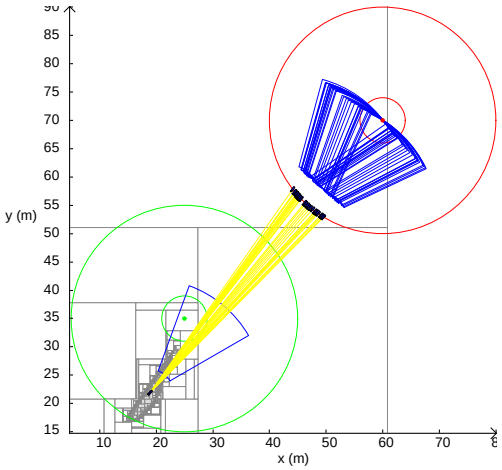
The first method is to start from a position that is able to detect the departure landmark. The registration map at this direction $\theta = 50°$ is given in Figure 6.51(b) and finally the departure area at this direction is given in Figure 6.51(c) to redetect for sure the goal landmark. This departure area indicates the set of robot positions such that it is able to detect the departure landmark for the desired direction $\theta_d = 50°$ and that is able to prove the detection of the goal landmark after the uncertain motion, i.e. after leaving the detection of the departure landmark.



(a) 2D backward reachability of the registration map associated to the goal landmark (red) at the defined direction $\theta_d = 50°$.



(b) Registration map of the departure landmark at the direction $\theta = 50°$.



(c) Intersection between the 2D backward reachability and the registration map of the departure landmark at the direction $\theta = 50°$. Departure area to guarantee the reachability at $\theta_d = 50°$.

Figure 6.51: First method of the reachability between two registration maps. Punctual landmarks are considered in this example. The green landmark is the departure landmark and the red one is the goal landmark.

Due to the motion ability of the robot to turn on itself or having a little turning radius, the second method consists simply by starting from a position that is in the field of influence of the departure landmark and is able to reach the desired landmark at the defined direction ($\theta_d = 50°$ in this case). This set of position is simply given by the following equation:

$$\mathbb{P}_{\mathbb{AB}}(\theta_d) = \{(x_1, x_2) \in \mathbb{R}^2 | BACK(\mathbb{B}, \theta_d) \cap proj^{\mathbb{X}_3}_{\mathbb{X}_1 \times \mathbb{X}_2}(\mathbb{A})\} \tag{6.271}$$

where θ_d is the desired direction for the reachability of $\mathbb{B}$, $\mathbb{A}$ is the registration map of the departure landmark and $\mathbb{B}$ is the registration map of the goal landmark.

According to the sensor proposed here, the term $proj^{\mathbb{X}_3}_{\mathbb{X}_1 \times \mathbb{X}_2}(\mathbb{A})$ corresponds simply to the set of robot positions by removing the constraint on the heading. Therefore, only the constraint on

the range is used to compute the proposed and find the departure area at a desired direction. The registration map of the departure landmark is then simply a subset of $\mathbb{R}^2$.

Define then:

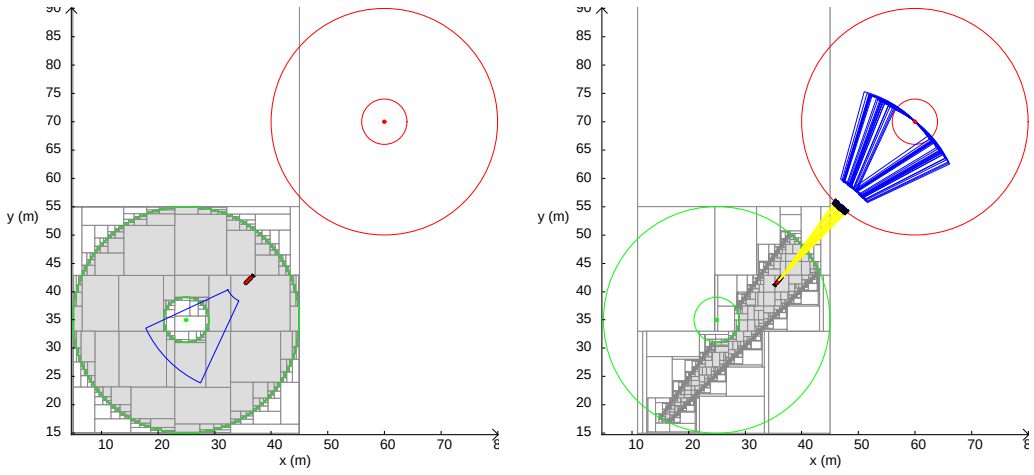
$$\mathbb{A}_{2D} = proj^{\mathbb{X}_3}_{\mathbb{X}_1 \times \mathbb{X}_2}(\mathbb{A}) \tag{6.272}$$

which can be computed by considering the visibility area defined in the subsection 5.2.1.1 about the isotropic sensor with limited ranges and by considering a transformation function $\mathbf{f}$ only composed of a translation. Considering the same sensor, this set is provided in Figure 6.52(a).

Equation 6.271 can be rewritten as follows:

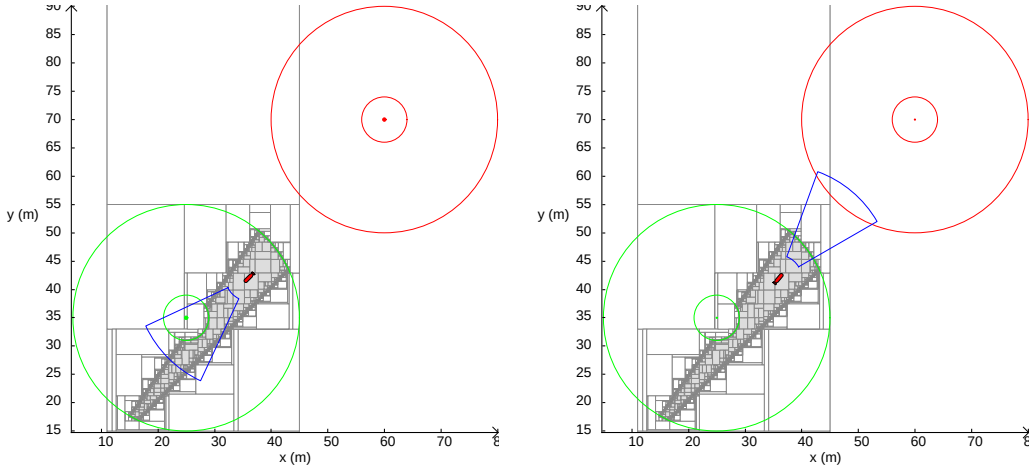
$$\mathbb{P}_{\mathbb{AB}}(\theta_d) = \{(x_1, x_2) \in \mathbb{R}^2 | BACK(\mathbb{B}, \theta_d) \cap \mathbb{A}_{2D}\} \tag{6.273}$$

The second method consists then by simply going in the area of the departure landmark given by Equation 6.273, depicted in Figure 6.52(b), detecting the landmark, as represented in Figure 6.52(c) and then orienting the head at the defined direction θ_d that guarantees the reachability of the goal landmark ($\theta_d = 50°$) as shown in Figure 6.52(d). The departure landmark might not be in the field of view of the sensor when it will start its motion, as shown in Figure 6.52(d), under the assumption that the rotation on itself does not change its position. An uncertainty on the position after the rotation can be added by using the theory developed before with uncertain initial position.



(a) 2D registration map $\mathbb{A}_{2D}$.

(b) Set of departure positions $\mathbb{P}_{\mathbb{AB}}(\theta_d)$ with $\theta_d = 50°$.

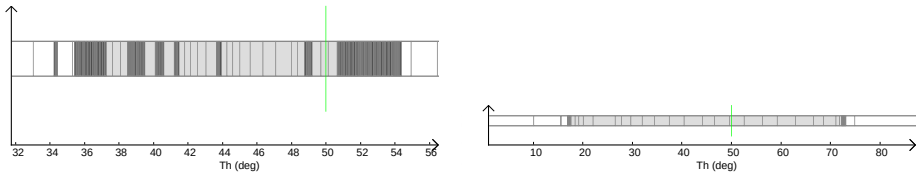


(c) Position before the rotation on itself, the landmark is detected.

(d) Position after the rotation at the desired direction $\theta_d = 50°$. Some trajectories are plotted in the figure (b).

Figure 6.52: Second method of the reachability between two registration maps based on the rotation ability of the robot on itself. The green landmark is the departure landmark and the red one is the goal landmark.

For both methods, it is possible to compute the set of directions $\Theta_{\mathbb{AB}}$, presented in Equation 6.64, that links the two registration maps $\mathbb{A}$ and $\mathbb{B}$ by using the projection as presented earlier. The results are given in Figure 6.53 where the set of directions using the first method is shown in Figure 6.53(a) and the one using the second method is represented in Figure 6.53(b). Notice that in the first case, the set of directions is hard to determine due to the projection in the set of directions with respect to the (x, y) dimension. Moreover, the set of directions in the second case is obviously larger than in the first case, since the robot can start from anywhere in the 2D registration $\mathbb{A}_{2D}$ shown in Figure 6.52(a).



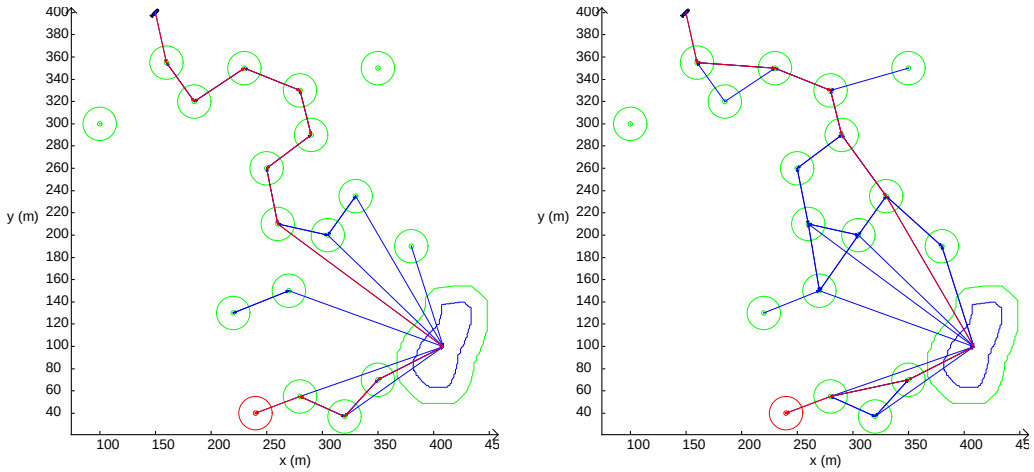
(a) Starting pose in $\mathbb{A}$, meaning the robot detects the departure landmark when it starts its motion.

(b) Starting position anywhere in $\mathbb{A}_{2D}$. The robot may not detect the departure landmark when it starts its motion.

Figure 6.53: $\Theta_{\mathbb{AB}}$ using both methods for the departure area. The selected direction ($\theta_d = 50°$) is the one that was used for the previous figures (green line).

Example

A simple environment is depicted in Figure 6.54 where many punctuals landmarks are represented by green dots and a shape landmark by a blue shape. A punctual target in red is at the bottom of the environment. The initial position of the AUV is at the top. It is assumed a forward-looking sensor that can detect between $R_{min} = 2m$ and $R_{max} = 15m$ and has an aperture angle $\theta_3 = 20°$. The whole aperture angle is then $2\theta_3 = 40°$. The error on the compass is fixed at 5°. The green and red circles are the visibility of the punctual landmarks and the assumed punctual targets at the minimal R_{min} and maximal R_{max} range of the sensor. The green shaped contour represents the visibility of the shape at R_{max}. The 2D areas correspond to the projection on the ground of the different registration maps. 6.54 shows the graph built for the two different methods of departure positions taking into account local ambiguous landmarks. It is assumed indistinguishable landmarks. The first method, i.e. starting from a position that can detect the departure landmark to reach another landmark, gives the graph and path in red in Figure 6.54(a) where blue links are other links built by the backward search. The result with the second method, i.e. is provided in Figure 6.54(b). Notice that the path is shorter with the second method since the robot is able to turn on itself before starting an uncertain motion to reach another landmark. It is assumed that the turn on itself does not increase the uncertainty on the robot position, i.e. the robot starts from a certain position. Uncertainty can be obviously added as shown before. The first method requires closer landmark positions since it starts from a further position to detect the departure landmark and to have a reliable starting position.

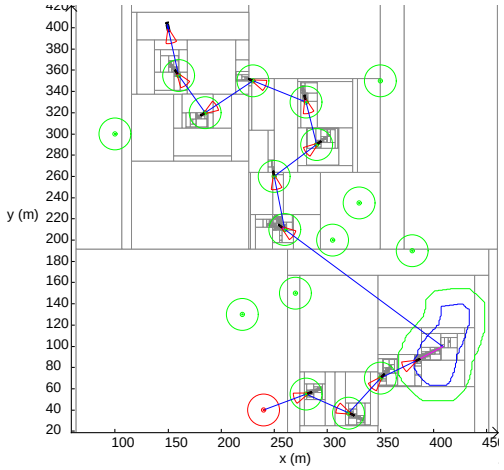


(a) Graph built with the first method by starting from a pose that is inside the registration map of a landmark.

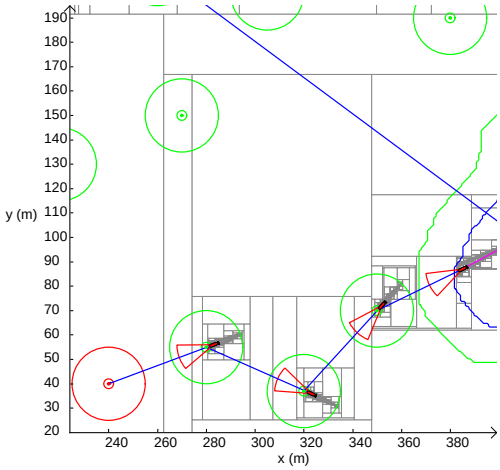
(b) Graph built with the second method by starting from a position and then turn at the selected direction.

Figure 6.54: Environment composed of punctual landmarks represented by green dots, a shape landmark shown by the blue shape and a target considered as punctual landmark in red. The circles or the contour of the shape in green are the visibility at R_{min} and R_{max} of the landmarks. The graph built is represented by blue links and the path found in red.

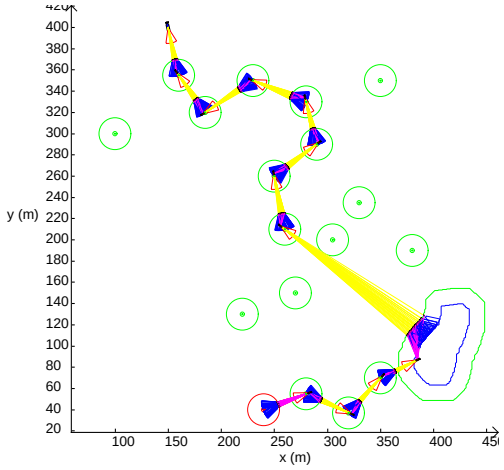
The strategy for the first method of departure positions is detailed in Figure 6.55 where directions are selected taking into account the local ambiguous landmarks. The set of departure positions according to the selected directions in each registration maps are depicted in Figure 6.55(a) where a zoom is provided in Figure 6.55(b). Some red AUVs positions are selected in the different departure sets as departure positions. The red pies corresponds to the visibility area before starting the motion between the registration maps. Magenta lines correspond to relocation process to place the robot at the departure AUV positions. The path execution is depicted in Figure 6.55(c) where again a zoom is provided in Figure 6.55(d). 50 trajectories are generated for each subpath by selecting a random error in the interval $[-5, 5](deg)$ for the error on the direction. Yellow lines are then the uncertain trajectories between the registration maps and magenta lines are the relocation processes. Blue pies correspond to the visibility area when the robot detects the the landmark, i.e. the intersection between the visibility area and the landmark is not empty.



