Interval state estimation with data association

Benoît Desrochers[1], Luc Jaulin[2]
[1]DGA Tn, Brest, France
[2]Lab-STICC, ENSTA Bretagne, Brest, France

August 2, 2018

1 Introduction

Interval analysis [8][12] has been for than 30 years for control [14] or for state estimation [13][9]. A classical state estimation problem in a bounded-error context [19] can be formalized has follows

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t)\right), \mathbf{u}(t)) & \text{(evolution equation)} \\ \mathbf{g}(\mathbf{x}(t_i)) \in [\mathbf{y}](t_i) & \text{(observation constraint)} \\ \mathbf{x}(0) \in \mathbb{X}_0 & \text{(initial state)} \end{cases} \tag{1}$$

where $\mathbf{x}$ is the unknown state vector, $\mathbf{y}$ an output measurement vector, $\mathbf{u}$ an input measurement vector. It is known that for all t, $\mathbf{u}(t) \in [\mathbf{u}]$ and $\mathbf{y}(t) \in [\mathbf{y}]$.

Often, the observation constraint may also appear in the following implicit form

$$\mathbf{g}(\mathbf{x}(t_i), \mathbf{y}(t_i)) = \mathbf{0}, \mathbf{y}(t_i) \in [\mathbf{y}](t_i). \tag{2}$$

Example. Assume that we have a robot at position (x_1, x_2) with a heading x_3. It measures a landmark located at $(4, 5)$. The corresponding measurement vector is composed of the distance y_1 and the bearing y_2. In such a case, we have

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} x_1 + y_1 \cdot \cos\left(x_3 + y_2\right) - 4 \\ x_2 + y_1 \cdot \sin\left(x_3 + y_2\right) - 5 \end{pmatrix}$$

as illustrated by Figure 1. If now, the robot only measures the distance to the landmark, we get

$$g(\mathbf{x}, y) = \left(x_1 - 4\right)^2 + \left(x_2 - 5\right)^2 - y^2.$$

In the more general case, the observation function is uncertain or more precisely, it depends on some parameters that may be uncertain. The observation constraint can be written as

$$\mathbf{g}(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)) = \mathbf{0}, \mathbf{y}(t_i) \in [\mathbf{y}](t_i), \mathbf{m}(t_i) \in [\mathbf{m}](t_i). \tag{3}$$

In a localization context, the parameter vector $\mathbf{m}(t_i)$ corresponds to a landmark the position of which is known with some bounded uncertainties.

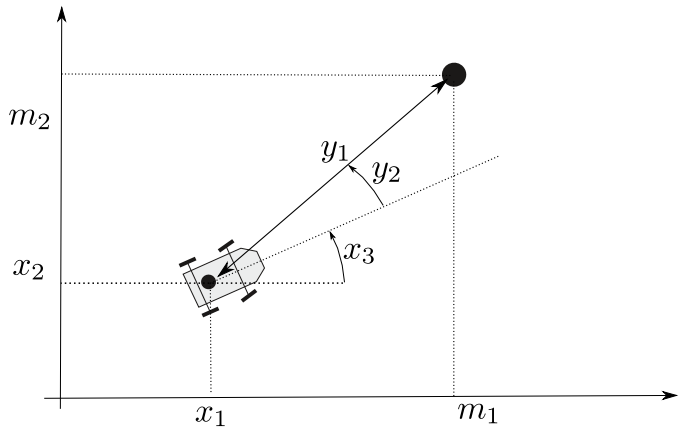

Figure 1: The landmark $\mathbf{m}$ is seen by the robot.

In some situations, the data are not associated. The observation constraint has the form

$$\begin{array}{c} \mathbf{g}(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)) = \mathbf{0} \\ \mathbf{m}(t_i) \in [\mathbf{m}_1] \vee \ldots \vee \mathbf{m}(t_i) \in [\mathbf{m}_\ell] \end{array} \tag{4}$$

or equivalently

$$\begin{array}{c} \mathbf{g}(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)) = \mathbf{0} \\ \mathbf{m}(t_i) \in \mathbb{M} = \{[\mathbf{m}_1], \ldots, [\mathbf{m}_\ell]\} \end{array}$$

It means that we do not know which is the right parameter vector associated to the observation constraint.

