Simultaneous localization and mapping in unstructured environments : a set-membership approach

Benoît Desrochers

To cite this version:

Benoît Desrochers. Simultaneous localization and mapping in unstructured environments : a set-membership approach. Robotics [cs.RO]. ENSTA Bretagne - École nationale supérieure de techniques avancées Bretagne, 2018. English. NNT : 2018ENTA0006 . tel-02365087

HAL Id: tel-02365087
https://theses.hal.science/tel-02365087

Submitted on 15 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simultaneous Localization and Mapping in Unstructured Environments

Benoît Desrochers

To cite this version:

Benoît Desrochers. Simultaneous Localization and Mapping in Unstructured Environments. Automatic. Université Bretagne Loire, 2018. English. tel-01850567

HAL Id: tel-01850567

https://hal.archives-ouvertes.fr/tel-01850567

Submitted on 27 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE
BRETAGNE
LOIRE

THESE / ENSTA Bretagne
sous le sceau de l'Université Bretagne Loire
pour obtenir le titre de

DOCTEUR de l'ENSTA Bretagne
Spécialité Robotique

Ecole Doctorale MathSTIC - ED601

présenté par

Benoît Desrochers

Préparé à ENSTA Bretagne (Lab-STICC, FR)
et à DGA Techniques Navales (FR)

Simultaneous Localization and Mapping in Unstructured Environments

A Set-Membership Approach

Thèse soutenue le 24 Mai 2018
devant le jury composé de:

Luc JAULIN, directeur de thèse
Professeur des Universités,, ENSTA Bretagne (Lab-STICC), Brest, FR.

Sylvie GALICHET, rapporteur
Professeur des Universités, LISTIC, Annecy le Vieux, FR.

Warwick TUCKER, rapporteur
Professeur des Universités, Université d'Uppsala, SE.

Simon LACROIX, examinateur
Directeur de recherches, LAAS-CNRS, Toulouse, FR.

Vincent CREUZE, examinateur
Maître de conférences, LIRMM, Monpellier, FR.

Fabrice LE BARS, examinateur
Maître de conférences, ENSTA Bretagne (Lab-STICC), FR.

Alain BERTHOLOM, examinateur
Ingénieur, Expert robotique navale, DGA TN, FR.

Nicolas DROGI, examinateur
Ingénieur, Sous directeur technique, DGA TN, FR.

Simultaneous Localization and Mapping in Unstructured Environments; A Set-Membership Approach

Benoît Desrochers

24 May 2018

Acknowledgments

Cette thèse a été réalisée en collaboration avec le département Drones Navals et Guerre des Mines de la Direction Générale de l'Armement - Techniques Navales sur le site de Brest et l'Ensta Bretagne. Je souhaite remercier l'ensemble de ces acteurs pour m'avoir fourni la possibilité de réaliser ces travaux de recherche. Mes premières pensées vont bien évidemment aux personnes qui m'ont encadré, conseillé et soutenu.

En premier lieu, je tiens à remercier Luc Jaulin pour avoir été un excellent directeur de thèse. Ces travaux n'auraient pu être aussi aboutis sans son aide précieuse, sa grande disponibilité et nos nombreuses discussions !

Je remercie également les membres de mon jury pour le temps qu'ils m'ont consacré ainsi que l'attention et l'intérêt qu'ils ont montrés lors de la relecture de mon manuscrit et ma présentation : leurs corrections, questions et remarques pertinentes et constructives m'ont permis d'améliorer ce manuscrit et de valoriser mon travail.

En particulier, je tiens à remercier mes rapporteurs : Sylvie Galichet, pour avoir apporté un point de vue issu de la communauté de la logique flou et Warwik Tucker pour son approche théorique, Simon Lacroix d'avoir présidé ce jury et Vincent Creuze, Fabrice Le Bars, Nicolas Drogi, Alain Bertholom d'avoir endossé les rôles d'examinateurs.

Je remercie également mes collègues de DGA Tn, qui m'ont apporté une vision pragmatique sur les problématiques de positionnement sous-marin et en particulier toute l'équipe du groupe Drone navale et Guerre des mines de Brest. Pour la partie théorique, je remercie vivement les différents personnels, doctorants, de l'Ensta Bretagne qui m'ont aidé durant ces 3 années. En particulier, je tiens à remercier Simon R., Thomas L. M., qui m'ont accueilli dans leur bureau lors de mes passages à l'Ensta Bretagne.

Je tiens à remercier tout particulièrement mes relecteurs Quentin B, Simon R. pour leur travail minutieux.

A titre personnel, je tiens à remercier l'ensemble de ma famille et en particulier mes parents et beaux-parents pour m'avoir encouragé durant mes nombreuses heures de travail. Et un grand merci pour le pot de thèse bien sûr !

Enfin, et par-dessus tout, je souhaite remercier ma femme Charlotte pour sa tendresse et son soutien indéfectible durant ces 3 années.

Benoît

Contents

List of Acronyms

AUV	Autonomous Underwater Vehicle
GPS	Global Positioning System
SLAM	Simultaneous Localization And Mapping
IMU	Inertial Measurement Unit
DVL	Doppler velocity log
INS	Inertial Navigation System
SSS	Side Scan Sonar Sensor
EKF	Extended Kalman Filter
PF	Particle Filter
GNSS	Global Navigation Satellite System
SONAR	SOund Navigation And Ranging
CN	Constraint Network
SVCN	Shape-Valued Constraint Network
SRAC	Shape Registration And Carving
LIDAR	LIght Detection And Ranging

Chapter 1

Introduction

This thesis focuses on new methods, based on Interval Analysis, able to deal with constraints that involve uncertain sets. Many engineering problems, such as robot localization, global motion path-planning or control, can be modeled by constraint networks. A *Constraint Network* (CN) is composed of a set of variables that belong to some domain and which must satisfy a given number of elementary facts and rules: so-called constraints. In classical formulations [Mackworth, 1977], variables are vectors of real numbers and domains are subsets of $\mathbb{R}^n$.

In this manuscript, we consider a larger class of problems that involve subsets of $\mathbb{R}^n$ as variables too. As the main application, we deal with the *Simultaneous Localisation And Mapping* (SLAM) problem in unstructured environments, *i.e* that cannot be described by geometrical features. Sensor readings will not be depicted by a model that can be described by parameters of $\mathbb{R}$ but directly by shapes or subsets of $\mathbb{R}^n$.

The uncertainties will then be handled, in a set-membership approach, by an interval of sets called *Thick Set*.

1.1 Motivation and background

Autonomous vehicles are used in support of humans to perform dull, dirty and dangerous tasks. The more dangerous and challenging the environment, the more autonomous they must be.

To be fully autonomous, vehicles must be able to perform long-term and long-range missions with little or no human intervention. They must position themselves with meaningful and reliable bounds on their positioning uncertainty that can be used for decision-making purposes. For instance, in the context of seabed mapping using an Autonomous Underwater Vehicle (AUV), the positioning uncertainty must be monitored, in order to assess that the mission requirements are successfully achieved. This can concern the assessment of the covered area by the vehicle before returning to the mother-ship, or that the positioning errors on payload data are acceptable with respect to the mission requirements. If not, the initial plan must be modified and specific actions need to be taken.

To localize itself, a modern autonomous vehicle relies on an Inertial Navigation System (INS) that merges proprioceptive measurements (acceleration, velocity, attitude, depth, ...) in order to continuously determine, via dead reckoning methods, the position, orientation and velocity of the vehicle. INS provides a good short-term accuracy but accumulates errors that grow unbounded with time and distance traveled. Consequently, this drift needs to be fixed with, for instance, data from *Global Navigation Satellites Systems* (GNSS). However, even if these systems can provide a centimeter position accuracy, they are not always available and remain sensible to jamming and spoofing. Another solution is to use physical features of the environment observed by embedded sensors, such as camera, LIDAR or sonar, to provide an onboard, real-time estimated, positioning solution.

1.1.1 Mapping

Environments, structured or not, are composed of features which are gathered in a map. In this thesis, structured environments are taken to mean terrains that can be described by geometric structures such as lines, segments, landmarks, ... For instance, man-made environments are generally structured. On the contrary, unstructured environments are composed of shapes that cannot be parameterized. This is the case of underwater environments since there are few natural occurring geometrical landmarks.

It exists several methods to depict the environment, but according to [Thrun, 1998] two main representations are commonly considered: topological and metric maps.

A topological map refers to a map that has been simplified so that only vital information remains and unnecessary details have been removed. Only the neighborhood relation between places is maintained. The tube map of the Paris underground is an example of such representation where only the connections between stations are depicted without considering scales and distances.

On the contrary, metric maps assign coordinates to objects of the environment that allows computing distances or angles between them. This representation can be classified into two categories [Thrun, 2002], which are referred to as feature-based map and location-based map.

1.1.1.1 Feature-based maps

Feature-based maps are composed of a set of features together with their Cartesian location. The map has a parametric structure where the features are points, segments, corners, or any other parametric shapes [Castellanos et al., 2001]. In the case of the SLAM problem, feature parameters are included among state variables [Dissanayake et al., 2001, Montemerlo et al., 2003] and are estimated concurrently with the pose of the vehicle. Probabilistic methods such as Kalman filtering, Bayesian estimation or particle filtering [Thrun, 2002] can be used. Interval-based methods have also been proved to be efficient to solve the feature-based SLAM [Le Bars et al., 2010, Jaulin, 2016].

However, geometrical features are spatially localized and do not include information on their environment. A particular attention must be paid when solving data association ambiguities between measurements and features [Neira and Tardos, 2001]. Loop closure methods [Ho and Newman, 2006, Aubry et al., 2013], that aim at finding places that have been previously visited, can provide a prior data association. This critical point made the localization of featured map a challenging problem in complex and unstructured environments like these faced in outdoor or underwater domains.

1.1.1.2 Location-based maps

On the contrary, location-based maps offer a label to any location in the world. A classical location-based map representation is known as occupancy map [Elfes, 1987] (also called pose-based map). They assign to each point of the world an occupancy value (a Boolean number or a probability of occupancy) that specifies whether or not a pose is occupied by an obstacle. For computational reasons, occupancy maps are most of the time based on a regular grid. Each cell of the grid stores the probability, between 0 and 1, to be occupied by an obstacle. The main difficulty with a grid-based representation is to correctly handle both the sensor and the pose uncertainties when updating the grid. This is especially the case when the map is only perceived through sensors, such as sonars [Thrun, 2003] or cameras, that partially observe the environment. To handle this difficulty properly, we introduce the notion of *shape*.

Shape. A *shape* is a subset of $\mathbb{R}^q$, where q is smaller than the dimension of the workspace. In this thesis, a shape will always be a subset of dimension 2. A shape corresponds to a part of the environment that is captured by some sensors of the robot. As shapes may be uncertain, we will also introduce *shape intervals* (a particular case of *thick sets*) to represent this uncertainty in a set-membership framework. A shape separates the environment into two complementary parts, associated with physical features that can be measured by vehicle's sensors.

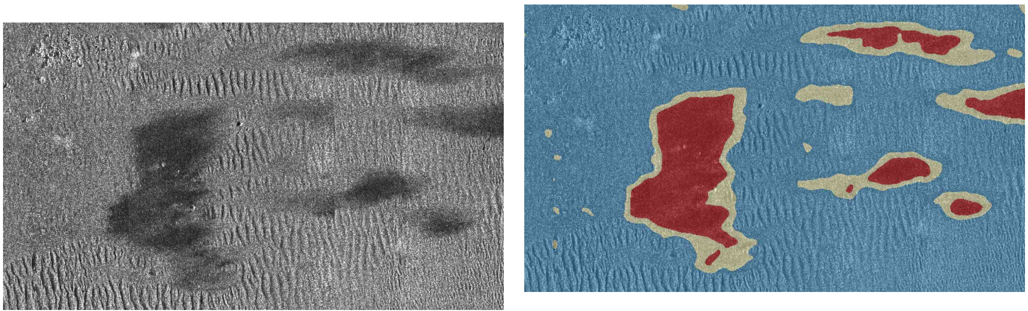
For examples, these features can be:

- the part of the space that can be proved to be free of obstacle. For instance, from range measurements acquired by laser or sonar sensors;
- contour map from *Digital Elevation Map* (DEM) that can be used to partition

the space based on a given elevation value;

- segmented images based on texturing information. For instance, from data collected by a side scan sonar, which provides an acoustic image of the seabed, it is possible to partition the space based on homogeneity properties [Picard et al., 2015] or a seabed classification [Leblond et al., 2005].

Given a particular feature, labeled as a class $\mathbb{A}$, the notion of uncertain shape, called *shape interval*, allows us to classify the space into three categories: (1) parts that can be classified for sure as $\mathbb{A}$, (2) those that can be classified as not $\mathbb{A}$ and (3) parts where nothing can be said, called the *penumbra*. From the raw image, Figure 1.1a, a segmentation algorithm separates the mud (darker area) and non-mud area. The boundaries, between these two classes, cannot be detected accurately and are assumed to be enclosed in the penumbra. The area outside the image, which has not been observed, also belongs to the penumbra. The second part of this manuscript deals with the interval shape (or thick set) representation.



(a) Side scan sonar image

(b) Shape Interval or thick set representation

Figure 1.1: The raw image 1.1a is segmented and interpreted as an uncertain shape 1.1b.

1.1.2 Shape based localization and mapping

From the previous set-based map representation, the localization problem needs to be defined. From a theoretical point of view, a localization problem can be described by the following set of equations.

$$\left\{\begin{array}{rcll} \dot{\mathbf{x}}(t) & = & \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) & \text{(evolution equation)} \\ \mathbb{Z}(t) & = & \mathbf{g}(\mathbf{x}(t), \mathbb{M}) & \text{(observation equation)} \\ \mathbf{x}(0) & \in & \mathbb{X}_0 & \text{(initial state)} \end{array}\right. \tag{1.1}$$

where $t \in \mathbb{R}$ is the time, $\mathbf{x} \in \mathbb{R}^n$ is the state vector of the vehicle, $\mathbf{u} \in \mathbb{R}^m$ is the input vector (in general associated with proprioceptive sensors), $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the evolution function. The set $\mathbb{M}$ is the map which is a subset of $\mathbb{R}^q$, where q is the dimension of the workspace (two or three in practice). The map $\mathbb{M}$ corresponds to a simplified representation of the environment. The extension to maps composed of several classes is considered as trivial and is not treated in our formalism, for clarity. The set $\mathbb{Z}(t) \subset \mathbb{R}^r$, where $r \leq q$, is a shape that represents a view of the map $\mathbb{M}$, is collected by some exteroceptive sensors (for instance a camera, a sonar, a telemeter) and expressed in the robot frame. In practice, due to the uncertainties, the measurement process does not provide us $\mathbb{Z}(t)$ directly, but an interval shape $[\mathbb{Z}](t)$ containing $\mathbb{Z}(t)$. The fact that the measurement corresponds to a shape instead of a vector is not classical in the SLAM community and is related to the location-based SLAM formalism. The treatment of the shape-SLAM is the main goal of this thesis. The observation function $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}^r$ links the measured shape $\mathbb{Z}(t)$ to the map $\mathbb{M}$. The initial state at $t = 0$ is known to belong to an initial domain $\mathbb{X}_0 \subset \mathbb{R}^n$. The functions $\mathbf{f}$ and $\mathbf{g}$ may be non-linear and uncertain.

With respect to the unknowns of the problem, three cases can be considered: the initial localization, the position tracking, and the Simultaneous Localization And Mapping problem.

The *initial localization* problem corresponds to the situation where only the map $\mathbb{M}$ is available and the initial state must be recovered. In the probabilistic context, in the specific case where the shape $\mathbb{Z}(t)$ is a vector of $\mathbb{R}^q$ or geometric features, this problem has been successfully addressed with Particle Filter (PF) methods such as Monte Carlo Localization (MCL) [Dellaert et al., 1999] or Rao-blackwellised Particle Filter (RBPF). However, proving the convergence and robustness of these algorithms remains a challenging task. On the contrary, interval methods [Jaulin et al., 2001a, Clérentin et al., 2008, Gning, 2006, Langerwisch, 2014] have been proved to be able to deal efficiently with this situation. The thesis [Guyonneau et al., 2013] provides an overview of set-membership methods applied to global localization.

The *pose tracking problem* arises when both the map $\mathbb{M}$ and the initial condition $\mathbb{X}_0$ are known. It consists of keeping track of the vehicle's current position by matching sensor readings with the map. It can be regarded as a special case of the initial localization problem where the search space is restricted. Probabilistic methods, such as the Kalman filter [Kalman, 1960] in the linear case, or the particle filter [Jensfelt and Christensen, 2001] in the non-linear case, have been used for a long time. However, due to changes in the environment or sensor reading uncertainties, the tracked solution can be lost. These algorithms may diverge and need to be initialized with a new starting position, that needs to be found.

Finally, the situation where $\mathbb{M}$ needs to be estimated corresponds to the SLAM problem[Leonard et al., 1992]. For an autonomous robot, moving in an unknown environment, the SLAM problem is to build a map while simultaneously using it to compute its location. Note that the map is built relatively to the initial state which, if undetermined, is fixed arbitrarily. A significant amount of work has been completed around this topic, still subject to further research [Leonard et al., 2002, Thrun, 2002].

Underwater SLAM [Hidalgo and Braunl, 2015] and terrain based navigation [Melo and Matos, 2017] remain challenging topics for autonomous underwater vehicles in long-term operations due to the limitations of subsea localization sensors and real-time payload exploitation. For these reasons, set-membership methods will be used since shape measurements are difficult to be handled by probabilistic methods. A vector of $\mathbb{R}$ can be easily modeled by a random variable, but it is not the case for a subset of $\mathbb{R}^n$. Some formulations of the SLAM problem have been proposed using Random Finite Sets [Molchanov, 2005b, Mullane et al., 2011] but remains limited to punctual features.

1.1.3 Probabilistic and set-membership approaches

In this thesis, the set-membership approach has been chosen to deal with localization issues. The following paragraph highlights the main difference between probabilistic methods and set-membership ones.

Probabilistic methods are largely used in mathematics and engineering and has been proved to be efficient. A random vector $\mathbf{x}$ of $\mathbb{R}^n$ is described (or encoded) using a density probability function: $\pi : \mathbb{R}^n \longrightarrow \mathbb{R}^+$, such as $\int_{\mathbb{R}^n} \pi(\mathbf{x})d\mathbf{x} = 1$. The variable is contained in a set $\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \pi(\mathbf{x}) \neq 0\}$, called *support*, which also provides a description of how this variable is spread over $\mathbb{X}$.

With a set membership approach, the variable is only represented by a set such as $\mathbf{x} \in \mathbb{X}$. The domain $\mathbb{X}$ can enclose inconsistent values, but all consistent values for $\mathbf{x}$ should be inside of $\mathbb{X}$. Consequently, the set-membership representation is poorer than the probabilistic approach but needs little knowledge about the statistical properties of variables and allows to handle a larger class of problems.

The following example explains the difference between the two approaches.

Example 1.1. Let us take three random variables x, y, z of $\mathbb{R}$ linked with the constraint $z = x + y$. From statistical properties on x and y, we aim at deducing those of z with probabilistic and set membership methods.

Probabilistic approach. Assume that x and y are uniform such as $\pi(x) = 1/5$ over $[0, 5]$ and $\pi(y) = 1/2$ over $[1, 3]$. Computing z requires the knowledge of the joint probability density for x and y. Assume for instance that x and y are independent, so the density of the couple is uniform with $\pi(x, y) = 1/10$ over the box $[0, 5] \times [1, 3]$. The Probability Density Function (PDF) for z is given by:

$$\pi(z) = \int_{\mathbb{R}} \pi_x(x) \cdot \pi_y(z - x)dx \qquad (1.2)$$

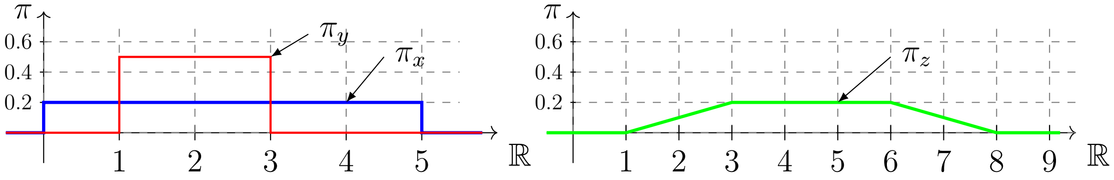


Figure 1.2: Density probability function of random variable x and y.

which gives us:

$$\pi(z) = 0 \text{ if } z \leq 1, \tag{1.3}$$

$$\pi(z) = \frac{1}{10}(z-1) \text{ if } z \in [1,3], \tag{1.4}$$

$$\pi(z) = \frac{1}{5} \text{ if } z \in [3,6], \tag{1.5}$$

$$\pi(z) = \frac{1}{10}(-z+8) \text{ if } z \in [6,8], \tag{1.6}$$

$$\pi(z) = 0 \text{ if } z \geq 8. \tag{1.7}$$

The function $\pi(z)$ is depicted on Figure 1.2. Note that the integral calculus is generally much harder even in the particular cases where x and y are Gaussian or uniform.

Set-membership approach. The uncertainty associated to x et y is given by intervals: $[x] = [0,5]$ and $[y] = [1,3]$. The random variable $z = x + y$ is contained in the interval $[z] = [0+1, 5+3] = [1,8]$. So the set membership approach leads to an easier computation with less hypotheses.

Note that representing a random vector x by a set $\mathbb{X}$ does not mean that its density function $\pi(x)$ is uniform over $\mathbb{X}$. It only supposes that x is in $\mathbb{X}$ or, in other words, that the support of $\pi(x)$ is included in $\mathbb{X}$. In the Example 1.1, the random variable z is described by the interval $[1,8]$ but is not uniform on $[1,8]$.

As illustrated by the Example 1.1, the computation of $\pi(z)$ is possible because the PDF has been assumed to be uniform and with a linear dependency between x, y and z. When the relation between variables is non-linear, Equation 1.2 becomes hard to solve in the general case. Moreover, the independence between variables must always be assumed mainly because the join law between variables is rarely available or because it does not correspond to the reality. It frequently happens when dealing with data from real sensors. However, when probabilistic properties are well-known, for instance when the problem is linear Gaussian, probabilistic methods are efficient and outperform largely set-membership methods.

In other cases, strong hypothesis must be done over the probabilistic properties of the problem and no proof can be done on the results. Filters, such as extended Kalman filters or particle filters, can converge to false or inconsistent values without raising any error. For this last point, interval methods allow to correctly, with realistic hypothesis, handle uncertainties.

In this manuscript, probabilistic methods will not be presented or compared. With respect to the bounded error hypothesis, set-membership methods provide reliable and guaranteed results. They can be used concurrently with probabilistic approaches to find the best estimate that lays into a guaranteed solution set. In other words, the hard part of strongly non-linear problems, such as data association or initial localization, can be made using interval analysis. Once done, probabilistic approaches can be used to efficiently find the best solution that lies in a guaranteed domain of validity.

1.2 Contributions and Outlines

The main contribution of this thesis is to consider that both the map and observations can be described by shapes, associated with physical features of the environment. We aim at tackling both initial localization and SLAM problems through a formalism that is not commonly used in the literature. For that purpose, the notion of *interval shape* is introduced, with its associated arithmetics and the definition of the uncertain set inversion problem. A specific contractor is developed to solve the so-called shape registration and carving problem.

Another contribution concerns a way to build minimal contractors and separators consistent with equations of the form $\mathbf{y} = \mathbf{f}(\mathbf{x})$. It allows us to introduce minimal operators associated with the change from polar to cartesian coordinates when range and bearing measurements are collected.

In this manuscript, we chose to only focus on the shape-based SLAM. For this reason, two main contributions, not directly related to the SLAM problem, have been placed in Appendix A and B. The first one is dedicated to a specific algorithm that solves the uncertain transformation of an interval shape when the uncertain function can be enclosed by an interval of functions. The contribution is dedicated to the characterization of the area explored by a robot when the dimension of the visible space at time t is smaller than that of the workspace. Both of them have required a huge amount of work and helped us to mature the concepts needed to properly formalise and solve the *Shape SLAM* problem.

This thesis is decomposed into two parts. The first one, from Chapters 2 to 4, is dedicated to the classical interval analysis used to solve problems involving variables that are vectors of $\mathbb{R}^n$. The second part, composed with Chapters 5 and 6, focuses on constraints that involve shapes as variables.

In Chapter 2, basic notions of interval analysis and contractors are introduced. A special focus on the *separator* operator is done in this chapter. This operator, introduced in [Jaulin and Desrochers, 2014], allows simple and efficient set manipulation and characterization.

In Chapter 3, we introduce a new theorem that can be used to build minimal contractors consistent with equations, and another new theorem to derive an optimal separator from a minimal contractor. As an application, we focus on the polar constraint associated with the change between Cartesian coordinates and Polar coordinates. We illustrate our method on the localization problem of an actual underwater robot where both range and goniometric measurements of landmarks are collected. The material of this chapter is published in [Desrochers and Jaulin, 2016a].

In Chapter 4, a new type of constraint is introduced, that links shapes *via* a transformation depending on some parameters. Based on the Polar separator, introduced in Chapter 3 that allows handling sectoral sensor, a first initial localization problem on an unstructured map is presented. This chapter makes the transition from classical constraint networks to shape-valued constraint networks where variables become shapes. This work is published in [Desrochers and Jaulin, 2017c].

Chapter 5 deals with intervals of shapes, which is a particular type of *Thick Sets.* Shape intervals provide an interval based representation of uncertain shapes. An uncertain shape $\mathbb{Y}$ is bracketed between an inner shape $\mathbb{Y}^-$ and an outer shape $\mathbb{Y}^+$. After defining operations that can be applied to shape intervals, tools used to deal with constraint networks involving shapes as variables are introduced. A special focus is done on the set inversion problem involving an uncertain function that depends on parameters. The shape registration constraint, presented in Chapter 4 is then extended to the uncertain case.

Chapter 6 deals with the SLAM problem where the observations are not based on scalar values or equations, but are subsets of $\mathbb{R}^n$. Two applications are considered. In the first one, the initial localization problem presented in Chapter 4 is extended to the context of the SLAM, where the map needs also to be estimated. In the second application, measurements taken by an imaging sensor are depicted by an uncertain shape. A SLAM method, based on this set-membership representation is then illustrated. This chapter summarizes the targeting application of this thesis.

Chapter 7 concludes this thesis.

Appendix A deals with the set inversion problem $\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$ in the case where the function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and the set $\mathbb{Y}$ are both uncertain. The uncertainty is treated under the form of intervals. More precisely, for all $\mathbf{x}$, $\mathbf{f}(\mathbf{x})$ is inside an interval of functions $[\mathbf{f}](\mathbf{x})$ and the uncertain set $\mathbb{Y}$ is depicted by a thick set. The introduction of new tools such as *thick intervals* and *thick boxes* will allow us to propose an efficient algorithm that handles the uncertainty of sets in an elegant and efficient manner. Some elementary test-cases that cannot be handled easily and properly by existing methods show the efficiency of the approach. This work is published in [Desrochers and Jaulin, 2017d].

Appendix B deals with the guaranteed characterization of the part of the space that has been explored by a robot. The main difficulty of the problem is to take into account the uncertainty associated with the trajectory and the fact that the dimension of the visible space at time t may be smaller than that of the workspace. An example involving an experiment made with an actual underwater robot is presented in order to illustrate the efficiency of the approach (published in [Desrochers and Jaulin, 2017b]).

1.3 PyIbex: an Interval Library

To support and capitalize the developments done during this thesis, a Python library called *pPyIbex* has been developed. The source code is available on GitHub at

https://github.com/benEnsta/pyIbex.

Based on the contractor programming library ibex-lib, it aims at providing a python interface to basic functionality, such as interval, contractor, separator, and adds extra-modules dedicated to tools introduced in this thesis.

PyIbex is presented as a multi-platform Python module. It was used by IAMOOC, a MOOC on Interval Analysis with application to parameter estimation and robot localization.

The documentation and examples are given in

http://benensta.github.io/pyibex

The figures of the manuscript are generated using VIBes [Drevelle and Nicola, 2014].

Part I

Interval Analysis

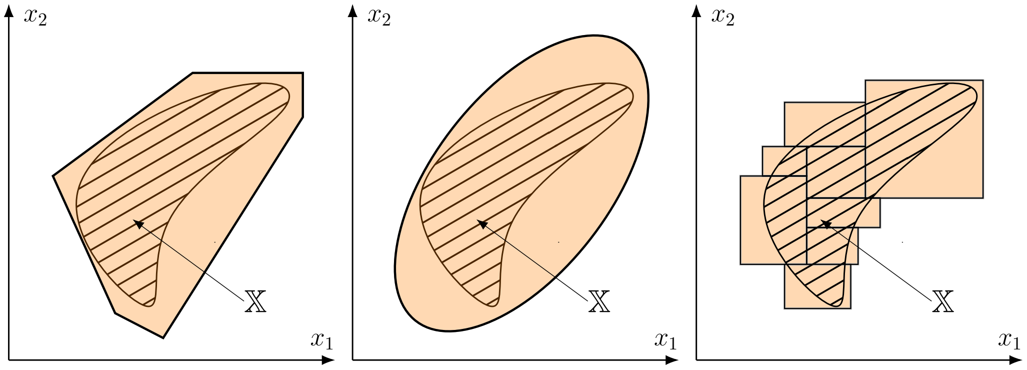
Chapter 2

Interval Analysis

Contents

2.1 Introduction

Our choice is to address robotic problems with set-membership methods. Basically, the solution of the problem, such as the estimating of the position of the robot, is characterized by a set defined by non-linear constraints. This set needs to be represented by an over approximation which can be manipulated by the computers. As illustrated in Figure 2.1, in a set membership framework, several approximations can be found in the literature such as zonotopes or polyhedral enclosures [Combastel, , Walter and Piet-Lahanier, 1989], ellipsoids [Rokityanski and Veres, 2005], intervals or subpavings [Jaulin and Walter, 1993c]. In this thesis, we will focus on the last representation which is related to interval analysis.



(a) Polyhedral enclosure (b) Ellipsoidal enclosure (c) Subpaving enclosure

Figure 2.1: Examples of different set-membership enclosures of a set $\mathbb{X}$.

Interval analysis has become over the past few years a strong alternative to traditional probabilistic approaches [Thrun and Montemerlo, 2005] to solve complex non-linear systems of equations.

Interval analysis makes it possible to characterize a set of $\mathbb{R}^n$ defined by constraints. For instance, the set may correspond to:

- the set of all parameters that are consistent with some interval measurements [Lévêque et al., 1997, Gning and Bonnifait, 2006, Kreinovich et al., 1997, Gning et al., 2013, Chabert and Jaulin, 2009a],
- to the set of all configuration vectors such that a robot does not meet any obstacles [Porta et al., 2007, Jaulin, 2001], to the set of all parameter vectors of a controller such that the closed loop system is stable [Wan et al., 2009, Didrit et al., 1995],
- to set all calibration parameters [Daney et al., 2006, Ramdani and Poignet, 2005]
- to attractors of dynamical systems [Tucker, 1999].

More formally, a mathematical problem can be formulated by means of a *Constraint Network* (CN) where a set of *variables*, that belong to some *domains*, must satisfy simultaneously a given number of elementary facts and rules: so-called *constraints* [Mackworth, 1977].

A *CN* is defined as a triple $\langle \mathbf{X}, \mathbb{D}, \mathcal{L} \rangle$ where

$$\begin{array}{rl} \mathbf{X} = \{x_1, \ldots, x_n\} & \text{is a set of variables,} \\ \mathbb{D} = \{\mathbb{X}_1, \ldots, \mathbb{X}_n\} & \text{is a set of the respective domains of values, and} \\ \mathcal{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_m\} & \text{is a set of constraints.} \end{array}$$

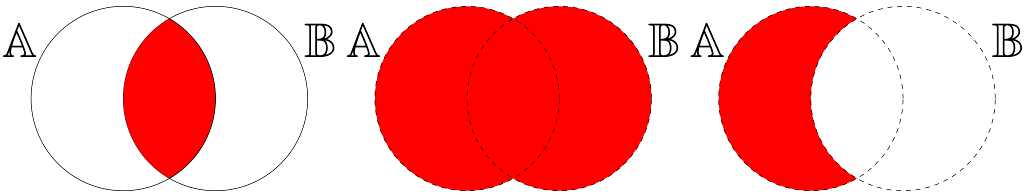
The problem to be considered here is to find the smallest sub-domains of the $\mathbb{X}_i$ for variables x_i consistent with all constraints. Note that, in this part, variables are real numbers or vector of $\mathbb{R}^n$ and domains are intervals, boxes, or sets. Later, the class of variables to be handled will be extended: they could correspond to shapes and their domains will be *shape intervals.*

Remark 2.1. In the literature, there exists a confusion between *Constraints Networks* (CN) and *Constraint Satisfaction Problem*, or CSP for short. Stress that a CSP is composed of a CN (which is the model) and a question. Solving a CSP is done by providing a solution of the corresponding CN, which is not exactly what we want to do here. Instead, we want to find an enclosure of all solutions of the CN.

The purpose of this chapter is to familiarize the reader with interval analysis concepts. In the first section, elementary notions on set theory are introduced. Then, in Section 2.3, basics of interval arithmetic are provided. In order to get an inner and an outer approximation of sets defined with constraints, contractors and separators are presented in Section 2.4. With respect to the applications of this manuscript, common separators associated to images, subpavings or polygons are introduced in Section 2.5. We will also give two applications illustrating how these operators can be combined to solve robotics problems.

2.2 Set Theory

The fundamental notion that is used at the heart of this thesis is the notion of *set.* A set is a well-defined collection of distinct objects. The objects that make up a set (also known as the set's elements or members) can be anything: numbers, people, letters of the alphabet, graphs, other sets, and so on. The set theory, that aims at studying sets, was developed at the end of the 19th century by George Cantor and Richard Dedekind [Cantor, 1847]. In this manuscript, sets are denoted in blackboard bold font (*mathbb*). The set of real numbers is denoted $\mathbb{R}$.



(a) $\mathbb{A} \cap \mathbb{B}$. (b) $\mathbb{A} \cup \mathbb{B}$. (c) $\mathbb{A} \setminus \mathbb{B}$.

Figure 2.2: Intersection, union and difference of two sets.

2.2.1 Operations

Set theory begins with a fundamental binary relation that defines whether or not an element a belongs to a set $\mathbb{A}$, denoted by $a \in \mathbb{A}$. From that relation, we are mainly considering sets composed of real numbers defined from predicates or constraints. For instance, the set of positive real numbers, commonly denoted by $\mathbb{R}^+$ is defined by:

$$\mathbb{R}^+ = \{x \in \mathbb{R} \mid x \geq 0\}\,. \tag{2.1}$$

Two special sets must be considered: the empty set denoted by $\emptyset$ that contains no element and the universe, denoted Ω that contains all elements. Basic operations on sets used in this manuscript are now introduced.

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

$$\overline{\mathbb{A}} = \{x \in \Omega \mid x \notin \mathbb{A}\}\,. \tag{2.2}$$

Given two sets $\mathbb{A}$ and $\mathbb{B}$, the inclusion between sets is defined by:

$$\mathbb{A} \subset \mathbb{B} \Leftrightarrow \forall x \in \mathbb{A}, x \in \mathbb{B}. \tag{2.3}$$

The equality between two sets is then defined by:

$$\mathbb{A} = \mathbb{B} \Leftrightarrow (\mathbb{A} \subset \mathbb{B} \wedge \mathbb{B} \subset \mathbb{A})\,. \tag{2.4}$$

where $\wedge$ stands for the logical *and* operator. The logical *or* is denoted $\vee$. Complex sets are defined from binary operations such as:

Intersection $\mathbb{A} \cap \mathbb{B} = \{x \mid x \in \mathbb{A} \wedge x \in \mathbb{B}\}$**.**

Union $\mathbb{A} \cup \mathbb{B} = \{x \mid x \in \mathbb{A} \vee x \in \mathbb{B}\}$**.**

Difference $\mathbb{A} \setminus \mathbb{B} = \{x \mid x \in \mathbb{A} \wedge x \notin \mathbb{B}\} = \mathbb{A} \cap \overline{\mathbb{B}}$**.**

Cartesian product $\mathbb{A} \times \mathbb{B} = \{(x, y) \mid x \in \mathbb{A} \wedge y \in \mathbb{B}\}$**.**

Figure 2.2 illustrates the intersection, union and difference of two sets.

In this thesis, a special intersection called the relaxed-intersection [Jaulin, 2009] will be used. It allows solving inconsistent CN by relaxing a small number of constraints. In the context of parameter estimation, it makes it possible to be robust to a given number of outliers (erroneous measurements).

Consider N sets $\mathbb{X}_1, \ldots, \mathbb{X}_N$ of $\mathbb{R}^n$. The $q-$relaxed intersection denoted by $\overset{\{q\}}{\bigcap} \mathbb{X}_i$ is the set of all $\mathbf{x} \in \mathbb{R}^n$ which belong to all $\mathbb{X}_i$'s, except q at most. Figure 2.3 illustrates this notion for N = 6 and q = 2, 3, 4. The relaxed intersection can be formulated in term of unions of intersections. For three sets $\mathbb{X}_1$, $\mathbb{X}_2$, $\mathbb{X}_3$, the 1-relaxed intersection is defined by:

$$\overset{\{1\}}{\bigcap} \mathbb{X}_i = (\mathbb{X}_1 \cap \mathbb{X}_2) \cup (\mathbb{X}_1 \cap \mathbb{X}_3) \cup (\mathbb{X}_2 \cap \mathbb{X}_3). \tag{2.5}$$

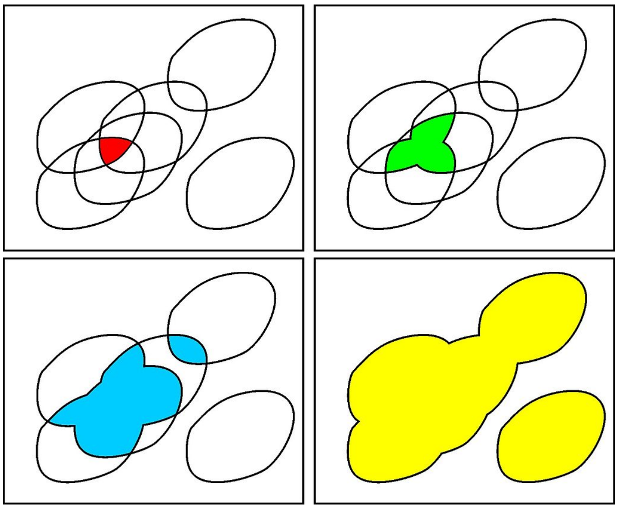


Figure 2.3: q-relaxed intersection of 6 sets for q=2 (red), q=3 (green), q=4 (blue) , q=5 (yellow)

2.2.2 Set image

Given a function $f : \mathbb{A} \rightarrow \mathbb{B}$ and $\mathbb{A}_1 \subset \mathbb{A}$, the direct image by f of $\mathbb{A}_1$ is defined by:

$$f(\mathbb{A}_1) = \{f(x) \in \mathbb{B} \mid x \in \mathbb{A}_1\} \tag{2.6}$$

The reciprocal image by f of $\mathbb{B}_1 \subset \mathbb{B}$ is then:

$$f^{-1}(\mathbb{B}_1) = \{x \in \mathbb{A} \mid f(x) \in \mathbb{B}_1\} \tag{2.7}$$

A canonical problem for which interval analysis has been proved to be efficient is the Set Inversion Problem. Let $\mathbf{f}$ be a function from $\mathbb{R}^n$ to $\mathbb{R}^m$ (possibly non-linear) and $\mathbb{Y}$ be a subset of $\mathbb{R}^m$. The Set Inversion problem aims at characterizing the reciprocal image of $\mathbb{Y}$ by $\mathbf{f}$:

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}) \tag{2.8}$$

2.2.3 Link with interval analysis

Elementary sets can be defined by constraints and combined in order to build more complex ones. The interval analysis framework provides algorithms and operators, such as *contractors* and *separators*, that makes it possible to compute, combine and represent sets of $\mathbb{R}^n$ that are defined by constraints. The following sections introduce these tools.

2.3 Interval Analysis

Interval Analysis was first developed to quantify the error on numerical computations [Moore, 1966]. In computers, real numbers are usually represented by floats with limited significant digits. This limitation leads to a small error that can drastically propagate and increase along successive operations. For instance, the decimal number "0.1" cannot be represented exactly in 32-bits floating-point pricision (8 bits exponent and 23 bits mantissa) in a computer. But it can be enclosed between two numbers such as:

$$0.0999999940395355224609375 < 0.1 < 0.100000001490116119384765625$$

Interval methods make it possible to enclose numerical and physical uncertainty, providing a rigorous theoretical framework. In this thesis, we will not focus on this low level uncertainty, which is absorbed by interval libraries such as *filib++* [Lerch et al., 2006], *gaol* [Goualard, 2008],..., but we will use interval analysis to manipulate the uncertainty on input parameters. For instance, a range sensor returns a distance, with an error, that will be represented by an interval. This section summarizes the basics of interval analysis. For a more detailed introduction, see [Jaulin et al., 2001b].

2.3.1 Interval arithmetics

2.3.1.1 Intervals

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

$$[x] = \left[x^-, x^+\right] = \left\{x \in \mathbb{R}, x^- \leq x \leq x^+\right\}. \tag{2.9}$$

An interval $[x]$ containing a single element is called a *degenerate interval* or a *singleton*. It is denoted with brackets $\{x\}$. The width w of an interval is defined by $w\left([x]\right) = x^+ - x^-$. For two intervals $[x]$ and $[y]$ of $\mathbb{R}^n$ and an operator $\diamond \in \{+, -, \cdot, /\}$, we define $[x] \diamond [y]$ as the smallest interval containing all feasible values for $x \diamond y$ when $x \in [x]$ and $y \in [y]$:

$$[x] \diamond [y] = [\{x \diamond y | x \in [x], y \in [y]\}] \tag{2.10}$$

where $[.]$ denotes the convex hull. The convex hull of a set $\mathbb{A} \subset \mathbb{R}$ is the smallest interval that contains $\mathbb{A}$.

Example 2.1. We have

$$\begin{aligned}
[-1,5] \cap [-3,2] &= [-1,2] && (2.11)\\
[-3,3] \cup [4,10] &= [-3,10] && (2.12)\\
[-3,3] - [4,10] &= [-13,-1] && (2.13)\\
w([-2,9]) &= 11 && (2.14)
\end{aligned}$$

2.3.1.2 Boxes

An interval vector $[\mathbf{x}]$ of $\mathbb{R}^n$, or a *box* for short, is a Cartesian product of n intervals. The set of all boxes of $\mathbb{R}^n$ is denoted by $\mathbb{IR}^n$.

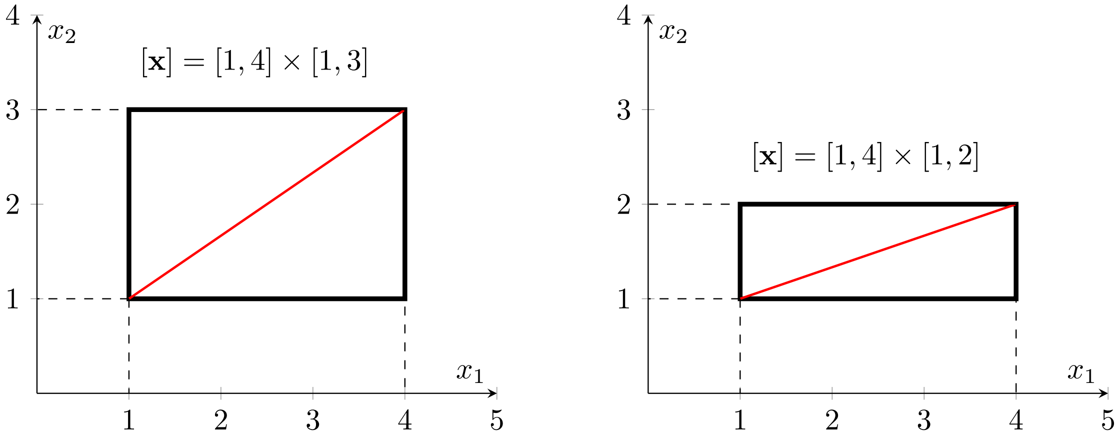
$$[\mathbf{x}] = [x_1] \times [x_2] \times \cdots \times [x_n]. \tag{2.15}$$

In this thesis, the *width* of a box $[\mathbf{x}] \in \mathbb{IR}^n$ is defined by:

$$w([\mathbf{x}]) = \sqrt{\sum_{i=1}^{n} w([x_i])^2} \tag{2.16}$$

This definition differs from the classical one which considers the length of the largest interval along all dimensions:

$$w_\infty\left([\mathbf{x}]\right) = \max_{1 \leq i \leq n} w([x_i]). \tag{2.17}$$



(a) Box 1: $w_\infty([\mathbf{x}]) = 3$, $w([\mathbf{x}]) = \sqrt{13}$

(b) Box 2: $w_\infty([\mathbf{x}]) = 3$, $w([\mathbf{x}]) = \sqrt{10}$

Figure 2.4: Boxes representation and comparison. The two boxes have the same width (w_∞) even if the area of the right one is smaller.

The first definition is preferred because, for two boxes $[\mathbf{x}]$ and $[\mathbf{y}]$, it satisfies the following relation :

$$\begin{cases} [\mathbf{x}] \subset [\mathbf{y}] \\ [\mathbf{x}] \neq [\mathbf{y}] \end{cases} \Rightarrow w([\mathbf{x}]) < w([\mathbf{y}]). \tag{2.18}$$

which is not true from w_∞. Consequently, it will be used to compare the accuracy of the approximation given by the proposed algorithms. This is illustrated in Figure 2.4 where w and w_∞ are compared.

Basic operations on real numbers or vectors can be extended to intervals by considering interval computations on each component of the box and to vector operations such as dot or cross products.

Example 2.2. For instance, with $[\mathbf{x}] = [1,2] \times [2,9]$ and $[\mathbf{y}] = [-1,2] \times [-5,5]$ we have:

$$\begin{aligned} [\mathbf{x}] + [\mathbf{y}] &= [0,4] \times [-3,14] && (2.19) \\ [\mathbf{x}] \cdot [\mathbf{y}] &= [-47,49] && (2.20) \\ w([\mathbf{x}]) &= \sqrt{50} && (2.21) \\ w_\infty([\mathbf{x}]) &= 7 && (2.22) \end{aligned}$$

where $\cdot$ denotes the product.

2.3.2 Inclusion function

One of the purposes of interval analysis is to provide, for a large class of functions $\mathbf{f}$, a fast and accurate way to compute a box which encloses the direct image of a box by $\mathbf{f}$. This operator is called an inclusion function of $\mathbf{f}$ and is now introduced.

2.3.2.1 Definition and properties

Consider $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. For any subset $\mathbb{X}$ of $\mathbb{R}^n$, the *image set* of $\mathbf{f}$ is defined by:

$$\mathbf{f}(\mathbb{X}) = \{\mathbf{y} \in \mathbb{R}^m \mid \exists \mathbf{x} \in \mathbb{X}, \mathbf{y} = \mathbf{f}(\mathbf{x})\} . \tag{2.23}$$

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

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathbf{f}\left([\mathbf{x}]\right) \subset [\mathbf{f}]\left([\mathbf{x}]\right) . \tag{2.24}$$

Thanks to interval arithmetic, and its implementation, the evaluation of an inclusion function provides a robust and reliable result taking into account round-off errors, imprecision on input data, numerical truncation, *etc.* To illustrate the notion of inclusion function, let us take $\mathbf{f}$ from $\mathbb{R}^2$ to $\mathbb{R}^2$ and a box $[\mathbf{x}] \in \mathbb{IR}^2$. The image set $\mathbf{f}([\mathbf{x}])$, in gray on Figure 2.5, can have any shape, be non convex and even disconnected. The inclusion function $[\mathbf{f}]$ of $\mathbf{f}$ makes it possible to compute a box $[\mathbf{f}]([\mathbf{x}])$ guaranteed to contain $\mathbf{f}([\mathbf{x}])$.

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

$$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow \mathbf{f}([\mathbf{x}]) \subset \mathbf{f}([\mathbf{y}]). \tag{2.25}$$

Consequently, $[\mathbf{f}]$ is *minimal* if for any $[\mathbf{x}]$, $[\mathbf{f}]([\mathbf{x}])$ is the smallest box that contains $\mathbf{f}([\mathbf{x}])$. The minimal inclusion function for $\mathbf{f}$ is unique and is denoted $[\mathbf{f}]^*$.

Enclosing the image set by a box axis-aligned introduces pessimism that can be seen as the set of extra points that belong to $[\mathbf{f}]([\mathbf{x}])$ but not to the image set $\mathbf{f}([\mathbf{x}])$.

The two main causes of pessimism are the multiple occurrences of variables in the expression of $\mathbf{f}$, that can be balanced by formal simplification [Araya et al., 2008] and the *wrapping effect.* This over-estimation can be reduced [Chabert and Jaulin, 2007] by using specific inclusion functions, multi-precision arithmetic [Revol and Rouillier, 2005], Taylor extensions [Berz and Makino, 1998], Bernstein expansions for polynomials [Garloff, 1985], *etc.*

An inclusion function $[\mathbf{f}]$ is *convergent* if, for any sequence of *nested* boxes $[\mathbf{x}](k)$

$$\lim_{k\rightarrow\infty} w\left([\mathbf{x}](k)\right) = 0 \Rightarrow \lim_{k\rightarrow\infty} w\left([\mathbf{f}]([\mathbf{x}](k))\right) = 0. \tag{2.26}$$

$[\mathbf{f}]$ is said to be *thin* if, for any degenerate interval vector $[\mathbf{x}] = \{\mathbf{x}\}$ we have $[\mathbf{f}]([\mathbf{x}]) = \mathbf{f}(\{\mathbf{x}\})$. In other words, the image of a singleton should be a singleton. This is not always the case. In Chapter 5 and Appendix A, we will consider specific methods to deal with *thick* functions. These thick (*i.e.,* non thin) functions associate to an element of $\mathbb{R}^n$, not a point, but a box of $\mathbb{IR}^m$.

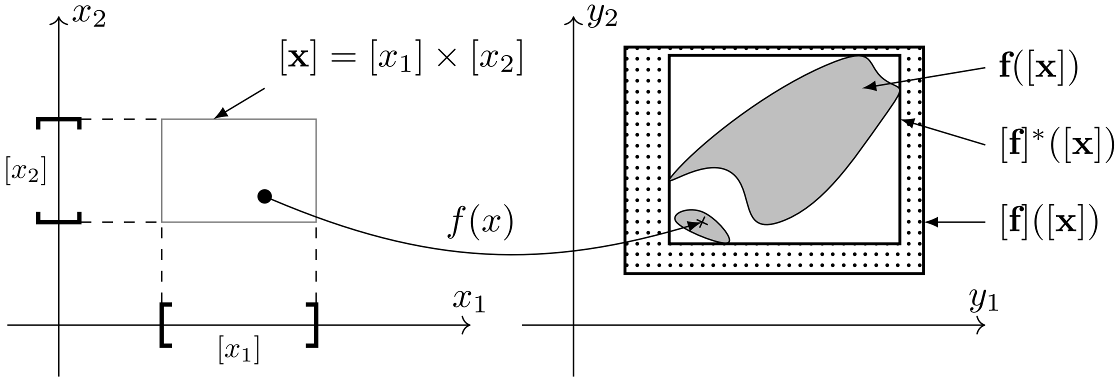

x_2
$[\mathbf{x}] = [x_1] \times [x_2]$
$[x_2]$
$f(x)$
x_1
$[x_1]$
y_2
$\mathbf{f}([\mathbf{x}])$
$[\mathbf{f}]^*([\mathbf{x}])$
$[\mathbf{f}]([\mathbf{x}])$
y_1

Figure 2.5: The image of the box $[\mathbf{x}]$ by the function $\mathbf{f}$, which is not necessarily a box, is depicted by the gray area. The dotted area represents the pessimism induced by $[\mathbf{f}]$ with respect to $[\mathbf{f}]^*$. The function $\mathbf{f}$ is thin because the image $\mathbf{f}(\mathbf{x})$ of $\mathbf{x}$ is a point.

2.3.2.2 Natural inclusion function

For any function obtained by the composition of elementary operators such as $+, -, \cdot, /, \cos, \sin, \exp, \ldots$, a *natural inclusion function* can be built by replacing these operators by their interval counterpart.

Example 2.3. The natural inclusion function $[\mathbf{f}]$ of $\mathbf{f}\,(x_1, x_2) = x_1 \cdot \cos\,(x_2)$ is:

$$[\mathbf{f}]\,([x_1], [x_2]) = [x_1] \cdot [\cos]\,([x_2]) \tag{2.27}$$

Its evaluation is done by applying interval arithmetic. For instance:

$$\begin{aligned} [\mathbf{f}]\left([-2,1],[\tfrac{\pi}{6},\tfrac{\pi}{3}]\right) &= [-2,1]\cdot[\cos]\left([\tfrac{\pi}{6},\tfrac{\pi}{3}]\right) \\ &= [-2,1]\cdot[\tfrac{1}{2},\tfrac{\sqrt{3}}{2}] \\ &= [-\sqrt{3},\tfrac{\sqrt{3}}{2}]. \end{aligned} \tag{2.28}$$

If $\mathbf{f}$ is continuous, its natural inclusion function is monotonic, convergent and thin, but generally not minimal.

2.4 Contractors and Separators

In order to characterize efficiently the domain of the variables involved in a CN [Ceberio and Granvilliers, 2000, Araya et al., 2008], interval analysis algorithms rely on operators called contractors and separators. These operators are able to remove parts of a box that do not satisfy the constraints. This section introduces these two operators.

2.4.1 Contractors

This contractor operator, associated with a constraint, is able to reduce, or *contract*, an initial box by removing parts that does not satisfy the constraint.

2.4.1.1 Definition

A *contractor* $\mathcal{C}$ is an operator $\mathbb{IR}^n \mapsto \mathbb{IR}^n$ (see *e.g.*, [Di Loreto et al., 2007, Chabert and Jaulin, 2009b]) such that

$$\begin{array}{lll} \mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}] & & \text{(contractance)} \\ [\mathbf{x}] \subset [\mathbf{y}] & \Rightarrow \; \mathcal{C}([\mathbf{x}]) \subset \mathcal{C}([\mathbf{y}]). & \text{(monotonicity)} \end{array} \tag{2.29}$$

A set $\mathbb{X}$ is *consistent* with the contractor $\mathcal{C}$ (we will write $\mathbb{X} \sim \mathcal{C}$) if for all $[\mathbf{x}]$, we have

$$\mathcal{C}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X}. \tag{2.30}$$

As a consequence, a *contractor* $\mathcal{C}_{\mathbb{X}}$ associated with the set $\mathbb{X}$ is an operator able to contract a box of $\mathbb{R}^n$ without removing a single point of the subset $\mathbb{X}$ of $\mathbb{R}^n$ to which it is associated. This is illustrated by Figure 2.6.

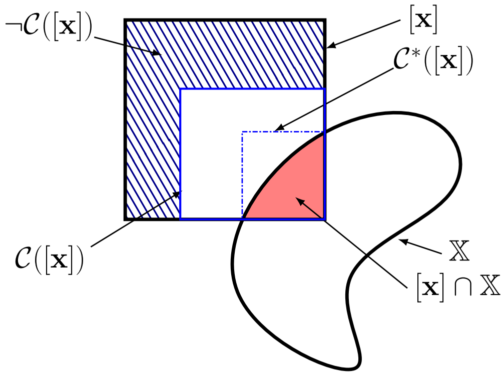

$\neg\mathcal{C}([\mathbf{x}])$
$[\mathbf{x}]$
$\mathcal{C}^*([\mathbf{x}])$
$\mathcal{C}([\mathbf{x}])$
$\mathbb{X}$
$[\mathbf{x}] \cap \mathbb{X}$

Figure 2.6: Contractor consistent with the set $\mathbb{X}$. The dashed blue area has been removed by the contractor. The minimal contractor is denoted $\mathcal{C}^*$ and returns the smallest box enclosing $[\mathbf{x}] \cap \mathbb{X}$, colored in red.

We define the inclusion between two contractors $\mathcal{C}_1$ and $\mathcal{C}_2$ as follows:

$$\mathcal{C}_1 \subset \mathcal{C}_2 \Leftrightarrow \forall\, [\mathbf{x}] \in \mathbb{IR}^n, \, \mathcal{C}_1([\mathbf{x}]) \subset \mathcal{C}_2([\mathbf{x}]). \tag{2.31}$$

Two contractors $\mathcal{C}$ and $\mathcal{C}_1$ are equivalent (we will write $\mathcal{C} \sim \mathcal{C}_1$) if we have:

$$\mathbb{X} \sim \mathcal{C} \Leftrightarrow \mathbb{X} \sim \mathcal{C}_1. \tag{2.32}$$

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

$$\mathcal{C} \sim \mathcal{C}_1 \Rightarrow \mathcal{C} \subset \mathcal{C}_1. \tag{2.33}$$

If $\mathcal{C}$ is a minimal contractor consistent with $\mathbb{X}$, then for all $[\mathbf{x}]$, we have $\mathcal{C}([\mathbf{x}]) \cap \mathbb{X} = [\![[\mathbf{x}] \cap \mathbb{X}]\!]$ where $[\![.]\!]$ is the *hull* operator. This means that $\mathcal{C}([\mathbf{x}])$ corresponds exactly to the smallest box that can be obtained by a contraction of $[\mathbf{x}]$ without removing a single point of $\mathbb{X}$. As a consequence, there exists a unique minimal contractor.

We define the *negation* $\neg\mathcal{C}$ of a contractor $\mathcal{C}$ as follows

$$\neg\mathcal{C}([\mathbf{x}]) = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{x} \notin \mathcal{C}([\mathbf{x}])\} \tag{2.34}$$

Note that, as illustrated by the blue dashed area on Figure 2.6, $\neg\mathcal{C}([\mathbf{x}])$ is not a box in general, but a union of boxes. As a consequence, the negation of a contractor is not a contractor. This is a problem which will motivate later the introduction of separators.

2.4.1.2 Building contractors

Contractors can be built for a large class of constraints. When constraints are defined by equations and inequalities, such that $\mathbf{f}(\mathbf{x}) \in [\mathbf{y}]$, a contractor based on an inclusion function $[\mathbf{f}]$ of $\mathbf{f}$ can be easily built. Given a box $[\mathbf{x}]$, it is defined by:

$$\mathcal{C}([\mathbf{x}]) = \begin{cases} \emptyset & \text{if } [\mathbf{f}]([\mathbf{x}]) \cap [\mathbf{y}] = \emptyset \\ [\mathbf{x}] & \text{otherwise} \end{cases} \tag{2.35}$$

This *binary contractor* contracts to the empty set every box such as $[\mathbf{f}]([\mathbf{x}]) \cap [\mathbf{y}] = \emptyset$. Of course, this contractor is not minimal.

An efficient improvement is the forward/backward contractor also called *HC4-revise* [Messine, 2008, Benhamou et al., 1999]. The principle is to evaluate $\mathbf{f}(\mathbf{x})$ using interval arithmetic (forward step) but instead of only considering the result of the intersection with $[\mathbf{y}]$ it retro-propagates the information to the initial domains. This is illustrated by the following simple example.

Example 2.4. Given three variables $x \in [x]$, $y \in [y]$ and $z \in [z]$, the contractor for the constraint $z = x - y$ is given by:

$$\mathcal{C}_- : \begin{pmatrix} [z] \\ [x] \\ [y] \end{pmatrix} = \begin{pmatrix} [z] \cap ([x] - [y]) & \text{forward step} \\ [x] \cap ([z] + [y]) & \text{backward step} \\ [y] \cap ([x] - [z]) & \end{pmatrix} \tag{2.36}$$

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

2.4.1.3 Contractors algebra

Contractors can be combined to solve systems of constraints. If $\mathcal{C}_1$ and $\mathcal{C}_2$ are two contractors, we define the following operations s[Chabert and Jaulin, 2009b]:

$$\begin{aligned}(\mathcal{C}_1 \cap \mathcal{C}_2)([\mathbf{x}]) &= \mathcal{C}_1([\mathbf{x}]) \cap \mathcal{C}_2([\mathbf{x}]), && (2.37)\\ (\mathcal{C}_1 \sqcup \mathcal{C}_2)([\mathbf{x}]) &= \mathcal{C}_1([\mathbf{x}]) \sqcup \mathcal{C}_2([\mathbf{x}]), && (2.38)\\ (\mathcal{C}_1 \circ \mathcal{C}_2)([\mathbf{x}]) &= \mathcal{C}_1\left(\mathcal{C}_2([\mathbf{x}])\right), && (2.39)\\ \mathcal{C}_1^{\infty}([\mathbf{x}]) &= \mathcal{C}_1 \circ \mathcal{C}_1 \circ \mathcal{C}_1 \circ \cdots \circ \mathcal{C}_1([\mathbf{x}]). && (2.40)\end{aligned}$$

where $\sqcup$ is the *union hull* defined by

$$[\mathbf{x}] \sqcup [\mathbf{y}] = [\![[\mathbf{x}] \cup [\mathbf{y}]]\!] \,. \tag{2.41}$$

Based on this formalism, primitive contractors can be combined in order to build more complex ones. Generally, these elementary contractors are built from dedicated algorithms.

Example 2.5. In order to illustrate the use of contractors, we consider the example of the goniometric localization of a robot. This problem consists in using angles measured between beacons, whose position is known, and the robot for localization. Let us consider the situation, depicted on Figure 2.7 with two beacons $\mathbf{m}_1$ and $\mathbf{m}_2$. We call *bearing* the angle θ_i between the vehicle and the beacon $\mathbf{m}_i$, measured with respect to the x-axis direction. The measurements, taken with an uncertainty $\Delta\theta$ of 3°, *i.e.* $[\theta_i] \in [\theta_i - \Delta\theta, \theta_i + \Delta\theta]$, are $\theta_1 = 45°$ and $\theta_2 = 120°$.

Recall that two vectors $\mathbf{u}, \mathbf{v}$ of $\mathbb{R}^2$ are colinear if their determinant is equal to zero, in other words if $det(\mathbf{u}, \mathbf{v}) = 0$. Thus, for each beacon $\mathbf{m}_i = (m_1^i, m_2^i)^\intercal$, we have the relation:

$$det\left(\begin{pmatrix} m_1^i - x_1 \\ m_2^i - x_2 \end{pmatrix}, \begin{pmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{pmatrix}\right) = 0. \tag{2.42}$$

The set of feasible positions for the vehicle is defined by:

$$\mathbb{X}_i = \left\{\mathbf{x} \in \mathbb{R}^2 \mid \exists \theta_i \in [\theta_i],\, (m_1^i - x_1) \cdot \sin(\theta_i) - (m_2^i - x_2) \cdot \cos(\theta_i) = 0\right\}. \tag{2.43}$$

These sets are depicted on Figure 2.7. Using a forward/backward propagation algorithm, an elementary contractor $\mathcal{C}_i$ consistent with $\mathbb{X}_i$ can be built. With two beacons, the position belongs to the intersection of the $\mathbb{X}_i$ and the contractor $\mathcal{C}$ consistent with the solution set $\mathbb{X}$ is then:

$$\begin{aligned}\mathbb{X} &= \mathbb{X}_1 \cap \mathbb{X}_2 && (2.44)\\ &\sim \mathcal{C} = \mathcal{C}_2 \circ \mathcal{C}_1 && (2.45)\end{aligned}$$

When we use contractors, we are looking for boxes as small as possible. For that

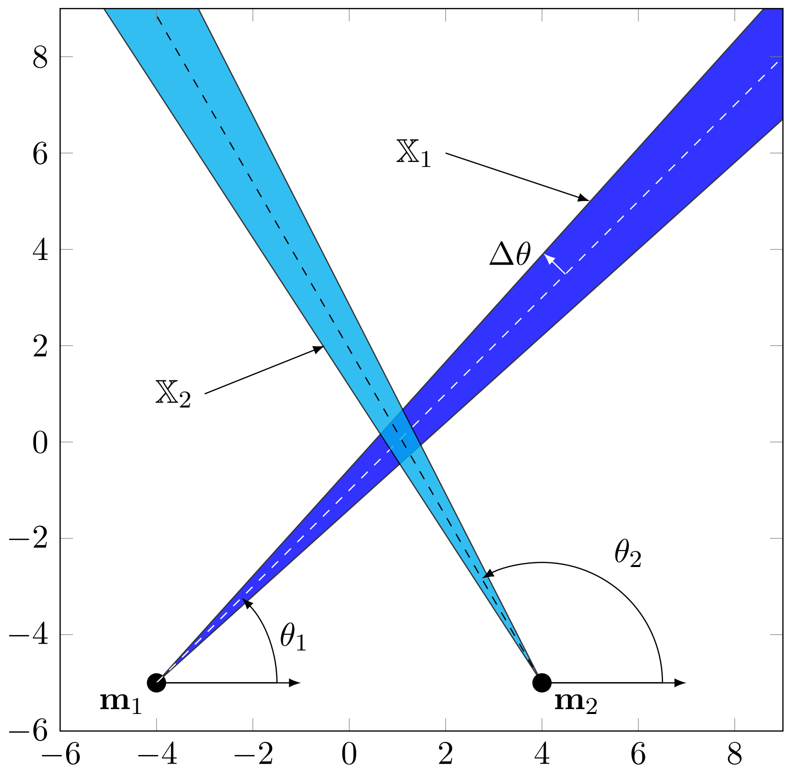


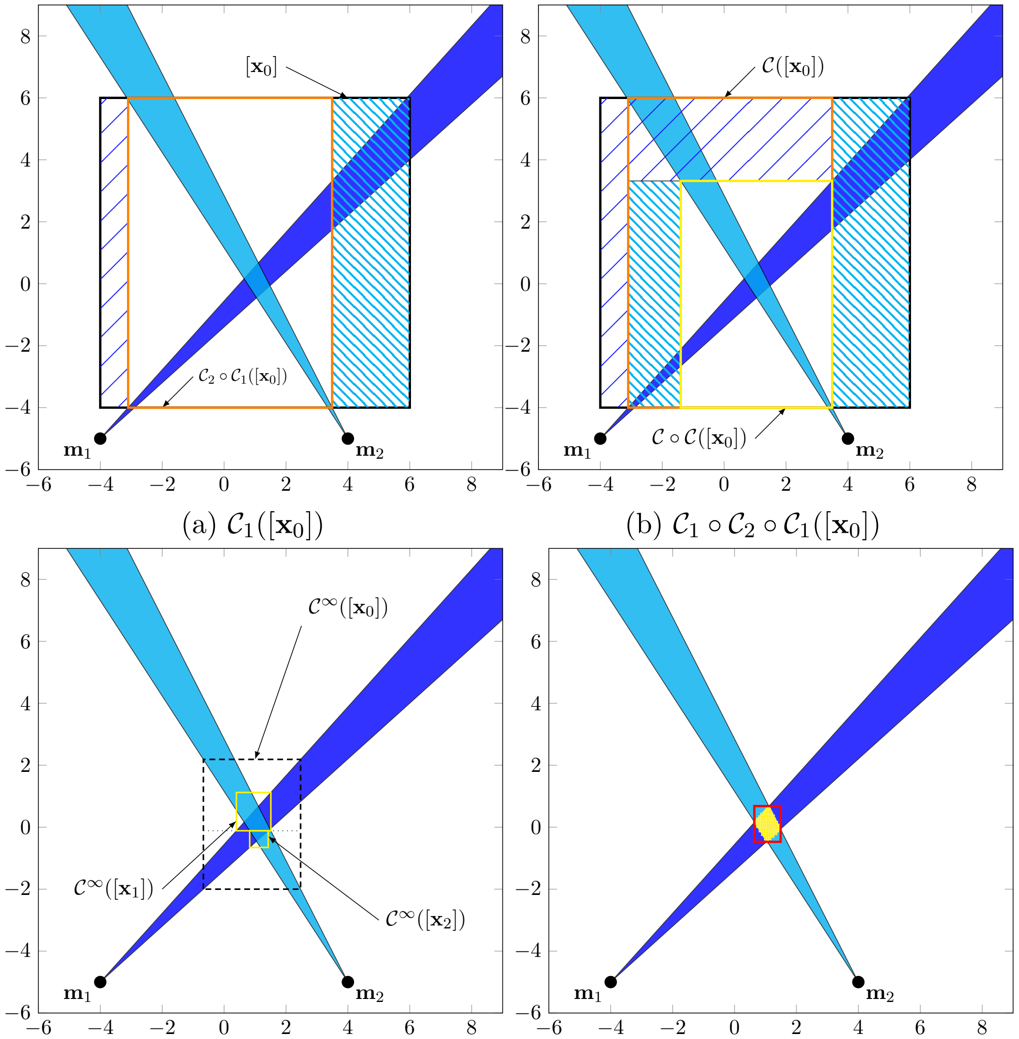
Figure 2.7: Goniometric localization

purpose, $\mathcal{C}$ is iteratively composed, denoted $\mathcal{C}^\infty = \mathcal{C} \circ \mathcal{C} \circ \cdots \circ \mathcal{C}$, in order to reach the fixed point. Since all contractors used are monotonic, it can be proved that the iterated composition $\mathcal{C}^\infty$ will always converge to the greatest box $[\mathbf{z}]$, such that $\mathcal{C}^\infty([\mathbf{z}]) = [\mathbf{z}]$, regardless the order of the contractors in composition [Montanari and Rossi, 1991].

Given an initial box $[\mathbf{x}_0] = [-4, 6]^2$, Figure 2.8 illustrates, step by step, the propagation process. On Figure 2.8a, the contraction $\mathcal{C}([\mathbf{x}_0])$, depicted by the orange box, is decomposed. The blue dashed area has been removed by $\mathcal{C}_1$, the cyan one by $\mathcal{C}_2$. The yellow box, on Figure 2.8b, shows the contraction done by applying $\mathcal{C}$ again on the orange box. This process is repeated until no more contraction happens anymore. The final result is shown on Figure 2.8c.

Now, in our case, with respect to the configuration of angles θ_1 and θ_2, the propagation process provides an enclosure of the solution set that is pessimistic. If a thinner approximation is required, a classical solution to reduce this pessimism is to bisect (split into two parts) this box, and appling $\mathcal{C}^\infty$ again on each sub-boxes. This process is illustrated in Figure 2.8c. This procedure can be recursively used until the width of resulting subboxes are smaller than a given threshold. It produces the subpaving (union of non-intersecting boxes) that encloses $\mathbb{X}$ as depicted in Figure 2.8d

This algorithm, which classifies the space, will be detailed in Section 2.4.2. The



(a) $\mathcal{C}_1([\mathbf{x}_0])$

(b) $\mathcal{C}_1 \circ \mathcal{C}_2 \circ \mathcal{C}_1([\mathbf{x}_0])$

(c) Bisection and contraction

(d) Subpaving

Figure 2.8: Illustration of the constraint propagation process. Results of contraction are depicted by red boxes.

main issue with this method is to choose the right number of bisections and to characterize the pessimism introduced by the approximation. Since boxes, which are completely included in $\mathbb{X}$, are never contracted and always bisected, their accumulations (see the yellow boxes in Figure 2.8d) lead to a huge computational cost. This justifies the introduction of separators, that aim at providing an inner approximation of the solution set, and the need for developing minimal contractors in order to avoid as much as possible bisections. Note also that the formulation of Equation (2.43) cannot be used to get an inner approximation of $\mathbb{X}$ and will be reformulated in Chapter 3 in order to build a dedicated contractor.

