

# A Nonlinear Set Membership Approach for the Localization and Map Building of Underwater Robots

Luc Jaulin

**Abstract**—This paper proposes a set membership method based on interval analysis to solve the simultaneous localization and map building (SLAM) problem. The principle of the approach is to cast the SLAM problem into a constraint satisfaction problem for which interval propagation algorithms are particularly powerful. The resulting propagation method is illustrated on the localization and map building of an actual underwater robot.

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

## I. INTRODUCTION

THIS PAPER proposes a set-membership approach to deal with *simultaneous localization and map building* (SLAM) in a submarine context. The 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. Most approaches for the SLAM cast the problem into a state estimation problem by including the landmark locations among the state variables [2], [7], [23]. Also, see [9], [29], and [33] for the case of autonomous underwater robots. Most of the proposed solutions are based on probabilistic estimation techniques (Kalman filtering, Bayesian estimation, particle filters) [30], [31], which aim at blending data with some state equations of the robot.

In this paper, a set-membership approach for SLAM (see, e.g., [5] and [6]) is considered, and it is shown that this approach leads us to a huge set of nonlinear equations that can be solved efficiently using interval analysis and constraint propagation (see [24] and [25] for classical interval analysis, [12] for interval optimization methods, and [15] for interval constraint propagation and applications). The approach will be illustrated on an experiment where an actual underwater vehicle is involved. In this problem, we will try to find an envelope for the trajectory of the robot and compute sets that contain some detected seamounts. Note that for our experiment, the detections of the seamounts are performed by a human operator, after the mission of the robot. Thus, our problem can be considered as an offline smoothing SLAM problem, i.e., the reconstruction of

the robot trajectory and the map building are performed once the measurements have been collected.

Set-membership methods have often been considered for the localization of robots (see, e.g., [11] and [22] if the problem is linear, and also [1] when the robot is underwater). In situations where strong nonlinearities are involved, interval analysis has been shown to be particularly useful (see, e.g., [21], where the first localization of an actual robot is solved with interval methods). Classical interval analysis has been shown to be efficient in several SLAM applications (see [8] and [26], where it is applied to SLAM of wheeled robots). Now, the resulting techniques perform bisections, and thus, the complexity of the resulting algorithms is exponential with respect to the number of variables that are bisected. In this paper, the approach is made more efficient by the use of interval propagation. With this approach, no bisection is performed and the resulting complexity becomes polynomial (almost linear in practice for the SLAM problem) with respect to the number of variables. Note that in our case, the number of variables grows with time, since they contain the robot pose at each time instant, as well as the mark position. Although never used in this context, there exist many other robotics applications where interval propagation has been successful (see, e.g., [27] for state estimation, [10] for dynamic localization of robots, [20] and [32] for control of robots, [3] for topology analysis of configuration spaces, etc.).

The paper is organized as follows. We shall first present a set membership approach for SLAM in Section II. By taking into account the state equations of the robot and the interpretation of the sensors in a bounded-error context, we will show that a SLAM problem can be cast into a set of nonlinear equations and inequalities. Section III will recall the basic notions of interval analysis and constraint propagation needed to solve the set of nonlinear constraints involved in our SLAM problem. In Section IV, we consider the particular situation of the SLAM in a submarine context that could be used by mine hunters. The efficiency of the interval approach will be illustrated on an actual experiment in Section V. A comparison between the proposed approach and other classical Bayesian techniques will be proposed in Section VI. Section VII will then conclude the paper.

## II. SET-MEMBERSHIP APPROACH FOR SLAM

### A. Principle

We shall describe our SLAM problem as follows:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) & \text{(evolution equation)} \\ \mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}) & \text{(observation equation)} \\ \mathbf{z}_i = \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{m}_i) & \text{(mark equation)} \end{cases} \quad (1)$$

Manuscript received January 11, 2008; revised July 18, 2008; accepted November 7, 2008. Current version published February 4, 2009. This paper was recommended for publication by Associate Editor P. Rives and Editor L. Parker upon evaluation of the reviewers' comments.

The author is with the Ecole Nationale Supérieure des Ingénieurs des Études et Techniques d'Armements (ENSIETA), Développement Technologies Nouvelles, Brest 29806, France, and also with the Groupe d'Étude Sous-Marine de l'Atlantique (GESMA), Brest 29200, France (e-mail: jaulinlu@ensieta.fr).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2008.2010358

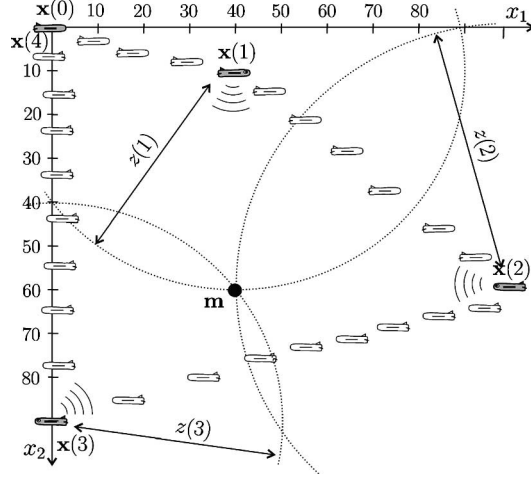


Fig. 1. Underwater robot with two state variables is able to measure the distance between a seamount and itself.

where  $\mathbf{x}(t)$  is the state vector of the robot,  $\mathbf{u}(t)$  is its input vector,  $\mathbf{y}(t)$  is its output vector,  $\mathbf{m}_i$  is the location of the  $i$ th mark,  $\mathbf{z}_i(t)$  is the measurement vector related to the  $i$ th mark, and  $t \in [t_0, t_f]$  is the time. In a set membership context, we shall assume the following.

- 1) For all  $t \in [t_0, t_f]$ , we have boxes  $[\mathbf{u}](t)$  and  $[\mathbf{y}](t)$  enclosing the vectors  $\mathbf{u}(t)$  and  $\mathbf{y}(t)$ . If some components of  $\mathbf{y}$  or  $\mathbf{u}$  are not measured, then the corresponding interval will be  $]-\infty, \infty[$ .
- 2) We have a finite subset  $\mathcal{M} \subset [t_0, t_f] \times \{1, \dots, i_{\max}\}$  and bounded boxes  $[\mathbf{z}_i](t)$ , where  $(t, i) \in \mathcal{M}$ , such that  $\mathbf{z}_i(t) \in [\mathbf{z}_i](t)$ . The number of elements of the set  $\mathcal{M}$  corresponds to the number of times a mark has been detected. If  $(t, i) \in \mathcal{M}$ , then the  $i$ th mark has been detected at time  $t$ .

In a set-membership context, the objective of SLAM is to contract as much as possible all set-membership domains for  $\mathbf{x}(t)$ ,  $\mathbf{m}_1, \dots, \mathbf{m}_{i_{\max}}$ , without removing any feasible value.

### B. Simple Example

In this section, an academic problem involving a 2-D underwater robot is presented. This problem has been chosen for the following reasons: 1) it illustrates the main idea of SLAM; 2) it will be used to show how interval propagation works (see Section III-D); and 3) it is not easily solved using usual SLAM methods such as the extended Kalman filter.

In the environment of the robot, we have one immobile seamount  $\mathbf{m}$  (see Fig. 1). The SLAM problem is described by the following equations:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{u} & \text{(state equation of the robot)} \\ \mathbf{y} = \mathbf{x} & \text{(GPS, when available)} \\ z = \|\mathbf{x} - \mathbf{m}\| & \text{(mark equation).} \end{cases} \quad (2)$$

At time  $t = 0$ , the robot is at the origin, and thus,  $\mathbf{x}(0) = (0, 0)$ . At time  $t = 4$ , the robot is at the surface and is able to measure its location (with the GPS). At time  $t \in \{1, 2, 3\}$ , the robot measures its distance  $z$  to the mark  $\mathbf{m}$ . During its mission, at

each time  $t \in [0, 4]$ , it also measures the input  $\mathbf{u}$  with a known accuracy.

