Constraints
DOI 10.1007/s10601-015-9231-9

APPLICATION

Range-only SLAM with indistinguishable landmarks; a constraint programming approach

Luc Jaulin[1]

Abstract This paper deals with the simultaneous localization and mapping problem (SLAM) for a robot. The robot has to build a map of its environment while localizing itself using a partially built map. It is assumed that (i) the map is made of point landmarks, (ii) the landmarks are indistinguishable, (iii) the only exteroceptive measurements correspond to the distance between the robot and the landmarks. This paper shows that SLAM can be cast into a constraint network the variables of which being trajectories, digraphs and subsets of $\mathbb{R}^n$. Then, we show how constraint propagation can be extended to deal with such generalized constraint networks. As a result, due to the redundancy of measurements of SLAM, we demonstrate that a constraint-based approach provides an efficient backtrack-free algorithm able to solve our SLAM problem in a guaranteed way.

Keywords Constraint programming · Interval analysis · Localization · Robotics · Simultaneous localization and mapping

1 Introduction

Simultaneous localization and mapping (SLAM) for a robot is the problem of building the environment (the *map*) while at the same time localizing itself inside that map [21] [28]. This paper deals with SLAM in the case where (i) the map is static; (ii) the map is made of indistinguishable point landmarks (or *mark* form short) and (iii) the marks are partially observable (i.e., multiple detections with different vehicle positions must be used to esti-

✉ Luc Jaulin
luc.jaulin@ensta-bretagne.fr

[1] LabSTICC, IHSEV, OSM, 2 rue François Verny, 29806, Brest, France

mate the position of a mark). SLAM is often referred to as a *chicken and egg* problem. A *chicken and egg* problem of cardinality n is a mixture of n interconnected problems such that if all problems are solved except one, the unsolved problem can be solved using known techniques. Since, here, the marks are indistinguishable, our SLAM problem can be seen as a chicken and egg problem of cardinality three: (i) if both the map and mark association (i.e., the matching between mark detections) are solved, finding the robot trajectory can be performed using any localization method, (ii) if the trajectory and the mark association are solved, we can build the map and (iii) if the trajectory of the robot and the map are known we can easily provide the association between marks. Fixed point methods are good candidates to deal efficiently with chicken and egg problems: fixed-point operators are called successively until no more improvement can be observed. Unfortunately, it is often impossible to guarantee that the process will converge and in case of convergence, it is difficult to predict that the fixed point will be a solution of our problem. When the conditions of the Tarski fixed point theorem [27] apply, fixed-point methods always converge. The Tarski theorem states that if $(\mathcal{L}, \leq)$ is a complete lattice and $f : \mathcal{L} \to \mathcal{L}$ is a monotonic function (i.e., $a \leq b \Rightarrow f(a) \leq f(b)$) which is Scott continuous (i.e., for each decreasing sequence x_k, we have $f(\lim x_k) = \lim f(x_k)$), then the sequence $x_{k+1} = f(x_k)$, where x_0 is the greatest element of $\mathcal{L}$, converges to the greatest fixed point of f in $\mathcal{L}$. Although this theorem seems to be far from our SLAM problem, it is an important element of the theoretical foundation of constraint propagation [2] [32] [31], which has already been used successfully to solve SLAM [12], [8] [14] and range-only localization problems [25]. The principle of constraint propagation is to apply polynomial-complexity contraction operators (called *contractors*) over domains containing the unknown variables. The contractors are called to shrink these domains until no more contraction can be observed. The approach is particularly efficient in case of redundant information [13] and provides guaranteed results.

In our SLAM problem, the unknown variables have an heterogenous nature: the marks are vectors of $\mathbb{R}^q$ (where $q \in \{2, 3\}$), is the dimension of the world), the trajectories belong to the set of functions from $\mathbb{R} \to \mathbb{R}^n$, the free space is a subset of $\mathbb{R}^q$ and the mark associations can be represented by a graph $\mathcal{G}$. This paper shows that constraint propagation methods can easily deal with heterogenous variables and that solving the SLAM problem amounts to use a fixed point method with specific heterogeneous contractors. The resulting method will inherit properties of constraint propagation tools: a backtrack-free algorithm (i.e., without combinatorial search) which provides guaranteed results. To illustrate the principle of what we call *heterogeneous constraint propagation*, we will consider the range-only SLAM (see [23] for probabilistic technics or [26] for set-membership approaches) in the case of distinguishable marks. The problem we consider in this paper is a difficult representative of the class of SLAM problems: the marks are partially observable (due to range-only measurements), the marks are indistinguishable and the collected ranges are huge in comparison to the movement of the robot between sequential measurements, which makes the problem inherently ill conditioned. Here, contrary to the range-only applications commonly treated in literature (except in [7] where the number of marks is known, which is not assumed here), the mark association problem will have to be solved.

The paper is organized as follows. Section 2 proposes a simple and new formulation for SLAM with indistinguishable marks. Section 3 presents the basic notions of heterogeneous constraint propagation. The resolution method is then presented in Section 4. Section 5 provides an illustration of the approach on a range-only SLAM problem with indistinguishable marks. Two test-cases are presented. The first one is based on simulations whereas the second one deals with actual data collected by an underwater robot. Section 6 concludes the paper.

2 Problem statement

Notations In this paper, the bold font is used to represent vectors of $\mathbb{R}^n$ (e.g., $\mathbf{a}$, $\mathbf{b}$) and the capital blackboard bold font is used to represent subsets of $\mathbb{R}^n$ (e.g., $\mathbb{A}$, $\mathbb{B}$). The ith component of a vector $\mathbf{a}$ is denoted by a_i. A *trajectory* $\mathbf{x}$ is a smooth function from $\mathbb{R} \to \mathbb{R}^n$ where $n \geq 1$. It will be denoted indifferently by $\mathbf{x}$ or $\mathbf{x}(\cdot)$. The vector $\mathbf{x}(t) \in \mathbb{R}^n$ is the value of $\mathbf{x}$ at time $t \in \mathbb{R}$. The derivative of the trajectory $\mathbf{x}$ with respect to t is denoted by $\mathbf{\dot{R}}$. Intervals are denoted between brackets.

Consider a robot the evolution of which is described by state equations. We assume that the initial state vector $\mathbf{x}(0)$ of the robot is known. This assumption is not restrictive since for SLAM, the environment is totally unknown and thus the origin and the orientation of the world frame can be fixed as the pose (i.e., the position and the orientation) of the robot at time $t = 0$. In the environment, we assume that there exist indistinguishable marks represented by the set $\mathcal{M} = \{\mathbf{m}(1), \mathbf{m}(2), \ldots\}$ of vectors of $\mathbb{R}^q$, where q is the dimension of the environment (typically two or three). The number of marks and their location are unknown and we may have an infinite number of marks. A *sector* is a subset of $\mathbb{R}^q$ which contains a single mark. For some time samples $t_1, t_2, \ldots, t_m$, we assume that the robot is able to collect sectors via exteroceptive sensors (like cameras or sonars). Note that these sectors are expressed with respect to a frame of reference attached to the robot (i.e., their coordinates are egocentric) and not in the world frame. More formally, we describe our SLAM problem as follows

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) & \text{(evolution equation)} \\ (t_i, \mathcal{H}_i(\mathbf{x})) & \text{(sector list)} \end{cases} \tag{1}$$

where $\mathbf{x}(t)$ is the state vector of the robot at time t, $\mathbf{u}(t)$ is the input vector (i.e., the actuators), $\mathbf{f}$ is the evolution function and $t \in [0, t_{\max}]$ is the time. For all $t \in [0, t_{\max}]$, we assume that boxes $[\mathbf{u}](t)$ enclosing the vectors $\mathbf{u}(t)$ are available. For $i \in \{1, \ldots, m\}$, the set-valued functions

$$\mathcal{H}_i : \begin{cases} \mathbb{R}^n & \to \mathcal{P}(\mathbb{R}^q) \\ \mathbf{x} & \mapsto \mathcal{H}_i(\mathbf{x}), \end{cases} \tag{2}$$

where $\mathcal{P}(\mathbb{R}^q)$ is the set of all subsets of $\mathbb{R}^q$, are called *sector functions*. Each sector $\mathbb{H}_i = \mathcal{H}_i(\mathbf{x}(t_i)) \subset \mathbb{R}^q$ of the world frame is known to contain one and only one mark. Note that since $\mathbf{x}(t_i)$ is unknown, the sector $\mathbb{H}_i$ is also unknown.