2.4.2 Separators

Given a set $\mathbb{X}$, the contractor consistent with $\mathbb{X}$ only provides an outer approximation. If $\mathbb{X}$ has a non-empty volume (there exists points in $\mathbb{X}$ which do not belong to its border), it can be useful to prove that a part of the initial space belongs effectively to the solution set. For that, the complementary contractor consistent with $\overline{\mathbb{X}}$ needs to be considered. In order to characterize an inner and outer approximation of the solution set, we introduce the notion of *separator*.

2.4.2.1 Definition

A separator $\mathcal{S}$ associated with the set $\mathbb{X}$ is an application such as

$$\begin{array}{rccc} \mathcal{S}: & \mathbb{IR}^n & \longrightarrow & \mathbb{IR}^n \times \mathbb{IR}^n \\ & [\mathbf{x}] & \longmapsto & ([\mathbf{x}_{in}], [\mathbf{x}_{out}]) \end{array} \tag{2.46}$$

with the following properties

$$\begin{array}{cccc} \text{(i)} & [\mathbf{x}] & = & [\mathbf{x}_{out}] \cup [\mathbf{x}_{in}] \\ \text{(ii)} & [\mathbf{x}_{out}] \cap \mathbb{X} & = & [\mathbf{x}] \cap \mathbb{X} \\ \text{(iii)} & [\mathbf{x}_{in}] \cap \overline{\mathbb{X}} & = & [\mathbf{x}] \cap \overline{\mathbb{X}} \end{array} \tag{2.47}$$

A *separator* can also be seen as a pair of contractors $\{\mathcal{S}^{\text{in}}, \mathcal{S}^{\text{out}}\}$ such that, for all $[\mathbf{x}] \in \mathbb{IR}^n$, we have

$$\mathcal{S}^{\text{in}}([\mathbf{x}]) \cup \mathcal{S}^{\text{out}}([\mathbf{x}]) = [\mathbf{x}] \qquad \text{(complementarity)}. \tag{2.48}$$

A set $\mathbb{X}$ is *consistent* with the separator $\mathcal{S}$ (we will write $\mathbb{X} \sim \mathcal{S}$), if

$$\mathbb{X} \sim \mathcal{S}^{\text{out}} \text{ and } \overline{\mathbb{X}} \sim \mathcal{S}^{\text{in}}. \tag{2.49}$$

where $\overline{\mathbb{X}} = \{\mathbf{x} \mid \mathbf{x} \notin \mathbb{X}\}$. This notion of separator is illustrated by Figure 2.9.

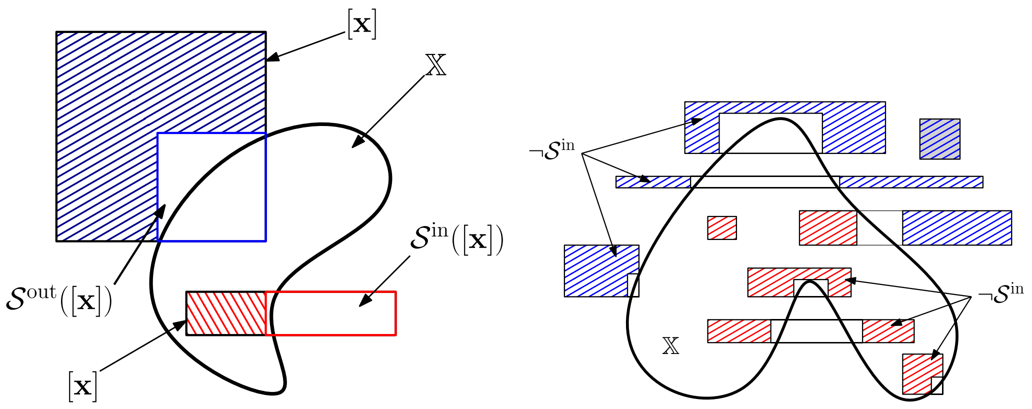

$[\mathbf{x}]$
$\mathbb{X}$
$\mathcal{S}^{\text{in}}([\mathbf{x}])$
$\mathcal{S}^{\text{out}}([\mathbf{x}])$
$[\mathbf{x}]$
$\neg\mathcal{S}^{\text{in}}$
$\neg\mathcal{S}^{\text{in}}$
$\mathbb{X}$

Figure 2.9: Illustration of a separator applied on different initial boxes. The outer contractor removes the blue dashed area and the red dashed area is removed by the inner contractor

We define the inclusion between two separators $\mathcal{S}_1$ and $\mathcal{S}_2$ as follows

$$\mathcal{S}_1 \subset \mathcal{S}_2 \Leftrightarrow \mathcal{S}_1^{\text{in}} \subset \mathcal{S}_2^{\text{in}} \text{ and } \mathcal{S}_1^{\text{out}} \subset \mathcal{S}_2^{\text{out}}. \tag{2.50}$$

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

$$\mathcal{S}_1 \subset \mathcal{S} \Rightarrow \mathcal{S}_1 = \mathcal{S}. \tag{2.51}$$

It is trivial to check that if $\mathcal{S}$ is minimal then the two contractors $\mathcal{S}^{\text{in}}$ and $\mathcal{S}^{\text{out}}$ are both minimal.

2.4.2.2 Separator algebra

The algebra for separators is a direct extension of the contractor algebra [Chabert and Jaulin, 2009b]. The main difference is that the contractor algebra does not allow any non monotonic (or decreasing) operation. It means that, if a contractor $\mathcal{C}$ is defined by an expression $\mathcal{E}$ of other contractors $\mathcal{C}_i$, then we always have

$$\forall i, \mathcal{C}_i \subset \mathcal{C}_i' \Rightarrow \mathcal{E}(\mathcal{C}_1, \mathcal{C}_2, \ldots) \subset \mathcal{E}\left(\mathcal{C}_1', \mathcal{C}_2', \ldots\right). \tag{2.52}$$

As a consequence the complementary $\overline{\mathcal{C}}$ of a contractor $\mathcal{C}$ or the restriction $\mathcal{C}_1 \backslash \mathcal{C}_2$ of two contractors $\mathcal{C}_1, \mathcal{C}_2$ (both correspond to non-monotonic operations) cannot be defined. The main advantage of separators is that it extends the operations allowed for contractors to non monotonic expressions. Let us now define some operations for separators. If $\mathcal{S} = \{\mathcal{S}^{\text{in}}, \mathcal{S}^{\text{out}}\}$ is a separator, we define the *complement* as

$$\overline{\mathcal{S}} = \left\{\mathcal{S}^{\text{out}}, \mathcal{S}^{\text{in}}\right\}. \tag{2.53}$$

Table 2.1: Center and radius of rings used in figure 2.10

	c_1	c_2	c_3
center	$(1,3)$	$(4,2)$	$(-3,-3)$
radius	$[3,6]$	$[1,5]$	$[3,4]$

Let us take $\mathcal{S}_i = \{\mathcal{S}_i^{in}, \mathcal{S}_i^{out}\}$, we could define the following operations:

$$\begin{array}{rcll} \mathcal{S}_1 \cap \mathcal{S}_2 & = & \{\mathcal{S}_1^{\text{in}} \cup \mathcal{S}_2^{\text{in}}, \mathcal{S}_1^{\text{out}} \cap \mathcal{S}_2^{\text{out}}\} & \text{(intersection)} \\ \mathcal{S}_1 \cup \mathcal{S}_2 & = & \{\mathcal{S}_1^{\text{in}} \cap \mathcal{S}_2^{\text{in}}, \mathcal{S}_1^{\text{out}} \cup \mathcal{S}_2^{\text{out}}\} & \text{(union)} \\ \overset{\{q\}}{\bigcap} \mathcal{S}_i & = & \left\{ \overset{\{m-q-1\}}{\bigcap} \mathcal{S}_i^{\text{in}}, \overset{\{q\}}{\bigcap} \mathcal{S}_i^{\text{out}} \right\} & \text{(relaxed intersection)} \\ \mathcal{S}_1 \backslash \mathcal{S}_2 & = & \mathcal{S}_1 \cap \overline{\mathcal{S}_2}. & \text{(difference)} \end{array} \tag{2.54}$$

These operations are used to combine constraints. More details about separators are given in [Jaulin and Desrochers, 2014].

The following example shows how separators can be used to manipulate sets defined with equations.

Example 2.6. Consider the set $\mathbb{X}$ defined by:

$$\mathbb{X} = (\mathbb{X}_1 \cap \mathbb{X}_2) \cup (\mathbb{X}_3 \setminus \mathbb{X}_1). \tag{2.55}$$

where $\mathbb{X}_1, \mathbb{X}_2, \mathbb{X}_3$ are three rings with center and radius defined in table 2.1. For each set, a separator $\mathcal{S}_i = \left\{\mathcal{C}_i, \overline{\mathcal{C}_i}\right\}$ can be built using a forward/backward propagation method. Using the separator algebra, the separator $\mathcal{S} = \left\{\mathcal{C}, \overline{\mathcal{C}}\right\}$ consistent with $\mathbb{X}$ is:

$$\mathcal{S} = (\mathcal{S}_1 \cap \mathcal{S}_2) \cup \left(\mathcal{S}_3 \cap \overline{\mathcal{S}_1}\right). \tag{2.56}$$

It will automatically apply the *De Morgan* rules:

$$\begin{array}{rcl} \mathcal{C} & = & (\mathcal{C}_1 \cap \mathcal{C}_2) \cup \left(\mathcal{C}_3 \cap \overline{\mathcal{C}}_1\right) \\ \overline{\mathcal{C}} & = & \left(\overline{\mathcal{C}}_1 \cup \overline{\mathcal{C}}_2\right) \cap \left(\overline{\mathcal{C}}_3 \cup \mathcal{C}_1\right) \end{array} \tag{2.57}$$

Combined with a *Paver* (see next section), the separator $\mathcal{S}$ can be used to characterize the set defined by Equation (2.55).

Separators allow us to deal efficiently with set operations by manipulating the inner and outer approximation of sets.

2.4.3 Paver

Separators are used by *pavers* to characterize sets defined by constraints or set operations. A paver is a branching algorithm which calls the separator $\mathcal{S}$ to classify

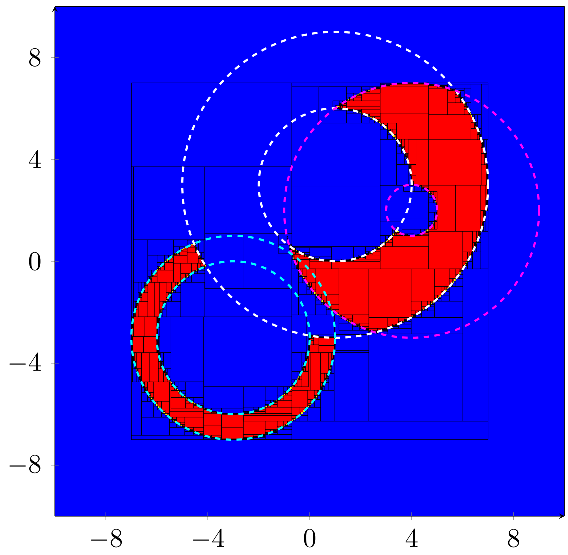


Figure 2.10: Characterization of $\mathbb{X}$ using $\mathcal{S}$ defined in Equation (2.56). Circle c_1 is drawn in white, c_2 in red and c_3 in cyan.

parts of the search space inside or outside the solution set $\mathbb{X}$ associated with $\mathcal{S}$. The algorithm is given in Algorithm (2.1). Step 1 initializes a list $\mathcal{L}$ containing all boxes to be studied. Step 2 takes one box $[\mathbf{x}]$ in $\mathcal{L}$. At Step 3, the separator $\mathcal{S}$ is then called to contract $[\mathbf{x}]$ into two boxes $[\mathbf{x}^{\text{in}}]$ and $[\mathbf{x}^{\text{out}}]$. Step 4 stores $\neg\mathcal{S}^{\text{in}}([\mathbf{x}])$, the part of $[\mathbf{x}]$ that is proved to be inside $\mathbb{X}$, into $\mathbb{X}^-$ and also into $\mathbb{X}^+$. Step 5 computes $\partial\mathcal{S}([\mathbf{x}])$ by intersecting $[\mathbf{x}^{\text{out}}]$ and $[\mathbf{x}^{\text{in}}]$. If this box is too small (*i.e.* with a width smaller than ε), it is stored inside $\mathbb{X}^+$ and will not be studied anymore. Otherwise, it is bisected, generally along its largest dimension, at Step 7 and stored into $\mathcal{L}$ waiting to be processed. After completion of the algorithm, we have the enclosure

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+. \tag{2.58}$$

Remark 2.2. . For the implementation, the resulting paving can be represented by a binary tree (*i.e.*, each node has two sons or is a leaf). The binary tree is said to be *minimal* if for any node i_1 (not the root) with brother i_2 and father j, we have

$$\begin{cases} \text{(i)} & [\mathbf{x}^{\text{in}}](i_1) \neq \emptyset,\ [\mathbf{x}^{\text{out}}](i_1) \neq \emptyset \\ \text{(ii)} & [\mathbf{x}^{\text{in}}](j) \cap [\mathbf{x}^{\text{out}}](j) = ([\mathbf{x}^{\text{in}}](i_1) \cap [\mathbf{x}^{\text{out}}](i_1)) \sqcup ([\mathbf{x}^{\text{in}}](i_2) \cap [\mathbf{x}^{\text{out}}](i_2))\,, \end{cases} \tag{2.59}$$

From a given tree generated by the paver, it is possible to simplify it into a minimal tree without changing the approximation for $\mathbb{X}$. This simplification can be done by scanning all nodes of the tree upward from the leaves to the root. If for a given node, properties (i) or (ii) of Equation (2.59) are not satisfied, then the two brother nodes i_1, i_2 with the father j are reunited into a single node. The procedure has a complexity which is linear with respect to the number of nodes of the tree. It

Algorithm 2.1 Paver(in: $[\mathbf{x}]$, $\mathcal{S}$, out: $\mathbb{X}^-$, $\mathbb{X}^+$)

```
$\mathcal{L} := \{[\mathbf{x}]\}$;
while $\mathcal{L} \neq \emptyset$ do
    Pull $[\mathbf{x}]$ from $\mathcal{L}$;
    $\{[\mathbf{x}^{\text{in}}], [\mathbf{x}^{\text{out}}]\} = \mathcal{S}([\mathbf{x}])$;
    Store $[\mathbf{x}] \backslash [\mathbf{x}^{\text{in}}]$ into $\mathbb{X}^-$ and also into $\mathbb{X}^+$;
    $[\mathbf{x}] = [\mathbf{x}^{\text{in}}] \cap [\mathbf{x}^{\text{out}}]$;
    if $w_\infty([\mathbf{x}]) < \varepsilon$ then
        store $[\mathbf{x}]$ in $\mathbb{X}^+$
    else
        bisect $[\mathbf{x}]$ and push into $\mathcal{L}$ the two resulting boxes
    end if
end while
```

will allow us to drastically reduce the number of elements of the subpaving. An illustration is provided on Figure 2.11.

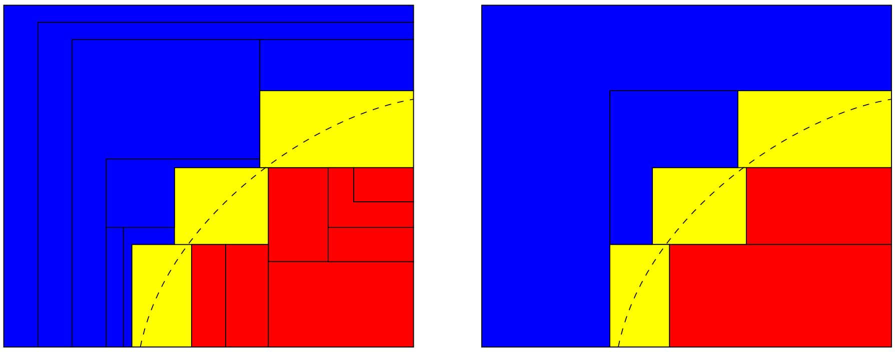


Figure 2.11: Simplification of the paving generated by the paver; left: before simplification, right: after. By convention, red boxes are proved to be inside the solution set, blue boxes are outside, and yellow boxes are undetermined.

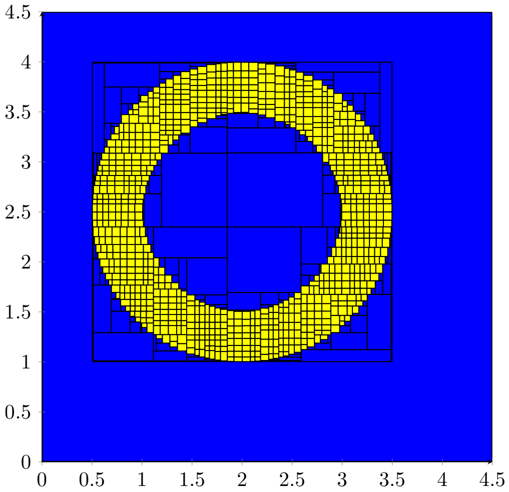
Remark 2.3. In the worst case, the complexity of the algorithm is exponential with respect to the dimension n of the initial box, and terminates after less than $(\frac{w([\mathbf{x}_0])}{\epsilon}+1)^n$ iterations. However, in practice, the number of iterations is $O(\mathcal{A}\left(\frac{1}{\epsilon}\right)^{n-1})$ where $\mathcal{A}$ is the area of the accumulation zone (in yellow) [Jaulin, 1994].

Remark 2.4. The area painted in red by the paver is not $[\mathbf{x}_{in}]$ (or $[\mathbf{x}_{out}]$) but $[\mathbf{x}]\backslash[\mathbf{x}_{in}]$ which corresponds to the part of the initial box which has been removed by the inner contractor.

Example 2.7. Consider the set

$$\mathbb{X} = \left\{\mathbf{x} \in \mathbb{R}^2, (x_1-2)^2 + (x_2-2.5)^2 \in [1, 1.5]^2\right\} \qquad (2.60)$$

which corresponds to a ring centered in $(2, 2.5)$. The minimal contractor $\mathcal{C}_{\mathbb{X}}$ consistent with $\mathbb{X}$ can be built using a forward-backward constraint propagation.



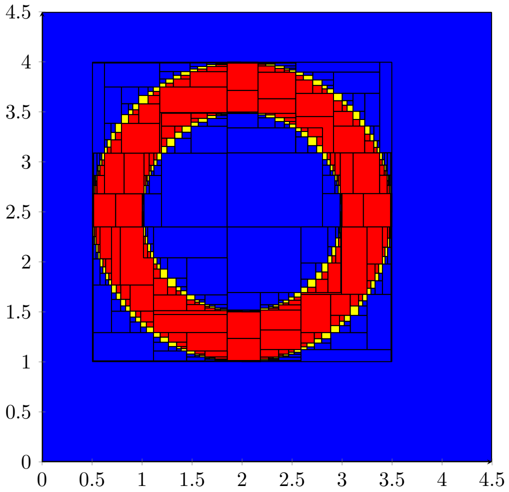


Figure 2.12: Pavings associated to Example 2.7, Left: paving obtained using the contractor, Right: paving obtained using the separator. Red boxes belong to $\mathbb{X}$ (the ring); blue boxes are outside of $\mathbb{X}$. No conclusion can be given on the yellow boxes.

Using a paver, this contractor is then used to obtain an outer approximation for $\mathbb{X}$. This is illustrated by Figure 2.12 (left) where $\mathcal{C}_{\mathbb{X}}$ removes parts of the space outside $\mathbb{X}$ (painted blue). But due to the consistency property, see Equation (2.30), $\mathcal{C}_{\mathbb{X}}$ has no effect on boxes included in $\mathbb{X}$. A box partially included in $\mathbb{X}$ cannot be eliminated and is bisected, except if its length is larger than a given value ε.

A separator consistent with $\mathbb{X}$ makes it possible to get an inner approximation. We are now able to quantify the pessimism introduced by the set inversion and to prove the existence of solutions. In a computational point of view, the paver becomes faster since the accumulation area is of dimension 1, instead of 2.

2.4.4 Inversion of separators

As it has been introduced previously, a set can be represented by its associated separator. The inversion of a separator will make it possible to get a separator for the inverse image of a set by a function. The inverse of a set $\mathbb{Y} \subset \mathbb{R}^n$ by a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is defined as

$$\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y}) = \{\mathbf{x} \mid \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\}. \tag{2.61}$$

The function $\mathbf{f}$ can be a translation, rotation, dilation, projection, or any other function. If $\mathcal{C}_{\mathbb{Y}}$ is a contractor for $\mathbb{Y}$, a contractor $\mathcal{C}_{\mathbb{X}}$ for $\mathbb{X}$ can be defined using a generalization of the forward-backward contractor as shown in [Chabert, 2013]. The contractor $\mathcal{C}_{\mathbb{X}}$ is called the inverse of $\mathcal{C}_{\mathbb{Y}}$ by $\mathbf{f}$ and we write $\mathcal{C}_{\mathbb{X}} = \mathbf{f}^{-1}(\mathcal{C}_{\mathbb{Y}})$. If $\mathcal{S}_{\mathbb{Y}}$ is a separator associated with a set $\mathbb{Y}$, we define the inverse of the separator $\mathcal{S}_{\mathbb{Y}}$

as follows

$$\mathbf{f}^{-1}\left(\mathcal{S}_{\mathbb{Y}}\right) = \left\{\mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{in}}), \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{out}})\right\}. \tag{2.62}$$

Theorem 2.1. *The separator* $\mathbf{f}^{-1}\left(\mathcal{S}_{\mathbb{Y}}\right)$ *is a separator associated with the set* $\mathbb{X} = \mathbf{f}^{-1}\left(\mathbb{Y}\right)$, i.e.,

$$\mathbf{f}^{-1}\left(\mathbb{Y}\right) \sim \mathbf{f}^{-1}\left(\mathcal{S}_{\mathbb{Y}}\right). \tag{2.63}$$

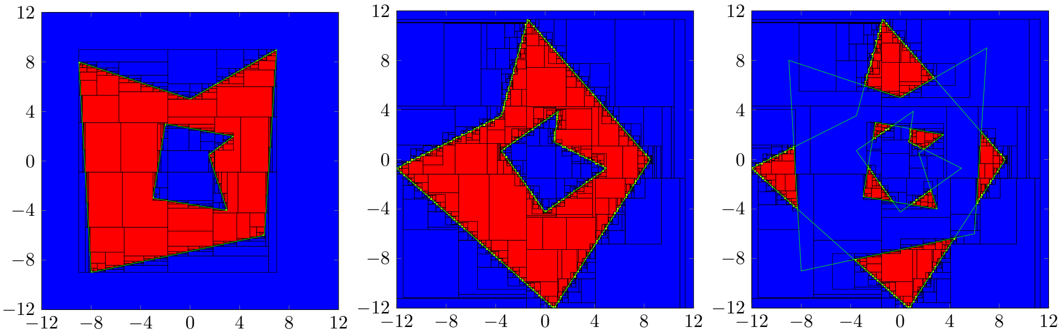
Proof. We have :

$$\begin{array}{lll}
 & \mathbb{Y} \sim \{\mathcal{S}_{\mathbb{Y}}^{\text{in}}, \mathcal{S}_{\mathbb{Y}}^{\text{out}}\} & \\
\Leftrightarrow & \mathbb{Y} \sim \mathcal{S}_{\mathbb{Y}}^{\text{out}},\ \overline{\mathbb{Y}} \sim \mathcal{S}_{\mathbb{Y}}^{\text{in}} & \text{(see 2.49)} \\
\Rightarrow & \mathbf{f}^{-1}\left(\mathbb{Y}\right) \sim \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{out}}),\ \mathbf{f}^{-1}\left(\overline{\mathbb{Y}}\right) \sim \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{in}}) & \text{(see [Chabert, 2013])} \\
\Leftrightarrow & \mathbf{f}^{-1}\left(\mathbb{Y}\right) \sim \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{out}}),\ \overline{\mathbf{f}^{-1}\left(\mathbb{Y}\right)} \sim \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{in}}) & \\
\Leftrightarrow & \mathbf{f}^{-1}\left(\mathbb{Y}\right) \sim \{\mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{in}}), \mathbf{f}^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{out}})\} & \text{(see ((2.49)))} \\
\Leftrightarrow & \mathbf{f}^{-1}\left(\mathbb{Y}\right) \sim \mathbf{f}^{-1}\left(\mathcal{S}_{\mathbb{Y}}\right) & \text{(see (2.63))}
\end{array} \tag{2.64}$$

which terminates the proof. □

The following example shows how the inversion of a separator can be used.

Example 2.8. Consider the polygon $\mathbb{M}$ defined on Figure 2.13a, for which a separator can be built (see Section 2.5.1 for more details) and a rotation of an angle of $\frac{\pi}{4}$ denoted $\mathcal{R}_{\frac{\pi}{4}}$. Figure 2.13b depicts the set $\mathcal{R}_{\frac{\pi}{4}}(\mathbb{M})$ computed using Theorem 2.1. Figure 2.13c shows how separators can be combined to compute the set $\mathcal{R}_{\frac{\pi}{4}}(\mathbb{M}) \cap \overline{\mathbb{M}}$.

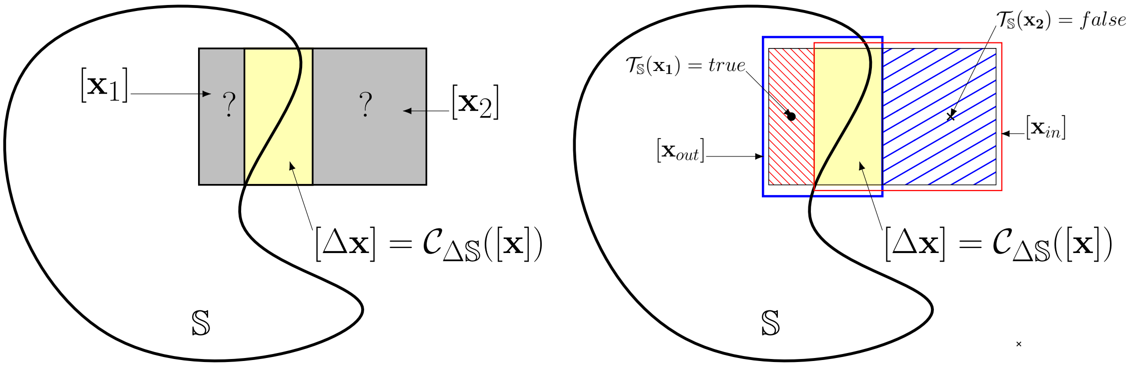


(a) $\mathbb{M}$ (b) $\mathcal{R}_{\frac{\pi}{4}}(\mathbb{M})$ (c) $\mathcal{R}_{\frac{\pi}{4}}(\mathbb{M}) \cap \overline{\mathbb{M}}$

Figure 2.13: Illustration of the inversion of separators. The polygon in (a) is rotated of an angle of $\frac{\pi}{4}$ in (b) and intersected with $\overline{\mathbb{M}}$ in (c).

2.5 Dedicated Contractors and Separators

A contractor can be used to represent an information we have on a vector of $\mathbb{R}^n$. This information can be an equation or an inequality, but it can also come from



$[\mathbf{x}_1]$
?
?
$[\mathbf{x}_2]$
$[\Delta\mathbf{x}] = \mathcal{C}_{\Delta\mathbb{S}}([\mathbf{x}])$
$\mathbb{S}$
$\mathcal{T}_{\mathbb{S}}(\mathbf{x_2}) = false$
$\mathcal{T}_{\mathbb{S}}(\mathbf{x_1}) = true$
$[\mathbf{x}_{out}]$
$[\mathbf{x}_{in}]$
$[\Delta\mathbf{x}] = \mathcal{C}_{\Delta\mathbb{S}}([\mathbf{x}])$
$\mathbb{S}$

(a) Boundary contraction using $\mathcal{C}_{\Delta\mathbb{S}}$

(b) Subboxes classification

Figure 2.14: Illustration of boundary-based separator. An initial box is contracted using $\mathcal{C}_{\Delta\mathbb{S}}$ (left). On each removed part of the initial box, a point is tested and, by continuity, the subbox is classified (right). Note that, for the illustration, both $[\mathbf{x}_{in}]$ and $[\mathbf{x}_{out}]$ have been enlarged.

a database. For instance, it can be the map of the surrounding environment or the shape of an object. This is an important point in our SLAM context where we have to combine equations (coming from observations or the evolution of the robot) with some other knowledge coming from a database. We will illustrate the principle by building some dedicated separators associated to data sets.

2.5.1 Boundary-based separator

Sometimes, isolating explicitly the corresponding inner and outer contractor associated to a set $\mathbb{S}$ is not so easy. However, if a contractor on the boundary of $\mathbb{S}$ and a test which is able to prove that a given point is inside or outside $\mathbb{S}$ are available, it is still possible to build a separator using the following boundary-based approach. Given a contractor $\mathcal{C}_{\Delta\mathbb{S}}$ on the border of set $\mathbb{S}$, and a test $\mathcal{T}_{\mathbb{S}}$ such that, for $\mathbf{x} \in \mathbb{R}^n$

$$\mathcal{T}_{\mathbb{S}}(\mathbf{x}) = \begin{cases} true & \text{if } \mathbf{x} \in \mathbb{S} \\ false & \text{otherwise} \end{cases} \tag{2.65}$$

the boundary-based separator $\mathcal{S}_{\Delta\mathbb{S}}$ works as follows. It first contracts an input box $[\mathbf{x}]$ with respect to the boundary of $\mathbb{S}$ using $\mathcal{C}_{\Delta\mathbb{S}}$ (see Figure 2.14a). Then, the test is used on a point of each subbox of $\neg\mathcal{C}_{\Delta\mathbb{S}}([\mathbf{x}])$ to determine if it belongs or not to $\mathbb{S}$. As illustrated on Figure 2.14b, the box $[\mathbf{x}_1]$ is classified inside $\mathbb{S}$ using point $\mathbf{x}_1$ and the same is done for $[\mathbf{x}_2]$ outside. The result of the separation is then

$$\begin{aligned} \mathcal{S}_{\Delta\mathbb{S}}([\mathbf{x}]) &= \{[\mathbf{x}_{in}], [\mathbf{x}_{out}]\} \\ &= \{[\Delta\mathbf{x}] \cup [\mathbf{x}_2], [\Delta\mathbf{x}] \cup [\mathbf{x}_1]\} \end{aligned} . \tag{2.66}$$

This method is now applied in order to build a separator associated to a polygon.

2.5.1.1 Polygon separator

Two-dimensional structured environments, such as rooms or buildings, can be described by a set of segments that forms a closed polygon, convex or not, with eventually holes inside. When a polygon is convex, this set is described by a system of linear inequalities that can be solved with linear solvers. When not, this method cannot be applied and the boundary-based approach can be used instead. We will now introduce how to build a separator for general polygons of $\mathbb{R}^2$.

Contractor on the border

Consider an oriented segment $[\mathbf{a}, \mathbf{b}]$ where $(\mathbf{a}, \mathbf{b}) \in \mathbb{R}^2$. The point $\mathbf{m}$ is in the segment if it satisfies the two following equations

$$\begin{cases} \det(\mathbf{b} - \mathbf{a}, \mathbf{a} - \mathbf{m}) = 0 \\ \min(\mathbf{a}, \mathbf{b}) \leq \mathbf{m} \leq \max(\mathbf{a}, \mathbf{b}) . \end{cases} \tag{2.67}$$

A general polygon $\mathcal{P}$ is composed of N oriented segments. The border $\Delta\mathcal{P}$ of the polygon satisfies the following constraint:

$$\Delta\mathcal{P} = \left\{ \mathbf{m} \in \mathbb{R}^2,\ \exists i \in [\![1, N]\!],\ \mathbf{m} \in [\mathbf{a}_i, \mathbf{b}_i] \right\} . \tag{2.68}$$

Let us take $\mathcal{C}_{a_i,b_i}$ as a contractor for the segment $[\mathbf{a}_i, \mathbf{b}_i]$. The contractor for $\Delta\mathcal{P}$ is:

$$\mathcal{C}_{\Delta\mathcal{P}} = \bigcup_{i=1}^{N} \mathcal{C}_{a_i,b_i} . \tag{2.69}$$

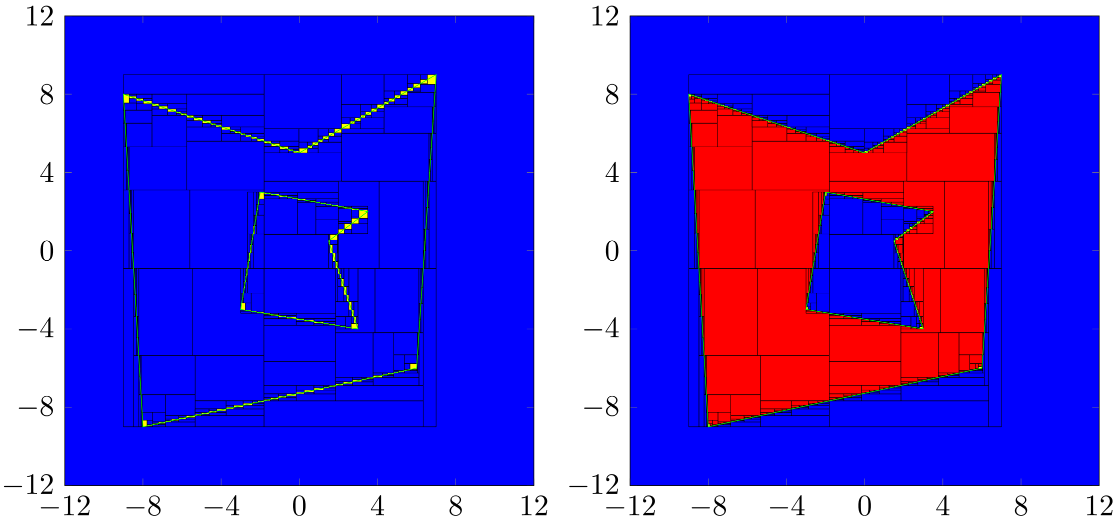
This contractor is illustrated on Figure 2.15a. As expected, it only focuses on the border of the polygon.

Remark 2.5. Because the union of minimal contractors is minimal (see Proposition 3.1), $\mathcal{C}_{\Delta\mathcal{P}}$ is a minimal contractor for the border of the polygon $\mathcal{P}$.

Test: point in polygon

To identify if a point is inside a polygon, we will use the *Winding Number* [Krantz, 2012] which represents the total number of times that a curve travels counterclockwise around the point. The winding number depends on the orientation of the curve, and is negative if the curve travels around the point clockwise. Let us take a polygon $\mathcal{P}$ with vertices $V_1, V_2, \ldots, V_n = V_1$ and $\mathbf{m}$ a point not on the border of $\mathcal{P}$. The *winding number* is defined by:

$$\mathbf{wn}(\mathbf{m}, \mathcal{P}) = \frac{1}{2\pi} \sum_{i=1}^{n} \theta_i = \frac{1}{2\pi} \sum_{i=1}^{n} \arccos \left(\frac{(V_i - \mathbf{m}).(V_{i+1} - \mathbf{m})}{\|(V_i - \mathbf{m})\| \|(V_{i+1} - \mathbf{m})\|} \right) \tag{2.70}$$



(a) Border based contractor

(b) Polygon separator

Figure 2.15: The union of contractors on segments is applied to characterize the border of the polygon (a). A test is then applied on a point on each box removed by the contractor to classify the box inside or outside the polygon (b).

So, as shown in Figure 2.16, if $\mathbf{m}$ is outside $\mathcal{P}$ we will have $\mathbf{wn}(\mathbf{m}, \mathcal{P}) = 0$, otherwise if $\mathbf{m}$ is inside, $\mathbf{wn}(\mathbf{m}, P) = 1$.

Remark 2.6. Faster algorithms that avoid the use of trigonometric functions can be found in [O'Rourke, 1998].

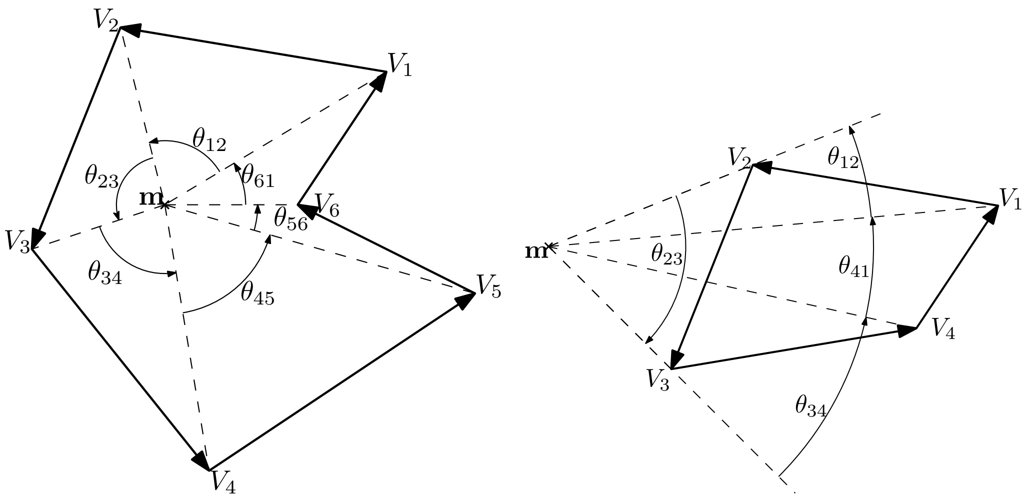
Finally, we build a separator $\mathcal{S}_{\mathcal{P}}$ for the polygon $\mathcal{P}$. Figure 2.15b shows the set $\mathbb{X}$ (in red) of all points inside the polygon.

Since this approach is generic, this type of separator can be extended to any set for which its border can be described by a union of elementary curves. The main difficulty is to find an efficient test to check whether a point belongs or not to the set.

2.5.2 Separator on subpavings

Data sets can also be stored in the form of a subpaving, *i.e.* a union of non-overlapping boxes generally organized as a binary tree, which represents an inner and an outer approximation of a set. From a given subpaving, a separator (or contractor) can be defined. Considering that each node $\mathcal{N}_j$ has two sons $\mathcal{N}_{i_1}$ and $\mathcal{N}_{i_2}$, Algorithm 2.2 allows to separate a box $[\mathbf{x}]$ into $[\mathbf{x}_{in}]$ and $[\mathbf{x}_{out}]$ in a recursive manner.

The complexity of such an algorithm is $\mathcal{O}(\log(n) + m)$ where n is the number of nodes and m is the number of nodes intersecting the initial box [de Berg et al., 2008].



(a) $\mathbf{wn}(\mathbf{m}, P) = 1$ (b) $\mathbf{wn}(\mathbf{m}, P) = 0$

Figure 2.16: In figure (a) $\mathbf{m}$ is inside $\mathcal{P}$ so $\mathcal{T}(\mathbf{m}) = true$ and in figure (b) $\mathbf{m}$ is outside P and the test is false

Algorithm 2.2 separate(in: $\mathcal{N}$, $[\mathbf{x}]$, out: $[\mathbf{x}_{in}], [\mathbf{x}_{out}]$)

```
if $[\mathbf{x}] = \emptyset$ or $\mathcal{N}$ is a leaf then
        Return $\{[\mathbf{x}_{in}(j)] \cap [\mathbf{x}], [\mathbf{x}_{out}(j)] \cap [\mathbf{x}]\}$
end if
  $[x] = [x] \cap [\mathbf{x}_{in}(j)] \cap [\mathbf{x}_{out}(j)]$
  $\{[\mathbf{x}_{in_1}], [\mathbf{x}_{out_1}]\} = $ separate($\mathcal{N}_{i_1}$, $[x]$)
  $\{[\mathbf{x}_{in_2}], [\mathbf{x}_{out_2}]\} = $ separate($\mathcal{N}_{i_2}$, $[x]$)
  Return $\{[\mathbf{x}_{in}(j)] \cup [\mathbf{x}_{in_1}] \cup [\mathbf{x}_{in_2}], [\mathbf{x}_{out}(j)] \cup [\mathbf{x}_{out_1}] \cup [\mathbf{x}_{out_2}]\}$
```

Note that any subpaving built using a paver can be simplified into a minimal subpaving (see Equation (2.59)) and used as a separator. Consequently, the results of long-running algorithms can be saved efficiently for future use.

2.5.3 Separator from images

An unstructured dataset $\mathbb{X}$ can be described by an occupancy grid, represented as a binary image, denoted i, in which the value of each pixel is 1 if it belongs to $\mathbb{X}$, 0 otherwise. The image contractor and separator are now introduced.

2.5.3.1 Image contractor

The *image contractor* was first introduced in [Sliwka, 2011, Guyonneau et al., 2013]. This contractor is based on the summed area table (also known as integral image), denoted I, widely used in computer vision [Viola and Jones, 2001] and defined by:

$$I\left(n_1, n_2\right) = \sum_{\substack{n_1' \leq n_1 \\ n_2' \leq n_2}} i\left(n_1', n_2'\right). \tag{2.71}$$

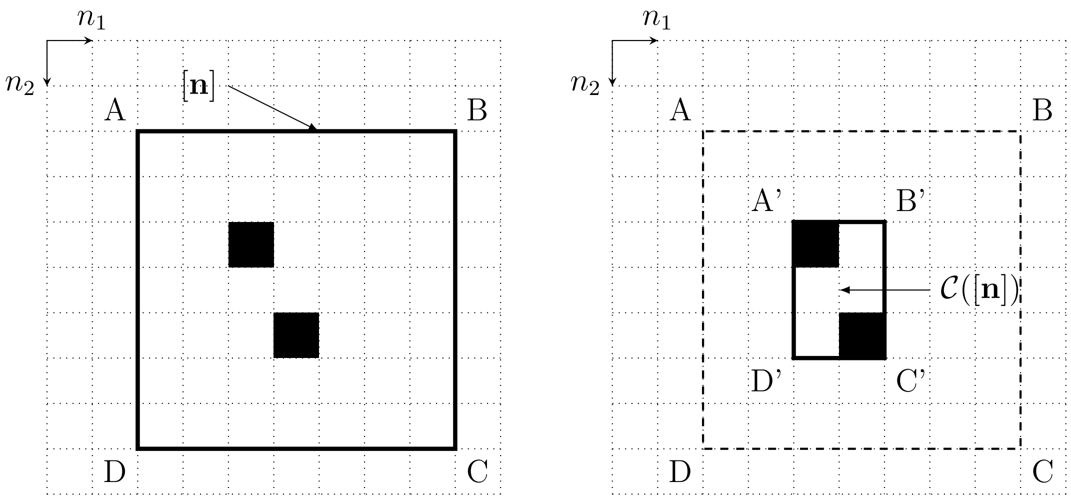
Using the integral image, the number of 1-valued pixels contained in any rectangular region can be computed in four operations. Let ϕ be the function which returns the number of occupied cells in a given box $[\mathbf{n}] = [n_1^-, n_1^+] \times [n_2^-, n_2^+]$ of $\mathbb{IN}$ aligned on the grid. Let $A = (n_1^-, n_2^-)$, $B = (n_1^+, n_2^-)$, $C = (n_1^+, n_2^+)$, $D = (n_1^-, n_2^+)$ be the coordinates of its corners, we have:

$$\begin{array}{ccc} \mathbb{IN}^2 & \longrightarrow \mathbb{N} & \\ \phi\left([\mathbf{n}]\right) & \longmapsto & I(A) + I(C) - I(B) - I(D) \end{array} \tag{2.72}$$

On Figure 2.17a, the box contains two black pixels, so $\phi\left([\mathbf{n}]\right) = 2$. From an initial box $[\mathbf{n}_0]$, the image contractor aims at finding the smallest box $[\mathbf{n}]$ included in $[\mathbf{x}_0]$ which contains exactly the same number of 1-valued pixels, *i.e*, $\phi([\mathbf{n}_0]) = \phi([\mathbf{n}])$. Denote by $\mathcal{C}_i$ the image contractor associated to the image i. Consider $[\mathbf{n}] \in \mathbb{IN}^2$ and $\mathcal{C}_i([\mathbf{n}]) = [\mathbf{m}]$ that satisfies:

$$\mathcal{L} : \begin{cases} m_1^- = \max(x \in [m_1], \phi([m_1^-; m] \times [m_2]) = 0) \\ m_1^+ = \min(x \in [m_1], \phi([m; m_1^+] \times [m_2]) = 0) \\ m_2^- = \max(x \in [m_2], \phi([m_1] \times [m_2^-; x]) = 0) \\ m_2^+ = \min(x \in [m_2], \phi([m_1] \times [m; m_2^+]) = 0) \end{cases} \tag{2.73}$$

The min and max can be computed using a dichotomy, which has a logarithmic complexity.



(a) ϕ function evaluation

(b) Contraction of the box

Figure 2.17: Image contractor

To apply a continuous contractors on a data set described by a geo-referenced image, world continuous coordinates must be converted into discrete image ones. Let $\mathbf{x}_0$ be the coordinate of the top-left corner and $\epsilon = (\epsilon_x, \epsilon_y)$ the horizontal and vertical pixel size. The change between image coordinate $[\mathbf{n}]$ to workspace coordinates $[\mathbf{x}]$ is

$$[\mathbf{x}] = \mathbf{x}_0 + [\mathbf{n}] \cdot (\{\mathbf{0}\} \cup \{\epsilon\}) \tag{2.74}$$

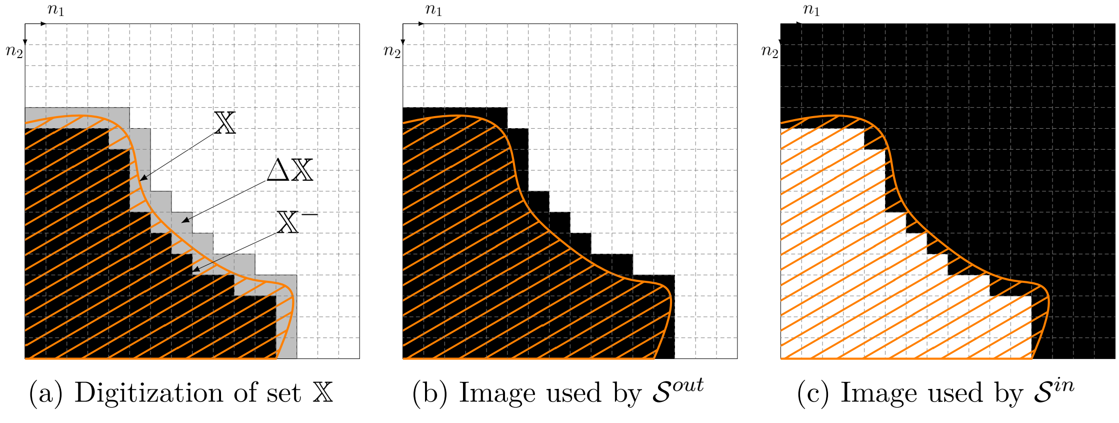
Remark 2.7. Because of the image convention, with vertical axis oriented downward, the value of ϵ_y is negative and $(\{\mathbf{0}\} \cup \{\epsilon\})$ is equal to $[0, \epsilon_x] \times [\epsilon_y, 0]$.

Remark 2.8. When a set $\mathbb{X}$ is digitized as an image, one valued pixel must describe an over approximation of $\mathbb{X}$ in order to keep the consistency property of the contractor. Consequently, the contractor is minimal with respect to the image but not with respect to $\mathbb{X}$ itself.

Remark 2.9. The image integral can be generalized to the n-dimensional case [Tapia, 2011]. In particular, the image contractor has been extended to the three-dimensional case in [Desrochers et al., 2015] to handle spacial occupancy grids, built from a 3D dense point clouds.

2.5.3.2 Image separator

A separator can be built using two complementary image contractors. Figure~2.18 shows how, from a digitized set $\mathbb{X}$, images representing an upper and lower approximations of $\mathbb{X}$ can be used to build the separator.



(a) Digitization of set $\mathbb{X}$ (b) Image used by $\mathcal{S}^{out}$ (c) Image used by $\mathcal{S}^{in}$

Figure 2.18: Image approximation of the set $\mathbb{X}$, orange dashed . Black pixels, which are completely included in $\mathbb{X}$, belong to a lower approximation $\mathbb{X}^-$. The upper approximation $\mathbb{X}^+ = \mathbb{X}^- \cup \Delta\mathbb{X}$ contains black pixels and those which intersect the boundary of $\mathbb{X}$. The outer contractor will be built using the left image while the inner contractor will be built using the complementary of $\mathbb{X}^-$ (right one).

Remark 2.10. The image separator, as well as the image contractor, has a very high speed of execution regardless the size of the input box. It only involves sums and differences of integer numbers. However, data are discretized on a regular grid and a trade off needs to be found between the storage space, the size of area, and the resolution of the grid. On the contrary, a subpaving can have better accuracy on the border of a set with a smaller memory footprint but with an high access cost.

Remark 2.11. The separators on images and subpaving are minimal with respect to the dataset. But, they are not minimal when approximated by an image or a subpaving. As illustrated by Figure 2.18a, the true border of the set, in orange, is enclosed by the gray pixels that introduce pessimism.

2.5.4 Differential contractor on tube

In the formalism presented in the introduction, the evolution of the state of the robot $\mathbf{x}(t), t \in \mathbb{R}$, is assumed to be constrained by a differential equation such as

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t)), \mathbf{u}(t)\right). \tag{2.75}$$

where $\mathbf{u}(t)$ is the input of the system. The temporal evolution of a variable $\mathbf{x}$ is described by a trajectory, denoted by $\mathbf{x}(.) : \mathbb{R} \to \mathbb{R}^n$. When this trajectory is uncertain, interval analysis makes it possible to enclose it in a *Tube*, which is an interval of trajectories.

The contractors and separators, introduced previously, are mainly dedicated to constraints that can be applied at a given time t. Now, we present briefly the tools needed to deal with an uncertain trajectory, constrained by a differential equation.

A *tube* $[\mathbf{x}](.)$ as defined in [Le Bars et al., 2012, Bethencourt and Jaulin, 2014], is an interval of trajectories $[\mathbf{x}^-(.), \mathbf{x}^+(.)]$ such that $\forall t, \mathbf{x}^-(t) \leq \mathbf{x}^+(t)$. A trajectory $\mathbf{x}(.)$ belongs to the tube $[\mathbf{x}](.)$ if $\forall t, \mathbf{x}(t) \in [\mathbf{x}](t)$. Figure 2.19 provides an illustration of a tube. It is important not to make the confusion between $\mathbf{x}(.)$ which is a trajectory and $\mathbf{x}(t)$ which is a vector of $\mathbb{R}^n$.

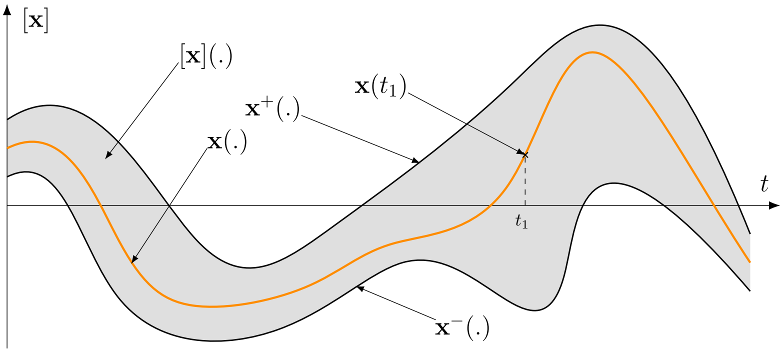


Figure 2.19: A tube $[\mathbf{x}](.)$, in grey, enclosing a trajectory $\mathbf{x}(.)$

In this thesis, tubes are mainly used to propagate a contraction, done at a given time step, to the whole trajectory. For that purpose, a dedicated contractor, denoted $\mathcal{C}_{\frac{d}{dt}}$, has been developed in [Rohou et al., 2017]. It is associated to the following constraint:

$$\begin{cases} \dot{\mathbf{x}}(.) = \mathbf{u}(.) \\ \mathbf{x}(.) \in [\mathbf{x}](.) \\ \mathbf{u}(.) \in [\mathbf{u}](.) \end{cases} \tag{2.76}$$

The contractor $\mathcal{C}_{\frac{d}{dt}}$ allows us to contract the tube $[\mathbf{x}](.)$ with respect to its derivative $\mathbf{u}(.)$ enclosed by a tube $[\mathbf{u}](.)$.

Remark 2.12. In the initial definition proposed in [Rohou et al., 2017], the contractor $\mathcal{C}_{\frac{d}{dt}}$ requires the derivative of the tube as an argument. It is written as

$$[\mathbf{x}](.) := \mathcal{C}_{\frac{d}{dt}}([\mathbf{x}](.), [\mathbf{u}](.)). \tag{2.77}$$

Now, since the tube $[\mathbf{u}](.)$ cannot be contracted with this contractor (the proof can be found in the corresponding article), we choose to simplify the notation by omitting the last argument. The tube $[\mathbf{u}](.)$ enclosing the derivativeis handled as an implicit parameter of the contractor.

Example 2.9. In order to illustrate how tubes are used, we consider the following example:

$$\begin{cases} \dot{x}(t) = u(t) \\ z_1 = x(t_1) \\ x(0) \in [-3, 3] \\ z_1 \in [2.2, 2.8] \end{cases} \tag{2.78}$$

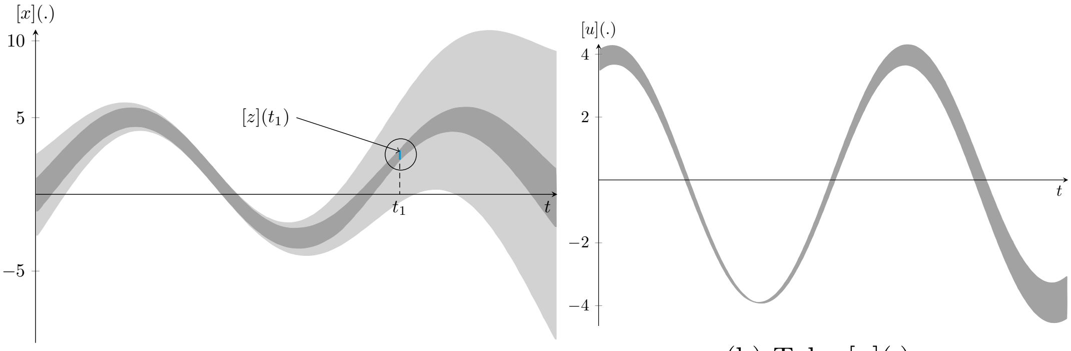
where $u(.)$ is assumed to belong to a bounded tube $[u](.)$, depicted in Figure 2.20b, and defined by:

$$\forall t > 0, [u](t) = 4 \cdot \sin(t - 5) + (t - 3.3) \cdot [-0.1, 0.1]. \tag{2.79}$$

In our robotics applications, $[u](.)$ is built from proprioceptive measurements, such as the speed or the angular velocity of the robot.

Figure 2.20a shows, in light gray, the result of $\mathcal{C}_{\frac{d}{dt}}$ applied on a tube $[x_0](.)$ defined by $\forall t > 0$, $[x_0](t) = [-\infty, \infty]$ and $[x_0](0) = [-3, 3]$. Only trajectories which are not consistent with the differential equation and the initial conditions are removed.

In the same way, a local information, such as $x(t_1) = z_1$, can be propagated using $\mathcal{C}_{\frac{d}{dt}}$ to the whole trajectory. This is illustrated by the dark grey tube on Figure 2.20a. The light grey tube is contracted in order to be consistent with the observation $[z_1]$. Note that contracting a tube at a given time step is not as trivial as doing $[x](t_1) := [x](t_1) \cap [z](t_1)$. It requires the use of a specific contractor which is not presented here. More details on tubes can be found in [Rohou, 2017]. In this thesis,



(a) Tube $[x](.)$

(b) Tube $[u](.)$

Figure 2.20: Illustration of $\mathcal{C}_{\frac{d}{dt}}$. The left figure shows the tube $[x](.)$ after the first call of $\mathcal{C}_{\frac{d}{dt}}$(light gray) and after using with $[z](t_1)$ (dark gray).

when we manipulate tubes, we use the open-source library called tubex-lib available at:

- http://www.simon-rohou.fr/research/tubex-lib

2.6 Applications

We now consider two classical applications that will illustrate how interval analysis, and mainly separators, can be used to solve robotics problems. The first application, taken from [Jaulin and Desrochers, 2014], deals with path-planning problems. The second one deals with the initial localization problem using a scanning laser rangefinder on a map defined by a polygon.

2.6.1 Application to path planning

The goal of path planning is to find a collision-free path for a robot in a given space with obstacles. The issue of path planning in a known environment has been addressed since many years (see, *e.g.,* [Laumond, 1986, Lozano-Perez, 1981, Ó'Dúnlaing and Yap, 1985, Koditschek, 1987]) and can easily be combined with set-membership techniques to take into account some uncertainties [Ceccarelli et al., 2006]. Most approaches are based on the concept of configuration space (*C-space*). Each coordinate of the C-space represents a degree of freedom of the object. An example of such robots are industrial robots which are kinematic chains of adjacent links are connected by n prismatic or rotary joints, each with one degree of freedom. The positions and orientations of each link can be characterized by n real numbers, which are the coordinates of a single n-dimensional point in the C-space [Lozano-Perez, 1983]. The *feasible configuration space* $\mathbb{M}$ is the subset of the C-space corresponding to feasible configurations of the robot. Partitioning

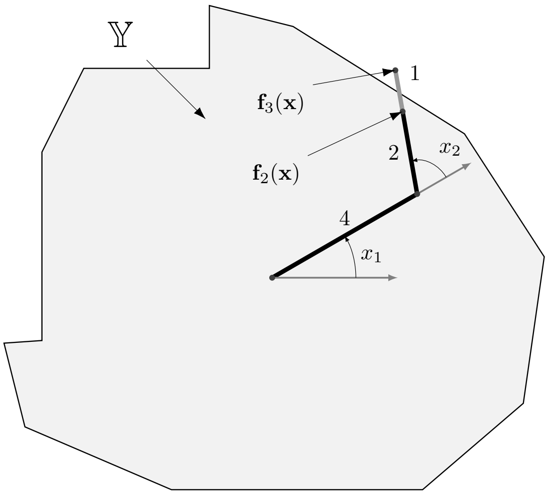


Figure 2.21: Two-dimensional wire loop game. For a feasible configuration (x_1, x_2), as represented here, the gray segment should cross the wire. Is it possible to follow the circular path ?

the C-space with subpavings makes it possible to solve the problem using graph algorithms [Brooks and Lozano-Perez, 1985, Jaulin, 2001].

The objective of this application (extracted from [Jaulin and Desrochers, 2014]) is to show how separators could solve the path planning problem on a simple example which is a 2D version of the wire loop game. This game involves a metal loop on a handle and a length of curved wire (see Figure 2.21). The player holds the loop in one hand and attempts to guide it along the curved wire without touching it. In our 2D version of this game, the player is an articulated robot with two rotary joints and the loop is a segment. The curved wire corresponds to the boundary of a set $\mathbb{Y}$ with an inside (grey in the figure) and an outside part. The length of the first and second arms are 4 and 2, respectively. The length of the loop is 1. The feasible configuration space is

$$\mathbb{M} = \{(x_1, x_2) \mid \mathbf{f}_2(\mathbf{x}) \in \mathbb{Y} \text{ and } \mathbf{f}_3(\mathbf{x}) \notin \mathbb{Y}\} = \mathbf{f}_2^{-1}(\mathbb{Y}) \cap \mathbf{f}_3^{-1}\left(\overline{\mathbb{Y}}\right), \tag{2.80}$$

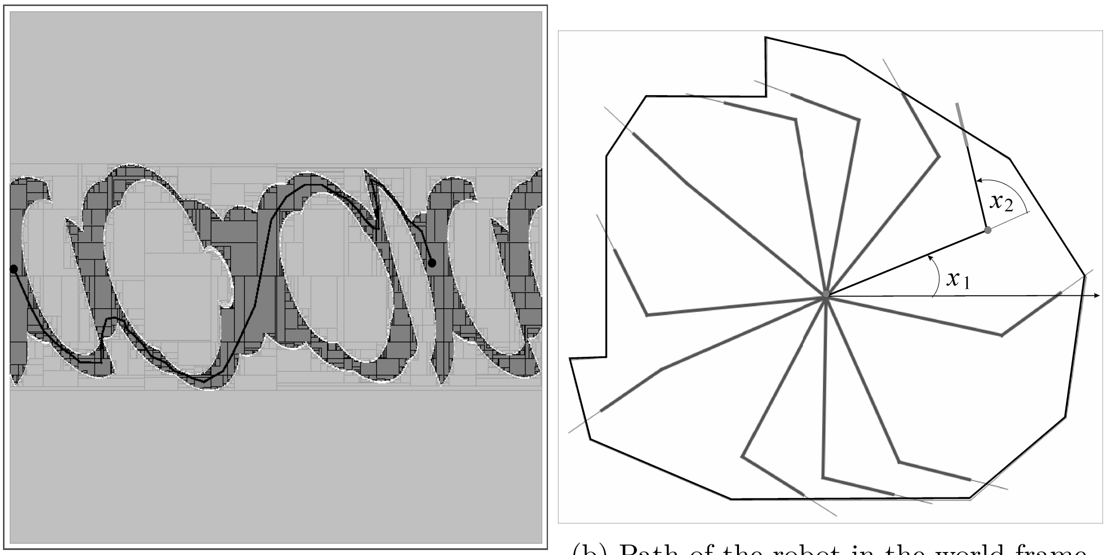
where

$$\mathbf{f}_\ell(\mathbf{x}) = 4\begin{pmatrix} \cos x_1 \\ \sin x_1 \end{pmatrix} + \ell \begin{pmatrix} \cos(x_1 + x_2) \\ \sin(x_1 + x_2) \end{pmatrix}, \quad \ell \in \{2, 3\}. \tag{2.81}$$

If $\mathcal{S}_{\mathbb{Y}}$ is a separator associated with $\mathbb{Y}$ (built as explained in Section 2.5.1), then a separator for $\mathbb{M}$ is

$$\mathcal{S}_{\mathbb{M}} = \mathbf{f}_2^{-1}(\mathcal{S}_{\mathbb{Y}}) \cap \mathbf{f}_3^{-1}\left(\overline{\mathcal{S}_{\mathbb{Y}}}\right). \tag{2.82}$$

The paver is able to approximate the feasible configuration space as illustrated by Figure 2.22a. A graph-based method is able to compute a feasible path (see the black path of the figure) which corresponds to a solution of our wire loop game.



(a) Feasible configuration space

(b) Path of the robot in the world frame

Figure 2.22: . A path, corresponding to one solution of the wire loop game, is represented. The corresponding motion in the world space is depicted on Figure 2.22b.

The minimality of the resulting subpaving is illustrated by the fact that each patch of the subpaving intersects the boundary of the solution set.

The corresponding motion in the world space is depicted on Figure 2.22b.

Remark. For our example, the set to be characterized is given by $\mathbb{M} = \mathbf{f}_2^{-1}(\mathbb{Y}) \cap \mathbf{f}_3^{-1}\left(\overline{\mathbb{Y}}\right)$ and the separator is obtained by the same expression $\mathcal{S}_{\mathbb{M}} = \mathbf{f}_2^{-1}(\mathcal{S}_{\mathbb{Y}}) \cap \mathbf{f}_3^{-1}\left(\overline{\mathcal{S}_{\mathbb{Y}}}\right)$. If we separate this expression, we get the contractor counterpart

$$\begin{cases} \mathcal{S}_{\mathbb{M}}^{\text{in}} & = & \mathbf{f}_2^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{in}}) \cup \mathbf{f}_3^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{out}}) \\ \mathcal{S}_{\mathbb{M}}^{\text{out}} & = & \mathbf{f}_2^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{out}}) \cap \mathbf{f}_3^{-1}(\mathcal{S}_{\mathbb{Y}}^{\text{in}}). \end{cases} \tag{2.83}$$

The separator formalism thus makes it possible to get directly the separator from the expression of the set that we want to characterize. This was not possible using contractor algebra which does not allow any decreasing operation. With contractors, the user has to rewrite the complementary expression. Moreover, the separator algebra requires less computations since a part of the common computation made for $\mathcal{S}_{\mathbb{M}}^{\text{in}}$ and $\mathcal{S}_{\mathbb{M}}^{\text{out}}$ is factorized.

2.6.2 Simple static map-based localization

A simple localization problem within an initial 2D map illustrates how separator operations can be used to compute the position of a robot. In order to build an inner and outer contractor, the map must divide the space into two complementary sets, with non-zero volume: $\mathbb{M}$ (gray in Figure˜2.23a) and $\overline{\mathbb{M}}$. Measurements will belong to the border of $\mathbb{M}$. In our case, the map is composed of a non-convex polygon with a hole, which for instance represents a room. The robot $\mathcal{R}$ is equipped with a laser sensor which returns the distance $d_i \in [d_i^-, d_i^+]$ between $\mathcal{R}$ and the nearest obstacle in a given direction α_i. At a given time step, N measurements are acquired with outliers. An outlier is defined as a measurement that is not consistent with the bounded error model used. The full situation is depicted on Figure 2.23a.

We also denote by $\mathbf{a}_i$ (sresp. $\mathbf{b}_i$) the nearest (resp. furthest) impact point associated with the distance d_i^- (resp. d_i^+). The parametrization is shown in Figure˜2.23b. The goal of this problem is to estimate the position $\mathbf{x} = (p_x, p_y)^\intercal$ of the robot on the map. Its heading θ is assumed to be known and is not estimated in this example and the uncertainty on α_i is assumed to be negligible.

Let $\mathbf{f}(\mathbf{x}, \alpha, d)$ be the function which translates the point $\mathbf{x}$ by a distance d in the direction given by $\alpha + \theta$. Let $\mathbf{f}_{i^-}(\mathbf{x}) = \mathbf{f}\left(\mathbf{x}, \alpha_i, d_i^-\right)$ (resp. $\mathbf{f}_{i^+}(\mathbf{x}) = \mathbf{f}\left(\mathbf{x}, \alpha_i, d_i^+\right)$) be the function which transforms the $\mathbf{x}$ into $\mathbf{a}_i$ (resp. $\mathbf{b}_i$). The set $\mathbb{X}_i$ of feasible configurations consistent with the i^{th} measurement is the set of all $\mathbf{x}$ such that the point $\mathbf{a}_i$ belongs to $\mathbb{M}$ and $\mathbf{b}_i$ belongs to $\overline{\mathbb{M}}$ given by:

$$\begin{aligned} \mathbb{X}_i &= \{\mathbf{x} \in \mathbb{R}^n | \mathbf{f}_{i^-}(\mathbf{x}) \in \mathbb{M} \text{ and } \mathbf{f}_{i^+}(\mathbf{x}) \notin \mathbb{M}\} \\ ie. \quad \mathbb{X}_i &= \mathbf{f}_{i^-}^{-1}(\mathbb{M}) \cap \mathbf{f}_{i^+}^{-1}\left(\overline{\mathbb{M}}\right) \end{aligned} \tag{2.84}$$

Given $\mathcal{S}_{\mathbb{M}}$ associated with $\mathbb{M}$, the separator $\mathcal{S}_i$ consistent with $\mathbb{X}_i$ is:

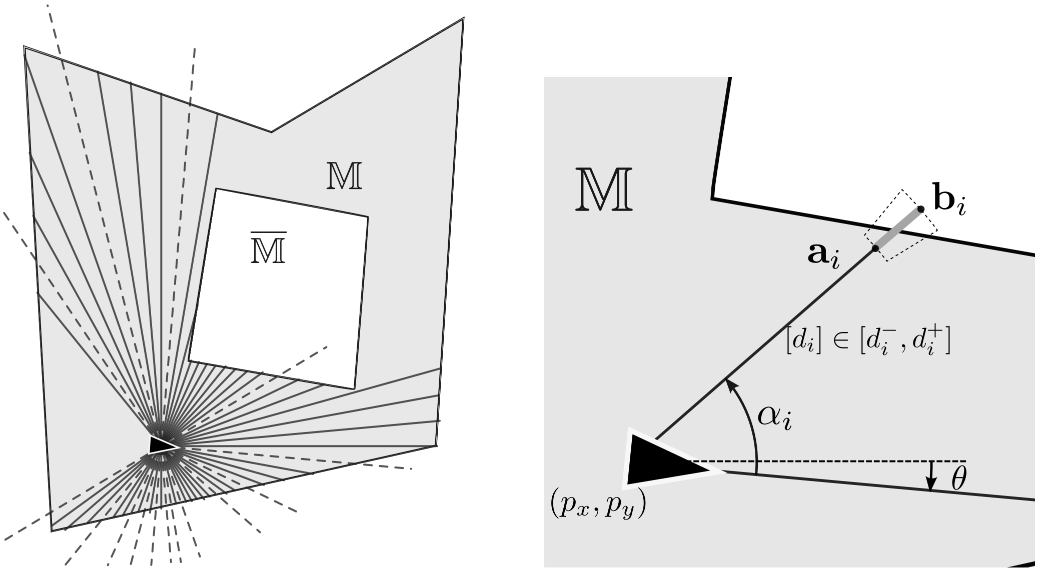
$$\mathcal{S}_i = \mathbf{f}_i^-(\mathcal{S}_{\mathbb{M}}) \cap \mathbf{f}_i^+(\overline{\mathcal{S}_{\mathbb{M}}}) \tag{2.85}$$

Without outliers, the feasible solution set must be consistent with the intersection of all $\mathbb{X}_i$. However, when dealing with outliers, this intersection is often empty. To solve this issue, the q-relaxed intersection is used to compute the set of feasible positions compatible with at least $N - q$ measurements.

$$\mathbb{X} = \bigcap_i^{\{q\}} \mathbb{X}_i \sim \bigcap_i^{\{q\}} \mathcal{S}_i \tag{2.86}$$

In this test case, 56 ranges are used with an uncertainty of +/- 0.1 meter. Results of the set inversion are shown in Figure 2.24 with $q = 15$. Because measurements are redundant and accurate, the solution set is small.

Finding automatically the number of allowed outliers is a hard task. Algorithms such as GOMNE (Guaranteed Outlier Minimal Number Estimator) can be used [Jaulin et al., 2002]. Roughly, a series of set inversion are performed with an increasing value of q until the solution set becomes not empty. An application with 3D data can be found in [Desrochers et al., 2015].



(a) Initial situation

(b) Modeling and parameters

Figure 2.23: Initial situation, the laser sensor returns 56 measurements with 15 outliers (dashed lines)

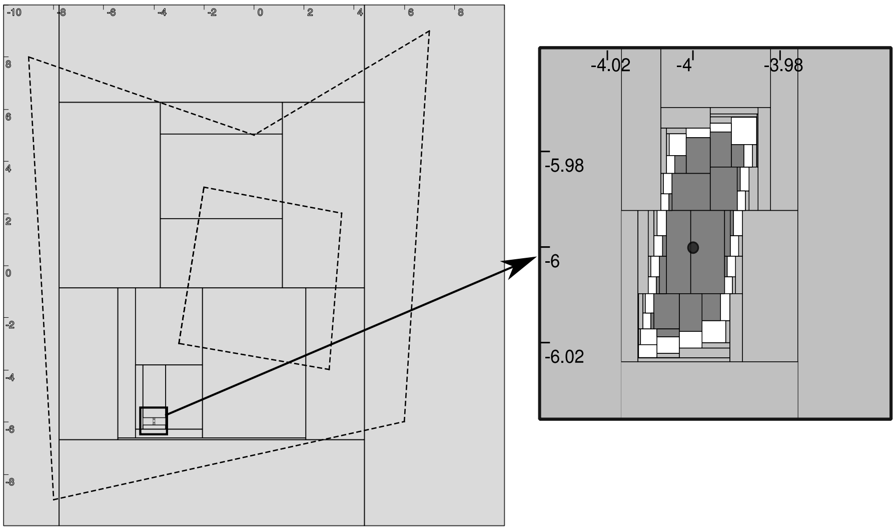


Figure 2.24: Resulting sub-pavings of the set inversion for $q = 15$. Dark gray boxes belong to $\mathbb{X}^-$ and light gray boxes are in $\Delta\mathbb{X}$. The black circle is the theoretical position of the robot

2.7 Conclusion

In this chapter, the interval analysis theory was introduced in the context of robotics localization. Many solution sets that may correspond to the position of a robot can be decomposed into a combination of unions, intersections, projections or inversions of elementary sets. Using operators such as contractors and separators, these sets can be characterized using interval analysis. For that purpose, constraint propagation and bisections mechanisms are combined inside an algorithm, called a *paver*, in order to get an inner and an outer approximation of a solution set. Now the performance of the paver relies on the efficiency of the elementary operators that are used. We introduced some dedicated contractors and separators associated to datasets which has been illustrated on two robotics examples. In the next chapter, another contractor and separator will be introduced to deal with the polar constraint, often met in localization when angle and bearing are measured.