*Remark 1:* For our simulation, the seamount is located at  $\mathbf{m} = (40, 60)$  and the chosen input  $\mathbf{u}(t)$  is given by

$$\begin{array}{c|c|c|c|c} t & [0, 1[ & [1, 2[ & [2, 3[ & [3, 4[ \\ \mathbf{u} & \begin{pmatrix} 40 \\ 10 \end{pmatrix} & \begin{pmatrix} 60 \\ 50 \end{pmatrix} & \begin{pmatrix} -100 \\ 30 \end{pmatrix} & \begin{pmatrix} 0 \\ -90 \end{pmatrix} \end{array}. \quad (3)$$

The exact position of the robot as well as the exact distance to the seamount can thus be computed for all  $t$ . As a consequence, we get the following values for  $\mathbf{x}$  and  $z$ :

$$\begin{array}{c|c|c|c|c} t & 1 & 2 & 3 & 4 \\ \mathbf{x} & \begin{pmatrix} 40 \\ 10 \end{pmatrix} & \begin{pmatrix} 100 \\ 60 \end{pmatrix} & \begin{pmatrix} 0 \\ 90 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ z & 50 & 60 & 50 & - \end{array}. \quad (4)$$

The exact values for  $\mathbf{u}(t)$  and  $\mathbf{m}$  are given here for a better understanding of the example and to allow the reader to check the consistency of the results. But in what follows, the location for  $\mathbf{m}$  is unknown and  $\mathbf{u}(t)$  is measured with a known accuracy.

The quantities that can actually be measured are  $\mathbf{u}(t)$  for each  $t \in [0, 4]$ ,  $z(t)$  for  $t \in \{1, 2, 3\}$ , and  $\mathbf{x}(4)$ . In a set membership context, these measurements translate into the intervals  $[\mathbf{u}](t)$ ,  $[z](t)$ , and  $[\mathbf{x}](4)$ , which enclose the true values for  $\mathbf{u}(t)$ ,  $z(t)$ , and  $\mathbf{x}(4)$ . Define the quantity  $\mathbf{v}(t) = \int_t^{t+1} \mathbf{u}(\tau) d\tau$ ,  $t \in \{0, 1, 2, 3\}$ . From the intervals  $[\mathbf{u}](t)$ , we are able to compute intervals  $[\mathbf{v}](t)$  enclosing  $\mathbf{v}(t)$ ,  $t \in \{0, 1, 2, 3\}$ . Our SLAM problem can then be cast into the following set of equations:

$$\begin{cases} \mathbf{x}(1) = \mathbf{x}(0) + \mathbf{v}(0), \dots, \mathbf{x}(4) = \mathbf{x}(3) + \mathbf{v}(3) \\ z(1) = \|\mathbf{x}(1) - \mathbf{m}\|, \dots, z(3) = \|\mathbf{x}(3) - \mathbf{m}\| \end{cases} \quad (5)$$

where some prior membership intervals (or *domains*) are known for  $\mathbf{v}(0)$ ,  $\mathbf{v}(1)$ ,  $\mathbf{v}(2)$ ,  $\mathbf{v}(3)$ ,  $\mathbf{x}(4)$ ,  $z(1)$ ,  $z(2)$ , and  $z(3)$ . These domains are given by

variable	domain
$z(1)$	$[50 - \delta_z, 50 + 2\delta_z]$
$z(2)$	$[60 - \delta_z, 60 + 2\delta_z]$
$z(3)$	$[50 - \delta_z, 50 + 2\delta_z]$
$\mathbf{v}(0)$	$[40 - \delta_v, 40 + 2\delta_v]$ $[10 - \delta_v, 10 + 2\delta_v]$
$\mathbf{v}(1)$	$[60 - \delta_v, 60 + 2\delta_v]$ $[50 - \delta_v, 50 + 2\delta_v]$
$\mathbf{v}(2)$	$[-100 - \delta_v, -100 + 2\delta_v]$ $[30 - \delta_v, 30 + 2\delta_v]$
$\mathbf{v}(3)$	$[-\delta_v, 2\delta_v]$ $[-90 - \delta_v, -90 + 2\delta_v]$
$\mathbf{x}(4)$	$[-\delta_x, 2\delta_x]$ $[-\delta_x, 2\delta_x]$

where  $\delta_z$ ,  $\delta_v$ , and  $\delta_x$  are positive real numbers. Note that all domains have the form  $[a] = [\hat{a} - \delta_a, \hat{a} + 2\delta_a]$ , where  $\hat{a}$  is the true value for the variable  $a$  and  $\delta_a$  is a tuning error coefficient. These intervals have been chosen in order to satisfy the following properties: 1)  $[a]$  always encloses the true value  $\hat{a}$  for  $a$ ;

2) if  $\delta_a = 0$ , then  $[a]$  becomes the degenerated interval  $[\hat{a}, \hat{a}]$  (i.e., a singleton  $\{\hat{a}\}$ ); 3) if  $\delta_a > 0$ , then the center of  $[a]$  is not  $\hat{a}$ ; this choice makes possible to avoid a particular unrealistic situation that could be favorable to some estimation methods. Any other arbitrary choice for the interval domains that would satisfy these three properties could have been done as well. Again, remember that the components of  $\mathbf{x}(1)$ ,  $\mathbf{x}(2)$ ,  $\mathbf{x}(3)$ , and  $\mathbf{m}$  are not measured, and thus, the associated prior intervals are  $]-\infty, \infty[$ .

Interval propagation is a set of numerical methods that make possible to contract the interval domains for the variables, without losing any feasible value. After a brief presentation of interval propagation, we will illustrate the efficiency of the approach on our simple SLAM problem.

### III. INTERVAL PROPAGATION

With an interval approach, a random variable  $x$  of  $\mathbb{R}$  is often represented by an interval  $[x]$  that encloses the support of its probability function. This representation is, of course, poorer than that provided by its probability density distribution, but it presents several advantages; 1) since an interval with non zero length is consistent with an infinite number of probability distribution functions, an interval representation is well adapted to represent random variables with imprecise probability density functions; 2) an arithmetic can be developed for intervals, which makes it possible to deal with uncertainties in a reliable and easy way, even when strong nonlinearities occur; 3) when the random variables are related by constraints (i.e., equations or inequalities), a propagation process (which will be explained later) makes it possible to get efficient polynomial algorithms to compute precise intervals that are guaranteed to contain all feasible values for the random variables.

#### A. Interval Arithmetic

An interval is a closed and connected subset of  $\mathbb{R}$ . Consider two intervals  $[x]$  and  $[y]$ , and an operator  $\diamond \in \{+, -, \dots, /\}$ , we define  $[x] \diamond [y]$  as the smallest interval that contains all feasible values for  $x \diamond y$ , if  $x \in [x]$  and  $y \in [y]$  (see [24]). If  $f$  is an elementary function such as  $\sin, \cos, \dots$ , we define  $f([x])$  as the smallest interval that contains all feasible values for  $f(x)$  if  $x \in [x]$ . Consider a constraint  $\mathcal{C}$  (i.e., an equation or an inequality), some variables  $x_1, x_2, \dots$  involved in  $\mathcal{C}$ , and prior interval domains  $[x_i]$  for the  $x_i$ 's. Interval arithmetic makes it possible to contract the domains  $[x_i]$  without removing any feasible values for the  $x_i$ 's. For instance, consider the equation  $z = x + y$ , where the domains for  $x, y$ , and  $z$  are given by  $[x] = [-\infty, 5]$ ,  $[y] = [-\infty, 4]$ , and  $[z] = [6, \infty]$ . These domains can be contracted to  $[x'] = [2, 5]$ ,  $[y'] = [1, 4]$ , and  $[z'] = [6, 9]$ . The resulting interval calculus is as follows:

$$\begin{aligned} z = x + y &\Rightarrow z \in [6, \infty] \cap ([-\infty, 5] + [-\infty, 4]) = [6, 9] \\ x = z - y &\Rightarrow x \in [-\infty, 5] \cap ([6, \infty] - [-\infty, 4]) = [2, 5] \\ y = z - x &\Rightarrow y \in [-\infty, 4] \cap ([6, \infty] - [-\infty, 5]) = [1, 4]. \end{aligned}$$

A contraction operator is called a *contractor*.

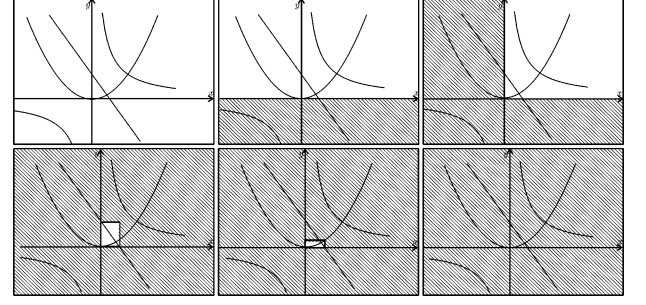


Fig. 2. Illustration of the propagation procedure.

