[12] D. Hsu, "The bridge test for sampling narrow passages with probabilistic roadmap planners," in *Proc. Int. Conf. Robot. Autom.*, 2003, pp. 4420–4426.
[13] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
[14] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *IEEE Pervasive Comput.*, vol. 3, no. 4, pp. 24–33, Oct.–Dec. 2004.
[15] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: A quantitative comparison," *Comput. Netw.*, vol. 43, no. 4, pp. 499–518, 2003.
[16] Q. Li and D. Rus, "Navigation protocols in sensor networks," *ACM Trans. Sensor Netw.*, vol. 1, no. 1, pp. 3–35, Aug. 2005.
[17] K. J. O'Hara, V. L. Bigio, E. R. Dodson, and A. J. Irani, "Physical path planning using the gnats," in *Proc. Int. Conf. Robot. Autom.*, 2005, pp. 709–714.
[18] E. Plaku and L. E. Kavraki, "Distributed sampling-based roadmap of trees for large-scale motion planning," in *Proc. Int. Conf. Robot. Autom.*, 2005, pp. 3868–3873.
[19] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Adv. Multimedia*, vol. 2009, pp. 1–22, 2009.
[20] B. Taati, M. Greenspan, and K. Gupta, "A dynamic load-balancing parallel search for enumerative robot path planning," *J. Intell. Robot. Syst.*, vol. 47, no. 1, pp. 55–85, Sep. 2006.
[21] A. Verma, H. Sawant, and J. Tan, "Selection and navigation of mobile sensor nodes using a sensor network," in *Proc. PerCom*, 2005, pp. 41–50.
[22] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Netw.*, vol. 10, no. 2, pp. 169–181, Mar. 2004.
[23] Z. Yao, "Distributed motion planning in robotic sensor networks" Ph.D. dissertation, School Eng. Sci., Simon Fraser Univ., Burnaby, BC, Canada, 2011.
[24] Z. Yao and K. Gupta, "Backbone-based roadmaps for robot navigation in sensor networks," in *Proc. Int. Conf. Robot. Autom.*, 2008, pp. 1023–1029.
[25] Z. Yao and K. Gupta, "Distributed roadmaps for robot navigation in sensor networks," in *Proc. Int. Conf. Robot. Autom.*, 2010, pp. 3078–3083.

Range-Only SLAM With Occupancy Maps: A Set-Membership Approach

Luc Jaulin

***Abstract*—This paper proposes a new set-membership approach to solve range-only simultaneous localization and mapping (SLAM) problems in the case where the map is described by an arbitrary occupancy set (i.e., we do not assume that the map is composed of segments, punctual marks, etc.). The principle is to transform the SLAM problem into a hybrid constraint satisfaction problem (CSP), where the variables can either be real numbers, vectors, trajectories, or subsets of $\mathbb{R}^n$. An extension of existing constraint propagation methods is then proposed to solve hybrid CSPs involving set-valued variables. A simulated test case is then proposed to show the feasibility of the approach.**

***Index Terms*—Interval analysis, interval propagation, localization, occupancy map, set-membership estimation, simultaneous localization and mapping (SLAM).**

Manuscript received January 4, 2011; accepted April 17, 2011. Date of publication May 31, 2011; date of current version October 6, 2011. This paper was recommended for publication by Associate Editor E. Marchand and Editor D. Fox upon evaluation of the reviewers' comments.

The author is with the École Nationale Supérieure de Techniques Avancées (ENSTA) Bretagne, Brest 29806, France (e-mail: luc.jaulin@ensta-bretagne.fr).

Digital Object Identifier 10.1109/TRO.2011.2147110

I. Introduction

The simultaneous localization and mapping (SLAM) problem [19] for an autonomous robot moving in an unknown environment is to build a map of this environment while simultaneously using this map to compute its location. The history and critical issues of SLAM are discussed in [10]. SLAM methods can be classified in two categories [27], which are referred to as *feature-based* SLAM and *location-based* SLAM. Feature-based SLAM assume that the map is composed of a set of features together with their Cartesian location. The map has, thus, a parametric structure where the features are points, segments, corners, or any other parametric shape [1]. The way to handle uncertain location vectors when using geometric features of the environment as map elements is defined in [25]. Many implementations use the segment [24] or the line as the main kind of feature, and some of them use corners or edges modeled as points. The feature-based SLAM problem can be cast into a state-estimation problem by including the feature parameters among the state variables [8], [22]. Probabilistic techniques (Kalman filtering, Bayesian estimation, and particle filters) [27] or set membership approaches (where sets can be representedby parallelotopes [6], [7] or by boxes [13]) have been proven to solve efficiently the feature-based SLAM problem. Now, feature-based maps are not well suited to model nonstructured environments, as it is the case for underwater robotics where landmarks have no particular geometric shape. Location-based maps offer a label to any location in the world. They contain information not only about obstacles in the environment but about the absence of obstacles as well. A classical location-based map representation is known as *occupancy map* [9] (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. When the occupancy value is binary, the map can be represented by a subset $\mathbb{M}$ (that will be called the *map* in this paper) that distinguishes free from occupied terrain. The robot's pose must always be in the free space, i.e., outside $\mathbb{M}$. The corresponding SLAM problem contains some unknown variables which are subsets of $\mathbb{R}^q$, where $q = 2$ or 3, depending of the dimension of the robot's environment (2-D or 3-D), and it cannot be cast into a state-estimation problem anymore. The problem has a totally different nature and is much harder to solve than when it is possible to detect parametric shapes. It becomes even more difficult when the map is only perceived through omnidirectional rangefinders, which only returns the distance to the closest obstacle with no angle information. The use of such a range-only sensor precludes easy matching between detected points of the map. 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 representative of a large class of SLAM problems that cannot be solved yet. The development of tools to solve properly and efficiently this problem will be useful to solve many other SLAM problems.

This paper proposes the first set-membership method to deal with the pose-based range-only SLAM problem in the case where the map is represented by an occupancy set. It first shows that the SLAM problem can be transformed into a hybrid constraint satisfaction problem (CSP) where the variables are subsets of $\mathbb{R}^q$. Then, this paper extends existing contractor methods in order to deal with problems involving sets. This can be done thanks to the notion of set intervals recently introduced in [14]. This paper is organized as follows. Section II proposes a formulation for the pose-based SLAM and presents the particular case where telemeters that measure distances between the robot and the map are considered. Section III introduces the principle of hybrid constraint propagation. Section IV shows how the range-only SLAM can be solved with contractor tools. In order to illustrate the principle

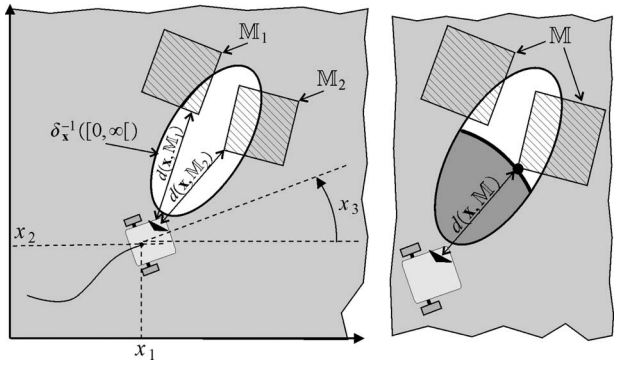

Fig. 1. Impact, covering, and dug zones.

of the approach, a testcase is treated in Section V. Finally, Section VI concludes this paper.