Chapter 3

Polar Separator

Contents

3.1 Introduction

In the previous chapter, interval analysis tools have been introduced. These tools, such as contractors and separators, are used to solve rigorously complex problems involving uncertainties and nonlinear equations [Ceberio and Granvilliers, 2000, Kreinovich et al., 1997]. Coupled with a Paver, they allow computing an inner and an outer approximation of a set defined by constraints.

Contractor programming relies on a catalog of elementary contractors. Most of the time, these elementary contractors are built using interval arithmetic [Moore, 1966]. Then, by combining all these elementary contractors, we can construct a more sophisticated contractor consistent with the solution set of the problem we want to solve. The principle can be extended to separator programming [Jaulin and Desrochers, 2014] in order to compute an inner and an outer approximation of the solution set.

Now, combining contractors introduces a pessimism which has to be balanced by additional bisections performed by the paver. For more efficiency, it is important to extend the catalog by adding some new specific contractors.

The following example motivates briefly the advantage of using a dedicated and minimal contractors or separators.

Example 3.1. The set $\mathbb{X} = f^{-1}([y])$ with $[y] = [3, 5]$ and:

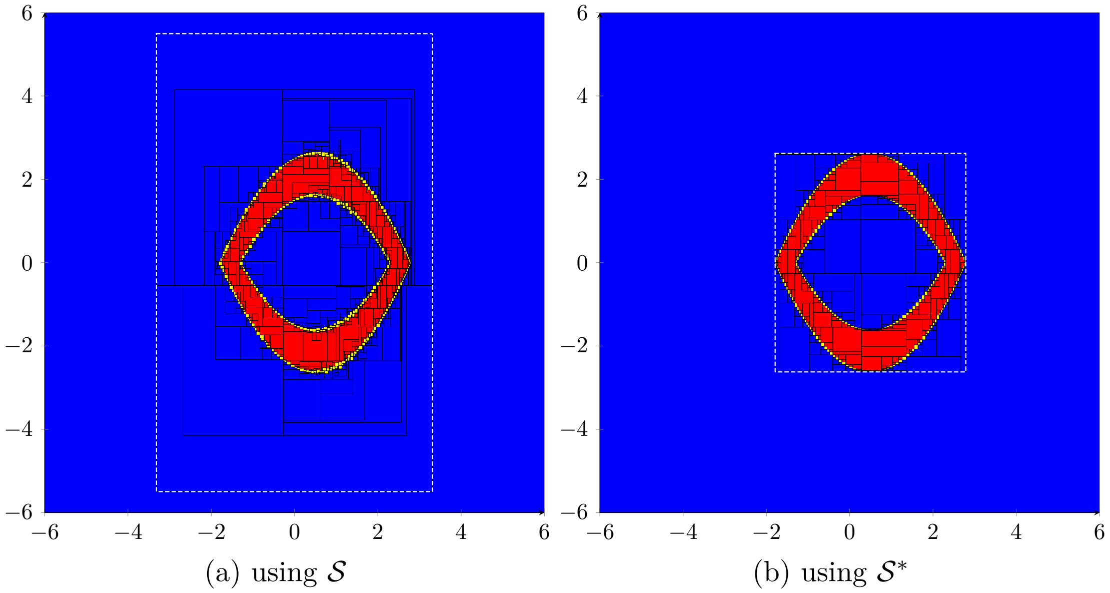
$$f(\mathbf{x}) = x_1^2 + |2x_2| - x_1 \tag{3.1}$$

can be characterized using two different separators. The first one $\mathcal{S}$ is built from the natural inclusion function of f. Because the variable x_1 appears twice, neither this inclusion function is minimal nor $\mathcal{S}$. The second separator $\mathcal{S}^*$ is built after changing the expression of f into:

$$f^*(\mathbf{x}) = (x_1 - \frac{1}{2})^2 - \frac{1}{4} + |2x_2| \tag{3.2}$$

in which multiple occurrences of x_1 have been removed. This leads to a minimal inclusion function and to the minimal separator. Figure 3.1 shows the resulting subpaving using $\mathcal{S}$ (left) and $\mathcal{S}^*$ (right). The minimal behavior of $\mathcal{S}^*$ can be observed by the fact that each box touches the border of $\mathbb{X}$. The number of bisections is also smaller when using a minimal separator. In this case, 839 bisections have been required to get the left subpaving while the right one requires 593 bisections only. The first outer approximation of the minimal separator, depicted by dashed white boxes, is more accurate. This point is critical when dealing with high dimensional problems, for which bisections need to be avoided. The more efficiently a separator is, the better the final approximation will be.

To be as efficient as possible, minimal separators, and therefore contractors, must be used when available. As illustrated by the previous example, they lead to



(a) using $\mathcal{S}$

(b) using $\mathcal{S}^*$

Figure 3.1: Resulting subpaving of Example 3.1 with $\mathcal{S}$ (left) and $\mathcal{S}^*$ (right). The dashed white box encloses the solution set and is obtained after the first call of the separator. It represents the outer approximation of the solution set and is more accurate with $\mathcal{S}^*$. All sides touch the border of $\mathbb{X}$. This is not the case with $\mathcal{S}$ where the pessimism has to be balanced with additional bisections.

an accurate outer approximation and reduce the number of bisections needed to compute an inner and an outer approximation of the solution set. This justifies the need for developing specific operators dedicated to our problems.

In this chapter, we propose some new theorems in order to build more easily optimal contractors/separators consistent with equations often used, for instance, in the field of robotics [Kieffer et al., 1999, Daney et al., 2006, Langerwisch and Wagner, 2012]. As an application, we will consider the *polar* constraint associated to the change of coordinates between Cartesian and polar form [Candau et al., 2006]. This constraint is essential for solving the localization of robots when both goniometric and/or distance measurements are available [Colle and Galerne, 2013, Di Marco et al., 2001]. Robotics applications, given in Section 3.4, will illustrate the use of the polar separator.

3.2 Building Minimal Contractors

Building minimal contractors for sets, defined by inequalities, can sometimes be done using interval based methods [Araya et al., 2008]. In the special case where each variable occurs only once in the expression and when all involved operators are continuous, for instance with the constraint $a + \sin(b + c \cdot d) = 0$, a simple interval evaluation followed by a backward propagation in the syntactic tree of the constraint provides a minimal contraction [Benhamou et al., 1999]. When the constraint is monotonic with respect to all variables, then again, it is possible to reach the minimality [Chabert and Jaulin, 2009c]. In this section, we propose some new results that will allow us to extend the class of constraints for which we can provide a minimal contractor.

3.2.1 Transformations of contractors and separators

We now present some results obtained in [Sliwka, 2011] and [Herréro, 2006] about the symmetries and the minimality of contractors. These results will be used in the next part to get the minimal contractor for specific constraints.

Proposition 3.1. If $\mathcal{C}_1$ and $\mathcal{C}_2$ are the two minimal contractors consistent with $\mathbb{X}_1$ and $\mathbb{X}_2$ then $\mathcal{C}_1 \sqcup \mathcal{C}_2$ is the minimal contractor consistent with $\mathbb{X}_1 \cup \mathbb{X}_2$.

Proof. The minimal contractor consistent with $\mathbb{X}_1 \cup \mathbb{X}_2$ is

$$\begin{aligned} [\![(\mathbb{X}_1 \cup \mathbb{X}_2) \cap [\mathbf{x}]]\!] &= [\![(\mathbb{X}_1 \cap [\mathbf{x}]) \cup (\mathbb{X}_2 \cap [\mathbf{x}])]\!] && ((\mathbb{A} \cup \mathbb{B}) \cap \mathbb{C} = (\mathbb{A} \cap \mathbb{C}) \cup (\mathbb{B} \cap \mathbb{C})) \\ &= [\![[\![\mathbb{X}_1 \cap [\mathbf{x}]]\!] \cup [\![\mathbb{X}_2 \cap [\mathbf{x}]]\!]]\!] && ([\![\mathbb{A} \cup \mathbb{B}]\!] = [\![[\![\mathbb{A}]\!] \cup [\![\mathbb{B}]\!]]\!]\ ) \\ &= [\![\mathbb{X}_1 \cap [\mathbf{x}]]\!] \sqcup [\![\mathbb{X}_2 \cap [\mathbf{x}]]\!] && (\ [\![[\mathbf{x}] \cup [\mathbf{y}]]\!] = [\mathbf{x}] \sqcup [\mathbf{y}]\ ) \\ &= \mathcal{C}_1([\mathbf{x}]) \sqcup \mathcal{C}_2([\mathbf{x}]) && (\text{minimality of } \mathcal{C}_1 \text{ and } \mathcal{C}_2) \end{aligned} \tag{3.3}$$

which terminates the proof. □

Definition 3.1. A bijective function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is *box-conservative* if for all $\mathbb{A} \subset \mathbb{R}^n$,

$$\mathbf{f}([\![\mathbb{A}]\!]) = [\![\mathbf{f}(\mathbb{A})]\!]. \tag{3.4}$$

Proposition 3.2. If $\mathbf{f}$ is box conservative so is $\mathbf{f}^{-1}$.

Proof.

$$\begin{aligned} \mathbf{f}^{-1}([\![\mathbb{A}]\!]) &= \mathbf{f}^{-1}([\![\mathbf{f} \circ \mathbf{f}^{-1}(\mathbb{A})]\!]) && (\mathbf{f} \text{ is bijective}) \\ &= \mathbf{f}^{-1} \circ \mathbf{f}([\![\mathbf{f}^{-1}(\mathbb{A})]\!]) && (\mathbf{f} \text{ is box conservative}) \\ &= [\![\mathbf{f}^{-1}(\mathbb{A})]\!]. \end{aligned} \tag{3.5}$$

□

Example 3.2. A rotation from $\mathbb{R}^2$ to $\mathbb{R}^2$ of angle α is box-conservative iff $\alpha = k \cdot \frac{\pi}{2}$, $k \in \mathbb{Z}$.

Definition 3.2. If $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is a bijective function, we define the image by $\mathbf{f}$ of a contractor as follows:

$$\mathbf{f}(\mathcal{C}_{\mathbb{X}}) = \mathbf{f} \circ \mathcal{C}_{\mathbb{X}} \circ \mathbf{f}^{-1}. \tag{3.6}$$

This new definition will make it possible to extend the contractor algebra to more complex operations. For instance $(\mathbf{f}^0 \sqcup \mathbf{f}^1 \sqcup \mathbf{f}^2)(\mathcal{C}_{\mathbb{X}})$ defines the following contractor

$$[\mathbf{x}] \mapsto \mathcal{C}_{\mathbb{X}}([\mathbf{x}]) \sqcup \left(\mathbf{f} \circ \mathcal{C}_{\mathbb{X}} \circ \mathbf{f}^{-1}([\mathbf{x}])\right) \sqcup \left(\mathbf{f}^2 \circ \mathcal{C}_{\mathbb{X}} \circ \mathbf{f}^{-2}([\mathbf{x}])\right). \tag{3.7}$$

This contractor is consistent with the set $\mathbb{X} \cup \mathbf{f}(\mathbb{X}) \cup \mathbf{f} \circ \mathbf{f}(\mathbb{X})$, as shown at least partly by the new following proposition.

Proposition 3.3. Define a set $\mathbb{X}$ for which we have a minimal contractor $\mathcal{C}_{\mathbb{X}}$. If $\mathbf{f}$ is box-conservative, then $\mathbf{f}(\mathcal{C}_{\mathbb{X}})$ is the minimal contractor for $\mathbf{f}(\mathbb{X})$.

Proof. The minimal contractor for $\mathbf{f}(\mathbb{X})$ is $[\mathbf{x}] \mapsto [\![\mathbf{f}(\mathbb{X}) \cap [\mathbf{x}]]\!]$. Now,

$$\begin{array}{rcll} [\![\mathbf{f}(\mathbb{X}) \cap [\mathbf{x}]]\!] & = & \mathbf{f} \circ \mathbf{f}^{-1}([\![\mathbf{f}(\mathbb{X}) \cap [\mathbf{x}]]\!]) & (\mathbf{f} \text{ is bijective}) \\ & = & \mathbf{f}([\![\mathbf{f}^{-1} \circ \mathbf{f}(\mathbb{X}) \cap \mathbf{f}^{-1}([\mathbf{x}])]\!]) & (\mathbf{f}^{-1} \text{ is box conservative}) \\ & = & \mathbf{f}([\![\mathbb{X} \cap \mathbf{f}^{-1}([\mathbf{x}])]\!]) & (\mathbf{f}^{-1} \circ \mathbf{f}(\mathbb{X}) = \mathbb{X}) \\ & = & \mathbf{f}(\mathcal{C}_{\mathbb{X}}(\mathbf{f}^{-1}([\mathbf{x}]))) & (\text{minimality of } \mathcal{C}_{\mathbb{X}}) \\ & = & (\mathbf{f}(\mathcal{C}_{\mathbb{X}}))([\mathbf{x}]) & (\text{definition of } \mathbf{f}(\mathcal{C}_{\mathbb{X}})\,) \end{array} \tag{3.8}$$

Thus $\mathbf{f}(\mathcal{C}_{\mathbb{X}}) = \mathbf{f} \circ \mathcal{C}_{\mathbb{X}} \circ \mathbf{f}^{-1}$ is the minimal contractor for $\mathbf{f}(\mathbb{X})$. □

Corrolary 1. If $\mathbf{f}$ is box conservative and if $\mathbf{I}$ is the identity function, from Proposition 3.3, the minimal contractor for the set $\mathbb{X} \cup \mathbf{f}(\mathbb{X})$ is $(\mathbf{I} \sqcup \mathbf{f})(\mathcal{C}_{\mathbb{X}})$.

Proof. The minimal contractor consistent with $\mathbb{X} \cup \mathbf{f}(\mathbb{X})$ is

$$\begin{array}{rll} [\![(\mathbb{X} \cup \mathbf{f}(\mathbb{X})) \cap [\mathbf{x}]]\!] & = [\![(\mathbb{X} \cap [\mathbf{x}]) \cup (\mathbf{f}(\mathbb{X}) \cap [\mathbf{x}])]\!] & ((\mathbb{A} \cup \mathbb{B}) \cap \mathbb{C} = (\mathbb{A} \cap \mathbb{C}) \cup (\mathbb{B} \cap \mathbb{C})) \\ & = [\![\mathbb{X} \cap [\mathbf{x}]]\!] \sqcup [\![\mathbf{f}(\mathbb{X}) \cap [\mathbf{x}]]\!] & (\,[\![\mathbb{A} \cup \mathbb{B}]\!] = [\![\mathbb{A}]\!] \sqcup [\![\mathbb{B}]\!]\,) \\ & = \mathcal{C}_{\mathbb{X}}([\mathbf{x}]) \ \sqcup \mathbf{f}(\mathcal{C}_{\mathbb{X}})([\mathbf{x}]) & \\ & = (\mathcal{C}_{\mathbb{X}} \sqcup \mathbf{f}(\mathcal{C}_{\mathbb{X}}))([\mathbf{x}]) & \\ & = ((\mathbf{I} \sqcup \mathbf{f})(\mathcal{C}_{\mathbb{X}}))([\mathbf{x}]). & \end{array} \tag{3.9}$$

□

Corrolary 2. If $\mathbf{f}$ is box-conservative and if $\mathcal{S}_{\mathbb{X}} = \{\mathcal{S}_{\mathbb{X}}^{\text{in}}, \mathcal{S}_{\mathbb{X}}^{\text{out}}\}$ is the minimal separator for $\mathbb{X}$, then the minimal separator for $\mathbf{f}(\mathbb{X})$ is

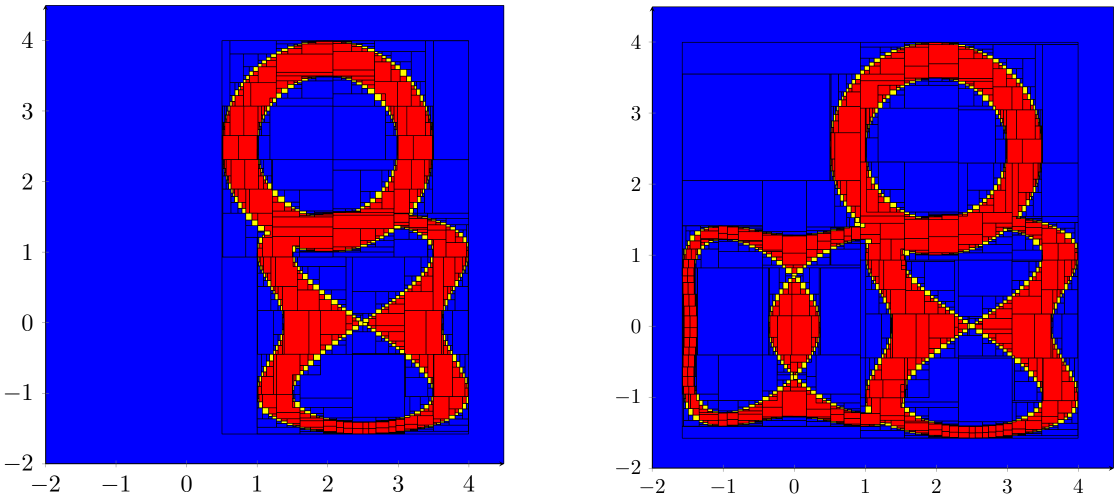
$$\mathbf{f}(\mathcal{S}_{\mathbb{X}}) = \left\{\mathbf{f} \circ \mathcal{S}_{\mathbb{X}}^{\text{in}} \circ \mathbf{f}^{-1}, \mathbf{f} \circ \mathcal{S}_{\mathbb{X}}^{\text{out}} \circ \mathbf{f}^{-1}\right\}. \tag{3.10}$$

Proof. This is a direct consequence of Proposition 3.3.

□

Example 3.3. Let us consider the set $\mathbb{X}$ defined in Example 2.7, and the following box-conservative function:

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \sqrt{3 - x_2} \\ x_1 \end{pmatrix}. \tag{3.11}$$



(a)$\mathbb{X} \cup \mathbf{f}(\mathbb{X})$ (b)$\mathbb{X} \cup \mathbf{f}(\mathbb{X}) \cup \mathbf{f}^2(\mathbb{X})$

Figure 3.2: The transformation of a minimal separator by a box-conservative function is minimal.

From a minimal separator for the set $\mathbb{X}$ as defined in Example 2.7, Corollary 2 allows us to obtain minimal separators consistent with the sets $\mathbb{X} \cup \mathbf{f}(\mathbb{X})$ and $\mathbb{X} \cup \mathbf{f}(\mathbb{X}) \cup \mathbf{f}^2(\mathbb{X})$. Corresponding subpavings are depicted on Figure 3.2.

3.2.2 Minimal contractors

We consider here an equation of the form $\mathbf{y} = \mathbf{f}(\mathbf{x})$. The following new theorem defines a way to build minimal contractors for this equation.

Theorem 3.1. *The minimal contractor consistent with* $\mathbb{S} = \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{y} = \mathbf{f}(\mathbf{x})\}$, *where* $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^p$, *is*

$$\mathcal{C}_{\mathbb{S}} \begin{pmatrix} [\mathbf{x}] \\ [\mathbf{y}] \end{pmatrix} = \begin{pmatrix} [\![[\mathbf{x}] \cap \mathbf{f}^{-1}([\mathbf{y}])]\!] \\ [\![[\mathbf{y}] \cap \mathbf{f}([\mathbf{x}])]\!] \end{pmatrix}. \tag{3.12}$$

Proof. Define $\mathbf{x}_{\neq i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ and $[\mathbf{x}_{\neq i}] = [x_1] \times \cdots \times [x_{i-1}] \times [x_{i+1}] \times \cdots \times [x_n]$ where the $[x_i]$'s are the interval components of the box $[\mathbf{x}]$. The optimal contraction for x_i is

$$\begin{aligned} & [\![\{x_i \in [x_i] \mid \exists \mathbf{x}_{\neq i} \in [\mathbf{x}_{\neq i}], \exists \mathbf{y} \in [\mathbf{y}], \mathbf{y} = \mathbf{f}(\mathbf{x})\}]\!] & (3.13) \\ = & \left[\!\!\left[\left\{ x_i \in [x_i], \exists \mathbf{x}_{\neq i} \in [\mathbf{x}_{\neq i}], \mathbf{x} \in \mathbf{f}^{-1}([\mathbf{y}]) \right\} \right]\!\!\right] & (3.14) \\ = & \left[\!\!\left[\mathrm{proj}_{x_i} \left([\mathbf{x}] \cap \mathbf{f}^{-1}([\mathbf{y}]) \right) \right]\!\!\right]. & (3.15) \end{aligned}$$

where the projection operator $proj_{x_i}$ will be defined in Chapter 4. Now, since for any subset of $\mathbb{R}^n$:

$$[\![\mathbb{A}]\!] = [\![\text{proj}_{x_1}(\mathbb{A})]\!] \times \cdots \times [\![\text{proj}_{x_n}(\mathbb{A})]\!], \tag{3.16}$$

we get

$$[\![[\mathbf{x}] \cap \mathbf{f}^{-1}([\mathbf{y}])]\!] = \left[\!\!\left[\text{proj}_{x_1}\left([\mathbf{x}] \cap \mathbf{f}^{-1}([\mathbf{y}])\right)\right]\!\!\right] \times \cdots \times \left[\!\!\left[\text{proj}_{x_n}\left([\mathbf{x}] \cap \mathbf{f}^{-1}([\mathbf{y}])\right)\right]\!\!\right]. \tag{3.17}$$

Let us apply the same reasoning with y_i. The optimal contraction for y_i is

$$\begin{aligned} & [\![\{y_i \in [y_i] \mid \exists \mathbf{y}_{\neq i} \in [\mathbf{y}_{\neq i}], \exists \mathbf{x} \in [\mathbf{x}], \mathbf{y} = \mathbf{f}(\mathbf{x})\}]\!] & (3.18)\\ = \; & [\![\{y_i \in [y_i] \mid \exists \mathbf{y}_{\neq i} \in [\mathbf{y}_{\neq i}], \mathbf{y} \in \mathbf{f}([\mathbf{x}])\}]\!] & (3.19)\\ = \; & [\![\text{proj}_{y_i}([\mathbf{y}] \cap \mathbf{f}([\mathbf{x}]))]\!]. & (3.20)\end{aligned}$$

Thus,

$$[\![[\mathbf{y}] \cap \mathbf{f}([\mathbf{x}])]\!] = [\![\text{proj}_{y_1}([\mathbf{y}] \cap \mathbf{f}([\mathbf{x}]))]\!] \times \cdots \times \left[\!\!\left[\text{proj}_{y_p}([\mathbf{y}] \cap \mathbf{f}([\mathbf{x}]))\right]\!\!\right]. \tag{3.21}$$

As a consequence, $\mathcal{C}_{\mathbb{S}}$ corresponds to the minimal contractor consistent with $\mathbb{S}$. □

3.2.3 Polar contractor

The results given in the previous section are applied here to build a minimal contractor for the *polar set* defined by:

$$\Pi = \left\{\mathbf{p} = (x, y, \rho, \theta) \in \mathbb{R}^4 \mid (x, y) = \boldsymbol{\pi}(\rho, \theta)\right\} \tag{3.22}$$

where

$$\boldsymbol{\pi}\begin{pmatrix}\rho\\ \theta\end{pmatrix} = \begin{pmatrix}\rho\cos\theta\\ \rho\sin\theta\end{pmatrix} \tag{3.23}$$

is the *polar function.* Define

$$\Pi_0 = [\mathbf{p}_0] \cap \Pi. \tag{3.24}$$

with

$$[\mathbf{p}_0] = \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \times \left[0, \frac{\pi}{4}\right]. \tag{3.25}$$

On $[\mathbf{p}_0]$, we have

$$\begin{pmatrix}x\\ y\end{pmatrix} = \boldsymbol{\pi}\begin{pmatrix}\rho\\ \theta\end{pmatrix} \Leftrightarrow \left\{\begin{array}{lcl}\rho & = & \sqrt{x^2+y^2}\\ \theta & = & \text{atan}(\frac{y}{x})\end{array}\right. \tag{3.26}$$

i.e.,

$$\begin{pmatrix}\rho\\ \theta\end{pmatrix} = \boldsymbol{\pi}^{-1}\begin{pmatrix}x\\ y\end{pmatrix} = \begin{pmatrix}\sqrt{x^2+y^2}\\ \text{atan}(\frac{y}{x})\end{pmatrix}. \tag{3.27}$$

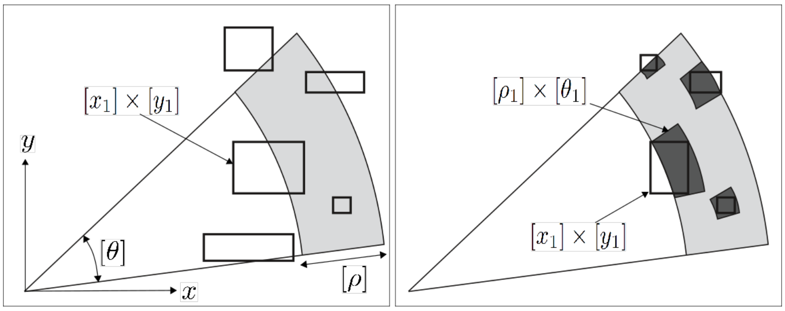


Figure 3.3: Illustration of the polar contractor. Left: before contraction, Right: after contraction.

From Theorem 3.1, the minimal contractor for Π_0 is

$$\mathcal{C}_{\Pi_0}\begin{pmatrix} [x] \times [y] \\ [\rho] \times [\theta] \end{pmatrix} = \begin{pmatrix} [\![[x] \times [y] \cap \boldsymbol{\pi}\left([\rho] \times [\theta]\right)]\!] \\ [\![[\rho] \times [\theta] \cap \boldsymbol{\pi}^{-1}\left([x] \times [y]\right)]\!] \end{pmatrix} \tag{3.28}$$

Figure 3.3 illustrates the contraction of five different boxes $[x] \times [y] \times [\rho] \times [\theta]$. The light gray pies is the initial domain for $[\rho]$ and $[\theta]$ while the dark gray pies are the resulting domains obtained after contraction. For instance, the box $[x_1] \times [y_1] \times [\rho] \times [\theta]$ on the left is contracted into the box $[x_1] \times [y_1] \times [\rho_1] \times [\theta_1]$ on the right.

The minimal contractor $\mathcal{C}_\Pi$ for Π can be deduced from $\mathcal{C}_{\Pi_0}$ using the following proposition.

Proposition 3.4. Define the following symmetries

$$\begin{aligned} \sigma_1 : (x, y, \rho, \theta) &\rightarrow \left(y, x, \rho, \frac{\pi}{2} - \theta\right) && (3.29) \\ \sigma_2 : (x, y, \rho, \theta) &\rightarrow (x, -y, \rho, -\theta) && (3.30) \\ \sigma_3 : (x, y, \rho, \theta) &\rightarrow (-x, y, \rho, \pi - \theta) && (3.31) \\ \sigma_4 : (x, y, \rho, \theta) &\rightarrow (x, y, -\rho, \pi + \theta) && (3.32) \\ \gamma : (x, y, \rho, \theta) &\rightarrow (x, y, \rho, \theta + 2\pi)\,. && (3.33) \end{aligned}$$

A minimal contractor for Π is

$$\mathcal{C}_\Pi = \left(\bigsqcup_i \gamma^i \circ (\mathbf{I} \sqcup \sigma_4) \circ (\mathbf{I} \sqcup \sigma_3) \circ (\mathbf{I} \sqcup \sigma_2) \circ (\mathbf{I} \sqcup \sigma_1)\right) \left(\mathcal{C}_{\Pi_0}\right). \tag{3.34}$$

Proof. By composing transformation functions, the initial domain, restricted to $\mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \times [0, \frac{\pi}{4}]$ is extended to $\mathbb{R}^4$. We have

$$
\begin{aligned}
\mathcal{C}_{\Pi_0} &\sim \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \times \left[0, \frac{\pi}{4}\right] \cap \Pi && (3.35)\\
(\mathbf{I} \sqcup \sigma_1)\, \mathcal{C}_{\Pi_0} &\sim \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \times \left[0, \frac{\pi}{2}\right] \cap \Pi && (3.36)\\
(\mathbf{I} \sqcup \sigma_2) \circ (\mathbf{I} \sqcup \sigma_1)\, \mathcal{C}_{\Pi_0} &\sim \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}^+ \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \cap \Pi && (3.37)\\
\bigcirc_{i\in\{3,2,1\}} (\mathbf{I} \sqcup \sigma_i)\, \mathcal{C}_{\Pi_0} &\sim \mathbb{R} \times \mathbb{R} \times \mathbb{R}^+ \times [-\pi, \pi] \cap \Pi && (3.38)\\
\bigcirc_{i\in\{4,3,2,1\}} (\mathbf{I} \sqcup \sigma_i)\, \mathcal{C}_{\Pi_0} &\sim \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times [-\pi, \pi] \cap \Pi && (3.39)\\
\bigsqcup_i \gamma \circ \bigcirc_{i\in\{4,3,2,1\}} (\mathbf{I} \sqcup \sigma_i)\, \mathcal{C}_{\Pi_0} &\sim \Pi. && (3.40)
\end{aligned}
$$

The minimality is a consequence of the fact that all transformations are box-conservative. □

3.3 Building Minimal Separators

The previous section introduced a new way to build a minimal contractor related to constraints which are built by composition of elementary constraints and box-conservative transformations. We now extend these results to build minimal separators.

3.3.1 Minimal separators

Consider the set

$$\mathbb{X} = \{\mathbf{x} \mid \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}), \tag{3.41}$$

where $\mathbf{f}$ is a function mapping $\mathbb{R}^n$ into $\mathbb{R}^m$. We assume here that $\mathbb{Y}$ is a subpaving (a finite union of boxes) which may overlap, *i.e.*,

$$\mathbb{Y} = \bigcup_i [\mathbf{y}_i]. \tag{3.42}$$

Since $\mathbb{Y}$ is a subpaving, its complementary set $\overline{\mathbb{Y}}$ is also a subpaving. For instance, in Figure 3.4, we have

$$
\begin{array}{rcl}
\mathbb{Y} & = & [1,2] \times [1,3] \cup [3,4] \times [1,3] \\
\overline{\mathbb{Y}} & = & (\mathbb{R} \times [3,\infty]) \cup (\mathbb{R} \times [-\infty,1]) \\
 & & \cup ([-\infty,1] \times [1,3]) \cup ([2,3] \times [1,3]) \\
 & & \cup ([4,\infty] \times [1,3]).
\end{array}
\tag{3.43}
$$

Definition 3.3. Consider a contractor $\mathcal{C}([\mathbf{x}],[\mathbf{y}])$. We define the partial contractor

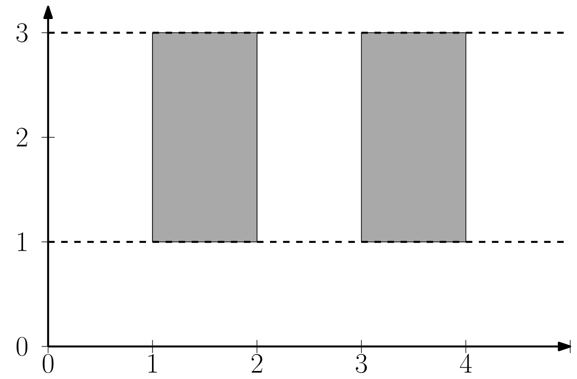


Figure 3.4: $\mathbb{Y}$ in gray and $\overline{\mathbb{Y}}$ can be represented by the union of boxes

with respect to $\mathbf{x}$ as the projection of the box $\mathcal{C}([\mathbf{x}],[\mathbf{y}])$ onto $\mathbf{x}$, *i.e.*,

$$\begin{aligned} \partial_{\mathbf{x}}\mathcal{C}([\mathbf{x}],[\mathbf{y}]) &= [\mathbf{a}] \text{ and } \partial_{\mathbf{y}}\mathcal{C}([\mathbf{x}],[\mathbf{y}]) = [\mathbf{b}]\,, && (3.44)\\ \text{where } ([\mathbf{a}],[\mathbf{b}]) &= \mathcal{C}([\mathbf{x}],[\mathbf{y}]). && (3.45) \end{aligned}$$

Theorem 3.2. *Denote by $\mathcal{C}\left([\mathbf{x}],[\mathbf{y}]\right)$, the minimal contractor consistent with the constraint $\mathbf{f}\left(\mathbf{x}\right)=\mathbf{y}$. If $\mathbb{Y}$ is a subpaving, then the minimal separator consistent with the set $\mathbb{X}=\mathbf{f}^{-1}\left(\mathbb{Y}\right)$, is*

$$\mathcal{S}\left([\mathbf{x}]\right)=\left\{\mathcal{S}^{in},\mathcal{S}^{out}\right\}\left([\mathbf{x}]\right)=\left\{\bigsqcup_{[\mathbf{y}]\in\overline{\mathbb{Y}}}\partial_{\mathbf{x}}\mathcal{C}\left([\mathbf{x}],[\mathbf{y}]\right),\bigsqcup_{[\mathbf{y}]\in\mathbb{Y}}\partial_{\mathbf{x}}\mathcal{C}\left([\mathbf{x}],[\mathbf{y}]\right)\right\}. \quad (3.46)$$

Proof. To prove that $\mathcal{S}$ is minimal, it suffices to prove that the two contractors $\mathcal{S}^{\text{in}}$ and $\mathcal{S}^{\text{out}}$ are minimal. Let us first prove that $\mathcal{S}^{\text{out}}$ is minimal. Define

$$\mathbb{X}=\left\{\mathbf{x}\mid\mathbf{f}\left(\mathbf{x}\right)\in\mathbb{Y}\right\}=\mathbf{f}^{-1}\left(\mathbb{Y}\right). \quad (3.47)$$

For a given box $[\mathbf{x}]$, the minimal contractor yields $[\![\mathbb{X}\cap[\mathbf{x}]]\!]$. Now

$$\begin{aligned} [\![\mathbb{X}\cap[\mathbf{x}]]\!] &= [\![\mathbf{f}^{-1}\left(\mathbb{Y}\right)\cap[\mathbf{x}]]\!] && \text{(definition of } \mathbb{X})\\ &= \left[\!\left[\mathbf{f}^{-1}(\textstyle\bigcup_{[\mathbf{y}]\in\mathbb{Y}}[\mathbf{y}])\cap[\mathbf{x}]\right]\!\right] && \text{(definition of } \mathbb{Y})\\ &= \left[\!\!\left[\left(\textstyle\bigcup_{[\mathbf{y}]\in\mathbb{Y}}\mathbf{f}^{-1}([\mathbf{y}])\right)\cap[\mathbf{x}]\right]\!\!\right] && (\mathbf{f}^{-1}(\mathbb{A}\cup\mathbb{B})=\mathbf{f}^{-1}\left(\mathbb{A}\right)\cup\mathbf{f}^{-1}\left(\mathbb{B}\right))\\ &= \left[\!\left[\textstyle\bigcup_{[\mathbf{y}]\in\mathbb{Y}}\left(\mathbf{f}^{-1}([\mathbf{y}])\cap[\mathbf{x}]\right)\right]\!\right] && ((\mathbb{A}\cup\mathbb{B})\cap[\mathbf{x}]=(\mathbb{A}\cap[\mathbf{x}])\cup(\mathbb{B}\cap[\mathbf{x}]))\\ &= \left[\!\left[\textstyle\bigcup_{[\mathbf{y}]\in\mathbb{Y}}[\![\mathbf{f}^{-1}([\mathbf{y}])\cap[\mathbf{x}]]\!]\right]\!\right] && (\,[\![\mathbb{A}\cup\mathbb{B}]\!]=[\![[\![\mathbb{A}]\!]\cup[\![\mathbb{B}]\!]]\!]\,)\\ &= \left[\!\left[\textstyle\bigcup_{[\mathbf{y}]\in\mathbb{Y}}\partial_{\mathbf{x}}\mathcal{C}\left([\mathbf{x}],[\mathbf{y}]\right)\right]\!\right] && \text{(minimality of } \mathcal{C}_{[\mathbf{y}]})\\ &= \textstyle\bigsqcup_{[\mathbf{y}]\in\mathbb{Y}}\partial_{\mathbf{x}}\mathcal{C}\left([\mathbf{x}],[\mathbf{y}]\right) && \text{(definition of } \sqcup\,)\\ &= \mathcal{S}^{\text{out}}\left([\mathbf{x}]\right). \end{aligned}$$

Let us now prove that $\mathcal{S}^{\text{in}}$ is minimal. Define

$$\overline{\mathbb{X}} = \{\mathbf{x} \mid \mathbf{f}(\mathbf{x}) \notin \mathbb{Y}\} = \mathbf{f}^{-1}\left(\overline{\mathbb{Y}}\right). \tag{3.48}$$

A reasoning, similar to the first part of the proof, gives us

$$\llbracket \overline{\mathbb{X}} \cap [\mathbf{x}] \rrbracket = \bigsqcup_{[\mathbf{y}] \in \overline{\mathbb{Y}}} \partial_{\mathbf{x}} \mathcal{C}([\mathbf{x}], [\mathbf{y}]) = \mathcal{S}^{\text{in}}([\mathbf{x}]), \tag{3.49}$$

which terminates the proof. □

Remark 3.1. This theorem shows that getting the optimal set inversion mostly depends on the function $\mathbf{f}$ and not on $\mathbb{Y}$. As a consequence, it is worthwhile to spend time to compute optimal contractors for the constraint $\mathbf{f}(\mathbf{x}) = \mathbf{y}$ in order to derive minimal separator for $\mathbf{f}(\mathbf{x}) \in \mathbb{Y}$.

3.3.2 Polar separator

We consider the problem of approximating the projection of Π (see Equation (3.22)) along the (ρ, θ) plan defined by:

$$[\rho] \times [\theta] \cap \boldsymbol{\pi}^{-1}([x] \times [y]). \tag{3.50}$$

Let $\mathcal{C}_{\Pi}$ be the minimal polar contractor consistent with Π. From Theorem 3.2, the minimal separator consistent with this set is

$$\mathcal{S}_{\pi}^{[x],[y]}([\rho], [\theta]) = \left\{ \bigsqcup_{[x_1] \times [y_1] \in \overline{[x] \times [y]}} \partial_{\rho\theta} \mathcal{C}_{\Pi}([x_1], [y_1], [\rho], [\theta]),\ \partial_{\rho\theta} \mathcal{C}_{\Pi}([x], [y], [\rho], [\theta]) \right\}, \tag{3.51}$$

where $\overline{[x] \times [y]}$ is a subpaving corresponding to the complement of the box $[x] \times [y]$. Here, $[x], [y]$ are the parameters of the separators and the contractions operate on the box $[\rho] \times [\theta]$. The same reasoning applied on ρ and θ, concludes that the minimal separator consistent with the set:

$$[x] \times [y] \cap \boldsymbol{\pi}([\rho] \times [\theta]) \tag{3.52}$$

is given by

$$\mathcal{S}_{\pi^{-1}}^{[\rho],[\theta]}([x], [y]) = \left\{ \bigsqcup_{[\rho_1] \times [\theta_1] \in \overline{[\rho] \times [\theta]}} \partial_{xy} \mathcal{C}_{\Pi}([x], [y], [\rho_1], [\theta_1]),\ \partial_{xy} \mathcal{C}_{\Pi}([x], [y], [\rho], [\theta]) \right\}. \tag{3.53}$$

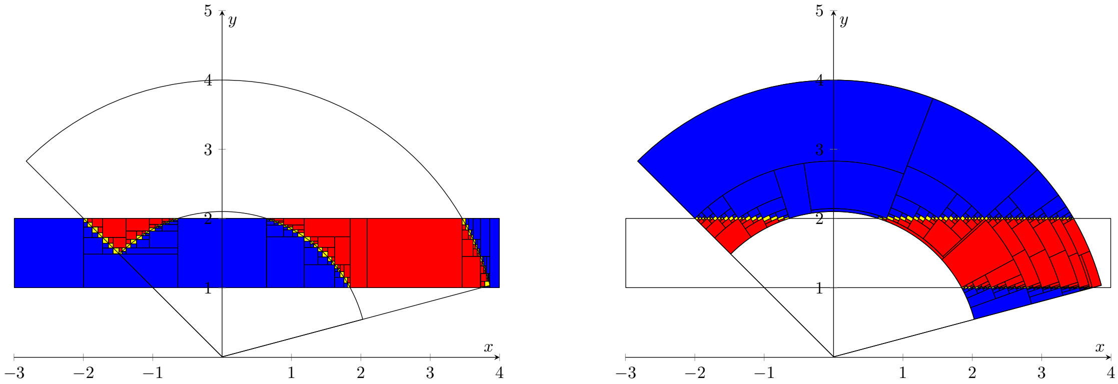


Figure 3.5: Action of the two separators defined as the projection of the polar contractor. The initial domains for (x, y) and (ρ, θ) are represented by the large rectangle and the large pie. Red boxes and pies belong to solutions sets while those in blue do not

where $\overline{[\rho] \times [\theta]}$ is a subpaving corresponding to the complement of the pie $[\rho] \times [\theta]$. The efficiency of these two separators are illustrated on Figure 3.5 with the initial intervals taken as $[x] = [-3, 4]$, $[y] = [1, 2]$, $[\rho] = [2.1, 4]$ and $[\theta] = [\frac{\pi}{12}, \frac{3\pi}{4}]$. The minimality of the separators can be observed by the fact that each box intersects the border of the pie in Figure 3.5 (left) and each pie intersects the boundary of the box in Figure 3.5 (right).

3.4 Application to Localization

Sensors returning range and goniometric measurements of a given landmark, such as sonar or LIDAR, are commonly used in robotics for localization. When measurements are related to some landmarks, the problem can be modeled using the polar constraint [Guyonneau et al., 2013]. Now, we propose examples of localization to illustrate the efficiency of the polar separator.

3.4.1 Static localization

3.4.1.1 Localization with one landmark

Consider one landmark at a known position $\mathbf{m} = (m_1, m_2)$. One robot at position $\mathbf{x} = (x_1, x_2)$ is able to measure the distance y_1 to $\mathbf{m}$ and the direction y_2 of $\mathbf{m}$ in the local reference frame with an interval accuracy. In this example, we take

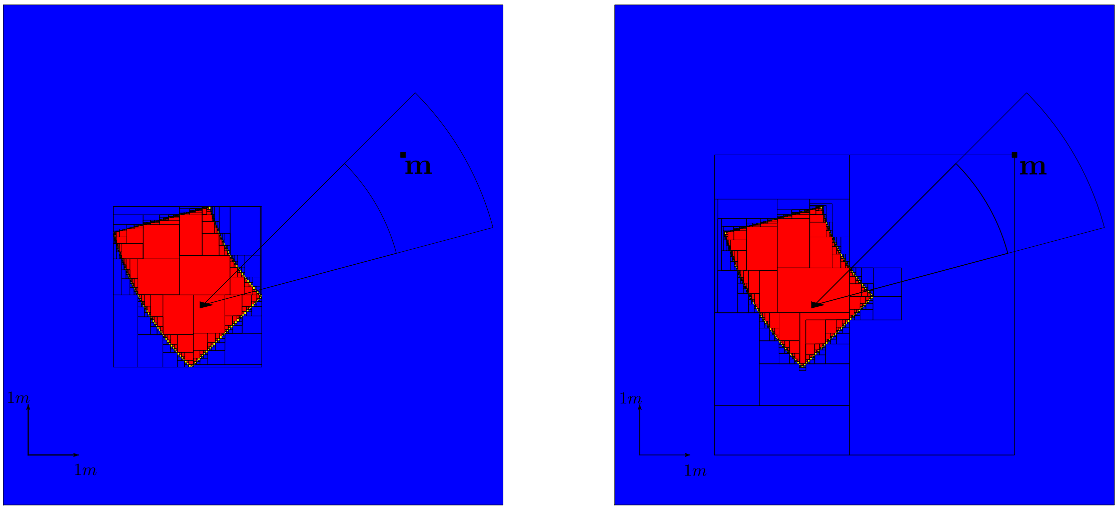


Figure 3.6: Approximation of $\mathbb{X}$ using the minimal separator (Left) and by using a classical forward-backward separator (Right). The initial domain is $\mathbb{X}_0 = [-6, 4]^2$.

$[\mathbf{y}] = [4, 6] \times \left[-\frac{\pi}{12}, \frac{\pi}{4}\right]$. The set $\mathbb{X}$ of feasible $\mathbf{x}$ is:

$$\begin{aligned} \mathbb{X} &= \{\mathbf{x} \in \mathbb{R}^2 \mid \exists \mathbf{y} \in [\mathbf{y}], \mathbf{m} - \mathbf{x} = \boldsymbol{\pi}(\mathbf{y})\} \\ &= \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{m} - \mathbf{x} \in \boldsymbol{\pi}([\mathbf{y}])\} \\ &= \mathbf{g_m} \circ \boldsymbol{\pi}\left([\mathbf{y}]\right). \end{aligned} \tag{3.54}$$

where $\boldsymbol{\pi}$ is defined in Equation (3.23) and

$$\mathbf{g_m}\left(\mathbf{x}\right) = \mathbf{m} - \mathbf{x}. \tag{3.55}$$

which is box-conservative. From Corollary 2, the minimal separator $\mathcal{S}_\mathbf{m}$, associated with landmark $\mathbf{m}$, consistent with $\mathbb{X}$ is

$$\mathcal{S}_\mathbf{m} = \mathbf{g_m} \circ \mathcal{S}_{\pi^{-1}}^{[\mathbf{y}]}. \tag{3.56}$$

where $\mathcal{S}_{\pi^{-1}}^{[\mathbf{y}]}$ is the minimal separator defined in Equation (3.53).
Figure 3.6 shows the results of the set inversion using our minimal separator (left) and a forward-backward separator (right). Note that the new Theorem 3.2 has to be used in order to be able to get an inner approximation, even with the forward/backward contractor.

3.4.1.2 Example with several landmarks

Assume now that we have 3 landmarks $\mathbf{m}_i$. The measurements $[\mathbf{y}_i]$ and the position of the landmarks are given in Table 3.1. The feasible set $\mathbb{X}$ is now

$$\mathbb{X} \;\; = \bigcap_i \;\; \mathbf{g}_{\mathbf{m}_i} \circ \boldsymbol{\pi}\left([\mathbf{y}_i]\right). \tag{3.57}$$

Table 3.1: Range and bearing measurements of the three landmarks

landmarks	1	2	3
m_1	6	-2	-3
m_2	12	-5	10
$[y_1]$(m)	[11, 12]	[8, 10]	[5, 7]
$[y_2]$(°)	[14, 100]	[-147, -75]	[63, 150]

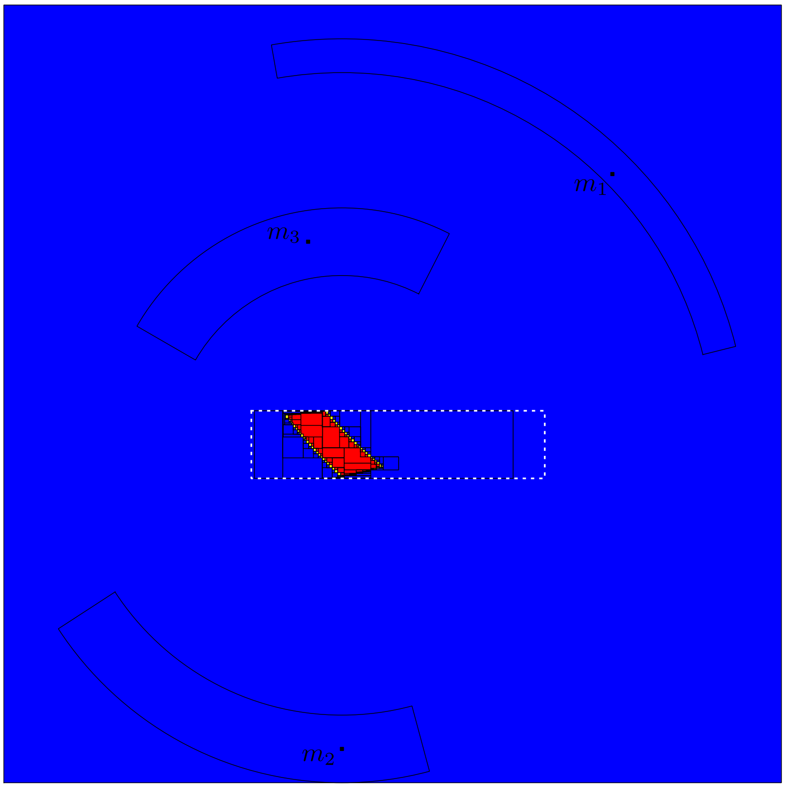


Figure 3.7: Paving obtained using the polar separator. The dashed white box is the result of the first call of a forward/backward separator

The corresponding separator $\mathcal{S}$ is

$$\mathcal{S} = \bigcap_i \mathbf{g}_{\mathbf{m}_i} \circ \mathcal{S}_{\pi^{-1}}^{[\mathbf{y}_i]}. \tag{3.58}$$

The paving obtained using $\mathcal{S}$ is shown in Figure 3.7. The white dashed box correspond to the result of the first call of the forward/backward separator and is bigger than the one obtained with $\mathcal{S}$.

Figure 3.8: VAMA (*Véhicule Anti Mine Autonome*), an AUV owned by DGA TN.

3.4.2 Underwater localization

We now illustrate the efficiency of the polar separator on a data-set extracted from a mission performed by the actual AUV (Autonomous Underwater Vehicle) VAMA (see Figure 3.8) in the Road-stead of Brest (France, Brittany), July 18, 2015. We focus on the transit phase of the mission where the AUV follows a set of waypoints in order to reach its mission area. During this phase, ranges and bearings $\mathbf{y}_k = (y_1^k, y_2^k)$ between the ship at position $\mathbf{m} = (m_1, m_2)$ and the AUV at time k are periodically measured using an *ultra short baseline* (USBL) and then sent to the robot through an acoustic communication. We assume that no communication delays exists. Moreover, thanks to the pressure sensor, the robot can measure its depth and the localization problem can be easily projected on the two-dimensional horizontal plane.

An estimated reconstitution of the part of the mission made by VAMA that will be used here is depicted on Figure 3.9. The motion of the robot is assumed to be described by the discrete-time state equation:

$$\mathbf{x}(k+1) = \boldsymbol{\varphi}_k(\mathbf{x}(k)) = \mathbf{x}(k) + \begin{pmatrix} \cos\psi(k) & -\sin\psi(k) \\ \sin\psi(k) & \cos\psi(k) \end{pmatrix} \cdot dt \cdot \mathbf{v}(k), \tag{3.59}$$

where $\mathbf{x}(k)$ corresponds to the 2D coordinates of the center of the robot at time k expressed in an absolute inertial frame, $\psi(k)$ is the heading and $\mathbf{v}(k)$ is the horizontal speed vector of the robot expressed in its own coordinate system. The speed and heading are measured using a MEMS IMU (Xsens Mti) and are assumed to be known with an uncertainty of $\pm 0.05\, m.s^{-1}$ and $\pm 1°$. The range is measured with an accuracy of $\pm 1\%$ and the bearing with an accuracy of $\pm 2°$.

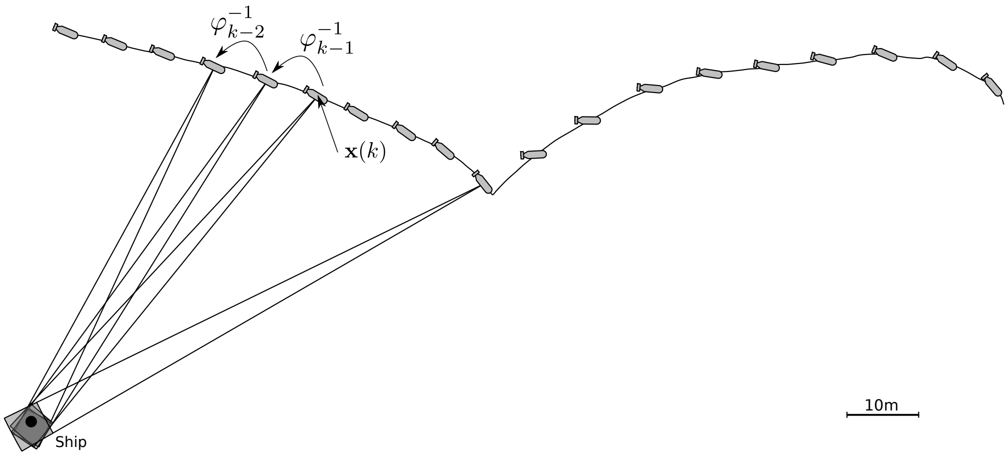


Figure 3.9: Part of the trajectory of VAMA. The robot changes its direction when it reached one way-point.

As a state estimator, we propose to compute the set $\mathbb{X}_k$ of feasible states $\mathbf{x}$ consistent with the last $\bar{i}$ measurements. It is given by

$$\mathbb{X}_k = \left\{\mathbf{x} \in \mathbb{R}^2 \mid \forall i \in \{0, \bar{i}\}, \exists \mathbf{y} \in [\mathbf{y}_i], \mathbf{m}_{k-i} - \boldsymbol{\varphi}_{k:k-i}(\mathbf{x}) = \boldsymbol{\pi}(\mathbf{y}_{k-i})\right\} \tag{3.60}$$

where $\boldsymbol{\varphi}_{k_1:k_2}$ is the flow defined as follows

$$\begin{cases} \boldsymbol{\varphi}_{k_1:k_2} & = & \boldsymbol{\varphi}_{k_2-1:k_2} \circ \cdots \circ \boldsymbol{\varphi}_{k_1:k_1+1} & \quad if\ k_2 \geq k_1 \\ \boldsymbol{\varphi}_{k_1:k_2} & = & \boldsymbol{\varphi}_{k_2:k_1}^{-1} & \quad \text{otherwise.} \end{cases} \tag{3.61}$$

We have

$$\begin{array}{rcl} \mathbb{X}_k & = & \bigcap_{i\in\{0,\bar{i}\}} \left\{\mathbf{x} \in \mathbb{R}^2 \mid \exists \mathbf{y} \in [\mathbf{y}_{k-i}], \mathbf{m}_{k-i} - \boldsymbol{\varphi}_{k:k-i}(\mathbf{x}) = \boldsymbol{\pi}(\mathbf{y})\right\} \\ & = & \bigcap_{i\in\{0,\bar{i}\}} \left\{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{m}_{k-i} - \boldsymbol{\varphi}_{k:k-i}(\mathbf{x}) \in \boldsymbol{\pi}([\mathbf{y}_{k-i}])\right\} \\ & = & \bigcap_{i\in\{0,\bar{i}\}} \boldsymbol{\varphi}_{k-i:k} \circ \mathbf{g}_{\mathbf{m}_{k-i}} \circ \boldsymbol{\pi}\left([\mathbf{y}_{k-i}]\right). \end{array} \tag{3.62}$$

where $\mathbf{g}_{\mathbf{m}_{k-i}}$ is the function associated with the position of the $k - i^{th}$ landmark, defined in Equation 3.55. The corresponding separator is

$$\mathcal{S}_{\mathbb{X}_k} = \bigcap_{i\in\{0,\bar{i}\}} \boldsymbol{\varphi}_{k-i:k} \circ \mathbf{g}_{\mathbf{m}_{k-i}} \circ \mathcal{S}_{\pi^{-1}}^{[\mathbf{y}_{k-i}]}. \tag{3.63}$$

For an implementation of the observer, Equation (3.63), $\mathcal{S}_{\pi^{-1}}^{[\mathbf{y}_{k-i}]}$ can either be the minimal polar separator or a separator based on a classical forward/backward propagation. For a comparison, let us first apply the separator $\mathcal{S}_{\mathbb{X}_k}$ on the initial box $[\mathbf{x}_0] = [-\infty, \infty]^2$ for all k without bisection and with $\bar{i} = 0$. As expected, no inner contraction occurs. The diameters of boxes $[\mathbf{x}_k]$, which enclose $\mathbb{X}_k$, are depicted in Figure 3.10 in blue. It also shows the diameter of the boxes $[\mathbf{x}_k]$ that are obtained with a forward/backward separator (green). The peak, between $t = 40s$

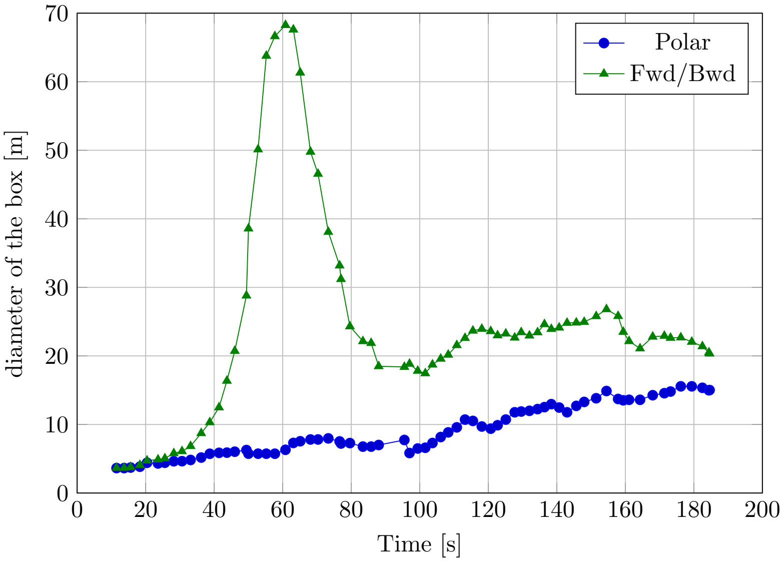


Figure 3.10: Diameter of the boxes$[\mathbf{x}_k]$ obtained by the minimal polar separator (blue) and the forward/backward separator (green). The green peak can be explained by the fact that the forward/backward contractor is less efficient when the angle between the measured angles are almost $\frac{\pi}{4}$.

and $t = 60s$, arises when the measured angles are close to $\frac{\pi}{4}$. We conclude that the forward/backward separator is indeed more pessimistic, which is consistent with the fact that the polar contractor is minimal (see also the video in [Desrochers and Jaulin, 2016a]).

Figure 3.11 shows an approximation of $\mathbb{X}_k$ for $\bar{i} = 5$ and $t = 55\,\text{sec}$. Since there is an acoustic measurement every 5 sec and since the sampling time is $\delta = 0.1\,\text{sec}$, a synchronization of all the data (not discussed here) had to be done. The correspondence between k and t is $k = 5 \cdot t$.

To be robust with respect to outliers, we may allow the observer to relax on the time window of length $\bar{i}$ at most q outliers [Desrochers et al., 2015]. The solution set $\mathbb{X}_k$ becomes:

$$\mathbb{X}_k \quad = \quad \bigcap_{i \in \{0,\bar{i}\}}^{\{q\}} \boldsymbol{\varphi}_{k-i:k} \circ \mathbf{g}_{\mathbf{m}_{k-i}} \circ \boldsymbol{\pi}\left([\mathbf{y}_{k-i}]\right) \tag{3.64}$$

where the q-relaxed intersection [Jaulin and Walter, 2002, Carbonnel et al., 2014] is used. The associated separator becomes

$$\mathcal{S}_{\mathbb{X}_k} = \bigcap_{i \in \{0,\bar{i}\}}^{\{q\}} \boldsymbol{\varphi}_{k-i:k} \circ \mathbf{g}_{\mathbf{m}_{k-i}} \circ \mathcal{S}_{\pi^{-1}}^{[\mathbf{y}_{k-i}]}. \tag{3.65}$$

The result obtained using this observer for $q \in \{0, 1, 2\}$ are illustrated on a video given in [Desrochers and Jaulin,].

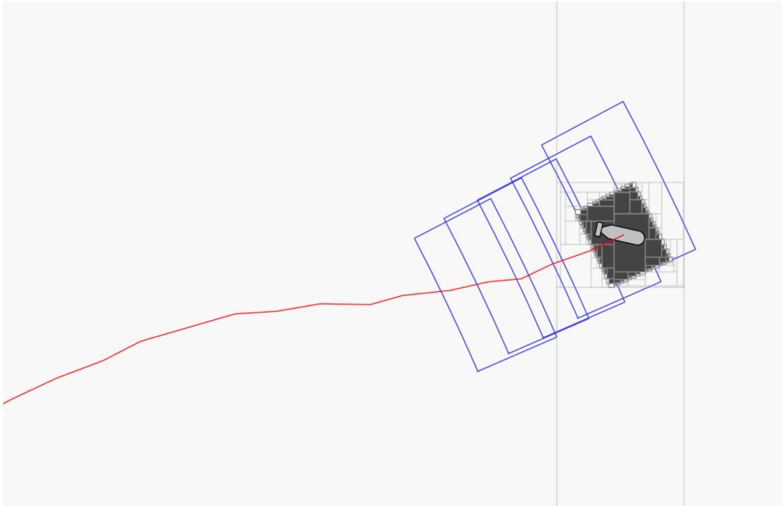


Figure 3.11: Localization of the robot for $t = 55\,\text{sec}$. Each of the blue pies corresponds to all feasible positions of the robot associated with one measurement.

3.4.3 Initial localization from known map

3.4.3.1 Feature-based localization

A feature-based localization problem can be formalized as follows

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) & \text{(evolution equation)} \\ \mathbf{g}(\mathbf{x}(t), \mathbf{y}(t)) \in \mathbb{M} & \text{(observation constraint)} \\ \mathbf{x}(0) \in \mathbb{X}_0 & \text{(initial state)} \end{cases} \tag{3.66}$$

where $\mathbf{x}$ is the unknown state vector, $\mathbf{y}$ an exteroceptive measurement vector, $\mathbf{u}$ a proprioceptive measurement vector. It is known that for all t, $\mathbf{u}(t) \in [\mathbf{u}]\,(t)$ and $\mathbf{y}(t) \in [\mathbf{y}]\,(t)$. The set $\mathbb{M}$ is a map, assumed to be known. Equivalently, the observation constraint can then be written as

$$\exists \mathbf{m}(t) \in \mathbb{M} \mid \mathbf{m}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{y}(t)).$$

If we take into account the uncertainty on $\mathbf{y}$, we can say that the state vector $\mathbf{x}(t)$ is consistent with the interval measurement $[\mathbf{y}]\,(t)$ if

$$\exists \mathbf{m}(t) \in \mathbb{M}, \exists \mathbf{y}(t) \in [\mathbf{y}]\,(t) \mid \mathbf{m}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{y}(t)).$$

The existential quantification $\exists \mathbf{m}(t) \in \mathbb{M}$ highlights the requirement of solving the so-called *data association problem* which aims at finding which point of the map is associated with the measurement vector $\mathbf{y}$. For us, $\mathbb{M}$ is composed with finite number of isolated points and we deal with the initial localization problem on a field of point landmarks that are indistinguishable. All measurements have the same aspect and cannot be associated directly with a particular point of the map.

This problem frequently arises when acoustic sensors are used to detect underwater environmental features [Fallon et al., 2013].

Remark. The observation constraint differs from the shape-based observation constraint given by $\mathbb{Z}(t) = \mathbf{g}(\mathbf{x}(t), \mathbb{M})$ as given in Chapter 2. For the shape observation, the measurement is a shape $\mathbb{Z}$ whereas here it is a vector $\mathbf{y}$. This is consistent with the classical feature-based localization. We have chosen here to consider the more classical feature-based localization to illustrate the use of the polar contractor described in this chapter. Later, it will be used in problems involving shape measurements.

3.4.3.2 Problem

We consider a robot moving on a plane, the motion of which is described by the state equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t)), \mathbf{u}(t)\right) = \left(\begin{array}{c} \left(\begin{array}{cc} \cos(\psi(t)) & -\sin(\psi(t)) \\ \sin(\psi(t)) & \cos(\psi(t)) \end{array} \right) \cdot \mathbf{v}(t) \\ \omega(t) \end{array} \right)$$

The state vector is $\mathbf{x} = (p_x, p_y, \psi)$, where $\mathbf{p} = (p_x, p_y)$ is its position and ψ is its heading. The input vector is $\mathbf{u} = (v_x, v_y, \omega)$, where $\mathbf{v} = (v_x, v_y)$ is the horizontal speed in the vehicle frame, measured for instance with a *Doppler Velocity Log* (DVL) (in case of an underwater robot), and $\omega(.)$ is the angular velocity measured by gyroscopes.

At some times $t_i \in \mathbb{T}$, the robot collects the range-bearing vector $\mathbf{y}(t_i) = (\rho(t_i), \varphi(t_i))^T$ to a landmark $\mathbf{m}(t_i) = (m_x(t_i), m_y(t_i))^T$ which belongs to the map $\mathbb{M}$, composed of a collection of georeferenced points. This leads to the following constraint

$$\exists \mathbf{m} \in \mathbb{M}, \mathbf{m} = \mathbf{g}\left(\mathbf{x}(t_i), \mathbf{y}(t_i)\right) \tag{3.67}$$

with

$$\mathbf{g}(\mathbf{x}(t_i), \mathbf{y}(t_i)) = \begin{pmatrix} p_x(t_i) \\ p_y(t_i) \end{pmatrix} + \rho(t_i) \cdot \begin{pmatrix} \cos(\psi(t_i) + \varphi(t_i)) \\ \sin(\psi(t_i) + \varphi(t_i) \end{pmatrix}. \tag{3.68}$$

The localization problem is thus described by the following set of equations:

$$\left\{ \begin{array}{r} \dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \\ \mathbf{m}(t_i) = \mathbf{g}(\mathbf{x}(t_i), \mathbf{y}(t_i)) \\ \mathbf{m}(t_i) \in \mathbb{M} \\ \mathbf{x}(0) \in \mathbb{X}_0 \end{array} \right. \tag{3.69}$$

where the trajectory $\mathbf{x}(.)$, and the landmark associated to the measurements taken at time t_i both need to be estimated. It can be decomposed into the following constraint network:

$$\begin{cases} (i) & \dot{\mathbf{x}}(.) = \mathbf{f}(\mathbf{x}(.), \mathbf{u}(.)) \\ (ii) & \mathbf{m}(t_i) \in \mathbb{M} \\ (iii) & \mathbf{a}_i = \mathbf{m}(t_i) - \mathbf{p}(t_i) \\ (iv) & \alpha_i = \psi(t_i) + \varphi(t_i) \\ (v) & \mathbf{a}_i = \rho \cdot \begin{pmatrix} \cos(\alpha(t_i)) \\ \sin(\alpha(t_i)) \end{pmatrix} \end{cases} \tag{3.70}$$

where a contractor can be defined for each constraint. In particular, the contractor $\mathcal{C}_{\frac{d}{dt}}$ (see Section 2.5.4) will be used to contract the tube $[\mathbf{x}](.)$ with respect to (i). The map can be depicted by a subpaving, or by an image, for which a contractor introduced in Section 2.5.2 can be used. For any box $[\mathbf{x}] \in \mathbb{IR}^2$, this contractor returns the smallest box which contains all landmarks included in $[\mathbf{x}]$. And the constraint (v) corresponds to the polar one where the minimal contractor, introduced in Section 3.2.3 can be used. The other contractors can be built as explained in Example 2.4.

3.4.3.3 Test-case

Consider an AUV starting its mission with a huge position uncertainty. This can happen during a dive in deep water [McPhail, 2009] or when, for discretion purposes, a long-range underwater transit phase is required to reach the working area.

For operational reasons, no external positioning system, such as acoustic beacons or USBL, are deployed. We assume that a part of the mission area has been previously mapped during a previous survey and this area is large enough to be reached by the AUV. The corresponding map $\mathbb{M}$ describing this area is modeled by a set of 280 point landmarks.

Our robot performs a small mission pattern as depicted in Figure 3.12. It senses its environment using a imaging forward-looking sonar oriented toward the seabed, the scope of which is represented by the blue pie. Every three seconds, it is able to measure the distance and bearing between its pose to some landmarks that range between 10 and 70 meters. The positions of the detected landmarks are depicted by green dots. The 90 red segments represent the measurements. Note that only a small number of mapped landmarks have been detected.

Only the 2D position of the robot and its heading need to be estimated since other state variables (roll, pitch, altitude, depth) are directly measured.

Assumptions. For simplicity, the fact that only landmarks inside the pie are seen by the AUV is not taken into account by the method. Moreover, in an underwater context, the detected landmarks cannot be distinguished from the others, since for instance, two different rocks can have the same aspect and dimension. Moreover, the landmark detection process is sensitive to change in the point of view of the sensor, some landmarks of the map cannot be seen during the survey. Thus, no reliable data associations based on the shape of the landmarks can be assumed. Moreover successive measurements corresponding to the same landmark could be associated by the sonar tracking system. Again, we consider this matching as non reliable and it will not be used for the localization.

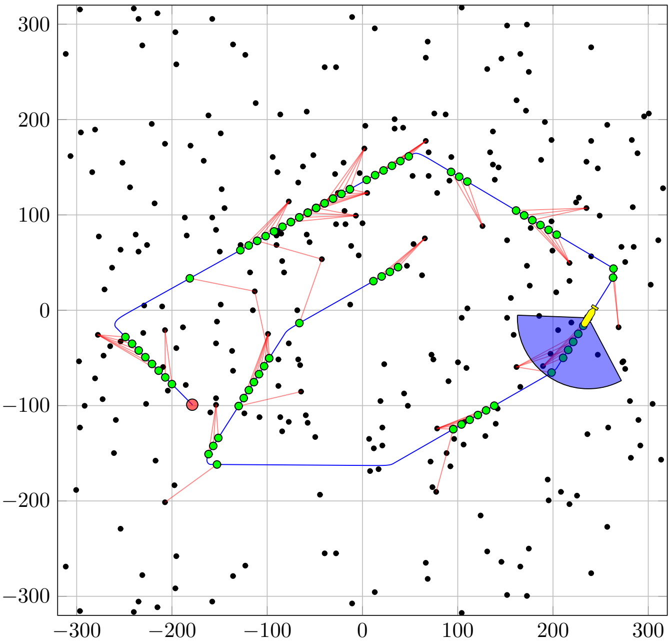


Figure 3.12: The simulated environment for initial localization. The trajectory of the AUV is depicted by the blue lines. Its starting point is drawn by the red dot. The map is composed of 230 landmarks represented by black dots.

This example aims at providing a practical illustration of how the constraint propagation methods can be used to find the set of feasible initial states of the vehicle and to solve the data association problem without using a combinatorial algorithm. Once these two issues are solved, any classical localization method, such as an EKF, can be used to get a more accurate estimate of the trajectory. The filter is initialized with a reliable starting point and only updated with observations which are correctly associated.