In this paper, we consider a state estimation problem in the case where we have to solve the data association problem [18]. This problem can be formalized by the following state equations:

$$\left\{ \begin{array}{rl} \dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t)\right), \mathbf{u}(t)) & \text{(evolution equation)} \\ \mathbf{g}(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)) = \mathbf{0} & \text{(observation constraint)} \\ \mathbf{y}(t_i) \in [\mathbf{y}](t_i), \mathbf{m}(t_i) \in \mathbb{M} & \\ \mathbf{x}(0) \in \mathbb{X}_0 & \text{(initial state)} \end{array} \right. \tag{5}$$

The set $\mathbb{M}$ is is assumed to be known. The membership constraint $\exists \mathbf{m}(t) \in \mathbb{M}$ highlights the requirement of solving the so-called *data association problem* which aims at finding which point of $\mathbb{M}$ is associated with the measurement vector $\mathbf{y}$. If $\mathbb{M}$ is composed with finite number of isolated points. Our problem copes with the initial localization problem on a field of point landmarks that are indistinguishable [6]. All measurements have the same aspect and cannot be associated directly with a particular point of the map. This problem frequently arises when acoustic sensors are used to detect underwater environmental features [3]. This is illustrated by Figure 2 where rocks can be used as marks.

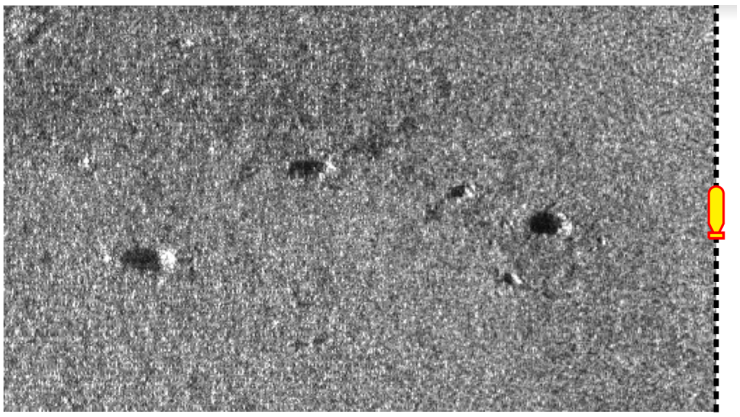

Figure 2: The robot (yellow) sees with its sonar some rocks at the bottom of the seafloor. Now, it is not able to make the difference between one rock and an other. This sonar image has been collected by the robot Daurade (DGA-TN, Shom, Brest) equipped with a Klein 5000.

In this paper, we propose an interval-based method [10] to solve the localization problem efficiently. [2] [7]. We will assume that all marks $\mathbf{m}$ are inside a set $\mathbb{M}$ which is made with small boxes as represented in Figure 5.

2 Problem

As an application, we consider a robot moving on a plane, the motion of which is described by the state equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t)\right), \mathbf{u}(t)) = \left(\begin{array}{c} \left(\begin{array}{cc} \cos(\psi(.)) & -\sin(\psi(.)) \\ \sin(\psi(.)) & \cos(\psi(.)) \end{array} \right) \cdot \mathbf{v}(.) \\ \omega(.) \end{array} \right). \tag{6}$$

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

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

$$\mathbf{g}\left(\mathbf{x}(t_i), \mathbf{y}(t_i), \mathbf{m}(t_i)\right) = \mathbf{0} \tag{7}$$