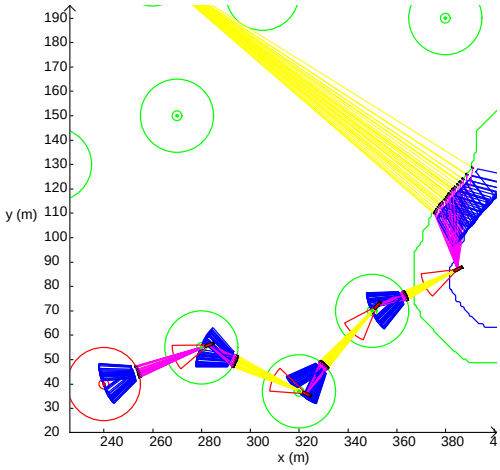
(a) Set of departure positions according to the different selected directions with some departure AUVs positions defined.



(b) Zoom on a part of the figure (a).



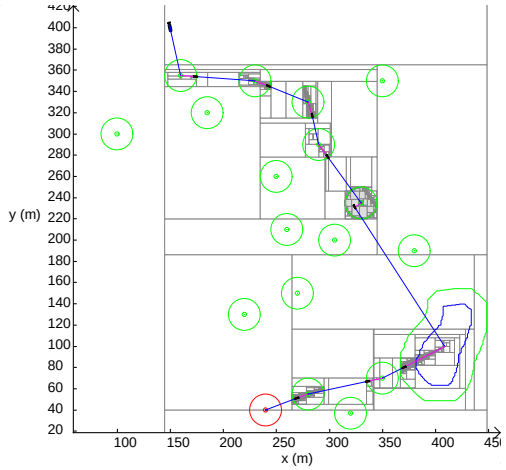
(c) Path execution with the departure red AUVs positions. Yellow trajectories are uncertain motions between relocation maps and magenta trajectories are relocation process. Blue and red pies are the visibility area of the sensor.



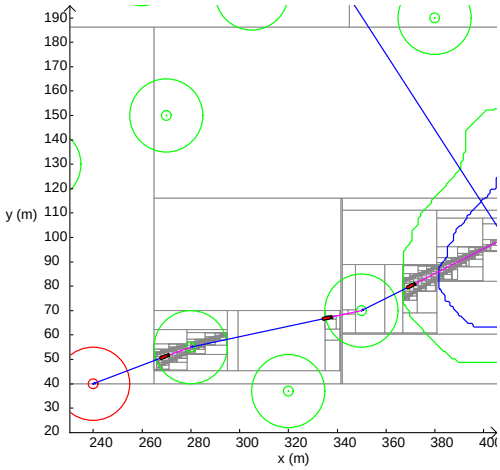
(d) Zoom on a part of the figure (c).

Figure 6.55: Strategy found with the first method of departure poses. One direction is selected for each subpath taking into account the possible local ambiguous landmarks.

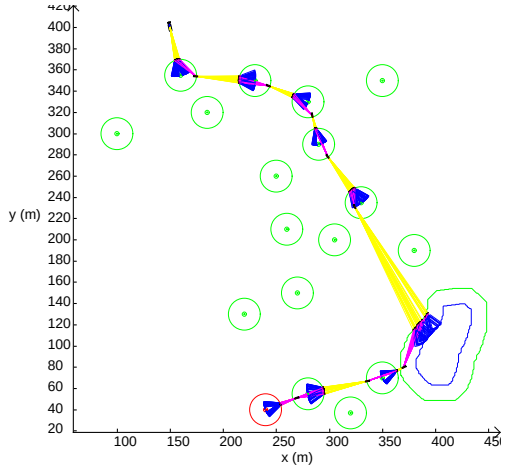
Similarly the strategy for the second method of departure positions is detailed in Figure 6.56 where the strategy is more straight-forward since the robot can turn on itself before leaving a landmark region of influence.



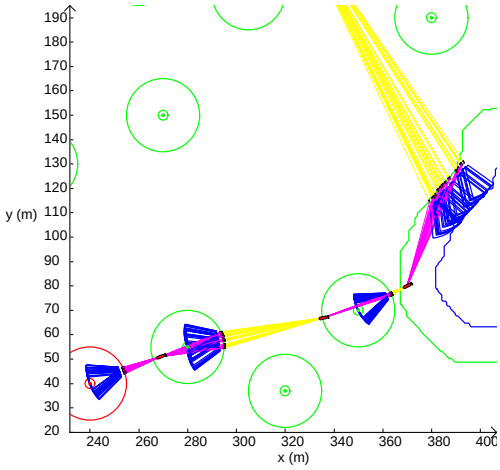
(a) Set of departure positions according to the different selected directions with some departure AUVs positions defined.



(b) Zoom on a part of the figure (a).



(c) Path execution with the departure red AUVs positions. Yellow trajectories are uncertain motions between relocation maps and magenta trajectories are relocation process. Blue pies are the visibility area of the sensor.



(d) Zoom on a part of the figure (c).

Figure 6.56: Strategy found with the second method of departure poses. One direction is selected for each subpath taking into account the possible local ambiguous landmarks.

6.6.2 Planning with point of view

The notion of point of view of the landmark is now taken into account to revisit the different landmark at an approximate same angle of view as in the survey mission. As depicted in Figure 2.27, an angular flexibility around the angle of detection was introduced since it is quite impossible to revisit a landmark at the exact same angle of view. This angular flexibility is defined at 15°. The motion planning problem will be illustrated through an example that involves only punctual landmarks again that are certainly located. The registration maps taking into account an angle of view was presented in Chapter 5 in section 5.2.5. The point of view was defined as a constraint in Equation 5.84 and depicted in Figure 5.16 for the point of view at 0°. Finally the registration map with the point of view is given by Equation 5.82 and was named $\mathbb{P}^{\mathbb{B}}([\theta_{detection}])$ (with $\mathbb{B} = \{\mathbf{b}\}$) for a punctual landmark located at $\mathbf{b} = (b_1, b_2)$. With an

angular flexibility of 15°, it gives $[\theta_{detection}] = \theta_{survey} + [-15, 15](deg)$. It will then be named $\mathbb{P}^{\mathbf{b}}([\theta_{detection}])$. It is obviously a subset of $\mathbb{R}^3$. Consider a sensor that can detect between 4 and $20m$ with an aperture angle of $2\theta_3 = 60°$. However the aperture angle of the sensor does not matter anymore since it is assumed that the vehicle is able to turn on itself to detect the landmark. Only the ranges are important then even if the registration map in Figure 5.16 was computed based on this aperture angle.

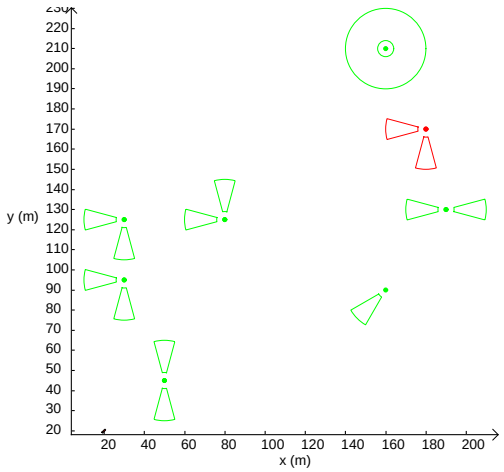
The set of robot poses that can detect the landmark is given $\mathbb{P}^{\mathbf{b}}([\theta_{detection}])$ and then the position of the robot taking into account the point of view is given by the following set:

$$\mathbb{P}^{\mathbf{b}}_{2D}([\theta_{detection}]) = proj^{\mathbb{X}_3}_{\mathbb{X}_1 \times \mathbb{X}_2}(\mathbb{P}^{\mathbf{b}}([\theta_{detection}])) \tag{6.274}$$

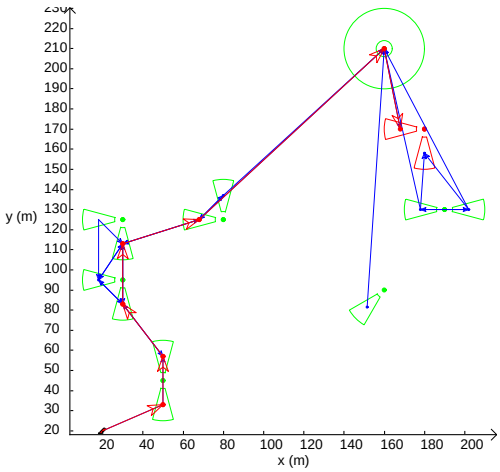
An environment is depicted in Figure 6.57(a) where many punctuals including the target in red were detected at a specific angle of view. Only one punctual landmark could have benefit from multiview and is detectable at any point of view. Using the motion model $\mathbf{f}_H$ in Equation 6.14 is not possible, a Dubins path would be more adapted since the vehicle can only see forward. However it is still possible to build but now the goal recognizability presented in the subsection 6.4.2.5 will be used. It is assumed then only 2D relocation areas as presented in Figure 6.57(a) and the reachability and the recognizability will be computed at the same time by determining the existence of direction and time command. The motion command is now (θ_d, t) as presented in subsection 6.4.2.5, meaning that the robot is able to reach the relocation area by following the direction θ_d despite the uncertain on the direction α_θ and stops in the relocation area to turn on itself after a certain time t based on the cruise speed $(1m/s)$ and the uncertainty on the speed $\alpha_v = 10\%$. The second motion model of linear trajectories $\mathbf{f}_{H2}$ in Equation 6.15 will consequently be used.

Starting from the two possible goal areas, the graph construction is provided in Figure 6.57(b) where blue arrows indicate the departure area and the reachable area, and the red path is the strategy found.

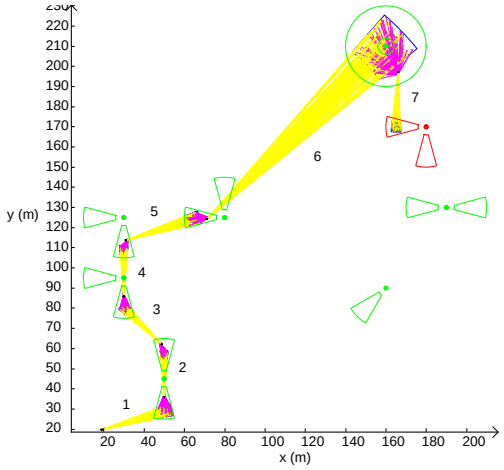
The execution of this strategy is proposed in Figure 6.57(c) where the uncertain trajectories outside relocation areas are painted in yellow and the relocation process is drawn in magenta to join a possible departure position to execute the motion command associated to this departure position and the intermediate or final relocation areas to reach. The AUV may have to turn on itself to relocate before executing the motion to be sure it is at the right position since it can only see forward. The set of departure positions in each departure area according to the strategy found has been computed in Figure 6.57(d) that indicates the position of the robot from where a motion command (θ_d, t) exists to reach the intermediate or goal area.



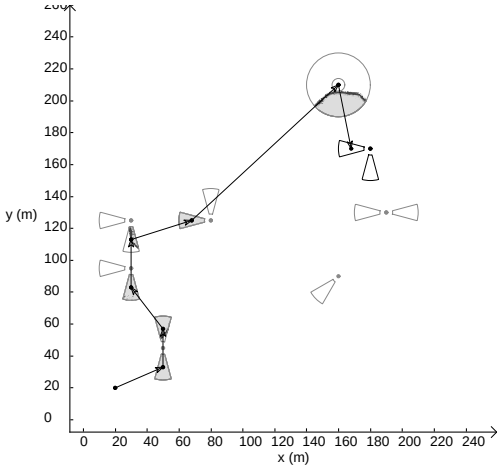
(a) Environment with punctual landmarks (green dots), a target (red dot) and their point of view taking into account the range of the sensor. The initial pose is in the bottom left corner represented by a red AUV.



(b) Graph building with the path found in red. Blue arrows show the reachability.



(c) Strategy execution with uncertain trajectories in yellow and with relocation process in magenta in the relocation areas. Number are given for the different sub-strategy.



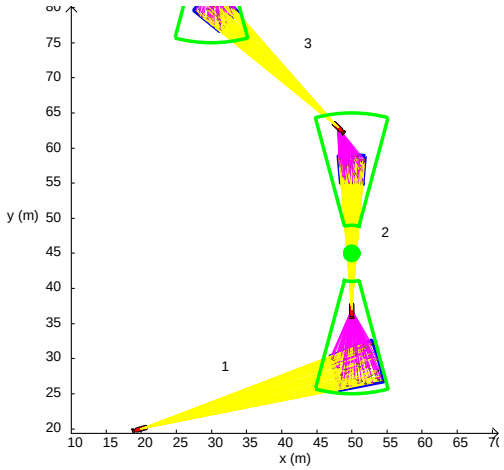
(d) Set of departure positions in each relocation area according to the strategy found. Grayscale convention for thin sets.

Figure 6.57: Graph construction, strategy finding and path execution.

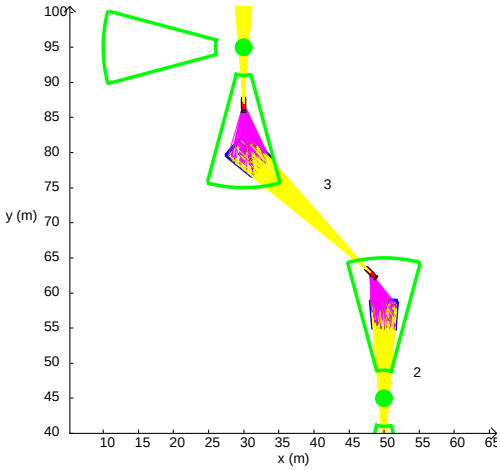
Several zooms of the path execution is provided in Figure 6.58. As it may be noticed in Figure 6.58(c) joining the isotropic detectable landmark was reduced to reach the disk defined by the maximum range. Indeed it guarantees to it will be in the field of influence of the landmark. Then the robot has just to turn on itself to detect the landmark. If it is not detected, the robot is likely in the blind zone in the middle due to the minimum range of detection. It has to perform then a little motion to try to escape from this blind zone and detect by turning on itself.

A focus on the path (or sub-strategy) 3 is given in Figure 6.58(b) where the set of motion commands (θ_d, t) has been computed in Figure 6.58(d) from this initial location. The motion commands choiced is represented by the blue dot. For each path, the set of motion commands has been generated according to the defined initial location and a single motion command is taken from the inner approximation. It has been proved that from this initial location at least

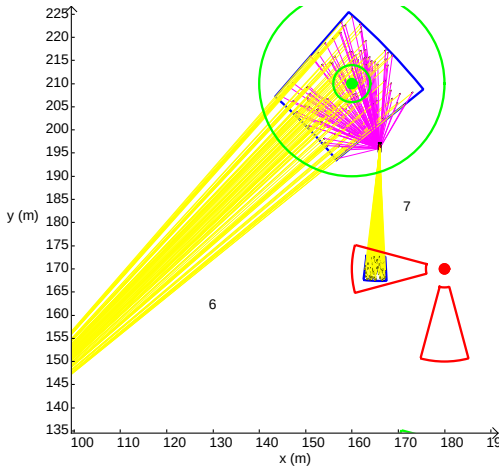
one motion command exists as shown in Figure 6.57(d).