Example 1 Consider a robot moving in a plane and located at coordinates (x_1, x_2). If at time t_3 the robot detects that there exists one and only one mark at a distance d inside $[4, 5]$ meters, the associated sector function is

$$\mathcal{H}_3(\mathbf{x}) = \left\{ \mathbf{a} \in \mathbb{R}^2 \mid (x_1 - a_1)^2 + (x_2 - a_2)^2 \in [16, 25] \right\}. \tag{3}$$

Example 2 Figure 1 represents two photos collected at different times by the camera of a robot moving inside a 2D environment where the marks are highlighters. On the left picture, the robot detects one mark and represents this information by two sectors (the circle and the square). Since each sector contains a single mark, the difference between the square and the circle does not contain any mark. This may be used to solve the mark association (i.e., to know if the mark detected at time t_i corresponds to the mark detected at time t_j). On the right picture, the robot detects two marks. Since a sector contains a unique mark, we need four sectors to represent this double detection. For the type sensor considered in this example, the marks are observable because when the robot sees one mark, it is able to

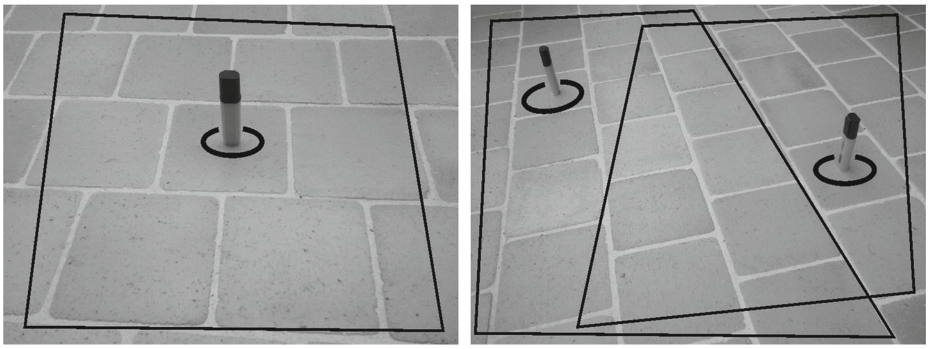

Fig. 1 Left: the robot detects one mark and represents this information by two sectors; Right: the robot detects two marks and uses four sectors

localize this mark in its own frame with some uncertainties. This was not the case in the previous example.

Example 3 Figure 2 illustrates a situation where the robot has detected a single mark (black star in the ring) inside the sector $\mathbb{H}_i$ (here a ring). This sector may have been obtained using an omnidirectional sonar, for instance. The visibility disk corresponds to the sector $\mathbb{H}_j$. Since in $\mathbb{H}_j$ there exists a unique mark, we conclude that the set $\mathbb{H}_j \backslash \mathbb{H}_i$ cannot contain any mark. In the figure, $\mathbb{H}_j \backslash \mathbb{H}_i$ has two connected components: the white disk around the robot

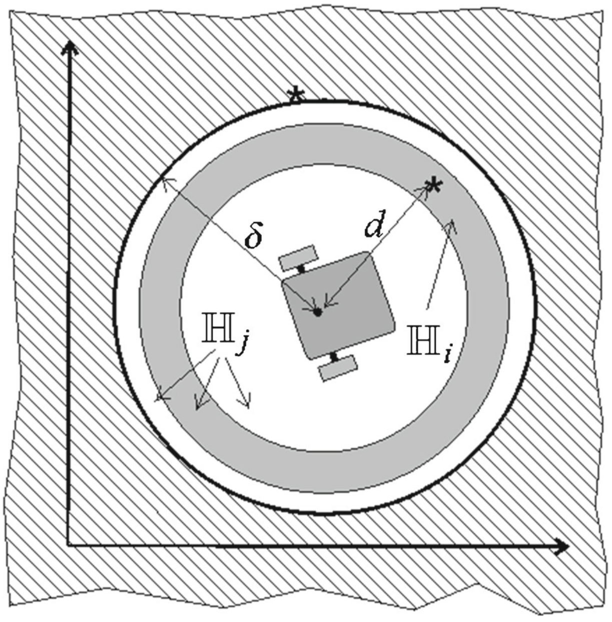

Fig. 2 The robot has detected the mark inside the ring. The other mark (which is not in the visibility disk of radius δ), is not detected

and the white ring. The nested pair $\left(\mathbb{H}_i, \mathbb{H}_j\right)$ represents both a zone with a detected mark, but also a zone without any mark. Note that due to the fact that $\mathbb{H}_i \subset \mathbb{H}_j$, we can associate the mark in $\mathbb{H}_i$ to the mark in $\mathbb{H}_j$. Both correspond to the star in the ring.

In the framework of this new formulation for SLAM, the following proposition shows how marks that have been detected at different times can be associated.

Proposition 1 *Consider a set of marks* $\mathcal{M} \subset \mathbb{R}^q$. *Define the free space as* $\mathbb{F} = \{\mathbf{p} \in \mathbb{R}^q \mid \mathbf{p} \notin \mathcal{M}\}$. *Consider m sectors* $\mathbb{H}_1, \ldots, \mathbb{H}_m$, *each of them containing exactly one mark and denote by* $\mathbf{a}(i)$ *the unique element in* $\mathcal{M} \cap \mathbb{H}_i$. *We have*

$$\begin{array}{ll} \text{(i)} & \mathbb{H}_i \subset \mathbb{H}_j \Rightarrow \mathbf{a}(i) = \mathbf{a}(j) \\ \text{(ii)} & \mathbb{H}_i \cap \mathbb{H}_j = \emptyset \Rightarrow \mathbf{a}(i) \neq \mathbf{a}(j) \\ \text{(iii)} & \mathbb{H}_i \subset \mathbb{H}_j \Rightarrow \mathbb{H}_j \backslash \mathbb{H}_i \subset \mathbb{F}. \end{array} \tag{4}$$

Proof Let us first prove (i). Denote by $\{\mathbf{a}(i)\}$ and $\{\mathbf{a}(j)\}$ the singletons containing $\mathbf{a}(i)$ and $\mathbf{a}(j)$. We have

$$\begin{aligned} \{\mathbf{a}(i)\} &= \mathcal{M} \cap \mathbb{H}_i \\ &= \mathcal{M} \cap \mathbb{H}_i \cap \mathcal{M} \cap \mathbb{H}_j \quad (\text{since } \mathbb{H}_i \subset \mathbb{H}_j) \\ &= \{\mathbf{a}(i)\} \cap \{\mathbf{a}(j)\}. \end{aligned} \tag{5}$$

Thus $\mathbf{a}(i) = \mathbf{a}(j)$. Let us now prove (ii)

$$\begin{aligned} \{\mathbf{a}(i)\} \cap \{\mathbf{a}(j)\} &= \mathcal{M} \cap \mathbb{H}_i \cap \mathcal{M} \cap \mathbb{H}_j \\ &= \emptyset \qquad\qquad (\text{since } \mathbb{H}_i \cap \mathbb{H}_j = \emptyset). \end{aligned} \tag{6}$$

Finally, let us prove (iii). If $\mathbb{H}_i \subset \mathbb{H}_j$, from (i), we have $\mathbf{a}(i) = \mathbf{a}(j)$, i.e., $\{\mathbf{a}(j)\} \setminus \{\mathbf{a}(i)\} = \emptyset$ or equivalently

$$\left(\mathbb{H}_j \cap \mathcal{M}\right) \setminus \left(\mathbb{H}_i \cap \mathcal{M}\right) = \left(\mathbb{H}_j \backslash \mathbb{H}_i\right) \cap \mathcal{M} = \emptyset.$$

Since $\mathbb{F}$ is the complement of $\mathcal{M}$, we get $\mathbb{H}_j \backslash \mathbb{H}_i \subset \mathbb{F}$. □

Example 4 Figure 3 represents 6 sectors $\mathbb{H}_1, \ldots, \mathbb{H}_6$ in the world frame. Each of them contains one and only one mark. Since $\mathbb{H}_3 \subset \mathbb{H}_1$, $\mathbb{H}_2 \subset \mathbb{H}_1$, $\mathbb{H}_3 \subset \mathbb{H}_5$, from (i) we conclude that $\mathbb{H}_1, \mathbb{H}_2, \mathbb{H}_3, \mathbb{H}_5$ all contain the same mark and that this mark belongs to $\mathbb{H}_1 \cap \mathbb{H}_2 \cap \mathbb{H}_3 \cap \mathbb{H}_5$. From (iii) we know that the hatched area cannot contain any mark. By subtracting the hatched area from $\mathbb{H}_4$, we conclude that $\mathbb{H}_4 \backslash \mathbb{H}_5$ (the black zone on the right) contains a single mark. Note that in our SLAM context, since exteroceptive measurements are egocentric, the $\mathbb{H}_i$ are only known through the sector functions $\mathcal{H}_i(\mathbf{x})$ and the state $\mathbf{x}$ is only approximately known.

Association graph Consider m mark detections $\mathbf{a}(1), \ldots, \mathbf{a}(m)$. We define the *association graph* as the graph with vertices $\mathbf{a}(i)$ such that one edge between $\mathbf{a}(i)$ and $\mathbf{a}(j)$ means that $\mathbf{a}(i) = \mathbf{a}(j)$.

Remark 1 When the robot is able to select a silent zone, i.e., a zone with no mark, we cannot take this information into account with our model. An extended model that could be considered is the following:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) & \text{(evolution equation)} \\ \left(t_i, \mathcal{H}_i(\mathbf{x}), \eta_i\right) & \text{(sector list)} \end{cases}$$

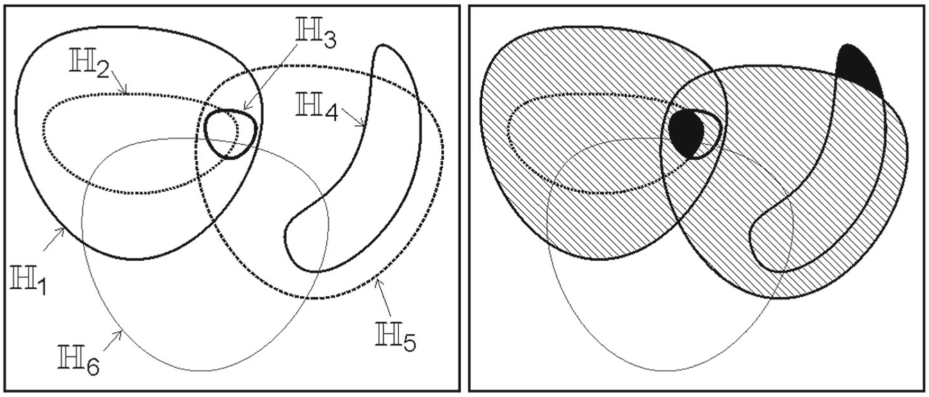

Fig. 3 From the configuration of the six sectors $\mathbb{H}_1, \ldots, \mathbb{H}_6$, we conclude that each of the two black zones contains a single mark and that no mark exists in the hatched area. Nothing can be concluded about the white area

where η_i is the exact number of marks inside the sector $\mathcal{H}_i(\mathbf{x})$. When no mark exists in $\mathcal{H}_i(\mathbf{x})$, then $\eta_i = 0$.