#### B. Propagation

When several constraints are involved, the contractions are performed sequentially until no more significant contraction can be observed (see [15] for more details). To illustrate the propagation process, consider the following three equations:

$$\begin{cases} (C_1): y = x^2 \\ (C_2): xy = 1 \\ (C_3): y = -2x + 1. \end{cases} \quad (6)$$

Using interval propagation, we want to prove that this system has no solution. To each of the variables, we assign the domain  $[-\infty, \infty]$ . Then, we contract the domains with respect to the constraints in the following order:  $C_1, C_2, C_3, C_1, C_2$ , and we get empty intervals for  $x$  and  $y$ . A geometric interpretation of the propagation is given in Fig. 2. The resulting interval computation is as follows:

$$\begin{aligned} (C_1) &\Rightarrow y \in [-\infty, \infty]^2 = [0, \infty] \\ (C_2) &\Rightarrow x \in 1/[0, \infty] = [0, \infty] \\ (C_3) &\Rightarrow y \in [0, \infty] \cap ((-2) \cdot [0, \infty] + 1) = [0, 1] \\ x &\in [0, \infty] \cap (-[0, 1]/2 + 1/2) = \left[0, \frac{1}{2}\right] \\ (C_1) &\Rightarrow y \in [0, 1] \cap [0, 1/2]^2 = [0, 1/4] \\ (C_2) &\Rightarrow x \in [0, 1/2] \cap 1/[0, 1/4] = \emptyset. \end{aligned}$$

*Remark 2:* The interval propagation method converges to a box that encloses all solutions (if there exists any), but the box is not necessarily the smallest one. The box is said to be *locally consistent* because it is consistent with all constraints taken independently. The smallest box that encloses all solutions is said to be *globally consistent*. The problem of computing this smallest box is NP-hard, and can be solved using bisection methods only for problems involving few variables. In this paper, since we want to solve large-dimensional problems, local consistency methods will be used.

#### C. Complex Constraints

Contractors can be developed for complex constraints, but an adaptation is often required (see, e.g., [15]). For instance, if we

consider the constraint  $\dot{\mathbf{p}}(t) = \mathbf{R}(t)\mathbf{v}_r(t)$ , where  $\forall t \in [t_0, t_1]$ ,  $\mathbf{R}(t) \in [\mathbf{R}]$  and  $\mathbf{v}_r(t) \in [\mathbf{v}_r]$ . Since

$$\mathbf{p}(t_1) = \mathbf{p}(t_0) + \int_{t_0}^{t_1} \mathbf{R}(t) \cdot \mathbf{v}_r(t) \in \mathbf{p}(t_0) + (t_1 - t_0) \cdot [\mathbf{R}] \cdot [\mathbf{v}_r]$$

the domains for  $\mathbf{p}(t_0)$  and  $\mathbf{p}(t_1)$  can be contracted as follows:

$$\begin{aligned} [\mathbf{p}](t_1) &= [\mathbf{p}](t_1) \cap ([\mathbf{p}](t_0) + (t_1 - t_0) \cdot [\mathbf{R}] \cdot [\mathbf{v}_r]) \\ [\mathbf{p}](t_0) &= [\mathbf{p}](t_0) \cap ([\mathbf{p}](t_1) + (t_0 - t_1) \cdot [\mathbf{R}] \cdot [\mathbf{v}_r]). \end{aligned}$$

For more information on the contraction of constraints described by differential equations; see, e.g., [13]. There exist a lot of sophisticated methods to build contractors adapted to complex constraints. Some of them combine interval methods with formal calculus (for instance, by computing the derivatives of the functions involved in the equations). Others have been built for specific important constraints (such as distance or angle constraints).

#### D. Forward-Backward Propagation

The interval propagation method converges to a box that contains all solution vectors of our set of constraints. If this box is empty, it means that there is no solution. It can be shown that the box to which the method converges does not depend on the order to which the contractors are applied [15], but the computing time is highly sensitive to this order. There is no optimal order in general, but, in practice, one of the most efficient is called *forward-backward propagation*. It consists of writing the whole set of equations under the form  $\mathbf{f}(\mathbf{x}) = \mathbf{y}$ , where  $\mathbf{x}$  and  $\mathbf{y}$  correspond to quantities that can be measured (i.e., some prior interval domains are given for them). Then, using interval arithmetic, the intervals are propagated from  $\mathbf{x}$  to  $\mathbf{y}$  in a first step (*forward propagation*) and, in a second step, the intervals are propagated from  $\mathbf{y}$  to  $\mathbf{x}$  (*backward propagation*). The forward-backward propagation is then repeated until no more significant contraction can be observed. To illustrate the principle, consider the equations (5). If  $\mathbf{x} \stackrel{\text{def}}{=} (\mathbf{x}(0), \mathbf{v}(0), \dots, \mathbf{v}(3), \mathbf{m})$  and  $\mathbf{y} \stackrel{\text{def}}{=} (z(1), z(2), z(3), \mathbf{x}(1), \dots, \mathbf{x}(4))$ , then (5) can be rewritten under the form  $\mathbf{f}(\mathbf{x}) = \mathbf{y}$ . The forward contraction can be described by the following algorithm:

FORWARD CONTRACTION	
1	for $k = 0$ to 3
2	$[\mathbf{x}](k+1) = ([\mathbf{x}](k) + [\mathbf{v}](k)) \cap [\mathbf{x}](k+1)$
3	end
4	for $k = 1$ to 3
5	$[\mathbf{d}](k) = [\mathbf{x}](k) - [\mathbf{m}]$
6	$[z](k) = \sqrt{([\mathbf{d}_1](k))^2 + ([\mathbf{d}_2](k))^2} \cap [z](k)$
7	end

where  $[\mathbf{d}](k)$  is the box  $[d_1](k) \times [d_2](k)$ . The backward propagation is described by

BACKWARD CONTRACTION	
1	for $k = 3$ to 1
2	$[d_1](k) = \sqrt{([z](k))^2 - ([d_2](k))^2} \cap [d_1](k)$
3	$[d_2](k) = \sqrt{([z](k))^2 - ([d_1](k))^2} \cap [d_2](k)$
4	$[\mathbf{x}](k) = ([\mathbf{d}](k) + [\mathbf{m}]) \cap [\mathbf{x}](k)$
5	$[\mathbf{m}] = ([\mathbf{x}](k) - [\mathbf{d}](k)) \cap [\mathbf{m}]$
6	end
7	for $k = 3$ to 0
8	$[\mathbf{x}](k) = ([\mathbf{x}](k+1) - [\mathbf{v}](k)) \cap [\mathbf{x}](k)$
9	$[\mathbf{v}](k) = ([\mathbf{x}](k+1) - [\mathbf{x}](k)) \cap [\mathbf{v}](k)$
10	end.

#### E. Results on Our Simple SLAM Example

Again consider the simple SLAM example of Section II-B. If we apply an elementary interval propagation for different values of  $\delta_x$ ,  $\delta_v$ , and  $\delta_z$ , we get the contracted intervals given in the following table.

	case 1	case 2	case 3
$z(1)$	[50]	[50]	[49, 52]
$z(2)$	[60]	[60]	[59, 62]
$z(3)$	[50]	[50]	[49, 52]
$\mathbf{m}$	[40] [60]	[38, 44.23] [58.93, 62]	[36, 45.34] [57.58, 64]
$\mathbf{v}(0)$	[40] [10]	[39, 42] [9, 12]	[39, 42] [9, 12]
$\mathbf{v}(1)$	[60] [50]	[59, 62] [49, 52]	[59, 62] [49, 52]
$\mathbf{v}(2)$	[-100] [30]	[-101, -98] [29, 32]	[-101, -98] [29, 32]
$\mathbf{v}(3)$	[0] [-90]	[-1, 2] [-91, -88]	[-1, 2] [-91, -88]
$\mathbf{x}(1)$	[40] [10]	[39, 42] [9, 12]	[39, 42] [9, 12]
$\mathbf{x}(2)$	[100] [60]	[98, 104] [58, 64]	[98, 104] [58, 64]
$\mathbf{x}(3)$	[0] [90]	[-3, 6] [87.49, 94.87]	[-3, 6] [87, 96]
$\mathbf{x}(4)$	[0] [0]	[-4, 8] [-4, 6.54]	[-4, 8] [-4, 8]

*Case 1:*  $\delta_v = 0, \delta_z = 0, \delta_x = \infty$ . We are in the unrealistic situation with no noise on the evolution and no noise on the distance measurements. An interval propagation contracts the domains for  $\mathbf{m}, \mathbf{x}(1), \mathbf{x}(2), \mathbf{x}(3)$ , and  $\mathbf{x}(4)$  (which were initially equal to  $] - \infty, \infty[^2$ ) to singletons. The obtained results are consistent with the exact solution.