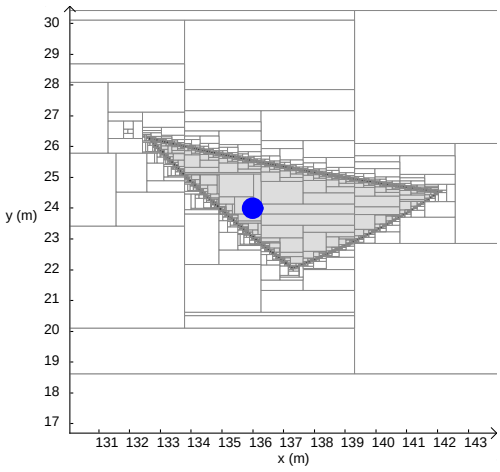
(a) Zoom on the beginning of the high level strategy. Same description as in Figure 6.57(c).



(b) Zoom on the path 3 with the motion command shown by the blue dot in the figure (d).



(c) Recognizability of the isotropic landmark.



(d) Set of motion commands for the path 3 at the defined location.

Figure 6.58: Graph construction, strategy finding and path execution.

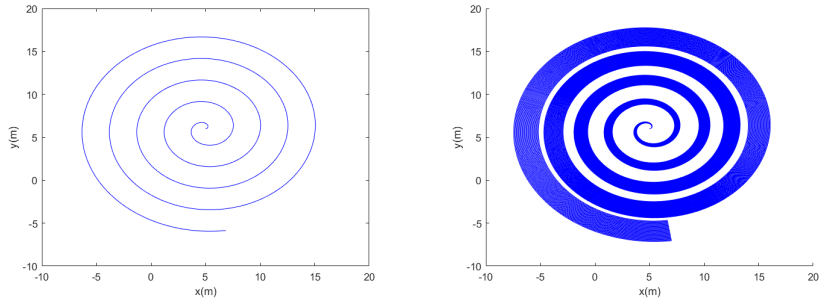
6.6.3 Spiral exploration

When the robot has to reach a set of indistinguishable landmarks, a "local" strategy to disambiguate the true position has to be developed to continue the high level strategy that relies on a well-defined position (or uncertainly defined) of the AUV. Different hypotheses of the localization of the robot are removed by maximizing the information gain in [206], or in other words by going in places where the sensor measurements can help in the disambiguation among the hypotheses. In this subsection, a simple model such as a spiral exploration is proposed. For simplicity reasons, only a 2D motion model will be proposed.

A spiral can be simply described by a parametric equation as follows:

$$\mathbf{f}_{spi} : \begin{cases} \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R} & \rightarrow & \mathbb{R}^2 \\ (\mathbf{x}_0, a, t) & \rightarrow & \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + a \cdot t \begin{pmatrix} \cos(t + \theta_0) \\ \sin(t + \theta_0) \end{pmatrix} \end{cases} \tag{6.275}$$

where $\mathbf{x}_0 = (x_0, y_0, \theta_0)$ is the initial pose, t is the time and a is the parameter that handles the shape of the ellipse.



(a) Without uncertainty.

(b) With an uncertainty $\alpha_a \in [-0.1, 0.1]$.

Figure 6.59: Spiral motion with $a = 0.4m/s$ from the initial pose $(5, 6, 0°)$.

Consider now that a is uncertainly defined to take into account a drift in position by following the spiral motion. The function is now defined as follows:

$$\mathbf{f}_{spi} : \begin{cases} \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} & \rightarrow & \mathbb{R}^2 \\ (\mathbf{x}_0, a, \alpha_a, t) & \rightarrow & \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + a \cdot (1 + \alpha_a) \cdot t \begin{pmatrix} \cos(t + \theta_0) \\ \sin(t + \theta_0) \end{pmatrix} \end{cases} \tag{6.276}$$

where α_a is a parameter to model an uncertainty on a. In the set-membership context, α_a belongs to an interval. In the following, this interval is defined by $[-0.1, 0.1]$ which corresponds to an error at 10% on the value of a.

By fixing the initial angle θ_0 at $0°$ for example, the spiral trajectory from the initial location $(5, 6)$ is drawn until $t = 30s$ with a defined at $0.4m/s$ in Figure 6.59 considering or not an uncertainty. An uncertainty leads to an undefined trajectory.

Consider a simple sensor with an isotropic field of view and a limited maximum range defined at $2m$ for example. The visibility area can be defined according to subsection 5.2.1.1. The robot can then detect any landmark at a distance up to $2m$ around it.

For simplicity reasons, consider a punctual landmark located at $(0, 0)$. The associated registration map $\mathbb{A}$ is then only a subset of $\mathbb{R}^2$.

Considering the motion model with uncertainty, the backward reach set can again be computed from this defined registration map. By fixing the initial angle at $\theta_0 = 0°$, Figure 6.61 shows the 2D backward reach set with a defined at 0.7 and $\alpha_a \in [-0.1, 0.1]$.

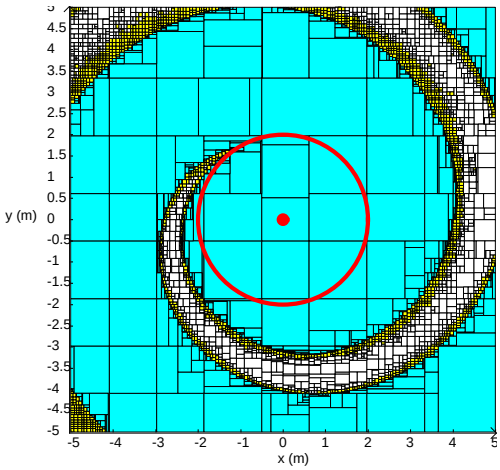


Figure 6.60: 2D backward reach set of the registration map represented by the red circle using the spiral motion with $a = 0.7$ and $\alpha_a \in [-0.1, 0.1]$. Cyan boxes are inside, white ones outside and yellow ones no conclusion can be made.

Based on the 2D backward reach set of the defined registration map it is possible to compute the set of uncertain initial location around the landmark location that guarantees the detection of the landmark. Define for example the initial possible location of the robot by a circle around the position of the landmark as follows:

$$\mathbb{P}(r) = \{\mathbf{x} \in \mathbb{R}^2 | \sqrt{(x_1 - m_1)^2 + (x_2 - m_2)^2} \leq r\} \tag{6.277}$$

where r is the radius of the circle and $\mathbf{m} = (m_1, m_2)$ is the position of the punctual landmark.

Consider now the following set:

$$\mathbb{M} = \{(a, r) \in \mathbb{R}^2 | \mathbb{P}(r) \subset BACK(\mathbb{A}, a)\} \tag{6.278}$$

where $\mathbb{A}$ is the registration map of the punctual landmark and $BACK(\mathbb{A}, a)$ is the backward reach set of $\mathbb{A}$ defined by the uncertain spiral motion shaped by the parameter a where α_a belongs to $[-0.1, 0.1]$. This is a shape registration problem.

According to the provided values, this set $\mathbb{M}$ is represented in Figure 6.61. Obviously, it can be noticed a step when the radius of the initial set of robot position is above the radius of visibility of the sensor $(2m)$.

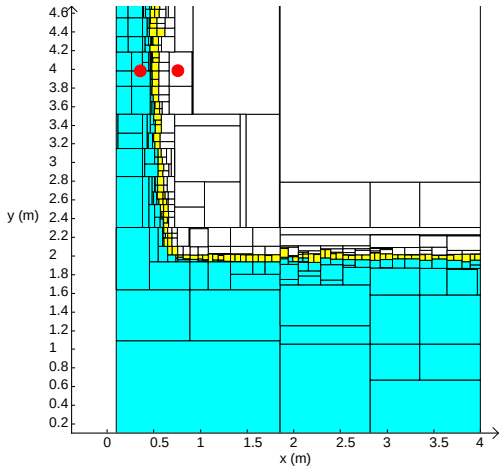
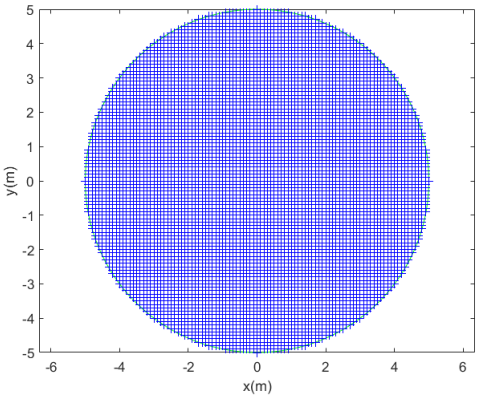
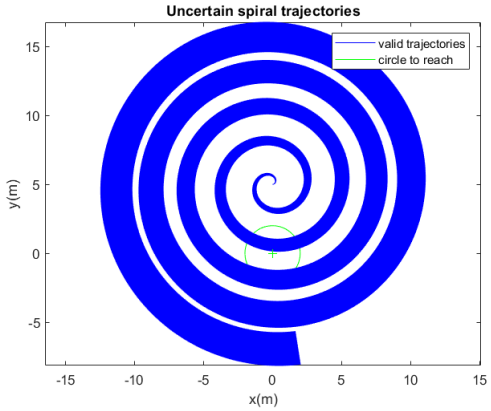


Figure 6.61: Set $\mathbb{M}$ giving the parameter a in function of the radius r of the initial set of robot position. Cyan boxes are inside, white ones outside and yellow ones no conclusion can be made.

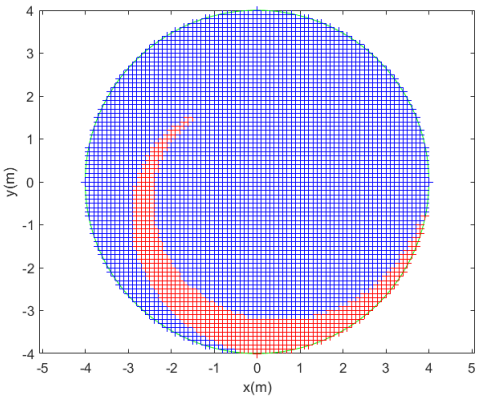
The results with some values (a, r) are presented in Figure 6.62. In Figure 6.62(a), a point inside the inner approximation is chosen ($a = 0.4$ and $r = 4m$) and all the initial locations inside the disk of radius $r = 4m$ enable to revisit the punctual landmark, otherwise some points would be red. The uncertain trajectory from one initial location is provided in Figure 6.62(b) where indeed all the possible trajectories from this initial location cross the registration map (blue trajectories). However, by choosing a point inside the outer approximation ($a = 0.7$ and $r = 4m$), all the initial points are not able to provide the guarantee of the detection of the landmark as shown in Figure 6.62(c) by the red initial points. The uncertain trajectory is represented from one initial non valid position in Figure 6.62(d) where it can be noticed that some trajectories in red never cross the registration map and then do not enable the detection of the landmark.



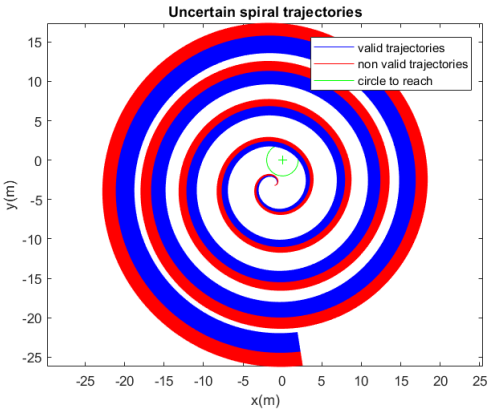
(a) Valid initial location in blue for $a = 0.4$ and $r = 4m$.



(b) All the trajectories (blue) cross the registration map for the defined initial location $(0, 5)$ for example.



(c) Valid (blue) and non valid (red) initial location for $a = 0.7$ and $r = 4m$. Red points do not guarantee the revisit of the landmark due to the uncertain parameter.



(d) Uncertain spiral trajectories from a non valid initial location $(1, 3.2)$. Red trajectories do not cross the registration map.

Figure 6.62: Examples of valid initial location and reachability. The landmark is drawn by a green dot and the registration map by a green circle.

When the initial set of position is inside a disk with a radius $r = 4$, the initial certainly visible area [73] is empty since the robot can detect only at a distance up to $2m$. Despite the fact that the robot cannot detect the landmark at the initial location, with a spiral parameter a defined at 0.4 for example, it guarantees it will detect the landmark at a certain time even with an uncertainty on the motion. The radius of the initial location may be defined by a percentage of the distance travelled (for example 10%).

Consequently, even if a landmark is not detected after a dead-reckoning motion, it may be detected after a spiral motion for example. Moreover, this spiral motion, despite the uncertainty, can extend the certainly visible area to detect possible other landmarks to disambiguate the true position of the robot.

However, this spiral motion may be constrained by the dynamics of the vehicle.

Chapter 7

Conclusion and perspectives

7.1 Conclusion

The underwater environment is a challenging environment where optical systems suffer from a weak visibility contrary to acoustics systems which enable to provide a lot of information about the environment thanks to SOund Navigation And Rangings (SONARs). Underwater robotics is highlighted in Chapter 1 with many types of applications that are nowadays performed by Remotely Operated Vehicles (ROVs) or Autonomous Underwater Vehicles (AUVs). Military applications are the main contributor to the development of unmanned robots to perform redundant, difficult or even dangerous tasks for Humans. Among them, the Mine Counter Measure (MCM) mission shows many interests to secure a naval trafic for commercially trade or to avoid the destruction of expensive ships or submarines. An MCM mission is generally composed of two main tasks. The first one is to survey an area with an highly resoluted sonar such as a side-scan sonar (SSS) or a Synthetic Aperture Sonar (SAS) in order to detect potentially dangerous objects such as mines. This task is performed since the beginning by heavy surface vessels that dispose of an absolute positioning through a Global Positioning System (GPS) measurement. AUVs are replacing more and more surface vessels but they cannot have access to an absolute positioning system since the electromagnetic waves do not penetrate underwater. Generally, the survey AUVs are equipped with an expensive Inertial Navigation System (INS) coupled with a Doppler Velocity Log (DVL) to have a good estimate of the position through a Kalman filter. These suspicious objects are then pretty well geolocalized and need a revisit to identify and neutralize them if necessary. This re-acquisition/identification is the second main task of a MCM mission where human divers performed initially this dangerous task. Due to the increasing abilities of unmanned underwater robots, this dangerous task can be replaced by robots. However, actual technologies show many difficulties to revisit a particular geolocalized object in underwater environment when the robot is not equipped with an high grade navigation solution. Indeed, due to time, economic and safety reasons, the mine killer robot is intended to self detonate to destroy a mine which may not necessary lay on the seabed. Therefore, the design of such robots is said "low cost". Moreover, due to commercially reasons, the future dictates to produce more and more low cost solutions to be competitive. An alternative solution would be to reuse the information gathered by the exteroceptive sensors during the survey missions or previous missions. Navigating directly on this *a priori* known environment does not mean the robot will be able to revisit the suspicious object. Indeed, it firstly needs to relocate the robot on the *a priori* map by matching features of interest known as landmarks detected during the revisit mission before being located relatively to the suspicious object. Heavy data association algorithms could retrieve the trajectory of the robot but it demands to navigate in features field environment to make the associations between the observed landmarks and the ones that are in the *a priori* map.

The solution proposed in this thesis is to provide an high level strategy to guarantee the revisit of the suspicious object based on the *a priori* map and on the exteroceptive abilities of the sensors embedded in the revisit vehicle. The problem to solved can be seen as a motion planning problem.