II. Formulation of the Range-Only Simultaneous Localization and Mapping

A range-only simultaneous localization and map building problem can be described by

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) & \text{(evolution equation)} \\ z(t) = d(\mathbf{x}(t), \mathbb{M}) & \text{(map equation)} \end{cases} \tag{1}$$

where $t \in [t] \subset \mathbb{R}$ 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), $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the evolution function, and d is the map function. The set $\mathbb{M} \in \mathcal{C}(\mathbb{R}^q)$ is the occupancy map, where $\mathcal{C}(\mathbb{R}^q)$ denotes the set of all compact sets of $\mathbb{R}^q$ and q is the dimension of the map (two or three in practice). The scalar z is an exteroceptive measurement collected by robot (for instance, by a sonar telemeter) and provides some information on $\mathbb{M}$. The map $\mathbb{M}$ is unknown and should be reconstructed together with the state $\mathbf{x}$. In this paper, a set-membership approach will be considered, i.e., we shall assume that $\mathbf{u}(t), z(t)$ are known to belong to some known intervals $[\mathbf{u}](t), [z](t)$. We shall also assume that the map function $d : \mathbb{R}^n \times \mathcal{C}(\mathbb{R}^q) \to \mathbb{R}^+$ corresponds to a rangefinder, i.e., for all $\mathbf{x}, \mathbb{M}_1, \mathbb{M}_2$, we have

$$\begin{cases} d(\mathbf{x}, \mathbb{M}_1 \cup \mathbb{M}_2) = \min\{d(\mathbf{x}, \mathbb{M}_1), d(\mathbf{x}, \mathbb{M}_2)\} \\ d(\mathbf{x}, \emptyset) = +\infty. \end{cases} \tag{2}$$

This assumption will provide us some conditions that will be used to characterize the map $\mathbb{M}$. Note that most rangefinders (laser, infrared, or ultrasound based) satisfy condition (2), whereas cameras (although often used for SLAM, see, e.g., [4], where Davison proposed a vision-based real-time SLAM) do not. Moreover, since the observations are only made by relative measurements between the robot and the environment, the initial state vector is assumed to be known. Define the function $\delta_{\mathbf{x}} : \mathbb{R}^q \to \mathbb{R}$ as

$$\delta_{\mathbf{x}}(\mathbf{a}) = d(\mathbf{x}, \{\mathbf{a}\}) \tag{3}$$

where $\{\mathbf{a}\}$ denotes the singleton containing $\mathbf{a}$. For a given state vector $\mathbf{x}$ and a given measurement z, we define the following sets:

$$\begin{aligned} &\text{covering zone } \delta_{\mathbf{x}}^{-1}([0,\infty[) = \{\mathbf{a}, \delta_{\mathbf{x}}(\mathbf{a}) < \infty\} \\ &\text{impact zone } \delta_{\mathbf{x}}^{-1}(\{z\}) = \{\mathbf{a}, \delta_{\mathbf{x}}(\mathbf{a}) = z\} \\ &\text{dug zone } \delta_{\mathbf{x}}^{-1}([0,z[) = \{\mathbf{a}, \delta_{\mathbf{x}}(\mathbf{a}) < z\} \end{aligned} \tag{4}$$

where $\{z\}$ denotes the singleton containing z. Fig. 1 represents a robot equipped with a rangefinder that can only detect points of the map inside an ellipse. The white ellipse on the left figure corresponds to $\delta_{\mathbf{x}}^{-1}([0,\infty[)$. The thick arc on the right figure corresponds to the set $\delta_{\mathbf{x}}^{-1}(\{z\})$, where $z = d(\mathbf{x}, \mathbb{M})$ and $\mathbb{M} = \mathbb{M}_1 \cup \mathbb{M}_2$. The part of the ellipse painted dark gray corresponds to $\delta_{\mathbf{x}}^{-1}([0,z[)$. The following two theorems will be used by our interval method to solve the range-only SLAM problem.

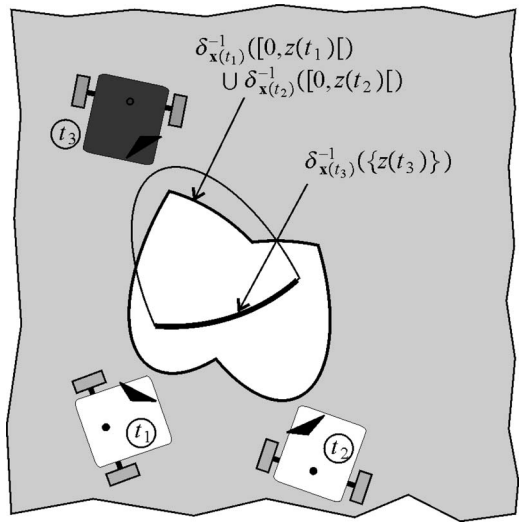

Fig. 2. Inconsistent situation where the impact zone at time t_3 is enclosed inside the zones dug at times t_1 and t_2.

Theorem 1 (dug zone) : Denote by $\mathbf{x}$ the state vector of the robot at a given time t, we have

$$z = d(\mathbf{x}, \mathbb{M}) \Rightarrow \delta_{\mathbf{x}}^{-1}([0,z[) \cap \mathbb{M} = \emptyset. \tag{5}$$

Proof: The proof is by contradiction. Assume that i) $z = d(\mathbf{x}, \mathbb{M})$; ii) $\mathbf{a} \in \delta_{\mathbf{x}}^{-1}([0,z[)$; and iii) $\mathbf{a} \in \mathbb{M}$. We have ii) $\Leftrightarrow \delta_{\mathbf{x}}(\mathbf{a}) \in [0,z[\overset{(3)}{\Leftrightarrow} d(\mathbf{x}, \{\mathbf{a}\}) < z \overset{\text{(i)}}{\Leftrightarrow} d(\mathbf{x}, \{\mathbf{a}\}) < d(\mathbf{x}, \mathbb{M})$. Now, since $\mathbf{a} \in \mathbb{M}$, we have $\{\mathbf{a}\} \subset \mathbb{M}$, and thus $d(\mathbf{x}, \{\mathbf{a}\}) \geq \min\{d(\mathbf{x}, \{\mathbf{a}\}), d(\mathbf{x}, \mathbb{M})\} \overset{(2)}{=} d(\mathbf{x}, \mathbb{M} \cup \{\mathbf{a}\}) \overset{\text{(iii)}}{=} d(\mathbf{x}, \mathbb{M})$. We have thus proved $d(\mathbf{x}, \{\mathbf{a}\}) < d(\mathbf{x}, \mathbb{M})$ and that $d(\mathbf{x}, \{\mathbf{a}\}) \geq d(\mathbf{x}, \mathbb{M})$. ■

The set $\mathbb{D} = \bigcup_{t \in [t]} \delta_{\mathbf{x}(t)}^{-1}([0, z(t)[)$ is called the *dug space*. A direct consequence of Theorem 1 is that $\mathbb{D}$ is inside the free space and thus does not intersect the map.