3.4.3.4 Results

Given $[\mathbf{x}(0)], \ldots, [\mathbf{x}(k_{max})]$ a set of boxes enclosing the state trajectory of the robot. All boxes are initialized to $\mathbb{R}^2 \times [-\pi, \pi]$. The initial heading is assumed to be known with an accuracy of ±10 degrees. Table 3.2 shows bounds used to quantify the errors on sensor readings. The map $\mathbb{M}$ is composed of 280 landmarks, and 90 observations have been done during the whole mission.

data	description	uncertainty	unit
$\psi(0)$	initial heading	$[-10, 10]$	*deg*
$\mathbf{v}(t)$	linear speed	$[-0.05, 0.05]^{\times 2}$	$m.s^{-1}$
$\omega(t)$	angular velocity	$[-0.001, 0.001]$	$deg.s^{-1}$
$\rho(t)$	range	$\rho(t) \cdot [-0.01, 0.01]$	m
$\psi(t)$	azimuth	$[-1, 1]$	*deg*

Table 3.2: Uncertainties on data used for the application.

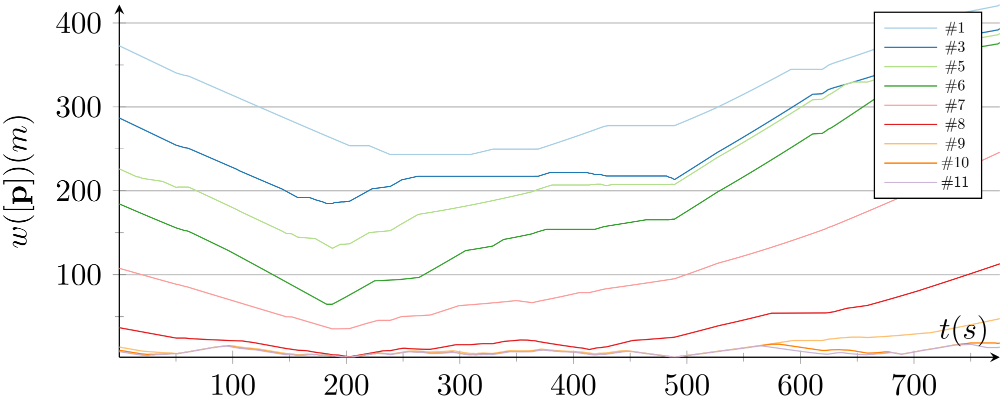


Figure 3.13: $w([\mathbf{p}])$ with respect to the iteration number.

Figure 3.15 shows the final trajectory obtained after 12 iterations in less than 1 minute on *i7-5600U CPU@2.60GHz*.

In Table 3.3, for each iteration, the times needed to contract the whole trajectory are given. This time is constant at each iteration. During the constraint propagation process, the thinner the trajectory, the smaller the number of landmarks contained in $[\mathbf{m}](t_i)$, and *vice versa*. As an indicator, Columns 3 (resp. 4) of Table 3.3 shows the minimal (resp. maximal) number of landmarks contained in $[\mathbf{m}](t_i)$ among all measurements. The last column corresponds to the number of correct associations, *i.e,* when $[\mathbf{m}](t_i)$ contains a single landmark. Figure 3.13 shows diameters of boxes along the trajectory for different iterations which illustrates the constraint propagation process. The constraint propagation methods is shown to be powerful in situations involving a huge number of possible data associations. In comparison, existing method [Fallon et al., 2013] often meet difficulty when both the initial position and the data associations are unknown.

3.5 Conclusion

Contractor-based techniques are particularly attractive when solving engineering applications, due to the fact that they can handle and propagate uncertainties in a context where the equations of the problem are non-linear and non-convex. Now,

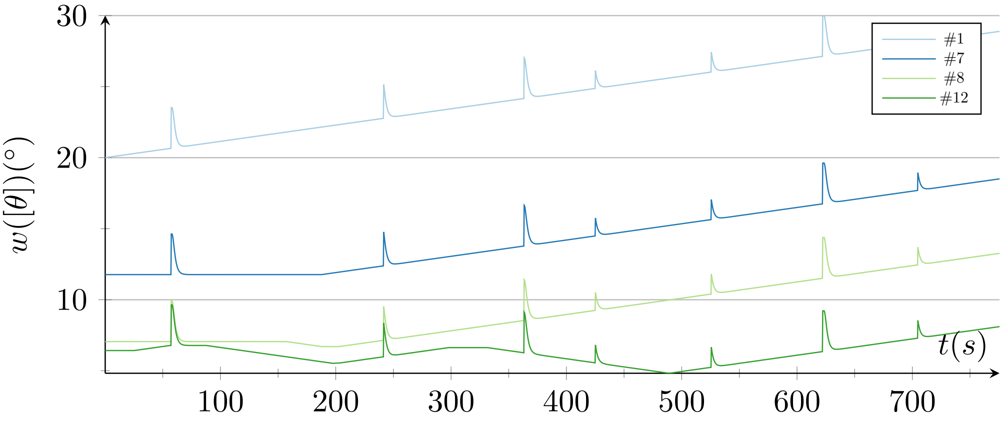


Figure 3.14: $w([\theta])$ with respect to the iteration number. The peaks in the graph are due to the wrapping effect when the vehicle turns.

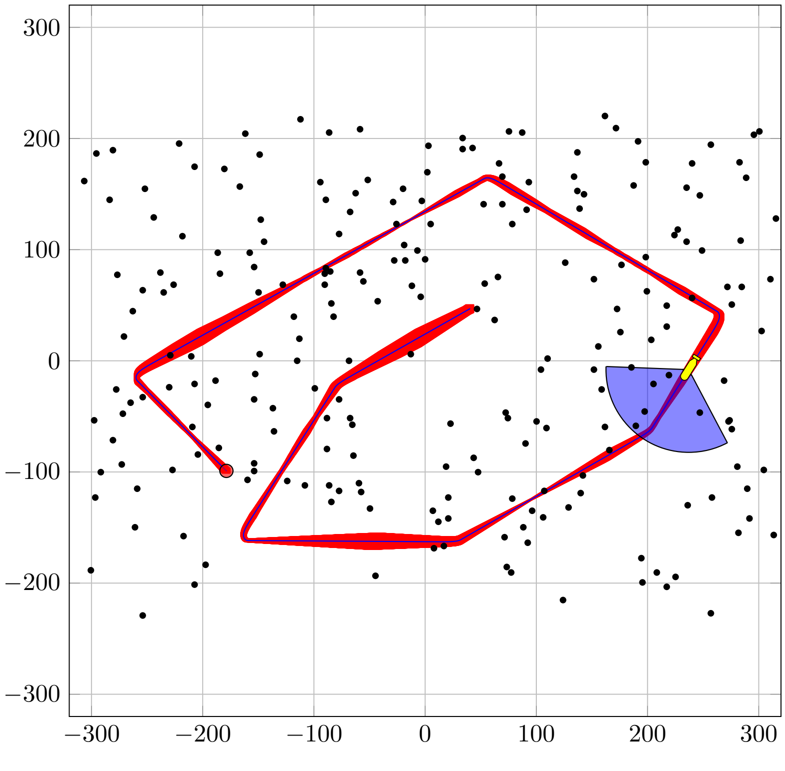


Figure 3.15: Final trajectory with a correct estimation of the initial position and landmarks association. The true trajectory, in blue, belongs to the red tube.

#	time(s)	#min	#max	#ok
1	2.61	202	230	0
2	2.65	27	81	0
3	2.99	22	75	0
4	2.77	15	73	0
5	2.62	11	71	0
6	2.48	4	71	0
7	2.33	2	68	0
8	2.02	1	29	10
9	2.61	1	6	64
10	2.44	1	3	82
11	2.41	1	2	88
12	2.40	1	1	90

Table 3.3: Time needed to contract the whole trajectory for each iteration. #min (resp. #max) is the smallest (resp. greatest) number of elements of $[\mathbf{m}](t_i)$ among all measurements. #ok denotes the number of good associations. The computing time is nearly constant.

the performances of paving methods are extremely sensitive to the accuracy of the contractors that are associated with the equations. One of the equations often met in practice is the polar equation which links Cartesian to polar coordinates.

In this chapter, we proposed two new theorems that could help to build more easily minimal contractors and separators consistent with some specific constraints. These theorems allowed us to build the minimal contractor and a minimal separator for the important polar constraint, which was not done before, to our knowledge.

The efficiency of these new operators and their ability to get an inner and outer approximation of the solution set was illustrated on the problem of the localization of a robot when both goniometric and range measurements are collected. In the next chapter, the polar contractor will be intensively used in the context of the shape-based localization that fits the formalism presented in the introduction of this thesis.

Chapter 4

Shape-based Localization

Contents

4.1 Introduction

In the previous chapter, a dedicated separator, associated with the polar constraint, has been introduced. It allows characterizing a set defined by angles and distances. The efficiency of this separator has been illustrated with localization problems in structured environments composed of landmarks. The variables of these problems are the landmark positions, the pose of the vehicle and the measurements ρ and ϕ. These variables are all real numbers. The domain which contains feasible values was computed using the polar separator. The objective of this previous chapter was to illustrate how minimal contractors / separators could be built for a constraint that is important in our localization context. Now, the existing catalog of primitive contractors built for dedicated geometrical constraints should be extended to incorporate all constraints that could be useful in a robotics context.

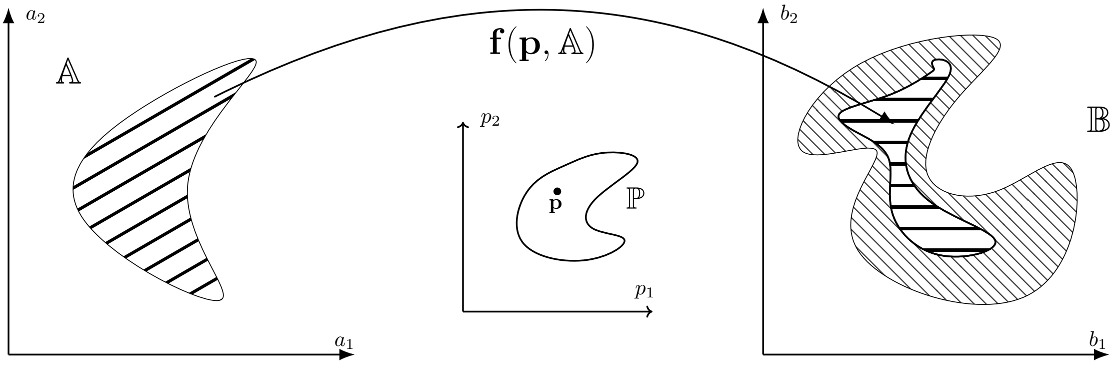


Figure 4.1: Example of a shape registration problem associated with the constraint $\mathbf{f}(\mathbf{p}, \mathbb{A}) \subset \mathbb{B}$. The feasible set $\mathbb{P}$ corresponds to the parameters of the function $\mathbf{f}$ such as the transformation of the shape $\mathbb{A}$ is included in $\mathbb{B}$.

Implementing these specific primitive contractors will allow us to be more efficient by limiting the decompositions (which yields pessimism).

In this chapter, we will assume that both the map $\mathbb{M}$ and measurements $\mathbb{Y}_i$ are subsets of $\mathbb{R}^q$, or shapes, where q is the dimension of the map (2 or 3 in practice). We also assume that shapes are perfectly known; *i.e.* they are thin sets. In the next chapters, this will not be the case since the shape will become uncertain, and we will introduce the notion of interval shape.

This chapter deals with a new type of constraint, called *shape*-registration. Given two shapes $\mathbb{A} \subset \mathbb{R}^n$, $\mathbb{B} \subset \mathbb{R}^m$ and $\mathbf{p} \in \mathbb{R}^p$ linked by a constraint $\mathcal{L}$ that depends on a parameter $\mathbf{p} \in \mathbb{R}^p$, the shape-registration problem aims at finding the set $\mathbb{P}$ of parameter vectors consistent with $\mathcal{L}$. For instance, it may correspond to finding the parameter vectors $\mathbf{p} \in \mathbb{P}$ of a non linear transformation $\mathbf{f}$ such that $\mathbf{f}(\mathbf{p}, \mathbb{A}) \subset \mathbb{B}$. This is illustrated by Figure 4.1.

These problems are related to the registration of points cloud, generally done by *Iterative Closest Points* methods [Pomerleau et al., 2015], images [Zitova and Flusser, 2003] and surfaces [Audette et al., 2000]. But we are in a set-membership context.

As the main illustration, the localization problem of a robot with sonar range-finder in an unstructured environment is handled. This problem is considered as difficult due to the fact that the sonar returns a measurement under the form of an impact point inside an emission cone. This specific type of measurement makes the problem partially observable. Moreover, our environment is not represented by geometric features such as segments or disks, but by a shape which cannot be translated into analytical constraints. Now, as shown in Figure 2.5, an unstructured map can be cast into a contractor form which allows us to use contractor/separator algebra.

Here, we propose first to use a separator-based method to perform a reliable simulation necessary to generate realistic data (see [Taha et al., 2015] for a survey on reliable simulation). Then, once these data have been generated, we consider the inverse problem, *i.e.*, the robot localization with large-cone sonar measurements in an unstructured map. This problem has never been considered yet, to our knowledge, at least in an unstructured environment (see *e.g.*, [Jaulin et al., 2000, Kieffer et al., 1999, Jaulin et al., 2002, Lévêque et al., 1997, Colle and Galerne, 2013] in the case where the map is made with geometrical features). We will also show the link with Minkowski operations and propose separator counterparts for these operations.

Section 4.2 presents the concept of shape registration and shows that it can be interpreted as a set projection problem. The principle is illustrated on a localization problem. Section 4.3 proposes to formulate the Minkowski operations as a specific shape registration problem where the transformations correspond to translations. Section 4.4 illustrates the application of the Minkowski operation to the problem of localization of a robot in an unstructured environment. Section 4.5 concludes the chapter.

4.2 Shape Registration as a Projection

Many sets that are defined with quantifiers can be described in terms of projection, inversion, complement, and composition. The projection operator is now defined with its associated separator.

4.2.1 Projection of separators

Definition 4.1. Given $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{Y} \subset \mathbb{R}^p$. If $\mathbb{Z} = \mathbb{X} \times \mathbb{Y}$, then the projection of a subset $\mathbb{Z}_1$ of $\mathbb{Z}$ onto $\mathbb{X}$ (with respect to $\mathbb{Y}$) is defined as:

$$proj_{\mathbb{X}}(\mathbb{Z}_1) = \{\mathbf{x} \in \mathbb{X} \mid \exists \mathbf{y} \in \mathbb{Y}, (\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_1\}\,. \tag{4.1}$$

Figure 4.2 illustrates the projection of a set along the $\mathbb{X}$ dimension.

When a separator is available for $\mathbb{Z}_1$, it is possible to define a separator consistent with $proj_{\mathbb{X}}(\mathbb{Z}_1)$ based on the following theorem.

Definition 4.2. Consider a separator $\mathcal{S}([\mathbf{x}], [\mathbf{y}]) = \{\mathcal{S}^{\text{in}}([\mathbf{x}], [\mathbf{y}]), \mathcal{S}^{\text{out}}([\mathbf{x}], [\mathbf{y}])\}$, the projection of $\mathcal{S}$ is defined by :

$$proj_{\mathbf{x}}(\mathcal{S})([\mathbf{x}]) = \left\{ \bigcap_{\mathbf{y}\in[\mathbf{y}]} \partial_{\mathbf{x}} \mathcal{S}^{in}([\mathbf{x}], [\mathbf{y}]),\ \bigcup_{\mathbf{y}\in[\mathbf{y}]} \partial_{\mathbf{x}} \mathcal{S}^{out}([\mathbf{x}], [\mathbf{y}]) \right\} \tag{4.2}$$

Theorem 4.1. *Given a separator $\mathcal{S}$ associated to $\mathbb{Z}$, we have:*

$$proj_{\mathbf{x}}(\mathbb{Z}) \sim proj_{\mathbf{x}}(\mathcal{S}). \tag{4.3}$$

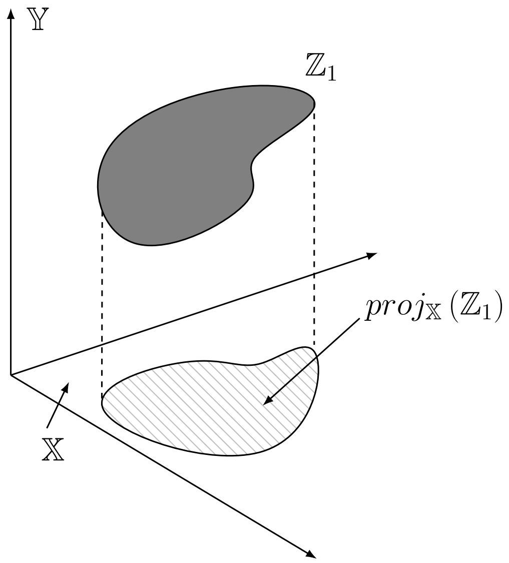


Figure 4.2: Set projection

Proof. To prove 4.3, we must prove that the outer contractor is consistent with $proj_{\mathbf{x}}(\mathbb{Z})$ and the inner with its complement. We have:

$$\begin{aligned} proj_{\mathbf{x}}(\mathbb{Z}) &= \{\mathbf{x} \mid \exists \mathbf{y} \in [\mathbf{y}], (\mathbf{x},\mathbf{y}) \in \mathbb{Z}\} && (4.4)\\ &= \bigcup_{\mathbf{y}\in[\mathbf{y}]} \{\mathbf{x}, (\mathbf{x},\mathbf{y}) \in \mathbb{Z}\} && (4.5)\\ &\sim \bigcup_{\mathbf{y}\in[\mathbf{y}]} \partial_{\mathbf{x}}\mathcal{S}^{out} && (4.6)\end{aligned}$$

Similarly, since $\overline{\mathbb{Z}} \sim \mathcal{S}^{in}$, we have:

$$\begin{aligned} \overline{proj_{\mathbf{x}}(\mathbb{Z})} &= \overline{\{\mathbf{x} \mid \exists \mathbf{y} \in [\mathbf{y}], (\mathbf{x},\mathbf{y}) \in \mathbb{Z}\}} && (4.7)\\ &= \{\mathbf{x} \mid \forall \mathbf{y} \in [\mathbf{y}], (\mathbf{x},\mathbf{y}) \notin \mathbb{Z}\} && (4.8)\\ &= \bigcap_{\mathbf{y}\in[\mathbf{y}]} \left\{\mathbf{x}, (\mathbf{x},\mathbf{y}) \in \overline{\mathbb{Z}}\right\} && (4.9)\\ &\sim \bigcap_{\mathbf{y}\in[\mathbf{y}]} \partial_{\mathbf{x}}\mathcal{S}^{in} && (4.10)\end{aligned}$$

that ends the proof. □

The implementation of the projection of separators is based on *Proj-Union* and *Proj-intersection* algorithms described in [Chabert and Jaulin, 2009b]. Roughly, the initial domain $[\mathbf{y}]$ is split into a list of small intervals $[\mathbf{y}_i]$. For the outer contraction, each sub-domain $[\mathbf{x}] \times [\mathbf{y}_i]$ is contracted using the outer contractor $\partial_{\mathbf{x}}\mathcal{S}^{out}([\mathbf{x}], [\mathbf{y}_i])$ and the hull of the results is returned. For the inner contraction, a sequence of points $\mathbf{y}_i$ are sampled from $[\mathbf{y}_i]$ and $\partial_{\mathbf{x}}\mathcal{S}^{in}$ is enforced on $[\mathbf{x}] \times \{\mathbf{y}_i\}$. The intersection of the results is returned.

In practice, many tricks are used to limit the number of separations. A specific implementation that tried to take advantage of the complementarity of the inner and outer contractions of $\mathcal{S}$ has been done. The implementation provided in *pyIbex* fits the generic separator formalism which can be used with border-based separators. It does not rely on explicit inner and outer contractors. Note that the proposed projection algorithm assumes that set $\mathbb{Z}$ has a non-empty volume in order to define an inner contractor.

4.2.2 Shape registration

In this section, we formulate the shape registration as a set projection. Consider a function

$$\mathbf{f} : \left\{ \begin{array}{ccc} \mathbb{R}^n \times \mathbb{R}^p & \longrightarrow & \mathbb{R}^m \\ (\mathbf{a}, \mathbf{p}) & \longrightarrow & \mathbf{f}(\mathbf{a}, \mathbf{p}) \end{array} \right. \tag{4.11}$$

For a given $\mathbf{p} \in \mathbb{R}^p$, $\mathbb{A} \subset \mathbb{R}^n$, $\mathbb{B} \subset \mathbb{R}^m$ we shall use the following notations:

$$\begin{array}{ll} \mathbf{f}(\mathbb{A}, \mathbf{p}) & = \{\mathbf{b} | \exists \mathbf{a} \in \mathbb{A}, \, \mathbf{b} = \mathbf{f}(\mathbf{a}, \mathbf{p})\} \\ \mathbf{f}^{-1}(\mathbb{B}) & = \{\mathbf{z} = (\mathbf{a}, \mathbf{p}) \, | \exists \mathbf{b} \in \mathbb{B}, \, \mathbf{b} = \mathbf{f}(\mathbf{a}, \mathbf{p})\} \end{array} \tag{4.12}$$

Consider the set defined by :

$$\mathbb{P} = \{\mathbf{p} \in \mathbb{R}^p \mid \mathbf{f}(\mathbb{A}, \mathbf{p}) \subset \mathbb{B}\}. \tag{4.13}$$

The vector $\mathbf{p}$ corresponds to a parameter vector associated to a transformation. A transformation vector $\mathbf{p}$ is consistent if after transformation of $\mathbb{A}$ it is included inside $\mathbb{B}$. We have

$$\begin{array}{cl} & \mathbf{f}(\mathbb{A}, \mathbf{p}) \subset \mathbb{B} \\ \Leftrightarrow & \forall \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B} \\ \Leftrightarrow & \neg \exists \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \overline{\mathbb{B}} \\ \Leftrightarrow & \neg \exists \mathbf{a} \in \mathbb{A}, (\mathbf{a}, \mathbf{p}) \in \mathbf{f}^{-1}\left(\overline{\mathbb{B}}\right) \\ \Leftrightarrow & \neg \exists \mathbf{a}, (\mathbf{a}, \mathbf{p}) \in \mathbb{A} \times \mathbb{R}^p \, \wedge \, (\mathbf{a}, \mathbf{p}) \in \mathbf{f}^{-1}\left(\overline{\mathbb{B}}\right). \end{array} \tag{4.14}$$

As a consequence,

$$\mathbb{P} = \overline{proj_{\mathbf{p}}\{(\mathbb{A} \times \mathbb{R}^p) \cap \mathbf{f}^{-1}\left(\overline{\mathbb{B}}\right)\}}. \tag{4.15}$$

Therefore, if we have separators $\mathcal{S}_{\mathbb{A}}, \mathcal{S}_{\mathbb{B}}$ for $\mathbb{A}, \mathbb{B}$ then a separator $\mathcal{S}_{\mathbb{P}}$ for $\mathbb{P}$ can be obtained using the separator algebra [Jaulin and Desrochers, 2014]. It is given by

$$\mathcal{S}_{\mathbb{P}} = \overline{proj_{\mathbf{p}}\{(\mathcal{S}_{\mathbb{A}} \times \mathcal{S}_{\mathbb{R}^p}) \cap \mathbf{f}^{-1}\left(\overline{\mathcal{S}_{\mathbb{B}}}\right)\}}. \tag{4.16}$$

Combining this separator with a paver, we are able to obtain an inner and outer approximation of $\mathbb{P}$.

Remark 4.1. The same method applies to build the separator consistent with the sets:

$$\begin{aligned} \mathbb{P}_{=} &= \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} = \emptyset\} & (4.17)\\ &= \left\{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}, \mathbf{p}) \subset \overline{\mathbb{B}}\right\} & (4.18)\\ \text{or} \quad \mathbb{P}_{\neq} &= \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} \neq \emptyset\} & (4.19)\\ &= \overline{\mathbb{P}_{=}} & (4.20) \end{aligned}$$

4.2.3 Example

A robot at position $(0, 0)$ is inside an environment defined by the map

$$\mathbb{M} = \{\mathbf{x} \in \mathbb{R}^2 \mid x_1 < 5 \text{ or } x_2 < 3\}. \tag{4.21}$$

It emits an ultrasonic sound in the cone with angles $\frac{\pi}{4} \pm \frac{\pi}{24}$. For a simulation purpose, we want to compute the distance d returned by the sonar. This distance corresponds to the shortest distance inside the emission cone to the complement of the map:

$$d = \inf\{d \mid \mathbf{f}(\mathbb{S}_1, d) \cap \overline{\mathbb{M}} \neq \emptyset\}, \tag{4.22}$$

where $\mathbf{f}(\mathbf{x}, d) = d \cdot \mathbf{x}$ is the scaling function and $\mathbb{S}_1$ is the unit cone defined by

$$\mathbb{S}_1 = \{(x, y) \mid x^2 + y^2 < 1 \text{ and } atan2(y, x) \in [\frac{5\pi}{24}, \frac{7\pi}{24}]\}. \tag{4.23}$$

Recall that for this type of set, a minimal separator has been introduced in Chapter 3. Equivalently, we have

$$d = \sup\{d \mid \mathbf{f}(\mathbb{S}_1, d) \subset \mathbb{M}\}. \tag{4.24}$$

To solve our problem, we first characterize the set:

$$\mathbb{D} = \{d \mid \mathbf{f}(\mathbb{S}_1, d) \subset \mathbb{M}\}. \tag{4.25}$$

which corresponds to a registration problem. We get

$$[0, 6.2988] \subset \mathbb{D} \subset [0, 6.3085]. \tag{4.26}$$

The situation is depicted on Figure 4.3. As a consequence, the true distance d returned by the sensor satisfies $d \in [6.2988, 6.3085]$

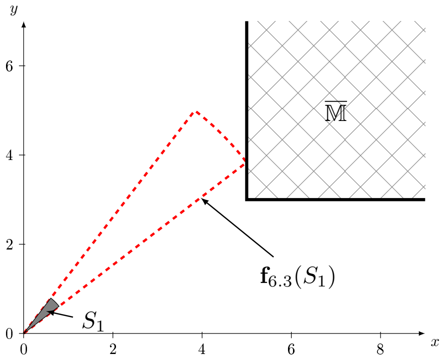


Figure 4.3: The map $\mathbb{M}$ is represented by the white space outside the hatched area while the unit pie $\mathbb{S}_1$ is painted in gray. The red pie represents the lower upper bound of $\mathbb{D}$ which almost touches the border of the map.

4.3 Minkowsky Sum and Difference

The elementary inclusion shape registration defined by Equation (4.13) has been illustrated in the previous section. Our aim is to derive efficient shape registration algorithms, based on Minkowski operations, to deal with the special case where the transformations are translations.

Minkowski operations are used in morphological mathematics to perform dilation or inflation of sets. As it will be shown in Section 4.4, it can also be used for a localization purpose. Efficient algorithms (see [Shoji, 1991]) have been proposed to perform Minkowski operations with sets represented by subpavings. In this section, we show that Minkowski sum and difference can be seen as a shape registration problem. This will allow us to build separators for these Minkowski operations. In the next paragraphs, Minkowski difference and addition are defined with their associated separators. This will be always true for our applications.

4.3.1 Minkowski difference

Definition 4.3. Given two sets $\mathbb{A} \in \mathcal{P}(\mathbb{R}^n)$, $\mathbb{B} \in \mathcal{P}(\mathbb{R}^n)$, the Minkowski difference [Najman and (Eds), 1000], denoted $\ominus$, is defined by

$$\mathbb{B} \ominus \mathbb{A} = \{\mathbf{p} \mid \mathbb{A} + \mathbf{p} \subset \mathbb{B}\}. \tag{4.27}$$

Proposition 4.1. Given two separators $\mathcal{S}_{\mathbb{A}}$ and $\mathcal{S}_{\mathbb{B}}$ for $\mathbb{A}$ and $\mathbb{B}$. The Minkowski difference of two separators, defined by

$$\mathcal{S}_{\mathbb{B}} \ominus \mathcal{S}_{\mathbb{A}} = \overline{proj_{\mathbf{p}}\{(\mathcal{S}_{\mathbb{A}} \times \mathcal{S}_{\mathbb{R}^n}) \cap \mathbf{f}^{-1}\left(\overline{\mathcal{S}_{\mathbb{B}}}\right)\}}. \tag{4.28}$$

where $\mathbf{f}(\mathbf{p}, \mathbf{a}) = \mathbf{a} + \mathbf{p}$, is a separator for $\mathbb{B} \ominus \mathbb{A}$.

Proof. Computing the Minkowski difference can be seen as a specific shape registration problem where $\mathbf{f}(\mathbf{p},\mathbf{a}) = \mathbf{a}+\mathbf{p}$, *i.e.*, the transformation corresponds to a translation of vector $\mathbf{p}$. As a consequence,

$$\mathbb{B}\ominus\mathbb{A} = \overline{proj_{\mathbf{p}}\{(\mathbb{A}\times\mathbb{R}^p)\cap\mathbf{f}^{-1}\left(\overline{\mathbb{B}}\right)\}}. \tag{4.29}$$

□

According to Equation (4.16) a separator can thus be built for $\mathbb{B}\ominus\mathbb{A}$ and a paver is then able to characterize $\mathbb{B}\ominus\mathbb{A}$. The following example illustrates the Minkowsky difference of a circle by a box.

Example 4.1. Let $\mathbb{A}$ be a rectangle of side's length of 4 x 2, and $\mathbb{B}$ be a disk of radius 5. The resulting solution set $\mathbb{B}\ominus\mathbb{A}$ is depicted in Figure 4.4.

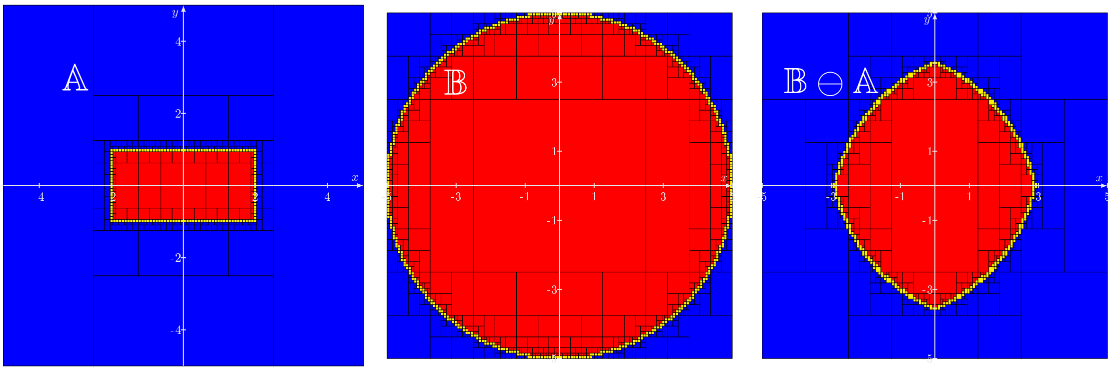


Figure 4.4: Minkowski difference of the disk $\mathbb{B}$ and the rectangle $\mathbb{A}$

4.3.2 Minkowski addition

Definition 4.4. Given two sets $\mathbb{A}\in\mathcal{P}(\mathbb{R}^n)$, $\mathbb{B}\in\mathcal{P}(\mathbb{R}^n)$, the Minkowski sum, denoted by $\oplus$, is defined by:

$$\mathbb{A}\oplus\mathbb{B} = \{\mathbf{a}+\mathbf{b}, \mathbf{a}\in\mathbb{A}, \mathbf{b}\in\mathbb{B}\}. \tag{4.30}$$

Proposition 4.2. Given two separators $\mathcal{S}_{\mathbb{A}}$ and $\mathcal{S}_{\mathbb{B}}$ for $\mathbb{A}$ and $\mathbb{B}$. The Minkowski sum of two separators defined by

$$\mathcal{S}_{\mathbb{A}}\oplus\mathcal{S}_{\mathbb{B}} = \overline{\overline{\mathcal{S}_{\mathbb{B}}}\ominus -\mathcal{S}_{\mathbb{A}}}. \tag{4.31}$$

is a separator for $\mathbb{A}\oplus\mathbb{B}$.

Proof. We have:

$$\begin{array}{rcl} \mathbb{A} \oplus \mathbb{B} & = & \{\mathbf{p} \mid \exists \mathbf{a} \in \mathbb{A}, \exists \mathbf{b} \in \mathbb{B}, \mathbf{p} = \mathbf{a} + \mathbf{b}\} \\ & = & \{\mathbf{p} \mid \exists \mathbf{a} \in \mathbb{A}, \exists \mathbf{b} \in \mathbb{B}, \mathbf{p} - \mathbf{a} = \mathbf{b}\} \\ & = & \{\mathbf{p} \mid (\mathbf{p} - \mathbb{A}) \cap \mathbb{B} \neq \emptyset\} \\ & = & \overline{\{\mathbf{p} \mid (\mathbf{p} - \mathbb{A}) \cap \mathbb{B} = \emptyset\}} \\ & = & \overline{\{\mathbf{p} \mid (\mathbf{p} + (-\mathbb{A})) \subset \overline{\mathbb{B}}\}} \text{(see Eq 4.27)} \\ & = & \overline{\overline{\mathbb{B}} \ominus -\mathbb{A}}. \end{array} \tag{4.32}$$

Thus, a separator for the set $\mathbb{A} \oplus \mathbb{B}$ is $\overline{\overline{\mathcal{S}_{\mathbb{B}}} \ominus -\mathcal{S}_{\mathbb{A}}}$. □

Example 4.2. Consider a triangle $\mathbb{A}$ and a square $\mathbb{B}$. The Minkowski addition $\mathbb{A} \oplus \mathbb{B}$ is shown on Figure 4.5.

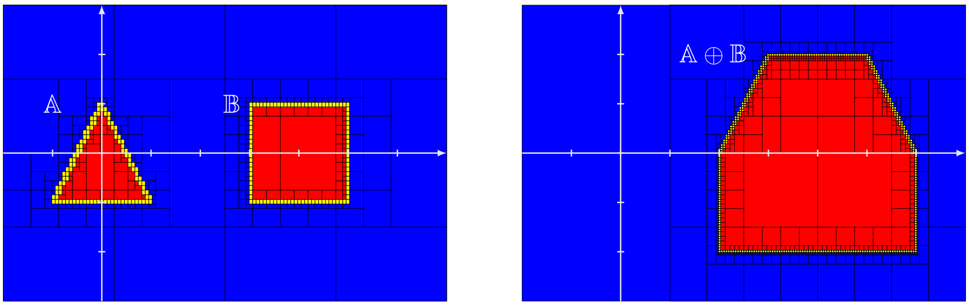


Figure 4.5: Minkowski sum of a square $\mathbb{B}$ and a triangle $\mathbb{A}$ computed using a set membership algorithm.

4.4 Localization in an Unstructured Environment

Consider a robot $\mathcal{R}$ at position $\mathbf{p} = (p_1, p_2)$ in an unstructured environment described by the set $\mathbb{M}$. We assume that the heading θ of $\mathcal{R}$ is known with a good accuracy (for instance, by using a compass) and does not need to be estimated. The robot is equipped with several sonars which return the distance between the robot and the map with respect to their emission cone. This section deals with the localization of the robots using a shape registration method. Several authors have already studied this problem using interval analysis [Lévêque et al., 1997, Meizel et al., 2002, Colle and Galerne, 2013] but in an environment made with segments.

Each sensor emits an acoustic wave in its direction α_i which propagates inside a cone of half angle γ corresponding to the aperture of the beam. By measuring the time lag between the emission and the reception of the wave, reflected by the map, an interval $[d_i] = [d_i^-, d_i^+]$ contains the true distance d_i to the nearest obstacle which lies in the scope of the sensor can be obtained. The situation is depicted in Figure 4.6. The area swept by the wave between 0 and d_i is free of obstacles whereas the map is hit by the wave at distance d_i. Define

$$\begin{array}{rcl} \mathbb{S}_i & = & \{(x,y) \mid x^2+y^2 < d_i^- \text{ and } atan2(y,x) \in [\alpha_i-\gamma, \alpha_i+\gamma]\} \\ \Delta\mathbb{S}_i & = & \{(x,y) \mid x^2+y^2 \in [d_i] \text{ and } atan2(y,x) \in [\alpha_i-\gamma, \alpha_i+\gamma]\} \end{array} \tag{4.33}$$

The set $\mathbb{S}_i$ is called the *free sector* and $\Delta\mathbb{S}_i$ is called the *impact pie*. These sets are depicted on Figure 4.6.

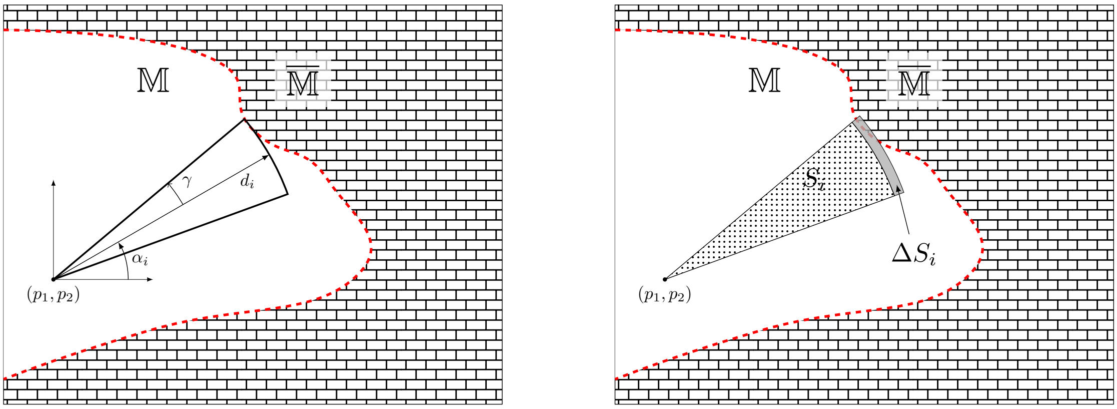


Figure 4.6: Sensor model used by the robot. The map $\mathbb{M}$, in white, corresponds to parts of the space without obstacles.

The set of all feasible positions $\mathbb{P}$ consistent with $[d_i]$ is

$$\begin{aligned} \mathbb{P}(i) &= \{\mathbf{p} \in \mathbb{R}^2 \mid (\mathbf{p}+\mathbb{S}_i) \subset \mathbb{M} \text{ and } (\mathbf{p}+\Delta\mathbb{S}_i) \cap \overline{\mathbb{M}} \neq \emptyset\} \\ &= (\mathbb{M} \ominus \mathbb{S}_i) \cap (\overline{\mathbb{M}} \oplus -\Delta\mathbb{S}_i). \end{aligned} \tag{4.34}$$

With several measurements $[d_i]$ the set of all positions consistent with all data is s

$$\mathbb{P} = \bigcap_i (\mathbb{M} \ominus \mathbb{S}_i) \cap (\overline{\mathbb{M}} \oplus -\Delta\mathbb{S}_i). \tag{4.35}$$

Denote by $\mathcal{S}_{\mathbb{M}}$, $\mathcal{S}_{\mathbb{S}_i}$, $\mathcal{S}_{\Delta\mathbb{S}_i}$ separators for $\mathbb{M}$, $\mathbb{S}_i$, $\Delta\mathbb{S}_i$. Then a separator for $\mathbb{P}$ is

$$\mathcal{S}_{\mathbb{P}} = \bigcap_i (\mathcal{S}_{\mathbb{M}} \ominus \mathcal{S}_{\mathbb{S}_i}) \cap (\overline{\mathcal{S}_{\mathbb{M}}} \oplus -\mathcal{S}_{\Delta\mathbb{S}_i}). \tag{4.36}$$

As an illustration, consider the situation described by Figure 4.7 (left), where a robot collects 6 sonar data. The width of the intervals corresponding to the range measurement is $\pm 1m$. The first measurement corresponding to $i = 1$ is painted in green. Figure 4.7 (right) corresponds to an approximation of the set $\mathbb{M} \ominus \mathbb{S}_1$, obtained using the separator $\mathcal{S}_{\mathbb{M}} \ominus \mathcal{S}_{\mathbb{S}_1}$ with a paver.

Figure 4.8 (left) corresponds to an approximation of the set $\overline{\mathbb{M}} \oplus -\Delta\mathbb{S}_1$, obtained using a paver with the separator $\overline{\mathcal{S}_{\mathbb{M}}} \oplus -\mathcal{S}_{\Delta\mathbb{S}_1}$. It corresponds to the set of positions for the robot such that the impact pie $\Delta\mathbb{S}_1$ intersects the outside of the map $\mathbb{M}$.

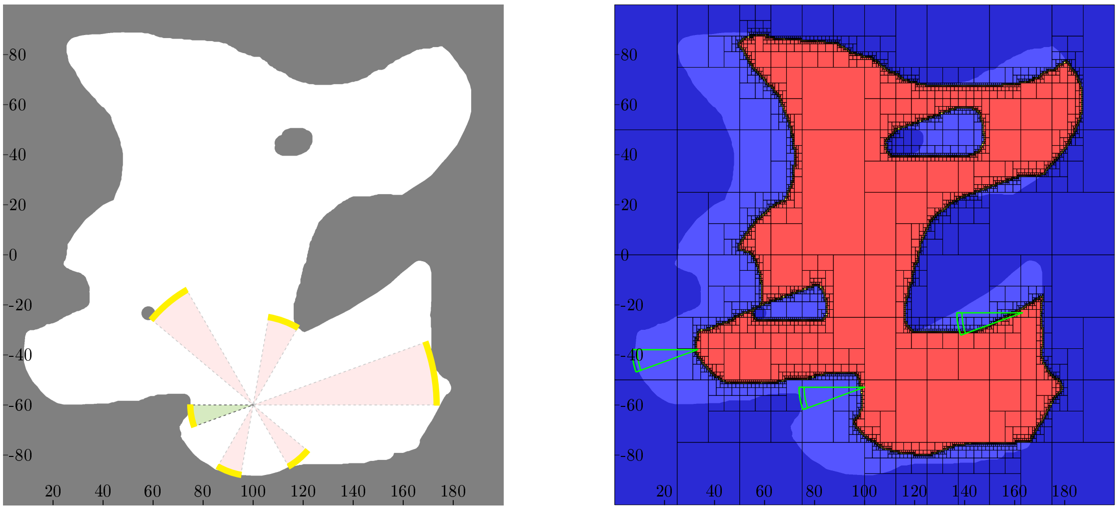


Figure 4.7: Left: a robot which collects 6 sonar range measurements. All free sectors $\mathbb{S}_i$ are included in the map $\mathbb{M}$ (in white) and the impact pies $\Delta\mathbb{S}_i$ in yellow intersect $\overline{\mathbb{M}}$. Right: set of all positions for the robot consistent with the fact that the free sector $\mathbb{S}_1$ (in green on the left picture) is inside $\mathbb{M}$.

Figure 4.8 (right) corresponds to the set $(\mathbb{M} \ominus \mathbb{S}_1) \cap (\overline{\mathbb{M}} \oplus -\Delta\mathbb{S}_1)$, *i.e.*, it contains the position consistent with both $\Delta\mathbb{S}_1$ and $\mathbb{S}_1$.

Figure 4.9 (left) corresponds to an approximation of the set $\mathbb{P}$, obtained using a paver on with the separator $\mathcal{S}_{\mathbb{P}}$. It corresponds to the set of positions for the robot that all six impact pies $\Delta\mathbb{S}_i$ intersect the outside of the map $\mathbb{M}$ and all six free sectors are inside $\mathbb{M}$. A zoom of the solution set is given in Figure 4.9 (right). The computing time is 127 sec. and 205 boxes have been generated. Note that obtaining an inner approximation of the solution set was not possible using existing approaches that are not based on separators.

4.5 Conclusion

Separator-based techniques are particularly attractive when solving engineering applications, due to the fact that they can handle and propagate uncertainties in a context where the equations of the problem are non-linear and non-convex.

Now, this chapter extended the class of constraints by considering the shape registration problem. For instance, the classical set inversion problem can be described by the following constraint network

$$\left\{\begin{array}{rl} \text{Variable:} & \mathbf{x} \in \mathbb{R}^n \\ \text{Constraint:} & \mathbf{f}(\mathbf{x}) \in \mathbb{Y} \\ \text{Domain:} & \mathbb{X} \subset \mathbb{R}^n \end{array}\right. \tag{4.37}$$

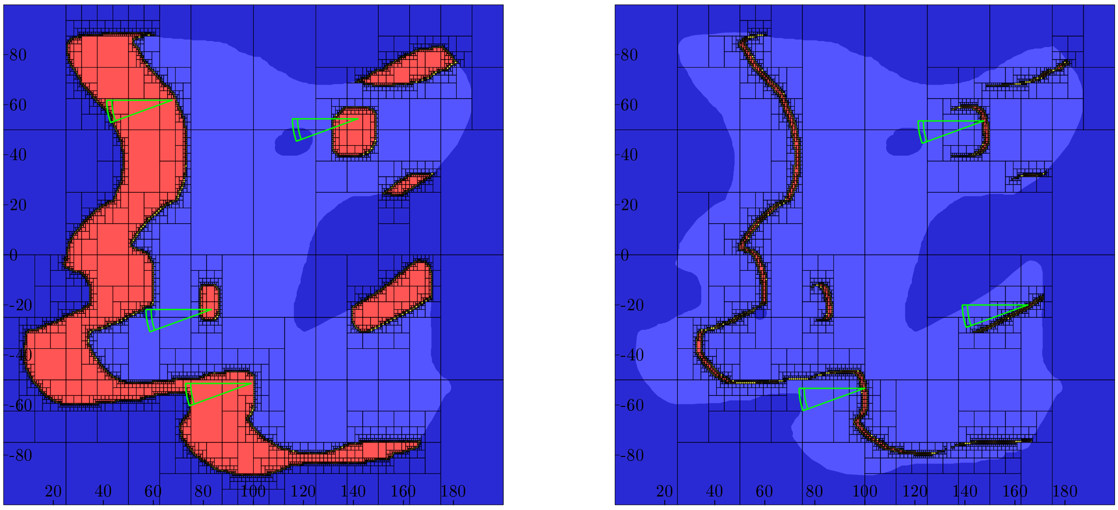


Figure 4.8: Left: Positions for the robot consistent with the impact pie $\Delta\mathbb{S}_1$. Right: Positions consistent with the free sector $\mathbb{S}_1$ and the impact pie $\Delta\mathbb{S}_1$.

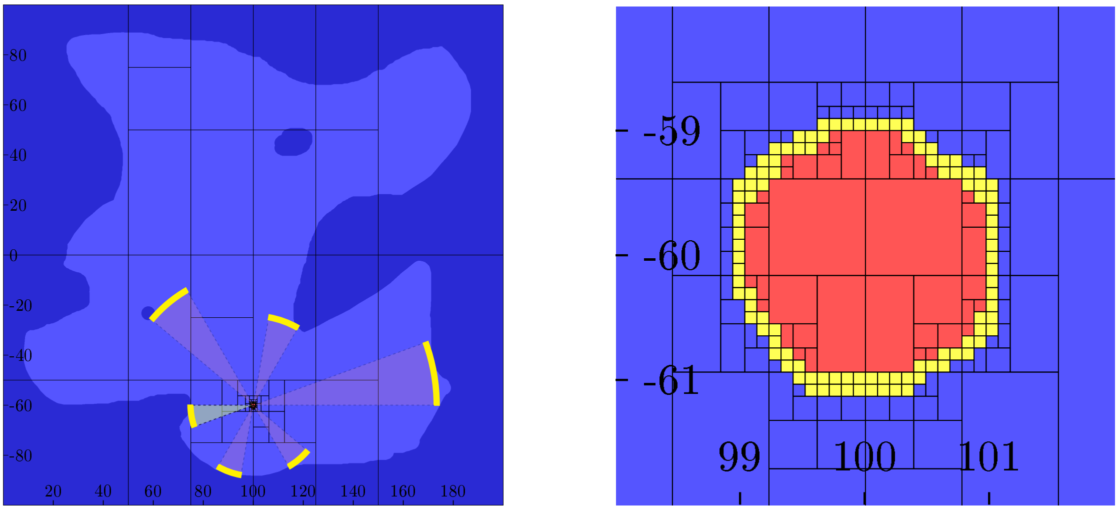


Figure 4.9: Left: Set of positions $\mathbb{P}$ for the robot consistent with all six free sectors $\mathbb{S}_i$ and all impact pies $\Delta\mathbb{S}_i$. Right: zoom around the solution set $\mathbb{P}$.

where $\mathbf{f} \in \mathcal{F}(\mathbb{R}^n, \mathbb{R}^m)$ and $\mathbb{Y}$ is a known set that comes from input data. Now, to deal with unstructured environments, a new constraint which links sets has been introduced

$$\left\{ \begin{array}{cc} \text{Variable :} & \mathbf{p} \in \mathbb{R}^p \\ \text{Constraint :} & \mathbf{f}_\mathbf{p}(\mathbb{X}) \subset \mathbb{Y} \\ \text{Domain:} & \mathbb{P} \subset \mathbb{R}^p \end{array} \right. \tag{4.38}$$

where $\mathbf{f} \in \mathcal{F}(\mathbb{R}^n \times \mathbb{R}^p, \mathbb{R}^m)$ is a parametric function and $\mathbb{X}$, $\mathbb{Y}$ are two known sets.

As illustrated by the application, this formulation allows us to deal with unstructured data sets (the shapes) that represent a part of the environment (the map) or a measurement (area obstacle-free). Sensor readings are not described by scalars, with an interval as a domain, but by a shape. The goal of the next part is to address the case where the shapes $\mathbb{X}$ and $\mathbb{Y}$ are variables of our *Constraint N*etwork. This happens with the SLAM problem inwhich the map is unknown. This issue will be discussed more deeply in Chapter 6.

Part II

Shape Valued Constraint Network

Chapter 5

Shape Carving

Contents

5.1 Introduction

Previous chapters introduced methods, based on interval analysis, able to solve constraint networks that involve real numbers as variables and subsets of $\mathbb{R}^n$ as domains. They have been used to compute the set of feasible values that satisfy a given number of constraints and have been applied to parameter estimation and robot localization.

Now, we will consider the case where variables are subsets of $\mathbb{R}^n$. This typically appears in engineering applications when arbitrary shapes (*i.e.* that cannot be parameterized) are involved or when the model used is uncertain. It corresponds, for instance, to the reconstruction of a three-dimensional object from photos [Bottino et al., 2003], mapping an environment from sonar measurements [Leonard and Durrant-Whyte, 1992] or SLAM [Jaulin, 2011]. For us, the uncertainty will be treated under the form of an interval of sets, called *Thick Set,* or *Interval Shape* in a robotics context. More precisely, in the context of the shape-based SLAM formalism, presented in the introduction of this manuscript, the observation equation takes the following form

$$\mathbb{Y} = \mathbf{f}(\mathbf{p}, \mathbb{X}). \tag{5.1}$$

where $\mathbf{g} : \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{p}$ is an element of $\mathbb{R}^p$, and $\mathbb{X}$ (resp. $\mathbb{Y}$) is a subset of $\mathbb{R}^n$ (resp. $\mathbb{R}^m$). All variables, and in particular $\mathbb{X}$ and $\mathbb{Y}$, of the equation are uncertain, and need to be estimated. This is the goal of this section which aims at providing tools and methods needed to solve the canonical problem, that we called *the shape registration and carving problem,* defined in Equation 5.1.

After recalling some notions on lattices, the first section introduces the concept of *intervals of sets* called *Thick Set*, or shape interval. This new notion will be used as domains for set-valued variables (or shapes). Section 5.3 extends the notion of separators to characterize thick sets. The canonical problem of set inversion is generalized in Section 5.4 for dealing with uncertain sets, and uncertain functions. Finally, Section 5.5 deals with contractors on thick sets and will be used to refine the approximation of an uncertain set defined by constraints.

5.2 Thick Sets

This section recalls more advanced notions on lattices and intervals that will be used to define the thick sets. Behind the provided mathematical background, this section aims to underline the fact that the thick sets theory is mainly an extension of interval analysis. Now intervals are in an infinite dimensional space corresponding to the space of shapes.

5.2.1 Lattices

Lattices. Interval methods can be applied as soon as the set of domains for the variables has a lattice structure [Apt, 1999]. A *lattice* $(\mathcal{E}, \leq)$ is a partially ordered set, closed under least upper and greatest lower bounds [Davey and Priestley, 2002].

The least upper bound of x and y is called the *join* and is denoted by $x \vee y$. The greatest lower bound is called the *meet* and is written as $x \wedge y$. Let us now give three examples.

- The set $(\mathbb{R}^n, \leq)$ is a lattice with respect to the partial order relation given by $\mathbf{x} \leq \mathbf{y} \Leftrightarrow \forall i \in \{1, \ldots, n\}, x_i \leq y_i$. We have $\mathbf{x} \wedge \mathbf{y} = (x_1 \wedge y_1, \ldots, x_n \wedge y_n)$ and $\mathbf{x} \vee \mathbf{y} = (x_1 \vee y_1, \ldots, x_n \vee y_n)$ where $x_i \wedge y_i = \min(x_i, y_i)$ and $x_i \vee y_i = \max(x_i, y_i)$.

- The set $(\mathbb{F}, \leq)$ of the functions that map $\mathbb{R}$ to $\mathbb{R}$ is a lattice with respect to the partial order relation given by $f \leq g \Leftrightarrow \forall t \in \mathbb{R}, f(t) \leq g(t)$. We have $f \wedge g : t \mapsto \min\{f(t), g(t)\}$ and $f \vee g : t \mapsto \max\{f(t), g(t)\}$.

- The set $\mathbb{IR}$ of closed intervals, as introduced by Moore [Moore, 1966], is a complete lattice with respect to the inclusion $\subset$. The meet corresponds to the intersection (generally denoted by $\cap$) and the join corresponds to the interval hull (generally denoted by $\sqcup$). For instance
$$[1,4] \cap [2,\infty] = [2,4] \text{ and } [1,4] \sqcup [8,9] = [1,9]. \tag{5.2}$$

A lattice $\mathcal{E}$ is *complete* if for all (finite or infinite) subsets $\mathcal{A}$ of $\mathcal{E}$, the least upper bound $\wedge\mathcal{A}$ and the greatest lower bound $\vee\mathcal{A}$ belong to $\mathcal{E}$. When a lattice $\mathcal{E}$ is not complete, we can add two elements corresponding to $\wedge\mathcal{A}$ and $\vee\mathcal{A}$ to make it complete. For instance, the set $\mathbb{R}$ is not a complete lattice whereas $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ is. By convention, for the empty set, we set $\wedge\emptyset = \vee\mathcal{E}$ and $\vee\emptyset = \wedge\mathcal{E}$.

Intervals. A *closed interval* (or *interval* for short) $[x]$ of a complete lattice $\mathcal{E}$ is a subset of $\mathcal{E}$ which satisfies $[x] = \{x \in \mathcal{E} \mid \wedge[x] \leq x \leq \vee[x]\}$. Both $\emptyset$ and $\mathcal{E}$ are intervals of $\mathcal{E}$. If we denote by $\mathbb{I}\mathcal{E}$ the set of all intervals of a complete lattice $(\mathcal{E}, \leq)$ then $(\mathbb{I}\mathcal{E}, \subset)$ is also a lattice. For two elements $[x] = [x^-, x^+]$ and $[y] = [y^-, y^+]$ of $\mathbb{I}\mathcal{E}$, we have:
$$\begin{array}{rcl} [x] \wedge [y] & = & [x^- \vee y^-, x^+ \wedge y^+] \\ [x] \vee [y] & = & [x^- \wedge y^-, x^+ \vee y^+]. \end{array} \tag{5.3}$$
The meet $[x] \wedge [y]$ is called the *intersection* and will be denoted by $[x] \cap [y]$. The join $[x] \vee [y]$ is the *interval hull*, denoted by $[x] \sqcup [y]$. It should not be confused with the classical union $\cup$ of two intervals.

Remark 5.1. The bracket notation is here used to denote an interval. The brackets can be interpreted as an operator which associates to an unknown variable x, an interval domain $[x]$ which contains it. This operator is used when solving *Constraint Satisfaction Problems* with intervals [Davis, 1987]. When we define spontaneously an interval $[x]$, then at the same time we define implicitly a variable x enclosed by $[x]$.

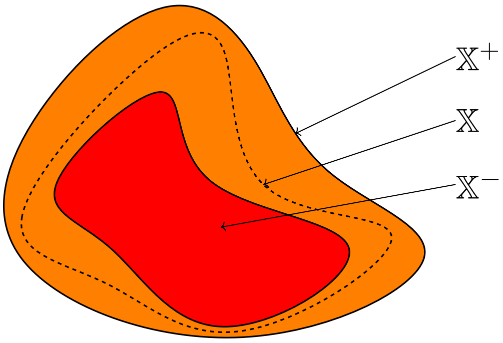


Figure 5.1: Example of thick set. The dark zone is outside the $\mathbb{X}^+$.

5.2.2 Thick set

Denote by $(\mathcal{P}(\mathbb{R}^n), \subset)$ the powerset of $\mathbb{R}^n$ equipped with the inclusion $\subset$ as an order relation. The set $\mathcal{P}(\mathbb{R}^n)$ is a complete lattice with respect to $\subset$. The meet operator corresponds to the intersection and the join to the union. A *thick set* $[\mathbb{X}]$ of $\mathbb{R}^n$ is an interval of $(\mathcal{P}(\mathbb{R}^n), \subset)$. If $[\mathbb{X}]$ is a thick set of $\mathbb{R}^n$, there exist [Yao et al., 2008] two subsets of $\mathbb{R}^n$ called the *lower bound* $\mathbb{X}^-$ and the *upper bound* $\mathbb{X}^+$ such that

$$\begin{aligned} [\mathbb{X}] &= [\mathbb{X}^-, \mathbb{X}^+] \\ &= \{\mathbb{X} \in \mathcal{P}(\mathbb{R}^n) \mid \mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+\}. \end{aligned} \tag{5.4}$$

As illustrated on Figure 5.1, a thick set partitions $\mathbb{R}^n$ into three zones: the *clear zone* $\mathbb{X}^-$, the *penumbra* $\mathbb{X}^+ \backslash \mathbb{X}^-$ and the *dark zone* $\mathbb{R}^n \backslash \mathbb{X}^+$.

Remark 5.2. For simplicity, the clear zone is also denoted by $\mathbb{X}^{in} = \mathbb{X}^-$, the dark zone by $\mathbb{X}^{out} = \overline{\mathbb{X}^+}$ and the penumbra by $\mathbb{X}^? = \mathbb{X}^+ \setminus \mathbb{X}^-$.

Remark 5.3. The convention used to draw sub-paving associated with a thick set is the following: red boxes belong to $\mathbb{X}^-$ (clear zone), orange boxes belong to the penumbra and blue boxes are outside $\mathbb{X}^+$ (dark zone).

A thick set $[\mathbb{X}]$ is a sub-lattice of $(\mathcal{P}(\mathbb{R}^n), \subset)$, *i.e.*, if $\mathbb{A} \in [\mathbb{X}]$, $\mathbb{B} \in [\mathbb{X}]$, then $\mathbb{A} \cap \mathbb{B} \in [\mathbb{X}]$ and $\mathbb{A} \cup \mathbb{B} \in [\mathbb{X}]$. The set of thick sets of $\mathbb{R}^n$ will be denoted by $\mathbb{I}\mathcal{P}(\mathbb{R}^n)$. If for the thick set $[\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+]$, we have $\mathbb{X}^- = \mathbb{X}^+$ then $[\mathbb{X}]$ is said to be *thin*. It corresponds to a singleton in $\mathcal{P}(\mathbb{R}^n)$ or equivalently a classical subset of $\mathbb{R}^n$.

Remark 5.4. Other possibilities to represent vagueness exist in the literature such as *Fuzzy Sets* [Dubois and Prade, 2015] or *Fuzzy Interval* used in fuzzy control [Galichet and Foulloy, 1995, Boukezzoula et al., 2004]. The closest definition, provided by Equation (5.4), is the one used to define the *Rough Set* [Pawlak, 1992]. In the Rough-set model, a given set is represented by a pair of ordinary sets called the lower and upper approximations. The approximation space is constructed based on an equivalence relation defined by a set of attributes. On the contrary, the thick set representation, also called *i-set* in [Jaulin, 2012] or interval-set [Yao et al., 2008, Gervet, 1997], is simply defined as an interval of sets. According to [Yao and Li, 1996], rough-set and thick set models are different extensions of set theory. More precisely with the rough-set theory, the domain of a random variable is described by a membership function which is also uncertain. We can talk about a modelization of the uncertainty of the uncertainty. On the contrary, with the thickset model, the variable is a subset of $\mathbb{R}^n$ and its domain is handled with the classical interval formalism.

The rough-set model introduces two additional set-theoretic operators based on an equivalence relation on the universe while the thick set model extends the standard set-theoretic operators like intersection, union or complement.

5.2.3 Operations

Some operations which can be applied on thick sets are now defined. Since *thick sets* are intervals of sets, two types of operators can be considered: extension of classical set operations to elements of thick sets and operations on intervals of sets.

The first class extends binary operators on subsets of $\mathbb{R}^n$ which correspond to classical operations on vectors of $\mathbb{R}^n$ (in the Minkowsky sense). If $\diamond$ is a binary operator in $\mathbb{R}^n$ (such as $+, -$, the multiplication when $n = 1$, the vector product when $n = 3$), this extension is:

$$[\mathbb{A}] \diamond [\mathbb{B}] = \{\mathbb{A} \diamond \mathbb{B}, \mathbb{A} \in [\mathbb{A}], \mathbb{B} \in [\mathbb{B}]\}. \tag{5.5}$$

The second class is composed of binary operators such as $\diamond \in \{\cup, \cap, \times, \backslash, \dots\}$ where $\times$ is the Cartesian product, $\backslash$ is the restriction (or trim) operator, for subsets of $\mathbb{R}^n$ that do not correspond to any extension of operators in $\mathbb{R}^n$.

Given a collection $\{\mathbb{X}_i\}_{i \in \mathbb{I}}$ of subsets of $\mathbb{R}^n$, we denote by $\square\{\mathbb{X}_{i,} i \in \mathbb{I}\}$ the smallest thick set which encloses all $\mathbb{X}_i, i \in \mathbb{I}$. We have:

$$\square\{\mathbb{X}_{i,} i \in \mathbb{I}\} = \left[\bigcap_{i \in \mathbb{I}} \mathbb{X}_i, \bigcup_{i \in \mathbb{I}} \mathbb{X}_i\right]. \tag{5.6}$$

It is possible to extend the operators to thick sets as follows:

$$[\mathbb{A}] \diamond [\mathbb{B}] = \square\{\mathbb{C}, \exists \mathbb{A} \in [\mathbb{A}], \exists \mathbb{B} \in [\mathbb{B}], \mathbb{C} = \mathbb{A} \diamond \mathbb{B}\}. \tag{5.7}$$

From the monotony of the following operators, we have:

$$
\begin{array}{rcl}
[\mathbb{A}] \cap [\mathbb{B}] & = & \left[\mathbb{A}^- \cap \mathbb{B}^-, \mathbb{A}^+ \cap \mathbb{B}^+\right] \qquad (5.8) \\
[\mathbb{A}] \cup [\mathbb{B}] & = & \left[\mathbb{A}^- \cup \mathbb{B}^-, \mathbb{A}^+ \cup \mathbb{B}^+\right] \qquad (5.9) \\
[\mathbb{A}] \times [\mathbb{B}] & = & \left[\mathbb{A}^- \times \mathbb{B}^-, \mathbb{A}^+ \times \mathbb{B}^+\right] \qquad (5.10) \\
[\mathbb{A}] \setminus [\mathbb{B}] & = & \left[\mathbb{A}^- \setminus \mathbb{B}^+, \mathbb{A}^+ \setminus \mathbb{B}^-\right] \qquad (5.11) \\
[\mathbb{A}] \oplus [\mathbb{B}] & = & \left[\mathbb{A}^- \oplus \mathbb{B}^-, \mathbb{A}^+ \oplus \mathbb{B}^+\right] \qquad (5.12)
\end{array}
$$

If $\mathbf{f}$ is a function from $\mathbb{R}^n$ to $\mathbb{R}^m$ the image of the thick set $[\mathbb{A}] \subset \mathcal{P}(\mathbb{R}^n)$ by $\mathbf{f}$ is defined as:

$$
\mathbf{f}([\mathbb{A}]) = \square\left\{\mathbf{f}(\mathbb{A}), \mathbb{A} \in \left[\mathbb{A}^-, \mathbb{A}^+\right]\right\} = \left[\mathbf{f}(\mathbb{A}^-), \mathbf{f}(\mathbb{A}^+)\right]. \qquad (5.13)
$$

For example, the following illustrates the two types of intersections that can be defined:

$$
\begin{array}{rcl}
[\mathbb{X}] \cap [\mathbb{Y}] & = & \{\mathbb{Z} \in \mathcal{P}(\mathbb{R}^n) \,|\, \mathbb{Z} = \mathbb{X} \cap \mathbb{Y}, \mathbb{X} \in [\mathbb{X}], \mathbb{Y} \in [\mathbb{Y}]\} \\
[\mathbb{X}] \sqcap [\mathbb{Y}] & = & \{\mathbb{Z} \in \mathcal{P}(\mathbb{R}^n) \,|\, \mathbb{Z} \in [\mathbb{X}], \mathbb{Z} \in [\mathbb{Y}]\}.
\end{array} \qquad (5.14)
$$

The first $\cap$ corresponds to the extension to $\mathbb{IP}(\mathbb{R}^n)$ of the intersection in $\mathcal{P}(\mathbb{R}^n)$ whereas the second $\sqcap$ corresponds to the intersection in $\mathbb{I}\mathcal{P}(\mathbb{R}^n)$. Therefore

$$
\begin{array}{rcl}
\mathbb{X} \in [\mathbb{X}], \mathbb{Y} \in [\mathbb{Y}] & \Rightarrow & \mathbb{X} \cap \mathbb{Y} \in [\mathbb{X}] \cap [\mathbb{Y}] \\
\mathbb{Z} \in [\mathbb{X}], \mathbb{Z} \in [\mathbb{Y}] & \Rightarrow & \mathbb{Z} \in [\mathbb{X}] \sqcap [\mathbb{Y}].
\end{array} \qquad (5.15)
$$

As shown in [Jaulin, 2012], we have:

$$
\begin{array}{rcl}
[\mathbb{X}] \cap [\mathbb{Y}] & = & [\mathbb{X}^- \cap \mathbb{Y}^-, \mathbb{X}^+ \cap \mathbb{Y}^+] \\
[\mathbb{X}] \sqcap [\mathbb{Y}] & = & [\mathbb{X}^- \cup \mathbb{Y}^-, \mathbb{X}^+ \cap \mathbb{Y}^+].
\end{array} \qquad (5.16)
$$

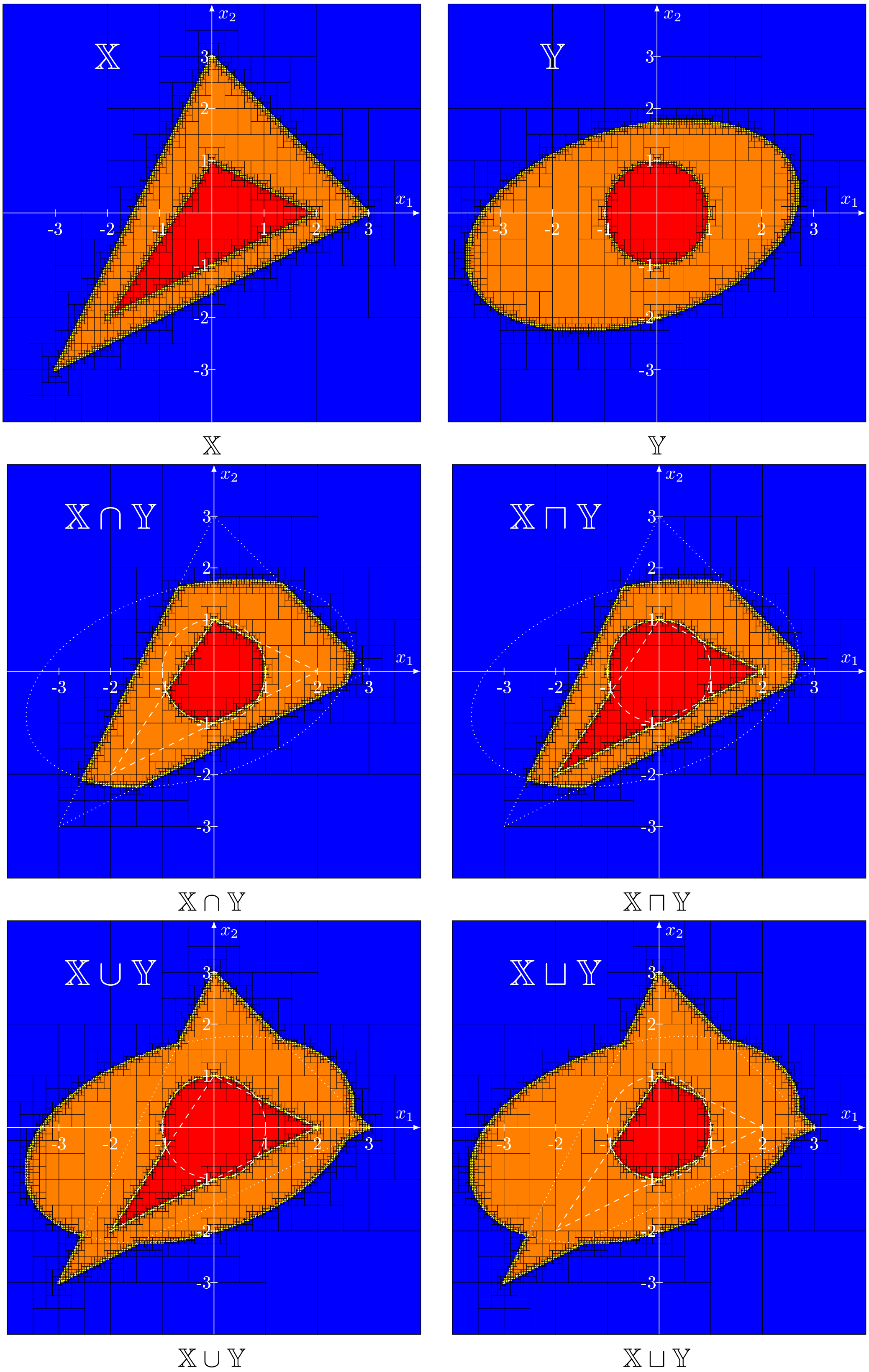
The same type of relations applies to the union

$$
\begin{array}{rcl}
[\mathbb{X}] \sqcup [\mathbb{Y}] & = & [\mathbb{X}^- \cap \mathbb{Y}^-, \mathbb{X}^+ \cup \mathbb{Y}^+] \\
[\mathbb{X}] \cup [\mathbb{Y}] & = & [\mathbb{X}^- \cup \mathbb{Y}^-, \mathbb{X}^+ \cup \mathbb{Y}^+]
\end{array} \qquad (5.17)
$$

Comparisons between $\cap$,$\sqcap$,$\cup$ and $\sqcup$ are depicted in Figure 5.2.

5.3 Thick Separators

To characterize a thin set using a paver, we may use a separator inside a paver. As for operations on sets, separators can be immediately generalized to thick sets.



$\mathbb{X}$

$\mathbb{Y}$

$\mathbb{X} \cap \mathbb{Y}$

$\mathbb{X} \sqcap \mathbb{Y}$

$\mathbb{X} \cup \mathbb{Y}$

$\mathbb{X} \sqcup \mathbb{Y}$

Figure 5.2: Illustration of the two different types of operators on thick sets.