Chapter 3 presents firstly the state-of-the-art on the classical motion planning problem without considering uncertainty. However, due to the presence of uncertainties in sensing, in motion and possibly in the map, planning under uncertainty is important for "low cost" robot and even vital in underwater environment. The state-of-the-art on the motion planning under uncertainty has then been developed to finally propose at the end of Chapter 3 two motion planners. The first one consists in planning in an extended state space (pose+uncertainty) and assumes a growing error in position proportional to the distance travelled. The second planner is based on the famous Probabilistic Path Planner (PPP) where the Dubins paths are used to take into account possible non-holonomic constraint of the AUV. The drift in position was modelled based on errors on the inputs. However, due to the unstructured aspect of the underwater environment and sometimes to the weak presence of landmarks to relocate the position of the robot, a more elaborated motion planning algorithm has to envisaged. Moreover the exteroceptive sensors, such as sonars, have a limited Field Of View (FOV) and therefore it has to be taken into account at the planning phase.

Chapter 4 introduces the basic tools necessary to understand and manipulate sets with interval analysis. Coupled with a *paver*, separators are able to provide an inner and an outer approximation of sets that are defined by constraints. Due to the presence of textured areas on the seabed in sonar images, the image separator is introduced where it defines as a set the binarized (or segmented) sonar image. The notion of projection is finally introduced, as it will be the main operator all along this thesis, where it can reduce the dimensionality of a problem by proving the existence of solution on a certain domain. It is a powerful tool when the dimension of the sets is greater than 2.

Under the assumption of a 2D environment and based on the *a priori* map where any shaped landmarks are present (may be punctual or bidimensional for example), the registration maps are introduced in Chapter 5 due to the limited FOV of the exteroceptive sensors. It indicates rigorously the set (thin) of robot configurations (x, y, θ) that are able to detect the landmark considered based on the assumed 2D visibility area of the sensors. By determining the 2D translation and rotation parameters, it simply provides the robot poses that can detect a part of a landmark which is sufficient to relocate the robot. The notion of uncertainty on the visibility area and on the map is then introduced as thick sets which are defined as intervals of sets. The notion of *penumbra* appears when manipulating thick sets and avoid to spend a lot of time to prove that an element is actually neither in the inner approximation nor in the outer approximation but in the penumbra (uncertain region).
The environment is then composed of a set of landmarks where a registration map can be computed for each of them. It corresponds to poses that can reduce the uncertainty on the robot position.

Under the assumption that the robot is equipped with a compass that provides the heading information with a bounded uncertainty, the motion planning problem is proposed in a set-membership manner in Chapter 6. The idea is to navigate between the registration maps to reduce each times the growing uncertainty on the robot position. This idea was initially proposed in the work in [186] where only disks were considered as relocation areas. The uncer-

tainties are modelled as uncertain parameters on two motion models. The first one corresponds to a parametric linear trajectory with an uncertain heading and speed, leading to a conic propagation of the position, and a second model based on the parametric Dubins paths with the same uncertain parameters. The reachability between two registration maps, defined as relocation areas, is computed based on the backward reach set of one registration map onto the another. Considering the parametric linear motion model with an uncertain but bounded heading, the directional and omnidirectional backprojections initially proposed in [186] is introduced as a backward reach set. The problem consists then in finding the set of position and direction in a relocation area to reach despite the uncertainties the another relocation area. The problem was reduced to find the set of directions thanks to the projections (1D representation). It is assumed that the vehicle can turn on itself and orient its head to be in the desired direction based on the compass measurements providing the heading (with an uncertainty). Starting from a goal area, a graph can be built backwardly by proving these guaranteed links between the relocation areas (registration maps) assuming the robot is able to move perfectly or imperfectly in a relocation area. It is then assumed some visual guided motion based to join a departure area. If the initial pose is included in a backward reach set of a relocation area, then a strategy has been found. Two motion planners are proposed making the difference between indistinguishable and distinguishable relocation areas. If all the relocation areas are distinguishable, i.e. the robot is able to say in which it is, complex strategies can be envisaged such as reaching a set of disconnected relocation areas. However, in the underwater environment, this assumption may be true for some landmarks but a second planner is proposed to avoid the ambiguity between the relocation areas. This second planner proposes then a strategy to follow without ambiguity of wrong data association. This reasoning may be useful when planning with obstacles. The directional and omnidirectional backprojection are then applied to the registration maps to take into the visibility area of the sensor. The notion of points of view of landmarks introduced in Chapter 2 is taken into account to propose a strategy to see again the intermediate landmarks at approximately the same point of view based on the fact that the robot is able to turn on itself. Finally, when dealing with indistinguishable landmarks, reaching a set of landmarks, that is proved, still requires to disambiguate the true position of the robot. It corresponds to the problem of the initial position recovery. Many pattern algorithms could be imagined. A spiral strategy has then been proposed to show that it could cover a large area without knowing the initial position, and guaranteeing the revisit of the landmarks in the set of landmarks.

The motion planners based on directional/omnidirectional backprojection assumed that a low level controller is able to orient the robot in the desired direction according to the heading measurements. This can be realized when considering an holonomic robot. If the robot has a minimum turning radius non null, it is still possible to consider the planners proposed but the autorelocation has to be proved which is also dependent on the detection of the landmark and then the visual guided motion.

The planner proposed in this thesis is inspired by the landmark based navigation proposed in [186]. However, in this previous work, the environment was only composed of disks as relocation areas (and obstacles) and the problem was solved analytically. In this thesis, the planner is extended to unstructured relocation areas since it only requires sets as inputs that can come from a binary image, a constraint network,... Moreover, the work is extended to take into account the limited visibility area of the sensor by proposing the registration maps. The planner proposes finally a guaranteed strategy to redetect intermediate landmarks before redetecting the target and is optimized through a graph expansion and search. Only few parameters have to be defined to compute the graph. The plan provided by the planners is a high level strategy which requires some local planning algorithms based on visual guided motion for example or

local strategy to disambiguate the position when reaching a set of landmarks.

Real experiments still have to be conducted. A project has been proposed to third-year robotics students at ENSTA Bretagne with a sail boat. A parallel of the strategy proposed in thesis can be made with the old navigation system based on a compass. By following directions with an uncertainty, the sail boat had to detect geolocalized orange buoys with a GoPro that was mounted at the front and could detect forward, to relocate the robot ans continue its strategy that was provided to students. The navigation of a sail boat is different from AUVs since it relies on the wind. However, some results have been obtained and the project should be reconduct next year due to the time to set up the autonomous sail boat.

7.2 Perspectives

Many algorithms were cited in Chapter 2 to extract information from sonar images, and especially sidesan sonar images. Algorithms to detect automatically textures and objects in Forward Looking Sonar (FLS) have not been treated but some of the proposed algorithms in sidescan sonar images may be applicable to FLS imagery. At the end of Chapter 2, it is assumed a simple filter function to remove all the landmarks that may not be detectable by the revisit system according to its payload. An adaptation between the two systems based on the possible resolutions was proposed to keep only the possibly detectable landmarks. This filter function needs to be more elaborated according to the sensors used and the landmarks that were previously detected.

The motion planners proposed assume that the robot was certainly or uncertainly located in a relocation area (registration map) whatever the shape of the landmark. When dealing with uncertain location, the uncertainty was the same everywhere in the associated registration map defined by a box, an ellipse or a circle. The link between two registration maps $\mathbb{A}$ and $\mathbb{B}$ was simply based on the backward reachability of $\mathbb{B}$ onto $\mathbb{A}$ considering or not the uncertain location in $\mathbb{A}$. In reality, this is not true as depicted in Figure 7.1 where a square landmark in blue is presented. The light red area corresponds to the projection in the $(x-y)$ plane of the associated registration map according to the visibility area of the sensor shown by a black pie at an AUV poses. As it may be guessed, a measurement of the landmark at the poses B will not provide a good estimate of the AUV poses contrary to the measurement in A. Indeed, detecting a corner and based on the heading measurement coming from the compass, the position of the AUV is pretty well defined contrary to the measurement in B which could be anywhere along the left side. Therefore, the motion planners should consider the four red disks as exit regions from this landmark and not the whole landmark as it was defined.

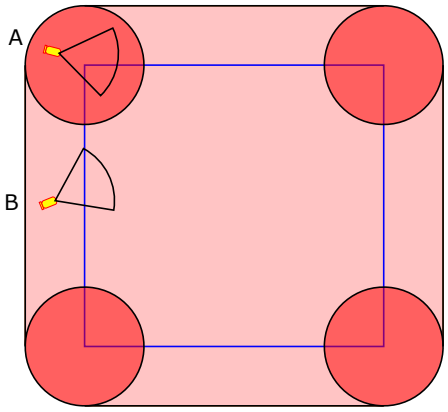


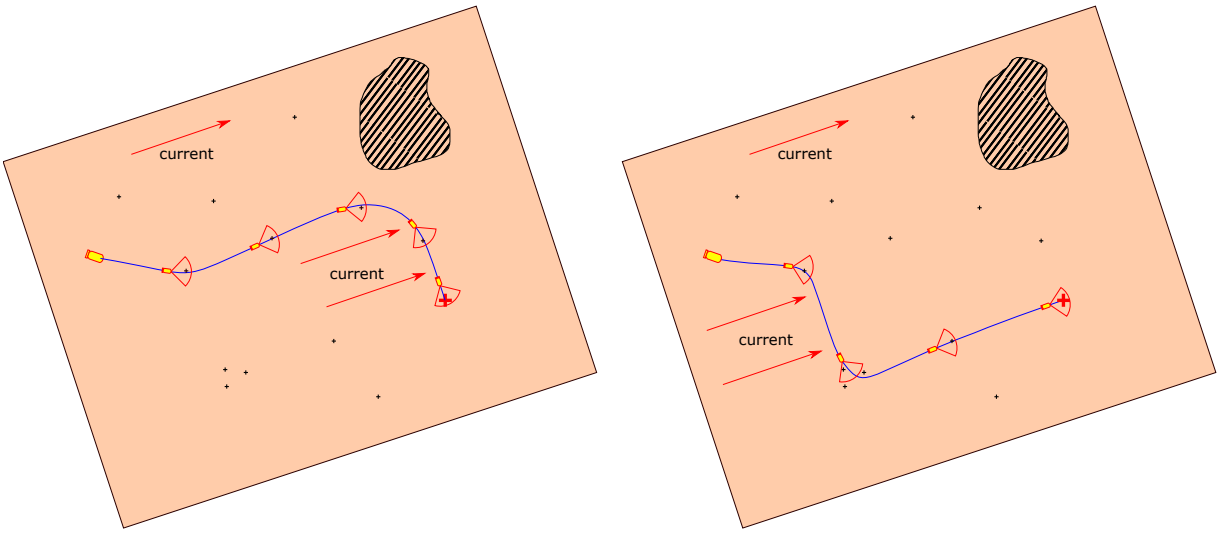
Figure 7.1: A landmark region represented by a blue square, the visibility area according to the sensor in light red and low error in positioning in red circles. A measurement in B would return an high uncertain position contrary to a measurement in A.

Due to the unstructured aspect of the underwater environment, a map could be computed in a 2D grid for example that indicates the relocation abilities at every positions. Orientations could added to feed a 3D grid. It would provide for each poses, if it takes a measurement of a landmark at these poses, the error on the robot poses according to the errors in the map and in sensing. Regions where a relocation is reliable should then be considered as exit regions (departure poses) in the test of connection between two registration maps. This corresponds to the concept of SUF (Sensory Uncertainty Field) initially proposed in [304] where a polygonal environment and range-laser measurements were used to compute the uncertainty on the robot poses after a measurement at every poses in a 3D grid. This could be adapted here by using the shape registration problem developed in Chapter 5 to match the part measured of a shape and the shape itself, presented in [71] and well suited for image registration in instructured environment when 2D landmarks are considered. w_∞ could be used to characterize the relocation abilities. If the landmark is punctual, the problem is a simple inversion problem.

Due to the presence of various elements in the underwater environment that may be very distinct from each other such as the difference between wrecks and sand ripples regions, the two motion planners considering distinguishable and indistinguishable landmarks should be merged to have a complete motion planner adapted to the elements contained in the map. Indeed, if two landmarks are very distinct from each other, if it is proved that a connection exists between them and if the registration maps are not connected, then they could be considered together to extend the range of exploration as in the distinguishable case. However, if another landmarks may lie between the actual location and the two particular landmarks, it has to be proved that a wrong data association cannot be made due to different acoustics backscatterer. If it is ambiguous, such complex strategies cannot be realized. Moreover the notion of point of view, which is important in underwater environment, could be added to a from a complete planner when some landmarks are not detectable isotropically.

The notion of currents, which is nevertheless inherent in oceans, has not been taken into account in the planners proposed. However, it is possible to inflate the uncertainty on the parameters to propose a strategy that requires more intermediate relocation areas in the plan. Another strategy would be to penalize directions that are perpendicularly to the direction of the current as depicted in Figure 7.2 where the direction is represented by a red arrow. Indeed, as it can be seen the minimum path length for example is provided in Figure 7.2(a) where some individual landmarks are revisited before reaching the target represented by the red cross

without taking into account the current. However, the two lasts portions of the motion have a direction perpendicular to the direction of the current that may lead to crabbing effect even if the distance to travel is small. The crabbing effect would displace the robot without changing its direction and the reachability is not guaranteed anymore. Moreover only individuals landmarks were considered in this strategy. When the robot has to navigate perpendicularly to the current, a solution would be to reach a large texture area or a cluster of punctual landmarks as proposed in Figure 7.2(b) where the strategy is to join the three landmarks that are more likely detectable despite the crabbing effect. A strategy has to be envisaged to disambiguate the true position of the robot if necessary. The idea is to manipulate the weights of the links between the relocation areas according to the situation.



(a) Minimum path length strategy returned by the algorithm.

(b) Strategy taking into account the current.

Figure 7.2: Strategy with currents. The target is represented by the red cross and the direction of the current by red arrows. Different poses during the paths are represented by AUVs with their visibility area.

When a step of the high-level strategy is to join a cluster of landmarks, another local strategies can be envisaged to disambiguate the true position of the robot such as *boustrophédon* pattern for example. Spirals were proposed in this thesis since a parametric equation is directly available but any patterns could be imagined to optimize the recovery of the true location of the robot according to the position of the landmarks.

In this thesis, the path optimization was only based on the distance between the centers/centroids of the landmarks that leads to a minimum path length, but different cost function may be imagined. Indeed, a landmark that is very different from the others locally may have a higher score or a landmark that has a strong acoustics backscatterer may be prioritized in the path finding. Moreover, taking into account the probability of detection may be important, especially when the intermediate landmark to search lies in a complex features field environment that could lead false alarms or even wrong data association.

An important point has still to be investigated and was roughly mentionned through this thesis. When executing the strategy, if the robot is not able to detect the considered landmark or target for any reasons, a solution has to be developed. It may simply resurface but this solution is prohibited in enemy area or in deep water. Having an estimate of the error in the

position when joining a landmark may be interesting to make some replanning techniques D* as in [110][109]. Based also on this position estimate, at the moment of the determination of a link between two registration maps, if another landmark can be reached despite the uncertainty in position due to the nondetection, an higher score could be attributed when building the graph. An alternative solution would be to explore locally and build a new map while localizing in this new map (SLAM). When the new map can be matched to the *a priori* map, leading to a position estimation of the robot in the *a priori* map, the motion planner can again be executed to find the strategy to join the target. Another solution would be to prioritize landmark that are close to other landmarks and to avoid isolated landmarks that are difficult to reach. The weighting in the graph would then be based on the density of the landmarks around the considered landmark. This solution could be merged with the previous alternative solution based on SLAM to reacquire a new map rapidly. If a particular landmark in this environment is detected, such as a wreck for example, the robot is directly relocated if this landmark was present in the *a priori* map. An association based only on the image (echo+shadow) could then be envisaged as in optical systems that are able to differentiate with machine learning techniques a cat from a dog thanks to a data base and some features. Recent techniques about deep learning may envisaged to make the associations only based on the sonar image and the information about the landmark.