*Case 2:*  $\delta_v = 1, \delta_z = 0, \delta_x = \infty$ . The evolution is now noisy and we are still able to localize the mark with a rather good accuracy. The localization of the mark allows a small improvement of the accuracy on the trajectory [see the domain for  $\mathbf{x}(3)$ ].

*Case 3:*  $\delta_v = 1, \delta_z = 1, \delta_x = \infty$ . Now, there is also an error in the distance measurement. The location of the mark

is less accurate than that for case 2. This additional error prevents any improvement in the accuracy of the trajectory envelope.

Other cases have been considered. For instance, if  $\delta_x$  is small, the localization of the mark as well as the estimation of the trajectory becomes more accurate. The computing time to get the contracted intervals is always less than 0.1 s on a standard laptop.

#### IV. SLAM IN A SUBMARINE CONTEXT

##### A. Mine Hunters

We shall consider a class of autonomous underwater robots, which are called *mine hunters*. These robots are used to detect near-shore underwater mines. Most mine hunters are equipped with a loch-Doppler, a gyrocompass, an altimeter, and a barometer. They also enclose one or two sidescan sonars to localize seamounts such as rocks or mines. The SLAM problem of mine hunters can be described by the SLAM equations (1), as will be shown now.

##### B. Evolution Equation

Around the zone covered by the robot, let us build the frame  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$ , where the vector  $\mathbf{i}$  indicates the north,  $\mathbf{j}$  indicates the east, and  $\mathbf{k}$  is oriented toward the center of the earth. Denote by  $\mathbf{p} = (p_x, p_y, p_z)$  the coordinates of the robot expressed in the frame  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$ . The robot motion can be described by

$$\dot{\mathbf{p}} = \mathbf{R}_{\text{Euler}}(\varphi, \theta, \psi) \cdot \mathbf{v}_r \quad (7)$$

where  $\mathbf{v}_r$  represents the speed vector of the robot measured by a loch-Doppler and  $\mathbf{R}_{\text{Euler}}(\varphi, \theta, \psi)$  is the Euler rotation matrix returned by the gyrocompass (the roll is  $\varphi$ , the pitch is  $\theta$ , and the head is  $\psi$ ). Equation (7) corresponds to the first equation of (1), where the state vector is  $\mathbf{p}$  and the input vector is  $\mathbf{u} = (\varphi, \theta, \psi, v_r^x, v_r^y, v_r^z)$ .

##### C. Observation Equation

The observation equation is given by  $\mathbf{y} = \mathbf{p}$  and corresponds to the second equation of (1). The first two coordinates  $p_x$  and  $p_y$  of  $\mathbf{p}$  can be measured by the GPS with an accuracy less than 2.5 m. Since electromagnetic waves (here, around 1.2 GHz) do not propagate through the water, the GPS is operational only when the robot is at the surface of the ocean. During a typical underwater mission, the robot is able to measure only its location with the GPS before diving and once it comes back to the surface. A barometer is used to measure the depth  $p_z$  of the robot (i.e., the distance between the robot and the surface of the ocean).

##### D. Mark Equation

To detect mines, a mine hunter uses a sidescan sonar that has either an antenna starboard or an antenna portside. It may also have two antennas on both sides. Sometimes, a mine hunter uses a camera located below it and oriented toward the bottom. A sidescan sonar sends a *ping*, i.e., a short ultrasonic wave on

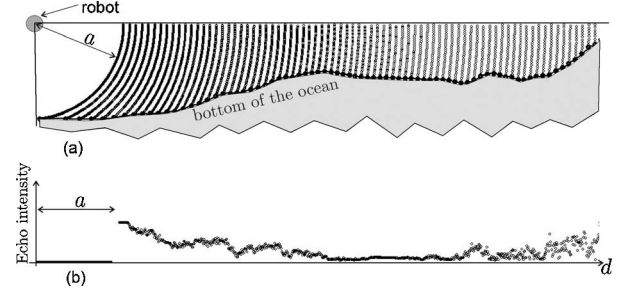


Fig. 3. When the robot emits a ping (i.e., a short ultrasonic sound), it receives an echo. (a) Robot and the propagation of the sound. (b) Intensity of the corresponding echo with respect to the distance. The quantity  $a$  is the distance between the robot and the bottom. When  $d < a$ , no echo can be detected.

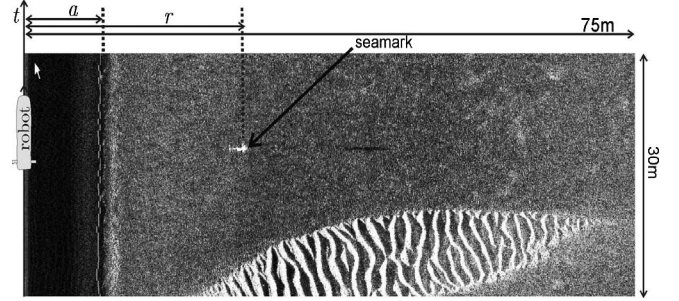


Fig. 4. Slice of the waterfall built by a sonar with one antenna starboard; this sonar image makes it possible to detect a seamount (here, a mine) and compute the distance  $r$  between the seamount and the robot.

a thin plane located on the right or on the left of the robot and perpendicular to its main axis. It receives an echo, as illustrated by Fig. 3.

From all these pings, the sidescan sonar builds a long image, and a horizontal slice of such an image is represented by Fig. 4. The image, called a *waterfall*, has a length that corresponds to the distance covered by the robot during its mission. Its width corresponds to the product of the distance covered by the sonar by the number of antennas (one or two). After the mission, a scrolling of the waterfall is performed by a human operator. He is then able to get an estimation  $\tilde{r}(t)$  of the distance  $r(t)$  between the robot and the seamount detected at time  $t$ . From the width of the black vertical strip on the left of the picture (called the *water column*), we are also able to compute an estimation  $\tilde{a}(t)$  of the altitude  $a(t)$  of the robot (distance between the robot and the bottom), but this estimation is less precise and less robust (i.e., it can generate outliers) than that provided by the loch-Doppler. Thus, we will not take it into account for the SLAM.

When the human operator detects a seamount on the waterfall, he is able to see if it is on the right or on the left of the robot, and he is also able to measure the distance between the robot and the mine. As a consequence, we shall define the *lateral distance* between the robot at position  $\mathbf{p}$  and the  $i$ th mark  $\mathbf{m}_i$  as follows:

$$r(\mathbf{p}, \psi, \theta, \varphi, \mathbf{m}_i) \stackrel{\text{def}}{=} \begin{cases} \frac{-\|\mathbf{e}_i\|}{\text{sign}(\mathbf{j}_r^T \cdot \mathbf{e}_i)}, & \text{if } \begin{cases} \mathbf{i}_r^T \cdot \mathbf{e}_i = 0 \\ \mathbf{k}_r^T \cdot \mathbf{e}_i \leq 0 \end{cases} \\ \infty, & \text{otherwise} \end{cases} \quad (8)$$

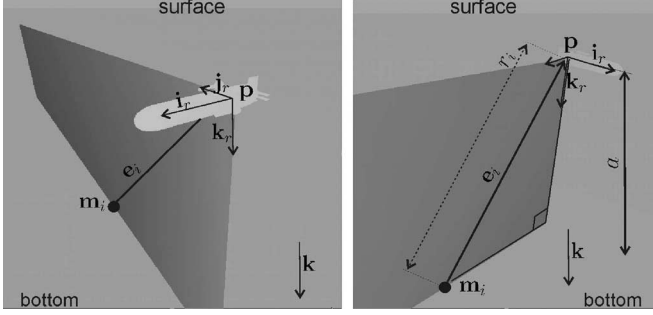


Fig. 5. Distance between the robot and a seamount obtained using a lateral sidescan sonar with an antenna starboard.