Now, the penumbra as a non-zero volume and, for efficiency reasons, it is important to avoid any accumulation of the paving deep inside the penumbra.

5.3.1 Definition

A *thick separator* $[\mathcal{S}]$ for the thick set $[\mathbb{X}]$ is an extension of the concept of separator to thick sets. As illustrated by Figure 5.3, a thick separator is a 3-uple of contractors $\left\{\mathcal{S}^{in}, \mathcal{S}^{?}, \mathcal{S}^{out}\right\}$ such that, for all $[\mathbf{x}] \in \mathbb{IR}^n$

$$\begin{array}{r} \mathcal{S}^{in}([\mathbf{x}]) \cap \mathbb{X}^{in} = [\mathbf{x}] \cap \mathbb{X}^{in} \\ \mathcal{S}^{?}([\mathbf{x}]) \cap \mathbb{X}^{?} = [\mathbf{x}] \cap \mathbb{X}^{?} \\ \mathcal{S}^{out}([\mathbf{x}]) \cap \mathbb{X}^{out} = [\mathbf{x}] \cap \mathbb{X}^{out} \end{array} \tag{5.18}$$

As for separators, which are built in practice from a pair of two complementary contractors, thick separators can be built from a pair of classical separators. Given separators $\mathcal{S}^- = \left\{\mathcal{S}^-_{in}, \mathcal{S}^-_{out}\right\}$ consistent with $\mathbb{X}^-$ and $\mathcal{S}^+ = \left\{\mathcal{S}^+_{in}, \mathcal{S}^+_{out}\right\}$ consistent with $\mathbb{X}^+$, the thick separator $[\mathcal{S}]$ consistent with $[\mathbb{X}]$ is:

$$[\mathcal{S}] = \left\{\mathcal{S}^-_{out}, \mathcal{S}^-_{in} \cup \mathcal{S}^+_{out}, \mathcal{S}^+_{in}\right\} \tag{5.19}$$

Proof. This is a direct consequence of definitions of $\mathcal{S}^-$, and $\mathcal{S}^+$. We have

$$\mathcal{S}^-_{out} \sim \mathbb{X}^- = \mathbb{X}^{in} \tag{5.20}$$
$$\mathcal{S}^+_{in} \sim \overline{\mathbb{X}^+} = \mathbb{X}^{out} \tag{5.21}$$
$$\mathcal{S}^-_{in} \cup \mathcal{S}^+_{out} \sim \overline{\mathbb{X}^-} \cap \mathbb{X}^+ = \mathbb{X}^{?} \tag{5.22}$$

□

The following example illustrates how a thick set can be defined from a grayscale image and characterized using a thick separator.

Example 5.1. We consider the problem of characterizing a thick set $[\mathbb{X}]$ defined from the ternary images depicted in Figure 5.4a. We assume that white pixels belong to $\mathbb{X}^-$, white and gray pixels belong to $\mathbb{X}^+$ and black pixels do not belong to $[\mathbb{X}]$. Using the image separator introduced in Section 2.5.3 on page 44, it is possible to build a thick separator $[\mathcal{S}]$ for $[\mathbb{X}]$. The result of the characterization of $[\mathbb{X}]$ is depicted in Figure 5.4b.

5.3.2 Algebra

Algebra for thick separators is defined in the same manner than what has been done for contractors [Chabert and Jaulin, 2009b] or for separators [Jaulin and Desrochers, 2014]. The main motivation is to provide methods to compute with

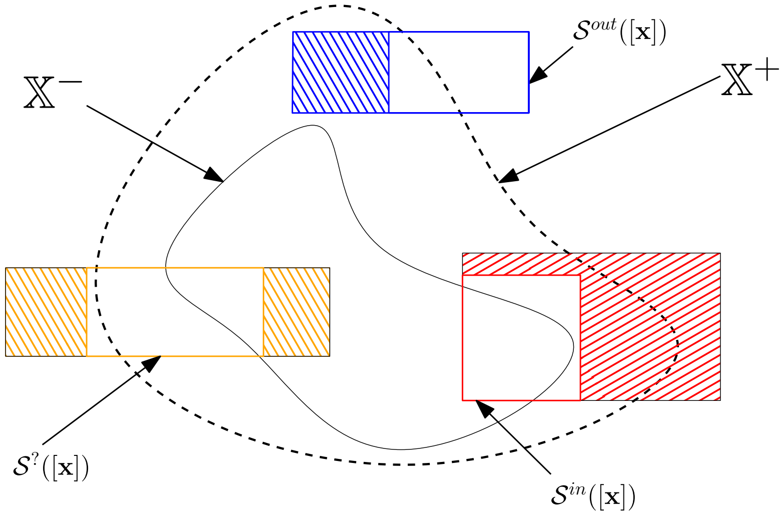
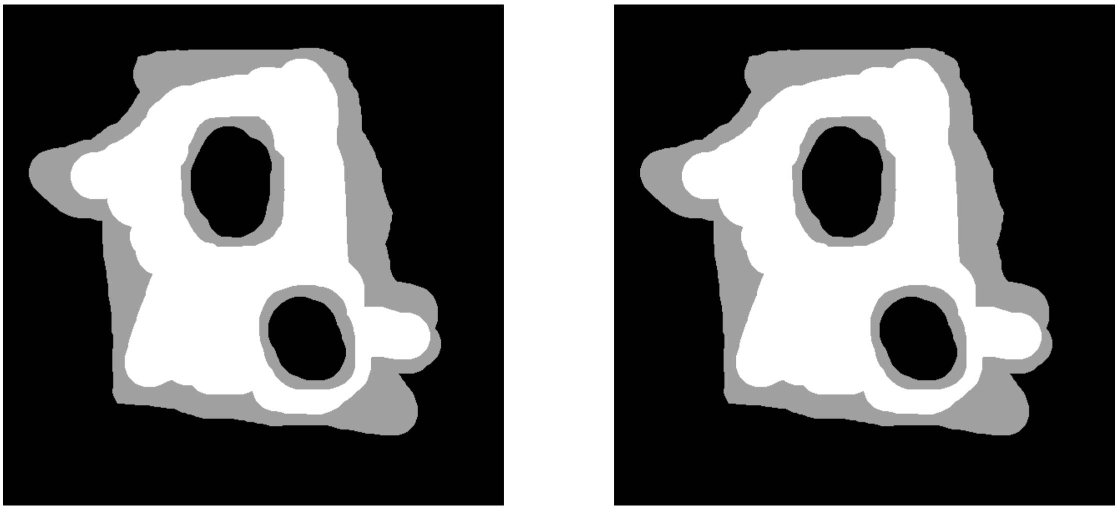

$\mathcal{S}^{out}([\mathbf{x}])$
$\mathbb{X}^-$
$\mathbb{X}^+$
$\mathcal{S}^?([\mathbf{x}])$
$\mathcal{S}^{in}([\mathbf{x}])$

Figure 5.3: Thick separator. $\mathcal{S}^{out}$ is consistent with $\mathbb{X}^+$, $\mathcal{S}^{in}$ with $\mathbb{X}^-$ and $\mathcal{S}^?$ with $\mathbb{X}^? = \mathbb{X}^+ \setminus \mathbb{X}^-$.



(a) Gray-scale image

(b) Resulting subpaving

Figure 5.4: Characterization of $[\mathbb{X}]$ using a paver defined by a grayscale image.

thick sets. We only focus on the intersection of thick separators but a similar reasoning can be done to define other operations, introduced in Section 5.1, such as the union, restriction, squared intersection, exclusive-or, or relaxed intersection [Desrochers and Jaulin, 2016b].

Intersection

Consider two thick sets $[\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+]$ and $[\mathbb{Y}] = [\mathbb{Y}^-, \mathbb{Y}^+]$ with their associated thick separators $[\mathcal{S}_\mathbb{X}] = \left\{\mathcal{S}_\mathbb{X}^{in}, \mathcal{S}_\mathbb{X}^{?}, \mathcal{S}_\mathbb{X}^{out}\right\}$ and $[\mathcal{S}_\mathbb{Y}] = \left\{\mathcal{S}_\mathbb{Y}^{in}, \mathcal{S}_\mathbb{Y}^{?}, \mathcal{S}_\mathbb{Y}^{out}\right\}$. A thick separator for the thick set

$$[\mathbb{Z}] = \left[\mathbb{Z}^-, \mathbb{Z}^+\right] = [\mathbb{X}] \cap [\mathbb{Y}] = \left[\mathbb{X}^- \cap \mathbb{Y}^-, \mathbb{X}^+ \cap \mathbb{Y}^+\right]. \tag{5.23}$$

is

$$\begin{array}{rcc} [\mathcal{S}_\mathbb{Z}] & = & \left\{\mathcal{S}_\mathbb{Z}^{in}, \mathcal{S}_\mathbb{Z}^{?}, \mathcal{S}_\mathbb{Z}^{out}\right\} \\ & = & \left\{\mathcal{S}_\mathbb{X}^{in} \cap \mathcal{S}_\mathbb{Y}^{in}, \left(\mathcal{S}_\mathbb{X}^{?} \cap \mathcal{S}_\mathbb{Y}^{in}\right) \sqcup \left(\mathcal{S}_\mathbb{X}^{?} \cap \mathcal{S}_\mathbb{Y}^{?}\right) \sqcup \left(\mathcal{S}_\mathbb{X}^{in} \cap \mathcal{S}_\mathbb{Y}^{?}\right), \mathcal{S}_\mathbb{X}^{out} \sqcup \mathcal{S}_\mathbb{Y}^{out}\right\}. \end{array} \tag{5.24}$$

Proof. We have

$$\begin{array}{ccccc} \mathbb{Z}^{in} & = & \mathbb{Z}^- & = & \mathbb{X}^- \cap \mathbb{Y}^- \\ & & & = & \mathbb{X}^{in} \cap \mathbb{Y}^{in} \\ \mathbb{Z}^{?} & = & \mathbb{Z}^+ \backslash \mathbb{Z}^- & = & \mathbb{X}^+ \cap \mathbb{Y}^+ \backslash (\mathbb{X}^- \cap \mathbb{Y}^-) \\ & & & = & \mathbb{X}^+ \cap \mathbb{Y}^+ \cap \overline{\mathbb{X}^- \cap \mathbb{Y}^-} \\ & & & = & \mathbb{X}^+ \cap \mathbb{Y}^+ \cap \left(\overline{\mathbb{X}^-} \cup \overline{\mathbb{Y}^-}\right) \\ & & & = & \left(\mathbb{X}^{in} \cup \mathbb{X}^{?}\right) \cap \left(\mathbb{Y}^{in} \cup \mathbb{Y}^{?}\right) \cap \left(\left(\mathbb{X}^{out} \cup \mathbb{X}^{?}\right) \cup \left(\mathbb{Y}^{out} \cup \mathbb{Y}^{?}\right)\right) \\ & & & = & \left(\mathbb{X}^{?} \cap \mathbb{Y}^{in}\right) \cup \left(\mathbb{X}^{?} \cap \mathbb{Y}^{?}\right) \cup \left(\mathbb{X}^{in} \cap \mathbb{Y}^{?}\right) \\ \mathbb{Z}^{out} & = & \overline{\mathbb{Z}^+} & = & \overline{\mathbb{X}^+ \cap \mathbb{Y}^+} \\ & & & = & \overline{\mathbb{X}^+} \cup \overline{\mathbb{Y}^+} \\ & & & = & \mathbb{X}^{out} \cup \mathbb{Y}^{out}. \end{array}$$

From the separator algebra, we get that a contractor for $\mathbb{Z}^{in}$ is $\mathcal{S}_\mathbb{Z}^{in} = \mathcal{S}_\mathbb{X}^{in} \cap \mathcal{S}_\mathbb{Y}^{in}$, a contractor for $\mathbb{Z}^{out}$ is $\mathcal{S}_\mathbb{Z}^{out} = \mathcal{S}_\mathbb{X}^{out} \sqcup \mathcal{S}_\mathbb{Y}^{out}$ and a contractor for $\mathbb{Z}^{?}$ is

$$\mathcal{S}_\mathbb{Z}^{?} = \left(\mathcal{S}_\mathbb{X}^{?} \cap \mathcal{S}_\mathbb{Y}^{in}\right) \sqcup \left(\mathcal{S}_\mathbb{X}^{?} \cap \mathcal{S}_\mathbb{Y}^{?}\right) \sqcup \left(\mathcal{S}_\mathbb{X}^{in} \cap \mathcal{S}_\mathbb{Y}^{?}\right). \tag{5.25}$$

□

Remark 5.5. In the case where $[\mathcal{S}_\mathbb{X}] = \left\{\mathcal{S}_\mathbb{X}^-, \mathcal{S}_\mathbb{X}^+\right\}$ and $[\mathcal{S}_\mathbb{Y}] = \left\{\mathcal{S}_\mathbb{Y}^-, \mathcal{S}_\mathbb{Y}^+\right\}$, the separator consistent with Equation (5.2) is

$$[\mathcal{S}_\mathbb{Z}] = \left\{\mathcal{S}_\mathbb{X}^- \cap \mathcal{S}_\mathbb{Y}^-, \mathcal{S}_\mathbb{X}^+ \cap \mathcal{S}_\mathbb{Y}^+\right\}. \tag{5.26}$$

The proof is a direct consequence of Equation (5.9). We have

$$\mathcal{S}_{\mathbb{X}}^{-} \cap \mathcal{S}_{\mathbb{Y}}^{-} \sim \mathbb{X}^{-} \cap \mathbb{Y}^{-} \text{ and } \mathcal{S}_{\mathbb{X}}^{+} \cap \mathcal{S}_{\mathbb{Y}}^{+} \sim \mathbb{X}^{+} \cap \mathbb{Y}^{+}.$$

5.4 Uncertain Set Inversion

The canonical problem of the set inversion $\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$ is well formalized and many algorithms, such as the ones presented in Chapters 2 and 3, are commonly used to solve it. In this section, we extend this problem to the thick set case where both $\mathbb{Y}$ and $\mathbf{f}$ are uncertain. A first theorem is given in order to provide a generic formulation of the uncertain set inversion problem.

However, in the general case, this problem is hard to solve. In order to provide explicit methods for solving this problem, assumptions on the nature of $\mathbf{f}$ have to be made. When $\mathbf{f}$ depends on an uncertain parameter vector $\mathbf{p} \in \mathbb{R}^p$, a method for dealing with the set inversion problem is proposed in this section. It allows handling uncertain nonlinear transformations such as rotations of an angle which belongs to a given interval. The introduced method mainly relies on the projection of separator, presented in Section 4.2.1, which can be slow when dealing with a high-dimensional parameter space. On the contrary, when $\mathbf{f}$ is assumed to belong to an interval of functions, a dedicated and more efficient method is introduced in Appendix A. Note also that this appendix, published in [Desrochers and Jaulin, 2017d], contains additional examples which have motivated the need for developing specific algorithms.

5.4.1 Problem statement

Given a function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and a set $\mathbb{Y} \subset \mathbb{R}^m$, set inversion aims at bracketing from inside and outside the set

$$\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y}). \tag{5.27}$$

This formalism has been used for more than 20 years with interval methods to solve problems in bounded-error parameter estimation [Vehi et al., 1997], robot localization [Meizel et al., 2002, Colle and Galerne, 2013, Drevelle and Bonnifait, 2013] and robust control [Malti et al., 2011, Herrero et al., 2005, Jaulin and Walter, 1993a]. Most interval algorithms for set-inversion alternate some interval tests or contractions [Chabert and Jaulin, 2009b] to certify that a box (*i.e.*, a Cartesian product of intervals) is inside or outside the solution set $\mathbb{X}$ and bisect the boxes for which no conclusion can be reached.

In the case where both $\mathbf{f}$ and $\mathbb{Y}$ are uncertain, a relaxation of the resulting uncertain constraints can be performed by adding quantifiers as made in [Goldsztejn,

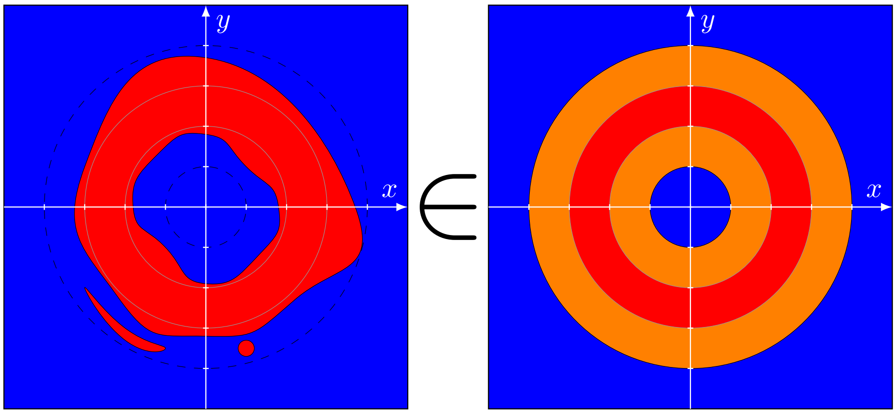


Figure 5.5: The thin set $\mathbb{X}$ (which is not a ring and even not connected) drawn in red on the left belongs to the thick set $[\mathbb{X}]$ on the right. $[\mathbb{X}]$ contains all sets that enclose the red ring and which do not intersect the blue zone.

2006, Kreinovich and Shary, 2016] for the linear case or by allowing a given number of constraints to be unsatisfied [Carbonnel et al., 2014]. A set inversion problem can be written as

$$\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y}),\ \mathbf{f} \in \mathbb{F} \text{ and } \mathbb{Y} \in [\mathbb{Y}]\,. \tag{5.28}$$

where $[\mathbb{Y}]$ is a thick set and $\mathbb{F}$ is a set of functions, *i.e,* a subset of $\mathcal{F}(\mathbb{R}^n, \mathbb{R}^m)$. For instance, $\mathbb{F}$ can be the set of all periodic functions, the set of positive functions, or a tube seen as an interval of functions from $\mathbb{R}$ to $\mathbb{R}^m$.

The set $\mathbb{X}$ is said to be a *feasible solution* if

$$\exists \mathbf{f} \in \mathbb{F}, \exists \mathbb{Y} \in [\mathbb{Y}]\,,\ \mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y}). \tag{5.29}$$

The set of all feasible solutions is not a thick set in general as illustrated by the following example. Solving a thick set-inversion problem will amount to finding the smallest thick set which encloses all feasible solutions for Equation (5.28).

Example 5.2. Let $\mathbf{f}(\mathbf{x}) = x_1^2 + x_2^2$ be a thin function from $\mathbb{R}^2$ to $\mathbb{R}$, and $\mathbb{Y}$ be a set such that $[2,3] \subset \mathbb{Y} \subset [1,4]$. If $\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$, we have:

$$\mathbf{f}^{-1}\left([2,3]\right) \subset \mathbb{X} \subset \mathbf{f}^{-1}\left([1,4]\right). \tag{5.30}$$

Now, all feasible $\mathbb{X}$ correspond to centered rings and it is clear that the inclusion (see Equation (5.30)) contains other types of sets as illustrated by Figure 5.5.

5.4.2 Set inversion theorem

The set inversion problem which involves an uncertain set and an uncertain function $\mathbf{f}$ is solved using the following theorem:

Theorem 5.1. *Given* $\mathbb{F} \subset \mathcal{F}(\mathbb{R}^n, \mathbb{R}^m)$ *and the thick set* $[\mathbb{Y}] = [\mathbb{Y}^-, \mathbb{Y}^+]$*, the smallest thick set which encloses all sets* $\mathbb{X}$ *such that*

$$\exists \mathbf{f} \in \mathbb{F},\, \exists \mathbb{Y} \in [\mathbb{Y}] \mid \mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y}). \tag{5.31}$$

is the thick set $[\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+]$ *where*

$$\begin{array}{rcl} \mathbb{X}^- & = & \bigcap\limits_{\mathbf{f}\in\mathbb{F}} \mathbf{f}^{-1}(\mathbb{Y}^-), \\ \mathbb{X}^+ & = & \bigcup\limits_{\mathbf{f}\in\mathbb{F}} \mathbf{f}^{-1}(\mathbb{Y}^+). \end{array} \tag{5.32}$$

Proof. Denote by $\{\mathbb{X}_i\}_{i\in\mathbb{I}}$ the set of all solutions of Equation (5.31). The smallest thick set $[\mathbb{X}]$ containing $\{\mathbb{X}_i\}_{i\in\mathbb{I}}$ is the thick set

$$[\mathbb{X}] = \left[\bigcap_{i\in\mathbb{I}}\{\mathbb{X}_i\}, \bigcup_{i\in\mathbb{I}}\{\mathbb{X}_i\}\right]. \tag{5.33}$$

Now, for each $\mathbb{X}_i \in \{\mathbb{X}_i\}_{i\in\mathbb{I}}$,

$$\exists \mathbf{f}_i \in \mathbb{F},\, \exists \mathbb{Y}_i \in [\mathbb{Y}] \mid \mathbb{X}_i = \mathbf{f}_i^{-1}(\mathbb{Y}_i). \tag{5.34}$$

Thus, $[\mathbb{X}]$ is given by

$$\left[\bigcap_{\mathbf{f}\in\mathbb{F}} \bigcap_{\mathbb{Y}\in[\mathbb{Y}]} \mathbf{f}^{-1}(\mathbb{Y}), \bigcup_{\mathbf{f}\in\mathbb{F}} \bigcup_{\mathbb{Y}\in[\mathbb{Y}]} \mathbf{f}^{-1}(\mathbb{Y})\right]. \tag{5.35}$$

Now, since $\mathbf{f}^{-1}$ is monotonic with respect to the inclusion $\subset$, we get

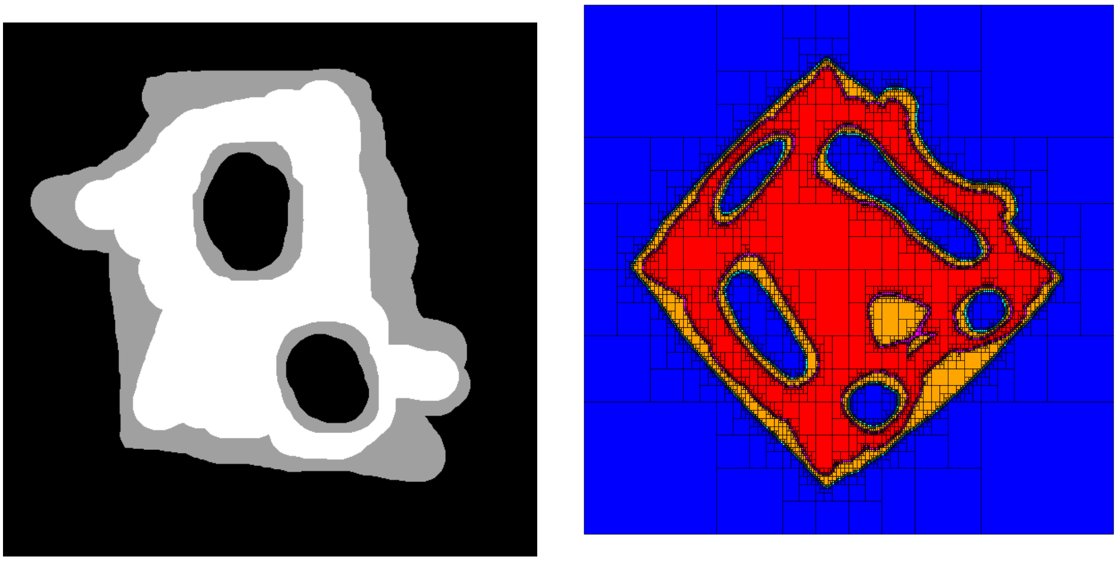
$$[\mathbb{X}] = \left[\bigcap_{\mathbf{f}\in\mathbb{F}} \mathbf{f}^{-1}(\mathbb{Y}^-), \bigcup_{\mathbf{f}\in\mathbb{F}} \mathbf{f}^{-1}(\mathbb{Y}^+)\right]. \tag{5.36}$$

□

Remark 5.6. Theorem 5.1 provides the exact formulation of the *thick set inversion* problem and defines the corresponding two sets we want to compute, *i.e.*, the two sets $\mathbb{X}^-$ and $\mathbb{X}^+$ as defined by Equation (5.32). The main difficulty is to get an inner approximation of the penumbra $\mathbb{X}^+ \backslash \mathbb{X}^-$.

Existing interval methods can still be used to deal with this type of uncertainty, but they accumulate inside the *penumbra*. This accumulation makes these methods inefficient, since they spend most of the computation time to test tiny boxes that are deep inside the penumbra, without proving that they are effectively in the penumbra. Consequently, specific algorithms have to be developed.

As it is illustrated by the next example, when $\mathbb{F}$ contains a single function $\mathbf{f}$ from $\mathbb{R}^n$ to $\mathbb{R}^m$, from Equation (5.13), the smallest thick set solution is



(a) $[\mathbb{Y}]$ as an image

(b) $[\mathbb{X}] = \mathbf{f}^{-1}([\mathbb{Y}])$

Figure 5.6: Left: $[\mathbb{Y}]$. Right: $[\mathbb{X}] = \mathbf{f}^{-1}([\mathbb{Y}])$.

$$[\mathbb{X}] = \left[\mathbf{f}^{-1}(\mathbb{Y}^-), \mathbf{f}^{-1}(\mathbb{Y}^+)\right], \tag{5.37}$$

where classical set inversion algorithms can be used.

Example 5.3. Consider $[\mathbb{Y}]$ as depicted in Figure 5.6, and the function $\mathbf{f} : \mathbb{R}^2 \to \mathbb{R}^2$ defined by:

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} x_1 \sin(x_1 + x_2) \\ x_2 \cos(x_2 - x_1) \end{pmatrix}. \tag{5.38}$$

The result of the set inversion $[\mathbb{X}] = \mathbf{f}^{-1}([\mathbb{Y}])$ is given by Figure 5.6. The thick separator, used by the paver, is built from a pair of separators consistent with the two classical set inversion problems: $\mathbf{f}^{-1}(\mathbb{Y}^-)$ and $\mathbf{f}^{-1}(\mathbb{Y}^+)$.

When $\mathbb{F}$ is a set of functions that depend on a parameter, elimination of quantified parameters can be done using projection algorithms (see Section 4.2.1 on page 81). This case is discussed in the next section. However, it requires additional bisections along the parameter space that become costly when dealing with high dimensional parameters.

On the contrary, when $\mathbb{F}$ can be described by an interval of functions, called a *Thick Function*, a new algorithm has been developed to compute the solution set without bisecting inside the parameter space. This method is discussed in Appendix A.

5.4.3 Set inversion with parametric function.

We consider here the case where $\mathbf{f}$ depends on a parameter vector $\mathbf{p} \in [\mathbf{p}] \subset \mathbb{R}^p$. The function should thus be written as $\mathbf{f}(\mathbf{x}, \mathbf{p})$, but for simplicity of notation, we will often write $\mathbf{f}_{\mathbf{p}}(\mathbf{x})$. For simplicity, the set inversion problem will be written as:

$$[\mathbb{X}] = \mathbf{f}_{[\mathbf{p}]}([\mathbb{Y}]) \tag{5.39}$$

From Theorem 5.1, the bounds of the smallest thick set $[\mathbb{X}^-, \mathbb{X}^+]$ are defined by:

$$\begin{aligned} \mathbb{X}^- &= \bigcap_{\mathbf{p} \in [\mathbf{p}]} \mathbf{f}_{\mathbf{p}}^{-1}(\mathbb{Y}^-) = \left\{ \mathbf{x} | \forall \mathbf{p} \in [\mathbf{p}], \mathbf{f}_{\mathbf{p}}(\mathbf{x}) \in \mathbb{Y}^- \right\} & (5.40) \\ \mathbb{X}^+ &= \bigcup_{\mathbf{p} \in [\mathbf{p}]} \mathbf{f}_{\mathbf{p}}^{-1}(\mathbb{Y}^+) = \left\{ \mathbf{x} | \exists \mathbf{p} \in [\mathbf{p}], \mathbf{f}_{\mathbf{p}}(\mathbf{x}) \in \mathbb{Y}^+ \right\}. & (5.41) \end{aligned}$$

Using the notion of projection introduced in Section 4.2, it is possible to build separators $\mathcal{S}^-$, consistent with $\mathbb{X}^-$, and $\mathcal{S}^+$ consistent with $\mathbb{X}^+$. Then the thick separator consistent with $[\mathbb{X}]$ can be defined and used inside a paver to characterize the solution set.

The following example illustrates the set inversion with the uncertain parametric rotation of a box of $\mathbb{R}^2$. It compares a sampling method used to guess the shape of the solution set, the classical set inversion algorithm which accumulates in the penumbra and the proposed set inversion using separators.

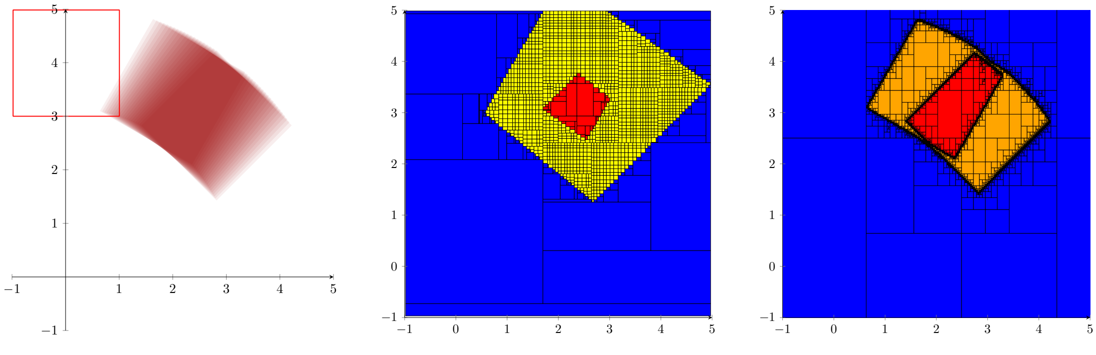
Example 5.4. Let $\mathbf{R}_\theta$ be a rotation of an angle $\theta \in [\frac{\pi}{6}, \frac{\pi}{3}]$ and $[\mathbf{y}] = [0, 3] \times [4, 5]$. The solution of the set inversion problem:

$$[\mathbb{X}] = \mathbf{R}_{[\theta]}([\mathbf{y}]) \tag{5.42}$$

is depicted in Figure 5.7 with and without the use of projections. The classical interval methods, on Figure 5.7b, accumulate in the penumbra and add a lot of pessimism. Both the red and blue areas are smaller than the true one, drawn in Figure 5.7c.

5.5 Shape Registration and Carving

Previous sections have defined the notions of thick sets and algorithms for solving the uncertain set inversion problem. As defined in the introduction, when a set is used as a variable of our constraint network we call it a *shape*. It belongs to a domain that we called an *interval shape*. In this section, the problem of the shape registration, presented in Chapter 4, is extended to the case where the shapes are not thin anymore and are uncertain. This problem is defined by the following



(a) sampling method (b) classical set inversion (c) projection algorithm

Figure 5.7: The result of the set inversion using a sampling method along the parameter space is depicted in Figure 5.7a. For each $\theta \in [\theta]$, the image of the red box by the rotation of an angle $-\theta$ is drawn using a transparency effect. It makes it possible to guess sets $\mathbb{X}^-$, $\mathbb{X}^+$. Figure 5.7b: the result with classical methods where the penumbra is not characterized. Figure 5.7c: the result of the set inversion using projection algorithms.

shape-valued constraint network :

$$\begin{cases} \mathbf{f}(\mathbb{A}) = \mathbb{B} \\ \mathbb{A} \in [\mathbb{A}] \\ \mathbb{B} \in [\mathbb{B}] \\ \mathbf{f} \in \mathbb{F} \end{cases} \tag{5.43}$$

Two sub-problems have to be considered in order to contract all domains of the involved variables. The *registration* problem which aims at finding the smallest set of functions, included in $\mathbb{F}$, compatible with the constraint $\mathbf{f}(\mathbb{A}) = \mathbb{B}$, $\mathbb{A} \in [\mathbb{A}]$ and $\mathbb{B} \in [\mathbb{B}]$. And, the *carving* problem which intends at refining the thick set approximation of $[\mathbb{A}]$ and $[\mathbb{B}]$ for a given set of functions $\mathbb{F}$.

For that purpose, the notion of contractor on thick sets is introduced in this section with some elementary contractors dedicated to Equation 5.43. Then the shape registration problem, presented in Section 4.2.2, will be extended to the thick set case. Finally, this section ends with an example which illustrates the method.

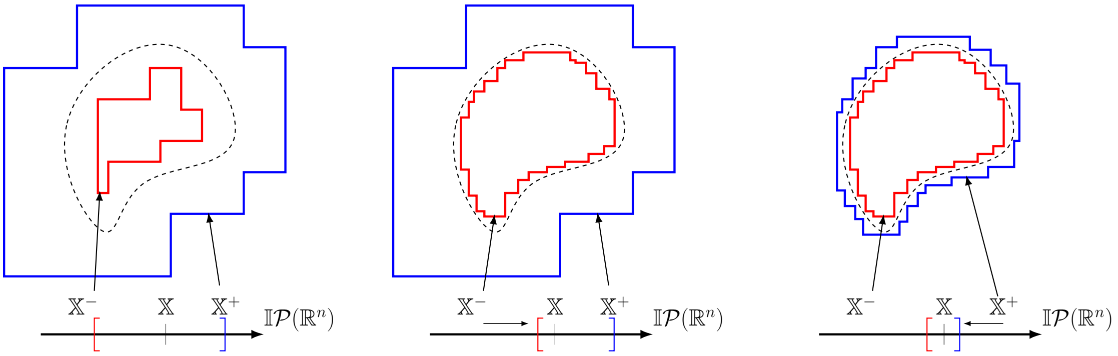
5.5.1 Contractor on thick sets

Since *Thick Sets* are intervals of sets, it is possible to define contractors for them in the same way as introduced in Section 2.4.1.1.

A contractor on a thick set is an operator from $\mathbb{IP}(\mathbb{R}^n)$ to $\mathbb{IP}(\mathbb{R}^n)$ such as :

$$\begin{array}{rcll} \mathcal{C}([\mathbb{X}]) & \subset & [\mathbb{X}] & \text{(contractance)} \\ [\mathbb{X}] \subset [\mathbb{Y}] & \Rightarrow & \mathcal{C}([\mathbb{X}]) \subset \mathcal{C}([\mathbb{Y}]) & \text{(monotonicity)} \end{array} \tag{5.44}$$

The effect of a contraction on a thick set is illustrated on Figure 5.8. A contraction of the lower bound corresponds to an inflation of the under approximation (Fig. 5.8c). Similarly, decreasing the upper bound of $[\mathbb{X}]$ contracts the upper approximation (Fig.5.8b). Since thick sets are interval of $\mathcal{P}(\mathbb{R}^n)$, it is only possible to move its bounds and have an outer contraction.



(a) initial domain

(b) lower contraction

(c) upper contraction

Figure 5.8: Illustration of contractor applied on a thick set. Recall that a thick set is an interval of $\mathcal{P}(\mathbb{R}^n)$. The thin set $\mathbb{X}$, delimited by the dashed line, is a singleton in this space.

5.5.2 Shape carving

The specific primitive contractors on thick sets are now presented. They will allow to contract shape domains of Equation (5.43).

Constraint $\mathbb{A} \subset \mathbb{B}$

Since the equality between two sets can be decomposed into two inclusion constraints, we consider the following proposition:

Proposition 5.1. Given $\mathbb{A} \in [\mathbb{A}]$, $\mathbb{B} \in [\mathbb{B}]$, the minimal contractor associated with the constraint $\mathbb{A} \subset \mathbb{B}$ is:

$$\mathcal{C}_{\subset}\begin{pmatrix} [\mathbb{A}] \\ [\mathbb{B}] \end{pmatrix} = \begin{pmatrix} [\mathbb{A}] \sqcap ([\mathbb{B}] \setminus ([\emptyset, \mathbb{R}^n])) \\ [\mathbb{B}] \sqcap ([\mathbb{A}] \cup ([\emptyset, \mathbb{R}^n])) \end{pmatrix} \tag{5.45}$$

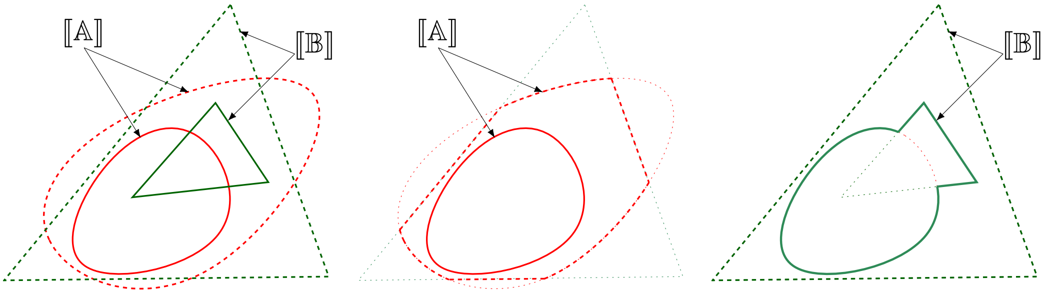
or equivalently:

$$\mathcal{C}_{\subset}\begin{pmatrix}[\mathbb{A}]\\ [\mathbb{B}]\end{pmatrix}=\begin{pmatrix}[\mathbb{A}^-,\mathbb{A}^+\cap\mathbb{B}^+]\\ [\mathbb{B}^-\cup\mathbb{A}^-,\mathbb{B}^+]\end{pmatrix}\tag{5.46}$$

Proof. The proof can be found in [Jaulin, 2012].

□

This contractor is illustrated in Figure 5.9.



(a) $[\mathbb{A}]$ and $[\mathbb{B}]$ (b) $[\mathbb{A}]\sqcap([\mathbb{B}]\setminus[\emptyset,\mathbb{R}^n])$ (c) $[\mathbb{B}]\sqcap([\mathbb{A}]\cup[\emptyset,\mathbb{R}^n])$

Figure 5.9: Contraction with respect to the constraint $\mathbb{A}\subset\mathbb{B}$.

5.5.2.1 Constraint $f(\mathbb{A})=\mathbb{B}$

We have the following theorem:

Proposition 5.2. Given $\mathbb{A}\in[\mathbb{A}]$, $\mathbb{B}\in[\mathbb{B}]$, the minimal contractor associated with the constraint $f(\mathbb{A})=\mathbb{B}$ where $\mathbf{f}$ is bijective is :

$$\mathcal{C}_{\mathbf{f}}\begin{pmatrix}[\mathbb{A}]\\ [\mathbb{B}]\end{pmatrix}=\begin{pmatrix}[\mathbb{A}]\sqcap\mathbf{f}^{-1}([\mathbb{B}])\\ [\mathbb{B}]\sqcap\mathbf{f}([\mathbb{A}])\end{pmatrix}=\begin{pmatrix}[\mathbb{A}^-\cup\mathbf{f}^{-1}(\mathbb{B}^-),\mathbb{A}^+\cap\mathbf{f}^{-1}(\mathbb{B}^+)]\\ [\mathbb{B}^-\cup\mathbf{f}(\mathbb{A}^-),\mathbb{B}^+\cap\mathbf{f}(\mathbb{A}^+)]\end{pmatrix}\tag{5.47}$$

Proof. The proof can be found in [Jaulin, 2012].

□

Finally, using the thick set inversion algorithm, introduced in Section 5.4 , the previous contractor can be extended to the case where $\mathbf{f}$ is uncertain, *i.e.*, $\mathbf{f}\in\mathbb{F}\subset\mathcal{F}(\mathbb{R}^n,\mathbb{R}^m)$.

Proposition 5.3. Given $\mathbb{A}\in[\mathbb{A}]$, $\mathbb{B}\in[\mathbb{B}]$, $\mathbf{f}\in\mathbb{F}$ the minimal contractor associated with the constraint $\mathbf{f}(\mathbb{A})\subset\mathbb{B}$ is :

$$\mathcal{C}_{\mathbb{F}}\begin{pmatrix}[\mathbb{A}]\\ [\mathbb{B}]\end{pmatrix}=\begin{pmatrix}[\mathbb{A}]\sqcap\mathbb{F}^{-1}([\mathbb{B}])\\ [\mathbb{B}]\sqcap\mathbb{F}([\mathbb{A}])\end{pmatrix}\tag{5.48}$$

where $\mathbb{F}^{-1}([\mathbb{B}])$ is the result of the set inversion as formalized in Section 5.4.

Proof. The proof is a direct consequence of Theorem 5.1 and Proposition 5.1.

□

In this section, only contractors needed to define the shape carving problem have been considered. Other contractors, such as those associated to the constraint $\mathbb{A} \cap \mathbb{B} = \emptyset$, can be found in [Jaulin, 2012], but are not needed for solving the Shape-based SLAM presented in the next chapter.

5.5.3 Shape registration and carving

The shape registration problem, introduced in Chapter 4.2, is now extended to the uncertain case by taking into account that the shapes involved in the transformation are not thin anymore. Given two shapes linked by a function $\mathbf{f}$ that depends on a parameter vector $\mathbf{p} \in \mathbb{R}^p$, the following propositions make it possible to contract the domain for the variables (parameters ans shapes).

5.5.3.1 Constraint $\mathbf{f}_\mathbf{p}(\mathbb{A}) \subset \mathbb{B}$

The following proposition allows to build a contractor, and even a separator, able to characterize an outer approximation of the set of feasible parameters associated to the constraint $\mathbf{f}_\mathbf{p}(\mathbb{A}) \subset \mathbb{B}$.

Proposition 5.4. Given $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$, $\mathbb{A} \in [\mathbb{A}]$ and $\mathbb{B} \in [\mathbb{B}]$, we have:

$$\{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}, \mathbf{p}) \subset \mathbb{B}\} \quad \subset \quad \left\{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}^-, \mathbf{p}) \subset \mathbb{B}^+\right\} \tag{5.49}$$

Proof. For $\mathbb{A} \in [\mathbb{A}]$, and $\mathbb{B} \in [\mathbb{B}]$,

$$\begin{aligned}
\{\mathbf{p}, \mathbf{f}(\mathbb{A}, \mathbf{p}) \subset \mathbb{B}\} \quad &= \quad \{\mathbf{p}, \forall \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}\} && (5.50)\\
&\overset{\mathbb{B} \subset \mathbb{B}^+}{\subset} \quad \left\{\mathbf{p}, \forall \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}^+\right\} && (5.51)\\
&\overset{\mathbb{A}^- \subset \mathbb{A}}{\subset} \quad \left\{\mathbf{p}, \forall \mathbf{a} \in \mathbb{A}^-, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}^+\right\} && (5.52)\\
&\subset \quad \left\{\mathbf{p}, \mathbf{f}(\mathbb{A}^-, \mathbf{p}) \subset \mathbb{B}^+\right\} && (5.53)
\end{aligned}$$

□

As a consequence, given $\mathbf{p} \in [\mathbf{p}]$ the contractor $\mathcal{C}_{\mathbf{f}\subset}$ associated to the constraint $\mathbf{f}(\mathbb{A}, \mathbf{p}) \subset \mathbb{B}$, $\mathbb{A} \in [\mathbb{A}]$, $\mathbb{B} \in [\mathbb{B}]$ is :

$$\mathcal{C}_{\mathbf{f}\subset}([\mathbf{p}], [\mathbb{A}], [\mathbb{B}]) = [\mathbf{p}] \cap \left[\!\left[\overline{proj_\mathbf{p}\left\{(\mathbb{A}^- \times \mathbb{R}^p) \cap \mathbf{f}^{-1}(\overline{\mathbb{B}^+})\right\}} \right]\!\right]. \tag{5.54}$$

where $[\![.]\!]$ denotes the convex hull.

Remark 5.7. In the case where $\mathbf{f_p}$ is bijective, the constraint $\mathbf{f_p}(\mathbb{A}) = \mathbb{B}$ can be decomposed into

$$\begin{cases} \mathbf{f_p}(\mathbb{A}) \subset \mathbb{B} \\ \mathbf{f_p^{-1}}(\mathbb{B}) \subset \mathbb{A} \end{cases} \tag{5.55}$$

for which Proposition 5.1 can be used twice. This leads to the following contractor:

$$\mathcal{C}_{\mathbf{f}=}([\mathbf{p}], [\mathbb{A}], [\mathbb{B}]) = \mathcal{C}_{\mathbf{f}\subset}([\mathbf{p}], [\mathbb{A}], [\mathbb{B}]) \cap \mathcal{C}_{\mathbf{f}^{-1}\subset}([\mathbf{p}], [\mathbb{B}], [\mathbb{A}]). \tag{5.56}$$

5.5.3.2 Constraint $\mathbf{f_p}(\mathbb{A}) \cap \mathbb{B} \neq \emptyset$

The *disjoint* constraint is also extended to the uncertain case. It has been used in the application of Chapter 4 to handle the impact area of sonar sensors.

Proposition 5.5. Given $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$, $\mathbb{A} \in [\mathbb{A}]$ and $\mathbb{B} \in [\mathbb{B}]$, we have:

$$\{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} \neq \emptyset\} \subset \left\{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}^+, \mathbf{p}) \cap \mathbb{B}^+ \neq \emptyset\right\} \tag{5.57}$$

Proof. Given $\mathbb{A} \in [\mathbb{A}]$, and $\mathbb{B} \in [\mathbb{B}]$, we have:

$$\begin{aligned} \{\mathbf{p} \in \mathbb{R}^p, \mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} \neq \emptyset\} &= \{\mathbf{p}, \exists \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}\} && (5.58) \\ &\overset{\mathbb{B}\subset\mathbb{B}^+}{\subset} \left\{\mathbf{p}, \exists \mathbf{a} \in \mathbb{A}, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}^+\right\} && (5.59) \\ &\overset{\mathbb{A}\subset\mathbb{A}^+}{\subset} \left\{\mathbf{p}, \exists \mathbf{a} \in \mathbb{A}^+, \mathbf{f}(\mathbf{a}, \mathbf{p}) \in \mathbb{B}^+\right\} && (5.60) \end{aligned}$$

□

As a consequence, the contractor $\mathcal{C}_{\mathbf{f}\neq}$ associated to the constraint $\mathbf{f}(\mathbb{A}, \mathbf{p}) \cap \mathbb{B} \neq \emptyset$, $\mathbb{A} \in [\mathbb{A}]$, $\mathbb{B} \in [\mathbb{B}]$ is :

$$\mathcal{C}_{\mathbf{f}\neq}([\mathbf{p}]) = [\mathbf{p}] \cap \left[proj_{\mathbf{p}} \left\{(\mathbb{A}^+ \times \mathbb{R}^p) \cap \mathbf{f}^{-1}(\mathbb{B}^+)\right\}\right]. \tag{5.61}$$

5.5.4 Application

As a preamble of applications presented in Chapter 6, the problem of *Shape Registration And Carving* (SRAC) is now considered. This situation corresponds, for instance, to a situation where a shape has partially been seen from two different positions. The transformation to move, from one point of view to another, depends on some parameters to be estimated. The goal of the shape registration and carving is to reduce the set of feasible parameters (registration) and contract observed shapes (carving).

Given two shapes $\mathbb{A} \in [\mathbb{A}]$, $\mathbb{B} \in [\mathbb{B}]$ linked by a bijective function $\mathbf{f_p}$ that depends on a parameter vector $\mathbf{p} \in [\mathbf{p}]$, the SRAC problem is described by the following SVCN:

$$\begin{cases} \mathbf{f_p}(\mathbb{A}) = \mathbb{B} \\ \mathbb{A} \in [\mathbb{A}] \\ \mathbb{B} \in [\mathbb{B}] \\ \mathbf{p} \in [\mathbf{p}] \end{cases} \tag{5.62}$$

From Proposition 5.4 a contractor can be built to contract the feasible parameter box $[\mathbf{p}]$. Then Proposition 5.1 is used to contract $[\mathbb{A}]$ and $[\mathbb{B}]$ with respect to $\mathbf{f}_{[\mathbf{p}]}$. The contraction procedure is given by Algorithm 5.1.

Algorithm 5.1 SRAC(in: $\mathbf{f}$, inout: $[\mathbf{p}]$, $[\mathbb{A}]$, $[\mathbb{B}]$)

```
1: $[\mathbf{p}] := \mathcal{C}_{\mathbf{f}=}([\mathbf{p}], [\mathbb{A}], [\mathbb{B}])$ (see Proposition 5.4)
2: $\begin{pmatrix} [\mathbb{A}] \\ [\mathbb{B}] \end{pmatrix} := \mathcal{C}_{\mathbf{f}_{[\mathbf{p}]}} \begin{pmatrix} [\mathbb{A}] \\ [\mathbb{B}] \end{pmatrix}$ (see Proposition 5.1)
```

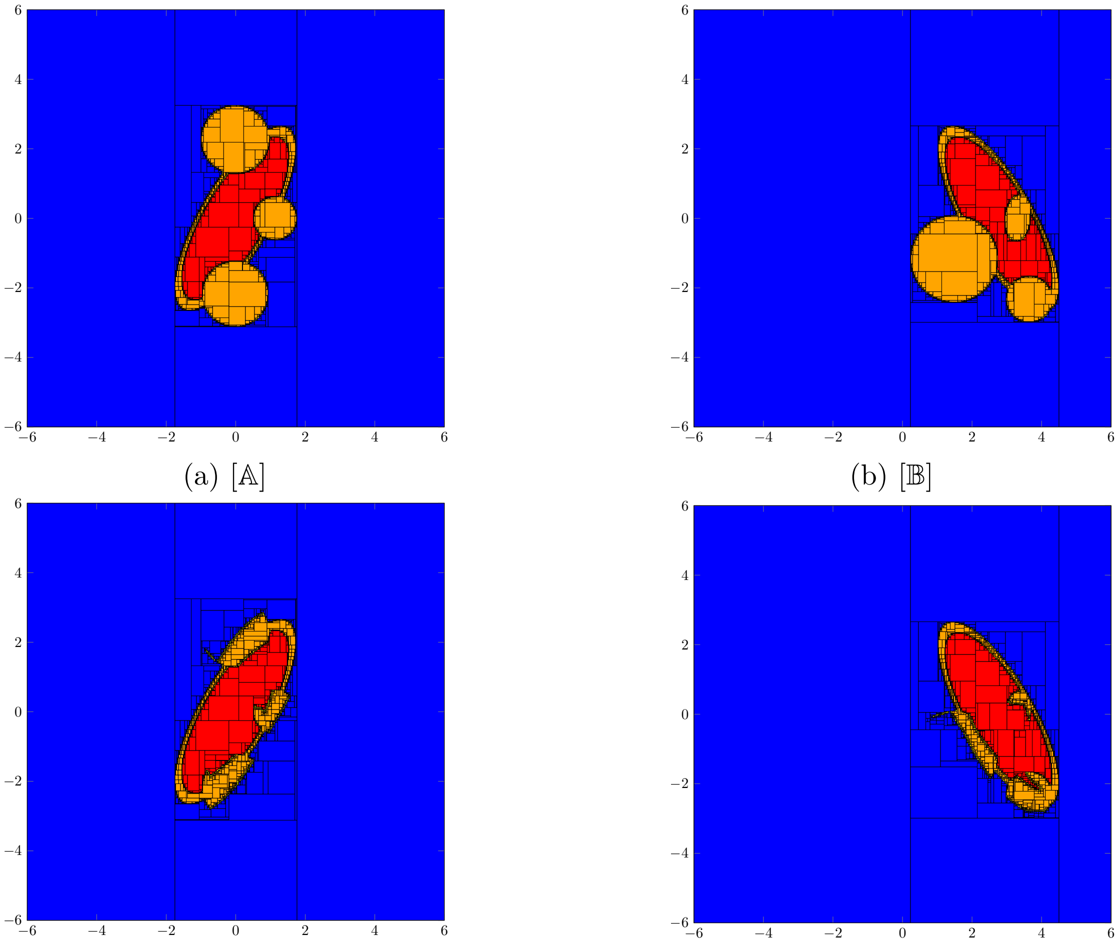
Example 5.5. Consider the two thick sets $[\mathbb{A}]$ and $[\mathbb{B}]$, depicted in Figure 5.10a and 5.10b, that are approximations of the same ellipse but from different positions. The transformation between these two views is defined by:

$$\mathbf{f_p}(\mathbf{x}) = \begin{pmatrix} p_1 \\ 0 \end{pmatrix} + \begin{pmatrix} \cos p_2 & -\sin p_2 \\ \sin p_2 & \cos p_2 \end{pmatrix} \cdot \mathbf{x} \tag{5.63}$$

The initial domain for $[\mathbf{p}]$ is $[0, 4] \times [-2, 2]$. Figure 5.11 shows the result of the set inversion using a separator.

Remark 5.8. From Equation (5.4), only an outer approximation of the set of feasible parameters can be computed. However, due to the huge pessimism introduced by $\mathbf{f}$, bisections are needed. The use of the separator, which provides an inner approximation, allows us to save computation time. It avoids useless bisections inside the solution set.

After the contraction, we obtained $[\mathbf{p}] = [2.65113,\ 2.90901] \times [-1.14117,\ -0.95238]$ which contains the theoretical value $\mathbf{p}^* = (2.8, -1)$ used to generate the images. This result is used to contract the thick set approximation. Figure 5.10 shows these sets after the contraction.



(a) $[\mathbb{A}]$

(b) $[\mathbb{B}]$

(c) $[\mathbb{A}] \sqcap \mathbf{f}_{[\mathbf{p}]}^{-1}([\mathbb{B}])$

(d) $[\mathbb{B}] \sqcap \mathbf{f}_{[\mathbf{p}]}([\mathbb{A}])$

Figure 5.10: Shape registration and carving. The initial set is approximated by 5.10a and 5.10b. Using the results of the carving process, the parameter box $[\mathbf{p}]$ (see Figure 5.11) has been contracted using Proposition 5.4. Then Proposition 5.1 is used to contract $[\mathbb{A}]$ (5.10c) and $[\mathbb{B}]$ (5.10d).

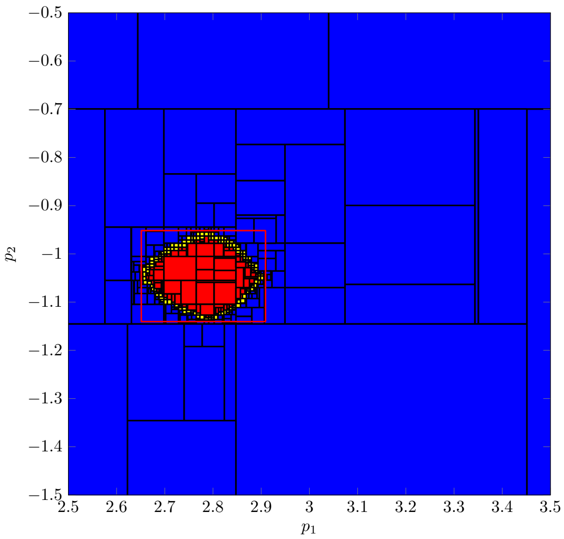


Figure 5.11: Contractions in the parameter space. The red box is $[\mathbf{p}] = [2.65113,\ 2.90901] \times [-1.14117,\ -0.95238]$

5.6 Conclusion

This chapter introduced the notion of interval of sets, called *thick set*. The thick set object is used as domains associated with set-valued variables (or shapes) that appear in our shaped-valued constraint network.

As an elementary constraint, the uncertain set inversion problem has been formalized. When the function can be parametric, a generic method that relies on the projection operator has been provided. This method can handle any non-linear function, but at a high computational cost.

Then, it has been used to extend the shape registration problem, introduced in the previous chapter, to the case where the shapes are also uncertain and need to be estimated. For that purpose, the carving problem, that aims at contracting their domains, has been introduced. Based on the notion of contractors on thick sets, which has been defined, a specific contractor applied to this problem was introduced. This contractor will be intensively used as a basic operator by applications presented in Chapter 6.

However, computing the set inversion with an uncertain function can be resource demanding and specific methods need to be developed in order to avoid bisections as much as possible. This is the goal of Appendix A, which introduces algorithms able to deal with the set inversion problem for which the uncertain function is described by an interval of functions.

This chapter concludes the presentation of the theoretical tools that have been developed. These tools will be used in a robotics context in order to solve the shape-based SLAM problem. Note that Appendix B provides another application of interval shapes. It focuses on a method able to deal with the guaranteed characterization of the part of the space that has been explored by a robot when the dimension of the visible space at time t may be smaller than that of the workspace.

Chapter 6

Shape-based SLAM

Contents

6.1 Introduction

The previous chapters have introduced all tools and algorithms needed to deal with the targeted applications of this manuscript. The main goal of this thesis is to provide an interval-based method to handle the SLAM problem in an unstructured environment (or Shape-based SLAM). In Chapter 4, the localization problem was first solved, but the map was assumed to be known without uncertainty. To tackle this issue, Chapter 5 introduced the notion of thick set, and defined a method to deal with the shape registration and carving problem. For the particular case of the translation, Appendix A introduces a method for solving the thick set inversion problem.

In the first section, a generic formulation of the shape-based SLAM is presented with a focus on how shapes are built from measurements. With respect to how a sensor reading is translated into a shape, two categories of interval shapes can be defined. For the first one, the shape provides an inner and an outer approximation of a part of the map. This allows us to formulate, in Section 6.3, the shape-based SLAM with inter-temporal constraints. As a direct application, we propose an efficient algorithm able to deal with the bathymetric SLAM problem. On the contrary, using sensors that only provide an inner approximation of the map, such as those used in Chapter 4, a different formulation will be needed. This formulation, called the *Dig-SLAM* problem, is introduced in Section 6.4. Section 6.5 concludes this chapter.

6.2 Problem Statement

In this section, the general formalism of the SLAM problem where both the map and measurements are shapes is given. Based on the thick-set contractors introduced in Chapter 5, a generic resolution method is then presented. The last part of this section focuses on the definition and the representation of physical features of the environment by shapes.

6.2.1 Formalism

A generic pose-based simultaneous localization and mapping problem can be described by:

$$\left\{\begin{array}{lcll} \dot{\mathbf{x}}(t) & = & \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) & \text{(evolution equation)} \\ \mathbb{Z}(t) & = & \mathbf{g}_{\mathbf{x}(t)}(\mathbb{M}) & \text{(map equation)} \\ \mathbf{x}(0) & = & \mathbf{x}_0 & \text{(initial position)} \end{array}\right. \tag{6.1}$$

where $t \in [t] \subset \mathbb{T}$ is the time, $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the input vector (in general associated with proprioceptive sensors), $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n$ is the evolution function and $\mathbf{x}_0$ is the initial state at $t = 0$. The set $\mathbb{M}$ is a subset of $\mathbb{R}^q$, where q is the dimension of the map (two or three in practice). The shape $\mathbb{Z}(t) \subset \mathbb{R}^r$ is an exteroceptive measurement collected by the robot (for instance by a camera, sonar, telemeter, ...), and expressed in the robot frame. The function $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^q \longrightarrow \mathbb{R}^r$ is the map function, which links the observed local shape $\mathbb{Z}(t)$ to the global map $\mathbb{M}$ with respect to the state of the vehicle $\mathbf{x}(t)$. For a given state vector $\mathbf{x}(t)$, $\mathbf{g}$ is assumed to be bijective *i.e*

$$\forall \mathbf{m} \in \mathbb{R}^q, \mathbf{g}_{\mathbf{x}(t)} \circ \mathbf{g}^{-1}_{\mathbf{x}(t)}(\mathbf{m}) = \mathbf{m} \tag{6.2}$$

Note that, most geometrical transformations, such as rotations, translations or dilations, are bijective functions that can be parametric.

In order to fit the set-membership formalism used in this thesis, the SLAM problem is handled by the following hybrid constraint network

$$\begin{cases} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbb{M},\ \mathbb{Z}(t_i) \\ \textbf{Constraints:} \\ \quad 1.\ \dot{\mathbf{x}}(\cdot) = \mathbf{f}\left(\mathbf{x}\left(\cdot\right), \mathbf{u}\left(\cdot\right)\right) \\ \quad 2.\ \mathbb{Z}(t_i) = \mathbf{g}_{\mathbf{x}(t_i)}(\mathbb{M}) \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbb{M}],\ [\mathbb{Z}]\left(t_i\right) \end{cases} \tag{6.3}$$

where measurements are taken at time $t_i \in \mathbb{T}$, and inputs of the system $\mathbf{u}(t)$ are assumed to be known and to belong to a bounded tube $[\mathbf{u}](.)$. Recall that the set-valued variables $\mathbb{Z}(t_i)$ are shapes. Their domains, defined by thick sets, are called *thick shapes*.

6.2.2 Generic resolution

Solving the SLAM problem implies generally to estimate both the trajectory and the map. The CN of Equation (6.3) is composed of two constraints for which contractors $\mathcal{C}_{\frac{d}{dt}}$, $\mathcal{C}_{SRAC}$ have been defined.

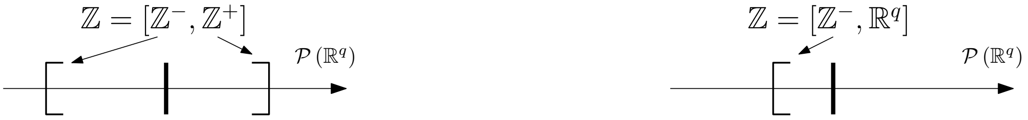
The first contractor $\mathcal{C}_{\frac{d}{dt}}$, defined in Section 2.5.4 on page 47, contracts the tube $[\mathbf{x}](t)$ with respect to the evolution equation $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ and the initial condition $\mathbf{x}_0$.

Then, the *Shape Registration And Carving* contractor $\mathcal{C}_{SRAC}$, introduced in Section 5.5.4 on page 114, is used to contract$[\mathbf{x}](t)$, the map, and the shape interval $[\mathbb{Z}(t)]$. Algorithm 6.1 provides a solution for solving the SLAM problem.

Note that this formulation is generic and, with respect to the nature of the measurements, needs to be adapted. Example of data that can be described by shapes are now given.

Algorithm 6.1 Explicit shape-based SLAM

```
$[\mathbb{M}] = [\emptyset, \mathbb{R}^q]$
$[\mathbf{x}](0) = \mathbf{x}_0$
repeat
    $[\mathbf{x}](.) := \mathcal{C}_{\frac{d}{dt}}([\mathbf{x}](.))$
    for $t_i$ in $\mathbb{T}$ do
        $\mathcal{C}_{SRAC}([\mathbf{x}](t_i), [\mathbb{Z}]\,(t_i), [\mathbb{M}])$
    end for
until no more contraction on $[\mathbf{x}](.)$
```


$\mathbb{Z} = [\mathbb{Z}^-, \mathbb{Z}^+]$
$\mathcal{P}(\mathbb{R}^q)$
$\mathbb{Z} = [\mathbb{Z}^-, \mathbb{R}^q]$
$\mathcal{P}(\mathbb{R}^q)$

(a) Double-sided observed shape

(b) One-sided observed shape

Figure 6.1: Observed thick shapes.

6.2.3 Shape-based measurements

As it has been said in the introduction of this manuscript, the map representation, based on subsets of $\mathbb{R}^n$, fits into the location-based map framework. Each position of the world is assumed to belong, or not, to a set (or shape) associated with a physical feature of the environment.

This shape is perceived by the embedded sensors of the robot. From raw data, we suppose that an algorithm is able to characterize an initial domain depicted by a thick shape. This initial domain has to take into account the uncertainty of the data acquisition process and the scope of the sensor.

Now, with respect to the map and the geometry of the sensor, two types of thick shapes need to be considered. The *double-sided observed shape*, for which we measure both an inner and outer approximation, and the *one-sided observed shape*, where only an inner approximation can be measured. These shapes are described in Figure 6.1

6.2.3.1 Double-sided observable shape

A double-sided observable shape can be defined when both $\mathbb{M}$ and $\overline{\mathbb{M}}$ can be measured. More formally, its interval shape approximation $[\mathbb{Z}] = [\mathbb{Z}^-, \mathbb{Z}^+]$ satisfies:

$$\begin{cases} \mathbb{Z}^- \subset \mathbb{M} \\ \overline{\mathbb{Z}^+} \subset \overline{\mathbb{M}} \end{cases} \tag{6.4}$$

This is illustrated by Figure 6.2. The map $\mathbb{M}$ is defined by the dashed area and the scope of the sensor, denoted by $\mathbb{V}$, is depicted by the dotted black rectangle. We assume that from sensor's data, it is possible to get an inner approximation $\mathbb{Z}^-$ (in red) of $\mathbb{M}$ and $\overline{\mathbb{Z}^+}$ (in blue) of $\overline{\mathbb{M}}$ that ranges in the scope of the sensor. Since robustly detecting the transitions between $\mathbb{M}$ and $\overline{\mathbb{M}}$ can be a challenging task, these transitions are enclosed inside the penumbra. The area, outside the scope of the sensor that has not been observed also belongs to the penumbra.

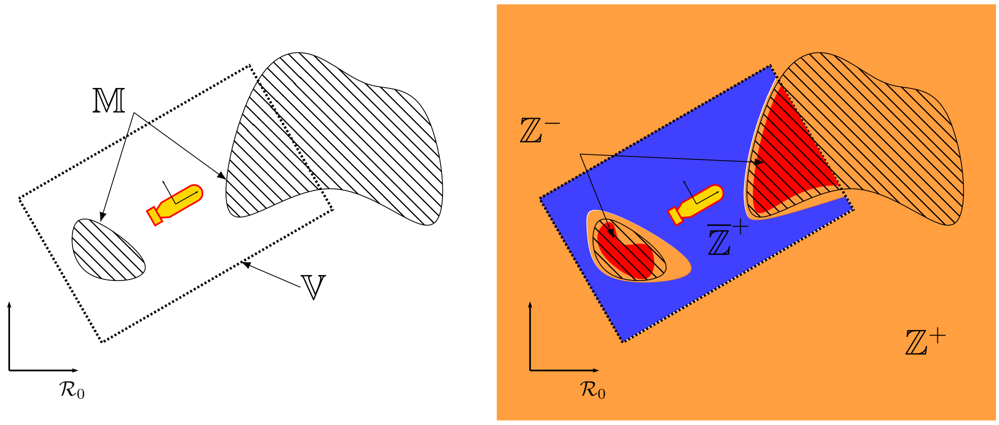


Figure 6.2: Double-sided observable shape. On the left-hand side, the map $\mathbb{M}$ is depicted by the dashed black lines. The black rectangle delimits the scope of the sensor which is the area that can be sensed by the vehicle. On the right-hand side, the thick shape $[\mathbb{Z}]$ which encloses the map i.e $\mathbb{Z}^- \subset \mathbb{M} \subset \mathbb{Z}^+$. The transition between $\mathbb{M}$ and $\overline{\mathbb{M}}$ belongs to the penumbra as well as all spaces that have not been observed, *i.e.* outside the scope of the sensor.

For instance, an airborne lidar measures distances between its carrier and the ground. These measurements, merged together into a local Digital Elevation Map (DEM), split the space into three areas:

- The shape $\mathbb{Z}^-$ that represents the space between the plane and the ground which is free of obstacles.
- The shape $\overline{\mathbb{Z}^+}$ that corresponds to the parts of the space below the ground which are assumed to be occupied.
- The shape $\mathbb{Z}^+$ that is composed of unobserved areas and the uncertainty of the bottom detection at the interface air/ground.

In this case, the DEM is represented as a shape of $\mathbb{R}^3$.

In Section 6.3, a new algorithm based on inter-temporal constraints will be introduced. It will take advantage of the fact that both $\mathbb{M}$ and $\overline{\mathbb{M}}$ can be observed.

6.2.3.2 One-sided observed shapes

With partially observed shapes $\mathbb{Z}$, the *interval* shape approximation only provides an inner view of the map, *i.e.*, $\mathbb{Z} \in [\mathbb{Z}^-, \mathbb{R}^q]$. It mainly happens when dealing with occupancy maps that separate free from occupied area sensed with ranging sensors. From distance readings, it is only possible to measure the free area between the vehicle and the nearest obstacle that lies within the scope of the sensor. Consequently, nothing can be inferred from the nature of the map outside this area. Figure 6.3 illustrates this concept with a pair of ultrasonic sensors. The regions of the working space that have not been sounded belong to the penumbra. Note that, the areas of the impact point, denoted $\Delta\mathbb{Z}$, have to be used for localization purpose.

This situation corresponds to the application introduced in Chapter 4. The Section 6.4 will extend this application to the SLAM problem.

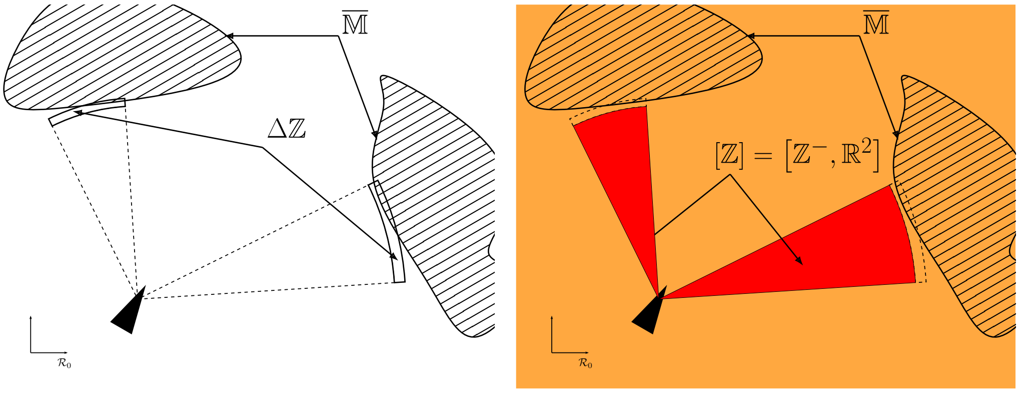


Figure 6.3: Occupancy based representation. On the left-hand side, the map is represented by the white area and obstacles are depicted in dashed black lines. Two sonar measurements are shown. For each of them, the impact point lies in a sector $\Delta\mathbb{Z}$. Since its position cannot be determined precisely, it makes this type of measurement partially observable. The thick shape associated to the measurements is drawn on the right side. The sounded part of the space, in red, is free from obstacles and belongs to $\mathbb{Z}^-$. Since nothing can be said about the nature of the environment sensed at a distance greater than the one measured, all the remaining space belongs to the penumbra.

6.3 Shape-based SLAM

The previous section introduced a general method to handle the SLAM problem with shapes. The proposed algorithm updates sequentially the trajectory, the map and the measurement shapes. However, building and updating large maps can be difficult and requires a lot of computing time. It involves solving many set inversion problems for which a specific algorithm has been introduced in Appendix A.

Sometimes, the map generated by the SLAM algorithm, is not a required output of the problem. Only the pose estimation is expected. In this situation, when shapes can be fully observed, the formalism of Equation (6.1) can be simplified. An inter-temporal formulation can be used to remove the map as a variable of the problem.

6.3.1 Inter-temporal SLAM

An inter-temporal constraint [Le Bars et al., 2012, Bethencourt and Jaulin, 2014] links two observations of the same object, taken at two different times. It can be somehow related to Graph SLAM methods [Thrun and Montemerlo, 2005] where reformulations and substitutions are performed to reduce the complexity of the problem.

The main idea behind this approach is to only consider transformations between two measurements, instead of between a measurement and the map.