with

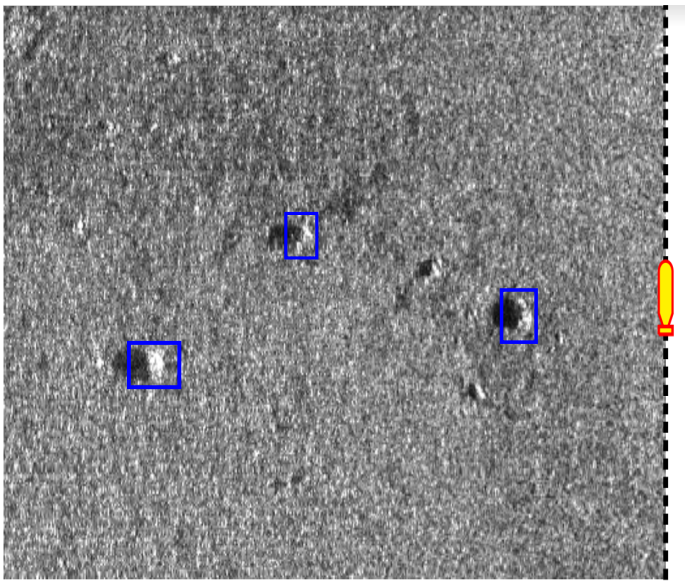

Figure 3: Each rock is assumed to belong to a small box which is georeferenced

$$\mathbf{g}(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)) = \begin{pmatrix} p_x(t_i) \\ p_y(t_i) \end{pmatrix} + \rho(t_i) \cdot \begin{pmatrix} \cos(\psi(t_i)+\varphi(t_i)) \\ \sin(\psi(t_i)+\varphi(t_i)) \end{pmatrix} - \begin{pmatrix} m_x(t_i) \\ m_y(t_i) \end{pmatrix}. \tag{8}$$

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

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t)\right), \mathbf{u}(t)) \\ \mathbf{g}(\mathbf{x}(t_i),\mathbf{y}(t_i),\mathbf{m}(t_i)) = \mathbf{0} \\ \mathbf{y}(t_i) \in [\mathbf{y}](t_i), \mathbf{m}(t_i) \in \mathbb{M} \\ \mathbf{x}(0) \in \mathbb{X}_0 \end{cases} \tag{9}$$

where the trajectory $\mathbf{x}(.)$, and the landmark associated to the measurements taken at time t_i both need to be estimated. This set of equations can decomposed into:

$$\begin{cases} (i) & \dot{\mathbf{x}}(.) = \mathbf{f}(\mathbf{x}(.), \mathbf{u}(.)) \\ (ii) & \mathbf{m}(t_i) \in \mathbb{M} \\ (iii) & \mathbf{a}_i = \mathbf{m}(t_i) - \mathbf{p}(t_i) \\ (iv) & \alpha_i = \psi(t_i) + \varphi(t_i) \\ (v) & \mathbf{a}_i = \rho \cdot \begin{pmatrix} \cos\alpha(t_i) \\ \sin\alpha(t_i) \end{pmatrix} \end{cases} \tag{10}$$

where a contractor [15] can be defined for each constraint. In particular, the contractor $\mathcal{C}_{\frac{d}{dt}}$ [16] is used to contract the tube $[\mathbf{x}](.)$ with respect to (i). The map $\mathbb{M}$ can be depicted by a subpaving, or by an image, for which a contractor