where  $\mathbf{i}_r$ ,  $\mathbf{j}_r$ , and  $\mathbf{k}_r$  are the three column vectors of the rotation matrix  $\mathbf{R}_{\text{Euler}}(\varphi, \theta, \psi)$  and

$$\mathbf{e}_i \stackrel{\text{def}}{=} \mathbf{p} - \mathbf{m}_i. \quad (9)$$

Let us now explain this definition (see Fig. 5). The lateral sonar is able to detect only a mark that is on its lateral plane, perpendicular to the main axis of the robot. Thus, if the mark is outside this plane (i.e.,  $\mathbf{i}_r^T \cdot \mathbf{e}_i \neq 0$ ), the lateral distance will be undefined (i.e., equal to  $\infty$ ). Moreover, the detected mark is always below the robot (i.e.,  $\mathbf{k}_r^T \cdot \mathbf{e}_i \leq 0$ ); otherwise, the lateral distance is undefined (i.e., equal to  $\infty$ ). If the mark is starboard (i.e.,  $\mathbf{j}_r^T \cdot \mathbf{e}_i \leq 0$ ), then, by convention, the lateral distance  $r_i$  is positive and equal to  $\|\mathbf{e}_i\|$ . If it is portside (i.e.,  $\mathbf{j}_r^T \cdot \mathbf{e}_i \geq 0$ ), then  $r_i = -\|\mathbf{e}_i\|$ . Moreover, if we assume that the ocean bottom is approximately flat (at least locally), the altitude of the robot and its depth can be used to get a measurement of the depth of the seamount by the relation

$$m_i^z = a + p_z + s_i \quad (10)$$

where  $s_i$  is a variable that could be bounded if we could give a bound on the slopes of the bottom of the ocean in the area covered by the robot. As a consequence, the mark equation is composed of the lateral distance of the seamount and the depth of the seamount. It has the form

$$\mathbf{z}_i = \mathbf{h}(\mathbf{p}, \mathbf{u}, \mathbf{m}_i) = \begin{pmatrix} r(\mathbf{p}, \psi, \theta, \varphi, \mathbf{m}_i) \\ m_i^z \end{pmatrix}. \quad (11)$$

Note that, since  $r(\mathbf{p}, \psi, \theta, \varphi, \mathbf{m}_i)$  is given by (8), the function  $\mathbf{h}$  is not continuous.

### E. SLAM Formulation for Mine Hunters

The previous developments lead to the following SLAM formulation [see (1)]:

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{R}_{\text{Euler}}(\varphi, \theta, \psi) \cdot \mathbf{v}_r \\ \mathbf{y} = \mathbf{p} \\ \mathbf{z}_i = \begin{pmatrix} r(\mathbf{p}, \psi, \theta, \varphi, \mathbf{m}_i) \\ m_i^z \end{pmatrix} \end{cases} \quad (12)$$

where  $\mathbf{u} = (\varphi, \theta, \psi, v_r^x, v_r^y, v_r^z)$ . Recall that in the context of mine hunters, we have accurate interval domains for all components of  $\mathbf{u}(t) \forall t \in t \in [t_0, t_f]$ . We also have accurate domains for  $r_i(t)$  and  $m_i^z$  for any pair  $(t, i)$  corresponding to a detection.

An interval propagation can thus be launched to find accurate interval domains for all other variables involved in our set of equations.

*Remark 3:* If a scalar decomposition of the vector equations is performed, the set of constraints associated with the SLAM problem involves  $n_t n_p + 3n_m$  unknown variables that cannot be measured, where  $n_t$  is the number of sampling times,  $n_p = 3$  is the dimension of  $\mathbf{p}$ , and  $n_m$  is the number of mines. The number of equations is  $n_t (n_p + n_z)$ , where  $n_z$  is the dimension of  $\mathbf{z}_i$ . For the actual experiment to be considered in Section V, we will have  $n_t = 60\,000$ ,  $n_m = 6$ , and thus, we have a problem with 180 018 variables for 300 000 equations. The fact that we have more equations than unknowns creates the redundancy needed by interval propagation methods to solve efficiently and rigorously such high-dimension problems.

### F. GESMI

In Section III, we have shown that a SLAM problem could be cast into a set of nonlinear constraints where interval domains were available for each variable. We have also presented an interval propagation approach to contract domains for the variables, and we have illustrated how interval propagation can be used on the simple SLAM problem presented in Section II-B. An interval SLAM solver named guaranteed estimation of seamounts with intervals (GESMI) [14] has been developed. The interval propagation performed by GESMI is mainly based on a forward-backward interval propagation (see Section III-D). The reason for choosing a forward-backward strategy is mainly due to the large number of constraints involved in our problem. There exist two versions of GESMI: one running under a *Windows* environment with a nice *OpenGL*-based interface and one running under *Linux* without graphical interface. Up to now, the *Windows* version does not implement the outward rounding needed to get the guarantee of the results with respect to rounding errors. On the other hand, the *Linux* version is based on the validated interval library IBEX (<http://ibex-lib.org/>) and performs an outward rounding for each interval operation. The *Windows* solver as well as the data collected during the experiment presented in the next section have been made available [14]. The software is easy to use and allows the reader to get a quick idea about the efficiency of the method.

## V. EXPERIMENTAL SETUP

### A. Description of the Robot

The robot to be considered in this experiment (see Fig. 6) is a mine hunter, named *Redermor* (means *sea runner*, in the Breton language). This robot, developed by the Groupe d'Etude Sous-Marine de l'Atlantique (GESMA), a center of the French Defense Ministry that supervises most of the research in French underwater robotics, has a length of 6 m, a diameter of 1 m, and a weight of 3800 kg. It has powerful propulsion and control system able to provide hovering capabilities. The main purpose of the *Redermor* is to evaluate improved navigation by the use of sonar information. Note that the use of sonars for improving

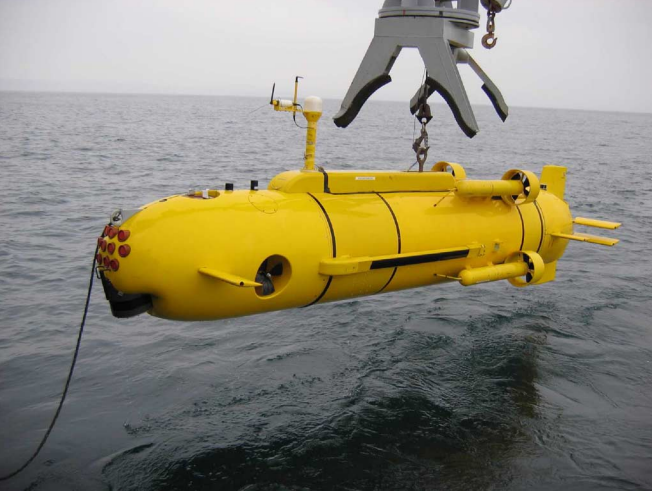


Fig. 6. Autonomous underwater vehicle, *Redermor*, built by the GESMA.

the localization and the navigation has already been proven on several robotics applications (see, e.g., [18]).