3 Heterogeneous constraint propagation

Constraint propagation [3] has been used in the context of finite domains for 40 years (see, *e.g.*, [33]). For 20 years, it has also been used as a numerical tool to solve nonlinear problems involving real numbers [32]. In most applications that can be found in the literature, the unknown variables are Boolean numbers, integers or real numbers. Constraint propagation has recently been extended to deal with unknown variables that can be trajectories [20] or subsets of $\mathbb{R}^n$ [15], as required for SLAM. This section recalls the principle of constraint propagation and extends the technique to deal with problems where the unknown variables are graphs, trajectories or subsets or $\mathbb{R}^n$, as needed to solve our SLAM problem. It also presents the notion of graph intervals, trajectory intervals (or tube) and set intervals.

3.1 Lattices

Constraint propagation can be applied as soon as the set of domains for the variables has a lattice structure. A *lattice* $(\mathcal{E}, \leq)$ is a partially ordered set, closed under least upper and greatest lower bounds [9]. 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 $x \wedge y$.

Example 5 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 $x_i \vee y_i = \max(x_i, y_i)$.

Example 6 If $\mathbb{E}$ is any set, the powerset $\mathcal{P}(\mathbb{E})$ of all subsets of $\mathbb{E}$ is a complete lattice with respect to the inclusion $\subset$. The meet operator corresponds to the intersection and the join to the union.

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, it is often possible to 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 $\vee\emptyset = \wedge\mathcal{E}$. The Cartesian product of two lattices $(\mathcal{E}_1, \leq_1)$ and $(\mathcal{E}_2, \leq_2)$ is the lattice $(\mathcal{E}, \leq)$ defined as the set of all $(a_1, a_2) \in \mathcal{E}_1 \times \mathcal{E}_2$ with the order relation $(a_1, a_2) \leq (b_1, b_2) \Leftrightarrow ((a_1 \leq_1 b_1) \text{ and } (a_2 \leq_2 b_2))$.

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 $\mathcal{E}$. An interval is a sub-lattice 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 $[x] = \left[x^-, x^+\right]$ and $[y] = \left[y^-, y^+\right]$ of $\mathbb{I}\mathcal{E}$, we have:

$$\begin{aligned} [x] \wedge [y] &= \left[x^- \vee y^-, x^+ \wedge y^+\right] \\ [x] \vee [y] &= \left[x^- \wedge y^-, x^+ \vee y^+\right]. \end{aligned} \tag{7}$$

The meet $[x] \wedge [y]$ is called the *intersection* and will denoted by $[x] \cap [y]$. The join $[x] \vee [y]$ is called the *interval union* and will be denoted by $[x] \sqcup [y]$.

3.2 Heterogeneous contractors

Many problems of estimation, control or robotics can be represented by *constraint networks* [18]. A constraint network (see, e.g., [31]) is composed of a set of variables $\{x_1, \ldots, x_n\}$, a set of constraints $\{c_1, \ldots, c_m\}$ and a set of domains $\{\mathbb{X}_1, \ldots, \mathbb{X}_n\}$. The domains $\mathbb{X}_i$ should belong to a complete lattice $(\mathcal{L}_i, \subset)$. In the context of our paper, the domains will be (i) subsets of $\mathbb{R}^q$ to represent the location of the marks, (ii) tubes (or interval of trajectories) to represent the unknown trajectory and (iii) intervals of subsets of $\mathbb{R}^q$ to represent the sectors and the free space. Denote by $\mathcal{L}$ the Cartesian product of all $\mathcal{L}_i$'s, i.e., $\mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$. An element $\mathbb{X}$ of $\mathcal{L}$ is the Cartesian product of n elements of $\mathcal{L}_i$, (i.e., it satisfies $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_n$). The set $\mathbb{X}$ will be called *heterogeneous domain* and the $\mathbb{X}_i$'s are the components of $\mathbb{X}$. A *heterogeneous contractor* is an operator

$$\mathcal{C} : \begin{array}{l} \mathcal{L} \to \mathcal{L} \\ \mathbb{X} \mapsto \mathcal{C}(\mathbb{X}) \end{array} \tag{8}$$

which satisfies

$$\begin{array}{ll} \mathbb{X} \subset \mathbb{Y} \Rightarrow \mathcal{C}(\mathbb{X}) \subset \mathcal{C}(\mathbb{Y}) & \text{(monotonicity)} \\ \mathcal{C}(\mathbb{X}) \subset \mathbb{X} & \text{(contractance)} \end{array} \tag{9}$$

The set of (heterogeneous) contractors forms also a complete lattice. As a consequence, the meet (or intersection) and join (or union) can also be defined. This leads us to the contractor algebra [6] which can naturally be extended to the heterogeneous case. When all variables of the constraint network belong to $\mathbb{R}$, contractor techniques have been shown to be very powerful [29] [4]. One contribution of the paper is to show that these techniques can still be efficient when the unknown variables are heterogeneous, i.e., when the variables have a different nature (real numbers, trajectories, subsets of $\mathbb{R}^n$, ...).

Heterogeneous constraint propagation The principle is to associate to each constraint $c_j \in \{c_1, \ldots, c_m\}$ of a heterogeneous constraint network, a contractor $\mathcal{C}_j(\mathbb{X})$ which does not remove any $(x_1, \ldots, x_n)$ consistent with c_j. Then, we build the contractor $\mathcal{C} = \mathcal{C}_1 \circ \cdots \circ \mathcal{C}_m$. We apply the contractor $\mathcal{C}$ until no more contraction can be observed. From the Tarski's theorem, we conclude that the process converges toward the largest subdomain

$\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_n$ of the initial heterogeneous domain which cannot be contracted by any $\mathcal{C}_i$.

Remark 2 In the literature the domains for the variables of a constraint network are intervals except in the case of finite domains. The interval nature is not needed as soon as the set of domains has a structure of lattice. For our SLAM resolution, the domains for the marks will not be boxes (i.e., intervals of $\mathbb{R}^q$), but subsets of $\mathbb{R}^q$. The propagation will be still valid due to the lattice structure of the $\mathcal{P}(\mathbb{R}^q)$.

Intervals for a set $\mathcal{E}$ can be defined as soon as $\mathcal{E}$ is a complete lattice. As previously written, we could have intervals of real numbers, intervals of sets, etc. In this section, intervals of graphs [5] [10] are presented for the following reasons: (i) graph intervals have never been presented in the context of robotics, (ii) they will illustrate how contractors can be built from a constraint, and (iii) they have a fundamental role in the resolution of the mark association problem. Then, we will present tubes [20] and set intervals [16], which have already been used in robotics, to represent uncertain trajectories of robots or uncertain maps.

3.3 Graph intervals

Consider m nodes $\mathcal{A} = \{a_1, \dots, a_m\}$ (for our SLAM problem, these nodes correspond to the marks $\mathbf{a}_i$ detected at time t_i). A graph $\mathcal{G}$ of $\mathcal{A}$ is a subset of $\mathcal{A} \times \mathcal{A}$. A graph can either be represented graphically using nodes and arcs or by a $m \times m$-matrix (g_{ij}) of Boolean numbers. For instance, the following three propositions are equivalent (i) $(a_i, a_j) \in \mathcal{G}$ (ii) there exists an arc between node i and node j, (iii) $g_{ij} = 1$ (here 1 corresponds to true and 0 to false). If the set of graphs of $\mathcal{A}$ is equipped with the following order relation

$$\mathcal{G} \leq \mathcal{H} \Leftrightarrow \forall i, j \in \{1, \dots, m\}, \; g_{ij} \leq h_{ij}, \tag{10}$$

then it is easy to check that this set corresponds to a complete lattice. Intervals of graphs of $\mathcal{A}$ can thus be defined. Figure 4 represents a graph which belongs to a graph interval. The corresponding matrix relation is

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \in \begin{pmatrix} [0,1] & [0,1] & 0 \\ 1 & [0,1] & [0,1] \\ [0,1] & [0,1] & [0,1] \end{pmatrix}. \tag{11}$$

Consider a constraint on a graph $\mathcal{G}$ which belongs to a graph interval $[\mathcal{G}]$. A contractor associated to this constraint is able to contract $[\mathcal{G}]$ without removing a single graph which satisfies the constraint. Consider for instance the optimal contractor $\mathcal{C}$ associated with the constraint "$\mathcal{G}$ is an equivalence relation". Since, for SLAM, the mark association graph is

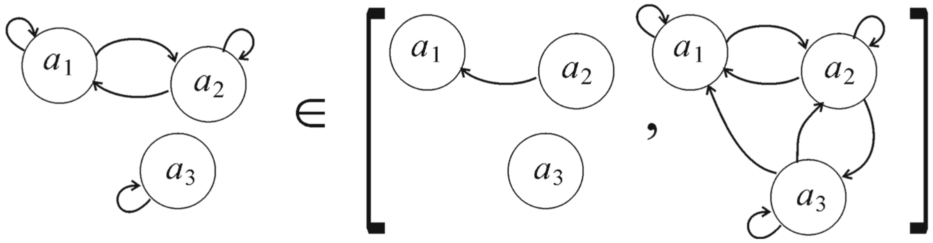

Fig. 4 The graph on the left belongs to the graph interval on the right

an equivalence relation, such a contractor will be used for the resolution of our problem. If we apply $\mathcal{C}$ to the graph interval given by the right hand side of relation (11), we get a degenerated interval which contains as a unique element the graph of the left hand side of (11).