The proposed high-level strategy through this thesis could be used by coupling different sensors and then different maps. For example consider the *a priori* known bathymetric map in Figure 7.3 where the different values correspond to the isobaths, the target is represented by the red cross and some landmarks are drawn by black crosses. The robot is equipped with an altimeter or a DVL and a camera or a sonar to detect/identify the target. The strategy proposed is to reach the isobath at $10m$, then follow to the North East this isobath before detecting for sure the landmark to provide a good position of the robot. The strategy consists then in reaching the cluster of landmarks, with a strategy of disambiguation for example, and finally to reach the desired target to identify the suspicious object. Using only the vision sensor would not provide any strategy from this initial location due to the far position but using the altitute could extend the reachability possibilities.

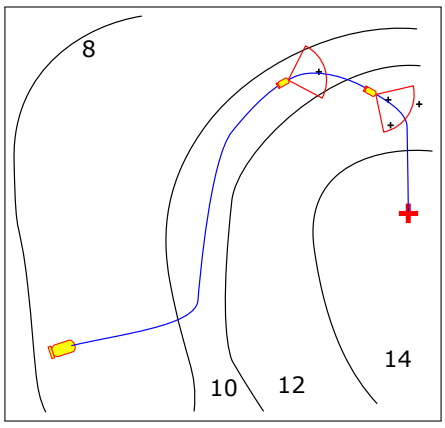


Figure 7.3: Strategy with bathymetry and landmarks detectable by a sonar or camera. The target is represented by the red cross. Different poses during the path are represented by AUVs with the visibility area of the camera or the sonar. Values indicate the isobaths.

It still has to be proved that the robot is able to follow the isobath and to localize the robot in this bathymetric map only based on the altimeter position and the proprioceptive measurements. Two PhD theses, supervised by Luc Jaulin, are conducted at ENSTA Bretagne to prove that the AUV is able to follow an isobath based only on an altimeter measurement

(Julien Damers) and to localize the robot on a bathymetric map only based on a altimeter measurement again (Joris Tillet).

The planner proposed in this thesis could obviously be adapted when the robot is building the map in a SLAM problem to guarantee the reachability of previously detected landmarks and therefore creating the so-called loop closure event.

Finally, I am highly convinced that this "simple" concept presented through this thesis to navigate between relocation areas could increase the autonomy ability of low cost and high grade robots in every *GPS-denied* environment. The principle is highly inspired of the humans sense of direction when they partially know their environment and move inside to reach a desired location. Moreover, this sense of direction has been the starting point of the world exploration when navigators followed the coasts to relocate themself or tried some directions based on the compass measurements to discover new lands.

Bibliography

[1] E. Acar, H. Choset, Y. Zhang, and M. Schervish. Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods. *The International journal of robotics research*, 22(7-8):441–466, 2003. 27

[2] E. Acar, Y. Zhang, H. Choset, M. Schervish, A. G. Costa, R. Melamud, D. C. Lean, and A. Graveline. Path planning for robotic demining and development of a test platform. In *International Conference on Field and Service Robotics*, volume 1, pages 161–168, 2001. 27

[3] A. Adam, E. Rivlin, and I. Shimshoni. Computing the sensory uncertainty field of a vision-based localization sensor. *IEEE Transactions on Robotics and Automation*, 17(3):258–267, 2001. 89

[4] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16(6):641–647, 1994. 55

[5] R. Alami and T. Simeon. Planning robust motion strategies for a mobile robot. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 1312–1318. IEEE, 1994. 90

[6] N. M. Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1, pages 113–120. IEEE, 1996. 86

[7] K. B. Ånonsen. *Advances in terrain aided navigation for underwater vehicles*. PhD thesis, 2010. 20

[8] K. B. Anonsen and O. Hallingstad. Terrain aided underwater navigation using point mass and particle filters. In *Position, location, and navigation symposium, 2006 IEEE/ION*, pages 1027–1035. IEEE, 2006. 22

[9] S. Anstee. Removal of range-dependent artifacts from sidescan sonar imagery. Technical report, DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION VICTORIA (AUSTRALIA) AERONAUTICAL AND MARITIME RESEARCH LAB, 2001. 45

[10] K. O. Arras, J. A. Castellanos, and R. Siegwart. Feature-based multi-hypothesis localization and tracking for mobile robots using geometric constraints. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1371–1377. IEEE, 2002. 24

[11] L. Atallah, C. Shang, and R. Bates. Object detection at different resolution in archaeological side-scan sonar images. In *Oceans 2005-Europe*, volume 1, pages 287–292. IEEE, 2005. 55

[12] J. Aulinas, A. Fazlollahi, J. Salvi, X. Lladó, Y. Petillot, J. Sawas, and R. Garcıa. Robust automatic landmark detection for underwater slam using side-scan sonar imaging. In *Proceedings of the 11th International Conference on Mobile Robots and Competitions*, pages 21–26, 2011. 19

[13] W. Baker and R. Clem. Terrain contour matching [tercom] primer. *Technical ReportASP-TR-77-61. AeronauticalSystems Division, Wright-Patterson AFB*, 1977. 20

[14] M. Barbier, H. Cao, S. Lacroix, C. Lesire, F. Teichteil-Königsbuch, and C. Tessier. Decision issues for multiple heterogeneous vehicles in uncertain environments. In *National conference on control architectures of robots (CAR)*, 2009. 28

[15] A. L. C. Barczak. Toward an efficient implementation of a rotation invariant detector using haar-like features. *Proceedings of the IVCNZ'05, Dunedin, New Zealand*, pages 31–36, 2005. 56

[16] S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba. Bathymetric slam with no map overlap using gaussian processes. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1242–1248. IEEE, 2011. 23

[17] J. Barraquand, L. Kavraki, J. C. Latombe, R. Motwani, T. Y. Li, and P. Raghavan. A random sampling scheme for path planning. *The International Journal of Robotics Research*, 16(6):759–774, 1997. 80

[18] J. Barraquand, B. Langlois, and J. C. Latombe. Numerical potential field techniques for robot path planning. *IEEE transactions on systems, man, and cybernetics*, 22(2):224–241, 1992. 83

[19] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008. 15

[20] M. J. Bays. *Stochastic Motion Planning for Applications in Subsea Survey and Area Protection*. PhD thesis, Virginia Tech, 2012. 28

[21] M. J. Bays, A. Shende, D. J. Stilwell, and S. A. Redfield. A solution to the multiple aspect coverage problem. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1531–1537. IEEE, 2011. 28, 71

[22] F. Benhamou, F. Goualard, L. Granvilliers, and J. F. Puget. Revising hull and box consistency. In *ICLP*, 1999. 119

[23] O. Bergem. Bathymetric navigation of autonomous underwater vehicles using a multibeam sonar and a kalman filter with relative measurement covariance matrices. *Dr. Scient thesis, University of Trondheim, Norway*, 1993. 21

[24] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302, 1986. 55

[25] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 155

[26] P. Bessiere, J. M. Ahuactzin, E. G. Talbi, and E. Mazer. The" ariadne's clew" algorithm: Global planning with local methods. In *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 2, pages 1373–1380. IEEE, 1993. 84

[27] Aymeric Bethencourt. *Analyse par intervalle pour la localisation en essaim. Application à la robotique sous-marine*. PhD thesis, Brest, 2014. 132

[28] L. Blackmore, M. Ono, and B. C. Williams. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6):1080–1094, 2011. 93

[29] P. Blondel. *The handbook of sidescan sonar*. Springer Science & Business Media, 2010. 44, 46, 47

[30] V. Boor, M. H. Overmars, and A. F. Van Der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Robotics and automation, 1999. proceedings. 1999 ieee international conference on*, volume 2, pages 1018–1023. IEEE, 1999. 86

[31] B. Bouilly, T. Simeon, and R. Alami. A numerical technique for planning motion strategies of a mobile robot in presence of uncertainty. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 2, pages 1327–1332. IEEE, 1995. vii, 90, 91

[32] M. Bouvet. *Traitements des signaux pour les systèmes sonar*. Masson Paris, 1992. 36, 41

[33] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001. 54

[34] A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 723–730. IEEE, 2011. vii, 83, 93, 94

[35] A. P. Bry. *Control, estimation, and planning algorithms for aggressive flight using onboard sensing*. PhD thesis, Massachusetts Institute of Technology, 2012. 93

[36] R. W. Button, J. Kamp, T. B. Curtin, and J. Dryden. A survey of missions for unmanned undersea vehicles. Technical report, RAND NATIONAL DEFENSE RESEARCH INST SANTA MONICA CA, 2009. 4

[37] J. F. Canny and M. C. Lin. An opportunistic global path planner. *Algorithmica*, 10(2-4):102–120, 1993. 79

[38] G. Cantor. Beiträge zur begründung der transfiniten mengenlehre. *Mathematische Annalen*, 46(4):481–512, 1895. 112

[39] C. Capus, I. T. Ruiz, and Y. Petillot. Compensation for changing beam pattern and residual tvg effects with sonar altitude variation for sidescan mosaicing and classification. In *7th. Eur. Conf. Underwater Acoustics, Delft, The Netherlands*, 2004. 46

[40] C. G. Capus, A. C. Banks, E. Coiras, I. T. Ruiz, C. J. Smith, and Y. R. Petillot. Data correction for visualisation and classification of sidescan sonar imagery. *IET radar, sonar & navigation*, 2(3):155–169, 2008. 46

[41] C. Carbonnel, G. Trombettoni, P. Vismara, and G. Chabert. Q-intersection algorithms for constraint-based robust parameter estimation. In *AAAI*, pages 2630–2636, 2014. 135

[42] S. Carreno, P. Wilson, P. Ridao, and Y. Petillot. A survey on terrain based navigation for auvs. In *OCEANS 2010*, pages 1–7. IEEE, 2010. 23

[43] A. Censi, D. Calisi, A. De Luca, and G. Oriolo. A bayesian framework for optimal motion planning with uncertainty. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1798–1805. IEEE, 2008. 92

[44] P. Cervenka and C. De Moustier. Sidescan sonar image processing techniques. *IEEE journal of oceanic engineering*, 18(2):108–122, 1993. 45

[45] P. Cervenka, C. De Moustier, and P. F. Lonsdale. Geometric corrections on sidescan sonar images based on bathymetry. application with seamarc ii and sea beam data. *Marine Geophysical Researches*, 16(5):365–383, 1994. 45

[46] G. Chabert. A simple slam example with ibex. In *Small Workshop on Interval Methods, SWIM*, 2013. 129

[47] G. Chabert and L. Jaulin. Contractor programming. *Artificial Intelligence*, 173:1079–1100, 2009. 118, 120, 124, 138, 169

[48] G. Chabert and L. Jaulin. A priori error analysis and spring arithmetic. *SIAM journal on Scientific Computing*, 31(3):2214–2230, 2009. 174

[49] C. Chailloux. *Recalage d'images sonar par appariemment de régions: application à la génération d'une mosaïque*. PhD thesis, Télécom Bretagne, 2007. 18

[50] C. Chailloux, J. M. Le Caillec, D. Gueriot, and B. Zerr. Intensity-based block matching algorithm for mosaicing sonar images. *IEEE Journal of Oceanic Engineering*, 36(4):627–645, 2011. 18

[51] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001. 61

[52] B. Chazelle. Approximation and decomposition of shapes. *Algorithmic and Geometric Aspects of Robotics*, 1:145–185, 1985. 79

[53] C. Chesnaud, P.e Réfrégier, and V. Boulet. Statistical region snake-based segmentation adapted to different physical noise models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1145–1157, 1999. 55

[54] H. Choset. Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomous Robots*, 9(3):247–253, 2000. 27, 79

[55] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005. 82

[56] B. Claus and R. Bachmayer. Development of a magnetometry system for an underwater glider. In *Proc. 17th 1nt. Symp. on Unmanned Untethered Submersible Technology (UUST)*, 2011. 23

[57] C. S. Clay and H. Medwin. Acoustical oceanography: principles and applications. 1977. 3

[58] C. Collet, P. Thourel, M. Mignotte, P. Pérez, and P. Bouthemy. Segmentation markovienne hiérarchique multimodèle d'images sonar haute résolution. *Traitement du Signal*, 15(3):231–250, 1998. 54

[59] C. Combastel. A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 7228–7234. IEEE, 2005. 111

[60] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. Use of active shape models for locating structures in medical images. *Image and vision computing*, 12(6):355–366, 1994. 55

[61] M. Couillard, J. Fawcett, and M. Davison. Optimizing constrained search patterns for remote mine-hunting vehicles. *IEEE Journal of Oceanic Engineering*, 37(1):75–84, 2012. 28, 65

[62] M. Couillard, J. Fawcett, M. Davison, and V. Myers. Optimizing time-limited multi-aspect classification. *Proceedings of the Institute of Acoustics*, 29(6):89–96, 2007. 63

[63] M. Cowie, N. Wilkinson, and R. Powlesland. Latest development of the terprom® digital terrain system (dts). In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 1219–1229. IEEE, 2008. 20

[64] A. Crawford and A. Skarke. Automatic detection of sand ripple features in sidescan sonar imagery. In *Oceans-St. John's, 2014*, pages 1–5. IEEE, 2014. 64

[65] J. L. da Rocha Melo. *Navigation Algorithms for Sensor-limited Autonomous Underwater Vehicles*. PhD thesis, University of Porto, 2016. v, 12, 22, 28

[66] O. Daniell, Y. Petillot, S. Reed, J. Vazquez, and A. Frau. Reducing false alarms in automated target recognition using local sea-floor characteristics. In *Sensor Signal Processing for Defence (SSPD), 2014*, pages 1–5. IEEE, 2014. 54

[67] O. J. Daniell. *Sonar image interpretation for sub-sea operations*. PhD thesis, Heriot-Watt University, 2015. 56

[68] S. Dektor and S. Rock. Improving robustness of terrain-relative navigation for auvs in regions with flat terrain. In *Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES*, pages 1–7. IEEE, 2012. 22

[69] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE, 1999. 15

[70] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using gibbs random fields. *IEEE Transactions on pattern analysis and machine intelligence*, (1):39–55, 1987. 55

[71] B. Desrochers. *SLAM in unstructured environments; a set-membership approach*. PhD thesis, Brest, 2018. 19, 23, 24, 66, 131, 132, 144, 155, 156, 172, 173, 309

[72] B. Desrochers and L. Jaulin. A minimal contractor for the polar equation: Application to robot localization. *Engineering Applications of Artificial Intelligence*, 55:83–92, 2016. 133

[73] B. Desrochers and L. Jaulin. Computing a guaranteed approximation of the zone explored by a robot. *IEEE Transactions on Automatic Control*, 62(1):425–430, 2017. 27, 44, 142, 182, 183, 304

[74] B. Desrochers and L. Jaulin. Minkowski operations of sets with application to robot localization. *arXiv preprint arXiv:1704.03103*, 2017. 136, 155, 156, 157

[75] B. Desrochers and L. Jaulin. Thick set inversion. *Artificial Intelligence*, 249:1–18, 2017. 19, 115, 117, 167, 169, 173

[76] O. Didrit, L. Jaulin, and E. Walter. Guaranteed analysis and optimisation of parametric systems with application to their stability degree. *European journal of control*, 3(1):68–80, 1997. 111

[77] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959. 73, 259

[78] G. J. Dobeck and J. C. Hyland. Automated detection and classification of sea mines in sonar imagery. In *Detection and Remediation Technologies for Mines and Minelike Targets II*, volume 3079, pages 90–111. International Society for Optics and Photonics, 1997. 54