Given two times $(t_1, t_2) \in \mathbb{T}^2$, the map equation can be changed into

$$\left\{\begin{array}{lcl} \mathbf{g}_{\mathbf{x}(t_i)}\left(\mathbb{Z}(t_i)\right) & = & \mathbb{M} \\ \mathbf{g}_{\mathbf{x}(t_j)}\left(\mathbb{Z}(t_j)\right) & = & \mathbb{M} \end{array}\right. \Leftrightarrow \mathbf{g}_{\mathbf{x}(t_i)}\left(\mathbb{Z}(t_i)\right) = \mathbf{g}_{\mathbf{x}(t_j)}\left(\mathbb{Z}(t_j)\right) \tag{6.5}$$

Now, in order to fit the shape transformation framework, we assume that the relation between the shapes $\mathbb{Z}(t_i)$ and $\mathbb{Z}(t_j)$ can be expressed by a function $\mathbf{h}_{\mathbf{p}_{ij}}$ that depends on a parameter $\mathbf{p}_{ij}$. This parameter links the states of the system between times t_i and t_j. This link is defined by a function ψ. From Equation (6.5), we have:

$$\mathbb{Z}(t_i) = \mathbf{g}^{-1}_{\mathbf{x}(t_i)} \circ \mathbf{g}_{\mathbf{x}(t_j)}\left(\mathbb{Z}(t_j)\right) \tag{6.6}$$

$$\Leftrightarrow \left\{\begin{array}{rcl} \mathbb{Z}(t_i) & = & \mathbf{h}_{\mathbf{p}_{ij}}(\mathbb{Z}(t_j)) \\ \mathbf{p}_{ij} & = & \psi(\mathbf{x}(t_i), \mathbf{x}(t_j)) \end{array}\right. \tag{6.7}$$

For instance, in the translation invariant case where $\mathbf{g}_{\mathbf{x}}(\mathbf{z}) = \mathbf{z} + \mathbf{x}$, we have

$$\mathbf{g}^{-1}_{\mathbf{x}(t_i)} \circ \mathbf{g}_{\mathbf{x}(t_j)}(\mathbf{z}) = \mathbf{x}(t_j) - \mathbf{x}(t_i) + \mathbf{z} \tag{6.8}$$

which becomes:

$$\mathbf{h}_{\mathbf{p}_{ij}}(\mathbf{z}) = \mathbf{p}_{ij} + \mathbf{z} \tag{6.9}$$

$$\mathbf{p}_{ij} = \phi(\mathbf{x}(t_i), \mathbf{x}(t_j)) = \mathbf{x}(t_j) - \mathbf{x}(t_i) \tag{6.10}$$

The SLAM problem is transformed into the following CN:

$$\left\{\begin{array}{l}\textbf{Variables: } \mathbf{x}(\cdot),\ \mathbb{M},\ \mathbb{Z}(t_i), \mathbf{p}_{ij}\\ \textbf{Constraints:}\\ \quad 1.\ \dot{\mathbf{x}}(\cdot) = \mathbf{f}\left(\mathbf{x}\left(\cdot\right), \mathbf{u}\left(\cdot\right)\right)\\ \quad 2.\ \mathbb{Z}(t_i) = \mathbf{h}_{\mathbf{p}_{ij}}\left(\mathbb{Z}(t_j)\right)\\ \quad 3.\ \mathbf{p}_{ij} = \phi(\mathbf{x}(t_i), \mathbf{x}(t_j))\\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbb{M}],\ [\mathbb{Z}]\,(t_i),\ [\mathbf{p}_{ij}]\end{array}\right. \tag{6.11}$$

where $(t_i, t_j) \in \mathbb{T}^2$ are times of measurements.

Algorithm 6.2 describes the propagation process used to solve the CN. As before, the trajectory is contracted using $\mathcal{C}_{\frac{d}{dt}}$ (line 4). For all pairs $(t_i, t_j) \in \mathbb{T}^2$, a two-step contraction process is used. Line 6, $\mathcal{C}_{SRAC}$ contractor is used to contract $[\mathbf{p}_{ij}]$ with respect to constraint 2. of CN (6.11). Line 7, the contractor $\mathcal{C}_\phi$ is used to propagate this information to the involved states. It can be built using classical forward-backward propagation methods. For instance, the contractor defined in Example 2.4 can be used for Equation (6.10).

Algorithm 6.2 Shape-based SLAM

```
1: $[\mathbf{x}](0) = \mathbf{x}_0$
2: $[\mathbf{p}_{ij}] := [-\infty, \infty]^n$
3: repeat
4:     $[\mathbf{x}](.) := \mathcal{C}_{\frac{d}{dt}}([\mathbf{x}](.))$
5:     for $(t_i, t_j)$ in $\mathbb{T}^2$ do
6:         $\mathcal{C}_{SRAC}\left([\mathbf{p}_{ij}], [\mathbb{Z}(t_i)], [\mathbb{Z}(t_j)]\right)$
7:         $\mathcal{C}_\psi([\mathbf{x}(t_i)], [\mathbf{x}(t_j)], [\mathbf{p}_{ij}])$
8:     end for
9: until no contraction on $[\mathbf{x}](.)$
```

This is illustrated by Figure 6.4 where two shapes are registered. Roughly, a given parameter vector is feasible if all red areas do not intersect the blue ones. The contractor based approach will guarantee that no feasible parameter vector will be removed. Consequently, this contractor can be applied to any pair $(t_i, t_j) \in \mathbb{T}^2$, without taking particular precaution.

6.3.2 Bathymetric SLAM

In order to illustrate the behavior of Algorithm 6.2, we consider a simulated test case that involves an AUV able to sense its environment using a multi-beam echo sounder. The vehicle made a survey pattern at a speed of 4 knots (around 2 m.s^{-1}) and covered an area of 600m^2 in 40 minutes. The AUV is equipped with an INS coupled with a DVL that provides speed and heading data every 0.1 seconds. Every

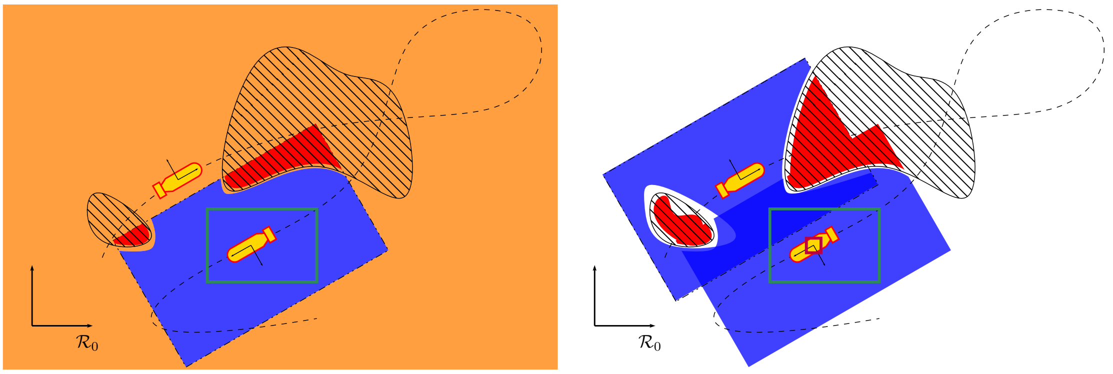


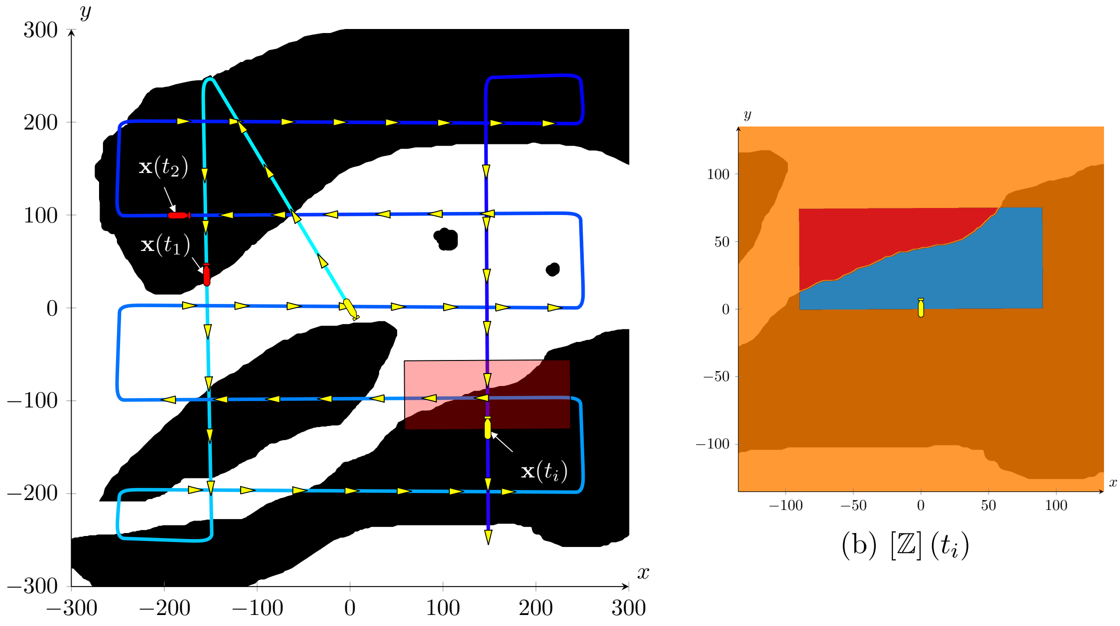
Figure 6.4: Example of the inter-temporal registration between two measurements.

25 seconds (around 40 meters), the last 30 seconds (75m) of bathymetric data are processed in order to build a local *Digital Elevation Map* (DEM). This DEM is segmented based on one or several elevation thresholds. The short-time positioning drift during this time window is assumed to be negligible with respect to data uncertainty. The attitude (roll, pitch and heading), depth and altitude of the AUV are directly measured. Their uncertainties are taken into account when the DEM is built. Consequently, only the horizontal position of the robot needs to be estimated. If this hypothesis is not acceptable, the method presented in Appendix B can be used to build a visibility mask in order to only take into account reliable data.

Figure 6.5 provides a simulation of the AUV trajectory on an unknown map. This map, associated to bathymetric data, splits the space into two complementary sets. A DEM can be seen as a shape of $\mathbb{R}^3$, which separates points above the seabed from those which are below. Now, in order to illustrate the proposed method, we choose to discretize it into a series of shapes of $\mathbb{R}^2$: $\mathbb{M}_1, \mathbb{M}_2, \dots, \mathbb{M}_m$ based on level sets extracted from the original DEM. The SLAM problem is then described by the following equations:

$$\left\{\begin{array}{rcll} \dot{\mathbf{x}}(t) & = & \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) & \text{(evolution equation)} \\ \mathbb{Z}_j(t) & = & \mathbf{g}(\mathbf{x}(t), \mathbb{M}_j) & \text{(observation equation)} \\ \mathbf{x}(0) & \in & \mathbb{X}_0 & \text{(initial state)} \end{array}\right.$$

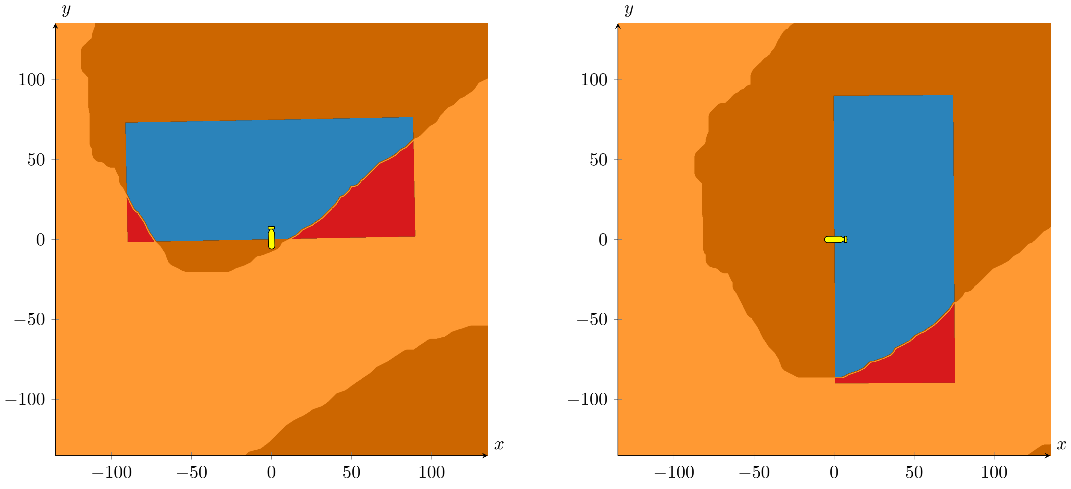
Now, combining several shapes associated to different level sets can be trivially done using the constraint network formalism. We consider only the shape associated with one level set. For instance, the white area corresponds to points with an elevation value that ranges between 10 and 13 meters. The black one corresponds to elevation values that are outside this interval. An example of a measured thick shape is given in Figure 6.5b.



(a) AUV trajectory

(b) $[\mathbb{Z}]\,(t_i)$

Figure 6.5: The map in black and white classifies the area into two complementary classes. The AUV starts at the center of the area and makes a survey pattern. On each yellow triangle, a measurement is taken. An example is shown on the right sub-figure which corresponds to the yellow AUV at time t_i. Note that the area which has not been sensed belongs to the penumbra. The scope of the sensor is depicted by the red rectangles. The measurements of the two red AUV, have been taken as an example to illustrate the inter-temporal contraction.



(a) $[\mathbb{Z}]\,(t_1)$

(b) $[\mathbb{Z}]\,(t_2)$

Figure 6.6: Thick shapes taken at times $t_1 = 4$ min 15 s and $t_2 = 26$ min 30 s.

This SLAM problem is described by the following equations:

$$\left\{\begin{array}{lclr} \dot{\mathbf{x}}(t) & = & \begin{pmatrix} \cos\psi(t) & -\sin\psi(t) \\ \sin\psi(t) & \cos\psi(t) \end{pmatrix} \cdot \mathbf{v}(t) & t \in \mathbb{R} \\ \mathbb{Z}(t_i) & = & \mathbb{M} - \mathbf{x}(t_i) & t_i \in \mathbb{T} \subset \mathbb{R} \end{array}\right. \tag{6.12}$$

where the state vector $\mathbf{x} = (p_x, p_y)^\intercal$ is the horizontal position of the AUV. The inputs are its horizontal speed vector $\mathbf{v}(t)$ and its headings $\psi(t)$. The quantities $\mathbf{v}(t)$ and $\psi(t)$ are measured every 0.1 seconds with an error of 0.02 $m.s^{-1}$ and 0.05° respectively. The initial state is assumed to be known and taken at $\mathbf{x}_0 = (0,0)^T$. Since the heading of the vehicle is known with a good accuracy, only translations are considered between the observation shapes and the map. This is modeled by the following system of equations:

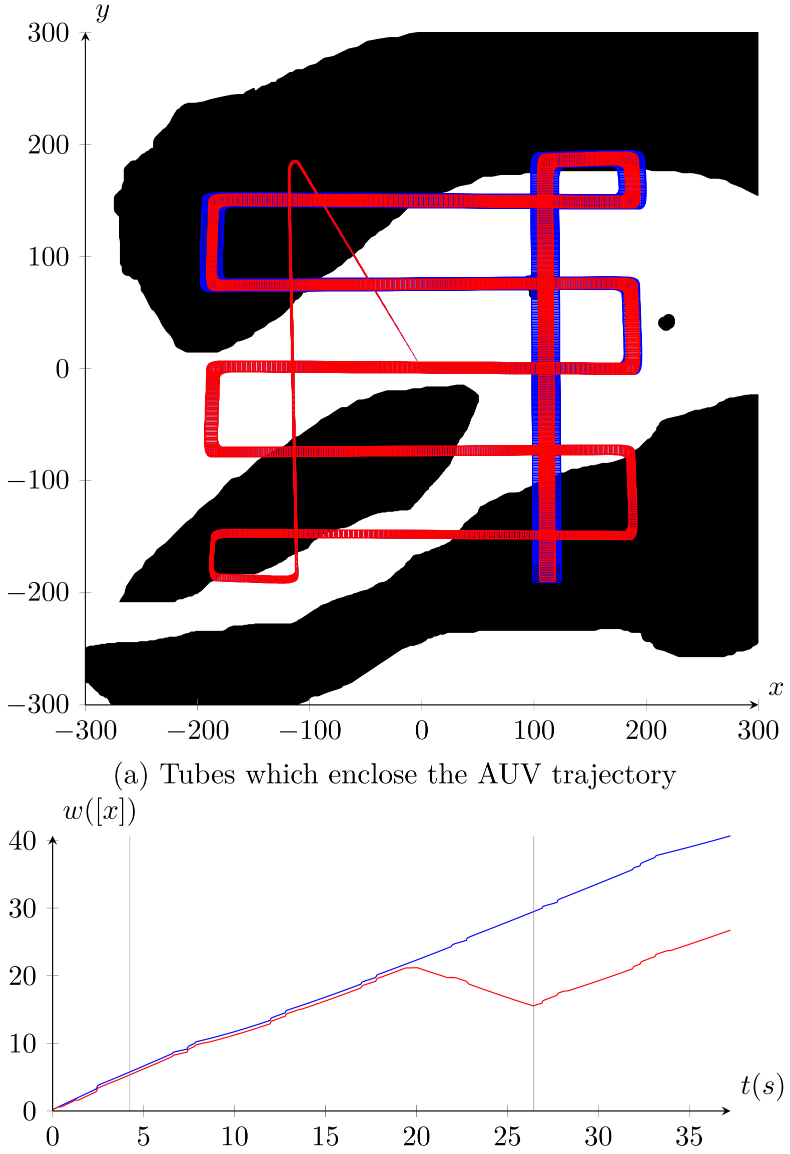
$$\left\{\begin{array}{l} \textbf{Variables: } \mathbf{x}(\cdot),\, \mathbb{M},\, \mathbb{Z}(t_i), \mathbf{p}_{ij} \\ \textbf{Constraints:} \\ \quad 1.\ \dot{\mathbf{x}}(\cdot) = \mathbf{f}\left(\mathbf{x}\left(\cdot\right), \mathbf{u}\left(\cdot\right)\right) \\ \quad 2.\ \mathbb{Z}(t_i) = \mathbb{Z}(t_j) + \mathbf{p}_{ij} \\ \quad 3.\ \mathbf{p}_{ij} = \mathbf{x}(t_j) - \mathbf{x}(t_i) \\ \textbf{Domains: } [\mathbf{x}](\cdot),\, [\mathbb{M}],\, [\mathbb{Z}]\,(t_i),\, [\mathbf{p}_{ij}] \end{array}\right. \tag{6.13}$$

where Algorithm 6.2 can be used.

The propagation process is now illustrated. The initial tube, which encloses the true trajectory is depicted in red on Figure 6.7. It was obtained by integrating the differential equation, from the initial vector $\mathbf{x}_0$, using the contractor $\mathcal{C}_{\frac{d}{dt}}$. The diameter of the tube is shown on Figure 6.7b. Since, only the proprioceptive measurements are used, its diameter grows almost linearly with time. In order to reduce the position drift, all pairs $(t_i, t_j) \in \mathbb{T}$ are registered.

The improvement obtained after a pair registration is illustrated with the two measurements, acquired by the two red AUV in Figure 6.5. These measurements, acquired at times t_1 = 4 min 15 s and t_2 = 26 min 30 s are depicted in Figure 6.6. The contractor $\mathcal{C}_{SRAR}$ is first used to estimate the translation vector between these two thick shapes. The values of the translation parameter vector before and after the contraction are given in Table 6.1. Then, the contractor $\mathcal{C}_{-}$ is enforced on the states at t_1, and t_2. Finally, the contractor $\mathcal{C}_{\frac{d}{dt}}$ is called to propagate the contractions on $[x](t_i)$ and $[x](t_j)$ to the whole trajectory. The result is shown by the green tube on Figure 6.7.

After 10 iterations Algorithm 6.2 stopped in less than 5 minutes. The resulting tube is shown on Figure 6.8 with its diameter. The position uncertainty remains bounded during the mission which is the expected result.



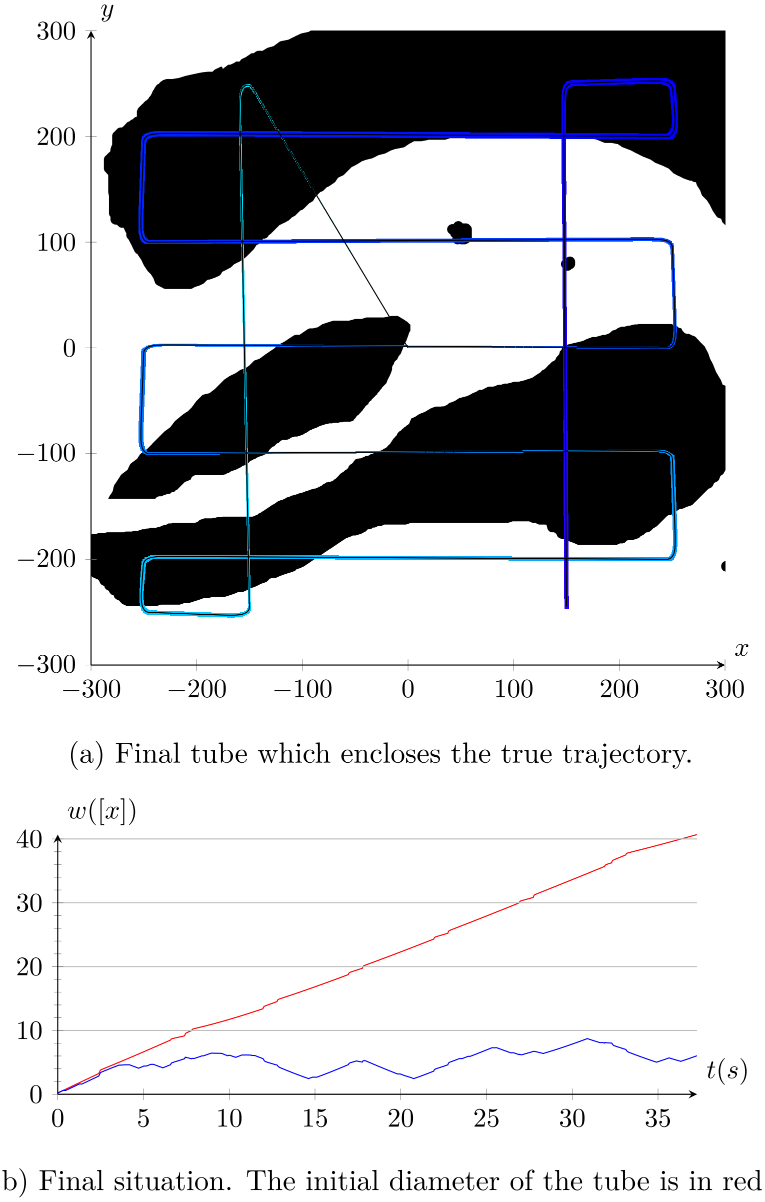
(a) Tubes which enclose the AUV trajectory

(b) Tube diameter, the vertical lines correspond to times t_1 and t_2 used to illustrate the algorithm.

Figure 6.7: Illustration of the contraction between timest_1 and t_2. The initial tube, in blue, is contracted into the red one.

	$[\mathbf{x}](t_1)$	$[\mathbf{x}](t_2)$	$[\mathbf{p}_{12}]$	$w([\mathbf{p}_{12}])$
before	$[-156.7, -154.3]$ $\times[31.2, 36.6]$	$[-200, -174.7]$ $\times[92.6, 107.7]$	$[-43.3, -20.3]$ $\times[61.2, 71.1]$	25.04
after	$[-156.7, -154.3]$ $\times[31.2, 36.6]$	$[-192.1, -181.8]$ $\times[93.8, 105.9]$	$[-37.8, -25.2]$ $\times[61.2, 71.1]$	16.02

Table 6.1: Shape registration. The two thick shapes, depicted on Figure 6.5, are registered using the contractor $\mathcal{C}_{SRAC}$. Then the contractor $\mathcal{C}_{-}$ is used to contract the states at t_1 and t_2. Note that only a contraction along the y-axis has been done which is consistent with the shapes used.



(a) Final tube which encloses the true trajectory.

(b) Final situation. The initial diameter of the tube is in red.

Figure 6.8: Final result after contraction

6.3.3 Discussion

The proposed method provides a way to deal with the SLAM problem where the observations and the map can be observed from double-sided (inside and outside). Thanks to the thick shape model, an efficient algorithm, based on inter-temporal constraints, has been proposed. It allows localizing the vehicle without explicitly building the map.

In the simulation, only the shape constraint, that links a given map representation with the measurement shape, has been used. Since the formalization of the problem fits the constraint network framework, additional constraints can be combined to improve the efficiency of the constraint propagation process.

Another improvement is to avoid testing all the pairs (t_i, t_j). Using the thick set inversion algorithm, it is possible to define a test, based on the scope of the sensor, which checks if two shape-measurements overlap or not. For two visibility sets $\mathbb{V}(t_i)$ and $\mathbb{V}(t_j)$ and a translation vector $[\mathbf{p}_{ij}]$, the following thick set is considered:

$$[\mathbb{X}] = [\mathbb{V}]\,(t_i) \cap ([\mathbb{V}]\,(t_j) + [\mathbf{p}_{ij}]) \tag{6.14}$$

If $\mathbb{X}^+ = \emptyset$, the two measurements do not overlap and the pair (t_i, t_j) should not be tested anymore. Else if $\mathbb{X}^- \neq \emptyset$, they overlap for sure. This information can be used, as in [Jaulin, 2016], to merge and associate data. In the last case, where $[\mathbb{X}] = [\emptyset, X^+]$, the translation is too uncertain to conclude and the pair must be tested.

If the measurement shapes $[\mathbb{Z}](t_i)$ are accurate enough, they do not need to be contracted. Consequently, only the transformation between two thick shapes needs to be estimated, which simplifies the contractor $\mathcal{C}_{SRAR}$. This hypothesis is done for this test case since we considered that bathymetric measurements are reliable and accurate.

Finally, the proposed method needs to be validated with real data. With respect to the targeted application, the specific process, that converts raw sensor data to thick shapes, needs to be developed.

6.4 Dig SLAM

The previous section deals with double-sided observable shapes that provide an inner and an outer observation of the map. In this section, the case of sensors that provide one-sided shape observation of the environment is discussed. This corresponds to the extension of the localization problem in an unstructured environment, introduced in Chapter 4.4, to the SLAM context.

6.4.1 Formulation of the Dig-SLAM

We consider the problem taken from [Jaulin, 2011], where a robot is moving in its environment and uses an omnidirectional range sensor to sense the free area around him. At a given time t, this sensor measures two sets: the *free zone*, denoted by $\mathbb{Z}(t)$, which represents the area in the scope of the sensor free of obstacles, and the *impact zone*, denoted $\Delta\mathbb{Z}(t)$ which is the border of $\mathbb{Z}(t)$ and contains the impact point. Note that measurements from most rangefinders (laser, infrared, ultrasound-based) can be cast into such two sets. Some examples are given in Figure 6.9.

This problem fits the formalism defined in this thesis and is modeled by:

$$\left\{\begin{array}{lcll} \dot{\mathbf{x}}(t) & = & \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) & \text{(evolution equation)} \\ \mathbb{Z}(t) & = & \mathbf{g}_{\mathbf{x}(t)}(\mathbb{M}) & \text{(observation equation)} \\ \mathbf{x}(0) & \in & \mathbb{X}_0 & \text{(initial state)} \end{array}\right. \tag{6.15}$$

where $\mathbb{M}$ depicts the set of points in the working space which are free of obstacles and $\mathbb{Z}(t)$ is the space dug by the measurement taken at time t. With respect to the uncertainty of the sensor, the measurement is changed into an interval shape $[\mathbb{Z}] = [\mathbb{Z}^-, \mathbb{Z}^+]$ with $\mathbb{Z}^+ = \mathbb{R}^2$. This corresponds to a one-sided observable shape, defined in Section 6.2.3.2.

The function $\mathbf{g} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$ is a transformation that links the local observation $\mathbb{Z}(t)$, expressed in the robot frame, to the map. For instance, in the translation invariant case, $\mathbf{g}$ is defined by:

$$\mathbf{g}_{\mathbf{x}(t)}(\mathbf{m}) = \mathbf{m} - \mathbf{x}(t). \tag{6.16}$$

Now, in order to take into account the fact that the impact zone $\Delta\mathbb{Z}$ intersects the map, the following constraint needs to be considered:

$$\Delta\mathbb{Z} \cap \mathbf{g}_{\mathbf{x}(t)}(\overline{\mathbb{M}}) \neq \emptyset. \tag{6.17}$$

In order to use the contractors previously introduced, the Dig-SLAM problem is formalized by the following constraint network:

$$\left\{\begin{array}{l} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbb{M},\ \mathbb{Z}(t_i) \\ \textbf{Constraints:} \\ \quad 1.\ \dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) \\ \quad 2.\ \mathbb{Z}(t) = \mathbf{g}_{\mathbf{x}(t)}(\mathbb{M}) \\ \quad 3.\ \Delta\mathbb{Z}(t) \cap \mathbf{g}_{\mathbf{x}(t)}(\overline{\mathbb{M}}) \neq \emptyset \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbb{M}],\ [\mathbb{Z}]\left(t\right) = [\mathbb{Z}(t), \mathbb{R}^q],\ [\Delta\mathbb{Z}]\left(t\right) = [\emptyset, \Delta\mathbb{Z}(t)] \end{array}\right. \tag{6.18}$$

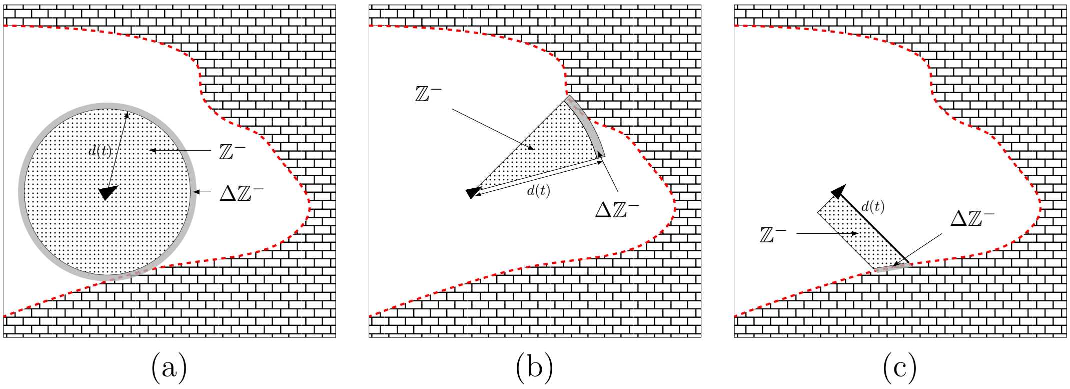


Figure 6.9: Examples of measurements which can be converted into interval shapes $[\mathbb{Z}] = [\mathbb{Z}^-, \mathbb{Z}^+]$ with $\mathbb{Z}^+ = \mathbb{R}^2$. The dotted area represents the *free space* $\mathbb{Z}^-$ and the light gray the *impact zone* $\Delta\mathbb{Z}(t)$. (a) an omni-directional rangefinder, where the impact point lies on a circle and the free space is a disk. (b) an ultrasonic rangefinder where $\mathbb{Z}^-$ is a sector and $\Delta\mathbb{Z}(t)$ is an arc. (c) represents the area swept by a laser rangefinder pointing perpendicularly to the path of the robot where methods introduced in Appendix B can be used. The set $\Delta\mathbb{Z}(t)$ is depicted by a curve.

where

1. describes the evolution of the robot with respect to its inputs using the evolution function.
2. corresponds to the measured free zone that must be inside the map.
3. corresponds to the impact zone that must intersect the border of the map.

However, because of the nature of $[\mathbb{Z}]$ and the fact that $[\mathbb{M}]$ needs to be estimated, Equation (2) of (6.18) can only be used to contract the inner approximation of the map. The following development justifies the need of considering the impact zone.

Given $[\mathbb{M}] = [\emptyset, \mathbb{R}^q]$ the initial map, a measurement $[\mathbb{Z}]\,(t) = [\mathbb{Z}(t), \mathbb{R}^q]$ and a box $[\mathbf{x}](t)$. The contractor $\mathcal{C}_{\mathbb{F}}$ defined in Proposition 5.3 on page 112 applied on $[\mathbb{Z}]$ and $[\mathbb{M}]$ leads to :

$$\mathcal{C}_{\mathbf{g}_{[\mathbf{x}](t)}} \begin{pmatrix} [\mathbb{Z}] \\ [\mathbb{M}] \end{pmatrix} = \begin{pmatrix} [\mathbb{Z}] \sqcap \mathbf{g}_{[\mathbf{x}](t)}\left([\mathbb{M}]\right) \\ [\mathbb{M}] \sqcap \mathbf{g}^{-1}_{[\mathbf{x}](t)}\left([\mathbb{Z}]\,(t)\right) \end{pmatrix} \tag{6.19}$$

$$= \begin{pmatrix} \left[\mathbb{Z} \cup \mathbf{g}_{[\mathbf{x}](t)}\left(\emptyset\right), \mathbb{R}^q \cap \mathbf{g}_{[\mathbf{x}](t)}\left(\mathbb{R}^q\right)\right] \\ \left[\emptyset \cup \mathbf{g}^{-1}_{[\mathbf{x}](t)}\left(\mathbb{Z}\right), \mathbb{R}^q \cap \mathbf{g}^{-1}_{[\mathbf{x}](t)}\left(\mathbb{R}^q\right)\right] \end{pmatrix} \tag{6.20}$$

$$= \begin{pmatrix} [\mathbb{Z}, \mathbb{R}^q] \\ \left[\mathbf{g}^{-1}_{[\mathbf{x}](t)}\left(\mathbb{Z}\right), \mathbb{R}^q\right] \end{pmatrix} \tag{6.21}$$

Consequently, only $\mathbb{M}^-$ can be contracted and $\mathbb{M}^+$ will remain undefined. This approximation is called on [Jaulin, 2011] the *dig zone* and represents the part of the free space that has been proved free of obstacles by the vehicle.

The registration contractor, defined by Proposition 5.4 needs an outer approximation of the map in order to contract the parameter's domain. This is why, Equation 3 of CN (6.18) needs to be used with Proposition 5.5 on page 114. We define the contractor

$$\mathcal{C}^{\neq}_{\mathbf{g}}\left([\mathbf{x}](t_i), \Delta\mathbb{Z}(t_i), \mathbb{M}^-\right) = [\mathbf{x}](t_i) \cap \left[\!\!\left[proj_{\mathbf{p}}\left\{(\Delta\mathbb{Z}(t_i) \times \mathbb{R}^q) \cap \mathbf{g}(\overline{\mathbb{M}^-})\right\}\right]\!\!\right] \tag{6.22}$$

The following algorithm can be used for solving the Dig-SLAM problem: Line 5, the map is updated and Line 6, the position of the robot is contracted by taking into account the fact that $\Delta\mathbb{Z}$ intersects $\overline{\mathbb{M}}$.

Algorithm 6.3 Dig SLAM(in: $\mathbb{T}$, $\mathbb{Z}(t_i)$, $\Delta\mathbb{Z}(t_i)$, inout: $[\mathbf{x}](.)$)

```
1: $[\mathbf{x}](0) = \mathbf{x}_0$
2: repeat
3:     $[\mathbf{x}](.) := \mathcal{C}_{\frac{d}{dt}}([\mathbf{x}](.))$
4:     for $t_i$ in $\mathbb{T}$ do
5:         $\mathbb{M}^- := \mathbb{M}^- \cup \mathbf{g}^{-1}_{[\mathbf{x}](t_i)}(\mathbb{Z})$
6:         $[\mathbf{x}](t_i) := \mathcal{C}^{\neq}_{\mathbf{g}}([\mathbf{x}](t_i), \Delta\mathbb{Z}, \mathbb{M}^-)$
7:     end for
8: until no contraction on $[\mathbf{x}](.)$
```

Note that the map needs to be explicitly built in order to contract the position of the vehicle. This algorithm is illustrated with the following application.

6.4.2 Application: Range only SLAM in an unstructured environment

From an academic point of view, the omnidirectional range-only posed-based SLAM can be seen as a canonical problem. It is the simplest and most significant of a large class of SLAM problems that are difficult to solve. The development of tools to solve properly and efficiently this problem will be useful to solve many other SLAM problems.

As an illustration, consider a mobile robot described by the following range-only SLAM equations:

$$\left\{\begin{array}{rcl} \dot{x_1}(t) & = & u_1(t)\cos(u_2(t)) \\ \dot{x_2}(t) & = & u_1(t)\sin(u_2(t)) \\ y(t) & = & d(\mathbf{x}(t), \mathbb{M}) \end{array}\right. \tag{6.23}$$

The inputs of the system are the speed u_1 and the heading u_2 of the robot. The measurement $y(t)$ corresponds to the closest distance of the robot to the map which could have been obtained by using an omni-directional sonar with angular aperture of 2π. As illustrated by Figure 6.9, the scalar measurement $y(t)$, assumed to belong to $[y(t)] = [y^-(t), y^+(t)]$, can be cast into the interval shape $[\mathbb{Z}] = [\mathbb{Z}^-, \mathbb{R}^2]$, with

$$\mathbb{Z}^- = \left\{ z \in \mathbb{R}^2 \mid \sqrt{z_1^2 + z_2^2} < y^-(t) \right\}.$$

which is a disk of center $(0, 0)$, *i.e.* expressed in the robot frame. The impact area $\Delta\mathbb{Z}$ is defined by

$$\Delta\mathbb{Z} = \left\{ z \in \mathbb{R}^2 \mid \sqrt{z_1^2 + z_2^2} \in [y(t)] \right\}.$$

The quantities u_1, u_2 and z are measured every $0.1\,s$ with an error of 0.001 m.s^{-1} , 0.001 rad.s^{-1} , and $0.01\,m$, respectively. The initial state, taken as $\mathbf{x} = (0, 0)^T$, is assumed to be known. Fig. 6.10 provides a simulation of the robot moving inside an unknown map. This map is composed by segments but this is not required by the method. The shape of the map could be arbitrary and no parametric representation of the map is needed.

This problem is a particular case of the generic formulation of Equation (6.1). The translation invariant case can be expressed in terms of Minkowsky sums. It is a generalization of the localization problem of Chapter 4. It can be described by the following CN:

$$\left\{ \begin{array}{l} \textbf{Variables: } \mathbf{x}(\cdot),\ \mathbb{M} \\ \textbf{Constraints:} \\ \quad 1.\ \dot{\mathbf{x}}(\cdot) = \mathbf{f}\left(\mathbf{x}\left(\cdot\right), \mathbf{u}\left(\cdot\right)\right) \\ \quad 2.\ \mathbb{Z}(t) + \mathbf{x}(t) \subset \mathbb{M} \\ \quad 3.\ \left(\Delta\mathbb{Z}(t) + \mathbf{x}(t)\right) \cap \overline{\mathbb{M}} \neq \emptyset \\ \textbf{Domains: } [\mathbf{x}](\cdot),\ [\mathbb{M}] \end{array} \right. \tag{6.24}$$

which can be solved with Algorithm 6.3.

An illustration of the interval propagation method is depicted in Figure 6.12. The propagation process stops after 4 iterations in less than 40 seconds. The code of [Jaulin, 2011] provided by the author runs in around 90 seconds on the same computer.

Figure 6.12 shows the computed tube with the true path in black and corresponds to contractions of the inner approximation of the map. The true map, defined with segments, is superimposed to show the accuracy of the inner approximation of the map.

Figure 6.11 shows how Equation (3) of CN 6.24 is used to contract the position with respect to the map.

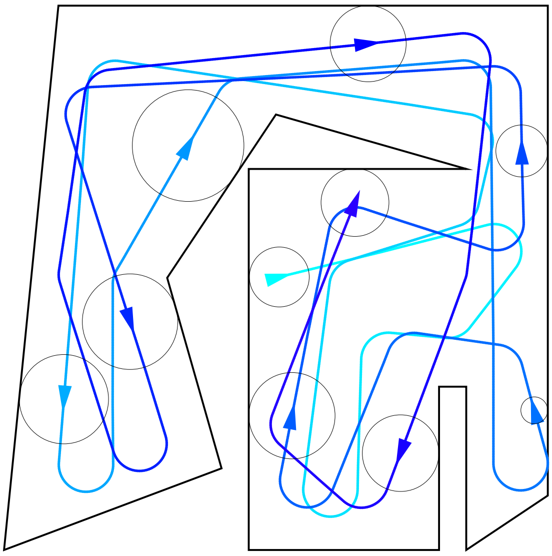


Figure 6.10: Trajectory of the robot. Feasible positions are inside the map, delimited by the polygon. Some measurements are depicted by black circles and correspond to the distance to the nearest obstacle around the robot. The framebox is $[-10, 10] \times [-10, 10]$.

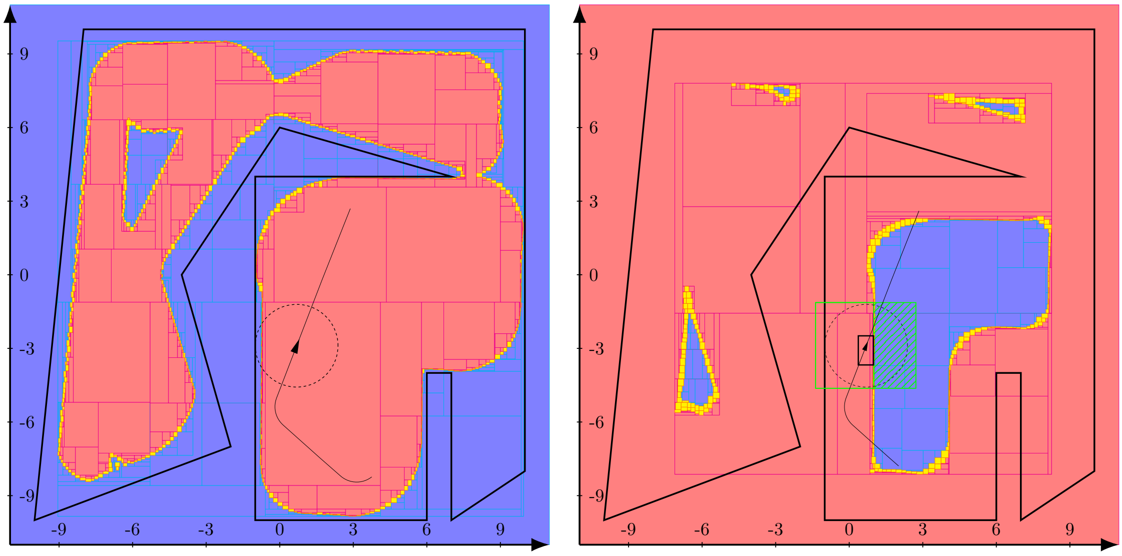


Figure 6.11: Illustration of the contraction of $[\mathbf{x}](t)$, for $t = 1230$ sec. Left: $\mathbb{M}^-$ with a measurement $\Delta\mathbb{Z}(t)$ depicted by the dashed circle. Right: The box $[\mathbf{x}](t)$ before contraction is depicted in green. The dashed green area has been removed by the contraction using Equation 3 of 6.24. The subpaving corresponds to $\overline{\mathbb{M}^-} \oplus -(\Delta\mathbb{Z}(t))$. The black box corresponds to $[\mathbf{x}](t)$ after calling $\mathcal{C}_{\frac{d}{dt}}$.

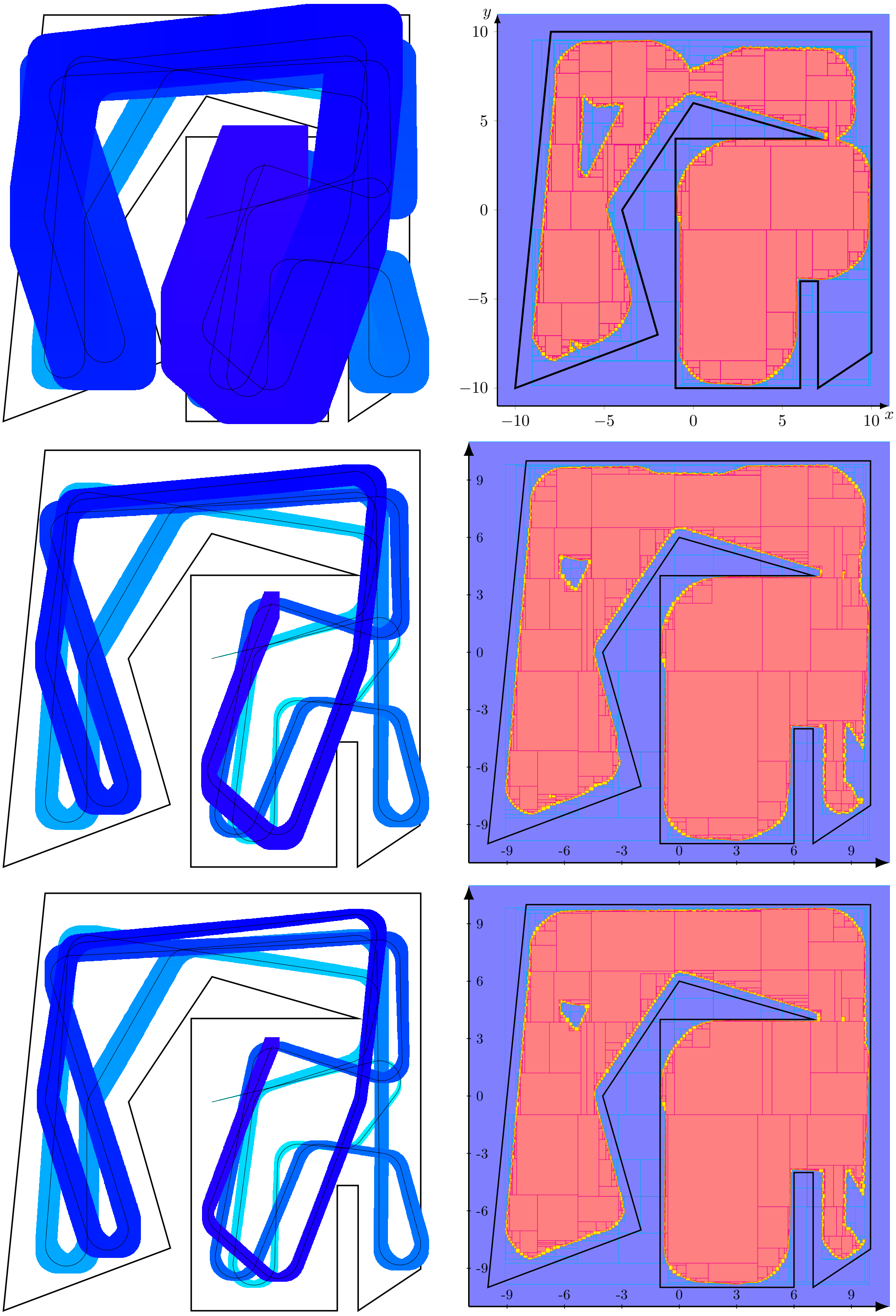


Figure 6.12: Left: Contraction on the tube $[\mathbf{x}](.)$ during the propagation process. Right: Associated estimation of $\mathbb{M}^{-}$. The true map is depicted by the black segments.

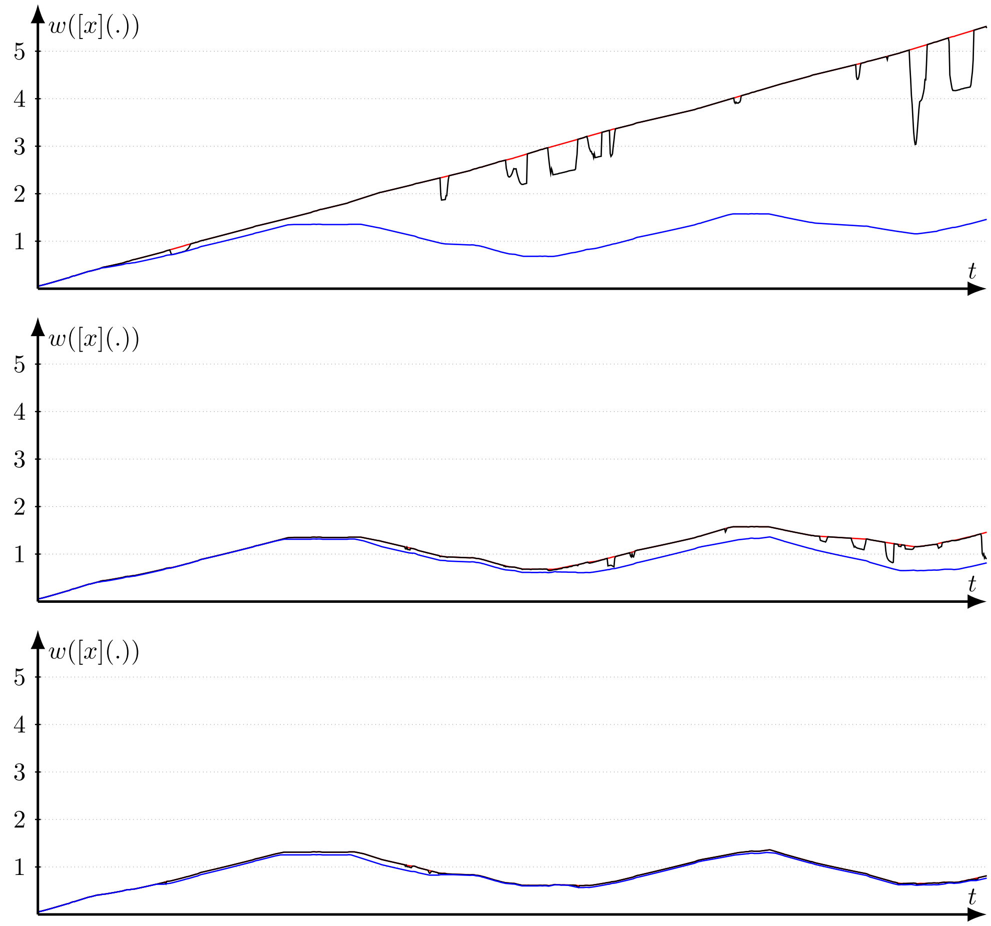


Figure 6.13: Evolution of the width of the tube. In red before contraction, black after calling $\mathcal{C}_2$ at each time, and blue after running the contractor $\mathcal{C}_{\frac{d}{dt}}$. After three iterations, no significant contraction happens. The maximal pose uncertainty is obtained when the robot is on the left side of the map. The minimum is obtained when it is near its starting position.

6.4.3 Discussion

This method is efficient to solve the SLAM problem where partially observed shapes are used to build the map and localize the map.

In the translation invariant case, the localization process is based on a dilation of the map by $\Delta\mathbb{Z}$. If $\mathbb{M}^-$ contains small holes, as it is the case in Figure 6.12, these holes grow which limits the efficiency of the contraction process. The computation of the inner approximation of the map must be done carefully.

Arbitrary shapes that are depicting the free space can be considered and, in a theoretical point of view, the heading can be estimated but with a higher computing cost.

6.5 Conclusion

In this chapter, the Simultaneous Localization and Mapping in an unstructured environment has been tackled using shapes. To our knowledge, this is new in a SLAM context and this contitutes the major contribution of this thesis. This problem is known as hard to solve due to the fact that no geometrical features can be extracted from the measurements. We proposed a formalism that allows us to handle rigorously nonlinearity and sensor uncertainties while keeping the guarantee and integrity of the results, with respect to the initial hypothesis.

In order to illustrate the proposed methods, two test-cases have been considered as examples. In the first one, the shape-based SLAM has been demonstrated on a simulated test-case based on bathymetric data. This type of datas allows to build a double-sided shape for which an inter-temporal-based approache has been used. The second test-case illustrated the dig SLAM formalism which involves sensors that are able to measure the parts of the working space which are free of obstacles. This problem is an extension of the application presented in Chapter 4 in the SLAM context.

The thick shape representation allows us to efficiently represent sensor uncertainties and the limit of the scope of the observation. This last point allows a paire-wise registration without preliminary associations. Merge different sensor.

The proposed method provides a generic abstraction of sensor reading that remains strongly bounded with physical features of the environment. The main difficulty is to be able to find robust algorithms to transform raw data into thick shapes.

Outliers can eventually be handled using the extension of the relaxed intersection on thick sets that has been presented in [Desrochers and Jaulin, 2016b]. The constraint network must be reformulated in order to efficiently take advantage of this operator.

Chapter 7

Conclusion

This PhD thesis introduces new methods based on Interval Analysis, able to deal with constraints that involve uncertain sets. These types of constraints are mainly encountered in robotic applications in order to handle unstructured environments. We proposed and illustrated a generic method that is able to solve the SLAM problem when measurements are shapes, *i.e.,* subsets of $\mathbb{R}^2$ of $\mathbb{R}^3$.

For that purpose, we introduced the shape *registration and carving* problem, where tools, algorithms, and applications have been presented along the chapters of this manuscript.

Summary of the Contributions

The key points of this thesis are now summarized. In Chapter 2, interval analysis is introduced. The main point of this chapter is the introduction of *separators*. Coupled with a *Paver*, which is a branching algorithm, separators allow to compute an inner and an outer approximation of a set defined by constraints. Their uses made it possible to characterize rigorously the pessimism introduced by interval-based algorithms. This work was published in [Jaulin and Desrochers, 2014] before the beginning of this thesis.

Now complex sets are defined by a combination of elementary ones, for which atomic separators are available. In Chapter 3, new theorems have been introduced in order to define a separator dedicated to the polar constraint, which is one of the important atomic constraints used in the context of localization. This constraint, associated with the change between Cartesian to Polar coordinates is often met in robotics, each time range and bearing measurements are involved. This work was published in [Desrochers and Jaulin, 2016a].

Based on the notion of projection of separators, the *shape registration* problem is presented in Chapter 4. this problem consists of estimating the smallest set of feasible parameters of the transformation which links two sets that we called *shapes*. This primitive constraint is at the center of our algorithms. A first localization application in an unstructured environment is then tackled using the polar separator (published in [Desrochers and Jaulin, 2017c]). Both the map and measurements are defined by shapes which, in this chapter, are assumed to be known.

In order to handle the shape uncertainty, the notion of intervals of sets, called a thick set, has been introduced in Chapter 5. Even if the notion of thickset is not new in the literature, a main contribution of this thesis is its exploitation for solving robotics problems. At the heart of the thick set theory, lies the notion of the *penumbra*. The penumbra characterizes the uncertainty that comes from the initial hypothesis of the problem. It allows us to prove that, for a box $[\mathbf{x}]$ inside the penumbra, nothing can be proved for any of its elements $\mathbf{x}$ with respect to the solution set.

This was illustrated in Appendix B where points that belong to the penumbra are seen for some feasible trajectories but not for all. Without uncertainty of the position of the vehicle, the penumbra would not have existed.

When dealing with sensor readings, the penumbra makes it possible to classify parts of the research space for which not enough information is available to conclude anything. Either because the area has not been sensed or because the process that builds the thick set from measurements cannot classify this part of the space. This can be the case with automatic segmentation algorithms.

The arithmetic and operations that can be performed on thick sets were also introduced. Another work which has not been included in the manuscript concerns the relaxed intersection of thick set presented in [Desrochers and Jaulin, 2016b].

The shape registration problem, introduced in Chapter 4 was extended to the uncertain case, were variables of the constraint network are shapes and their domains are thick sets, that we called *thick shapes*.

Based on the notion of contraction on thick sets, the shape carving problem has been defined. Given two thick shapes linked by an uncertain function, it aims at contracting the thick shapes using shape contractors. This contraction was done by solving an uncertain set inversion problem. When the uncertain function depends on parameters, a first method based on projection has been provided.

With respect to the SLAM problem, this constraint is mainly used to build the map or filter the measurements.

Finally, Chapter 6 deals with our targeted application. A SLAM algorithm that involves shapes has been introduced. The formalism, based on thick shapes, allows us to handle efficiently unstructured environments. It has been illustrated through two examples.

Open-source library

A significant contribution of this work is the development by the author of the new open-source library Pyibex, freely available at:

- http://www.ensta-bretagne.fr/desrochers/pyibex

This project gathers all the elementary tools presented in this document. The reader will be able to process the simulated examples and build its own solvers for the resolution of shape-based problems.

Overall Prospects

We are convinced that the proposed methods are relevant to address the SLAM problem in complex environments. But, in order to be used, they require a way to be tried and diffused. For that purpose, the development of the *pyIbex* library will be continued in order to make the use of interval analysis in robotics applications easier.

The shape registration problem relies mainly on the use of the projection algorithm. Its efficiency must be improved in order to be able to handle more complex transformations, such as the rotation.

The key point toward real applications is the processing chain that converts raw sensor data to interval shapes. This chain needs to robustly and reliably classify sensor outputs taking into account the notion of the penumbra. For a given set, the classical representation binaries the world. A point must belong or not to the shape. On the contrary, the notion of interval shape allows us to handle all data, which cannot be reliably classified inside or outside a measurement shape. For instance, with side scan sonar images, shadows are classified in the penumbra.

The interest of the shape carving process has not been fully demonstrated in this manuscript. The shape carving method may be used to perform pattern recognition. For instance, an object partially defined by an interval shape can be matched with another one. The interval shape representation allows to only consider the reliable parts of the initial object which are compared.

Outliers could also be taken into account by using the thick relaxed intersection. This needs to be correctly formalized and validated on actual experiments.

Appendix A

Thick Set Inversion

Contents

A.1 Introduction

In this thesis, the set inversion problem $\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$, where both $\mathbf{f}$ and $\mathbb{Y}$ are uncertain has been introduced. In the case where $\mathbf{f}$ depends on a parameter, a method has been given but it relies on projection algorithms that bisect along the parameter space.

In this chapter, we assume that $\mathbf{f} \in [\mathbf{f}] = [\mathbf{f}^-, \mathbf{f}^+]$ where $\mathbf{f}^-, \mathbf{f}^+$ are two known functions from $\mathbb{R}^n$ to $\mathbb{R}^m$. We also assume that the uncertain set $\mathbb{Y}$ belongs to an interval of sets, denoted by $[\![\mathbb{Y}]\!]$. We say that $[\mathbf{f}]$ is a *thick function* and that $[\![\mathbb{Y}]\!]$ is a *thick set*. As illustrated by the following example, existing interval methods can still be used to deal with this type of uncertainties, but they accumulate on a thick boundary which is called the *penumbra*. This accumulation makes classical interval methods inefficient, since they spend most of the computation time to test tiny boxes that are inside the penumbra.

Example A.1. Consider the set inversion problem $\mathbb{X} = f^{-1}([y])$ with $[y] = [0, 4]$. We assume that f is uncertain and that we only know that for all $\mathbf{x}$

$$f(\mathbf{x}) \in [f](\mathbf{x}) = (x_1 - [a_1])^2 + (x_2 - [a_2])^2 . \tag{A.1}$$

with $[a_1] = [0, 1]$, $[a_2] = [0, 1]$. Note that f is not necessarily a circular paraboloid, and may correspond to any weird function satisfying the enclosure condition. Since, for all $\mathbf{x}$, $[f](\mathbf{x})$ is an interval of $\mathbb{R}$, the function $[f]$ is a thick function. More precisely, we have

$$[f](\mathbf{x}) = \left[f^-(\mathbf{x}), f^+(\mathbf{x})\right] \tag{A.2}$$

where

$$f^-(\mathbf{x}) = \min_{\mathbf{a}\in[0,1]\times[0,1]} (x_1 - a_1)^2 + (x_2 - a_2)^2 \tag{A.3}$$

and

$$f^+(\mathbf{x}) = \max_{\mathbf{a}\in[0,1]\times[0,1]} (x_1 - a_1)^2 + (x_2 - a_2)^2 . \tag{A.4}$$

Using a classical interval arithmetic [Moore, 1979], we can easily test if a box $[\mathbf{x}] = [x_1] \times [x_2]$ is inside or outside the solution set $\mathbb{X}$:

$$\begin{array}{lccc} (i) & ([x_1] - [a_1])^2 + ([x_2] - [a_2])^2 \subset [y] & \Rightarrow & [\mathbf{x}] \subset \mathbb{X} \\ (ii) & ([x_1] - [a_1])^2 + ([x_2] - [a_2])^2 \cap [y] = \emptyset & \Rightarrow & [\mathbf{x}] \cap \mathbb{X} = \emptyset. \end{array}$$

Now, we are not able conclude anything if none of these conditions is satisfied. Figure A.1 (left) corresponds to the result of a paver based on these two tests (see the Set Inversion algorithm recalled at Subsection A.4.1). Red boxes satisfy the inner test *(i)*, blue boxes satisfy the outer test (ii) and yellow boxes satisfy neither Test (i) nor Test (ii). The yellow boxes are not bisected by the paver since they reached the required accuracy. They cover a zone, called the *penumbra*, which corresponds to the part of the plane for which both the inner test and the outer test fail. Of course, if we were able to conclude that a box is inside the penumbra, many bisections would have been avoided. We would thus get a picture similar to Figure A.1 (right) which is an approximation of a thick set with the inner part (red), the outer part (blue) and the penumbra (orange).

Now, when dealing with practical applications, the penumbra often exists as for instance when we want to characterize the zone that has actually been explored by a robot, detailed in the Appendix B, in case of partial observability [Grimson and Lozano-Perez, 1985] or in the shape carving problem introduced in Chapter 5. Characterizing the penumbra from inside will allow us to save computing time, but also to make the difference between the uncertainty due to the computation and that due to the initial uncertainties of the input parameters. The objective of this chapter is to extend set inversion to the thick case (where a penumbra exists) and to show how to conclude that a box is inside the penumbra without bisections

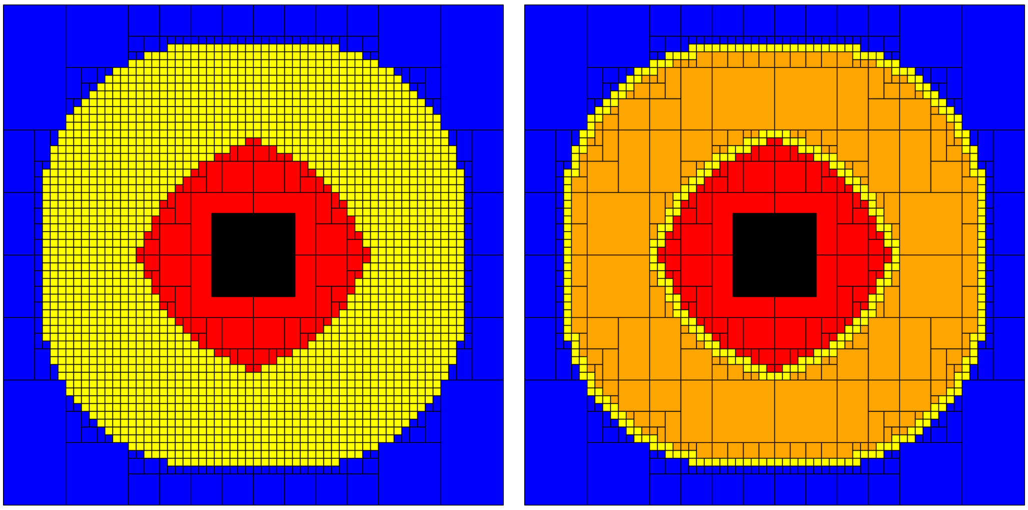


Figure A.1: Left: Classical interval methods accumulate on the thick boundary (the penumbra). Right: the method we propose here will allow a fast treatment of the penumbra. The frame box is $[-2,4]\times[-2,4]$ and the black box corresponds to $[\mathbf{a}]$.

along $[f]$.

The chapter is organized as follows. Section A.3 presents the new notion of thick intervals and thick boxes to be used for solving the thick set-inversion problem. Section A.4 generalizes the classical set-inversion algorithm to the thick case by introducing the new notion of thick inclusion function. Section A.5 illustrates the principle of the method on five test cases with one involving an actual underwater robot. Section B.6 concludes the chapter.

The results in this chapter are extracted from [Desrochers and Jaulin, 2017d]. *Thick objects* (functions, intervals, sets) are denoted with a double brackets notation $[\![.]\!]$.

A.2 Problem Statement

Notation. We introduce a specific notation involving the quantifier $\forall$ when dealing with thick sets. Given two thick sets $[\![\mathbb{A}]\!]$ and $[\![\mathbb{B}]\!]$, we define:

$$\begin{array}{rcl}
([\![\mathbb{A}]\!]\subset[\![\mathbb{B}]\!])^{\forall} & \Leftrightarrow & \forall\mathbb{A}\in[\![\mathbb{A}]\!],\,\forall\mathbb{B}\in[\![\mathbb{B}]\!],\mathbb{A}\subset\mathbb{B}\\
([\![\mathbb{A}]\!]\not\subset[\![\mathbb{B}]\!])^{\forall} & \Leftrightarrow & \forall\mathbb{A}\in[\![\mathbb{A}]\!],\,\forall\mathbb{B}\in[\![\mathbb{B}]\!],\mathbb{A}\not\subset\mathbb{B}\\
([\![\mathbb{A}]\!]\cap[\![\mathbb{B}]\!]=\emptyset)^{\forall} & \Leftrightarrow & \forall\mathbb{A}\in[\![\mathbb{A}]\!],\,\forall\mathbb{B}\in[\![\mathbb{B}]\!],\mathbb{A}\cap\mathbb{B}=\emptyset\\
([\![\mathbb{A}]\!]\cap[\![\mathbb{B}]\!]\neq\emptyset)^{\forall} & \Leftrightarrow & \forall\mathbb{A}\in[\![\mathbb{A}]\!],\,\forall\mathbb{B}\in[\![\mathbb{B}]\!],\mathbb{A}\cap\mathbb{B}\neq\emptyset.
\end{array}\tag{A.5}$$

Thick function. Based on the notion of lattices, introduced in Section 5.2.1, the definition of the intervals of function, called a *thick function* is now presented.

Denote by $(\mathcal{F}(\mathbb{R}^n, \mathbb{R}^m), \leq)$ the set of all functions from $\mathbb{R}^n \to \mathbb{R}^m$ equipped with the order relation $\leq$ defined as follows

$$\mathbf{f} \leq \mathbf{g} \Leftrightarrow \forall i \in \{1, \ldots, m\}, \forall \mathbf{x} \in \mathbb{R}^n, f_i(\mathbf{x}) \leq g_i(\mathbf{x}). \tag{A.6}$$

the set $\mathcal{F}(\mathbb{R}^n, \mathbb{R}^m)$ is a lattice where the meet and the join are defined by

$$\mathbf{f} \wedge \mathbf{g}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \wedge g_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \wedge g_m(\mathbf{x}) \end{pmatrix}, \tag{A.7}$$

and

$$\mathbf{f} \vee \mathbf{g}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \vee g_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \vee g_m(\mathbf{x}) \end{pmatrix}. \tag{A.8}$$

A *thick function* $[\mathbf{f}]$ from $\mathbb{R}^n$ to $\mathbb{IR}^m$ is an interval of $(\mathcal{F}(\mathbb{R}^n, \mathbb{R}^m), \leq)$. For such a thick function $[\mathbf{f}]$, there exist two functions $\mathbf{f}^-$ and, called the *lower bound* and the *upper bound* such that

$$\begin{aligned} [\mathbf{f}] &= [\mathbf{f}^-, \mathbf{f}^+] \\ &= \{\mathbf{f} \in \mathcal{F}(\mathbb{R}^n, \mathbb{R}^m) \mid \forall \mathbf{x} \in \mathbb{R}^n, \mathbf{f}^-(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}) \leq \mathbf{f}^+(\mathbf{x})\}. \end{aligned}$$

A thick function $[\mathbf{f}]$ is a sublattice of $(\mathcal{F}(\mathbb{R}^n, \mathbb{R}^m), \leq)$, *i.e.*, if $\mathbf{f} \in [\mathbf{f}]$, $\mathbf{g} \in [\mathbf{f}]$, then $\mathbf{f} \wedge \mathbf{g} \in [\mathbf{f}]$ and $\mathbf{f} \vee \mathbf{g} \in [\mathbf{f}]$. Again, if $\mathbf{f}^- = \mathbf{f}^+$, $[\mathbf{f}]$ is said to be *thin* and corresponds to a singleton of$\mathcal{F}(\mathbb{R}^n, \mathbb{R}^m)$, or equivalently to a classical function from $\mathbb{R}^n$ to $\mathbb{R}^m$.

Remark A.1. The class of thick functions is not so restrictive. For instance, all set-valued functions of the form

$$F(\mathbf{x}) = \{f(\mathbf{x}, \mathbf{a}) \in \mathbb{R}, \mathbf{a} \in [\mathbf{a}] \subset \mathbb{R}^m\}, \tag{A.9}$$

where $[\mathbf{a}]$ is a box and where f is continuous with respect to $\mathbf{a}$, are thick functions. If now the box $[\mathbf{a}]$ is replaced by a disconnected set or when the function f is not scalar anymore, the function $F(\mathbf{x})$ has no reason to be a thick function.

Thick set inversion problem

A thick set inversion problem can be written as

$$\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y}),\ \mathbf{f} \in [\mathbf{f}] \text{ and } \mathbb{Y} \in [\![\mathbb{Y}]\!] \tag{A.10}$$

where $[\![\mathbb{Y}]\!]$ is a thick set and $[\mathbf{f}]$ is a thick function.

The set $\mathbb{X}$ is said to be a *feasible solution* if

$$\exists \mathbf{f} \in [\mathbf{f}], \exists \mathbb{Y} \in [\![\mathbb{Y}]\!], \mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y}). \tag{A.11}$$

As explained in Section 5.4, when $[\mathbf{f}]$ can be parametric, projection algorithm can be used but required additional bisection along the domain of the parametric variable. Although generic, this method induces an extra computational cost and failed to handle efficiently high dimensional problems. To avoid this, additional notions such as thick intervals are introduced in the following section.

A.3 Thick Intervals

Denote by $\mathbb{IR}$ the set of all intervals of $\mathbb{R}$. A *thick interval* $[\![x]\!]$ (see, *e.g.*, [Chabert and Jaulin, 2009a]) is a subset of $\mathbb{IR}$ which can be written under the form

$$\begin{aligned} [\![x]\!] &= [\![[x^-]\,,[x^+]]\!] \\ &= \{[x^-,x^+] \in \mathbb{IR} \mid x^- \in [x^-] \text{ and } x^+ \in [x^+]\}\,. \end{aligned} \tag{A.12}$$

Here, $[x^-]\,,[x^+]$ are two intervals containing the *lower bound* $x^- \in \mathbb{R}$ and the *upper bound* $x^+ \in \mathbb{R}$ of an uncertain interval $[x^-,x^+]$. If we define the two intervals of $\mathbb{R}$

$$[x^{\subset}] = \bigcap \left\{[x^-,x^+] \in \mathbb{IR} \mid x^- \in [x^-], x^+ \in [x^+]\right\}, \tag{A.13}$$

and

$$[x^{\supset}] = \bigcup \left\{[x^-,x^+] \in \mathbb{IR} \mid x^- \in [x^-], x^+ \in [x^+]\right\}, \tag{A.14}$$

called the *subset bound* and the *supset bound* of $[\![x]\!]$ then

$$[\![x]\!] \subset \{[x] \in \mathbb{IR} \mid [x^{\subset}] \subset [x] \subset [x^{\supset}]\}\,, \tag{A.15}$$

with an equality if $[x^-] \cap [x^+] \neq \emptyset$. As a consequence, a thick interval is not necessarily a thick set: it is more precise or equivalently it is narrower. This can be explained using the *endpoints diagram* [Kulpa, 1994] (see Figure A.2) where an interval is seen as a point of $\mathbb{R}^2$. For instance, to the interval $[1,7]$, we associate the point with coordinates $(1,7)$. The degenerated intervals, such as $[2,2]$, all belong to the diagonal. This representation provides a geometrical representation of the relation between intervals. For instance, $[x] \subset [y]$ if $[y]$ is at the top left of $[x]$. The intersection between two intervals (or the interval hull) is obtained by taking the bottom-right corner (or the top left corner) of the smallest box which encloses the two interval points. For instance, if $[x] = [1,4]$ and $[y] = [2,5]$, the enclosing interval box is painted red. The top left interval is $[x] \cup [y] = [1,5]$ and the bottom-right interval is $[x] \cap [y] = [2,4]$. This red box corresponds to the thick box $[\![a]\!] = [\![[1,2],[4,5]]\!]$. The subset and supset bounds are $[a^{\subset}] = [2,4]$ and $[a^{\supset}] = [1,5]$.