Theorem 2 (Impact Zone): For all $\mathbf{x}$, the impact zone intersects the map, i.e,

$$z = d(\mathbf{x}, \mathbb{M}) \Rightarrow \delta_{\mathbf{x}}^{-1}(\{z\}) \cap \mathbb{M} \neq \emptyset.$$

Proof: We have $z = d(\mathbf{x}, \mathbb{M}) \overset{(2)}{=} \min_{\mathbf{m} \in \mathbb{M}} d(\mathbf{x}, \{\mathbf{m}\})$ (recall that $\mathbb{M}$ is compact). Denote by $\mathbf{a}$ one minimizer of $d(\mathbf{x}, \{\mathbf{m}\})$ over $\mathbb{M}$. We have $z = d(\mathbf{x}, \{\mathbf{a}\})$, and thus, $\mathbf{a} \in \delta_{\mathbf{x}}^{-1}(\{z\})$. Therefore, $\mathbf{a}$ belongs to both $\delta_{\mathbf{x}}^{-1}(\{z\})$ and $\mathbb{M}$. ■

Fig. 2 illustrates the principle of the resolution method that will be proposed to solve our SLAM problem. The robot poses painted white represent the actual poses of the robot at times t_1, t_2 with the corresponding dug zones. From Theorem 1, we have

$$\left(\delta_{\mathbf{x}(t_1)}^{-1}([0, z(t_1)[) \cup \delta_{\mathbf{x}(t_2)}^{-1}([0, z(t_2)[)\right) \cap \mathbb{M} = \emptyset$$

which provides an outer approximation $\mathbb{M}^+$ of $\mathbb{M}$ represented by the light gray zone that covers a large part of the workspace. Since

$$\delta_{\mathbf{x}(t_3)}^{-1}(\{z(t_3)\}) \subset \delta_{\mathbf{x}(t_1)}^{-1}([0, z(t_1)[) \cup \delta_{\mathbf{x}(t_2)}^{-1}([0, z(t_2)[)$$

we have $\delta^{-1}_{\mathbf{x}(t_3)}(\{z(t_3)\}) \cap \mathbb{M} = \emptyset$, and thus, from Theorem 2, we conclude that the dark gray pose at time t_3 is inconsistent. Theorem 1 will be used for the map building (since it is able to find an outer approximation $\mathbb{M}^+$ of $\mathbb{M}$), whereas Theorem 2 will be used for the localization (since it is able to remove inconsistent poses). This is the main idea of the contraction approach proposed in the next sections.

III. Hybrid Constraint Propagation

Constraint propagation is a numerical method to solve nonlinear problems. In the literature, the unknown variables are Boolean numbers, integers, or real numbers. This section explains its principle and extends the classical technique in order to allow us to solve more general problems (such as those where the variables are functions or sets). This extension is necessary to solve our SLAM problem.

A. Lattices

A *lattice* $(\mathcal{E}, \leq)$ is a partially ordered set, which is closed under least upper and greatest lower bounds (see [3] for more details). The least upper bound (or infimum) of x and y is called the *join* and is denoted by $x \vee y$. The greatest lower bound (or *supremum*) is called the *meet* and is written as $x \wedge y$.

Example: The set $\mathbb{R}^n$ 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

$$\begin{aligned} \mathbf{x} \wedge \mathbf{y} &= (x_1 \wedge y_1, \ldots, x_n \wedge y_n) \\ \mathbf{x} \vee \mathbf{y} &= (x_1 \vee y_1, \ldots, x_n \vee y_n) \end{aligned} \tag{6}$$

where $x_i \wedge y_i = \min(x_i, y_i)$, and $x_i \vee y_i = \max(x_i, y_i)$.

A lattice $\mathcal{E}$ is *complete* if for all (finite of infinite) subsets $\mathcal{A}$ of $\mathcal{E}$, the least upper bound (which is denoted $\wedge\mathcal{A}$) and the greatest lower bound (which is denoted $\vee\mathcal{A}$) belong to $\mathcal{A}$. When a lattice $\mathcal{E}$ is not complete, it is possible to add new elements (corresponding the supremum or infimum of infinite subsets of $\mathcal{E}$ that do not belong to $\mathcal{E}$) 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 product of two lattices $(\mathcal{E}_1, \leq_1)$ and $(\mathcal{E}_2, \leq_2)$ is the lattice $(\mathcal{E}, \leq)$ that is defined as the set of all $(a_1, a_2) \in \mathcal{E}_1 \times \mathcal{E}_2$ with the order relation $(a_1, a_2) \leq (b_1, b_2) \Leftrightarrow ((a_1 \leq_1 b_1) \text{ and } (a_2 \leq_2 b_2))$.

B. Intervals

A *closed interval* (or *interval* for short) $[x]$ of a complete lattice $\mathcal{E}$ is a subset of $\mathcal{E}$ that 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 sublattice of $\mathcal{E}$. An interval $[x]$ of $\mathcal{E}$ will also be denoted by $[x] = [\wedge[x], \vee[x]]_{\mathcal{E}}$. For example, the sets $\emptyset = [\infty, -\infty]_{\mathbb{R}}$; $\mathbb{R} = [-\infty, \infty]_{\mathbb{R}}$; $[0, 1]_{\mathbb{R}}$; and $[0, \infty]_{\mathbb{R}}$ are intervals of $\mathbb{R}$; the set $\{2, 3, 4, 5\} = [2, 5]_{\mathbb{N}}$ is an interval of the set of integers $\mathbb{N}$; and the set $\{4, 6, 8, 10\} = [4, 10]_{2\mathbb{N}}$ is an interval of $2\mathbb{N}$. The *interval hull* (or *hull*, for short) of a subset $\mathcal{A}$ of $\mathcal{E}$ is the smallest interval of $\mathcal{E}$ which contains $\mathcal{A}$. We shall now introduce the notions of tubes and set intervals that will be used to solve the range-only SLAM problem.

Tubes: The set $\mathcal{F}$ of all functions from $\mathbb{R}$ to $\mathbb{R}^n$ is a complete lattice with the following partial order $\mathbf{f} \leq \mathbf{g} \Leftrightarrow \forall t \in \mathbb{R}, \ \mathbf{f}(t) \leq \mathbf{g}(t)$. An interval of $\mathcal{F}$ is called a *tube* [18], [21].

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 $\subset$.

Fig. 3 illustrates the notions of tubes and set intervals. In Fig. 3(left), the function f is bracketed by two stair functions f^-, f^+. In the computer, the interval $[f^-, f^+]$ is represented as a list of boxes. In Fig. 3(right), the uncertain set $\mathbb{X}$ is approximated by the set interval $[\mathbb{X}^-, \mathbb{X}^+]$, where $\mathbb{X}^-$ is the union of black boxes, and $\mathbb{X}^+$ is the union of black and white boxes. The two bounds $\mathbb{X}^-, \mathbb{X}^+$ of the interval $[\mathbb{X}^-, \mathbb{X}^+]$ are represented in the computer as union of boxes (or subpavings [26]).

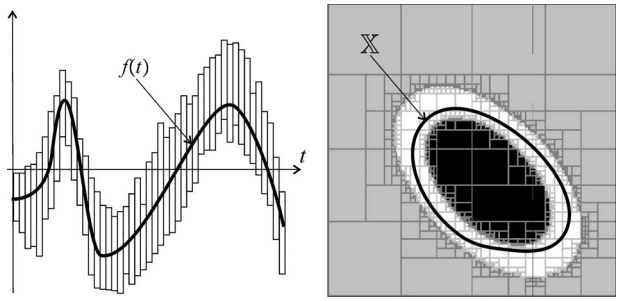

Fig. 3. Interval function (or tube) and a set interval.