[79] M. F. Doherty, J. G. Landowski, P. F. Maynard, G. T. Uber, D. W. Fries, and F. H. Maltz. Side scan sonar object classification algorithms. In *Unmanned Untethered Submersible Technology, 1989. Proceedings of the 6th International Symposium on*, pages 417–424. IEEE, 1989. 54

[80] Bruce R Donald. The complexity of planar compliant motion planning under uncertainty. *Algorithmica*, 5(1-4):353–382, 1990. 197

[81] G. T. Donovan. Development and testing of a real-time terrain navigation method for auvs. In *OCEANS 2011*, pages 1–9. IEEE, 2011. 22

[82] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc., 2000. 15

[83] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, 79(3):497–516, 1957. 103, 185, 188

[84] V. Ekütekin. *Navigation and control studies on cruise missiles*. PhD thesis, 2007. 20

[85] M. Elbanhawi and M. Simic. Sampling-based robot motion planning: A review. *Ieee access*, 2:56–77, 2014. 94

[86] A. Elfes. A sonar-based mapping and navigation system. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 1151–1156. IEEE, 1986. 24, 80

[87] J. Engström. Tools for classification of mine-like objects in synthetic aperture sonar images. 2007. 55

[88] M. Erdmann. Using backprojections for fine motion planning with uncertainty. *The International Journal of Robotics Research*, 5(1):19–45, 1986. 224

[89] M. A. Erdmann. *On motion planning with uncertainty*. PhD thesis, 1984. 87, 194, 224

[90] L. H. Erickson, J. Knuth, J. M. O'Kane, and S. M. LaValle. Probabilistic localization with a blind robot. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1821–1827. IEEE, 2008. 91

[91] R. Eustice, H. Singh, J. J. Leonard, M. R. Walter, and R. Ballard. Visually navigating the rms titanic with slam information filters. In *Robotics: Science and Systems*, volume 2005, pages 57–64, 2005. 16

[92] R. M. Eustice. *Large-area visually augmented navigation for autonomous underwater vehicles*. PhD thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2005. 16

[93] M. F. Fallon, J. Folkesson, H. McClelland, and J. J. Leonard. Relocating underwater features autonomously using sonar-based slam. *IEEE Journal of Oceanic Engineering*, 38(3):500–513, 2013. 19, 28

[94] M. F. Fallon, M. Kaess, H. Johannsson, and J. J. Leonard. Efficient auv navigation fusing acoustic ranging and side-scan sonar. 2011. 12

[95] R. Fandos. *ADAC system design and its application to mine hunting using SAS imagery*. PhD thesis, Technische Universität, 2012. 55

[96] R. Fandos and A. M. Zoubir. Optimal feature set for automatic detection and classification of underwater objects in sas images. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):454–468, 2011. 55

[97] J. Fawcett, A. Crawford, D. Hopkin, V. Myers, and B. Zerr. Computer-aided detection of targets from the citadel trial klein sonar data. *Defence Research and Development Canada Atlantic TM*, 115, 2006. 54

[98] J. Fawcett, V. Myers, D. Hopkin, A. Crawford, M. Couillard, and B. Zerr. Multiaspect classification of sidescan sonar images: Four different approaches to fusing single-aspect information. *IEEE Journal of Oceanic Engineering*, 35(4):863–876, 2010. 63

[99] F. Ferreira, V. Djapic, M. Micheli, and M. Caccia. Forward looking sonar mosaicing for mine countermeasures. *Annual Reviews in Control*, 40:212–226, 2015. 49

[100] M. Ferrera, J. Moras, P. Trouvé-Peloux, and V. Creuze. Real-time monocular visual odometry for turbid and dynamic underwater environments. *arXiv preprint arXiv:1806.05842*, 2018. 49

[101] J. Folkesson and J. J. Leonard. Autonomy through slam for an underwater robot. In *Robotics Research*, pages 55–70. Springer, 2011. 19

[102] T. Fraichard and R. Mermond. Path planning with uncertainty for car-like robots. In *ICRA*, pages 27–32, 1998. 86, 92, 103

[103] Th. Fraichard and R. Mermond. Integrating uncertainty and landmarks in path planning for car-like robots. In *Proc. IFAC Symp. on Intelligent Autonomous Vehicles March*, volume 25, page 27, 1998. 92, 105

[104] B. Garau, M. Bonet, A. Alvarez, S. Ruiz, and A. Pascual. Path planning for autonomous underwater vehicles in realistic oceanic current fields: Application to gliders in the western mediterranean sea. *Journal of Maritime Research*, 6(2):5–22, 2009. 9

[105] M. Geilhufe and Ø. Midtgaard. Quantifying image complexity for mcm performance estimation. 06 2014. 56

[106] A. Gning and P. Bonnifait. Constraints propagation techniques on intervals for a guaranteed localization using redundant data. *Automatica*, 42(7):1167–1175, 2006. 111

[107] J. P. Golden. Terrain contour matching (tercom): a cruise missile guidance aid. In *Image processing for missile guidance*, volume 238, pages 10–19. International Society for Optics and Photonics, 1980. v, 21

[108] J. P. Gonzalez and A. Stentz. Planning with uncertainty in position an optimal and efficient planner. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2435–2442. IEEE, 2005. 91

[109] J. P. Gonzalez and A. Stentz. Planning with uncertainty in position using high-resolution maps. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1015–1022. IEEE, 2007. 77, 80, 91, 92, 95, 96, 98, 100, 311

[110] J. P. Gonzalez and A. Stentz. Replanning with uncertainty in position: Sensor updates vs. prior map updates. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1806–1813. IEEE, 2008. 77, 311

[111] H. H. González-Banos and J. C. Latombe. Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research*, 21(10-11):829–848, 2002. 91

[112] J. W. Goodman. Some fundamental properties of speckle. *JOSA*, 66(11):1145–1150, 1976. 40

[113] N. Gracias and J. Santos-Victor. Underwater video mosaics as visual navigation maps. *Computer Vision and Image Understanding*, 79(1):66–91, 2000. 16

[114] N. Gracias and J. Santos-Victor. Trajectory reconstruction with uncertainty estimation using mosaic registration. *Robotics and Autonomous Systems*, 35(3-4):163–177, 2001. 16

[115] N. R. Gracias, S. Van Der Zwaan, A. Bernardino, and J. Santos-Victor. Mosaic-based navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 28(4):609–624, 2003. 16

[116] J. Groen, E. Coiras, and D. P. Williams. Detection rate statistics in synthetic aperture sonar images. In *Proc. Intl. Conf. & Exh. Underwater Acoustic Measurements*, pages 367–374, 2009. 64

[117] S. Guillaudeux, S. Daniel, and E. Maillard. Optimization of a sonar image processing chain: A fuzzy rules based expert system approach. In *OCEANS'96. MTS/IEEE. Prospects for the 21st Century. Conference Proceedings*, volume 3, pages 1319–1323. IEEE, 1996. 54

[118] O. K. Hagen, K. B. Ånonsen, and T. O. Sæbø. Low altitude auv terrain navigation using an interferometric sidescan sonar. In *OCEANS 2011*, pages 1–8. IEEE, 2011. 21, 22

[119] A. Hait, T. Siméon, and M. Taïx. A landmark-based motion planner for rough terrain navigation. In *Experimental Robotics V*, pages 216–226. Springer, 1998. 91

[120] A. Hait, T. Siméon, and M. Taïx. Robust motion planning for rough terrain navigation. In *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, volume 1, pages 11–16. IEEE, 1999. 91

[121] R. E. Hansen. Introduction to synthetic aperture sonar. In *Sonar systems*. InTech, 2011. 20

[122] R. E. Hansen, T. O. Sæbø, H. J. Callow, P. E. Hagen, and E. Hammerstad. Synthetic aperture sonar processing for the hugin auv. *Oceans 2005-Europe*, 2:1090–1094, 2005. 20

[123] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 16

[124] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. 73, 259

[125] C. C. Hau. *Handbook of pattern recognition and computer vision*. World Scientific, 2015. 42

[126] M. P. Hayes and P. T. Gough. Synthetic aperture sonar: a review of current status. *IEEE Journal of Oceanic Engineering*, 34(3):207–224, 2009. 20

[127] R. He, S. Prentice, and N. Roy. Planning in information space for a quadrotor helicopter in a gps-denied environment. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1814–1820. IEEE, 2008. 86, 105

[128] J. D. Hernández, E. Vidal, G. Vallicrosa, E. Galceran, and M. Carreras. Online path planning for autonomous underwater vehicles in unknown environments. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1152–1157. IEEE, 2015. 83

[129] C. Holleman and L. E. Kavraki. A framework for using the workspace medial axis in prm planners. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1408–1413. IEEE, 2000. 86

[130] D. Hsu, T. Jiang, J. Reif, and Z. Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 4420–4426. IEEE, 2003. 86

[131] D. Hsu, J. C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 3, pages 2719–2726. IEEE, 1997. 84

[132] M. K Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187, 1962. 63

[133] N. Hurtós, Y. Petillot, and J. Salvi. Fourier-based registrations for two-dimensional forward-looking sonar image mosaicing. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5298–5305. Ieee, 2012. 19

[134] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern recognition*, 24(12):1167–1186, 1991. 56

[135] L. Jaulin. A nonlinear set membership approach for the localization and map building of underwater robots. *IEEE Transactions on Robotics*, 25(1):88–98, 2009. 18

[136] L. Jaulin. Robust set-membership state estimation; application to underwater robotics. *Automatica*, 45(1):202–206, 2009. 113

[137] L. Jaulin. Solving set-valued constraint satisfaction problems. *Computing*, 94(2-4):297–311, 2012. 168

[138] L. Jaulin. Range-only slam with indistinguishable landmarks; a constraint programming approach. *Constraints*, 21(4):557–576, 2016. 111, 115

[139] L Jaulin. Isobath following using an altimeter as a unique exteroceptive sensor. In *International Robotic Sailing Conference 2018 (IRSC 2018)*, 2018. 28, 71

[140] L. Jaulin and B. Desrochers. Introduction to the algebra of separators with application to path planning. *Engineering Applications of Artificial Intelligence*, 33:141–147, 2014. 123, 124, 128, 129, 169

[141] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*, volume 1. Springer Science & Business Media, 2001. 15, 114

[142] L. Jaulin, M. Kieffer, E. Walter, and D. Meizel. Guaranteed robust nonlinear estimation with application to robot localization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(4):374–381, 2002. 111

[143] L. Jaulin, M. Legris, and F. Dabe. Gesmi, un logiciel pour l'aide à localisation de mines sous-marines. *JIME*, 2006. 18

[144] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993. 111, 125

[145] L. Jaulin and E. Walter. Guaranteed robust nonlinear minimax estimation. *IEEE Transactions on Automatic Control*, 47(11):1857–1864, 2002. 19, 135

[146] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt. *Computational ocean acoustics*. Springer Science & Business Media, 2000. 3

[147] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. J. Leonard. Imaging sonar-aided navigation for autonomous underwater harbor surveillance. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4396–4403. IEEE, 2010. 19

[148] A. E. Johnson and J. F. Montgomery. Overview of terrain relative navigation approaches for precise lunar landing. In *Aerospace Conference, 2008 IEEE*, pages 1–10. IEEE, 2008. 20

[149] J. Johnson. Analysis of image forming systems. In *Selected papers on infrared design. Part I and II*, volume 513, page 761, 1985. 67

[150] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998. 87

[151] R. E. Kalman. Contributions to the theory of optimal control. *Bol. Soc. Mat. Mexicana*, 5(2):102–119, 1960. 45

[152] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960. 12

[153] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011. 80, 84, 86

[154] I. Karoui, R. Fablet, J. M. Boucher, and J. M. Augustin. Segmentation texturales des images sonar des fonds marins par une approche variationnelle basée région. 2008. 55

[155] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988. 55

[156] L. Kavraki. *Random networks in configuration space for fast path planning*. Number 1535. stanford university, 1994. 82

[157] L. Kavraki and J. C. Latombe. Randomized preprocessing of configuration for fast path planning. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 2138–2145. IEEE, 1994. 86

[158] L. Kavraki, P. Svestka, and M. H. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*, volume 1994. Unknown Publisher, 1994. 84

[159] A. Kelly. Linearized error propagation in odometry. *The International Journal of Robotics Research*, 23(2):179–218, 2004. 96

[160] M. Khatib. Sensor-based motion control for mobile robots. *Maher Khatib Sensor-based motion control for mobile robots PHD thesis, LAAS-CNRS December*, 1996. 82

[161] M. Khatib, B. Bouilly, T. Siméon, and R. Chatila. Indoor navigation with uncertainty using sensor-based motions. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 4, pages 3379–3384. IEEE, 1997. 90

[162] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, pages 396–404. Springer, 1986. 82

[163] S. Koenig and M. Likhachev. Incremental a. In *Advances in neural information processing systems*, pages 1539–1546, 2002. 77

[164] S. Koenig and M. Likhachev. Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics*, 21(3):354–363, 2005. 77

[165] J. Košecká. Visually guided navigation. In *Intelligent Robots—Sensing, Modeling And Planning*, pages 77–95. World Scientific, 1997. 242

[166] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 995–1001. IEEE, 2000. 83

[167] C. Kunz, C. Murphy, H. Singh, C. Pontbriand, R. A. Sohn, S. Singh, T. Sato, C. Roman, K. Nakamura, M. Jakuba, et al. Toward extraplanetary under-ice exploration: Robotic steps in the arctic. *Journal of Field Robotics*, 26(4):411–429, 2009. 13

[168] H. Kurniawati, T. Bandyopadhyay, and N. M. Patrikalakis. Global motion planning under uncertain motion, sensing, and environment map. *Autonomous Robots*, 33(3):255–272, 2012. 175

[169] A. B. Kurzhanski and P. Varaiya. On reachability under uncertainty. *SIAM Journal on Control and Optimization*, 41(1):181–216, 2002. 111

[170] N. H. Kussat, C. D. Chadwell, and R. Zimmerman. Absolute positioning of an autonomous underwater vehicle using gps and acoustic measurements. *IEEE Journal of Oceanic Engineering*, 30(1):153–164, 2005. 13

[171] H. Lahanier, E. Walter, and R. Gomeni. Omne: a new robust membership-set estimator for the parameters of nonlinear models. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(2):203–219, 1987. 19

[172] A. Lambert and Th. Fraichard. Landmark-based safe path planning for car-like robots. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 3, pages 2046–2051. IEEE, 2000. 86, 92

[173] A. Lambert and D. Gruyer. Safe path planning in an uncertain-configuration space. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 4185–4190. IEEE, 2003. 92, 93, 95, 103

[174] A. Lambert and N. Le Fort-Piat. Safe task planning integrating uncertainties and local maps federations. *The International Journal of Robotics Research*, 19(6):597–611, 2000. 93

[175] L. Lapierre and B. Jouvencel. Robust nonlinear path-following control of an auv. *IEEE Journal of Oceanic Engineering*, 33(2):89–102, 2008. 187

[176] L. Lapierre and D. Soetanto. Nonlinear path-following control of an auv. *Ocean engineering*, 34(11-12):1734–1744, 2007. 187

[177] M. Laranjeira, C. Dune, and V. Hugel. Catenary-based visual servoing for tethered robots. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 732–738. IEEE, 2017. 7

[178] J. C. Latombe. *Motion planning with uncertainty: The preimage backchaining approach*. Department of Computer Science, Stanford University, 1988. 88

[179] J. C. Latombe. *Robot Motion Planning*. Springer Science & Business Media, 1991. 79, 80, 82, 88