### B. Description of the Mission and Sensors

The experiment to be presented now has been performed by the GESMA in the Douarnenez bay, Brittany, France. At time  $t_0 = 0.0$  s, the robot has been dropped approximately around the position  $\tilde{\ell}^0 = (\tilde{\ell}_x^0, \tilde{\ell}_y^0) = (-4.458227931^\circ, 48.212920614^\circ)$ , measured by the GPS, where  $\tilde{\ell}_x^0$  is the west/east longitude and  $\tilde{\ell}_y^0$  is the south/north latitude. The error of the GPS is known to be less than  $0.0000135^\circ$  for  $\tilde{\ell}_x^0$  and less than  $0.0000090^\circ$  for  $\tilde{\ell}_y^0$  in this part of the earth that corresponds approximately to a position error less than 2.5 m. When the robot surfaces, at time  $t_f = 5999.4$  s, its position is also measured as  $\tilde{\ell}^f = (\tilde{\ell}_x^f, \tilde{\ell}_y^f) = (-4.454660760^\circ, 48.219129760^\circ)$ , with the same accuracy. The loch-Doppler is a *Workhorse Navigator Doppler Velocity Log* from *RD-instruments company*. The frequency of the waves are around 300 kHz. The actual speed (in meters per second) is known to satisfy

$$\tilde{\mathbf{v}}_r \in \tilde{\mathbf{v}}_r + 0.004 * [-1, 1] \cdot \tilde{\mathbf{v}}_r + 0.004 * [-1, 1] \quad (13)$$

where  $\tilde{\mathbf{v}}_r$  denotes the 3-D speed vector returned by the sensor (see [www.rdinstruments.com/navigator.html](http://www.rdinstruments.com/navigator.html) for more information about these bounds). The loch-Doppler is also able to provide the altitude  $a$  of the robot with an error less than 10 cm. The gyrocompass that has been used is an Octans III from IXSEA. If we denote by  $(\tilde{\varphi}, \tilde{\theta}, \tilde{\psi})$  the angles (in radians) returned by our gyrocompass, then the actual Euler angles for our robot should satisfy (see [www.ixsea.com](http://www.ixsea.com) for the technical characteristics of the sensor)

$$\begin{pmatrix} \tilde{\varphi} \\ \tilde{\theta} \\ \tilde{\psi} \end{pmatrix} \in \begin{pmatrix} \tilde{\varphi} \\ \tilde{\theta} \\ \tilde{\psi} \end{pmatrix} + \begin{pmatrix} 1.75 \times 10^{-4} \cdot [-1, 1] \\ 1.75 \times 10^{-4} \cdot [-1, 1] \\ 5.27 \times 10^{-3} \cdot [-1, 1] \end{pmatrix}. \quad (14)$$

If  $\tilde{d}$  is the depth (in meters) collected by the barometer, then the actual depth of the robot satisfies  $\tilde{p}_z(t) \in [-1.5, 1.5] + \tilde{d} +$

$0.2 * [-1, 1] \cdot \tilde{d}$ . The interval  $[-1.5, 1.5]$  may change depending on the strength of waves and tides.

Thus, for each time  $t \in \mathcal{T} \stackrel{\text{def}}{=} \{0.0, 0.1, 0.2, \dots, 5999.4\text{s}\}$ , the vector of measurements  $(\tilde{\varphi}(t), \tilde{\theta}(t), \tilde{\psi}(t), \tilde{v}_r^x(t), \tilde{v}_r^y(t), \tilde{v}_r^z(t), \tilde{a}(t), \tilde{d}(t))$  is collected. This vector can be represented by a point in the 8-D sensor space (where 8 is the number of corresponding sensors) and approximates the actual value

$$(\tilde{\varphi}(t), \tilde{\theta}(t), \tilde{\psi}(t), \tilde{v}_r^x(t), \tilde{v}_r^y(t), \tilde{v}_r^z(t), \tilde{a}(t), \tilde{p}_z(t)) \quad (15)$$

for the measurement vector, with an error that can be bounded by taking into account the characteristics of the sensors. For each  $t \in \mathcal{T}$ , we are thus able to get a box in  $\mathbb{R}^8$  that contains the actual measurement vector.

The sidescan sonar used for detection is a KLEIN 5400 (see [www.l-3klein.com](http://www.l-3klein.com) for detailed characteristics of this sensor). It has a single-antenna-located starboard (i.e., on right-hand side) and its scope is about 75 m. Every 0.1 s, the sonar sends a *ping*, i.e., a short ultrasonic wave with frequency 455 kHz, on a thin plane located on its right and perpendicular to the main axis of the robot. From the sonar waterfall, provided by the robot after its mission, six mines  $\mathbf{m}_0, \dots, \mathbf{m}_5$  have been detected manually. From these detections, the following *mark table* can thus be built:

$t(\text{sec})$	$i$	$\tilde{r}_i$ (meters)
1054	1	52.42
1092	2	12.47
1374	1	54.40
1748	0	52.68
3038	1	27.73
3688	5	26.98
4024	4	37.90
4817	3	36.71
5172	3	37.37
5232	4	31.03
5279	5	33.51
5688	1	15.05

The table provides: 1) the time  $t$  a mine has been detected starboard; 2) the number  $i$  of the detected mine; and 3) a measure  $\tilde{r}_i(t)$  of the lateral distance between the robot and the mine  $\mathbf{m}_i$ . The actual distance  $\hat{r}_i(t)$  is assumed to satisfy the relation  $\hat{r}_i(t) \in [\tilde{r}_i(t) - 1, \tilde{r}_i(t) + 1]$ . The set  $\mathcal{M}$  of all  $(t, i)$  such that the  $i$ th mine has been detected at time  $t$  can be built from the mark table. We get

$$\mathcal{M} = \{(1054, 1), (1092, 2), (1373, 1), \dots, (5279, 5), (5688, 1)\}.$$

For the flatness characteristics of the bottom, we took a maximum slope of 1%, which makes a measurement of the depth of the mine  $m_i^z$  with an accuracy less than  $0.1 + 2.5 + 0.01 * 75 = 3.35$  m (0.1 m for the accuracy of  $a$ , 2.5 m for  $p_z$ , and 75 m for the distance covered by the lateral sonar). Fig. 4 is related to the detection of a mine (here, it is a spherical floating mine connected to its sinker by a tether).



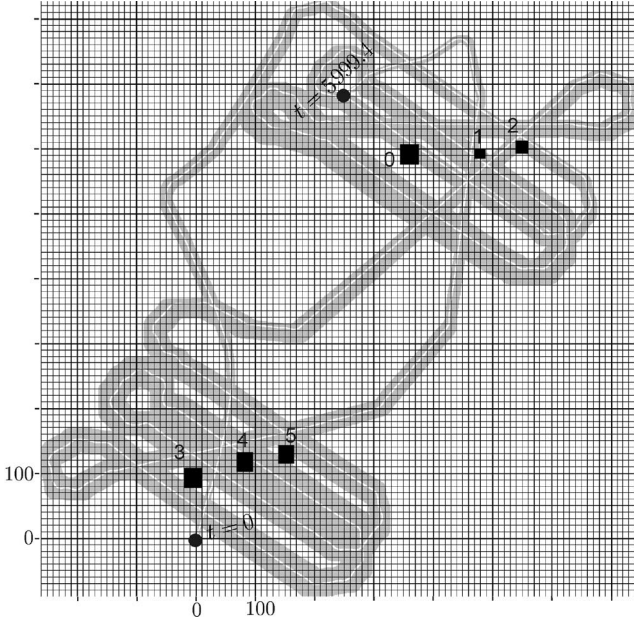


Fig. 7. Envelope enclosing the trajectory of the robot and the six boxes containing the six mines detected by the sonar.

### C. Results

We applied GESMI to solve the SLAM problem related to this experiment. The results obtained after ten interval forward-backward propagations are illustrated in Fig. 7. The envelope painted gray encloses the trajectory of the robot, the six black boxes contain the six mines detected by the sonar, and the two black circles represent the initial and final poses of the robot. These enclosures result from a merge of the information given by the GPS (available at the beginning and at the end of the mission), the detected mines, the loch-Doppler, the gyrocompass, and the barometer. The accuracy of the locations of the robot and the mines is always better than 30 m. The computing time to get these results is less than 30 s with a Pentium III under the Windows version of GESMI (without the outward rounding). With the Linux version of GESMI, which performs an outward rounding, the computing time is less than 55 s. Note that both versions provide exactly the same interval domains (up to five significant digits).

*Remark 4:* We are not supposed to know the location of these six mines. However, when we dropped them, we measured our location with a GPS, and we used this information to check the consistency of results obtained by the interval propagation.

*Remark 5:* Remember that our interval estimation has been done after the mission, and thus, the robot did not use SLAM to control its trajectory. During its mission, the robot localizes itself using a dead reckoning approach (i.e., it does not use the lateral sonar, but only the loch-Doppler, the gyrocompass, and the barometer). The robot tries to follow a trajectory using an elementary PID controller. The trajectory has been defined through a list of waypoints that have to be reached. These waypoints have been given by the user before the mission. Dead reckoning generates a drift that can be estimated here to 100 m for 1-h mission, and thus, the waypoints have been reached with

a bad precision. However, this can be considered as sufficient for controlling the robot for a 2-h mission in the ocean without being lost. Now, the aim of our robot is to detect and localize mines. This localization should be as precise as possible (less than 30 m) in order to be able to send another teleoperated robot equipped with cameras, to destroy the detected mines. Such a reliable accuracy of 30 m could not have been obtained without using a SLAM approach.