3.4 Tubes

The set $\mathcal{F}$ of all functions from $\mathbb{R}$ to $\bar{\mathbb{R}}^n$ is a complete lattice with the following partial order $\mathbf{x}(\cdot) \le \mathbf{y}(\cdot) \Leftrightarrow \forall t \in \mathbb{R},\ \mathbf{f}(t) \le \mathbf{g}(t)$. An interval of $\mathcal{F}$ is called a *tube* [19]. Since tubes are defined as intervals of a complete lattice, the set of tubes $(\mathbb{I}\mathcal{F}, \subset)$ is also a lattice. Tubes make it possible to represent the uncertain trajectory of the robot [20]. When non empty, a tube $[\mathbf{x}](\cdot)$ can be represented by its lower and its upper bound:

$$[\mathbf{x}](\cdot) = \left[\mathbf{x}^-(\cdot), \mathbf{x}^+(\cdot)\right] = \left\{\mathbf{x}(\cdot) \ \mid \forall t, \mathbf{x}^-(t) \le \mathbf{x}(t) \le \mathbf{x}^+(t)\right\}. \tag{12}$$

Contractors associated with constraints on trajectories can be built. For instance, the minimal contractor associated with the constraint $\dot{x} = y$ can be obtained by achieving the forward interval integration:

$$[x](t) = [x](t) \cap ([x](t - dt) + dt \cdot [y](t)) \tag{13}$$

followed by the backward interval integration:

$$[x](t) = [x](t) \cap ([x](t + dt) - dt \cdot [y](t)). \tag{14}$$

No contraction can be obtained for $[y]$. If the constraint is given by a state equation of the form $\dot{x}(t) = f(x(t), u(t))$, a contractor can be obtained performing the decomposition

$$\begin{cases} \dot{x}(t) = y(t) \\ y(t) = f(x(t), u(t)) \end{cases} \tag{15}$$

and then by a composition of the two resulting contractors. Now, it is much more efficient to use specific interval integration methods (see, e.g., [34]).

In the computer, a tube is represented as a list of nonoverlapping boxes (or *slices*) in the $\mathbb{R}^n \times \mathbb{R}$ space. Figure 5a represents a scalar (i.e., $n = 1$) tube $[x](t)$ for $t \in [-2, 2]$ and for a sampling time $\delta = 0.1$. The tube $[x](t)$ is thus made with 40 slices. To illustrate how contractor-based techniques can be implemented in practice, assume that the tube $[x](t)$ contains an unknown function $x(t)$ which is periodic with a period of 1. This means that we have the constraint $\forall t,\ x(t) = x(t + 1)$ for the trajectory $x(\cdot)$. We now illustrate how the tube $[x](\cdot)$ can be contracted without removing any feasible $x(\cdot)$. We first compute the tube $[y](t) = [x](t + 1)$ (painted darkgrey in Fig. 5b). The large slices for $t \in [-2, -1]$ correspond to $[-\infty, \infty]$, due to the fact that nothing is known for $x(t)$, when $t < -2$. An intersection in thus performed between the two tubes $[x](t)$ and $[y](t)$ (see Fig. 5c). This operation is repeated for a left/right shift until no more contractions are observed. The fixed-point tube is painted darkgrey on Fig. 5d. On Fig. 5b–d, the lightgrey tube behind the darkgrey tube corresponds to the initial tube $[x](\cdot)$.

3.5 Set intervals

The set $\mathcal{P}(\mathbb{R}^n)$ of all subsets of $\mathbb{R}^n$ is a complete lattice with respect to the inclusion $\subset$. We can thus define the set $\mathbb{I}\mathcal{P}(\mathbb{R}^n)$ of intervals of $\mathcal{P}(\mathbb{R}^n)$ which is also a lattice with respect to $\subset$. An element $[\mathbb{X}] = \left[\mathbb{X}^-, \mathbb{X}^+\right]$ is called a *set interval* and can be used to represent an uncertain subset $\mathbb{X}$ of $\mathbb{R}^n$, i.e.,

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+, \tag{16}$$

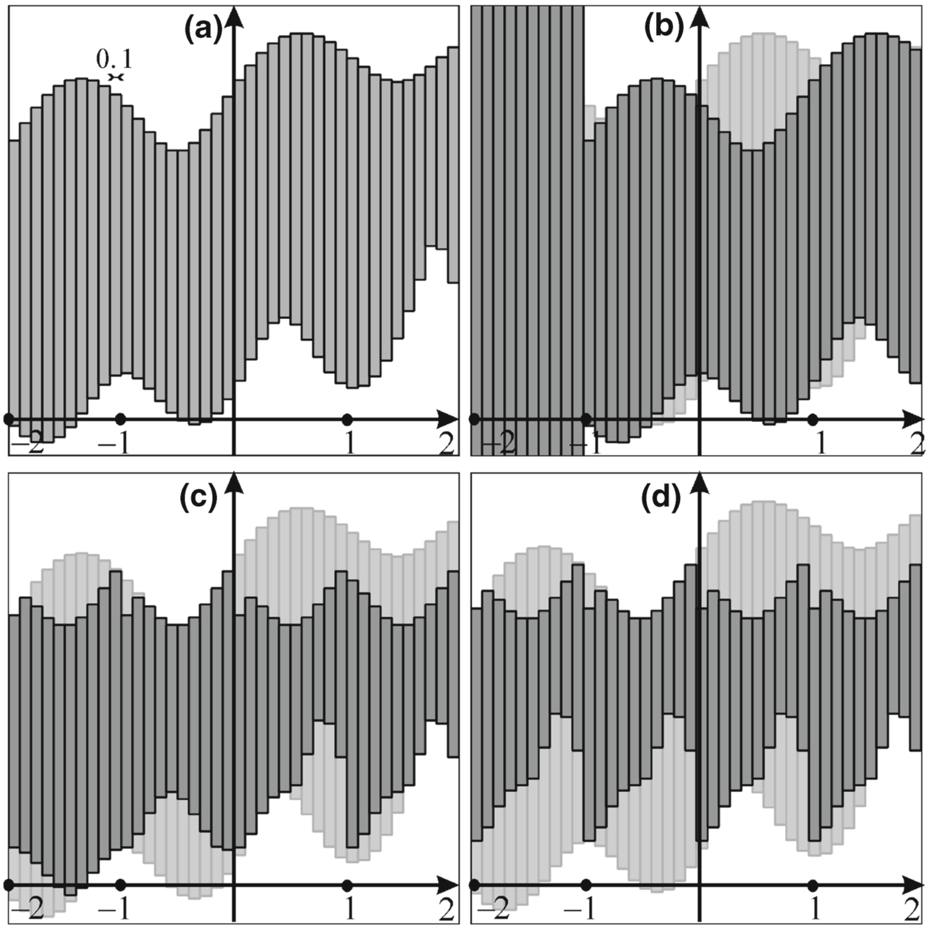

Fig. 5 Illustration of the contraction process over interval functions (or tubes). The darkgrey tube $[x](\cdot)$ encloses the unknown trajectory $x(\cdot)$

or equivalently $\mathbb{X} \in [\mathbb{X}]$. The top of $\mathbb{IP}(\mathbb{R}^n)$ is $\top = \left[\emptyset, \mathbb{R}^n\right]$ and encloses all sets of $\mathbb{R}^n$. The bottom $\perp$ of $\mathbb{IP}(\mathbb{R}^n)$ is the empty set of $\mathbb{IP}(\mathbb{R}^n)$. It should not be confused with the interval $[\emptyset, \emptyset]$ which a singleton: it contains as a single element the empty set of $\mathbb{R}^n$. Contractors can also be defined for set intervals. For instance, the minimal contractor $\mathcal{C}$ associated with the constraint $\mathbb{A} \subset \mathbb{B}$ is (see [16])

$$\mathcal{C}\begin{pmatrix} [\mathbb{A}^-, \mathbb{A}^+] \\ [\mathbb{B}^-, \mathbb{B}^+] \end{pmatrix} = \begin{pmatrix} [\mathbb{A}^-, \mathbb{A}^+ \cap \mathbb{B}^+] \\ [\mathbb{B}^- \cup \mathbb{A}^-, \mathbb{B}^+] \end{pmatrix}.$$

To illustrate the principle of constraint propagation in $\mathcal{P}(\mathbb{R}^n)$, consider an unknown subset $\mathbb{X}$ of $\mathbb{R}^2$ which satisfies the constraint $\mathbf{f}(\mathbb{X}) = \mathbb{X}$, where $\mathbf{f}$ is a rotation of $\mathbb{R}^2$ around $\mathbf{0}$ with an angle $\frac{\pi}{6}$. A possible contractor for $[\mathbb{X}]$ is

$$\mathcal{C}([\mathbb{X}]) = \left[\mathbb{X}^- \cup \mathbf{f}\left(\mathbb{X}^-\right), \mathbb{X}^+ \cap \mathbf{f}\left(\mathbb{X}^+\right)\right].$$