[180] J. P. Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. *Proc. International Conference on Intelligent Autonomous Systems*, pages 346–354, 1986. 72, 80

[181] J. P. Laumond, S. Sekhavat, and F. Lamiraux. Guidelines in nonholonomic motion planning for mobile robots. In *Robot motion planning and control*, pages 1–53. Springer, 1998. 83

[182] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998. 83

[183] S. M. LaValle. *Planning algorithms*. Cambridge university press, 2006. vi, 72, 80, 83, 84, 85, 86, 194, 224

[184] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. 2000. 83, 84

[185] A. Lazanas. *Reasoning about uncertainty in robot motion planning*. PhD thesis, to the Department of Computer Science.Stanford University, 1994. 88

[186] A. Lazanas and J. C. Latombe. Landmark-based robot navigation. *Algorithmica*, 13(5):472–501, 1995. vii, 88, 89, 106, 184, 186, 187, 193, 194, 196, 197, 202, 244, 246, 306, 307

[187] A. Lazanas and J. C. Latombe. Motion planning with uncertainty: a landmark approach. *Artificial intelligence*, 76(1-2):287–317, 1995. 88, 186, 197, 202, 246

[188] G. Le Chenadec. *Analyse de descripteurs énergétiques et statistiques de signaux sonar pour la caractérisation des fonds marins*. PhD thesis, Université de Bretagne Occidentale, 2004. 40, 46

[189] T. Le Mézo, L. Jaulin, and B. Zerr. Eulerian state estimation. 2017. 9

[190] I. Leblond, M. Legris, and B. Solaiman. Use of classification and segmentation of sidescan sonar images for long term registration. In *Oceans 2005-Europe*, volume 1, pages 322–327. IEEE, 2005. 47, 56

[191] J. Lee, J. H. Kwon, and M. Yu. Performance evaluation and requirements assessment for gravity gradient referenced navigation. *Sensors*, 15(7):16833–16847, 2015. 23

[192] G. Lehureau, F. Tupin, C. Tison, G. Oller, and D. Petit. Registration of metric resolution sar and optical images in urban areas. In *Synthetic Aperture Radar (EUSAR), 2008 7th European Conference on*, pages 1–4. VDE, 2008. 65

[193] J. J. Leonard, A. A. Bennett, C. M. Smith, H. Jacob, and S. Feder. Autonomous underwater vehicle navigation. In *MIT Marine Robotics Laboratory Technical Memorandum*. Citeseer, 1998. 13

[194] J. S. Lewis and J. M. O'Kane. Guaranteed navigation with an unreliable blind robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5519–5524. IEEE, 2010. 91

[195] J. S. Lewis and J. M. O'Kane. Reliable indoor navigation with an unreliable robot: Allowing temporary uncertainty for maximum mobility. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 160–165. IEEE, 2012. 91

[196] J. S. Lewis and J. M. O'Kane. Planning for provably reliable navigation using an unreliable, nearly sensorless robot. *The International Journal of Robotics Research*, 32(11):1342–1357, 2013. 91

[197] M. L'Hour and V. Creuze. French archaeology's long march to the deep—the lune project: Building the underwater archaeology of the future. In *Experimental Robotics*, pages 911–927. Springer, 2016. 5

[198] H. Li, B. S. Manjunath, and S. K. Mitra. A contour-based approach to multisensor image registration. *IEEE transactions on image processing*, 4(3):320–334, 1995. 65

[199] H. Lichte. On the influence of horizontal temperature layers in sea water on the range of underwater sound signals. 1977. 3

[200] S. R. Lindemann and S. M. LaValle. Current issues in sampling-based motion planning. In *Robotics Research. The Eleventh International Symposium*, pages 36–54. Springer, 2005. 80

[201] T. Lolla. *Path planning in time dependent flow fields using level set methods*. PhD thesis, MIT, 2012. 82

[202] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 15

[203] T. Lozano-Perez, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 3(1):3–24, 1984. 87, 91, 186

[204] L. Lucido, J. Opderbecke, V. Rigaud, R. Deriche, and Z. Zhang. A terrain referenced underwater positioning using sonar bathymetric profiles and multiscale analysis. In *OCEANS'96. MTS/IEEE. Prospects for the 21st Century. Conference Proceedings*, volume 1, pages 417–421. IEEE, 1996. 21

[205] A. K. Mackworth. Consistency in networks of relations. *Artificial intelligence*, 8(1):99–118, 1977. 111

[206] F. Maurelli. *Probabilistic Approaches for Intelligent AUV Localisation*. PhD thesis, Heriot-Watt University, 2014. 300

[207] F. Maurelli, P. Patrón, J. Cartwright, J. Sawas, Y. Petillot, and D. Lane. Integrated mcm missions using heterogeneous fleets of auvs. In *OCEANS, 2012-Yeosu*, pages 1–7. IEEE, 2012. 28

[208] E. Mazer, J. M. Ahuactzin, and P. Bessiere. The ariadne's clew algorithm. *Journal of Artificial Intelligence Research*, 9:295–316, 1998. 84

[209] H. H. G. McClelland. *Using Negative Information in simultaneous localization and mapping*. PhD thesis, Massachusetts Institute of Technology, 2011. 19

[210] G. McLachlan. *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons, 2004. 56

[211] D. K. Meduna. *Terrain relative navigation for sensor-limited systems with application to underwater vehicles*. Stanford University, 2011. 14, 20, 21

[212] D. K. Meduna, S. M. Rock, and R. McEwen. Low-cost terrain relative navigation for long-range auvs. In *OCEANS 2008*, pages 1–7. IEEE, 2008. 22

[213] D. K. Meduna, S. M. Rock, and R. S. McEwen. Closed-loop terrain relative navigation for auvs with non-inertial grade navigation sensors. In *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*, pages 1–8. IEEE, 2010. 21, 28

[214] N. A. Melchior and R. Simmons. Particle rrt for path planning with uncertainty. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1617–1624. IEEE, 2007. 93

[215] J. Melo and A. Matos. Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Engineering*, 139:250–264, 2017. 20, 23

[216] F. Meyer and S. Beucher. Morphological segmentation. *Journal of visual communication and image representation*, 1(1):21–46, 1990. 55

[217] M. Mignotte. Segmentation d'images sonar par approche markovienne hi erarchique non supervis ee et classification d'ombres port ees par mod eles statistiques. *These de doctorat, Ecole Navale, Brest, France*, 1998. 54

[218] M. Mignotte, C. Collet, P. Perez, and P. Bouthemy. Markov random field model and fuzzy formalism-based data modeling for the sea-floor classification. In *Mathematical Modeling, Bayesian Estimation, and Inverse Problems*, volume 3816, pages 229–241. International Society for Optics and Photonics, 1999. 54

[219] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy. Three-class markovian segmentation of high-resolution sonar images. *Computer Vision and Image Understanding*, 76(3):191–204, 1999. 55

[220] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy. Markov random field and fuzzy logic modeling in sonar imagery: application to the classification of underwater floor. *Computer Vision and Image Understanding*, 79(1):4–24, 2000. 55

[221] N. C. Mitchell and M. L. Somers. Quantitative backscatter measurements with a long-range side-scan sonar. *IEEE Journal of Oceanic Engineering*, 14(4):368–374, 1989. 46

[222] U. Montanari and F. Rossi. Constraint relaxation may be perfect. *Artificial intelligence*, 48(2):143–170, 1991. 121

[223] M. Montemerlo. *astslam: A factored solution to the simultaneous localization and mapping problem with unknown data association*. PhD thesis, The Robotics Institute Carnegie Mellon University Pittsburgh, 2003. 24

[224] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598, 2002. 23

[225] R. E. Moore. *Interval Analysis*. 1969. 114

[226] H. Moravec. Robot spatial perceptionby stereoscopic vision and 3d evidence grids. *Perception*, 1996. 80

[227] H. Mu, M. Wu, X. Hu, and H. Ma. Geomagnetic surface navigation using adaptive ekf. In *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, pages 2821–2825. IEEE, 2007. 23

[228] E. M. Murphy. *Planning and Exploring Under Uncertainty*. PhD thesis, University of Oxford, 2010. 75, 80

[229] V. Myers and D. P. Williams. Adaptive multiview target classification in synthetic aperture sonar images using a partially observable markov decision process. *IEEE Journal of Oceanic Engineering*, 37(1):45–55, 2012. 28, 63

[230] L. Najman and H. Talbot. *Mathematical morphology: from theory to applications*. John Wiley & Sons, 2013. 156, 157

[231] US Navy. The navy unmanned undersea vehicle (uuv) master plan. *US Navy, November*, 9:90, 2004. 4

[232] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on robotics and automation*, 17(6):890–897, 2001. 19

[233] B. Nguyen, M. J. Bays, A. Shende, and D. J. Stilwell. An approach to subsea survey for safe naval transit. In *OCEANS 2011*, pages 1–6. IEEE, 2011. 28

[234] F. Nicolas, A. Arnold-Bos, I. Quidu, and B. Zerr. Markov-based approaches for ternary change detection between two high resolution synthetic aperture sonar tracks. In *OCEANS 2017-Aberdeen*, pages 1–9. IEEE, 2017. 18

[235] F. Nicolas, A. Arnold-Bos, I. Quidu, and B. Zerr. Fast fourier-based block-matching algorithm for sonar tracks registration in a multiresolution framework. In *Marine Robotics and Applications*, pages 1–14. Springer, 2018. 18

[236] I. Nygren. *Terrain navigation for underwater vehicles*. PhD thesis, KTH, 2005. 21

[237] I. Nygren. Robust and efficient terrain navigation of underwater vehicles. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 923–932. IEEE, 2008. 21

[238] J. M. O'Kane. *A theory for comparing robot systems*. PhD thesis, University of Illinois at Urbana-Champaign, 2007. 91

[239] L. A. Page and A. C. Sanderson. A path-space search algorithm for motion planning with uncertainties. In *Assembly and Task Planning, 1995. Proceedings., IEEE International Symposium on*, pages 334–340. IEEE, 1995. 89

[240] L. A. Page and A. C. Sanderson. Robot motion planning for sensor-based control with uncertainties. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 2, pages 1333–1340. IEEE, 1995. 89

[241] Y. Pailhas, Y. Petillot, and C. Capus. High-resolution sonars: What resolution do we need for target recognition? *EURASIP Journal on Advances in Signal Processing*, 2010:42, 2010. 67

[242] L. Paull. *Robust Online Adaptive Sensor-Driven Survey Planning for Single and Multiple Autonomous Underwater Vehicles*. PhD thesis, UNIVERSITY OF NEW BRUNSWICK, 2014. v, 17

[243] L. Paull, S. Saeedi, and H. Li. Path planning for autonomous underwater vehicles. In *Marine Robot Autonomy*, pages 1–46. Springer, 2013. vi, 80

[244] L. Paull, M. Seto, and H. Li. Area coverage planning that accounts for pose uncertainty with an auv seabed surveying application. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6592–6599. IEEE, 2014. 27, 183

[245] R. Pepy. *Vers une planification robuste et sûre pour les systèmes autonomes*. PhD thesis, Université Paris Sud-Paris XI, 2009. 93, 105, 107

[246] R. Pepy, M. Kieffer, and E. Walter. Reliable robust path planning. *International Journal of Applied Mathematics and Computer Science*, 19(3):413–424, 2009. 83, 93

[247] R. Pepy, M. Kieffer, and E. Walter. Reliable robust path planning with application to mobile robots. *International Journal of Applied Mathematics and Computer Science*, 19(3):413–424, 2009. 83

[248] R. Pepy, M. Kieffer, and E. Walter. Reliably safe path planning using interval analysis. In *Progress in Industrial Mathematics at ECMI 2008*, pages 583–588. Springer, 2010. 93

[249] R. Pepy and A. Lambert. Safe path planning in an uncertain-configuration space using rrt. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5376–5381. IEEE, 2006. 93

[250] Y. Petillot, F. Maurelli, N. Valeyrie, A. Mallios, P. Ridao, J. Aulinas, and J. Salvi. Acoustic-based techniques for autonomous underwater vehicle localization. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 224(4):293–307, 2010. 21

[251] Y. Petillot, I. T. Ruiz, and D. Lane. Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar. *IEEE Journal of Oceanic Engineering*, 26(2):240–251, 2001. 49

[252] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane. Path planning for autonomous underwater vehicles. *IEEE Transactions on Robotics*, 23(2):331–341, 2007. vii, 82

[253] C. Pêtrès and P. Patrón. Path planning for unmanned underwater vehicles. In *Proc. IJCAI Workshop Planning, Learning in A Priori Unknown or Dynamic Domains*, page 47, 2005. 82

[254] L. Picard. *Prise en compte de l'environnement marin dans le processus de reconnaissance autmatique de cibles sous-marines*. PhD thesis, Brest, 2017. v, vi, 38, 42, 53, 56, 63

[255] L. Picard, A. Baussard, G. Le Chenadec, and I. Quidu. Detection of mlo in sand ripple seafloor using the monogenic signal and intrinsic dimensionality. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–9. IEEE, 2016. 56, 64

[256] G. Piella. Diffusion maps for multimodal registration. *Sensors*, 14(6):10562–10577, 2014. 65

[257] V. Pilania and K. Gupta. A localization aware sampling strategy for motion planning under uncertainty. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 6093–6099. IEEE, 2015. 93, 105

[258] V. Pilania and K. Gupta. Localization aware sampling and connection strategies for incremental motion planning under uncertainty. *Autonomous Robots*, 41(1):111–132, 2017. 93, 105

[259] J. Pineau, G. Gordon, S. Thrun, et al. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI*, volume 3, pages 1025–1032, 2003. 87, 90

[260] C. Pradalier and S. Sekhavat. " localization space": a framework for localization and planning, for systems using a sensor/landmarks module. 2002. 142

[261] S. Prentice and N. Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research*, 28(11-12):1448–1465, 2009. 92

[262] I. Quidu. *Classification multi-vues d'un objet immergé à partir d'images sonar et de son ombre portée sur le fond*. PhD thesis, Université de Bretagne occidentale-Brest, 2001. 28, 63

[263] I. Quidu, J. P. Malkasse, G. Burel, and P. Vilbé. Mine classification using a hybrid set of descriptors. In *OCEANS 2000 MTS/IEEE Conference and Exhibition*, volume 1, pages 291–297. IEEE, 2000. 54

[264] N. Ramdani and P. Poignet. Robust dynamic experimental identification of robots with set membership uncertainty. *IEEE/ASME Transactions on Mechatronics*, 10(2):253–256, 2005. 111

[265] D. Rao and S. B. Williams. Large-scale path planning for underwater gliders in ocean currents. In *Australasian Conference on Robotics and Automation (ACRA)*, pages 2–4, 2009. 9, 12

[266] S. Reed, Y. Petillot, and J. Bell. An automatic approach to the detection and extraction of mine features in sidescan sonar. *IEEE journal of oceanic engineering*, 28(1):90–105, 2003. 54, 55

[267] S. Reed, I. T. Ruiz, C. Capus, and Y. Petillot. The automatic fusion of classified sidescan sonar mosaics using cml-rts and markov random fields. In *Oceans 2005-Europe*, volume 2, pages 883–888. IEEE, 2005. 54, 55

[268] D. Reid. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 24(6):843–854, 1979. 18, 63

[269] Z. Reut, N. G. Pace, and M. J. P. Heaton. Computer classification of sea beds by sonar. *Nature*, 314(6010):426, 1985. 56

[270] D. Ribas, P. Ridao, and J. Neira. *Underwater SLAM for structured environments using an imaging sonar*, volume 65. Springer, 2010. 20