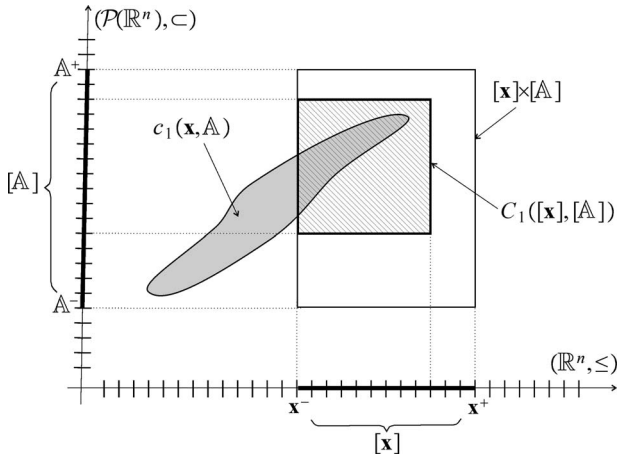

Fig. 4. Hybrid contractor $\mathcal{C}_1$ associated with the hybrid constraint $c_1(\mathbf{x}, \mathbb{A})$. The hybrid box $[\mathbf{x}] \times \mathbb{A}$ is contracted into the hatched box.

C. Contractors

Many problems of estimation, control, robotics, etc., can be represented by a continuous CSP [17]. A CSP [29] is composed of a set of variables $\mathcal{V} = \{x_1, \ldots, x_n\}$, a set of constraints $\mathcal{C} = \{c_1, \ldots, c_m\}$, and a set of interval domains $\{[x_1], \ldots, [x_n]\}$. Each variable x_i should belong to a complete lattice $(\mathcal{E}_i, \leq_i)$. When the lattices $\mathcal{E}_i$ have a different nature, the CSP is said to be *hybrid*. The Cartesian product of all interval domains of a hybrid CSP is called a *hybrid box*. In the context of this paper, a hybrid CSP will be considered and the x_i's will be real numbers, vectors, subsets of $\mathbb{R}^n$, or trajectories (i.e., functions of time). Propagation techniques contract as much as possible the interval domains for the variables without losing a solution [2], [28]. They have been shown to be efficient in several robotic applications such as localization [20], state estimation [11], or parametric SLAM [13]. Denote by $[\mathbf{x}]$ the Cartesian product of all domains $[x_i]$. A *contractor* associated with the constraint c_i is an operator C_i such that

$$\begin{aligned} (c_i \cap [\mathbf{x}]) \subset C_i([\mathbf{x}]) \quad &\text{(completeness)} \\ C_i([\mathbf{x}]) \subset [\mathbf{x}] \quad &\text{(contractance).} \end{aligned} \tag{7}$$

Fig. 4 shows a hybrid contractor associated with the hybrid constraint $c_1(\mathbf{x}, \mathbb{A})$. The x-axis corresponds to the lattice $\mathbb{R}^n$, whereas the y-axis corresponds $\mathcal{P}(\mathbb{R}^n)$ the set of subsets of $\mathbb{R}^n$. The discretization illustrates that only a finite number of elements of $\mathbb{R}^n$ and $\mathcal{P}(\mathbb{R}^n)$ can be represented by the computer. More precisely, these machine

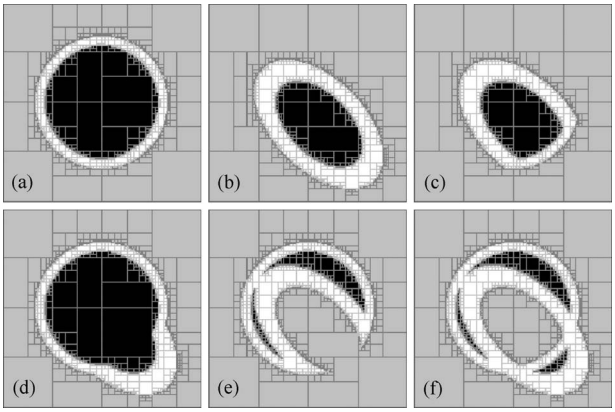

Fig. 5. If the two sets $\mathbb{A}, \mathbb{B}$ belong to the set intervals $[\mathbb{A}]$,$[\mathbb{B}]$ of (a) and (b), then the sets $\mathbb{A} \cap \mathbb{B}$, $\mathbb{A} \cup \mathbb{B}$, $\mathbb{A} \backslash \mathbb{B}$, $(\mathbb{A} \cup B) \setminus (\mathbb{A} \cap \mathbb{B})$ will belong to the set intervals of (c)–(f).

numbers are floating-point vectors for $\mathbb{R}^n$ and subpavings (i.e., union of boxes with floating point vectors as vertices). The hybrid box $[\mathbf{x}] \times [\mathbb{A}]$ has two components, namely, the box $[\mathbf{x}]$ and the set interval $[\mathbb{A}]$. The bounds of $[\mathbf{x}]$ are $\mathbf{x}^-$ and $\mathbf{x}^+$, whereas the bounds of $[\mathbb{A}]$ are the sets $\mathbb{A}^-$ and $\mathbb{A}^+$. Since $(c_1(\mathbf{x}, \mathbb{A}) \cap ([\mathbf{x}] \times [\mathbb{A}])) \subset C_1([\mathbf{x}], [\mathbb{A}])$ and $C_1([\mathbf{x}], [\mathbb{A}]) \subset [\mathbf{x}] \times [\mathbb{A}]$, the completeness and the contractance properties are satisfied.

To solve ahybrid CSP, the principle of the *propagation* is to contract all hybrid boxes $[\mathbf{x}]$ by calling all available contractors $C_1, \ldots, C_m$ until a fixed point is reached.

IV. Resolution

The implementation of the contractors that are described in Section III-C requires an implementation of an arithmetic for set intervals [14]. This arithmetic makes it possible to easily handle uncertain sets (such as the map for our SLAM problem), as illustrated in Fig. 5 (for the graphical representation of a set interval $[\mathbb{A}] = [\mathbb{A}^-, \mathbb{A}^+] = \{\mathbb{A}, \mathbb{A}^- \subset \mathbb{A} \subset \mathbb{A}^+\}$, the black boxes are inside $\mathbb{A}^-$, the gray boxes are outside $\mathbb{A}^+$, and the white boxes are inside $\mathbb{A}^+$ and outside $\mathbb{A}^-$). If the two sets $\mathbb{A}, \mathbb{B}$ belong to the set intervals $[\mathbb{A}]$,$[\mathbb{B}]$ [see Fig. 5(a) and (b)], then the sets $\mathbb{A} \cap \mathbb{B}$, $\mathbb{A} \cup \mathbb{B}$, $\mathbb{A} \backslash \mathbb{B}$, $(\mathbb{A} \cup \mathbb{B}) \setminus (\mathbb{A} \cap \mathbb{B})$ will belong to the set intervals shown in Fig. 5(c)–(f), respectively. The set interval arithmetic can be used to contract set intervals with respect to some constraints. Consider, for example, the constraint $\mathbb{A} \cap \mathbb{B} = \emptyset$ between the two sets $\mathbb{A}, \mathbb{B}$, and assume that $\mathbb{A} \in [\mathbb{A}], \mathbb{B} \in [\mathbb{B}]$. The contractions of $[\mathbb{A}], [\mathbb{B}]$ are given by the following operations: $[\mathbb{A}] := [\mathbb{A}^-, \mathbb{A}^+ \backslash \mathbb{B}^-]$ and $[\mathbb{B}] := [\mathbb{B}^-, \mathbb{B}^+ \backslash \mathbb{A}^-]$. Note that in the case of Fig. 5, it is not possible to find $\mathbb{A} \in [\mathbb{A}]$ and $\mathbb{B} \in [\mathbb{B}]$ such that $\mathbb{A}$ and $\mathbb{B}$ are disjoint. The contractions will provide set intervals with bad ordered bounds (i.e., $\mathbb{A}^- \not\subset \mathbb{A}^+ \backslash \mathbb{B}^-$ and $\mathbb{B}^- \not\subset \mathbb{B}^+ \backslash \mathbb{A}^-$). Such set intervals do not contains any set and should, thus, be interpreted as empty.