### D. Waterfall

Fig. 8 contains the reconstructed waterfall (left side) and one zoom (right side). Each column corresponds to one of the six mines ( $i = 0, \dots, 5$ ). The gray areas contain the set of all feasible pairs  $(t, \|e_i(t)\|)$  obtained after the interval propagation. The 12 small black disks correspond to the mines detected by the human operator. From each disk, we can get the time  $t$  at which the mine has been detected ( $y$ -axis), the number of the mine, and the distance  $r_i$  between the robot and the mine ( $x$ -axis). Thus, from the 12 black disks in the figure, we are able to reconstruct the mark table. Black areas correspond to all feasible  $(t, r_i)$ . Some of these areas are tiny and are covered by a black disk. Some are larger and do not contain any black disk. In such a case, an existing mine has been missed by the operator during the scrolling of the waterfall. As a consequence, with the help of Fig. 8, the operator could again scan the waterfall and find undetected mines much more efficiently. If the operator is able to detect at least one more mark, then the propagation algorithm can run again to get a thinner envelope for the trajectory, thinner black areas in the reconstructed waterfall, and thus, a higher probability to detect new mines on the waterfall. The operator can thus be seen as a contractor inside an interval propagation process. Up to now, the detection procedure is done by hand. Now, new techniques for automatic detections of seamounts are more and more reliable (see, e.g., [28] in the situation where the seamounts are mines). Such automatic detections could, of course, be included in our propagation process.

## VI. COMPARISON WITH EXISTING METHODS

The class of SLAM problems treated with the interval propagation approach can also be treated using more classical Bayesian smoothers. For instance, in the experiment considered in the previous section, a Kalman smoother leads to the trajectory envelope represented by Fig. 9. To apply a Kalman smoother, the Euler angles  $\varphi$ ,  $\theta$ , and  $\psi$  are assumed to be measured without any error (otherwise, the noise would become multiplicative and Kalman filter would not apply) and the mark equation has to be linearized. For our application, we observed that the resulting envelope as well as the ellipses for the marks are accurate and consistent with the results obtained by an interval propagation. This consistency of the results is mainly due to the fact that linearization as well as the assumption on the exact knowledge of the Euler angles do not affect the quality of the results. Let us now give a list of comparisons between the interval propagation approach and other more classical Bayesian approaches for SLAM.



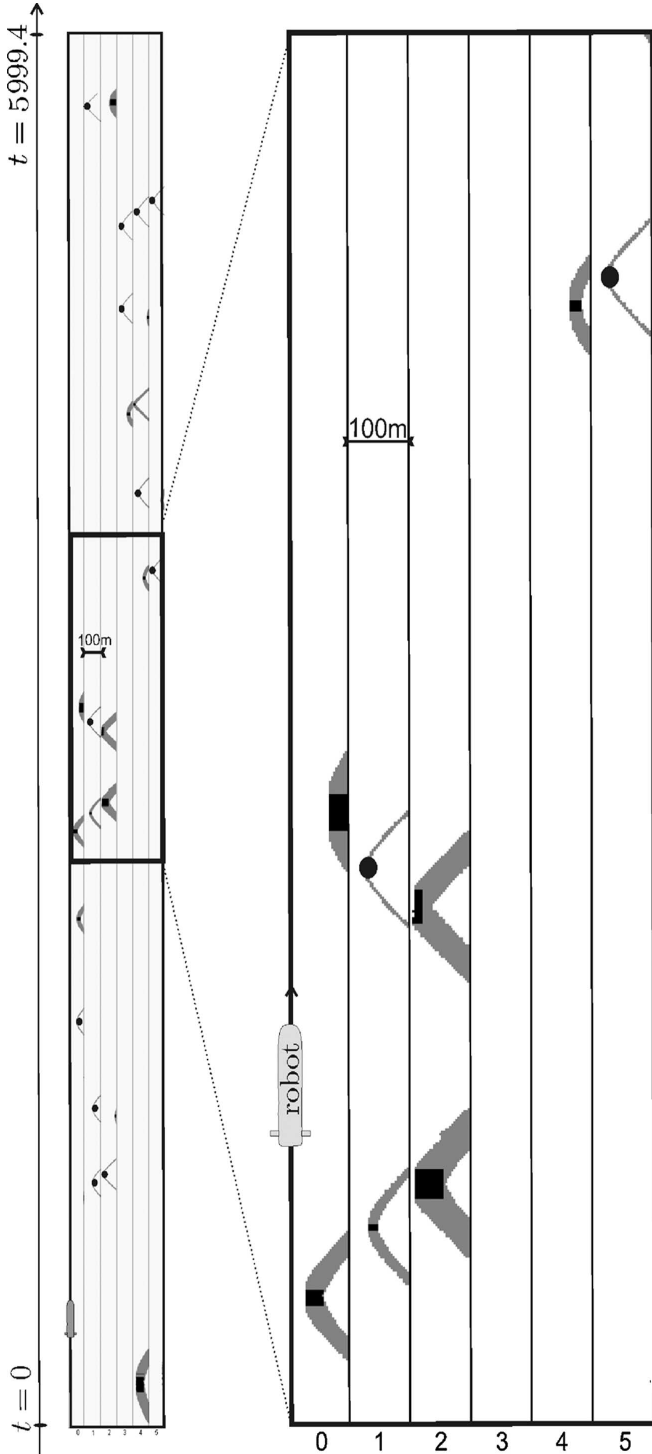


Fig. 8. Reconstructed waterfalls can help the human operator to find undetected mines.

- 1) *Reliability*: If the assumption on both the model and the bound errors are correct, then the interval method will provide guaranteed results. This is not the case of other Bayesian smoothers. The Kalman-based methods do not provide any bound on the linearization errors and particle-based methods do not provide any quantification of the error due to the finite number of particles. Only the interval

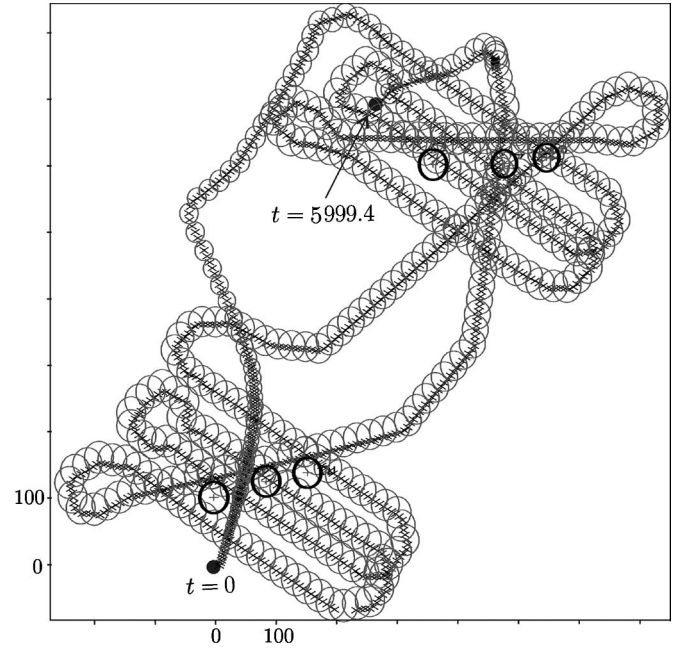


Fig. 9. Confidence ellipses generated by a classical extended Kalman smoother; the six thick ellipses correspond to the six mines.

propagation method is able to provide an envelope that encloses all feasible trajectories of the robot in a nonlinear context.

- 2) *Inconsistency*: When the model is not anymore valid (for instance, when the robot meets the bottom), when outliers occur in data, then the interval propagation method generally yields an empty set. An inconsistency often translates some anomalies during the mission of the robot, and should thus be identified in order to realize a reliable navigation system. Detecting inconsistencies can thus be useful in practice to detect conception bugs that affect the behavior of the robot.
- 3) *Validation*: On the one hand, if some sensors are not reliable or if the model is not precise enough, the interval propagation system usually detects an inconsistency. On the other hand, if the interval propagation provides a consistent and small envelope for the trajectory and the marks, then the robot equipped with its sensors can be considered as reliable (this is not the case for the Kalman filter that often provides a thin envelope even if some assumptions on the sensors are wrong). As a consequence, the interval propagation can be considered as a reliable way to validate the quality of a navigation system.
- 4) *Mark association*: Interval propagation can easily be combined with other propagation methods to deal with problems involving Boolean or integer variables. For instance, if, in the mark table (16), we remove all mark numbers in column 2, and if we only give the information that we have six marks, then the propagation is able to find all correspondences between marks in less than 3 m.
- 5) *Approximation*: Interval propagation methods never linearize the equations (contrary to the Kalman smoother);

they do not assume that the noise is additive; they do not approximate probability density functions (contrary to most Bayesian smoothers); they do not assume that the computer computes with real numbers instead of floating point numbers. Of course, they consider that the noise is bounded, reliable boundaries are known, the model is correct, and no outlier occurs, but these assumptions are already in the definition of the problem. Interval propagation does not bring any further approximation as it is the case for other nonlinear SLAM methods.