Assume now that the initial domain is $[\mathbb{X}] = \left[\mathbb{X}^-, \mathbb{X}^+\right]$, as represented on Fig. 6a. Figure 6b represents $\mathcal{C}([\mathbb{X}])$ and Fig. 6c depicts the fixed point $\mathcal{C}^\infty([\mathbb{X}]) = \mathcal{C} \circ \cdots \circ \mathcal{C}([\mathbb{X}])$. In the

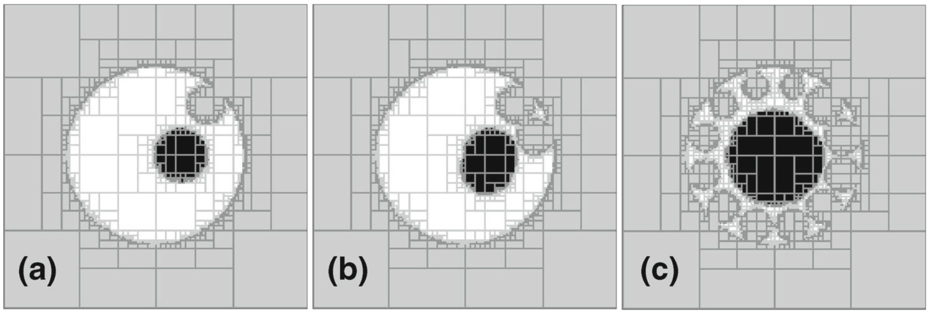

Fig. 6 **a** initial set interval $[\mathbb{X}]$; **b**: $\mathcal{C}([\mathbb{X}])$; **c**: the fixed point $\mathcal{C}^{\infty}([\mathbb{X}])$ .The color code for the representation of a set interval is as follows: the black boxes correspond to the lower bound and the black+white boxes correspond to the upper bound

computer, all set intervals are represented by a finite collection of boxes. This explains why the fixed point set interval do not have the rotational symmetry that we could have expected.

4 SLAM as a heterogeneous constraint network

This section formulates the SLAM problem described by (1) as a *constraint network*. We now describe the associated variables, constraints and domains.

Variables Among the unknown variables of our SLAM problem, we have (i) the trajectory of the robot $\mathbf{x}$ (ii) the sectors $\mathbb{H}_i$ containing a unique mark $\mathbf{a}(i)$: the one which corresponds to the ith detection. (iii) the location of the mark $\mathbf{a}(i)$ detected at time t_i, (iv) the association graph $\mathcal{G}$ and (v) the free space $\mathbb{F}$.

Domains The domains are sets enclosing the true value for the variables. Depending on the nature of the set $\mathbb{V}$ to which a given unknown variable v belongs, the domain can have different representations. For instance, when $\mathbb{V}$ is finite, the domain for v can be described by extension; when $\mathbb{V}$ is a lattice, the domain for v can be represented by intervals or by union of intervals (also called subpaving) [24]. In our context, the trajectory $\mathbf{x}$ belongs to a lattice and its domain will be represented by an interval of trajectories $\left[\mathbf{x}^-, \mathbf{x}^+\right]$. The vector $\mathbf{a}_i$ belongs to the lattice $\mathbb{R}^q$ and thus we could represent its domain as an interval of $\mathbb{R}^q$. Now, to get better accuracy, this domain will be represented by a subset $\mathbb{A}(i)$ of $\mathbb{R}^q$. The set variable $\mathbb{H}_i$ is a subset of $\mathbb{R}^q$. Since the $\mathcal{P}(\mathbb{R}^q)$ is a lattice, we thus represent the domain of $\mathbb{H}_i$ by an interval $\left[\mathbb{H}_i^-, \mathbb{H}_i^+\right]$ of $\mathcal{P}(\mathbb{R}^q)$. The free space $\mathbb{F}$ will also be represented by an interval $\left[\mathbb{F}^-, \mathbb{F}^+\right]$ of $\mathcal{P}(\mathbb{R}^q)$. Again, since the set of graphs with nodes $\mathbf{a}_i$ is a complete lattice, we represent the domain of the graph $\mathcal{G}$ by an interval of graphs $\left[\mathcal{G}^-, \mathcal{G}^+\right]$. As a consequence, we will write: $\mathbf{x}(\cdot) \in [\mathbf{x}](\cdot) = \left[\mathbf{x}^-(\cdot), \mathbf{x}^+(\cdot)\right]$, $\mathbf{a}(i) \in \mathbb{A}(i)$, $\mathbb{H}_i \in [\mathbb{H}_i] = \left[\mathbb{H}_i^-, \mathbb{H}_i^+\right]$, $\mathbb{F} \in [\mathbb{F}] = \left[\mathbb{F}^-, \mathbb{F}^+\right]$ and $\mathcal{G} \in \left[\mathcal{G}\right] = \left[\mathcal{G}^-, \mathcal{G}^+\right]$.

Initialization Let us now describe how the domains are initialized. Since we only know the initial state, we initialize the domain for the trajectory as follows: $[\mathbf{x}](t) \leftarrow [-\infty, \infty]$ if $t > 0$ and $[\mathbf{x}](0) \leftarrow \mathbf{0}$. The prior position for each mark is also unknown. As a consequence, for all i, we take $\mathbb{A}(i) \leftarrow \mathbb{R}^q$. The initial value for the sectors are also unknown and thus $[\mathbb{H}_i] \leftarrow \left[\emptyset, \mathbb{R}^q\right]$. This is also the case for the free space, i.e., $[\mathbb{F}] \leftarrow \left[\emptyset, \mathbb{R}^q\right]$. The domain for the graph $\mathcal{G}$ is initialized by $\left[\mathcal{G}\right] \leftarrow \left[\mathcal{G}^-, \mathcal{G}^+\right]$ where the lower bound $\mathcal{G}^-$ corresponds to a graph with no arc and an upper bound $\mathcal{G}^+$ corresponds to a complete graph (i.e., with edges between any two vertices).

Constraints The constraints relating all variables $\mathbf{x}$, $\mathbf{a}\,(i)$, $\mathbb{H}_i$, $\mathbb{F}$, $\mathcal{G}$, are the following

$$\begin{array}{ll}
\text{(i)} & \dot{\mathbf{x}} = \mathbf{f}\,(\mathbf{x}, \mathbf{u}) \\
\text{(ii)} & \mathbb{H}_i = \mathcal{H}_i\,(\mathbf{x}\,(t_i))\,, \quad \forall i \\
\text{(iii)} & \mathbf{a}\,(i) \in \mathbb{H}_i, \quad \forall i \\
\text{(iv)} & \mathbf{a}\,(i) = \mathbf{a}\,(j) \Leftrightarrow g_{ij} = 1, \quad \forall i, \forall j \\
\text{(v)} & \mathbf{a}\,(i) \in \mathbb{H}_j \Leftrightarrow g_{ij} = 1, \quad \forall i, \forall j \\
\text{(vi)} & g_{ij} = 1 \Rightarrow \mathbb{H}_j \backslash \mathbb{H}_i \subset \mathbb{F}, \quad \forall i, \forall j \\
\text{(vii)} & \mathbf{a}\,(i) \notin \mathbb{F}, \quad \forall i
\end{array} \tag{17}$$

for all i and j. Constraint (i) claims that the robot evolution must satisfy the state equations. From the initial state, it is possible to get a (poor) idea of the trajectory via dead reckoning, i.e., by considering the state equation (i) only for the prediction. Constraint (ii) defines the sectors $\mathbb{H}_i$ from the state of the robot. If the $\mathbf{x}\,(t_i)$ are approximately known this constraint allows us to get an approximation concerning the sectors $\mathbb{H}_i$. Constraint (iii) expresses that the mark $\mathbf{a}\,(i)$ belongs to the corresponding sector. Equivalence (iv) defines the graph $\mathcal{G}$. Equivalence (v) formulates the main set-membership condition used to prove the association (and the non-association) between marks (see Proposition 1). Relation (vi) provides a condition to dig the free space. Constraint (vii) states that a mark cannot belong to the free space.

Contractors To each constraint, we associate a contractor which is an operator able to contract the domains without removing any value which satisfies the constraint. *Constraint* (i): several specific contractors for state equations can be used in the literature (see, e.g., [30]). *Constraint* (ii): the contractor for (ii) is described in [15] and makes it possible to contract the set intervals $[\mathbb{H}_i]$ containing the sectors $\mathbb{H}_i$. *Constraint* (iii): the contractor is $\mathbb{A}\,(i) \leftarrow \mathbb{A}\,(i) \cap \mathbb{H}_i^+$, where $\mathbb{H}_i^+$ is the upper bound for the interval $[\mathbb{H}_i]$. *Constraint* (iv): if the domains for $\mathbf{a}\,(i)$ and $\mathbf{a}\,(j)$ do not intersect, no association between $\mathbf{a}\,(i)$ and $\mathbf{a}\,(j)$ exists, i.e., $g_{ij} = 0$. And if $g_{ij} = 1$, we can intersect these two domains. The resulting contractor is thus

$$\begin{array}{l}
\text{if } \mathbb{A}\,(i) \cap \mathbb{A}\,(j) = \emptyset \text{ then } \left[g_{ij}\right] \leftarrow \left[g_{ij}\right] \cap [0, 0] \\
\text{if } \left[g_{ij}\right] = [1, 1] \text{ then } \mathbb{A}\,(i) \leftarrow \mathbb{A}\,(i) \cap \mathbb{A}\,(j)\,.
\end{array} \tag{18}$$

Constraint (v): if the domain $\mathbb{A}\,(i)$ for $\mathbf{a}\,(i)$ is a subset of $\mathbb{H}_j^-$, the association can be done. The corresponding contractor is thus

$$\text{if } \mathbb{A}\,(i) \subset \mathbb{H}_j^- \text{ then } \left[g_{ij}\right] \leftarrow \left[g_{ij}\right] \cap [1, 1]\,. \tag{19}$$

Constraint (vi): if the association between $\mathbf{a}\,(i)$ and $\mathbf{a}\,(j)$ has been proved, then we can increase the free space, i.e.

$$\text{if } g_{ij} = [1, 1] \text{ then } [\mathbb{F}] \leftarrow \left[\mathbb{F}^- \cup \mathbb{H}_j^- \backslash \mathbb{H}_i^+, \mathbb{F}^+\right]. \tag{20}$$

Constraint (vii): The domain $\mathbf{a}\,(i)$ can be contracted by the free space using the following contractor: $\mathbb{A}\,(i) \leftarrow \mathbb{A}\,(i) \backslash \mathbb{F}^-$.