Our range-only SLAM problem can be cast into a hybrid CSP. The unknown variables are the trajectory $\mathbf{x}(t)$, the map $\mathbb{M}$, and the dug space $\mathbb{D}$. Since they have a different nature, the resulting CSP is hybrid. The constraints of the hybrid CSP are as follows:

$$
\left.
\begin{array}{ll}
1) & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\
2) & \mathbb{D} = \bigcup_{t \in [t]} \delta^{-1}_{\mathbf{x}(t)}([0, z(t)[)\,; \\
3) & \mathbb{D} \cap \mathbb{M} = \emptyset; \\
4) & \delta^{-1}_{\mathbf{x}(t)}(\{z(t)\}) \cap \mathbb{M} \neq \emptyset.
\end{array}
\right\} : z(t) = d(\mathbf{x}(t), \mathbb{M})
$$

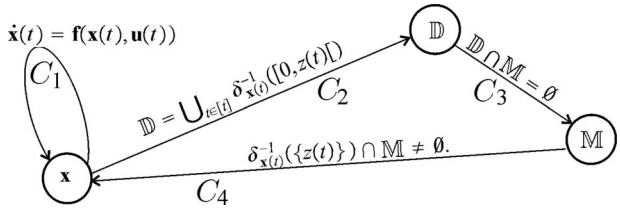

Fig. 6. Constraint diagram of the SLAM problem.

Constraints 2–4 correspond to a decomposition of the map constraint $z(t) = d(\mathbf{x}(t), \mathbb{M})$. This decomposition, which is a consequence of Theorems 1 and 2, is necessary to build the corresponding contractors. The prior domains for the set variables $\mathbb{M}$ and $\mathbb{D}$ are the set intervals $[\mathbb{M}] = [\mathbb{D}] = [\emptyset, \mathbb{R}^q]$ that enclose all subsets of $\mathbb{R}^q$. It translates the fact that no prior information on the map and the dug space is available. The prior domain for the trajectory $\mathbf{x}(t)$ is a tube $[\mathbf{x}](t)$. For the SLAM problem to be considered later, we have $[\mathbf{x}](t) = \mathbb{R}^n$ for $t > 0$ and $[\mathbf{x}](0)$ will be a singleton, which means that the initial state $\mathbf{x}(0)$ is known without any error.

Our CSP is composed of four constraints, the diagram of which is depicted in Fig. 6. To each constraint, we have to build a contractor. The first one $\mathcal{C}_1([\mathbf{x}](t))$ contracts the tube $[\mathbf{x}](t)$ with respect to the evolution equation $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$. Recall that a bounded tube $[\mathbf{u}](t)$ for $\mathbf{u}(t)$ is assumed to be known. The tube $[\mathbf{x}](t)$ can be contracted without losing a feasible value by an interval integration [5], [12], [23] using a forward and a backward propagation. The hybrid contractor $\mathcal{C}_2([\mathbb{D}], [\mathbf{x}](t))$ is related to the hybrid constraint $\mathbb{D} = \bigcup_{t \in [t]} \delta^{-1}_{\mathbf{x}(t)}([0, z(t)[)$. It makes possible to contract the set interval $[\mathbb{D}]$. The contractor $\mathcal{C}_3([\mathbb{D}], [\mathbb{M}])$ associated with the constraint $\mathbb{D} \cap \mathbb{M} = \emptyset$ yields contractions of the set interval $[\mathbb{M}]$ (see [14] for more explanations about this contractor). The hybrid contractor $\mathcal{C}_4([\mathbb{M}], [\mathbf{x}](t))$, which is associated with the constraint $\delta^{-1}_{\mathbf{x}(t)}(\{z(t)\}) \cap \mathbb{M} \neq \emptyset$ that should be satisfied for all t, provides contractions for the tube $[\mathbf{x}]$ (see [16] for more about this contractor).

The resulting propagation algorithm is given in the following table.

CONTRACT(inout: $[\mathbf{x}](t)$; out: $[\mathbb{D}], [\mathbb{M}]$)	
1	$[\mathbb{M}] := [\emptyset, \mathbb{R}^q]\,; [\mathbb{D}] = [\emptyset, \mathbb{R}^q]\,;$
2	Repeat
3	$\quad [\mathbf{x}](t) := \mathcal{C}_1([\mathbf{x}](t))\,;$
4	$\quad [\mathbb{D}] := \mathcal{C}_2([\mathbb{D}], [\mathbf{x}](t))\,;$
5	$\quad [\mathbb{M}] := \mathcal{C}_3([\mathbb{D}], [\mathbb{M}])\,;$
6	$\quad [\mathbf{x}](t) := \mathcal{C}_4([\mathbb{M}], [\mathbf{x}](t))\,;$
7	Until no more contraction.

For a given precision, the complexity of contractor-based propagation methods is polynomial if all contractors have a polynomial complexity. Now, some of the contractors that are used have a complexity which is exponential with respect to q, the dimension of the map (which is equal to 2 or 3). Now, since q can be considered as a fixed parameter, the complexity of the method is polynomial with respect to all other parameters of the problem (size of the world, time of the mission, etc.).

V. Testcase

In order to illustrate the behavior of the algorithm presented in Section IV, consider a mobile robot described by the following range-only SLAM equations:

$$
\left\{
\begin{array}{l}
\dot{x}_1(t) = u_1(t) \cos(u_2(t)) \\
\dot{x}_2(t) = u_1(t) \sin(u_2(t)) \\
z(t) = d(\mathbf{x}(t), \mathbb{M}).
\end{array}
\right. \tag{8}
$$

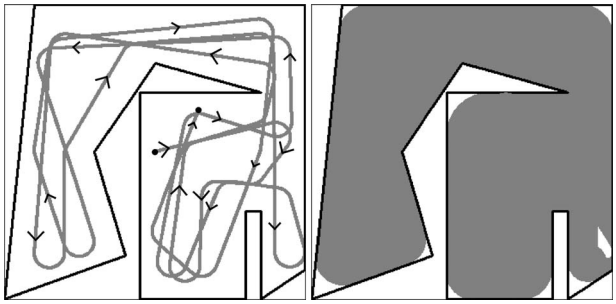

Fig. 7. (Left) Actual trajectory of the robot. (Right) Corresponding dug space. The frame box corresponds to $[-10, 10] \times [-10, 10]$.