[271] D. Ribas, P. Ridao, J. Neira, and J. D. Tardos. Slam using an imaging sonar for partially structured underwater environments. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5040–5045. IEEE, 2006. 20

[272] H. Rice, S. Kelmenson, and L. Mendelsohn. Geophysical navigation technologies and applications. In *Position Location and Navigation Symposium, 2004. PLANS 2004*, pages 618–624. IEEE, 2004. 23

[273] O. C. Rodriguez. General description of the bellhop ray tracing program. *Physics Department Signal Processing Laboratory Faculty of Sciences and the University of the Algarve Tecnologia (Galician), Version*, 1, 2008. v, 4

[274] S. Rohou. *Reliable robot localization: a constraint programming approach over dynamical systems*. PhD thesis, Brest, 2017. viii, 2, 23, 95, 105, 115, 124, 192

[275] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, and S. M. Veres. Guaranteed computation of robot trajectories. *Robotics and Autonomous Systems*, 93:76–84, 2017. 105

[276] D. Y. Rokityanskiy and S. M. Veres. Application of ellipsoidal estimation to satellite control design. *Mathematical and Computer Modelling of Dynamical Systems*, 11(2):239–249, 2005. 111

[277] N. Roy, W. Burgard, D. Fox, and S. Thrun. Coastal navigation-mobile robot navigation with uncertainty in dynamic environments. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 35–40. IEEE, 1999. 90, 91, 92

[278] N. Roy and S. Thrun. Coastal navigation with mobile robots. In *Advances in Neural Information Processing Systems*, pages 1043–1049, 2000. vii, 90

[279] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011. 15

[280] I. T. Ruiz, S. De Raucourt, Y. Petillot, and D. M. Lane. Concurrent mapping and localization using sidescan sonar. *IEEE Journal of Oceanic Engineering*, 29(2):442–456, 2004. 18, 55

[281] I. T. Ruiz, D. M. Lane, and M. J. Chantler. A comparison of inter-frame feature measures for robust object classification in sector scan sonar image sequences. *IEEE Journal of Oceanic Engineering*, 24(4):458–469, 1999. 63

[282] I. T. Ruiz, Y. Petillot, D. M. Lane, and C. Salson. Feature extraction and data association for auv concurrent mapping and localisation. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2785–2790. IEEE, 2001. 18, 24, 63

[283] A. Sakai, D. Ingram, J. Dinius, K. Chawla, A. Raffin, and A. Paques. Pythonrobotics: a python code collection of robotics algorithms. 2018. vi, 75

[284] P. Salaris, D. Fontanelli, L. Pallottino, and A. Bicchi. Shortest paths for a robot with nonholonomic and field-of-view constraints. *IEEE Transactions on Robotics*, 26(2):269–281, 2010. 242

[285] J.m Salvi, Y. Petillo, S. Thomas, and J. Aulinas. Visual slam for underwater vehicles using video velocity log and natural landmarks. In *OCEANS 2008*, pages 1–6. IEEE, 2008. 16

[286] F. Salzenstein and W. Pieczynski. Unsupervised bayesian segmentation using hidden markovian fields. In *icassp*, pages 2411–2414. IEEE, 1995. 55

[287] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme. Visually guided landing of an unmanned aerial vehicle. *IEEE transactions on robotics and automation*, 19(3):371–380, 2003. 242

[288] J. Sawas, Y. Petillot, and Y. Pailhas. Cascade of boosted classifiers for rapid detection of underwater objects. In *Proceedings of the European Conference on Underwater Acoustics*, pages 1507–1516, 2010. 56

[289] M. S. I. Seddik. *Localization of a Swarm of Underwater Robots Using Set-Membership Methods*. PhD thesis, Brest, 2015. 132

[290] J. A. Sethian. Fast marching methods. *SIAM review*, 41(2):199–235, 1999. 81

[291] J. A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, 1999. 81

[292] K. Shoji. Quadtree decomposition of binary structuring elements. In *Nonlinear Image Processing II*, volume 1451, pages 148–158. International Society for Optics and Photonics, 1991. 156

[293] T. Siméon, J. P. Laumond, and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. *Advanced Robotics*, 14(6):477–493, 2000. 85, 86

[294] M. Sistiaga, J. Opderbecke, M. J. Aldon, and V. Rigaud. Map based underwater navigation using a multibeam echosounder. In *OCEANS'98 Conference Proceedings*, volume 2, pages 747–751. IEEE, 1998. 21

[295] J. Sliwka. *Using set membership methods for robust underwater robot localization*. PhD thesis, Ensta Bretagne, 2011. 130

[296] A. Stentz. Optimal and efficient path planning for partially-known environments. In *ICRA*, volume 94, pages 3310–3317, 1994. 75

[297] A. Stentz et al. The focussed dˆ* algorithm for real-time replanning. In *IJCAI*, volume 95, pages 1652–1659, 1995. 77

[298] A. Stentz, D. Fox, and M. Montemerlo. Fastslam: A factored solution to the simultaneous localization and mapping problem with unknown data association. In *In Proceedings of the AAAI National Conference on Artificial Intelligence*. Citeseer, 2003. 23

[299] W. E. Stevens and F. Brimberg. Automatic target detector, June 14 1977. US Patent 4,030,096. 54

[300] P. Svestka. *A probabilistic approach to motion planning for car-like robots*, volume 93. Unknown Publisher, 1993. 86

[301] P. Švestka and M. H. Overmars. Motion planning for carlike robots using a probabilistic learning approach. *The International Journal of Robotics Research*, 16(2):119–143, 1997. 86

[302] P. Švestka and M. H. Overmars. Probabilistic path planning. In *Robot motion planning and control*, pages 255–304. Springer, 1998. 84

[303] H. Takeda, C. Facchinetti, and J. C. Latombe. Planning the motions of a mobile robot in a sensory uncertainty field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1002–1017, 1994. 89

[304] H. Takeda and J. C. Latombe. Sensory uncertainty field for mobile robot navigation. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 2465–2472. IEEE, 1992. 86, 89, 141, 309

[305] M. R. Teague. Image analysis via the general theory of moments. *JOSA*, 70(8):920–930, 1980. 63

[306] F. J. C. M. Teixeira. *Terrain-aided navigation and geophysical navigation of autonomous underwater vehicles*. PhD thesis, 2007. 22, 23

[307] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998. 23

[308] S. Thrun. Probabilistic algorithms in robotics. *Ai Magazine*, 21(4):93, 2000. 190

[309] S. Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002. 22, 24, 175

[310] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006. 80

[311] B. Triggs. Motion planning for nonholonomic vehicles: An introduction. 1993. 72

[312] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995. 81

[313] J. D. Tucker, M. R. Azimi-Sadjadi, and G. J. Dobeck. Canonical coordinates for detection and classification of underwater objects from sonar imagery. In *Proc. of IEEE OCEANS 2007 Conference Europe*, pages 1–6, 2007. 54

[314] W. Tucker. The lorenz attractor exists. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 328(12):1197–1202, 1999. 111

[315] M. Tur, K. C. Chin, and J. W. Goodman. When is speckle noise multiplicative? *Applied optics*, 21(7):1157–1159, 1982. 40

[316] P. L. Tyack and C. W. Clark. Communication and acoustic behavior of dolphins and whales. In *Hearing by whales and dolphins*, pages 156–224. Springer, 2000. 35

[317] R.J. Urick. *Principles of Underwater Sound*. Peninsula Publishing, 1983. 2

[318] D. Vaman. Trn history, trends and the unused potential. In *Digital Avionics Systems Conference (DASC), 2012 IEEE/AIAA 31st*, pages 1A3–1. IEEE, 2012. 20

[319] J. Van Den Berg, P. Abbeel, and K. Goldberg. Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, 30(7):895–913, 2011. 93

[320] J. Van Den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research*, 31(11):1263–1278, 2012. 93

[321] J. Van Den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using differential dynamic programming in belief space. In *Robotics Research*, pages 473–490. Springer, 2017. 93

[322] P. Vandrish, A. Vardy, D. Walker, and O. A. Dobre. Side-scan sonar image registration for auv navigation. In *Underwater Technology (UT), 2011 IEEE Symposium on and 2011 Workshop on Scientific Use of Submarine Cables and Related Technologies (SSC)*, pages 1–7. IEEE, 2011. 18

[323] I. Vasilescu, C. Detweiler, M. Doniec, D. Gurdan, S. Sosnowski, J. Stumpf, and D. Rus. Amour v: A hovering energy efficient underwater robot capable of dynamic payloads. *The International Journal of Robotics Research*, 29(5):547–570, 2010. 13

[324] K. Vencatasamy, L. Jaulin, and B. Zerr. Secure a zone from intruders with a group robots. In *Marine Robotics and Applications*, pages 101–116. Springer, 2018. 5

[325] K. Vickery. Acoustic positioning systems. a practical overview of current systems. In *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings of the 1998 Workshop on*, pages 5–17. IEEE, 1998. 13

[326] Hurtós Vilarnau. *Forward-looking sonar mosaicing for underwater environments*. PhD thesis, Universitat de Girona, 2014. 19, 49, 52

[327] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001. 56

[328] P. Viola and M. Jones. Robust real-time object detection. *International journal of computer vision*, 4(34-47):4, 2001. 19

[329] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 56

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

[331] S. Wang, H. Zhang, K. Yang, and C. Tian. Study on the underwater geomagnetic navigation based on the integration of tercom and k-means clustering algorithm. In *OCEANS 2010 IEEE-Sydney*, pages 1–4. IEEE, 2010. 23

[332] C. W. Warren. A technique for autonomous underwater vehicle route planning. *IEEE Journal of Oceanic Engineering*, 15(3):199–204, 1990. 83

[333] A. Watt and F. Policarpo. *The computer image*. 1998. 47

[334] D. P. Williams. On adaptive underwater object detection. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4741–4748. IEEE, 2011. 55

[335] D. P. Williams. Auv-enabled adaptive underwater surveying for optimal data collection. *Intelligent Service Robotics*, 5(1):33–54, 2012. 28

[336] D. P. Williams. Fast target detection in synthetic aperture sonar imagery: A new algorithm and large-scale performance analysis. *IEEE Journal of Oceanic Engineering*, 40(1):71–92, 2015. 55, 56, 64

[337] D. P. Williams, F. Baralli, M. Micheli, and S. Vasoli. Adaptive underwater sonar surveys in the presence of strong currents. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2604–2611. IEEE, 2016. 2, 28, 71

[338] D. P. Williams and E. Coiras. On sand ripple detection in synthetic aperture sonar imagery. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1074–1077. IEEE, 2010. 64

[339] D. P. Williams and J. Groen. A fast physics-based, environmentally adaptive underwater object detection algorithm. In *OCEANS, 2011 IEEE-Spain*, pages 1–7. IEEE, 2011. 55

[340] S. B. Williams, O. R. Pizarro, M. V. Jakuba, C. R. Johnson, N. S. Barrett, R. C. Babcock, G. A. Kendrick, P. D. Steinberg, A. J. Heyward, P. J. Doherty, et al. Monitoring of benthic reference sites: using an autonomous underwater vehicle. *IEEE Robotics & Automation Magazine*, 19(1):73–84, 2012. 28

[341] P. Woock. Survey on suitable 3d features for sonar-based underwater navigation. In *OCEANS, 2012-Yeosu*, pages 1–6. IEEE, 2012. 19

[342] J. Yao, Y. Yao, V. Kreinovich, P. P. da Silva, S. A. Starks, G. Xiang, and H. T. Nguyen. Towards more adequate representation of uncertainty: from intervals to set intervals, with the possible addition of probabilities and certainty degrees. In *Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence). IEEE International Conference on*, pages 983–990. IEEE, 2008. 167

[343] B. Zerr, E. Bovio, and B. Stage. Automatic mine classification approach based on auv manoeuvrability and the cots side scan sonar. In *Proceedings of Goats 2001 Conference, La Spezia, Italy*, 2001. 28

[344] B. Zerr, G. Mailfert, A. Bertholom, and H. Ayreault. Sidescan sonar image processing for auv navigation. In *Oceans 2005-Europe*, volume 1, pages 124–130. IEEE, 2005. 18

[345] B. Zerr, B. Stage, and A. Guerrero. Automatic target classification using multiple sidescan sonar images of different orientations. *SACLANT CEN Memorandum SM-309*, 1997. 63

[346] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57, 2001. 54

[347] J. Zhao, J. Yan, H. Zhang, and J. Meng. A new radiometric correction method for side-scan sonar images in consideration of seabed sediment variation. *Remote Sensing*, 9(6):575, 2017. 46

[348] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and vision computing*, 21(11):977–1000, 2003. 18, 155

Titre : Etude et développement de solutions de relocalisation d'objets sous-marins par des véhicules sous-marins hétérogènes.

Mots clés : Sous-marin, robotique, planification, incertitudes, graphe sonar, détection

Résumé : Dans le contexte de guerre des mines (MCM), il est important de revisiter les objets potentiellement dangereux afin de les identifier et les neutraliser s'il s'agit effectivement de mines. Cette dangereuse tâche était habituellement réalisée par des plongeurs démineurs qui sont petit à petit remplacés par des drones sous-marins. Le design « low cost » des robots de revisite/destruction de mines ne permet en général pas de garantir la revisite d'une cible géolocalisée en allant droit dessus. De plus, le robot pourrait commencer sa mission sous-marine à une distance éloignée et l'absence de positionnement GPS en environnement sous-marin imposent l'élaboration d'une stratégie afin de garantir la revisite de la cible. En se basant sur des informations *a priori* de l'environnement et notamment la présence de points de repère (amers), le problème est résolu par la planification d'une stratégie à suivre en prenant en compte les incertitudes de déplacement inhérentes en milieu sous-marin.

Dans un contexte d'erreurs bornées, une approche ensembliste est proposée.
Premièrement, en se basant sur la position et la forme des amers, ainsi que la zone visible du capteur embarqué, les cartes de recalages sont construites afin de définir les poses du robot qui permettent de détecter les differents amers afin de réduire l'incertitude de position du robot. Deuxièmement, en se basant sur un modèle paramétrique de déplacement avec des paramètres incertains, une stratégie haut-niveau est proposée à travers l'optimisation d'un graphe. La stratégie consiste à naviguer entre les cartes de recalage afin de réduire l'incertitude de position du robot et finallement garantir la revisite de la cible souhaitée.

Title: Study and development of relocation solutions of underwater objects by underwater heterogeneous underwater

Keywords: Underwater, robotics, motion planning, uncertainty, graph, sonar, detection

Abstract: In the Mine Counter Measure (MCM) context in the underwater environment, it is vital to revisit some potentially dangerous objects to identify and neutralize them if they are actually mines. This dangerous task was usually performed by human divers but more and more it is conducted by unmanned underwater robots. Due to the low cost design of the revisit/mine-killer robot, going straightforward to the geolocalized suspicious object does not guarantee that the robot will redetect it. Moreover the robot may dive at a far position from the target and the lack of absolute positioning system in underwater environment demands a strategy to follow to guarantee the revisit of this target. Based on *a priori* information in the working area and especially the presence of geolocalized landmarks, the problem is solved as a motion planning problem considering uncertainties due to the increasing error when navigating underwater.

In the context of bounded errors, the problem is solved in a set-membership manner.
Firstly, based on the location and the shape of the landmarks, and on the visibility area of the sensor embedded, the registration maps are computed indicating the sets of robot poses to detect the different landmarks considered in order to reduce the uncertainty on the robot position. Secondly, based on a parametric motion model with uncertain parameters, an high level strategy is provided through a graph optimization. The strategy consists in navigating between the registration maps to reduce each times the uncertainty in position of the robot and finally to guarantee the reachability of a goal area corresponding to the redetection of the target.