To make the contractions more efficient, redundant constraints can be added [31] to the set of constraints (17). For speeding (or improving) the solving process, we added the two constraints

$$\begin{array}{ll} \text{(viii)} & \mathbf{a}(i) \in \mathcal{H}_i(\mathbf{x}(t_i)), \quad \forall i \\ \text{(ix)} & \mathcal{G} \text{ is an equivalence relation} \end{array} \tag{21}$$

Constraint (viii) is derived from constraints (ii) and (iii) and corresponds to an inequality involving $\mathbf{a}(i)$ and $\mathbf{x}(t_i)$ (see Example 1). The contractor can be built using the method proposed in [22]. *Constraint (ix)*: It is a consequence of constraint (iv). It forces the graph to be reflexive, symmetric and transitive. Figure 7 represents the contractor graph. The dotted arrows correspond to the redundant constraints. To solve the SLAM problem, we call all 9 contractors, represented by the (hyper)-arcs in the figure, once and we repeat the procedure several times until no more significant contraction can be observed. From Tarski's theorem, the resulting propagation method converges and the method is guaranteed (no solution can be lost).

Example 7 Consider the situation, described by Fig. 3, where the sectors $\mathbb{H}_i$, $i \in \{1, \ldots, 6\}$ are exactly known (which is not the case in our SLAM context where we only know a set interval $[\mathbb{H}_i]$ such that $\mathbb{H}_i \in [\mathbb{H}_i]$). If nothing is known for the association graph $\mathcal{G}$, i.e.,

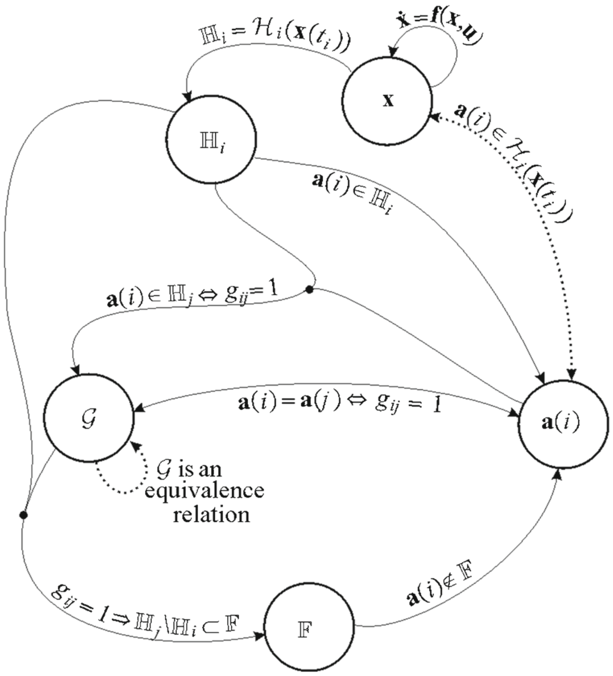
$\mathbb{H}_i = \mathcal{H}_i(\mathbf{x}(t_i))$
$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$
$\mathbf{x}$
$\mathbb{H}_i$
$\mathbf{a}(i) \in \mathcal{H}_i(\mathbf{x}(t_i))$
$\mathbf{a}(i) \in \mathbb{H}_i$
$\mathbf{a}(i) \in \mathbb{H}_j \Leftrightarrow g_{ij} = 1$
$\mathcal{G}$
$\mathbf{a}(i) = \mathbf{a}(j) \Leftrightarrow g_{ij} = 1$
$\mathbf{a}(i)$
$\mathcal{G}$ is an equivalence relation
$g_{ij} = 1 \Rightarrow \mathbb{H}_j \backslash \mathbb{H}_i \subset \mathbb{F}$
$\mathbf{a}(i) \notin \mathbb{F}$
$\mathbb{F}$

Fig. 7 Contractor digraph. Each node represents an unknown variable. The 9 (hyper)-arcs represent the constraints or the contractors. The arrow indicates which variable will be contracted by the contractor

$\mathcal{G} = [\perp, \top]$, and for the $\mathbf{a}(i)$, *i.e.*, $\mathbb{A}(i) = \mathbb{R}^2$ then the contractors (iii), (iv) and (v) provide the following contraction for $[\mathcal{G}]$:

$$[\mathcal{G}] = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & [0,1] \\ 1 & 1 & 1 & 0 & 1 & [0,1] \\ 1 & 1 & 1 & 0 & 1 & [0,1] \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & [0,1] \\ [0,1] & [0,1] & [0,1] & 0 & [0,1] & 1 \end{pmatrix}$$

Moreover, the domains $\mathbb{A}(1), \mathbb{A}(2), \mathbb{A}(3), \mathbb{A}(5)$ are all contracted to $\mathbb{H}(1) \cap \mathbb{H}(2) \cap \mathbb{H}(3) \cap \mathbb{H}(5)$, i.e., the black zone on the left of Fig. 3. The domain $\mathbb{A}(4)$ for $\mathbf{a}(4)$ is contracted to the black zone on the right of the figure and $\mathbb{A}(6)$ is contracted to $\mathbb{H}(6) \setminus \mathbb{Z}$ where $\mathbb{Z}$ corresponds to the hatched area.

5 Test-cases

To illustrate the principle of the method, we consider two test-cases. The first test case is based on a simulation and will be used to check that the method is guaranteed and also to illustrate how the computing time depends on different parameters such as the number of marks or the accuracy of the sensors. The second example is based on an actual experiment [14] that has been performed by an underwater robot.

5.1 Test-case 1

Generation of the data We have generated a set of data from a simulated robot following a cycloid for 100sec. We have also added randomly 10 marks inside the square $[-8, 8] \times [-8, 8]$, in meters . The robot is equipped with a specific sonar which collects a measure $\tilde{d}$ of the distance d to the nearest mark with an accuracy of 1cm. If the echo is clear, the sensor is able to conclude the unicity of the mark inside the ring at a distance $d \in [d] = \tilde{d} \pm 0.01$. Now, due to the silence before and after the first echo, our sensor is also able to detect a disk with radius δ which contains a single mark (see Fig. 2). The radius δ is smaller than the distance of the robot to the second nearest mark. If two marks are inside the echo ring, the sonar does not detect a clear echo and the data is rejected.

Resolution We assume that the robot is described by the state equations

$$\begin{cases} \dot{x}_1 = u_1 \cos u_2 \\ \dot{x}_2 = u_1 \sin u_2 \end{cases} \tag{22}$$

where the speed u_1 and the heading u_2 are measured every 0.01 sec. with an accuracy of ± 0.0001. The initial state is taken as $\mathbf{x}(0) = (0, 0)^{\mathrm{T}}$. Every second, from the sonar measurement, we define the set-valued sector functions

$$\begin{aligned} \mathcal{H}_i(\mathbf{x}(t_i)) &= \{\mathbf{a} \mid \|\mathbf{a} - \mathbf{x}(t_i)\| \in [d_i]\} \\ \mathcal{H}_{i+1}(\mathbf{x}(t_{i+1})) &= \{\mathbf{a} \mid \|\mathbf{a} - \mathbf{x}(t_{i+1})\| < \delta_{i+1}\} \end{aligned}$$

where i is odd and $t_i = t_{i+1}$. As illustrated by Example 3, this decomposition into the two sector functions allows us to represent both the existence and the absence of mark without changing our formalism. A new solver, named MSLAM, implementing the propagation is made available [17], with the C++ code and one tutorial video. MSLAM has been able to generate Fig. 8 in order to illustrate the off-line propagation. This figure has 5 lines and 2