The inputs of the system are the speed u_1 and the heading u_2 of the robot. The measurement z corresponds to the closest distance of the robot to the map which could have been obtained by using a omnidirectional (with angular aperture of 2π) sonar. The quantities u_1, u_2, and z are measured every 0.1 s with an error of 0.01 ms^{-1}, 0.01 rads^{-1}, and 0.01 m, respectively. The initial state, which is taken as $\mathbf{x} = (0, 0)^{\mathrm{T}}$, is assumed to be known. Fig. 7 provides a simulation of the robot moving inside an unknown map. As shown in Fig. 7, the 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. The gray zone in the right part of Fig. 7 represents the unknown dug space $\mathbb{D}$. It means that, in the ideal situation where the trajectory $\mathbf{x}(t)$ is known exactly, the map $\mathbb{M}$ can be approximated by $\mathbb{D}$ (in the sense that $\mathbb{M} \cap \mathbb{D} = \emptyset$).

An illustration of the interval propagation method is depicted in Fig. 8, which has been computed in about 15 min with a classical laptop. Fig. 8(a), (c), (e), (g), and (i) shows the computed tubes $[\mathbf{x}](t)$ (painted gray) with the true path (painted black) of the robot. As expected, the true path is always included inside the tube $[\mathbf{x}](t)$. Fig. 8(b), (d), (f), (h), and (j) corresponds to inner approximations (painted gray) of the dug space $\mathbb{D}$. The segments of the true map are also represented to illustrate how accurate is the approximation of the map. The width w of the tubes $[\mathbf{x}](t)$ is given in Fig. 9. After the first call to $\mathcal{C}_1$ (see Step 3), we get the tube shown in Fig. 8(a). The error increases linearly as shown in Fig. 9(a). After running all contractions, we get Fig. 8(b) as an inner approximation of the dug space $\mathbb{D}$. After a second run of the loop, we get Fig. 8(c) and (d). A third and fourth run yields Fig. 8(e)–(h). The fixed point that is reached is depicted in Fig. 8(i) and (j). As shown in Fig. 9, the width of the tube $[\mathbf{x}]$ decreases. Oscillations with respect to t are due to the fact that on the right part of the room (which was first observed), the robot succeeds to have a more accurate localization than on the left part of the room. When the right part of the room was observed for the first time, the robot did not accumulate uncertainties in its localization and was thus able to get an accurate map. When the robot came back to the right part, it was then able to take advantage of the accurate mapping to improve its localization. This is consistent with the loop-closure effect classically observed in a SLAM context. An educational windows program associated with this testcase with all C++ codes is made available [15].

Remark: Let us now give more details concerning the very beginning of the contraction procedure. At the first step, the contractor C_1 contracts the tube $[\mathbf{x}](t)$ by propagating forward the knowledge of the initial condition $\mathbf{x}(0)$. The corresponding tube $[\mathbf{x}](t)$ is represented in Fig. 8(a). The contractor C_2 contracts the set interval $[\mathbb{D}]$. More precisely, it provides an inner approximation $\mathbb{D}^-$ of the dug space (i.e., all points in $\mathbb{D}^-$ are obstacle free). The contractor C_3 contracts the set interval $[\mathbb{M}]$. More precisely, it performs the contraction $\mathbb{M}^+ = \mathbb{M}^+ \setminus \mathbb{D}^-$.

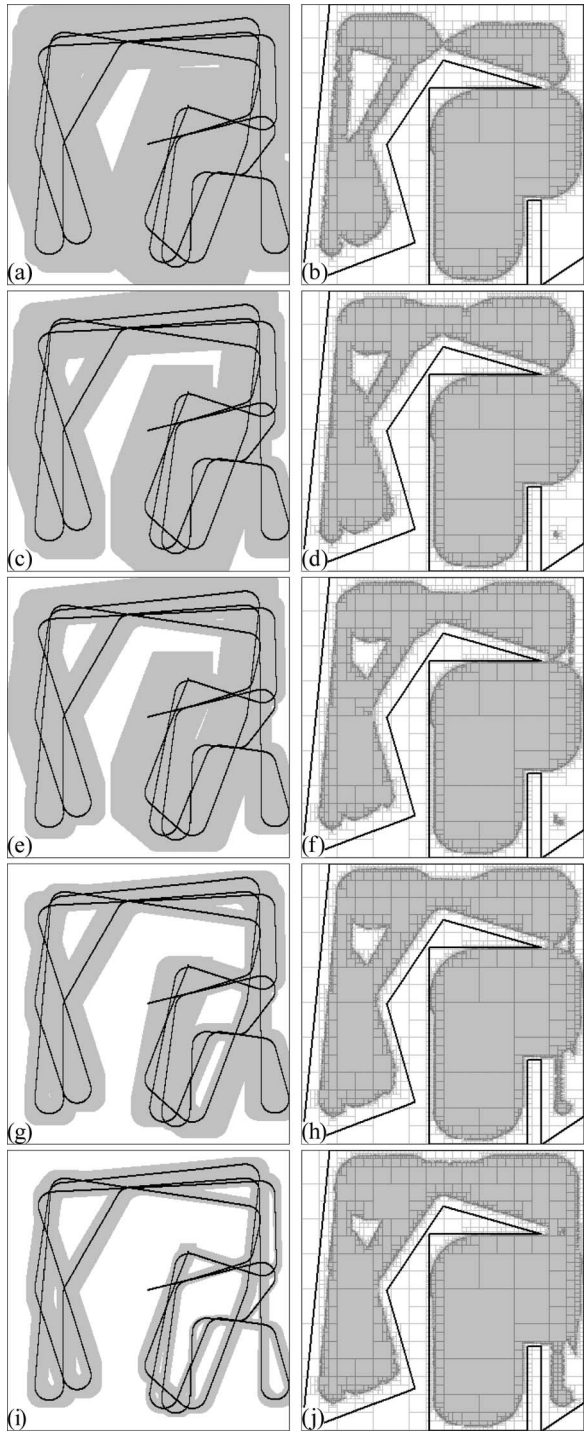

Fig. 8. (a), (c), (e), (g), and (i) Contraction of the tube $[\mathbf{x}](t)$ during the propagation. (b), (d), (f), (h), and (j) Evolution of the approximation the dug space $\mathbb{D}$.

The contractor C_4 contracts the tube $[\mathbf{x}](t)$ (see Fig. 6). At this level, the tube $[\mathbf{x}](t)$, which is represented in Fig. 10 (top left), appears to be the same as before the call to C_4 [see Fig. 8(a)]. This is due to the fact that only small parts of $[\mathbf{x}](t)$ have been contracted, and due to the superposition of all boxes, the contractions are not visible. The width $w([\mathbf{x}](t))$ of the tube is similar to the previous one (almost linear), except for some t (illustrated by the clouds inside the two ellipses at the bottom of Fig. 10). In Fig. 10 (top right), the subtube of $[\mathbf{x}](t)$

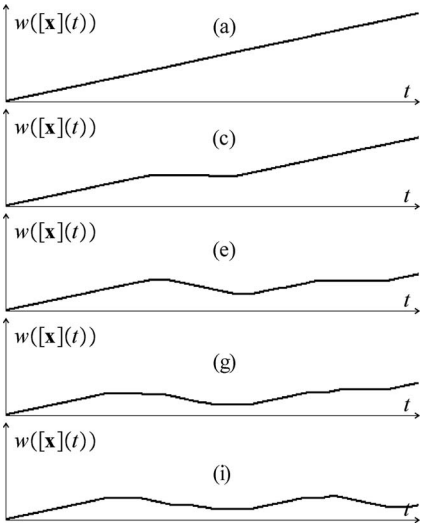

Fig. 9. Width of the tubes $[\mathbf{x}]\,(t)$ represented on the left part of Fig. 8.