In this figure, the orange polygon corresponds to the thick interval $[\![b]\!] = [\![[3,7],[6,8]]\!]$. The subset and supset bounds are $[b^-] = \emptyset$ and $[b^+] = [3,8]$. As illustrated by the gray zone of Figure A.2 (left), a subset-supset representation adds pessimism when

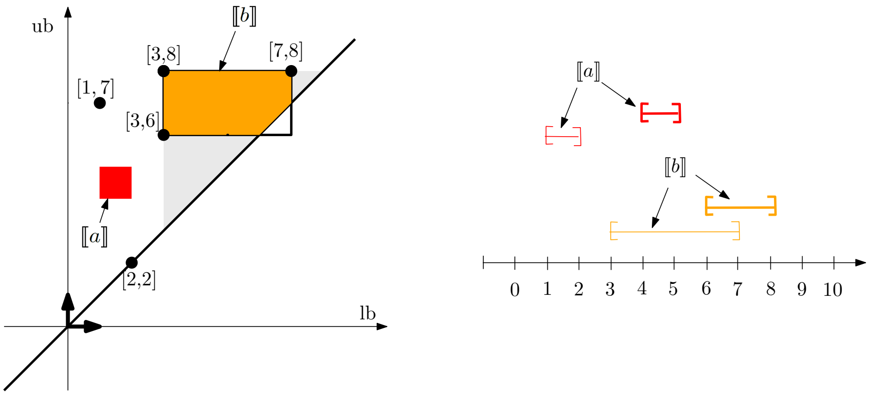


Figure A.2: In the endpoints diagram, an interval is represented by a point (here the small black disks)

the subset bound is empty, *i.e.*, a subset-supset representation may contain more intervals. For instance, if $[b] = [4,5]$, we have $\emptyset \subset [b] \subset [3,8]$, but $[b] \notin [\![[3,7],[6,8]]\!]$. The corresponding lower and upper interval bounds are represented on FigureA.2 (right). Note that when the subset bound is not empty (as for the thick box $[\![\mathbf{a}]\!]$ in red), no pessimism is added and both representations are equivalent.

Due to this pessimism, we will prefer to use a representation based on lower-upper bounds instead of the notation based on the subset-supset bounds. As already seen for thick sets (see (5.16)), set membership operations such as the union or the intersection can easily be extended to thick intervals. An extension for all classical operators of interval arithmetic is also valid. More precisely, if $\diamond \in \{+,-,\cdot,\cap,\sqcup,\dots\}$, we define

$$[\![x]\!] \diamond [\![y]\!] \quad = \quad \{[x] \diamond [y] \mid [x] \in [\![x]\!], [y] \in [\![y]\!]\}. \tag{A.16}$$

For instance, if $[\![a]\!] = [\![[1,2],[4,5]]\!]$ and $[\![b]\!] = [\![[3,7],[6,8]]\!]$, we have

$$\begin{aligned}
[\![a]\!] + [\![b]\!] &= [\![[1,2]+[3,7],[4,5]+[6,8]]\!] \\
&= [\![[4,9],[10,13]]\!] \\
[\![a]\!] \cap [\![b]\!] &= [\![\max([1,2],[3,7]),\min([4,5],[6,8])]\!] \\
&= [\![[3,7],[4,5]]\!] \\
[\![a]\!] \sqcup [\![b]\!] &= [\![\min([1,2],[3,7]),\max([4,5],[6,8])]\!] \\
&= [\![[1,2],[6,8]]\!] .
\end{aligned}$$

An interpretation for these formulas is the following: if $[a] \in [\![a]\!]$ and $[b] \in [\![b]\!]$ then the intervals $[x] = [a] + [b]$, $[y] = [a] \cap [b]$, $[z] = [a] \sqcup [b]$ satisfy $[x] \in [\![a]\!] + [\![b]\!]$,$[y] \in [\![a]\!] \cap [\![b]\!]$, $[z] \in [\![a]\!] \sqcup [\![b]\!]$.

The following proposition shows how to compare, from a practical point of view, two thick intervals.

Proposition A.1. Given two thick intervals $[\![a]\!]$ =$[\![[a^-],[a^+]]\!]$ and $[\![b]\!]$=$[\![[b^-],[b^+]]\!]$, we have

$$\begin{array}{lll}
(i) & ([\![a]\!] \subset [\![b]\!])^{\forall} \Leftrightarrow \left\{ \begin{array}{ll} [b]^- - [a]^- \subset \mathbb{R}^- & \wedge \\ [a]^+ - [b]^+ \subset \mathbb{R}^- & \end{array} \right. \\
(ii) & ([\![a]\!] \not\subset [\![b]\!])^{\forall} \Leftrightarrow \left\{ \begin{array}{ll} [a]^- - [b]^- \subset \mathbb{R}^- & \vee \\ [b]^+ - [a]^+ \subset \mathbb{R}^- & \end{array} \right. \\
(iii) & ([\![a]\!] \cap [\![b]\!] = \emptyset)^{\forall} \Leftrightarrow \left\{ \begin{array}{ll} [a]^+ - [b]^- \subset \mathbb{R}^- & \vee \\ [b]^+ - [a]^- \subset \mathbb{R}^- & \end{array} \right. \\
(iv) & ([\![a]\!] \cap [\![b]\!] \neq \emptyset)^{\forall} \Leftrightarrow \left\{ \begin{array}{ll} [b]^- - [a]^+ \subset \mathbb{R}^- & \wedge \\ [a]^- - [b]^+ \subset \mathbb{R}^- & \end{array} \right.
\end{array}$$

Proof. (i) Consider two intervals $[a]$ and $[b]$ of $\mathbb{R}$. (i) The inclusion $[a] \subset [b]$ is satisfied iff

$$b^- \leq a^- \text{ and } a^+ \leq b^+. \tag{A.17}$$

Thus, the inclusion is true for all $[a] \in [\![a]\!]$ and all $[b] \in [\![b]\!]$ iff (i) is satisfied.

(ii) We have $[a] \not\subset [b]$ iff $b^- > a^-$ or $a^+ > b^+$. Thus, the inclusion is unsatisfied for all $[a] \in [\![a]\!]$ and all $[b] \in [\![b]\!]$ iff (ii) is satisfied.

(iii) The two intervals $[a]$ and $[b]$ are disjoint iff $b^- > a^+$ or $a^- > b^+$. Therefore, they are disjoint for all $[a] \in [\![a]\!]$ and all $[b] \in [\![b]\!]$ iff (iii) is true.

(iv) The two intervals $[a]$ and $[b]$ overlap iff $b^- \leq a^+$ and $a^- \leq b^+$. Thus, they overlap for all $[a] \in [\![a]\!]$ and all $[b] \in [\![b]\!]$ iff (iv) is true. □

Example A.2. The two intervals $[a] = [1, 5]$ and $[b] \in [\![b]\!] = [\![[2, 4], [3, 6]]\!]$, depicted in Figure A.3, overlap for all feasible $[b]$, *i.e.*, $([a] \cap [\![b]\!] \neq \emptyset)^{\forall}$. This can be checked using Proposition A.1, iv:

$$[2, 4] - 5 \subset \mathbb{R}^- \text{ and } 1 - [3, 6] \subset \mathbb{R}^-. \tag{A.18}$$

Note that using the subset-supset bounds, we could not reach this conclusion. Indeed, the subset-supset approximation of $[\![b]\!]$ is

$$\emptyset \subset [b] \subset [2, 6]. \tag{A.19}$$

The interval $[b] = [6, 6]$ is consistent with this inclusion and does not intersect $[a]$. The green zone represents $[a] \cap [\![b]\!]$, the set of all intervals $[a] \cap [b]$ such that $[b] \in [\![b]\!]$.

Thick boxes. Denote by $\mathbb{IR}^n$ the set of all boxes of $\mathbb{R}^n$. A *thick box* $[\![\mathbf{x}]\!]$ is a set of boxes of $\mathbb{IR}^n$ which can be defined as

$$[\![\mathbf{x}]\!] = \left\{ \left[\mathbf{x}^-, \mathbf{x}^+\right] \in \mathbb{IR}^n \mid \mathbf{x}^- \in \left[\mathbf{x}^-\right], \mathbf{x}^+ \in \left[\mathbf{x}^+\right] \right\} \tag{A.20}$$

where $[\mathbf{x}^-]$ and $[\mathbf{x}^+]$ are boxes of $\mathbb{R}^n$. The set of thick boxes of $\mathbb{R}^n$ is denoted by $\mathbb{IIR}^n$. A thick box can be seen as an interval of boxes, *i.e.*, an interval of intervals of $\mathbb{R}^n$. This is illustrated by Figure A.4 which shows four thin boxes all contained

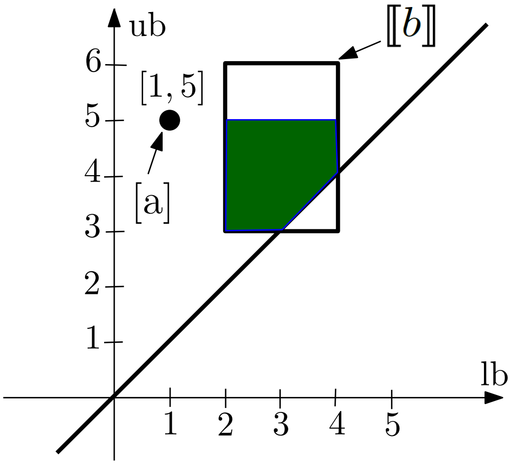


Figure A.3: A subset-supset approximation of $[\![b]\!]$ adds pessimism and thus fails to conclude that $[a]$ and $[b]$ always overlap. The green zone corresponds to the thick interval $[a] \cap [\![b]\!]$.

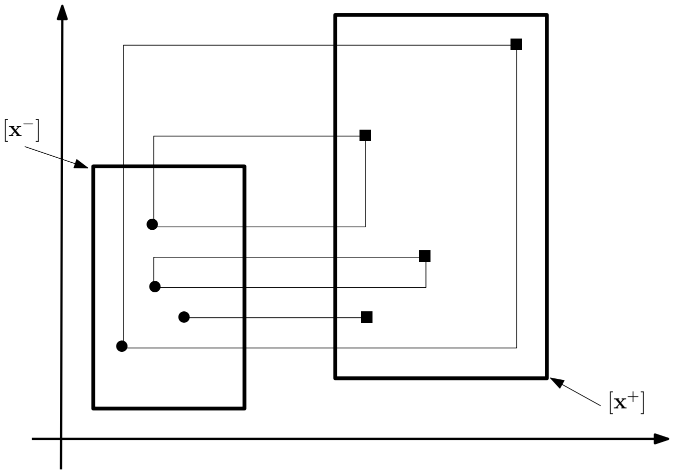


Figure A.4: The four boxes (thin) all belong to the thick box $[\![\mathbf{x}]\!] = [\![[\mathbf{x}^-], [\mathbf{x}^+]]\!]$

in the thick box $[\![\mathbf{x}]\!] = [\![[\mathbf{x}^-], [\mathbf{x}^+]]\!]$. Since the two box bounds of $[\mathbf{x}^-]$ and $[\mathbf{x}^+]$ are boxes of $\mathbb{R}^n$, we could decompose them as the Cartesian product of n intervals:

$$\begin{array}{rcl} [\mathbf{x}^-] & = & [x_1^-] \times \cdots \times [x_n^-] \\ [\mathbf{x}^+] & = & [x_1^+] \times \cdots \times [x_n^+]. \end{array} \tag{A.21}$$

We define the ith component $[\![x_i]\!]$ of the thick box $[\![[\mathbf{x}^-], [\mathbf{x}^+]]\!]$ as the thick interval $[\![x_i]\!] = \left[\!\!\left[\left[x_i^-\right], \left[x_i^+\right]\right]\!\!\right]$.

The following proposition will allow us to compare two thick boxes, with respect to the inclusion, from their interval components.

Proposition A.2. Given two thick boxes $[\![\mathbf{a}]\!]{=}[\![[\mathbf{a}^-], [\mathbf{a}^+]]\!]$ and $[\![\mathbf{b}]\!]{=}[\![[\mathbf{b}^-], [\mathbf{b}^+]]\!]$ of

$\mathbb{R}^n$, we have

$$\begin{array}{cc} (i) & (\llbracket \mathbf{a} \rrbracket \subset \llbracket \mathbf{b} \rrbracket)^{\forall} \Leftrightarrow \forall i \in \{1,\ldots,n\}, (\llbracket a_i \rrbracket \subset \llbracket b_i \rrbracket)^{\forall} \\ (ii) & (\llbracket \mathbf{a} \rrbracket \not\subset \llbracket \mathbf{b} \rrbracket)^{\forall} \Leftrightarrow \exists i \in \{1,\ldots,n\}, (\llbracket a_i \rrbracket \not\subset \llbracket b_i \rrbracket)^{\forall} \\ (iii) & (\llbracket \mathbf{a} \rrbracket \cap \llbracket \mathbf{b} \rrbracket = \emptyset)^{\forall} \Leftrightarrow \exists i \in \{1,\ldots,n\}, (\llbracket a_i \rrbracket \cap \llbracket b_i \rrbracket = \emptyset)^{\forall} \\ (iv) & (\llbracket \mathbf{a} \rrbracket \cap \llbracket \mathbf{b} \rrbracket \neq \emptyset)^{\forall} \Leftrightarrow \forall i \in \{1,\ldots,n\}, (\llbracket a_i \rrbracket \cap \llbracket b_i \rrbracket \neq \emptyset)^{\forall} \end{array} \tag{A.22}$$

Proof. This proof is a direct consequence of Proposition A.1 and of the fact that

$$\begin{array}{rcl} [\mathbf{a}] \subset [\mathbf{b}] & \Leftrightarrow & \forall i, [a_i]{\subset}[b_i] \\ [\mathbf{a}] \not\subset [\mathbf{b}] & \Leftrightarrow & \exists i, [a_i]{\not\subset}[b_i] \\ [\mathbf{a}] \cap [\mathbf{b}] = \emptyset & \Leftrightarrow & \exists i, [a_i]{\cap}[b_i] = \emptyset \\ [\mathbf{a}] \cap [\mathbf{b}] \neq \emptyset & \Leftrightarrow & \forall i, [a_i]{\cap}[b_i] \neq \emptyset.\blacksquare \end{array} \tag{A.23}$$

□

A.4 Thick Set Inversion

This section generalizes to set inversion algorithm [Jaulin and Walter, 1993c] to the thick case, as defined by Theorem 5.1.

A.4.1 Set inversion

Given a function $\mathbf{f}$ from $\mathbb{R}^n$ to $\mathbb{R}^m$ and a (thin) set $\mathbb{Y} \subset \mathbb{R}^n$, solving the set inversion problem, denoted by $\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$, is classically performed using an *inclusion function* $[\mathbf{f}] : \mathbb{IR}^n \to \mathbb{IR}^m$ of $\mathbf{f}$ [Moore, 1979], *i.e.*, an interval function such that

$$\mathbf{a} \in [\mathbf{x}] \Rightarrow \mathbf{f}(\mathbf{a}) \in [\mathbf{f}]([\mathbf{x}]). \tag{A.24}$$

Most algorithms for set-inversion decompose $\mathbb{R}^n$ into boxes [Moore, 1992][Jaulin and Walter, 1993c]. If a given box $[\mathbf{x}]$ satisfies $[\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y}$ then it is proved to be inside the solution set $\mathbb{X}$. If $[\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset$ then it is proved to be outside $\mathbb{X}$. If it satisfies none of the previous tests, it is bisected until it becomes too small. A possible implementation for set inversion is given by Algorithm A.1 which is called *Sivia* (Set Inversion Via Interval Analysis) [Jaulin and Walter, 1993c].

When the algorithm terminates, we have [Jaulin and Walter, 1993c]

$$\begin{array}{l} \bigcup \mathcal{L}^{clear} \subset \mathbb{X} \\ \bigcup \mathcal{L}^{dark} \cap \mathbb{X} = \emptyset. \end{array}$$

For the thick case, we have a thick function $[\mathbf{f}]$ from $\mathbb{R}^n$ to $\mathbb{IR}^m$ and a thick set $\llbracket \mathbb{Y} \rrbracket \in \mathbb{IP}(\mathbb{R}^n)$. We want to compute an approximation of the set of all sets

Algorithm A.1 Set inversion algorithm: *Sivia*

```
Input: $[\mathbf{x}]$, $\varepsilon$, $\mathbb{Y}$, $\llbracket \mathbf{f} \rrbracket$
Output: $\mathcal{L}^{clear}$, $\mathcal{L}^{dark}$
$\mathcal{L} = \{[\mathbf{x}]\}$, $\mathcal{L}^{clear} = \emptyset$, $\mathcal{L}^{dark} = \emptyset$
while $\mathcal{L} \neq \emptyset$ do
        unstack $\mathcal{L}$ into $[\mathbf{x}]$
    if $([\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y})$ then
        push $[\mathbf{x}]$ into $\mathcal{L}^{clear}$
    else if $([\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset)$ then
        push $[\mathbf{x}]$ into $\mathcal{L}^{dark}$
    else if width$([\mathbf{x}]) > \varepsilon$ then
        bisect $[\mathbf{x}]$ perpendicularly to its largest side and push the two resulting boxes in$\mathcal{L}$
    end if
end while
```

$\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$, assuming that $\mathbf{f} \in [\mathbf{f}]$ and $\mathbb{Y} \in \llbracket \mathbb{Y} \rrbracket$. This problem, formalized by Theorem 5.1, is called a *thick set inversion problem*, denoted by

$$\llbracket \mathbb{X} \rrbracket = [\mathbf{f}]^{-1} \left(\llbracket \mathbb{Y} \rrbracket \right). \tag{A.25}$$

We propose to compute an approximation of $\llbracket \mathbb{X} \rrbracket$ by decomposing $\mathbb{R}^n$ into three subsets: the clear zone $\mathbb{X}^-$, the penumbra $\mathbb{X}^+ \backslash \mathbb{X}^-$ and the dark zone $\mathbb{R}^n \backslash \mathbb{X}^+$. In our approach, a paver performs the decomposition of $\mathbb{R}^n$ into boxes and a thick extension of an inclusion function is used to classify boxes.

A.4.2 Thick inclusion function

The function $\llbracket \mathbf{f} \rrbracket : \mathbb{IR}^n \to \mathbb{IIR}^m$ is a *thick inclusion function* of the thick function $[\mathbf{f}] : \mathbb{R}^n \to \mathbb{IR}^m$ if

$$\mathbf{a} \in [\mathbf{x}] \Rightarrow [\mathbf{f}] (\mathbf{a}) \in \llbracket \mathbf{f} \rrbracket ([\mathbf{x}]) \,. \tag{A.26}$$

Theorem A.1. *Consider a thick function* $[\mathbf{f}] (\mathbf{x})$ *and denote by* $[\mathbf{f}^-], [\mathbf{f}^+]$ *two inclusion functions for the bounds* $\mathbf{f}^-, \mathbf{f}^+$ *of* $[\mathbf{f}]$*. The function* $\llbracket \mathbf{f} \rrbracket : \mathbb{IR}^n \to \mathbb{IIR}^m$ *defined by*

$$\llbracket \mathbf{f} \rrbracket ([\mathbf{x}]) = \left\llbracket \left[\mathbf{f}^- \right] ([\mathbf{x}]) \,, \left[\mathbf{f}^+ \right] ([\mathbf{x}]) \right\rrbracket \,, \tag{A.27}$$

is a thick inclusion function for $[\mathbf{f}] (\mathbf{x})$.

Proof. Since $[\mathbf{f}^-], [\mathbf{f}^+]$ are two inclusion functions for $\mathbf{f}^-, \mathbf{f}^+$, we have

$$\mathbf{a} \in [\mathbf{x}] \Rightarrow \begin{cases} \mathbf{f}^- (\mathbf{a}) & \in & [\mathbf{f}^-] ([\mathbf{x}]) \\ \mathbf{f}^+ (\mathbf{a}) & \in & [\mathbf{f}^+] ([\mathbf{x}]) \end{cases} \tag{A.28}$$

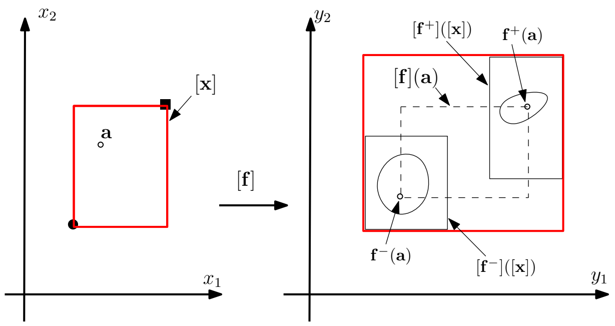

x_2
$[\mathbf{x}]$
$\mathbf{a}$
x_1
$[\mathbf{f}]$
y_2
$[\mathbf{f}^+]([\mathbf{x}])$
$\mathbf{f}^+(\mathbf{a})$
$[\mathbf{f}](\mathbf{a})$
$\mathbf{f}^-(\mathbf{a})$
$[\mathbf{f}^-]([\mathbf{x}])$
y_1

Figure A.5: The thick inclusion function $[\![\mathbf{f}]\!]([\mathbf{x}])$ encloses all boxes $[\mathbf{f}](\mathbf{a})$ where $\mathbf{a} \in [\mathbf{x}]$.

Now, the right-hand side is equivalent to $[\mathbf{f}](\mathbf{a}) \in [\![\mathbf{f}]\!]([\mathbf{x}])$. ■ □

As illustrated by Figure A.5, the thick box $[\![\mathbf{f}]\!]([\mathbf{x}])$ encloses the set of all boxes $[\mathbf{f}](\mathbf{a})$ with $\mathbf{a} \in [\mathbf{x}]$. The vector $\mathbf{a} \in [\mathbf{x}] \subset \mathbb{R}^2$ has an image $[\mathbf{f}](\mathbf{a})$ which is a box of $\mathbb{R}^2$ with a lower bound $\mathbf{f}^-(\mathbf{a})$ and an upper bound $\mathbf{f}^+(\mathbf{a})$. Using a classical interval arithmetic, we are able to get inclusion functions $[\mathbf{f}^-]$ and $[\mathbf{f}^+]$ for $\mathbf{f}^-$ and $\mathbf{f}^+$. The boxes $[\mathbf{f}^-]([\mathbf{x}])$ and $[\mathbf{f}^+]([\mathbf{x}])$ contain $\mathbf{f}^-(\mathbf{a})$ and $\mathbf{f}^+(\mathbf{a})$. Therefore, the box $[\mathbf{f}](\mathbf{a})$ is inside the thick box $[\![\mathbf{f}]\!]([\mathbf{x}]) = [\![[\mathbf{f}^-]([\mathbf{x}]), [\mathbf{f}^+]([\mathbf{x}])]\!]$.

A.4.3 Algorithm

Algorithm A.2, named *ThickSivia* (Thick Set Inversion Via Interval Analysis), provides an approximation of the solution of the thick set inversion problem $[\![\mathbb{X}]\!] = [\mathbf{f}]^{-1}([\![\mathbb{Y}]\!])$. The input of this algorithm are (1) the box $[\mathbf{x}]$ which is assumed to be large enough to contains $\mathbb{X}^+$, the upper bound of $[\![\mathbb{X}]\!]$, (2) an accuracy $\varepsilon > 0$, (3) the thick inclusion function $[\![\mathbf{f}]\!]$, and (4) thick set $[\![\mathbb{Y}]\!]$. The output is an approximation of the thick set $[\![\mathbb{X}]\!] = [\![\mathbb{X}^-, \mathbb{X}^+]\!]$. The algorithm decomposes the initial box $[\mathbf{x}]$ into four non-overlapping subpavings: (1) The inner subpaving $\mathcal{L}^{clear}$ which contains boxes that have been proved to be inside the clear zone $\mathbb{X}^-$, (2) the outer subpaving $\mathcal{L}^{dark}$ which contains boxes that have been proved to be outside $\mathbb{X}^+$ (*i.e.*, inside the dark zone), (3) the subpaving $\mathcal{L}^{penumbra}$ which contains boxes that have been proved to be inside the penumbra $\mathbb{X}^+ \backslash \mathbb{X}^-$ and (4) the subpaving made with boxes that have been rejected (for which nothing is known and with a width smaller than the desired level of precision ε).

Remark A.2. In the algorithm, we have several tests on thick boxes that require comparisons on thick boxes as introduced in Section A.3. Consider for instance the test $([\![\mathbf{f}]\!]([\mathbf{x}]) \cap \mathbb{Y}^+ = \emptyset)^{\forall}$ and assume that $\mathbb{Y}^+$ is made with boxes $\{[\mathbf{y}](1), \dots, [\mathbf{y}](\bar{k})\}$ (this will be the case in the applications presented in Section A.5). Since we have

$$\left([\![\mathbf{f}]\!]([\mathbf{x}]) \cap \mathbb{Y}^+ = \emptyset\right)^{\forall} \Leftrightarrow \forall k, ([\![\mathbf{f}]\!]([\mathbf{x}]) \cap [\mathbf{y}](k) = \emptyset)^{\forall}.$$

Algorithm A.2 Thick set inversion algorithm: *ThickSivia*

```
Input: $[\mathbf{x}]$, $\varepsilon$, $[\![\mathbb{Y}]\!]$, $[\![\mathbf{f}]\!]$
Output: $\mathcal{L}^{clear}$, $\mathcal{L}^{penumbra}$, $\mathcal{L}^{dark}$
$\mathcal{L} = \{[\mathbf{x}]\}$, $\mathcal{L}^{clear} = \emptyset$, $\mathcal{L}^{penumbra} = \emptyset$, $\mathcal{L}^{dark} = \emptyset$
while $\mathcal{L} \neq \emptyset$ do
    Unstack $\mathcal{L}$ into $[\mathbf{x}]$
    if $([\![\mathbf{f}]\!]([\mathbf{x}]) \subset \mathbb{Y}^-)^\forall$ then
        push $[\mathbf{x}]$ into $\mathcal{L}^{clear}$
    else if $([\![\mathbf{f}]\!]([\mathbf{x}]) \cap \mathbb{Y}^+ = \emptyset)^\forall$ then
        push $[\mathbf{x}]$ into $\mathcal{L}^{dark}$
    else if $([\![\mathbf{f}]\!]([\mathbf{x}]) \cap (\mathbb{Y}^+ \backslash \mathbb{Y}^-) \neq \emptyset)^\forall$ then
        push $[\mathbf{x}]$ into $\mathcal{L}^{penumbra}$
    else if width$([\mathbf{x}]) > \varepsilon$ then
        bisect $[\mathbf{x}]$ perpendicularly to its largest side and push the two resulting boxes in $\mathcal{L}$
    end if
end while
```

Our test amounts to checking that $([\![\mathbf{a}]\!] \cap [\mathbf{b}] = \emptyset)^\forall$ where $[\![\mathbf{a}]\!] = [\![\mathbf{f}]\!]([\mathbf{x}])$ and $[\mathbf{b}] = [\mathbf{y}](k)$. Therefore,

$$\begin{array}{lll} & ([\![\mathbf{a}]\!] \cap [\mathbf{b}] = \emptyset)^\forall & \\ \Leftrightarrow & \exists i \in \{1,\ldots,n\}, ([\![a_i]\!] \cap [b_i] = \emptyset)^\forall & \text{(see Proposition A.2)} \\ \Leftrightarrow & \exists i \in \{1,\ldots,n\}, b_i^- - [a_i]^+ \subset \mathbb{R}^- \wedge [a_i]^- - b_i^+ \subset \mathbb{R}^- & \text{(see Proposition A.1)} \\ \Leftrightarrow & \exists i \in \{1,\ldots,n\},\ b_i^- - lb\left([a_i]^+\right) \leq 0 \wedge (ub\,([a_i]^-)) - b_i^+ \leq 0 & \end{array}$$

where $lb([a_i]^+)$ is the lower bound of the interval $[a_i]^+$ and $ub([a_i]^-)$ is the upper bound of $[a_i]^-$. We thus get Algorithm A.3. The same type of algorithm applies to test if $([\![\mathbf{f}]\!]([\mathbf{x}]) \subset \mathbb{Y}^-)^\forall$ or $([\![\mathbf{f}]\!]([\mathbf{x}]) \cap (\mathbb{Y}^+ \backslash \mathbb{Y}^-) \neq \emptyset)^\forall$.

Algorithm A.3 Test if $([\![\mathbf{f}]\!]([\mathbf{x}]) \cap \mathbb{Y}^+ = \emptyset)^\forall$

```
Input: $[\![\mathbf{a}]\!] = [\![\mathbf{f}]\!]([\mathbf{x}]), \mathbb{Y}^+ = \{[\mathbf{y}](1),\ldots,[\mathbf{y}](\bar{k})\}$
for $k = 1$ to $\bar{k}$ do
    $[\mathbf{b}] = [\mathbf{y}](k)$
    for $i = 1$ to $n$ do
        If $b_i^- - lb\left([a_i]^+\right) > 0 \;\vee\; (ub\,([a_i]^-)) - b_i^+ > 0$ then Return False. End.
    end for
end for
Return True
```

i	$1-\frac{1}{i}$
1	0
2	$\frac{1}{2}$
3	$\frac{2}{3}$
∞	1

Table A.1: results of Formula A.31 for different values of i.

A.4.4 Properties

Termination

The algorithm always terminates in less than $\lambda = \left(\frac{\text{with}([\mathbf{x}])}{\varepsilon}\right)^{dim(\mathbf{x})}$ iterations, where $[\mathbf{x}]$ it the input box. The number λ is huge and corresponds to the worst case situation where all tests fail and the algorithm returns $\mathbb{X}^{clear} = \emptyset$, $\mathbb{X}^{penumbra} = \emptyset$, $\mathbb{X}^{dark} = \emptyset$. In practice, as shown in [Jaulin and Walter, 1993b] the number of iterations is $O\left(\mathcal{A}\left(\frac{1}{\varepsilon}\right)^{dim(\mathbf{x})-1}\right)$, where $\mathcal{A}$ is the area of the accumulation zone which is composed here with the boundary of the penumbra.

Now, we can find some examples where the number of iterations is larger than $O\left(\left(\frac{1}{\varepsilon}\right)^{dim(\mathbf{x})-1}\right)$. Consider for instance the problem of solving $f(x) < 0$ where $f(x) = x^i$. Take an inclusion function: $[f]([x]) = [x]^i + [x] - [x]$. Assume that $[x] = [a, b]$ where $a > 0$. We have:

$$[f]\left([a,b]\right) = \left[a^i, b^i\right] + [a-b, b-a] = \left[a^i + a - b, b^i + b - a\right] \tag{A.29}$$

The interval is not classified by our algorithm if $a^i + a - b < 0$, *i.e.*, if $a^i < w([a,b])$. Since in our algorithm, for all interval $[a, b]$, we have $\varepsilon < w\left([a,b]\right)$ (otherwise, the interval is eliminated), we conclude that if $a^i < \varepsilon$, or equivalently, $a < \varepsilon^{\frac{1}{i}}$ the box cannot be eliminated. Thus, the number of remaining yellow boxes η is of order $\varepsilon^{\frac{1}{i}}.\varepsilon^{-1} = \varepsilon^{\frac{1}{i}-1}$. Therefore,

$$\frac{\log \eta}{\log \varepsilon^{\frac{1}{i}-1}} = \frac{\log \eta}{\left(\frac{1}{i}-1\right)\log \varepsilon} \tag{A.30}$$

should converge to 1 or equivalently

$$\lim_{\varepsilon \to 0} \frac{\log \eta}{\log \varepsilon^{-1}} = 1 - \frac{1}{i}. \tag{A.31}$$

The Table A.1 illustrates this formula for different values of i.

We tried to get an experimental validation of this formula, and we obtained the following figure which is consistent with the convergence formula. Now, even for tiny ε, $\log\varepsilon^{-1}$ is never large if we want the results in a reasonable time. It is thus difficult to get a complete experimental validation of our formula.

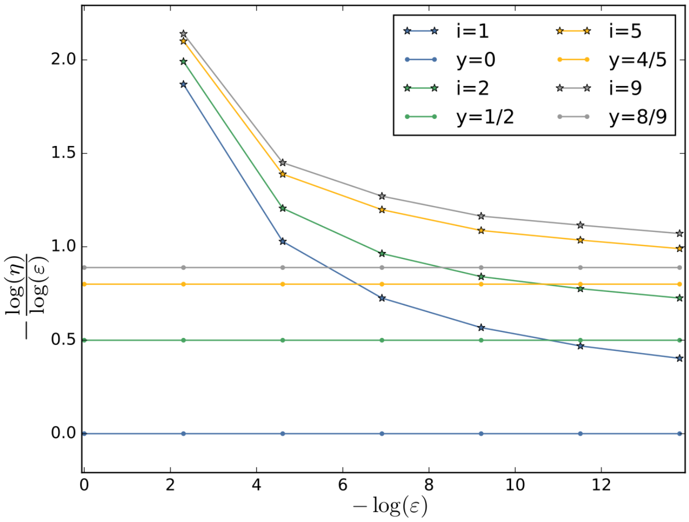


Figure A.6: Experimental validation of the number of generated boxes η with respect to ε. This figure draws $\log\eta$ with respect tolog ε^{-1} for different values of i.

Enclosure

The algorithm computes guaranteed inner and outer approximations of the solution set of the thick set inversion problem. This is asserted by the following theorem.

Theorem A.2. *The algorithm returns an approximation of the thick set inversion problem* $[\![\mathbb{X}^-,\mathbb{X}^+]\!] = [\mathbf{f}]^{-1}([\![\mathbb{Y}]\!])$ *under the form of 3 subpavings (i.e., union of boxes):* $\mathcal{L}^{clear}$, $\mathcal{L}^{penumbra}$, $\mathcal{L}^{dark}$. *This approximation satisfies*

$$\begin{array}{ll} (i) & \bigcup\mathcal{L}^{clear} \subset \mathbb{X}^- \\ (ii) & \bigcup\mathcal{L}^{penumbra} \subset \mathbb{X}^+\backslash\mathbb{X}^- \\ (iii) & \bigcup\mathcal{L}^{dark} \cap \mathbb{X}^+ = \emptyset. \end{array}$$

Proof. To prove the Theorem, we need to show that, for a box $[\mathbf{x}]$, we have (see Figure A.7):

$$\begin{array}{cccc} (i) & \left([\![\mathbf{f}]\!]([\mathbf{x}]) \subset \mathbb{Y}^-\right)^{\forall} & \Rightarrow & [\mathbf{x}] \subset \mathbb{X}^- \\ (ii) & \left([\![\mathbf{f}]\!]([\mathbf{x}]) \cap \mathbb{Y}^+ = \emptyset\right)^{\forall} & \Rightarrow & [\mathbf{x}] \cap \mathbb{X}^+ = \emptyset \\ (iii) & \left([\![\mathbf{f}]\!]([\mathbf{x}]) \cap (\mathbb{Y}^+\backslash\mathbb{Y}^-) \neq \emptyset\right)^{\forall} & \Rightarrow & [\mathbf{x}] \subset \mathbb{X}^+\backslash\mathbb{X}^-. \end{array} \tag{A.32}$$

Let us first prove (i). The left-hand side of (i) implies that

$$\forall\, [\mathbf{a}] \in [\![\mathbf{f}]\!]\,([\mathbf{x}]),\ [\mathbf{a}] \subset \mathbb{Y}^{-}. \tag{A.33}$$

Take $\mathbf{x} \in [\mathbf{x}]$, and we show that $\mathbf{x} \in \mathbb{X}^{-}$. Since $\mathbf{x} \in [\mathbf{x}]$, we have $[\![\mathbf{f}]\!]\,(\mathbf{x}) \subset [\![\mathbf{f}]\!]\,([\mathbf{x}])$ and the previous formula implies.

$$\forall\, [\mathbf{a}] \in [\![\mathbf{f}]\!]\,(\mathbf{x}),\ [\mathbf{a}] \subset \mathbb{Y}^{-}. \tag{A.34}$$

Now, $[\![\mathbf{f}]\!]\,(\mathbf{x})$ is a singleton in $\mathbb{IR}^n$ which contains the single box $[\mathbf{f}]\,(\mathbf{x})$. Thus, (A.34) becomes $[\mathbf{f}]\,(\mathbf{x}) \subset \mathbb{Y}^{-}$, which implies

$$\forall \mathbf{f} \in [\mathbf{f}]\,,\ \mathbf{f}(\mathbf{x}) \in \mathbb{Y}^{-} \tag{A.35}$$

or equivalently $\forall \mathbf{f} \in [\mathbf{f}]\,,\ \mathbf{x} \in \mathbf{f}^{-1}\,(\mathbb{Y}^{-})$. We get

$$\mathbf{x} \in \bigcap_{\mathbf{f}\in[\mathbf{f}]} \mathbf{f}^{-1}(\mathbb{Y}^{-}) \overset{(5.32)}{=} \mathbb{X}^{-}. \tag{A.36}$$

The same reasoning applies to prove (ii). For (iii), assume that the left-hand side of (iii) is satisfied. Take one $\mathbf{x} \in [\mathbf{x}]$, the quantity $[\![\mathbf{f}]\!]\,([\mathbf{x}])$ becomes a singleton in $\mathbb{IR}^n$, *i.e.*, a box of $\mathbb{R}^n$. We have

$$\begin{array}{ll} & ([\mathbf{f}]\,(\mathbf{x}) \cap (\mathbb{Y}^{+}\backslash\mathbb{Y}^{-}) \neq \emptyset) \\ \Leftrightarrow & \exists \mathbf{f} \in [\mathbf{f}]\,, \mathbf{x} \in \mathbf{f}^{-1}(\mathbb{Y}^{+}\backslash\mathbb{Y}^{-}) \\ \Leftrightarrow & \exists \mathbf{f} \in [\mathbf{f}]\,, \mathbf{x} \in \mathbf{f}^{-1}(\mathbb{Y}^{+}) \wedge \mathbf{x} \notin \mathbf{f}^{-1}(\mathbb{Y}^{-}) \\ \Leftrightarrow & \mathbf{x} \notin \bigcap_{\mathbf{f}\in[\mathbf{f}]} \mathbf{f}^{-1}(\mathbb{Y}^{-}) \ \wedge\ \mathbf{x} \in \bigcup_{\mathbf{f}\in[\mathbf{f}]} \mathbf{f}^{-1}(\mathbb{Y}^{+}). \end{array}$$

Thus, from (5.32), we get $[\mathbf{x}] \subset \mathbb{X}^{+}\backslash\mathbb{X}^{-}$. ■ □

Convergence

We now provide some convergence properties of our algorithm. We need first to define the convergence of a thick inclusion function $[\![\mathbf{f}]\!]$. This convergence can be interpreted as the continuity of the thick function $[\![\mathbf{f}]\!]\,([\mathbf{x}])$ around intervals $[\mathbf{x}]$ which are degenerated (i.e., the box $[\mathbf{x}]$ is a singleton).

Definition A.1. The thick inclusion function $[\![\mathbf{f}]\!]\,([\mathbf{x}])$ for $[\mathbf{f}]([\mathbf{x}])$ is said to be *convergent* if for all $\mathbf{a} \in \mathbb{R}^n$, for all sequences of boxes $[\mathbf{x}]\,(k)$ and $[\mathbf{y}]\,(k)$, we have

$$\left.\begin{array}{r} d_H([\mathbf{x}](k), \{\mathbf{a}\}) \overset{k\to\infty}{\longrightarrow} 0 \\ [\mathbf{y}](k) \in [\![\mathbf{f}]\!]\,([\mathbf{x}]\,(k)) \end{array}\right\} \Rightarrow d_H([\mathbf{y}](k), [\mathbf{f}](\mathbf{a})) \overset{k\to\infty}{\longrightarrow} 0,$$

where d_H is the Hausforff distance between compact sets [Berge, 1963].

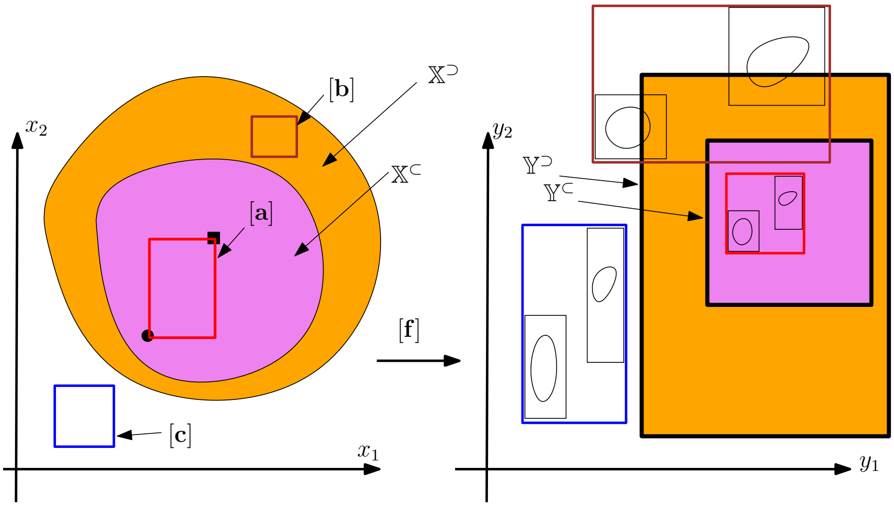


Figure A.7: Tests used for the thick set inversion. The box [**a**] is proved to be inside the clear zone $\mathbb{X}^-$; The box [**b**] is proved to be inside the penumbra $\mathbb{X}^+ \backslash \mathbb{X}^-$. The box [**c**] is proved to be inside the dark zone, i.e., outside $\mathbb{X}^+$.

Theorem A.3. *For a given ε, our algorithm provides three lists $\mathcal{L}^{clear}(\varepsilon)$,$\mathcal{L}^{penumbra}(\varepsilon)$ and $\mathcal{L}^{dark}(\varepsilon)$. Take a point* $\mathbf{a}$. *For ε sufficiently small we have*

$$\begin{array}{lccc} (i) & [\mathbf{f}](\mathbf{a}) \subset int\,(\mathbb{Y}^-) & \Rightarrow & \mathbf{a} \in \bigcup \mathcal{L}^{clear} \\ (ii) & [\mathbf{f}](\mathbf{a}) \subset int\left(\overline{\mathbb{Y}^+}\right) & \Rightarrow & \mathbf{a} \in \bigcup \mathcal{L}^{dark} \\ (iii) & [\mathbf{f}](\mathbf{a}) \cap int\,(\mathbb{Y}^+ \backslash \mathbb{Y}^-) \neq \emptyset & \Rightarrow & \mathbf{a} \in \bigcup \mathcal{L}^{penumbra} \end{array} \tag{A.37}$$

where $int(\mathbb{A})$ denotes the interior of the set $\mathbb{A}$ [Berge, 1963].

Proof. The proof is by contradiction. Assume that for all ε the box containing $\mathbf{a}$ is never classified. It means that there exists a sequence of boxes $[\mathbf{x}](k)$ converging to $\mathbf{a}$ such that none of the three tests is satisfied for all $[\mathbf{x}](k)$.
(i) Since for all k, $(\llbracket \mathbf{f} \rrbracket ([\mathbf{x}](k)) \subset \mathbb{Y}^-)^{\forall}$ is false, there exists a sequence $[\mathbf{y}](k) \in \llbracket \mathbf{f} \rrbracket ([\mathbf{x}](k))$ such that $([\mathbf{y}](k) \subset \mathbb{Y}^-)$ is false. Now, since $\llbracket \mathbf{f} \rrbracket$ is a convergent thick inclusion function for $[\mathbf{f}]$, $d_H([\mathbf{y}](k), [\mathbf{f}](\mathbf{a})) \to 0$. Since $int\,(\mathbb{Y}^-)$ is an open set, we cannot have $[\mathbf{f}](\mathbf{a}) \subset int\,(\mathbb{Y}^-)$.
(ii) Since for all k, $(\llbracket \mathbf{f} \rrbracket ([\mathbf{x}])(k) \cap \mathbb{Y}^+ = \emptyset)^{\forall}$ is false, using the same reasoning as for (i), we get that we cannot have $[\mathbf{f}](\mathbf{a}) \subset int\left(\overline{\mathbb{Y}^+}\right)$.
(iii) Since for all k,$(\llbracket \mathbf{f} \rrbracket ([\mathbf{x}])(k) \cap (\mathbb{Y}^+ \backslash \mathbb{Y}^-) \neq \emptyset)^{\forall}$ is false, again, we conclude that we cannot have $([\mathbf{f}](\mathbf{a}) \cap int\,(\mathbb{Y}^+ \backslash \mathbb{Y}^-) \neq \emptyset)$.
As a consequence, we get that if either (i), (ii) or (iii) is satisfied then $\mathbf{a}$ will be classified inside one of the three lists. Moreover, from Equation (A.32), we get that $\mathbf{a}$ will be classified on the right list.■ □

Remark A.3. For any box $[\mathbf{y}]$, we always have $[\mathbf{y}] \subset \mathbb{Y}^-$ or $[\mathbf{y}] \subset \overline{\mathbb{Y}^+}$ or $[\mathbf{y}] \cap (\mathbb{Y}^+ \backslash \mathbb{Y}^-) \neq \emptyset$. Moreover, in a generic situation, we have $[\mathbf{y}] \subset int\,(\mathbb{Y}^-)$ or $[\mathbf{y}] \subset$

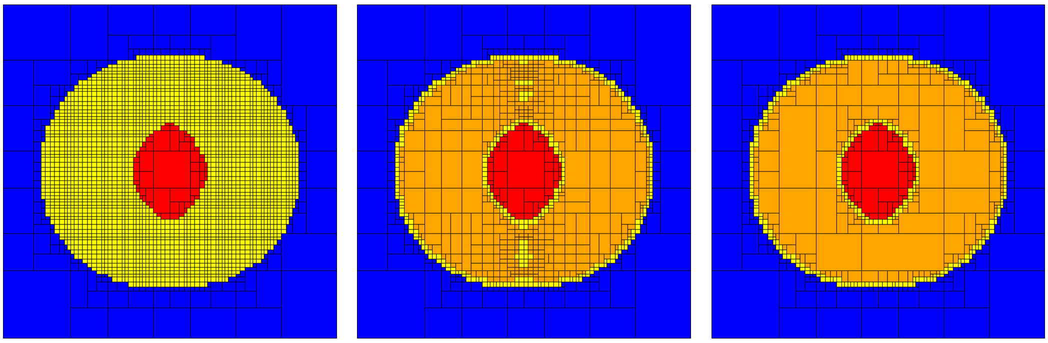


Figure A.8: Left: With classical intervals; Center: with subset-supset based thick intervals; Right: with lower-upper bounds based thick intervals

$int\left(\overline{\mathbb{Y}^+}\right)$ or $[\mathbf{y}] \cap int\left(\mathbb{Y}^+ \backslash \mathbb{Y}^-\right) \neq \emptyset$. Therefore, Theorem A.3 tells us that the part of the search space which will not be classified are rare. Except in atypical situations, these regions will correspond to the boundaries of the penumbra $\mathbb{X}^+ \backslash \mathbb{X}^-$.

A.5 Test-Cases

This section provides five test-cases to illustrate the efficiency of our method. All these test cases solve a thick inversion problem $[\![\mathbb{X}]\!] = [\mathbf{f}]^{-1}\left([\![\mathbb{Y}]\!]\right)$. In the figures, all red boxes are shown to be inside the clear zone $\mathbb{X}^-$; all blue boxes are inside the dark zone, *i.e.*, outside $\mathbb{X}^+$; all orange boxes are proved to be inside the penumbra. Nothing is known for the small yellow boxes.

Test-case 1. *Thick translation.* Consider the thick set $[\![\mathbb{Y}]\!] = [\mathbb{Y}^-, \mathbb{Y}^+]$ where $\mathbb{Y}^-, \mathbb{Y}^+$ are two disks with center $(0, 0)$ and with radius $r^- = 1$ and $r^+ = 2$, respectively. Consider the thick function $[\mathbf{f}](\mathbf{x}) = \mathbf{x} - [\mathbf{v}]$ where $[\mathbf{v}] = [0.7, 1.3] \times [-0.02, 0.02]$. Figure A.8 represents an approximation of $[\![\mathbb{X}]\!] = [\mathbf{f}]^{-1}\left([\![\mathbb{Y}]\!]\right)$ using three types of intervals: the classical intervals (left), the thick intervals with a subset-supset representation (center) and the thick intervals defined by lower-upper interval bounds (right). These results have been obtained with $\varepsilon = 0.1$ and the frame box corresponds to $[-2, 4] \times [-2, 4]$. These figures have been generated in 2.1sec for the left figure; 0.21sec for the centered figure and 0.19 for the right figure. As we can see on this figure, the penumbra is better (*i.e.*, without any uncertain boxes inside) characterized with a lower/upper bound representation for the thick intervals.

We compared the computing time (on a processor i5-2520M@2.50GHz) and the number of bisections with the traditional approach (which does not characterize the penumbra) and our method. We get the table below. We observe that when ε is small, classical methods are much less efficient due to the fact many bisections take place inside the penumbra. This observation is also valid for all five test-cases considered in this paper.

	Classical method		Our method	
ε	#bisection	time(s)	#bisection	time(s)
0.5	195	0.0003	159	0.0012
0.1	2215	0.0031	783	0.0047
0.05	8043	0.0130	1623	0.0117
0.01	470407	0.6701	13383	0.1779
0.005	1867707	2.6515	26823	0.5634
0.001	29722487	42.0044	107443	6.1724

Remark A.4. With respect to notation introduced in Chapter 4, the thick translation can be expressed in term of Minkowsky sum and difference. From the notations used in this test case we have:

$$\begin{aligned} [\mathbf{f}]^{-1}\left(\llbracket \mathbb{Y} \rrbracket\right) &= \llbracket \mathbb{Y} \rrbracket \oplus [\mathbf{v}] && \text{(A.38)} \\ &= \left\llbracket \mathbb{Y}^{-} \ominus [\mathbf{v}], \mathbb{Y}^{+} \oplus [\mathbf{v}] \right\rrbracket . && \text{(A.39)} \end{aligned}$$

To make the link with morphological operations, the lower set $\mathbb{X}^{-}$ corresponds to an erosion and $\mathbb{X}^{+}$ to a dilation.

Test-case 2. *Tolerable-United solution sets.* Consider the interval linear system [Kreinovich and Shary, 2016]

$$\begin{pmatrix} [2,4] & [-2,0] \\ [-1,1] & [2,4] \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \begin{pmatrix} [-1,1] \\ [0,2] \end{pmatrix}. \tag{A.40}$$

The left-hand side corresponds to a thick function and the right-hand side corresponds to a thin set. The solution set $\llbracket \mathbb{X} \rrbracket = \llbracket \mathbb{X}^{-}, \mathbb{X}^{+} \rrbracket$ has for subset bound the *tolerable solution set* $\mathbb{X}^{-}$ for supset bound the *united solution set* $\mathbb{X}^{+}$ [Goldsztejn and Chabert, 2006] . Some techniques have been developed to approximate these sets [Shary, 1995] in the linear case. They are mainly based on the Kaucher interval arithmetic [Kaucher, 1980, Goldsztejn, 2005] and may be used to find boxes inside the penumbra. Now, these methods have mainly been developed to deal with linear interval problems and cannot be used to find boxes inside the penumbra for general nonlinear problems as for the two following test-cases. For this example, the thick set-inversion algorithm provides the paving of Figure A.9, in less than 0.4 sec, for $\varepsilon = 0.01$.

Test-case3. *Parameter estimation.*Consider the parametric model

$$y_m(\mathbf{x}, t) = x_1 e^{-x_2 t}, \tag{A.41}$$

where $\mathbf{x} = (x_1, x_2)$ is the parameter vector and $t \in \mathbb{R}$ is the time. At time t_i, we collect measurements y_i with some interval uncertainties as written in Table A.2. Note that one of the main difficulties of this problem is that uncertainties exist on the independent variable (here the time) [Jaulin and Walter, 1999, Cerone, 1991]. In our formulation, the uncertainty of the t_i is stored inside the model under the

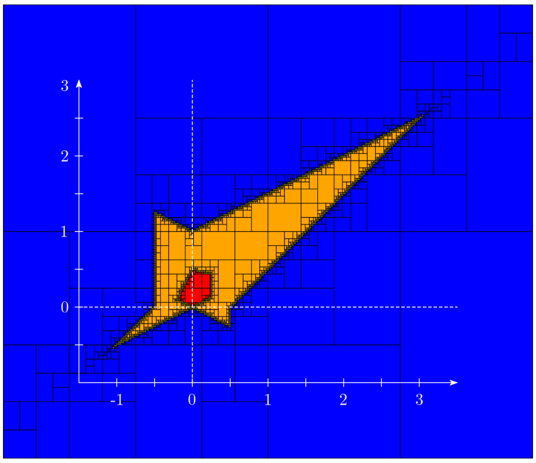


Figure A.9: Approximation of the tolerable-united solution sets of the interval linear system of Test-case 2

form of a thick function.

i	$[t_i]$	$[y_i]$
1	$[0.03, 0.06]$	$[4, 8]$
2	$[0.09, 0.12]$	$[2, 6]$
3	$[0.15, 0.18]$	$[2, 5]$
4	$[0.21, 0.24]$	$[1, 3]$
5	$[0.27, 0.3]$	$[0, 2]$

Table A.2: Measurements (t_i, y_i) used for estimation

The set of all feasible parameter vectors is

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^2 \mid \forall i \in \{1, \dots, 5\}, \exists t \in [t_i], x_1 e^{-x_2 t} \in [y_i]\}. \tag{A.42}$$

If we define the thick function

$$[\mathbf{f}](\mathbf{x}) = \begin{pmatrix} x_1 e^{-x_2 [t_1]} \\ \vdots \\ x_1 e^{-x_2 [t_5]} \end{pmatrix} \tag{A.43}$$

and the box

$$[\mathbf{y}] = [y_1] \times \cdots \times [y_5], \tag{A.44}$$

then the thick set $[\![\mathbb{X}]\!] = [\![\mathbb{X}^-, \mathbb{X}^+]\!] = [\mathbf{f}]^{-1}(\mathbb{Y})$ is composed with the two sets

$$\mathbb{X}^- = \{\mathbf{x} \in \mathbb{R}^2 \mid \forall i \in \{1, \dots, 5\}, \forall t \in [t_i], x_1 e^{-x_2 t} \in [y_i]\} \tag{A.45}$$

and

$$\mathbb{X}^+ = \{\mathbf{x} \in \mathbb{R}^2 \mid \forall i \in \{1, \dots, 5\}, \exists t \in [t_i], x_1 e^{-x_2 t} \in [y_i]\}. \tag{A.46}$$

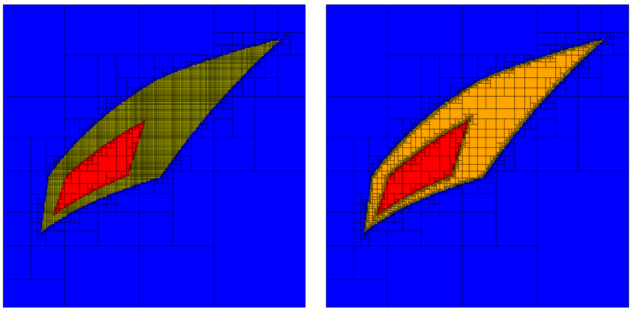


Figure A.10: Representation of the thick set $\llbracket \mathbb{X}^-, \mathbb{X}^+ \rrbracket$ associated to the estimation problem. All blue boxes are outside $\mathbb{X}^+$ and all red boxes are inside $\mathbb{X}^-$. The orange boxes (on the right figure) are outside $\mathbb{X}^-$ and inside $\mathbb{X}^+$. The left Figure, obtained with classical interval tests, do not classify any box in the penumbra. With a thick set inversion approach, we get an inner approximation of the penumbra.

For $\varepsilon = 0.1$, the thick set-inversion algorithm computes an approximation of the thick set $\llbracket \mathbb{X}^-, \mathbb{X}^+ \rrbracket$ as represented by Figure A.10. The left figure is obtained in 1.8 sec and contains 10337 boxes. The right figure is obtained in 0.2 sec and contains 2744 boxes.

Test-case 4. *Communication area.*Consider p marks $\mathbf{m}\,(i)$ located at position $(m_1\,(i)\,, m_2\,(i))$ given by Table A.3 and a robot at the position $\mathbf{x} = (x_1, x_2)$.

i	1	2	3	4
$\mathbf{m}_1\,(i)$	1 ± 0.5	10 ± 0.5	10 ± 0.5	-2 ± 0.5
$\mathbf{m}_2\,(i)$	3 ± 0.5	-1 ± 0.5	6 ± 0.5	-5 ± 0.5

Table A.3: Location of the marks

The robot is able to communicate with the mark $\mathbf{m}\,(i)$ if its distance to the mark is smaller than 10m, *i.e.*, if $\|\mathbf{x} - \mathbf{m}(i)\| < 10$. The communication is not possible if the distance is larger than 20m. With a distance inside $[10, 20]$, the communication is uncertain. The set of all positions for the robot such that the robot can communicate with all marks is a thick set defined by

$$\llbracket \mathbb{X} \rrbracket = [\mathbf{f}]^{-1}\left(\llbracket \mathbb{Y}^-, \mathbb{Y}^+ \rrbracket\right) \tag{A.47}$$

where

$$\mathbb{Y}^- = [0, 10]^{\times 4},\ \mathbb{Y}^+ = [0, 20]^{\times 4} \tag{A.48}$$

and

$$[f_i](\mathbf{x}) = \|\mathbf{x} - [\mathbf{m}](i)\|. \tag{A.49}$$

Our thick inversion algorithm provides in less than 0.3 seconds, the paving represented on Figure A.11. In the example, the clear and the dark boxes could have been obtained using existing interval algorithms. But these methods have to bisect

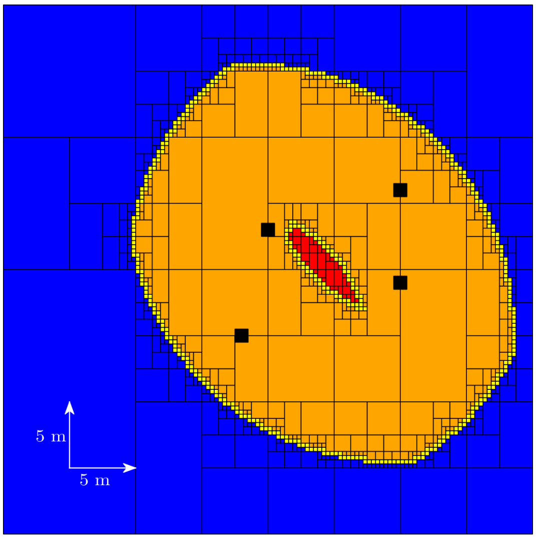


Figure A.11: Thick set $[\![\mathbb{X}]\!]$ representing the communication region. The dark zone (blue) corresponds to position where the robot cannot communicate with all marks. In the clear zone (red), the robot is able to communicate with all marks.

everywhere inside the penumbra. Using thick interval arithmetic, we are able to conclude for an orange box that there is no need to bisect it.

A.6 Conclusion

This chapter deals with the set-inversion problem $\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$ in the case where both $\mathbf{f}$ and $\mathbb{Y}$ are uncertain, *i.e.*, $\mathbf{f}$ belongs to the interval of functions $[\mathbf{f}^-, \mathbf{f}^+]$ and $\mathbb{Y}$ belongs to a thick set, *i.e.*, an interval of sets $[\![\mathbb{Y}]\!] = [\![\mathbb{Y}^-, \mathbb{Y}^+]\!]$. After introducing the new notions of thick intervals and thick boxes, a new algorithm for set inversion has been proposed. It is able to compute a thick solution set $[\![\mathbb{X}]\!] = [\![\mathbb{X}^-, \mathbb{X}^+]\!]$ containing all feasible solution sets.

From the computational point of view, thick intervals allow us to have a better understanding of the uncertainty. For instance, for the set inversion problem, we are able to detect that a box is included in the penumbra $\mathbb{X}^+ \backslash \mathbb{X}^-$. In this penumbra, we can conclude that any bisection would be useless. This could not have been detected using classical intervals. As a consequence, the accumulation zone (*i.e.*, the part of the search space where tiny boxes are still bisected) for thick interval based algorithms has a zero volume, since it corresponds to the boundary of the penumbra. Using classical intervals instead, we could obtain similar results, but the accumulation zone would correspond to the whole penumbra, which has a nonzero volume. As a result, a large part of the computational burden made by traditional interval algorithms is done on a part of the search space which has no influence on the final result.

To make the link with the shape registration and carving problem, the thick set inversion algorithm allows to deal efficiently with uncertain transformations that can be modeled by thick functions. The translation of an uncertain vector, that will be used in applications of chapter 5, fit this definition.

Appendix B

Guaranteed Area Explored

Contents

B.1 Introduction

In Appendix A, the thick set inversion problem for which the uncertain function is described by a thick function, has been defined. A new algorithm has been provided to efficiently characterize the penumbra and was illustrated with some examples. In Chapter 5 a generic method, based on the projection algorithms, has also been introduced.

Now, this chapter deals with a new method able to characterize the area explored by a robot taking into account its trajectory uncertainty. When the dimension of the observer space is smaller that those of the working space, previous methods cannot be used. The temporal evolution of the vehicle need to be correctly handled in by the set inversion algorithm.

Behind the practical case of the characterization of the area explored by a robot, the exposed method will allow us to compute a set that encloses the sensor data taking into account its geometry, its positioning uncertainty and the movement of its carrier.

It is organized as follows. Section B.2 defines the notion of explored zone and proposes a formalization of the problem. Section B.3 extends the formalism in order to take into account the uncertainty associated with the trajectory of the robot. Section B.4 gives the new algorithm which encloses the explored zone $\mathbb{Z}$ between two subpavings (union of boxes). An experiment involving the underwater robot *Daurade* is treated in Section B.5 in order to validate the feasibility of the approach. A conclusion is then given on Section B.6.

Note that, this chapter is extracted, almost with any changes, from [Desrochers and Jaulin, 2017b]. The original paper was written before the thick set formalism was clearly defined, but it helped us to better understand and define the notion of penumbra.

B.2 Problem Statement

We consider a robot moving inside an unknown environment in a dead reckoning manner, *i.e.*, using the proprioceptive sensors only. As it is the case for most industrial underwater robots, the robot is equipped with some exteroceptive sensors (such as sonars or cameras) that are used exploration only and not for navigation. After the mission, it has to find which part of the environment it has explored, taking into account the uncertainty on the localization. We assume that the interpretation of exploration sensors is not reliable enough to allow us using SLAM techniques such as in [Leonard et al., 1992] or [Frese, 2006]. When the mission is short and the quality of the proprioceptive sensors is good, *occupancy map* techniques [Elfes, 1987] can be used to mark all points that have been observed and a probability of being explored can be associated with each part of the space [Paull, 2013]. Now, in practice, due to the state noise, the prediction of the location of the robot is getting less accurate, and we have to take into account these uncertainties properly. We are in a typical situation where the uncertainty is combined with inaccuracy [Dubois and Prade, 2015]. The unknown variable of our problem is the explored set $\mathbb{Z}$, and characterizing uncertain sets with classical probabilistic methods requires elaborated mathematical tools such as random sets [Molchanov, 2005a]. These random sets have already been used in the context of robot mapping (see [Mullane et al., 2011]) but the approach is limited to finite sets. In a set membership approach, an uncertain set $\mathbb{Z}$ can be bracketed by two sets $\mathbb{Z}^-$ and $\mathbb{Z}^+$ such that $\mathbb{Z}^- \subset \mathbb{Z} \subset \mathbb{Z}^+$. This representation is particularly adapted to represent uncertain maps [Langerwisch and Wagner, 2013]. Interval-type uncertainties can easily be propagated through nonlinear functions (see e.g., [Gning and Bonnifait, 2006]) or nonlinear state equations, as shown in [Collins and Goldsztejn, 2008] to get an inner and an outer characterization of the reachable space or in [Jaulin, 2011] for SLAM involving unstructured maps.

The problem to be considered here is the characterization of the *explored zone* $\mathbb{Z}$ of a robot. The set $\mathbb{Z}$ is defined by

$$\begin{cases} \text{(i)} & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}),\ \mathbf{x}(0) = \mathbf{x}_0,\ \mathbf{u}(t) \in [\mathbf{u}](t) \\ \text{(ii)} & \mathbb{Z} = \bigcup_{t\geq 0} \ \mathbb{V}(\mathbf{x}(t)) \end{cases} \tag{B.1}$$

where (i) describes the evolution of the robot and (ii) defines the explored zone $\mathbb{Z}$. In the state equation (i) of the robot, $\mathbf{x} \in \mathbb{R}^n$ is the state vector and $\mathbf{u}$ is the input vector. In order to take into account some state noise, we assume that, for all t, a box $[\mathbf{u}](t)$ which contains $\mathbf{u}(t)$ is available. Moreover, the initial condition $\mathbf{x}_0$ is assumed to be known. For each t, a scanner on the robot is able to observe a part of its environment. More precisely, for each t, there exists a subset of the environment $\mathbb{V}(\mathbf{x}(t)) \subset \mathbb{R}^q$, $q \in \{2,3\}$ that is visible by the robot. This set is called the *visible set* (see, *e.g.*, [Guyonneau et al., 2013] in the context of localization). Note that our robot does not use any exteroceptive sensors for state estimation, for localization or to control its trajectory. It is only able to estimate its position from the proprioceptive sensors $\mathbf{u}$ with a given bounded accuracy. Equivalently, we consider that the scan sensors of the robot are used to collect data for exploration that will only be analyzed after the mission by some human users. The following examples illustrate how the visible sets could be defined in practice.

Example 1. Consider a robot, the pose of which is $\mathbf{x} = (x, y, \theta)$, moving in a plane. This robot is able to scan the environment up to a distance of 3 meters in front of it inside a cone of $\pm\frac{\pi}{6}$rad. Therefore, the visible set $\mathbb{V}(\mathbf{x})$ contains all $\mathbf{z} \in \mathbb{R}^2$ which satisfy the following inequalities

$$\begin{cases} (z_1 - x)^2 + (z_2 - y)^2 & \leq 9 \\ \cos\theta \cdot (z_1 - x) + \sin\theta \cdot (z_2 - y) & \geq \cos\frac{\pi}{6} \end{cases} \tag{B.2}$$

Example 2. We consider the robot of Example 1 except that now, this robot is able to scan only points that are exactly on its right at a distance in $[2, 3]$. Then $\mathbb{V}(\mathbf{x})$ is the set of all $\mathbf{z} \in \mathbb{R}^2$ such that

$$\begin{cases} (z_1 - x)^2 + (z_2 - y)^2 & \in [4, 9] \\ \cos\theta \cdot (z_1 - x) + \sin\theta \cdot (z_2 - y) & = 0 \\ \cos\theta \cdot (z_2 - y) - \sin\theta\,(z_1 - x) & \leq 0. \end{cases} \tag{B.3}$$

Example 3. The same robot is now able to scan only points that are exactly on its right at a distance of 3 meters. Then $\mathbb{V}(\mathbf{x})$ is the set of all $\mathbf{z} \in \mathbb{R}^2$ such that

$$\begin{cases} (z_1 - x)^2 + (z_2 - y)^2 - 3^2 & = 0 \\ \cos\theta \cdot (z_1 - x) + \sin\theta \cdot (z_2 - y) & = 0 \\ \cos\theta \cdot (z_2 - y) - \sin\theta\,(z_1 - x) & \leq 0. \end{cases} \tag{B.4}$$

As motivated by the following examples, we may consider three different types of exploration.

1. *Patch exploration* [Drevelle et al., 2013]. The visible sets $\mathbb{V}(\mathbf{x})$ have a dimension q, as for Example 1. In this case, dim($\mathbb{Z}$)=q and it will be possible to compute an inner and an outer approximation of $\mathbb{Z}$. This type of exploration exists when the sensor is a camera: a 3D zone is explored for each t.
2. *Sweep exploration.* The visible sets $\mathbb{V}(\mathbf{x})$ have a dimension $q-1$, as for Example 2. Thus, $\mathbb{Z}$ has a dimension of q. Again, it will be possible [Olivier et al., 2013] to bracket $\mathbb{Z}$ from both inside and outside. This is the case when the robot is equipped with a side scan sonar, a multi-beam echo sounder or an airborne LiDAR.
3. *Pen exploration.* The dimension of $\mathbb{V}(\mathbf{x})$ is smaller than $q-2$, as for Example 3, and thus dim($\mathbb{Z}$) $\leq q-1$. It will neither be possible to get an inner approximation for $\mathbb{Z}$ nor to prove that a given point is actually inside $\mathbb{Z}$. This is the case when the robot is equipped with a laser range-finder and takes one range measurement every second.

When the whole trajectory $\mathbf{x}(t)$ is known, the set $\mathbb{Z}$ is clearly defined. Different techniques could be thought in order to characterize the set $\mathbb{Z}$. For instance, we could use subpaving-based methods [Jaulin et al., 2001b] or occupancy grid approaches. Now, in this paper, the trajectory of the robot is not known precisely and the set $\mathbb{Z}$ becomes uncertain which makes the problem much more difficult [Paull, 2013].