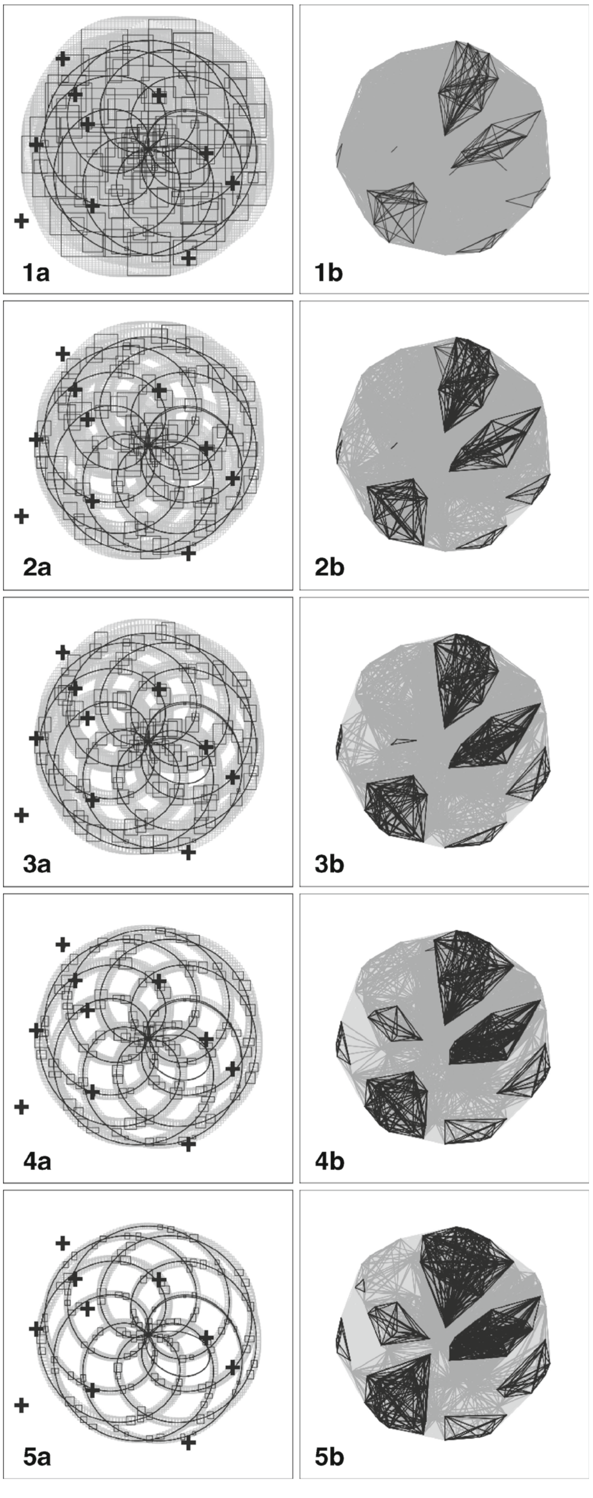

Fig. 8 Illustration of the propagation. Left: the tube $[\mathbf{x}](\cdot)$ becomes more and more accurate. Right: The association graph $\mathcal{G}$ has more and more black edges (i.e., associations)

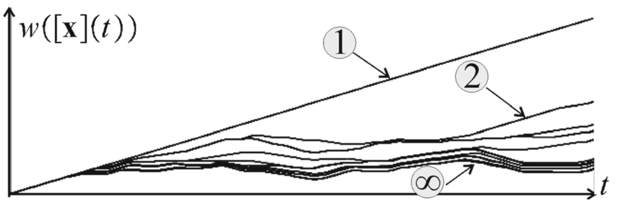

Fig. 9 Width $w(t)$ of the tube $[\mathbf{x}](t)$. When the number of iterations increases, $w(t)$ decreases

columns. The four first lines, which correspond to the four first iterations of the propagation, have been obtained in less than 2 minutes on a classical laptop (Intel 2.8GHz, 4.0 Go RAM). Line 5 corresponds to the fixed point obtained in less than 5 minutes (the fixed point is considered as reached when there are no more significant contractions). Column a on the left depicts the tubes (grey) for $\mathbf{x}(\cdot)$ and the true trajectory (thin black cycloid). The black boxes correspond to the value of the tube at time $t \in \{t_1, \ldots, t_m\}$ and the thick black **+** are the 10 marks. Column b on the right corresponds to the graph intervals. The vertices have been placed at the true pose $\mathbf{x}(t_i)$ of the robot. The black, darkgrey, lightgrey segments correspond to $g_{ij} = 1, [0, 1], 0$, respectively. At the initialization, all edges are darkgrey (i.e., we can have either $\mathbf{a}(i) = \mathbf{a}(j)$ or $\mathbf{a}(i) \neq \mathbf{a}(j)$). During the propagation, the edges become lightgrey (i.e., $\mathbf{a}(i) \neq \mathbf{a}(j)$) or black (i.e., $\mathbf{a}(i) = \mathbf{a}(j)$) whereas the trajectory envelope becomes more and more accurate.

Figure 9 shows that the width $w(t) = w([\mathbf{x}](t))$ of the tubes decreases at each step of the propagation. After one iteration, $w(t)$ is a line which is a consequence of the dead reckoning: since the initial position is known $w(0) = 0$ and when t increases, the robot becomes more and more lost. At iteration 2, the SLAM makes it possible to contract the trajectory by taking into account other constraints. After several iterations, the uncertainty remains limited. This means that the robot could evolve in its environment (which was not initially known) without getting lost.

Figure 10 provides the inner approximation $\mathbb{F}^-$ (white) of the free space $\mathbb{F}$ obtained when the fixed point is reached. At the fixed point, 3888 associations have been found, 29128 pairs $(\mathbf{a}(i), \mathbf{a}(j))$ have been proven disjoint and 5400 pairs $(\mathbf{a}(i), \mathbf{a}(j))$ have not been classified.

Remark 3 The solver MSLAM [17] also provides a simulator which makes it possible to change easily different parameters such as the size of the world, the density of marks, the trajectory of the robot, the sampling time, the frequency of detections, etc. When we change these parameters, we always observe that after 20 contractions we do not have any significant contractions anymore. As a consequence, we could provide an empirical complexity for our SLAM method: linear with respect to the number of time samples and quadratic with respect to the number of detections, due to the number of arcs $(\mathbf{a}(i), \mathbf{a}(j))$ in the graph.

Remark 4 The main difficulty with the SLAM problem considered in this paper is that the search space is of huge dimension (trajectories + graphs + marks) and the attraction basin of the solution is small. Existing methods, such as Monte-Carlo based SLAM, assume that if the trajectory is known, the mark association is almost solved. This is the case only if the marks are observable, which is not the case here: we only measure the distance, not the directions.

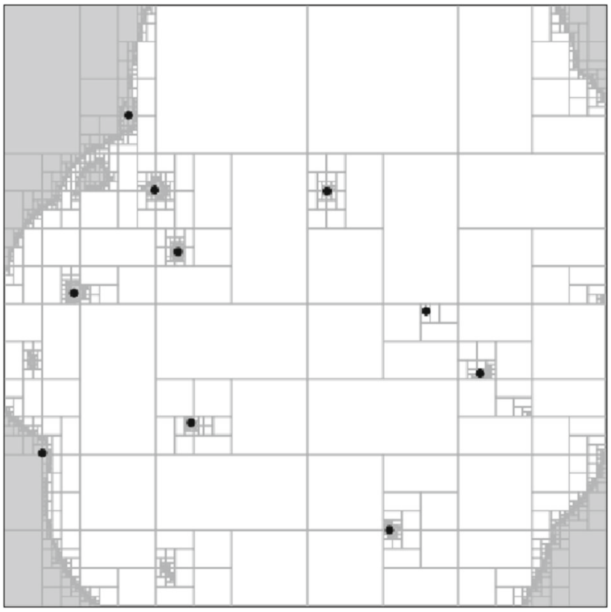

Fig. 10 Representation of the free space $\mathbb{F}$. The white zone (which corresponds to $\mathbb{F}^{-}$) is proven to be free of mark

5.2 Test-case 2

The second test-case is taken from [14]. A two-hours experiment has been made by an underwater robot (see Fig. 11) in the Douarnenez bay, Brittany (France) in order to test SLAM algorithms. On this experiment, we have six seamarks. Even if the location of the robot and of the seamarks is known, we assume here that we only measure proprioceptive

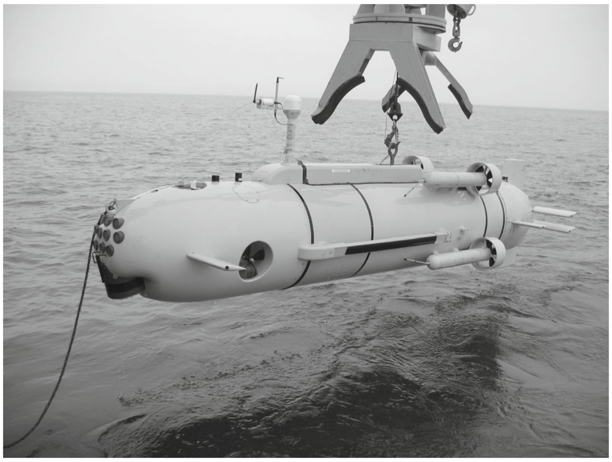

Fig. 11 The autonomous underwater vehicle, *Redermor*, built by the GESMA (Groupe d'Etude Sous-Marine de l'Atlantique)