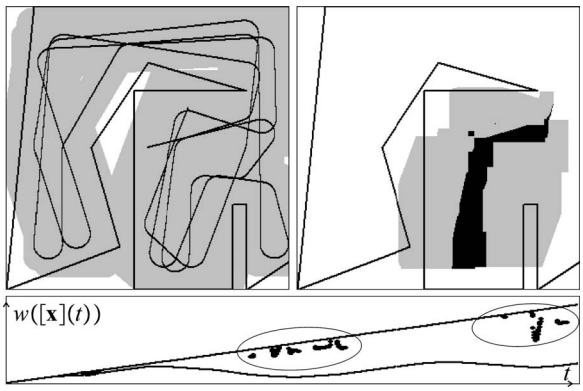

Fig. 10. First step of the propagation procedure. (Left) Tube $[\mathbf{x}]\,(t)$ obtained by propagating the initial vector. (Right) Part of the tube $[\mathbf{x}]\,(t)$ that has been contracted by removing values $\mathbf{x}\,(t)$ that are inconsistent with the distance $z(t)$ between the robot and the dug space. (Bottom) Widths of the tube $[\mathbf{x}]\,(t)$ with respect to the contractions.

contracted by C_4 is represented. The black zone corresponds to the part of $[\mathbf{x}]\,(t)$ that has been removed by C_4, and the gray zone corresponds the subtube after contraction. Each inconsistent point of the black zone is such that the disk of radius $z(t)$ is strictly inside the current inner approximation $\mathbb{D}^-$ of the dug space.

VI. Conclusion

This paper has presented a new contractor method to solve the SLAM problem in the case where the map cannot be represented by a parametric structure. In such a case, the map can be represented as a set of an infinite number of punctual marks or equivalently by arbitrary compact subsets of $\mathbb{R}^q$. As a consequence, the SLAM problem encloses unknown variables that are sets of $\mathbb{R}^q$ and containing an uncountable number of elements. To solve these type of set-valued nonlinear problems with set-membership methods is not easy and atypical in the robotic or control community. An extension of existing contractor methods has then been proposed in order to allow using contractors associated with set-valued variables. The principle of the resulting hybrid contractor approach for SLAM has been illustrated through a range-only offline SLAM testcase with simulated data. To the author's knowledge, no other existing deterministic method could solve a similar problem.

However, even if the principle of solving the pose-based range-only SLAM in a reliable way has been demonstrated, the proposed technique exhibits some limitations: 1) When outliers occur during the mission, the trajectory tube quickly becomes an empty tube and no more estimation of the map and the trajectory can be produced anymore; 2) the approach cannot be easily extended to situations where moving obstacles exist in the environment; 3) a prior box enclosing the map and the trajectory is needed, which is not well suited for exploration; 4) it is computationally expensive to match representation of the space with subpavings; and 5) the method is only able to perform offline SLAM, which is not suited for real-time applications. Further research is, therefore, necessary to make the approach effective to solve online SLAM problems that involve real robots.

References

[1] J. Castellanos, J. Neira, and J. Tardós, "Multisensor fusion for simultaneous localization and map building," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 908–914, Dec. 2001.

[2] G. Chabert and L. Jaulin, "Contractor programming," *Artif. Intell.*, vol. 173, pp. 1079–1100, 2009.

[3] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[4] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. Int. Conf. Comput. Vis.*, 2003, pp. 1403–1410.

[5] Y. Deville, M. Janssen, and P. V. Hentenryck, "Consistency techniques in ordinary differential equations," in *Proc. 4th Int. Conf. Principles Practice Constraint Program.*, Lecture Notes in Computer Science, 1998, pp. 289–315.

[6] M. Di Marco, A. Garulli, A. Giannitrapani, and A. Vicino, "A set theoretic approach to dynamic robot localization and mapping," *Auton. Robots*, vol. 16, no. 1, pp. 23–47, 2004.

[7] M. Di Marco, A. Garulli, S. Lacroix, and A. Vicino, "Set membership localization and mapping for autonomous navigation," *Int. J. Robust Nonlinear Control*, vol. 7, no. 11, pp. 709–734, 2001.

[8] G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 3, no. 17, pp. 229–241, Jun. 2001.

[9] A. Elfes, "Sonar-based real world mapping and navigation," *IEEE Trans. Robot. Autom.*, vol. 3, no. 3, pp. 249–265, Jun. 1987.

[10] U. Frese, "A discussion of simultaneous and mapping," *Auton. Robots*, vol. 20, pp. 25–42, 2006.

[11] A. Gning and P. Bonnifait, "Constraints propagation techniques on intervals for a guaranteed localization using redundant data," *Automatica*, vol. 42, no. 7, pp. 1167–1175, 2006.

[12] L. Jaulin, "Nonlinear bounded-error state estimation of continuous-time systems," *Automatica*, vol. 38, pp. 1079–1082, 2002.

[13] L. Jaulin, "A nonlinear set-membership approach for the localization and map building of an underwater robot using interval constraint propagation," *IEEE Trans. Robot.*, vol. 25, no. 1, pp. 88–98, Feb. 2009.

[14] L. Jaulin, "Solving set-valued constraint satisfaction problems," presented at the SCAN 2010, Lyon, France.

[15] L. Jaulin. (2011). "Programs for range only SLAM with occupancy maps," ENSTA-Bretagne [Online]. Available: www.ensta-bretagne.fr/jaulin/dig.html

[16] L. Jaulin and G. Chabert, "Resolution of nonlinear interval problems using symbolic interval arithmetic," *Eng. Appl. Artif. Intell.*, vol. 23, no. 6, pp. 1035–1049, 2010.

[17] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. London, U.K.: Springer-Verlag, 2001.

[18] A. Kurzhanski and I. Valyi, *Ellipsoidal Calculus for Estimation and Control*. Boston, MA: Birkhäuser, 1997.

[19] J. J. Leonard and H. F. Durrant-Whyte, "Dynamic map building for an autonomous mobile robot," *Int. J. Robot. Res.*, vol. 11, no. 4, 1992.

[20] D. Meizel, O. Lévêque, L. Jaulin, and E. Walter, "Initial localization by set inversion," *IEEE Trans. Robot. Autom.*, vol. 18, no. 6, pp. 966–971, Dec. 2002.

[21] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, *Bounding Approaches to System Identification*. New York: Plenum, 1996.
[22] D. K. Montemerlo, S. Thrun, and B. Wegbreit, "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, pp. 1151–1156.
[23] N. Nedialkov, K. Jackson, and G. Corliss, "Validated solutions of initial value problems for ordinary differential equations," *Appl. Math. Comput.*, vol. 105, no. 1, pp. 21–68 1999.
[24] P. Newman, J. Leonard, J. Tardós, and J. Neira, "Explore and return: Experimental validation of real-time concurrent mapping and localization," in *Proc. Int. Conf. Robot. Autom. 2002*, Washington, DC, pp. 1802–1809.
[25] R.Smith, M. Self and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*. vol. 8, New York: Springer, 1990, pp. 167–193.
[26] D. Sam-Haroud, "Constraint consistency techniques for continuous domains" Ph.D. dissertation, Swiss Federal Inst. Technol., Lausanne, Switzerland, 1995.
[27] S. Thrun, W. Bugard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.
[28] M. van Emden, "Algorithmic power from declarative use of redundant constraints," *Constraints*, vol. 4, no. 4, pp. 363–381, 1999.
[29] P. van Hentenryck, L. Michel, and Y. Deville, *Numerica—A Modelling Language for Global Optimization*. Cambridge, MA: MIT Press, 1997.