## VII. CONCLUSION

In this paper, we have shown that interval constraint propagation could be seen as an efficient alternative to Bayesian approaches for solving SLAM problems in a bounded-error context. Of course, as presented here, the SLAM method is offline and requires a human operator to detect seamounts and make the mark associations. A lot of work is left to obtain an online SLAM method. However, the approach has been demonstrated on a real problem with an actual underwater robot (the *Redermor*).

The main advantages of the interval propagation approach are its generality, its simplicity, its reliability, and its efficiency. For instance, if we have several robots communicating together, with different kind of sensors, the SLAM problem can be cast into a huge nonlinear set of constraints. Without any approximation (such as a linearization) or tuning (such as the number of particles), the interval propagation is able to provide an envelope that encloses the trajectories of the robots and build an interval map for the surrounding marks. Whereas classical Bayesian can be considered as more efficient to get the best trajectory and the best mark positions, an interval propagation is more adequate to find an envelope for them.

In practice, when strong outliers occur, an interval approach generally returns the empty set and we are not able to have an estimation of the location of the robot. If we really need such an estimation, a possible alternative is to maximize the number of constraints that can be satisfied [17]. This can be done efficiently using interval propagation [16]. The corresponding approach has been shown to be robust with respect to outliers.

In the near future, we are planning to merge several experiments made by one or different underwater robots. A promising work in this direction has been proposed by Di Marco *et al.* [4] in a set membership context. With the approach proposed here, merging several experiments can be done by concatenating the equalities and inequalities associated to each experiment. We suspect the interval propagation to be efficient for this type of problems.

## REFERENCES

- [1] A. Caiti, A. Garulli, F. Livide, and D. Prattichizzo, "Set-membership acoustic tracking of autonomous underwater vehicles," *Acta Acust. United Acust.*, vol. 88, no. 5, pp. 648–652, 2002.
- [2] J. A. Castellanos and J. D. Tardos, *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Norwell, MA: Kluwer, 1999.
- [3] N. Delanoue, L. Jaulin, and B. Cottenceau, "Using interval arithmetic to prove that a set is path-connected," *Theor. Comput. Sci.*, vol. 351, no. 1, pp. 119–128, 2006.
- [4] M. Di Marco, A. Garulli, A. Giannitrapani, and A. Vicino, "Simultaneous localization and map building for a team of cooperating robots: A set membership approach," *IEEE Trans. Robot. Autom.*, vol. 19, no. 2, pp. 238–249, Apr. 2003.
- [5] 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.
- [6] M. Di Marco, A. Garulli, S. Lacroix, and A. Vicino, "Set membership localization and mapping for autonomous navigation," *Int. J. Robust Nonlinear Control*, vol. 11, no. 7, pp. 709–734, 2001.
- [7] 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. 17, no. 3, pp. 229–241, Jun. 2001.
- [8] C. Drocourt, L. Delahoché, B. M. E. Brassart, and A. Clerentin, "Incremental construction of the robot's environmental map using interval analysis," in *Proc. 2nd Int. Workshop, COCOS 2003, Global Optim. Constraint Satisf.*, 2005, vol. 3478, pp. 127–141.
- [9] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard, "Visually navigating the rms titanic with slam information filters," in *Proc. Robot. Sci. Syst. (RSS)*, Cambridge, MA, Jun. 2005, pp. 57–64.
- [10] 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.
- [11] E. Halbwachs and D. Meizel, "Bounded-error estimation for mobile vehicle localization," in *Proc. CESA 1996 IMACS Multiconf. (Symp. Modelling, Anal. Simul.)*, pp. 1005–1010.
- [12] E. R. Hansen, *Global Optimization Using Interval Analysis*. New York: Marcel Dekker, 1992.
- [13] L. Jaulin, "Nonlinear bounded-error state estimation of continuous-time systems," *Automatica*, vol. 38, pp. 1079–1082, 2002.
- [14] L. Jaulin, ENSIETA-GESMA. (2007). Solver GESMI (Global Estimation of SeaMarks with Intervals) [Online]. Available: <http://www.ensieta.fr/jaulin/gesmi.zip>
- [15] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. London: Springer-Verlag, 2001.
- [16] L. Jaulin and E. Walter, "Guaranteed robust nonlinear minimax estimation," *IEEE Trans. Autom. Control*, vol. 47, no. 11, pp. 1857–1864, Nov. 2002.
- [17] H. Lahanier, E. Walter, and R. Gomeni, "OMNE: A new robust membership-set estimator for the parameters of nonlinear models," *J. Pharmacokinet. Biopharm.*, vol. 15, pp. 203–219, 1987.
- [18] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*. Boston, MA: Kluwer, 1992.
- [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, pp. 286–298, 1992.
- [20] F. Lydoire and P. Poignet, "Nonlinear predictive control using constraint satisfaction," in *Proc. 2nd Int. Workshop Global Constrained Optim. Constraint Satisf. (COCOS)*, 2003, pp. 179–188.
- [21] 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.
- [22] D. Meizel, A. Preciado-Ruiz, and E. Halbwachs, "Estimation of mobile robot localization: Geometric approaches," in *Bounding Approaches to System Identification*, M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, Eds. New York: Plenum, 1996, pp. 463–489.
- [23] D. K. Montemero, 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. (IJCAI)*, 2003, pp. 1151–1156.
- [24] R. E. Moore, *Methods and Applications of Interval Analysis*. Philadelphia, PA: SIAM, 1979.
- [25] A. Neumaier, *Interval Methods for Systems of Equations*. Cambridge, U.K.: Cambridge Univ. Press, 1990.
- [26] J. Porta, "Cuikslam: A kinematics-based approach to slam," in *Proc. 2005 IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 2436–2442.
- [27] T. Raissi, N. Ramdani, and Y. Candau, "Set membership state and parameter estimation for systems described by nonlinear differential equations," *Automatica*, vol. 40, pp. 1771–1777, 2004.

- [28] S. Reed, Y. Petillot, and J. Bell, "An automatic approach to the detection and extraction of mine features in sidescan sonar," *IEEE J. Ocean. Eng.*, vol. 28, no. 1, pp. 90–105, Jan. 2003.
- [29] I. Ruiz, S. de Raucourt, Y. Petillot, and D. Lane, "Concurrent mapping and localization using sidescan sonar," *IEEE J. Ocean. Eng.*, vol. 39, no. 2, pp. 442–456, Apr. 2004.
- [30] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.
- [31] S. Thrun and M. Montemerlo, "The GraphSLAM algorithm with applications to large-scale mapping of urban structures," *Int. J. Robot. Res.*, vol. 25, no. 5/6, pp. 403–430, 2005.
- [32] P. H. Vinas, M. A. Sainz, J. Vehi, and L. Jaulin, "Quantified set inversion algorithm with applications to control," *Reliable Comput.*, vol. 11, no. 5, pp. 369–382, 2006.
- [33] S. Williams, G. Dissanayake, and H. Durrant-Whyte, "Towards terrain-aided navigation for underwater robotics," *Adv. Robot.*, vol. 15, no. 5, pp. 533–549, 2001.

**Luc Jaulin** was born in Nevers, France, in 1967. He received the Ph.D. degree in automatic control from the University of Orsay, Paris, France in 1993.

Since 2004, he has been a Professor of robotics at the Ecole Nationale Supérieure des Ingenieurs des Etudes et Techniques d'Armements (ENSIETA), Développement Technologies Nouvelles, Brest, France. He is also with the Groupe d'Etude Sous-Marine de l'Atlantique (GESMA), Brest. He was engaged in research on underwater robotics using interval methods and constraint propagation.