B.3 Uncertain Explored Zones

When the trajectory $\mathbf{x}(\cdot)$ is uncertain, the explored zone $\mathbb{Z}$ cannot be approximated with an arbitrary accuracy. In a probabilistic context, we can associate to each $\mathbf{z} \in \mathbb{R}^q$ a probability of being explored. Now, even if this probability can be estimated using Monté-Carlo methods, due to the large (here infinite) dimension of the set of trajectories to be explored, the computational burden is high which makes probabilistic methods not so attractive. Denote by $\mathcal{X}(\cdot)$, the set of all feasible trajectories, *i.e.*, the trajectories $\mathbf{x}(\cdot)$ consistent with (B.1) and with the initial condition $\mathbf{x}(0)$. It is important not to make the confusion between $\mathbf{x}(\cdot)$ which is a trajectory and $\mathbf{x}(t)$ which is a vector of $\mathbb{R}^n$. We define the two following sets

$$\begin{aligned} \mathbb{Z}^- &= \bigcap_{\mathbf{x}(\cdot)\in\mathcal{X}(\cdot)} \bigcup_{t\geq 0} \mathbb{V}(\mathbf{x}(t)) \\ \mathbb{Z}^+ &= \bigcup_{\mathbf{x}(\cdot)\in\mathcal{X}(\cdot)} \bigcup_{t\geq 0} \mathbb{V}(\mathbf{x}(t)) \end{aligned} \tag{B.5}$$

The set $\mathbb{Z}^-$ is called the *certainly explored zone* or the *clear zone.* It corresponds to the set of all points $\mathbf{z}$ of the environment that have certainly been seen by the

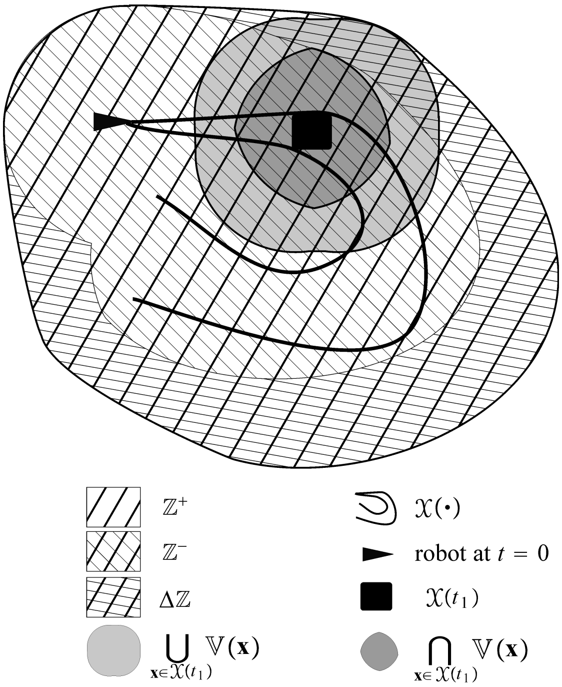


Figure B.1: The space is partitioned into three zones : $\overline{\mathbb{Z}^+}$ is the *dark zone*, $\overline{\mathbb{Z}^-}\cap \mathbb{Z}^+$ is the *penumbra* and $\mathbb{Z}^-$ is the *clear zone*

robot taking into account that its trajectory $\mathbf{x}(\cdot)$ is feasible. The complementary set $\overline{\mathbb{Z}^+}$ of $\mathbb{Z}^+$ is called the *certainly unexplored zone* or the *dark zone*. The set $\mathbb{Z}^+$ is called *non-dark zone*. Moreover, we define the *penumbra* as the set $\Delta\mathbb{Z} = \mathbb{Z}^+ \backslash \mathbb{Z}^-$ (see Figure B.1). It corresponds to the set of all $\mathbf{z}$ that have been seen by some feasible trajectories and not seen by some other feasible trajectories. Figure B.1 illustrates a situation of a patch exploration where the patch is a disk. At time t_1 the robot knows that it is inside the black box. Taking this uncertainty into account, it concludes that the dark gray zone is certainly illuminated (*i.e.*, all corresponding points $\mathbf{z}$ are certainly seen at time t_1) and that all points inside the light gray zone have possibly been seen at time t_1. The unknown explored zone $\mathbb{Z}$ obviously satisfies

$$\mathbb{Z}^- \subset \mathbb{Z} \subset \mathbb{Z}^+. \tag{B.6}$$

Note that the clear zone $\mathbb{Z}^-$ is larger than the union of the certainly visible sets, i.e.,

$$\underbrace{\bigcup_{t\geq 0}\bigcap_{\mathbf{x}\in\mathcal{X}(t)} \mathbb{V}(\mathbf{x})}_{\{\mathbf{z}\ \mid\ \exists t\ \forall \mathbf{x}\in\mathcal{X}(t),\ \mathbf{z}\in\mathbb{V}(\mathbf{x})\}} \subset \mathbb{Z}^- = \underbrace{\bigcap_{\mathbf{x}(\cdot)\in\mathcal{X}(\cdot)}\bigcup_{t\geq 0}\mathbb{V}(\mathbf{x}(t))}_{\{\mathbf{z}\ \mid\ \forall \mathbf{x}(\cdot)\in\mathcal{X}(\cdot),\exists t,\ \mathbf{z}\in\mathbb{V}(\mathbf{x})\}} . \tag{B.7}$$

This inclusion is due to the fact that, $\cap$ and $\cup$ do not commute in general. The consequence of this non commutativity is illustrated by Figure B.2 where two different trajectories for the robot are represented. The pies correspond to the set

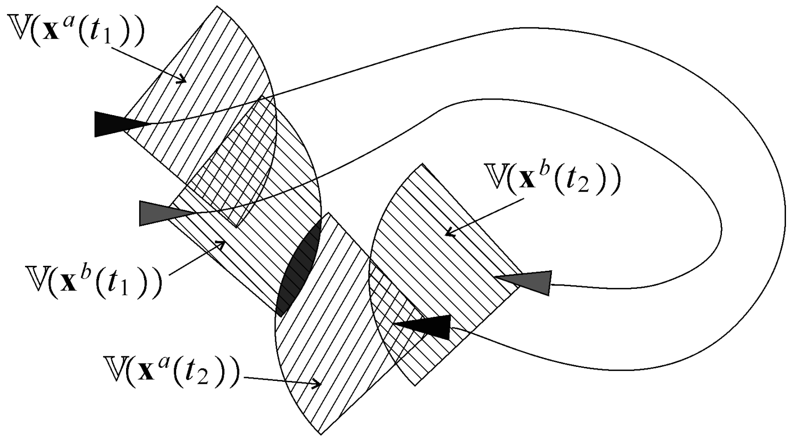

$\mathbb{V}(\mathbf{x}^a(t_1))$
$\mathbb{V}(\mathbf{x}^b(t_2))$
$\mathbb{V}(\mathbf{x}^b(t_1))$
$\mathbb{V}(\mathbf{x}^a(t_2))$

Figure B.2: Even if we do not know which one of the two trajectories is the true one, we are certain that the gray set has been explored; the pies correspond to illuminated zones.

of all $\mathbf{z}$ that have been seen by the robot at times $t_1$and t_2 for the two trajectories. To characterize $\mathbb{Z}^-$, it is not sufficient to compute the union of all certainly visible sets ; we also need to add all cross intersections. For instance, whatever is the true trajectory among $\mathbf{x}^a$ or $\mathbf{x}^b$, we are certain that all points of the dark gray set (at the intersection between the two opposite pies) in the figure have been explored. These cross intersections are not taken into account by existing methods [Drevelle et al., 2013].

Now, for sweep exploration, the dimension of the visible sets is $q-1$ and the certainly visible sets $\bigcap_{\mathbf{x}\in\mathcal{X}(t)} \mathbb{V}(\mathbf{x})$ become empty. All the information about $\mathbb{Z}^-$ can thus only be obtained from these cross intersections, *i.e.*, by taking into account different t and different trajectories all together or equivalently, by considering the right-hand side of (B.7) for the characterization of $\mathbb{Z}^-$.

B.4 Characterization of the Explored Zone

We now focus our attention on the sweep exploration where visible sets have the form

$$\mathbb{V}(\mathbf{x}) = \{\mathbf{z}\in\mathbb{R}^q \mid \varphi(\mathbf{z},\mathbf{x}) = 0 \text{ and } \psi(\mathbf{z},\mathbf{x}) \leq 0\}. \tag{B.8}$$

In this formula, $\varphi : \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}$ is the *visibility function* which is assumed to be continuous and $\psi : \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}$ models the scope of the sensor. Examples 1, 2, 3 of Section B.2 provide an illustration of these functions.

This section proposes a new algorithm able to bracket the explored zone between two subpavings (union of non-overlapping boxes). More precisely, the workspace will be partitioned into 4 types of boxes: (a) the *clear* boxes that have been proved to be inside $\mathbb{Z}^-$, (b) the *dark* boxes that have been proved to be outside $\mathbb{Z}^+$, (c) the *penumbra* boxes which are neither dark nor clear and (d) the boxes which have not been classified yet. When a box cannot be classified, it is bisected into two non overlapping subboxes. To ensure the convergence of the algorithm, boxes which have a width smaller than a small given value are not bisected.

To classify boxes (clear, dark, penumbra), we need first to get a guaranteed enclosure of the set of feasible trajectories $\mathcal{X}(\cdot)$. This can be done using interval guaranteed integration [Wilczak and Zgliczynski, 2011], [Tucker, 2002], [Revol et al., 2005]. These techniques make it possible to compute a tube (*i.e.*, a function $[\mathbf{x}](\cdot)$ mapping $\mathbb{R}$ into $\mathbb{IR}^n$, the set of boxes of $\mathbb{R}^n$) which encloses the set of trajectories $\mathcal{X}(\cdot)$. In what follows, we shall thus assume that the trajectory $\mathbf{x}(\cdot)$ is inside a tube denoted by $[\mathbf{x}](\cdot)$. As a consequence, from (B.5), we have

$$\bigcap_{\mathbf{x}(\cdot)\in[\mathbf{x}](\cdot)} \bigcup_{t\geq 0} \mathbb{V}(\mathbf{x}(t)) \subset \mathbb{Z} \subset \bigcup_{\mathbf{x}(\cdot)\in[\mathbf{x}](\cdot)} \bigcup_{t\geq 0} \mathbb{V}(\mathbf{x}(t)). \tag{B.9}$$

B.4.1 Notion of logic

The presentation of the tests that will be used to classify the boxes of the workspace requires some notions on logic, involving quantifiers. Some useful rules are now presented. If A is a predicate, if $\varphi(\cdot)$ is continuous, if $\mathbf{x}(\cdot)$ is a trajectory and $[\mathbf{x}](\cdot)$ is a tube, we have

$$\begin{array}{ll}
\text{(i)} & \exists \mathbf{x}(\cdot) \in [\mathbf{x}](\cdot),\ \forall t,\ A(\mathbf{x}(t)) \\
& \quad \Leftrightarrow \forall t,\ \exists \mathbf{a} \in [\mathbf{x}](t),\ A(\mathbf{a}) \\
\text{(ii)} & \exists t,\ \varphi(t) = 0 \wedge \psi(t) \leq 0 \\
& \quad \Leftrightarrow \exists t_1\, \exists t_2, \begin{cases} \varphi(t_1)\,\varphi(t_2) \leq 0 \\ \psi([t_1,t_2]) \subset \mathbb{R}^- \end{cases} \\
\text{(iii)} & \forall \mathbf{x}(\cdot) \in [\mathbf{x}](\cdot),\ \exists t,\ A(\mathbf{x}(t)) \\
& \quad \Leftrightarrow \exists t, \forall \mathbf{a} \in [\mathbf{x}](t),\ A(\mathbf{a}).
\end{array} \tag{B.10}$$

B.4.2 Clarity test

The clarity test aims at proving that a box $[\mathbf{z}]$ has been explored whatever the uncertainties are. This test is based on the following theorem.

Theorem B.1. *Given an interval* $[t_1,t_2] \subset \mathbb{R}^+$ *and a tube* $[\mathbf{x}](\cdot)$*, we have :*

$$\left.\begin{array}{c} \varphi(\mathbf{z},[\mathbf{x}](t_1)) \cdot \varphi(\mathbf{z},[\mathbf{x}](t_2)) \subset \mathbb{R}^- \\ \wedge\ \psi(\mathbf{z},[\mathbf{x}]([t_1,t_2])) \subset \mathbb{R}^- \end{array}\right\} \Rightarrow \mathbf{z} \in \mathbb{Z}^{\text{-}}. \tag{B.11}$$

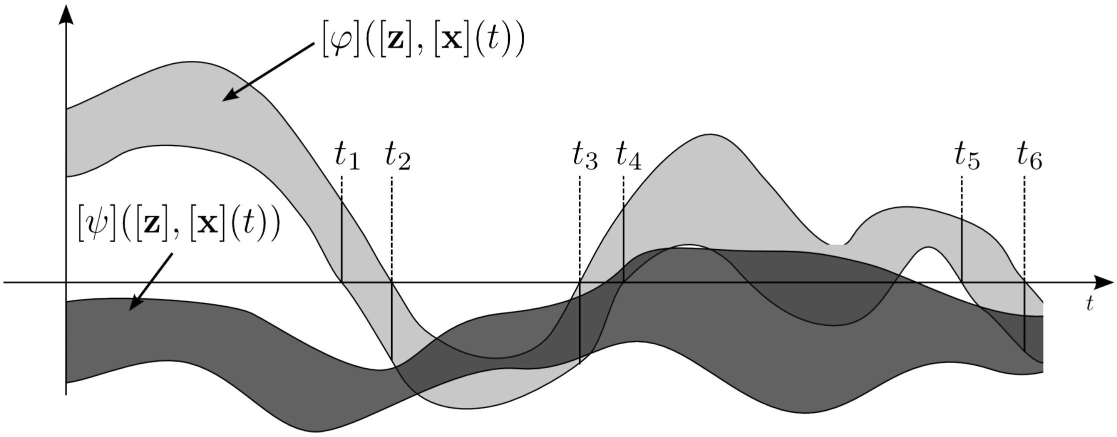


Figure B.3: The fact that φ changes its sign inside $[t_1,t_2]$ whereas ψ remains negative implies that $[\mathbf{z}]\subset\mathbb{Z}$

Proof: Consider a point $\mathbf{z}\in\mathbb{R}^q$. The left-hand side of (B.11) is equivalent to

$$\begin{array}{cl} & \forall\mathbf{x}(\cdot)\in[\mathbf{x}](\cdot),\quad \begin{array}{l}(\varphi(\mathbf{z},\mathbf{x}(t_1))\cdot\varphi(\mathbf{z},\mathbf{x}(t_2))\le 0)\\ \quad\wedge\ \psi(\mathbf{z},\mathbf{x}([t_1,t_2]))\subset\mathbb{R}^-\end{array}\\ \overset{(B.10,ii)}{\Leftrightarrow} & \forall\mathbf{x}(\cdot)\in[\mathbf{x}](\cdot),\ \exists t\ge 0,\ \begin{array}{l}(\ \varphi(\mathbf{z},\mathbf{x}(t))=0\\ \wedge\ \psi(\mathbf{z},\mathbf{x}(t))\le 0\ )\end{array}\\ \Leftrightarrow & \mathbf{z}\in\bigcap_{\mathbf{x}(\cdot)\in[\mathbf{x}](\cdot)}\ \bigcup_{t\ge 0}\left\{\mathbf{z}\in\mathbb{R}^q\mid \begin{array}{l}(\ \varphi(\mathbf{z},\mathbf{x}(t))=0\\ \wedge\ \psi(\mathbf{z},\mathbf{x}(t))\le 0\ )\end{array}\right\}\\ \overset{(B.8)}{\Leftrightarrow} & \mathbf{z}\in\bigcap_{\mathbf{x}(\cdot)\in[\mathbf{x}](\cdot)}\ \bigcup_{t\ge 0}\mathbb{V}(\mathbf{x}(t))\\ \overset{(B.9)}{\Rightarrow} & \mathbf{z}\in\mathbb{Z}^-.\ \blacksquare\end{array}$$

Clarity test. Given t_1,$t_2\ge 0$ and inclusion functions $[\varphi]$,$[\psi]$ for φ,ψ, we define the clarity test as:

$$T_{\text{clarity}}([\mathbf{z}],[\mathbf{x}](\cdot)):\left\{\begin{array}{c}[\varphi]([\mathbf{z}],[\mathbf{x}](t_1))\cdot[\varphi]([\mathbf{z}],[\mathbf{x}](t_2))\subset\mathbb{R}^-\\ \wedge[\psi]([\mathbf{z}],[\mathbf{x}]([t_1,t_2]))\subset\mathbb{R}^-\end{array}\right.$$

From Theorem B.1, we have

$$T_{\text{clarity}}([\mathbf{z}],[\mathbf{x}](\cdot))\Rightarrow[\mathbf{z}]\subset\mathbb{Z}^-. \tag{B.12}$$

An illustration of this proposition is given by Figure B.3. We choose the pairs (t_1,t_2) as in the figure, i.e., such that t_2-t_1 are small and such that

$$[\varphi]([\mathbf{z}],[\mathbf{x}](t_1))\cdot[\varphi]([\mathbf{z}],[\mathbf{x}](t_2))\subset\mathbb{R}^-. \tag{B.13}$$

Note that the test also concludes for the pair (t_5,t_6) but fails for (t_3,t_4). Since here, the test concludes for at least one pair, we get that the whole box $[\mathbf{z}]$ has been explored.

B.4.3 Dark test

The dark test aims at proving that no point in the box $[\mathbf{z}]$ has been explored whatever the uncertainties are. The dark test is based on the following theorem.
Theorem B.2. *If* $[\mathbf{x}](\cdot)$ *is a tube, then we have*

$$\begin{array}{c} \forall t \geq 0, \forall \mathbf{x} \in [\mathbf{x}](t), (\varphi(\mathbf{z},\mathbf{x}) \neq 0) \\ \vee (\psi(\mathbf{z},\mathbf{x}) > 0) \Rightarrow \mathbf{z} \notin \mathbb{Z}^{+} \end{array} \tag{B.14}$$

Proof. The proof is by contradiction, *i.e.*, we show that assuming $\mathbf{z} \in \mathbb{Z}^{+}$ implies that left-hand side of (B.14) is false. We have

$$\begin{array}{cl} & \mathbf{z} \in \mathbb{Z}^{+} \\ \overset{(B.9)}{\Rightarrow} & \mathbf{z} \in \bigcup\limits_{\mathbf{x}(\cdot)\in[\mathbf{x}](\cdot)} \bigcup\limits_{t\geq 0} \mathbb{V}(\mathbf{x}(t)) \\ \overset{(B.8)}{\Leftrightarrow} & \mathbf{z} \in \bigcup\limits_{\mathbf{x}(\cdot)\in[\mathbf{x}](\cdot)} \bigcup\limits_{t\geq 0} \left\{ \mathbf{z} \in \mathbb{R}^{q} \mid \begin{array}{l} (\ \varphi(\mathbf{z},\mathbf{x}(t)) = 0 \\ \wedge\ \psi(\mathbf{z},\mathbf{x}(t)) \leq 0\ ) \end{array} \right\} \\ \Leftrightarrow & \exists \mathbf{x}(\cdot) \in [\mathbf{x}](\cdot), \exists t \geq 0, \begin{array}{l} (\ \varphi(\mathbf{z},\mathbf{x}(t)) = 0 \\ \wedge\ \psi(\mathbf{z},\mathbf{x}(t)) \leq 0\ ) \end{array} \end{array}$$

which corresponds to the negation of left-hand side of (B.14). ■

Dark test. Given two inclusion functions $[\varphi]$,$[\psi]$ for φ, ψ. Define the *dark test* as

$$T_{\text{dark}}([\mathbf{z}],[\mathbf{x}](\cdot)) : \forall t \geq 0, \left\{ \begin{array}{c} (0 \notin [\varphi]([\mathbf{z}],[\mathbf{x}](t))) \\ \vee ([\psi]([\mathbf{z}],[\mathbf{x}](t)) \subset \mathbb{R}^{+}) . \end{array} \right. \tag{B.15}$$

From Theorem B.2, we have

$$T_{\text{dark}}([\mathbf{z}],[\mathbf{x}](\cdot)) \Rightarrow [\mathbf{z}] \cap \mathbb{Z}^{+}. \tag{B.16}$$

An illustration of this proposition is given by Figure B.4. In the situation represented in this figure, $\forall t \geq 0, \forall \mathbf{x} \in [\mathbf{x}](t), \forall \mathbf{z} \in [\mathbf{z}]$, either$\varphi(\mathbf{z},\mathbf{x}) \neq 0$or $(\psi(\mathbf{z},\mathbf{x}) > 0)$, which means that no point inside $\mathbf{z} \in [\mathbf{z}]$ could have been seen.

B.4.4 Penumbra test

Proving that a point $\mathbf{z}$ belongs to the penumbra $\Delta\mathbb{Z}$ is difficult and not so useful. The main interest to have an inclusion test for the penumbra is to limit the computation burden. Indeed, if $[\mathbf{z}] \subset \Delta\mathbb{Z}$ then we will never be able to prove that $[\mathbf{z}] \subset \mathbb{Z}^{-}$ or to prove that $[\mathbf{z}] \cap \mathbb{Z}^{+} = \emptyset$, and thus there is no need to bisect. Now, instead of proving that $[\mathbf{z}] \subset \Delta\mathbb{Z}$, we propose here to test either the darkness test or the clarity test always fail for all subbox of $[\mathbf{z}]$. In such a case, $[\mathbf{z}]$ will not be bisected. In such a situation, we shall say that $[\mathbf{z}]$ satisfies the *penumbra test.*

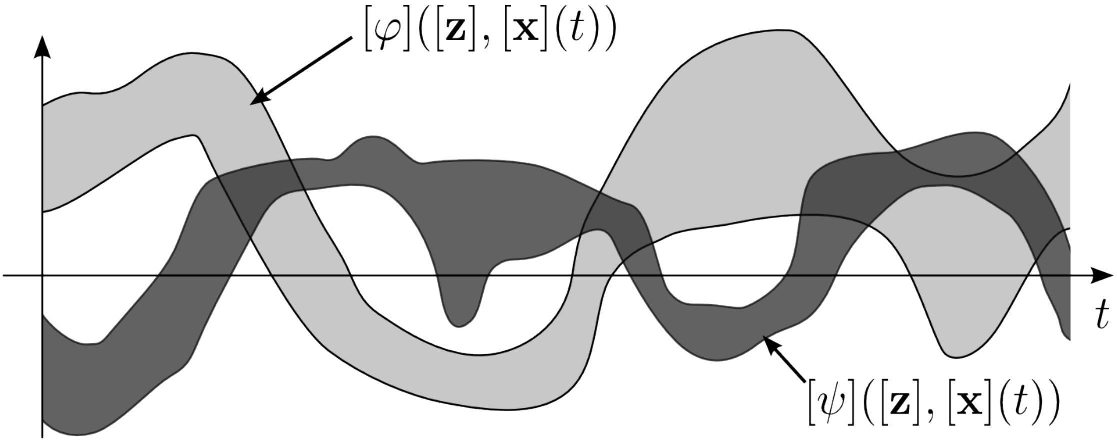

$[\varphi]([\mathbf{z}],[\mathbf{x}](t))$
t
$[\psi]([\mathbf{z}],[\mathbf{x}](t))$

Figure B.4: This configuration for the tubes $[\varphi](\cdot)$ and $[\psi](\cdot)$ allows us to conclude that $[\mathbf{z}]$ is dark

Theorem B.3. *Given* $\mathbf{x}(\cdot) \in [\mathbf{x}](\cdot)$ *and* $\mathbf{z} \in [\mathbf{z}]$. *If*

$$\forall t, \left\{ \begin{array}{c} \varphi(\mathbf{z}, \mathbf{x}(t)) \neq 0 \\ \vee \ \psi(\mathbf{z}, \mathbf{x}(t)) > 0 \end{array} \right. \Rightarrow \neg T_{clarity}([\mathbf{x}](.), [\mathbf{z}]). \tag{B.17}$$

Proof. The proof is by contradiction. Assume that $T_{\text{clarity}}([\mathbf{x}](.), [\mathbf{z}])$ is true, i.e.,

$$\left\{ \begin{array}{c} [\varphi]([\mathbf{z}], [\mathbf{x}](t_1)) \cdot [\varphi]([\mathbf{z}], [\mathbf{x}](t_2)) \subset \mathbb{R}^- \\ \wedge [\psi]([\mathbf{z}], [\mathbf{x}]([t_1, t_2])) \subset \mathbb{R}^-. \end{array} \right.$$

Since $\mathbf{x}(\cdot) \in [\mathbf{x}](\cdot)$ and $\mathbf{z} \in [\mathbf{z}]$, we get

$$\left\{ \begin{array}{c} \varphi(\mathbf{z}, \mathbf{x}(t_1)) \cdot \varphi(\mathbf{z}, \mathbf{x}(t_2)) \subset \mathbb{R}^- \\ \wedge \psi(\mathbf{z}, \mathbf{x}([t_1, t_2])) \subset \mathbb{R}^-. \end{array} \right.$$

From (B.10,ii), we get that there exists $t_0 \in [t_1, t_2]$ such that

$$\varphi(\mathbf{z}, \mathbf{x}(t_0)) = 0 \wedge \psi(\mathbf{z}, \mathbf{x}(t_0)) \subset \mathbb{R}^-$$

which is inconsistent with the left-hand side of (B.17). ■

Theorem B.4. *Given* $\mathbf{x}(\cdot) \in [\mathbf{x}](\cdot), \mathbf{z} \in [\mathbf{z}]$ *and* $t_1, t_2 \geq 0$. *We have:*

$$\left. \begin{array}{c} \varphi(\mathbf{z}, \mathbf{x}(t_1)) . \varphi(\mathbf{z}, \mathbf{x}(t_2)) \leq 0 \\ \wedge \ \forall t \in [t_1, t_2], \psi(\mathbf{z}, \mathbf{x}(t)) \leq 0 \end{array} \right\} \Rightarrow \neg T_{dark}([\mathbf{x}](\cdot), [\mathbf{z}]). \tag{B.18}$$

Proof. The proof is by contradiction. Assume that $T_{\text{dark}}([\mathbf{x}](\cdot), [\mathbf{z}])$ is true, *i.e.*,

$$\forall t \geq 0, \ 0 \notin [\varphi]([\mathbf{z}], [\mathbf{x}](t)) \ \ \vee \ \ [\psi]([\mathbf{z}], [\mathbf{x}](t)) \subset \mathbb{R}^+$$

Since $\mathbf{x}(\cdot) \in [\mathbf{x}](\cdot)$ and $\mathbf{z} \in [\mathbf{z}]$, we get

$$\forall t \geq 0,\ 0 \neq \varphi\left(\mathbf{z}, \mathbf{x}\left(t\right)\right) \ \vee \ \psi\left(\mathbf{z}, \mathbf{x}\left(t\right)\right) > 0.$$

Assume now that the left-hand side of (B.18) is also true. From (B.10,ii), we get that for a specific $t_0 \in [t_1, t_2]$,

$$\varphi\left(\mathbf{z}, \mathbf{x}(t_0)\right) = 0 \wedge \psi\left(\mathbf{z}, \mathbf{x}(t_0)\right) \leq 0$$

which is not consistent with the previous proposition.

Penumbra test. Given t_1,$t_2 \geq 0$, two trajectories $\mathbf{x}^a\left(\cdot\right), \mathbf{x}^b\left(\cdot\right) \in \left[\mathbf{x}\right]\left(t\right)$ and inclusion functions $[\varphi]$,$[\psi]$ for φ,ψ. Define the *penumbra test* $T_{\text{penumbra}}\left([\mathbf{z}]\right)$ as

$$\left\{ \begin{array}{c} \forall t, [\varphi]\left(\left[\mathbf{z}\right], \mathbf{x}^a\left(t\right)\right) \neq 0 \ \vee \ \ [\psi]\left(\left[\mathbf{z}\right], \mathbf{x}^a(t)\right) > 0 \\ [\varphi]\left(\left[\mathbf{z}\right], \mathbf{x}^b(t_1)\right) \cdot [\varphi]\left(\left[\mathbf{z}\right], \mathbf{x}^b(t_2)\right) \leq 0 \\ \wedge \ \forall t \in [t_1, t_2], [\psi]\left(\left[\mathbf{z}\right], \mathbf{x}^b(t)\right) \leq 0. \end{array} \right. \tag{B.19}$$

From Theorems B.3 and B.4, we get that T_{penumbra} implies that neither T_{clarity} nor T_{dark} will able to conclude anything for any subbox of $[\mathbf{z}]$.

B.5 Experiment

This section illustrates our sweep exploration method on a real experiment. November 2015, a 46 minutes mission has been performed in the Road-stead of Brest (France, Brittany) with the underwater robot *Daurade* which realized a classical survey pattern composed of a set of parallel tracks with an altitude of about 10 meters. This robot (see Figure B.5) has been built by ECA robotics and used by DGA Tn (Direction Général de l'Armement - Techniques Navales) and SHOM (Service Hydrographique et Océanographique de la Marine) for performing REA (Rapid Environment Assessment) studies. REA is intended to survey the environmental conditions of a particular location in order to identify any existing or potential dangers. In the counter mine warfare context, attention is focused on mapping the sea floor with acoustic sensors. *Daurade* is equipped with a Side Scan Sonar (Klein 5500) used to detect potential mines. With this sonar, data are recorded on a line perpendicular to the path of the sensor and images are formed by putting side by side these lines. Characterizing the zone seen by the sonar enters inside the framework of *sweep exploration*.

For the navigation, *Daurade* relies on an inertial central (Phins II IXBlue) coupled with a DVL (Doppler Velocity Log), which returns the speed with respect to the ground. Once under the water, no GPS data are available and the estimated position of the robot drifts with the time. A key point of REA missions is to guarantee that the area of interest has been totally explored without any gap.

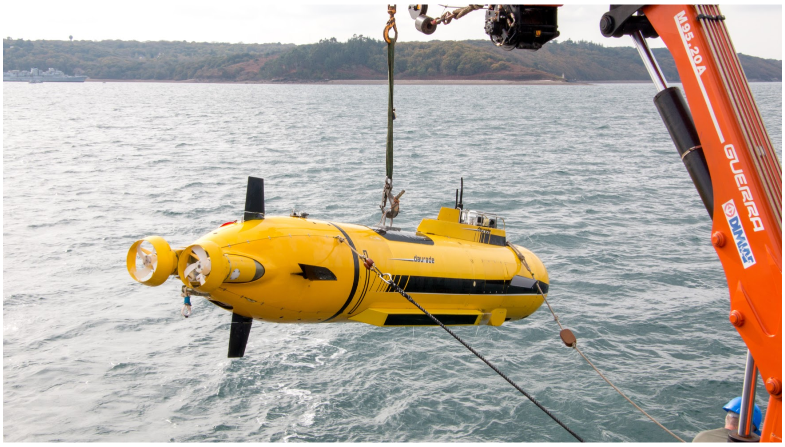


Figure B.5: *Daurade*: the underwater robot used for our experiment. Photo: S. Rohou

Assessment of the covered area is usually done manually by an operator which looks at the sonar images. The proposed algorithm can be used to validate the mission plan or, at the end of the mission, to check the area to be explored has indeed been covered.

The horizontal kinematic model the robot is taken as

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix} \cdot \mathbf{v} \tag{B.20}$$

where (x_1, x_2) corresponds to the 2D coordinates of the center of the robot expressed in an absolute inertial frame, ψ is the heading and $\mathbf{v}$ is the horizontal speed vector of the robot expressed in its own coordinate system. The aperture angle of the side scan sonar is taken as $\alpha = 80°$. This means that, with a flat seabed assumption, if the robot has an altitude a then the sonar is able to sense the part of the environment which is perpendicular to the robot and at a distance less than $\ell = a \cdot \sin\alpha$. The horizontal online estimated trajectory is depicted on Figure B.6. Note that due to the fact that the controller uses this trajectory for control, it looks perfect, which is not the case for the true trajectory (dotted line) directly measured by an USBL (Ultra-Short Base Line). Now, in our experiment, this true trajectory is neither used for control nor for estimation of the explored zone. It is only used for the validation of the results.

We assume that the initial position is $x_1 = x_2 = 0$. Taking the interval uncertainties into account and using a guaranteed integration of the state equation (B.20), we are able to compute a tube which contains the true trajectory. At the end of the mission the position error is around 17 meters. On this experiment, our method provides in less than 5 minutes an estimation of the explored zone as given by Figure B.7. This result is consistent with the true trajectory obtained by the USBL and by an observation of the sonar images.

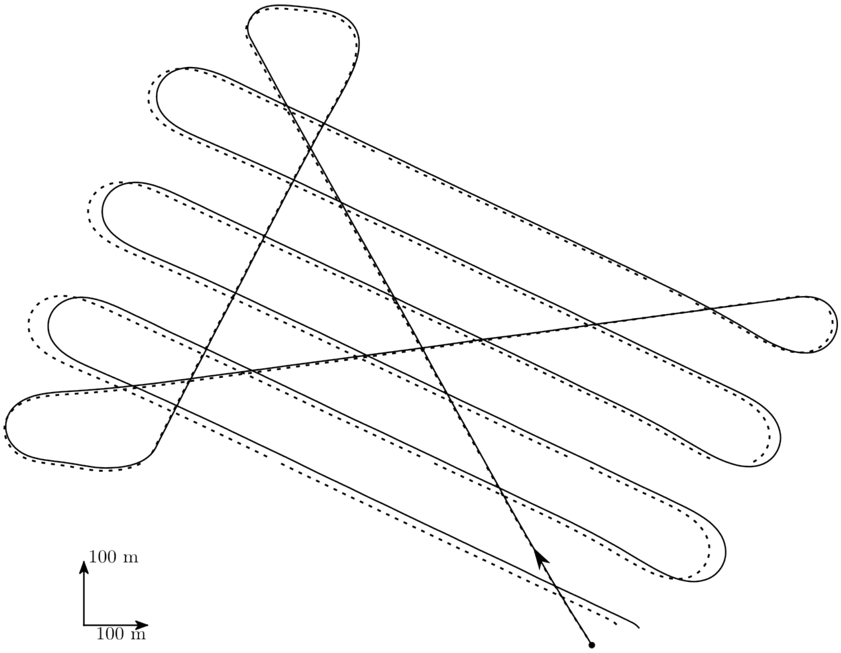


Figure B.6: Online estimated trajectory (plain) and true trajectory (dotted line)

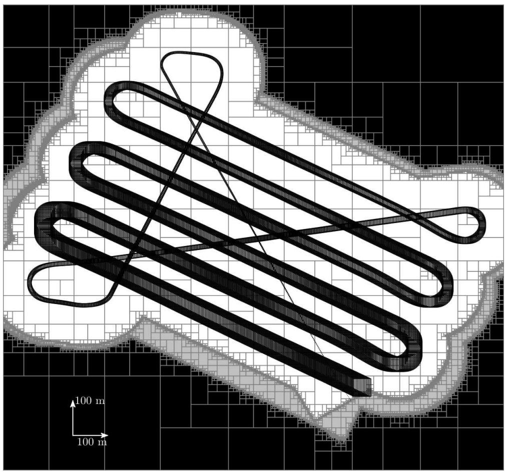


Figure B.7: Enclosure of the explored area. The tube in black encloses the true trajectory

Consider the same experiment, but, instead of having a sweep exploration, we assume that at each t the visible set $\mathbb{V}(t)$ is a disk of radius $a \cdot \sin\alpha$ (instead of a segment). In this case, the explored zone can still be obtained, but we do not need anymore to use the continuity of the trajectory. At each $t \in \{1, 2, \ldots, m\}$ where $m = 2760$, the visible set (*i.e.*, the part of the bottom which is seen by the sonar of the robot) is a disk of radius 50 meters around the robot, the position of which is not exactly known. The *explored zone* $\mathbb{X}$ corresponds to the union of all patches that have been seen:

$$\mathbb{X} = \bigcup_{t\in[1,2,\ldots,m\}} f_t^{-1}([0, 50[), \tag{B.21}$$

where

$$f_t(\mathbf{x}) = \sqrt{(x_1 - a_1(t))^2 + (x_2 - a_2(t))^2} \tag{B.22}$$

The complementary set of $\mathbb{X}$ is

$$\overline{\mathbb{X}} = \bigcap_{t\in\{1,2,\ldots,m\}} f_t^{-1}([50, \infty]) = \mathbf{f}^{-1}([50, \infty]^m),$$

where $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$. The function $\mathbf{f}(\mathbf{x})$ is consistent with the intervals $[a_1](t)$ and $[a_2](t)$ containing the positions of the robot iff

$$\mathbf{f}(\mathbf{x}) \in [\mathbf{f}](\mathbf{x}) = \begin{pmatrix} \sqrt{(x_1 - [a_1](1))^2 + (x_2 - [a_2](1))^2} \\ \vdots \\ \sqrt{(x_1 - [a_1](m))^2 + (x_2 - [a_2](m))^2} \end{pmatrix}$$

Since the function $[\mathbf{f}](\mathbf{x})$ is thick, the characterization of the thick set $\left[\overline{\mathbb{X}}\right]$ is a thick set inversion problem which can be characterized by algorithm presented in Appendix A. Its complementary $[\mathbb{X}]$ can thus be derived. The resulting enclosure of $[\mathbb{X}]$, given on Figure B.8, is computed in less than 3 minutes. Note that the penumbra (orange) is larger in zones when the estimation of the position of the robot is less accurate. The black tube corresponds to $[\mathbf{a}](t)$. From the information given by the inertial system, for any point $\mathbf{x}_o$ in the orange zone, it is impossible to known if $\mathbf{x}_o$ has been seen or not by the sonar. This ambiguity comes from the uncertainty related to the position $\mathbf{a}(t)$ of the robot. Note also that Figures B.8 and B.7 look similar (which is consistent with our intuition) except near the initial position (bottom right) where we can observe the difference between the patch (here a disk) exploration and the sweep (here a segment) exploration.

B.6 Conclusion

This chapter has presented a new method to characterize the zone $\mathbb{Z}$ explored by a mobile robot. The method only uses the proprioceptive sensors for localization, which makes the approach reliable. The method is indeed not sensitive *(i)* to

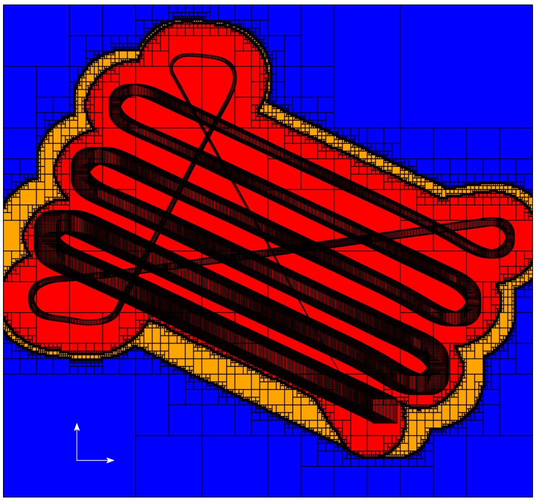


Figure B.8: Patch exploration in the case where the visible sets are disks The dark zone (blue) has certainly been unexplored and the clear zone (red) has certainly been explored. Here, certainly means: 'for all feasible trajectories'

outliers (which mainly occur on exteroceptive sensors), *(ii)* to data association errors, or *(iii)* to any information resulting from an interaction between the robot and the environment. The method can also be used before the mission in order to guarantee that a given area will all be covered by the robot, whatever the noise on the proprioceptive sensors are. We only need to assume that the state noise is bounded with known bounds in order to be able to enclose the trajectory $\mathbf{x}(\cdot)$ inside a tube $[\mathbf{x}](\cdot)$.

To solve the problem, we had to introduce new interval tests to classify boxes as clear (certainly explored), dark (certainly unexplored) or in the penumbra. This classification requires the development of new quantifier elimination procedures, based on four new theorems. The efficiency of the resulting algorithm has been validated on an actual experiment made by an autonomous underwater robot.

Appendix C

Publications

Journal Articles

- Jaulin, L. and Desrochers, B. (2014). Introduction to the algebra of separators with application to path planning. *Engineering Applications of Artificial Intelligence*, 33(0):141–147
- Desrochers, B. and Jaulin, L. (2017b). Computing a Guaranteed Approximation of the Zone Explored by a Robot. *IEEE Transactions on Automatic Control*, 62(1):425–430
- Desrochers, B. and Jaulin, L. (2016a). A minimal contractor for the polar equation: Application to robot localization. *Engineering Applications of Artificial Intelligence*, 55(Supplement C):83–92
- Jaulin, L., Desrochers, B., and Massé, D. (2016). Bisectable Abstract Domains for the Resolution of Equations Involving Complex Numbers. *Reliable Computing*, 23(1):35–46
- Desrochers, B. and Jaulin, L. (2017d). Thick set inversion. *Artificial Intelligence*, 249:1–18

Congress with Selection Committee

- Desrochers, B., Lacroix, S., and Jaulin, L. (2015). Set-membership approach to the kidnapped robot problem. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2015-Decem, pages 3715–3720
- Jaulin, L. and Desrochers, B. (2016). Thick separators. In *COPROD'16*, Uppsala (Sweden)

- Desrochers, B. and Jaulin, L. (2016b). Relaxed intersection of thick sets. In *SCAN 2016*, Uppsala (Sweden)
- Desrochers, B. and Jaulin, L. (2017a). Chain of Set Inversion Problems Application to reachability analysis. *IFAC-PapersOnLine*, 50(1):9266–9271
- Desrochers, B. and Jaulin, L. (2017c). Minkowski Operations of Sets with Application to Robot Localization. In *Electronic Proceedings in Theoretical Computer Science*, volume 247, pages 34–45

Bibliography

[Apt, 1999] Apt, K. R. (1999). The essence of constraint propagation. *Theoretical Computer Science*, 221(1-2):179–210.

[Araya et al., 2008] Araya, I., Neveu, B., and Trombettoni, G. (2008). Exploiting common subexpressions in numerical CSPs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5202 LNCS, pages 342–357, LNCS 5202.

[Aubry et al., 2013] Aubry, C., Desmare, R., and Jaulin, L. (2013). Loop detection of mobile robots using interval analysis. *Automatica*, 49(2):463–470.

[Audette et al., 2000] Audette, M. A., Ferrie, F. P., and Peters, T. M. (2000). An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis*, 4(3):201–217.

[Benhamou et al., 1999] Benhamou, F., Goualard, F., Granvilliers, L., and Puget, J. (1999). Revising hull and box consistency. In *Int. Conf. on Logic Programming*, pages 230–244.

[Berge, 1963] Berge, C. (1963). *Topological Spaces: Including a Treatment of Multi-Valued Functions, Vector Spaces and Convexity.* Dover Books on Mathematics.

[Berz and Makino, 1998] Berz, M. and Makino, K. (1998). Verified Integration of ODEs and Flows Using Differential Algebraic Methods on High-Order Taylor Models. *Reliable Computing*, 4(4):361–369.

[Bethencourt and Jaulin, 2014] Bethencourt, A. and Jaulin, L. (2014). Solving non-linear constraint satisfaction problems involving time-dependant functions. *Mathematics in Computer Science*, 8(3).

[Bottino et al., 2003] Bottino, A., Jaulin, L., and Laurentini, A. (2003). Reconstructing 3D Objects from Silhouettes with Unknown Viewpoints: The Case of Planar Orthographic Views. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 2905, pages 153–162, Havana, Cuba.

[Boukezzoula et al., 2004] Boukezzoula, R., Galichet, S., and Foulloy, L. (2004). Observer-Based Fuzzy Adaptive Control for a Class of Nonlinear Systems: Real-

Time Implementation for a Robot Wrist. *IEEE Transactions on Control Systems Technology*, 12(3):340–351.

[Brooks and Lozano-Perez, 1985] Brooks, R. A. and Lozano-Perez, T. (1985). A Subdivision Algorithm in Configuration Space For Find Path with Rotation. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(2):224 – 233.

[Candau et al., 2006] Candau, Y., Raissi, T., Ramdani, N., and Ibos, L. (2006). Complex interval arithmetic using polar form. *Reliable Computing*, 12(1):1–20.

[Cantor, 1847] Cantor, G. (1847). Ueber eine Eigenschaft des Inbegriffs aller reellen algebraischen Zahlen. *Journal für die reine und angewandte Mathematik*, 1874(77):258–262.

[Carbonnel et al., 2014] Carbonnel, C., Trombettoni, G., and Chabert, G. (2014). Q-intersection Algorithms for Constraint-Based Robust Parameter Estimation. In *Twenty-Eighth Conference on Artificial Intelligence*.

[Castellanos et al., 2001] Castellanos, J., Neira, J., and Tardos, J. (2001). Multisensor fusion for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 17(6):908–914.

[Ceberio and Granvilliers, 2000] Ceberio, M. and Granvilliers, L. (2000). Solving Nonlinear Systems by Constraint Inversion and Interval Arithmetic. In *Intelligence*, volume 1930, pages 127–141, LNCS 5202.

[Ceccarelli et al., 2006] Ceccarelli, N., Marco, M., Garulli, A., Giannitrapani, A., and Vicino, A. (2006). Set Membership Localization and Map Building for Mobile Robots. *Current Trends in Nonlinear Systems and Control SE - Systems and Control: Foundations & Applications*, pages 289–308.

[Cerone, 1991] Cerone, V. (1991). Parameter Bounds for Models with Bounded Errors in all Variables. In Milanese, M., Norton, J., Piet-Lahanier, H., and Walter, E., editors, *IFAC Proceedings Volumes*, volume 24, pages 895–900. Plenum, New York, NY.

[Chabert, 2013] Chabert, G. (2013). A simple SLAM example with IBEX. In *SWIM13*, Brest.

[Chabert and Jaulin, 2007] Chabert, G. and Jaulin, L. (2007). Computing the Pessimism of Inclusion Functions. *Reliable Computing*, 13(6):489–504.

[Chabert and Jaulin, 2009a] Chabert, G. and Jaulin, L. (2009a). A Priori Error Analysis with Intervals. *SIAM Journal on Scientific Computing*, 31(3):2214–2230.

[Chabert and Jaulin, 2009b] Chabert, G. and Jaulin, L. (2009b). Contractor programming. *Artificial Intelligence*, 173(11):1079–1100.

[Chabert and Jaulin, 2009c] Chabert, G. and Jaulin, L. (2009c). Hull Consistency Under Monotonicity. In *CP'2009*.

[Clérentin et al., 2008] Clérentin, A., Delafosse, M., Delahoche, L., Marhic, B., and Jolly-Desodt, A. M. (2008). Uncertainty and imprecision modeling for the mobile robot localization problem. *Autonomous Robots*, 24(3):267–283.

[Colle and Galerne, 2013] Colle, E. and Galerne, S. (2013). Mobile robot localization by multiangulation using set inversion. *Robotics and Autonomous Systems*, 61(1):39–48.

[Collins and Goldsztejn, 2008] Collins, P. and Goldsztejn, A. (2008). The Reach-and-Evolve Algorithm for Reachability Analysis of Nonlinear Dynamical Systems. *Electronic Notes in Theoretical Computer Science*, 223:87–102.

[Combastel,] Combastel, C. A State Bounding Observer for Uncertain Non-linear Continuous-time Systems based on Zonotopes. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 7228–7234. IEEE.

[Daney et al., 2006] Daney, D., Andreff, N., Chabert, G., and Papegay, Y. (2006). Interval method for calibration of parallel robots: Vision-based experiments. *Mechanism and Machine Theory*, 41(8):929–944.

[Davey and Priestley, 2002] Davey, B. A. and Priestley, H. A. (2002). *Summary for Policymakers*, volume 1. Cambridge University Press, Cambridge.

[Davis, 1987] Davis, E. (1987). Constraint propagation with interval labels. *Artificial Intelligence*, 32(3):281–331.

[de Berg et al., 2008] de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. (2008). More Geometric Data Structures. In *Computational Geometry*, chapter 10, page 220. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Dellaert et al., 1999] Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte Carlo localization for mobile robots. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 2, pages 1322–1328, Detroit, Michigan. IEEE.

[Desrochers and Jaulin,] Desrochers, B. and Jaulin, L. Underwater usbl tracking using the polar separator.

[Desrochers and Jaulin, 2016a] Desrochers, B. and Jaulin, L. (2016a). A minimal contractor for the polar equation: Application to robot localization. *Engineering Applications of Artificial Intelligence*, 55(Supplement C):83–92.

[Desrochers and Jaulin, 2016b] Desrochers, B. and Jaulin, L. (2016b). Relaxed intersection of thick sets. In *SCAN 2016*, Uppsala (Sweden).

[Desrochers and Jaulin, 2017a] Desrochers, B. and Jaulin, L. (2017a). Chain of Set Inversion Problems Application to reachability analysis. *IFAC-PapersOnLine*, 50(1):9266–9271.

[Desrochers and Jaulin, 2017b] Desrochers, B. and Jaulin, L. (2017b). Computing a Guaranteed Approximation of the Zone Explored by a Robot. *IEEE Transactions on Automatic Control*, 62(1):425–430.

[Desrochers and Jaulin, 2017c] Desrochers, B. and Jaulin, L. (2017c). Minkowski Operations of Sets with Application to Robot Localization. In *Electronic Proceedings in Theoretical Computer Science*, volume 247, pages 34–45.

[Desrochers and Jaulin, 2017d] Desrochers, B. and Jaulin, L. (2017d). Thick set inversion. *Artificial Intelligence*, 249:1–18.

[Desrochers et al., 2015] Desrochers, B., Lacroix, S., and Jaulin, L. (2015). Set-membership approach to the kidnapped robot problem. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2015-Decem, pages 3715–3720.

[Di Loreto et al., 2007] Di Loreto, M., Dao, M., Jaulin, L., Lafay, J.-F., and Loiseau, J. J. (2007). Applied Interval Computation: A New Approach for Time-Delays Systems Analysis. *In Application of Time-Delay Systems, Lecture Notes in Control and Information Science*, 352:175–197.

[Di Marco et al., 2001] Di Marco, M., Garulli, A., Lacroix, S., and Vicino, A. (2001). Set membership localization and mapping for autonomous navigation. *International Journal of Robust and Nonlinear Control*, 11(7):709–734.

[Didrit et al., 1995] Didrit, O., Jaulin, L., and Walter, E. (1995). Guaranteed Analysis and Optimization of Parametric Systems, with Application to their Stability Degree. In *Proceedings of 3rd European Control Conference*, pages 1412–1417, Rome.

[Dissanayake et al., 2001] Dissanayake, M., Newman, P. M., Clark, S., Durrant-Whyte, H. F., and Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241.

[Drevelle and Bonnifait, 2013] Drevelle, V. and Bonnifait, P. (2013). Reliable Positioning Domain Computation for Urban Navigation. *IEEE Intelligent Transportation Systems Magazine*, 5(3):21–29.

[Drevelle et al., 2013] Drevelle, V., Jaulin, L., and Zerr, B. (2013). Guaranteed Characterization of the Explored Space of a Mobile Robot by using Subpavings. In *IFAC Proceedings Volumes*, volume 46, pages 44–49, Toulouse.

[Drevelle and Nicola, 2014] Drevelle, V. and Nicola, J. (2014). VIBes: A Visualizer for Intervals and Boxes. *Mathematics in Computer Science*, 8(3):563–572.

[Dubois and Prade, 2015] Dubois, D. and Prade, H. (2015). *Possibility Theory.* Plenum, New York.

[Elfes, 1987] Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal on Robotics and Automation*, 3(3):249–265.

[Fallon et al., 2013] Fallon, M. F., Folkesson, J., McClelland, H., and Leonard, J. J. (2013). Relocating underwater features autonomously using sonar-based SLAM. *IEEE Journal of Oceanic Engineering*, 38(3):500–513.

[Frese, 2006] Frese, U. (2006). A Discussion of Simultaneous Localization and Mapping. *Autonomous Robots*, 20(1):25–42.

[Galichet and Foulloy, 1995] Galichet, S. and Foulloy, L. (1995). Fuzzy Controllers: Synthesis and Equivalences. *IEEE Transactions on Fuzzy Systems*, 3(2):140–148.

[Garloff, 1985] Garloff, J. (1985). Convergent bounds for the range of multivariable polynomials. In Nickel, K., editor, *Interval Mathematics 1985*, pages 37–56. Springer-Verlag.

[Gervet, 1997] Gervet, C. (1997). Interval propagation to reason about sets: Definition and implementation of a practical language. *Constraints*, 1:191–246.

[Gning, 2006] Gning, A. (2006). *Localisation garantie d'automobiles. Contribution aux techniques de satisfaction de contraintes sur les intervalles.* Phd dissertation, Université de Technologie de Compiègne, Compiègne, France.

[Gning and Bonnifait, 2006] Gning, A. and Bonnifait, P. (2006). Constraints propagation techniques on intervals for a guaranteed localization using redundant data. *Automatica*, 42(7):1167–1175.

[Gning et al., 2013] Gning, A., Ristic, B., Mihaylova, L., and Abdallah, F. (2013). An introduction to box particle filtering. *IEEE Signal Processing Magazine*, 30(4):166–171.

[Goldsztejn, 2005] Goldsztejn, A. (2005). A Right-Preconditioning Process for the Formal-Algebraic Approach to Inner and Outer Estimation of AE-solution Sets. *Reliable Computing*, 11(6):443–478.

[Goldsztejn, 2006] Goldsztejn, A. (2006). A branch and prune algorithm for the approximation of non-linear AE-solution sets Generalized intervals. In *Proceedings of the 21st ACM Symposium on Applied Computing track Reliable Computations and their Applications, Dijon, France, April 2006 (SAC 2006)*, pages 1650–1654.

[Goldsztejn and Chabert, 2006] Goldsztejn, A. and Chabert, G. (2006). On the approximation of linear AE-solution sets. In *12th International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics, Duisburg, Germany, (SCAN 2006).*

[Goualard, 2008] Goualard, F. (2008). Fast and Correct SIMD Algorithms for Interval Arithmetic. In Springer, editor, *9th International Workshop on State-of-the-Art in Scientific and Parallel Computing (PARA '08)*, volume 6126-6127. aaa.

[Grimson and Lozano-Perez, 1985] Grimson, W. and Lozano-Perez, T. (1985). Recognition and localization of overlapping parts from sparse data in two and three dimensions. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 61–66.

[Guyonneau et al., 2013] Guyonneau, R., Lagrange, S., and Hardouin, L. (2013). A visibility information for multi-robot localization. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1426–1431.

[Herréro, 2006] Herréro, P. (2006). *Quantified real constraint solving using modal intervals with applications to control.* Phd dissertation, University of Girona, Spain.

[Herrero et al., 2005] Herrero, P., Sainz, M. A., Veh, J., and Jaulin, L. (2005). Quantified Set Inversion Algorithm with Applications to Control. *Reliable Computing*, 11(5):369–382.

[Hidalgo and Braunl, 2015] Hidalgo, F. and Braunl, T. (2015). Review of underwater SLAM techniques. *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 306–311.

[Ho and Newman, 2006] Ho, K. L. and Newman, P. (2006). Loop closure detection in SLAM by combining visual and spatial appearance. *Robotics and Autonomous Systems*, 54(9):740–749.

[Jaulin, 1994] Jaulin, L. (1994). *Solution globale et garantie de problèmes ensemblistes ; application à l'estimation non linéaire et à la commande robuste.* PhD dissertation, Université Paris-Sud, Orsay, France.

[Jaulin, 2001] Jaulin, L. (2001). Path planning using intervals and graphs. *Reliable Computing*, 7(1):1–15.

[Jaulin, 2009] Jaulin, L. (2009). Robust set-membership state estimation; application to underwater robotics. *Automatica*, 45(1):202–206.

[Jaulin, 2011] Jaulin, L. (2011). Range-only SLAM with occupancy maps: A set-membership approach. *IEEE Transactions on Robotics*, 27(5):1004–1010.

[Jaulin, 2012] Jaulin, L. (2012). Solving set-valued constraint satisfaction problems. *Computing*, 94(2):297–311.

[Jaulin, 2016] Jaulin, L. (2016). Range-only SLAM with indistinguishable landmarks; a constraint programming approach. *Constraints*, 21(4):557–576.

[Jaulin and Desrochers, 2014] Jaulin, L. and Desrochers, B. (2014). Introduction to the algebra of separators with application to path planning. *Engineering Applications of Artificial Intelligence*, 33(0):141–147.

[Jaulin and Desrochers, 2016] Jaulin, L. and Desrochers, B. (2016). Thick separators. In *COPROD'16*, Uppsala (Sweden).

[Jaulin et al., 2016] Jaulin, L., Desrochers, B., and Massé, D. (2016). Bisectable Abstract Domains for the Resolution of Equations Involving Complex Numbers. *Reliable Computing*, 23(1):35–46.

[Jaulin et al., 2001a] Jaulin, L., Kieffer, M., Braems, I., and Walter, E. (2001a). Guaranteed non-linear estimation using constraint propagation on sets. *International Journal of Control*, 74(18):1772–1782.

[Jaulin et al., 2001b] Jaulin, L., Kieffer, M., Didrit, O., and Eric, W. (2001b). *Applied Interval Analysis (Springer).pdf*. Springer London, London.

[Jaulin et al., 2002] Jaulin, L., Kieffer, M., Walter, E., and Meizel, D. (2002). 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.

[Jaulin and Walter, 1993a] Jaulin, L. and Walter, E. (1993a). Guaranteed nonlinear estimation and robust stability analysis via set inversion. In *Proceedings of the 2nd European Control Conference*, pages 818–821.

[Jaulin and Walter, 1993b] Jaulin, L. and Walter, E. (1993b). Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis. *Mathematics and Computers in Simulation*, 35(2):123–137.

[Jaulin and Walter, 1993c] Jaulin, L. and Walter, E. (1993c). Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064.

[Jaulin and Walter, 1999] Jaulin, L. and Walter, E. (1999). Guaranteed parameter bounding for nonlinear models with uncertain experimental factors. *Automatica*, 35(5):849–856.

[Jaulin and Walter, 2002] Jaulin, L. and Walter, E. (2002). Guaranteed robust nonlinear minimax estimation. *IEEE Transactions on Automatic Control*, 47(11):1857–1864.

[Jaulin et al., 2000] Jaulin, L., Walter, E., Lévêque, O., and Meizel, D. (2000). Set inversion for χ-algorithms, with application to guaranteed robot localization. *Mathematics and Computers in Simulation*, 52(3-4):197–210.

[Jensfelt and Christensen, 2001] Jensfelt, P. and Christensen, H. I. (2001). Pose tracking using laser scanning and minimalistic environmental models. *IEEE Transactions on Robotics and Automation*, 17(2):138–147.

[Kalman, 1960] Kalman, R. E. (1960). Contributions to the Theory of Optimal Control. *Bolentin de la Sociedad Matematica Mexicana*, 5:102–119.

[Kaucher, 1980] Kaucher, E. (1980). Interval analysis in the extended interval space IR. *Computing*, 2:33–49.

[Kieffer et al., 1999] Kieffer, M., Jaulin, L., Walter, E., and Meizel, D. (1999). Guaranteed Mobile Robot Tracking Using Interval Analysis. In *Proceedings of the MISC'99 Workshop on Applications of Interval Analysis to Systems and Control*, pages 347–360, Girona, Spain.

[Koditschek, 1987] Koditschek, D. E. (1987). Exact robot navigation by means of potential functions: some topological considerations. In *"Proceedings of the IEEE International Conference on Robotics and Automation"*, pages 1–6, Raleigh.

[Krantz, 2012] Krantz, S. G. (2012). The Index or Winding Number of a Curve about a Point." §4.4.4. In *Handbook of Complex Variables*, pages 49–50. Birkhäuser Boston, Boston, MA.

[Kreinovich et al., 1997] Kreinovich, V., Lakeyev, A. V., Rohn, J., and Kahl, P. T. (1997). Computational Complexity and Feasibility of Data Processing and Interval Computations. *Reliable Computing*, 4(4):405–409.

[Kreinovich and Shary, 2016] Kreinovich, V. and Shary, S. (2016). Interval Methods for Data Fitting under Uncertainty: A Probabilistic Treatment. *Reliable Computing*, 23:105–140.

[Kulpa, 1994] Kulpa, Z. (1994). Diagrammatic representation and reasoning. *Machine Graphics and Vision*, 3(1/2):77–103.

[Langerwisch, 2014] Langerwisch, M. (2014). *Kartierung und Lokalisation durch mobile Serviceroboter unter der Annahme unbekannter aber begrenzter Sensorfehler.* Schriftenreihe Technische Forschungsergebnisse. Kovac, Dr. Verlag.

[Langerwisch and Wagner, 2012] Langerwisch, M. and Wagner, B. (2012). Guaranteed mobile robot tracking using robust interval constraint propagation. *Intelligent Robotics and Applications, Springer*, 7507:354–365.

[Langerwisch and Wagner, 2013] Langerwisch, M. and Wagner, B. (2013). Building variable resolution occupancy maps assuming unknown but bounded sensor errors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo Big Sight, Japan.

[Laumond, 1986] Laumond, J. P. (1986). Feasible trajectories for mobile robots with kinematic and environment constraints. In *Proceedings of the International Conference on Intelligent Autonomous Systems*, pages 246–354, Amsterdam, the Netherlands.

[Le Bars et al., 2010] Le Bars, F., Bertholom, A., Sliwka, J., and Jaulin, L. (2010). Interval SLAM for underwater robots; A new experiment. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, pages 42–47, Italy.

[Le Bars et al., 2012] Le Bars, F., Sliwka, J., Jaulin, L., and Reynet, O. (2012). Set-membership state estimation with fleeting data. *Automatica*, 48(2):381–387.

[Leblond et al., 2005] Leblond, I., Legris, M., and Solaiman, B. (2005). Use of classification and segmentation of sidescan sonar images for long term registration. In *Europe Oceans 2005*, pages 322–327 Vol. 1. IEEE.

[Leonard and Durrant-Whyte, 1992] Leonard, J. J. and Durrant-Whyte, H. F. (1992). *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer, Boston.

[Leonard et al., 1992] Leonard, J. J., Durrant-Whyte, H. F., and Cox, I. J. (1992). Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4):286–289.

[Leonard et al., 2002] Leonard, J. J., Rikoski, R. J., Newman, P. M., and Bosse, M. (2002). Mapping Partially Observable Features from Multiple Uncertain Vantage Points. *The International Journal of Robotics Research*, 21(10-11):943–975.

[Lerch et al., 2006] Lerch, M., Tischler, G., Gudenberg, J. W. V., Hofschuster, W., and Krämer, W. (2006). FILIB++, a Fast Interval Library Supporting Containment Computations. *ACM Trans. Math. Softw.*, 32(2):299–324.

[Lévêque et al., 1997] Lévêque, O., Jaulin, L., Meizel, D., and Walter, E. (1997). Vehicle Localization from Inaccurate Telemetric Data: A Set-Inversion Approach. In *IFAC Proceedings Volumes*, volume 30, pages 173–179, Nantes, France.

[Lozano-Perez, 1981] Lozano-Perez, T. (1981). Automatic Planning of Manipulator Transfer Movements. *"IEEE Transactions on Systems Man and Cybernetics"*, 11(10):681–698.

[Lozano-Perez, 1983] Lozano-Perez, T. (1983). Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers*, 32(2):108–120.

[Mackworth, 1977] Mackworth, A. K. (1977). Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118.

[Malti et al., 2011] Malti, R., Moreau, X., Khemane, F., and Oustaloup, A. (2011). Stability and resonance conditions of elementary fractional transfer functions. *Automatica*, 47(11):2462–2467.

[McPhail, 2009] McPhail, S. (2009). Autosub6000: A Deep Diving Long Range AUV. *Journal of Bionic Engineering*, 6(1):55–62.

[Meizel et al., 2002] Meizel, D., Lévêque, O., Jaulin, L., and Walter, E. (2002). Initial localization by set inversion. *IEEE Transactions on Robotics and Automation*, 18(6):966–971.

[Melo and Matos, 2017] Melo, J. and Matos, A. (2017). Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Engineering*, 139(April):250–264.

[Messine, 2008] Messine, F. (2008). Deterministic global optimization using interval constraint propagation techniques. *RAIRO Operations Research*, 42(3):415–431.

[Molchanov, 2005a] Molchanov, I. (2005a). *The Theory of Random Sets*. Springer, New York.

[Molchanov, 2005b] Molchanov, I. (2005b). *Theory of Random Sets*. Probability and Its Applications. Springer-Verlag, London.

[Montanari and Rossi, 1991] Montanari, U. and Rossi, F. (1991). Constraint relaxation may be perfect. *Artificial Intelligence*, 48(2):143–170.

[Montemerlo et al., 2003] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2003). {FastSLAM 2.0}: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *Proceedings of the Intl. Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 1151–1156.

[Moore, 1966] Moore, R. E. (1966). *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ.

[Moore, 1979] Moore, R. E. (1979). *Methods and {A}pplications of {I}nterval {A}nalysis*. SIAM, Philadelphia, PA.

[Moore, 1992] Moore, R. E. (1992). Parameter sets for bounded-error data. *Mathematics and Computers in Simulation*, 34(2):113–119.

[Mullane et al., 2011] Mullane, J., Vo, B.-N., Adams, M., and Vo, B.-T. (2011). *Random Finite Sets for Robot Mapping and SLAM*, volume 72 of *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Najman and (Eds), 1000] Najman, L. and (Eds), H. T. (1000). *Mathematical morphology: from theory to applications*. ISTE-Wiley.

[Neira and Tardos, 2001] Neira, J. and Tardos, J. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897.

[Ó'Dúnlaing and Yap, 1985] Ó'Dúnlaing, C. and Yap, C. K. (1985). A "retraction" method for planning the motion of a disc. *Journal of Algorithms*, 6(1):104–111.

[Olivier et al., 2013] Olivier, M., Goubault, É., Kieffer, M., and Putot, S. (2013). General inner approximation of vector-valued functions. *Reliable Computing*, 18(0):117–143.

[O'Rourke, 1998] O'Rourke, J. (1998). *Computational Geometry in C*. Cambridge University Press, 2nd edition.

[Paull, 2013] Paull, L. (2013). *Robust Online Adaptive Sensor-Driven Survey Planning for Single and Multiple Autonomous Underwater Vehicles*. Phd dissertation, University of New Brunswick.

[Pawlak, 1992] Pawlak, Z. (1992). *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, Norwell, MA, USA.

[Picard et al., 2015] Picard, L., Baussard, A., Le Chenadec, G., and Quidu, I. (2015). Potential of the intrinsic dimensionality for characterizing the seabed in the ATR context. In *OCEANS 2015 - Genova*, pages 1–7. IEEE.

[Pomerleau et al., 2015] Pomerleau, F., Colas, F., and Siegwart, R. (2015). A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Found. Trends Robot*, 4(1):1–104.

[Porta et al., 2007] Porta, J. M., Cortés, J., Ros, L., and Thomas, F. (2007). A space decomposition method for path planning of loop linkages. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1882–1888.

[Ramdani and Poignet, 2005] Ramdani, N. and Poignet, P. (2005). Robust dynamic experimental identification of robots with set membership uncertainty. *IEEE/ASME Transactions on Mechatronics*, 10(2):253–256.

[Revol et al., 2005] Revol, N., Makino, K., and Berz, M. (2005). Taylor models and floating point arithmetic: proof that arithmetic operations are bounded in Cosy. *The Journal of Logic and Algebraic Programming*, 64:135–154.

[Revol and Rouillier, 2005] Revol, N. and Rouillier, F. (2005). Motivations for an Arbitrary Precision Interval Arithmetic and the MPFI Library. *Reliable Computing*, 11(4):275–290.

[Rohou, 2017] Rohou, S. (2017). *Reliable robot localization: a constraint programming approach over dynamical systems.* PhD thesis.

[Rohou et al., 2017] Rohou, S., Jaulin, L., Mihaylova, L., Le Bars, F., and Veres, S. M. (2017). Guaranteed computation of robot trajectories. *Robotics and Autonomous Systems*, 93:76–84.

[Rokityanski and Veres, 2005] Rokityanski, D. and Veres, S. (2005). Application of ellipsoidal estimation to satellite control. *Mathematical and Computer Modelling of Dynamical Systems*, 11(2):239–249.

[Shary, 1995] Shary, S. P. (1995). On Optimal Solution of Interval Linear Equations. *SIAM Journal on Numerical Analysis*, 32(2):610–630.

[Shoji, 1991] Shoji, K. (1991). Quadtree decomposition of binary structuring elements. In *Nonlinear Image Processing II.* Proc. SPIE 1451.

[Sliwka, 2011] Sliwka, J. (2011). *Using set membership methods for robust underwater robot localization.* PhD thesis, Ensta Bretagne.

[Taha et al., 2015] Taha, W., Duracz, A., Zeng, Y., Atkinson, K., Bartha, F., and Ames, A. (2015). Acumen : An Open-source Testbed for Cyber- Physical Systems Research. In *International Internet of Things Summit*, pages 118–130.

[Tapia, 2011] Tapia, E. (2011). A note on the computation of high-dimensional integral images. *Pattern Recognition Letters*, 32(2):197–201.

[Thrun, 1998] Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71.

[Thrun, 2002] Thrun, S. (2002). *Probabilistic robotics*, volume 45. MIT Press, Cambridge, M.A.

[Thrun, 2003] Thrun, S. (2003). Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, 15(2):111–127.

[Thrun and Montemerlo, 2005] Thrun, S. and Montemerlo, M. (2005). The Graph-SLAM Algorithm With Applications to Large-Scale Mapping of Urban Structures. *International Journal on Robotics Research*, 25(5/6):403–430.

[Tucker, 1999] Tucker, W. (1999). The Lorenz attractor exists. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 328(12):1197–1202.

[Tucker, 2002] Tucker, W. (2002). A Rigorous ODE Solver and Smale's 14th Problem. *Foundations of Computational Mathematics*, 2(1):53–117.

[Vehi et al., 1997] Vehi, J., Armengol, J., Rodellar, J., and Sainz, M. A. (1997). Using Interval Methods for Control Systems Design in the Parameter Space. In *IFAC Proceedings Volumes*, volume 30, pages 353–357, Gent.

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–511–I–518. IEEE Comput. Soc.

[Walter and Piet-Lahanier, 1989] Walter, E. and Piet-Lahanier, H. (1989). Exact recursive polyhedral description of the feasible parameter set for bounded-error models. *IEEE Transactions on Automatic Control*, 34(8):911–915.

[Wan et al., 2009] Wan, J., Vehi, J., and Luo, N. (2009). A numerical approach to design control invariant sets for constrained nonlinear discrete-time systems with guaranteed optimality. *"Journal of Global Optimization"*, 44:395–407.

[Wilczak and Zgliczynski, 2011] Wilczak, D. and Zgliczynski, P. (2011). Cr-Lohner algorithm. *Schedae Informaticae*, 20:9–46.

[Yao et al., 2008] Yao, J. T., Yao, Y. Y., Kreinovich, V., da Silva, P. P., Starks, S. A., Xiang, G., and Nguyen, H. T. (2008). Towards More Adequate Representation of Uncertainty: From Intervals to Set Intervals, with the Possible Addition of Probabilities and Certainty Degrees. In *Proceedings of the IEEE World Congress on Computational Intelligence WCCI'2008*, pages 983–990, Hong Kong, China.

[Yao and Li, 1996] Yao, Y. Y. and Li, X. (1996). Comparison of Rough-set and Interval-set Models for Uncertain Reasoning. *Fundamenta Informaticae*, 27(2):289–298.

[Zitova and Flusser, 2003] Zitova, B. and Flusser, J. (2003). Image registration methods : a survey. *Image (Rochester, N.Y.)*, 21(11):977–1000.

List of Figures

List of Tables

Résumé

Cette thèse étudie le problème de la localisation et de la cartographie simultanée (SLAM), dans des environnements non structurés, c'est-à-dire, qui ne peuvent pas être décrits par des équations ou des formes géométriques. Ces types d'environements sont souvent rencontrés dans le domaine sous-marin.

Contrairement aux approches classiques, l'environnement n'est pas modélisé par une collection de descripteurs ou d'amers ponctuels, mais directement par des ensembles. Ces ensembles, appelés forme ou shape, sont associés à des caractéristiques physiques de l'environnement, comme par exemple, des textures, du relief ou, de manière plus symbolique, à l'espace libre autour du véhicule.

D'un point de vue théorique, le problème du SLAM, basé sur des formes, est formalisé par un réseau de contraintes hybrides dont les variables sont des vecteurs de $\mathbb{R}^n$ et des sous-ensembles de $\mathbb{R}^n$. De la même façon que l'incertitude sur une variable réelle est représentée par un intervalle de réels, l'incertitude sur les formes sera représenté par un intervalle de forme. La principale contribution de cette thèse est de proposer un formalisme, basé sur le calcul par intervalle, capable de calculer ces domaines. En application, les algorithmes développés ont été appliqués au problème du SLAM à partir de données bathymétriques recueillies par un véhicule sous-marin autonome (AUV).

Mots-clés: analyse par intervalles, localisation, SLAM, robotique

---

Abstract

This thesis deals with the simultaneous localization and mapping (SLAM) problem in unstructured environments, *i.e.* which cannot be described by geometrical features. This type of environment frequently occurs in an underwater context.

Unlike classical approaches, the environment is not described by a collection of punctual features or landmarks, but directly by sets. These sets, called shapes, are associated with physical features such as the relief, some textures or, in a more symbolic way, the space free of obstacles that can be sensed around a robot.

In a theoretical point of view, the SLAM problem is formalized as an hybrid constraint network where the variables are vectors and subsets of $\mathbb{R}^n$. Whereas an uncertain real number is enclosed in an interval, an uncertain shape is enclosed in an interval of sets. The main contribution of this thesis is the introduction of a new formalism, based on interval analysis, able to deal with these domains. As an application, we illustrate our method on a SLAM problem based on bathymetric data acquired by an autonomous underwater vehicle (AUV).

Keyworks: SLAM, AUV, interval analysis, localisation