Online Trajectory Generation: Straight-Line Trajectories

Torsten Kröger, *Member, IEEE*

***Abstract*—A concept of online trajectory generation for robot motion control systems that enables instantaneous reactions to unforeseen sensor events was introduced in a former publication. This concept is now extended with the important feature of homothety. Homothetic trajectories are 1-D straight lines in a multidimensional space and are relevant for all straight-line motion operations in robotics. This paper clarifies 1) how online concepts can be used to generate homothetic trajectories and 2) how we can instantaneously react to (sensor) events with homothetic trajectories. To underline the practical relevance, real-world experimental results with a seven-degree-of-freedom (DOF) robot arm are shown.**

***Index Terms*—Homothety, online trajectory generation (OTG), robot motion control, sensor integration.**

I. Introduction and Problem Formulation

Homothetic trajectories belong to the most common ones in commercially available robotic manipulator control. They represent a motion along a 1-D straight line in a multidimensional space (Euclidian space, Euler space, spherical coordinates, joint space, etc.). If we consider a mechanical system with multiple degrees of freedom (DOFs) that is equipped with one or more sensors delivering digital and/or analog sensor signals, it is an essential feature to *instantaneously* react to *unforseen* sensor signals and events. In [1], a framework for the online generation of time-synchronized robot motion trajectories was introduced, which generates trajectories from *arbitrary* initial states of motion. This paper now extends the framework of [1] by enabling phase-synchronized (homothetic) trajectories that are generated within low-level control cycle (typically, 1 ms or less).

Let us define a trajectory $\mathcal{M}_i(t)$, which is calculated at a discrete-time instant T_i, as

$$\mathcal{M}_i(t) = \Big\{ \left({}^1\mathbf{m}_i(t),\, {}^1\mathcal{V}_i\right), \ldots, \left({}^l\mathbf{m}_i(t),\, {}^l\mathcal{V}_i\right) \ldots, \left({}^L\mathbf{m}_i(t),\, {}^L\mathcal{V}_i\right) \Big\} \tag{1}$$

where the elements ${}^l\mathbf{m}_i(t)$ are the matrices of the motion polynomials

$$\begin{aligned} {}^l\mathbf{m}_i(t) &= \left({}^l\vec{p}_i(t),\, {}^l\vec{v}_i(t),\, {}^l\vec{a}_i(t),\, {}^l\vec{j}_i(t)\right) \\ &= \left({}^l_1\vec{m}_i(t), \ldots, {}^l_k\vec{m}_i(t), \ldots, {}^l_k\vec{m}_i(t)\right)^T. \end{aligned} \tag{2}$$

Here, K is the total number of DOFs, and a trajectory segment l of a single DOF k is described by the motion polynomials

$${}^l_k\vec{m}_i(t) = \left({}^l_k p_i(t),\, {}^l_k v_i(t),\, {}^l_k a_i(t),\, {}^l_k j_i(t)\right) \tag{3}$$

where ${}^l_k p_i(t)$ represents the position progression, ${}^l_k v_i(t)$ represents the velocity progression, ${}^l_k a_i(t)$ represents the acceleration progression, and ${}^l_k j_i(t)$ represents the jerk progression. According to (1), a complete trajectory is described by L segments, and each segment l is accompanied by a set of time intervals

$${}^l\mathcal{V}_i = \left\{ {}^l_1\vartheta_i, \ldots, {}^l_k\vartheta_i, \ldots, {}^l_k\vartheta_i \right\} \quad \text{where} \quad {}^l_k\vartheta_i = \left[{}^{l-1}_k t_i,\, {}^l_k t_i \right] \tag{4}$$

such that a single set of motion polynomials ${}^l_k\vec{m}_i(t)$ is only valid within the interval ${}^l_k\vartheta_i$.

In [2], a good introduction about homothety is given, and in [3], it is applied to robot trajectory generation. To generate homothetic trajectories in the K-dimensional space, we can take an arbitrary DOF, i.e., $\kappa \in \{1, \ldots, K\}$, as the reference DOF and design the trajectory parameters, such that the condition

$$\begin{aligned} &\forall\, (k,l) \in \{1, \ldots, K\} \times \{1, \ldots, L\} \\ &\quad {}^l_k v_i(t) = {}_k\varrho_i \cdot {}^l_\kappa v_i(t) \quad \text{with} \quad t \in {}^l_k\vartheta_i \end{aligned} \tag{5}$$

is fulfilled. This naturally also implies that

$$\begin{aligned} &\forall\, (k,l) \in \{1, \ldots, K\} \times \{1, \ldots, L\} \\ &\left.\begin{array}{l} {}^l_k a_i(t) = {}_k\varrho_i \cdot {}^l_\kappa a_i(t) \\ {}^l_k j_i(t) = {}_k\varrho_i \cdot {}^l_\kappa j_i(t) \\ {}^l_k d_i(t) = {}_k\varrho_i \cdot {}^l_\kappa d_i(t) \end{array}\right\} \quad \text{with} \quad t \in {}^l_k\vartheta_i \end{aligned} \tag{6}$$

are fulfilled. The constant vector

$$\vec{\varrho}_i = \left({}_1\varrho_i, \ldots, {}_k\varrho_i, \ldots, {}_k\varrho_i\right)^T \quad \text{with } {}_\kappa\varrho_i = 1 \tag{7}$$

defines the ratios between the reference DOF κ and all other DOFs $\{1, \ldots, K\}\backslash\{\kappa\}$. Usually, homothetic trajectories are generated as described by (5)–(7): A scalar function specifies the velocity progression for one DOF, which is referred to as a reference DOF, and the motion of the other DOFs is calculated by the use of $\vec{\varrho}_i$ [3].

Fig. 1 illustrates the path of a simple 2-DOF point-to-point motion with zero velocities in $\vec{P}_0$ and $\vec{P}_0^{\,\mathrm{trgt}}$. One can clearly recognize the differences between the phase-synchronized (homothetic), time-synchronized (cf., [1]), and nonsynchronized trajectories. In particular, the elements of the kinematic motion constraints

$$\mathbf{B}_i = \left(\vec{V}_i^{\,\max}, \vec{A}_i^{\,\max}, \vec{J}_i^{\,\max}, \vec{D}_i^{\,\max}\right) \tag{8}$$

Manuscript received December 23, 2010; revised April 13, 2011; accepted May 18, 2011. Date of publication June 16, 2011; date of current version October 6, 2011. This paper was recommended for publication by Associate Editor A. Albu-Schäffer and Editor J.-P. Laumond upon evaluation of the reviewers' comments. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation).

The author is with the Artificial Intelligence Laboratory, Stanford University, Stanford, CA 94305-9010 USA (e-mail: tkr@stanford.edu).

Digital Object Identifier 10.1109/TRO.2011.2158021