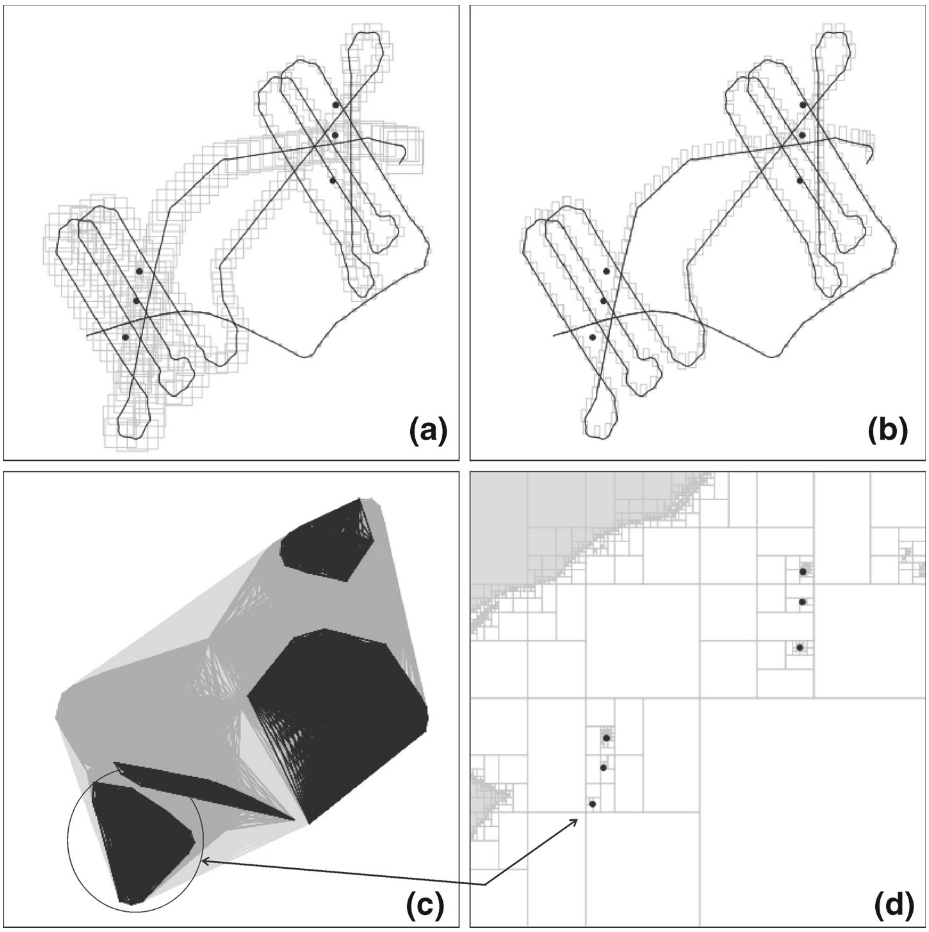

Fig. 12 Localization and mapping for an experiment invoving an actual robot; **a** initial envelope for the trajectory of the robot; **b** final envelope obtained after SLAM, **c** correspondance graph after SLAM; **d** inner approximation of the free space $\mathbb{F}$

data (from the gyrocompass and the loch Doppler), the distances (not the directions) to the nearest seamark and the distances to the second nearest seamark. We are thus in a similar situation as for Testcase 1 and we can therefore use the same software for the two test-cases. The distances are measured every 20 seconds. To reduce the computation time, we took a sampling time of 2 sec (instead of 0.1sec as taken in [14]).

After less than 100 minutes on the samel laptop, the propagation is able to get the approximation depicted on Fig. 12. To get the fixed point, 15 calls to each contractor have been needed. Figure 12a gives us the initial envelope obtained using only proprioceptive sensors. Figure 12b is the final envelope obtained after propagation by taking into account the distances to the seamarks. Figure 12c is the association graph obtained after propagation. A black edge links two positions of the robot that are proved to see the same seamark. A lightgrey edge corresponds to a validated non-association between two positions (they see different seamarks). Darkgrey edges correspond to unknown associations. For this testcase, we obtain 32,368 validated associations, 251,592 validated non-associations and 57,096 unknown associations. Figure 12d provides an inner approximation of the free space. No seamark can belong to the white area, i.e., all marks are inside the grey zone. It is proved that we have at least six seamarks, but of course, no upper bound for the number of marks may be computed. The set of black edges in the circle of Fig. 12b corresponds to the associations between positions where the robot detects the seamark located at the bottom of Fig. 12d.

All files (C++ codes for the simulator, the solver, data) associated to the two test-cases can be found in [17]. A tutorial video explaining the theory introduced in this paper is also given.

6 Conclusion

Range-only SLAM is a difficult problem, mainly due to partially observable marks and the fact that the problem is ill-conditioned. To solve the problem, this paper has first proposed a new formulation for SLAM which encompasses the information needed to guarantee possible associations. Then, a constraint programming approach has been followed. This has been made possible by introducing the new notion of heterogeneous constraint network. As a result, a backtrack-free fixed point method has been provided to compute sets that are proved to enclose the trajectory of the robot, the position of the marks and the free space. Moreover, the resulting method has been able to solve the mark association problem, which was not yet solved rigorously in the context of range-only SLAM. As most interval-based methods, the proposed approach could be combined with probabilist methods [1] and made robust with respect to outliers by relaxing a given number of constraints [11].

References

1. Abdallah, F., Gning, A., & Bonnifait, P. (2008). Box particle filtering for nonlinear state estimation using interval analysis. *Automatica*, *44*(3), 807–815.
2. Apt, K. (1998). The essence of constraint propagation. *Theoretical Computer Science*, *221*(2), 179–210.
3. Apt, K. (2003). *Principles of constraint programming*: Cambridge University Press.
4. Araya, I., Trombettoni, G., & Neveu, B. (2012). *A Contractor Based on Convex Interval Taylor, vol. 7, 1–16, LNCS 7298*: Springer.
5. Beldiceanu, N., Carlsson, M., Demassey, S., & Petit, T. (2006). Graph properties based filtering, Constraint Programming. *Principles and Practice of Constraint Programming*, *13*, 59–74.
6. Chabert, G., & Jaulin, L. (2009). Contractor Programming. *Artificial Intelligence*, *3*(173), 1079–1100.
7. Chabert, G., Jaulin, L., & Lorca, X. (2009). A constraint on the number of distinct vectors with application to localization, CP 2009, Vol. 7.
8. Colle, E., & Galerne, M.obile.r.obot.l.ocalization.b.y.m.ultiangulation.u.sing.s.et.i.nversion. (2013). *Robotics and Autonomous Systems*, *61*, 39–48.
9. Davey, B.A., & Priestley, H.A. (2002). *Introduction to Lattices and Order*: Cambridge University Press. (ISBN 0521784514).
10. Dooms, G., Deville, Y., & Dupont, P. (2005). Introducing a graph computation domain in constraint programming. *Principles and Practice of Constraint Programming*, *7*, 211–225.
11. Drevelle, V., & Bonnifait, P. (2009). High integrity gnss location zone characterization using interval analysis, ION GNSS, Vol. 7.
12. Drocourt, C., Delahoche, L., Brassart, E., & Clerentin, A. (2005). Incremental construction of the robot environmental map using interval analysis, Global Optimization and Constraint Satisfaction: Second International Workshop, COCOS 2003, 127–141.
13. Gning, A., & Bonnifait, P. (2006). Constraints propagation techniques on intervals for a guaranteed localization using redundant data. *Automatica*, *42*, 1167–1175.

14. Jaulin, L. (2009). A Nonlinear Set-membership Approach for the Localization and Map Building of an Underwater Robot using Interval Constraint Propagation. *IEEE Transaction on Robotics*, 88–98.
15. Jaulin, L., & maps, R.ange.-o.nly.S.LAM.w.ith.o.upancy (2011). A set-membership approach. *IEEE Transaction on Robotics*, 1004–1010.
16. Jaulin, L. (2012). Solving set-valued constraint satisfaction problems. *Computing*, 297–311.
17. Jaulin, L. (2015). MSLAM, a solver for range-only SLAM with indistinguisable landmarks, available at www.ensta-bretagne.fr/jaulin/mslam.html, LabSTICC, ENSTA Bretagne.
18. Jaulin, L., Kieffer, M., Didrit, O., & Walter, E. (2001). *Applied interval analysis, with examples in parameter and state estimation, robust rontrol and robotics*. London: Springer-Verlag.
19. Kurzhanski, A., & Valyi, I. (1997). *Ellipsoidal Calculus for Estimation and Control*. MA: Birkhauser.
20. Le Bars, F., Sliwka, J., Reynet, O., & Jaulin, L. (2012). State estimation with fleeting data. *Automatica*, 381–387.
21. Leonard, J.J., & Durrant-Whyte, H.F. (1992). *Dynamic map building for an autonomous mobile robot*.
22. Messine, F. (2004). Deterministic global optimization using interval constraint propagation techniques. *Operations Research*, 277–293.
23. Newman, P., & Leonard, J. (2003). Pure range-only sub-sea slam, ICRA03, pp. 1921–1926. Washington.
24. Sainudiin, R. (2010). *Machine interval experiments: Accounting for the physical limits on empirical and numerical resolutions*. Germany: LAP Academic, Koln.
25. Soares, G., Arnold-Bos, A., Jaulin, L., Vasconcelos, J.A., & Maia, C.A. (2008). An interval-based target tracking approach for range-only multistatic radar. *IEEE Transactions on Magnetics*, 1350–1353.
26. Spletzer, J. (2004). *A new approach to range-only slam for wireless sensor networks, LU-CSE-04-008*. Lehigh University.
27. Tarski, A. (1955). A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 285–309.
28. Thrun, S., Bugard, W., & Fox, D. (2005). *Probabilistic Robotics*. Cambridge: MIT Press.
29. Trombettoni, G., & Chabert, G. (2007). Constructive Interval Disjunction. In *Proc. CP, Constraint Programming, LNCS 4741* (pp. 635–650).
30. Tucker, W. (2002). A Rigorous ODE Solver and Smale's 14th Problem. *Foundations of Computational Mathematics*, 53–117.
31. van Emden, M. (1999). Algorithmic power from declarative use of redundant constraints, p. 363–381.
32. van Hentenryck, P., Deville, Y., & Michel, L. (1997). *Numerica: A modeling language for global optimization*. MA: MIT Press.
33. Waltz, D. (1975). Generating semantic descriptions from drawings of scenes with shadows. In P. H. Winston (Ed.), *The Psychology of Computer Vision* (pp. 19–91). New York: McGraw-Hill.
34. Wilczak, D., & Zgliczynski, P. (2011). Cr-lohner algorithm. *Schedae Informaticae*, 9–46.