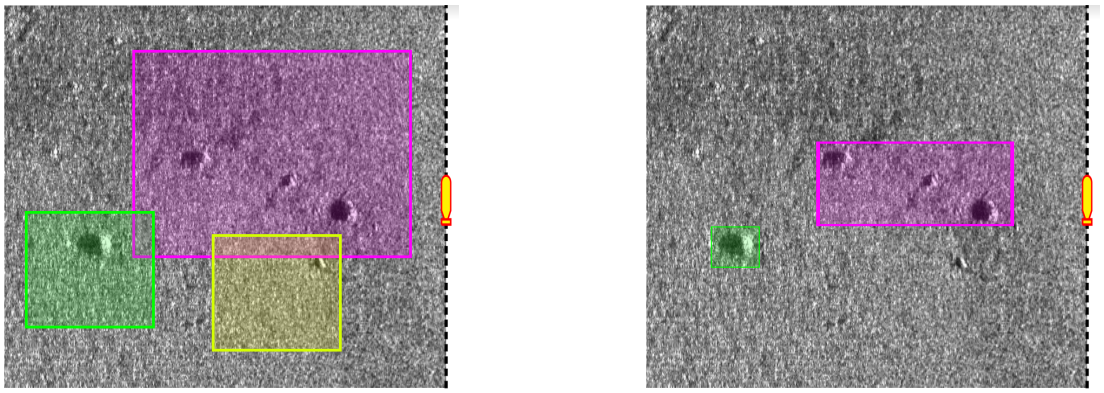

Figure 4: Map contractor. This operator does not remove any significant rock

such as the one used in [17] can be used. For any box $[\mathbf{x}] \in \mathbb{IR}^2$, this contractor returns the smallest box which contains all landmarks included in $[\mathbf{x}]$, as illustrated by Figure 4. This contractor which does not exist in the literature is given in the next section. The constraint (v) corresponds to the polar constraint where the minimal contractor [1] can be used.

3 Constellation contractor

This section proposes a new contractor, named the *constellation contractor*, which will allow us to solve the data association problem. Consider a constellation of ℓ points $\mathbb{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_\ell\}$ of $\mathbb{R}^d$, and a box $[\mathbf{x}]$,we want to compute the smallest box $\mathcal{C}([\mathbf{x}])$ which contains $\mathbb{M} \cap [\mathbf{x}]$, or equivalently

$$\mathcal{C}([\mathbf{x}]) = \bigsqcup_i [\mathbf{x}] \cap \{\mathbf{m}_i\}.$$

where $\sqcup$ returns the smallest enclosing the union of its arguments.

The operator $\mathcal{C}$ corresponds to the *constellation contractor*. Since we will have many different boxes $[\mathbf{x}]$ whereas the constellation is fixed, a preprocessing step has to be done. This step will allows us to have a logarithmic complexity for the contractor with respect to ℓ.

Preprocessing step. We build a balanced binary tree corresponding to a R-tree [4]. The R-tree has be following properties.

- *(i)* To each node β of the tree is associated a box $\square\beta$ containing at least one point of the constellation $\mathbb{M}$ and such that $\mathcal{C}(\square\beta) = \square\beta$. It means that $\square(\beta)$ is the enveloping box of all points of $\mathbb{M}$ it contains.
- *(ii)* If β is the root, $\mathbb{M} \subset \square\beta$.
- *(iii)* If β_1, β_2 are sons of β then $(\square\beta_1 \cap \mathbb{M}) \cup (\square\beta_2 \cap \mathbb{M}) = (\square\beta \cap \mathbb{M})$.

Algorithm 1 $\mathcal{C}([\mathbf{x}], \beta)$

	$\mathcal{C}([\mathbf{x}], \beta)$
1	If $\square\beta \subset [\mathbf{x}]$ return $\square\beta$
2	If $\square\beta \cap [\mathbf{x}] = \emptyset$ return $\emptyset$
4	Return $(\mathcal{C}([\mathbf{x}], \text{left\_son}(\beta))) \sqcup (\mathcal{C}([\mathbf{x}], \text{right\_son}(\beta)))$

- *(iv)* If β_1, β_2 are are brothers then $\square\beta_1 \cap \square\beta_2 = \emptyset$. When this property is satisfied, the R-tree is called a R* tree. This property is not restrictive in our context since we assume that the $\mathbf{m}_i$ are completely known beforehand.
- *(v)* The tree is balanced with respect to the space. This means that the bisection direction for the branching of the node β is decided with respect to the largest width of $\square(\beta)$.
- *(vi)* The tree is balanced with respect to the constellation. It means that bisection position of β is decided with respect to the median in order to minimize the difference $|\text{card}(\square\beta_1 \cap \mathbb{M}) - \text{card}(\square\beta_2 \cap \mathbb{M})|$.

These properties are illustrated by Figure 5 in the case where the constellation has $\ell = 10$ points (represented by the tiny blue boxes). The root of the tree is β_0 which has two sons β_1, β_2. Note that $|\text{card}(\square\beta_1 \cap \mathbb{M}) - \text{card}(\square\beta_2 \cap \mathbb{M})| = 1$ which corresponds to the minimum that can be obtained. The corresponding bisection is obtained from the median. The node β_2 has two sons β_3, β_4. The node β_5 is a leaf of the tree and $\square\beta_5$ corresponds to a single point of the constellation $\mathbb{M}$.

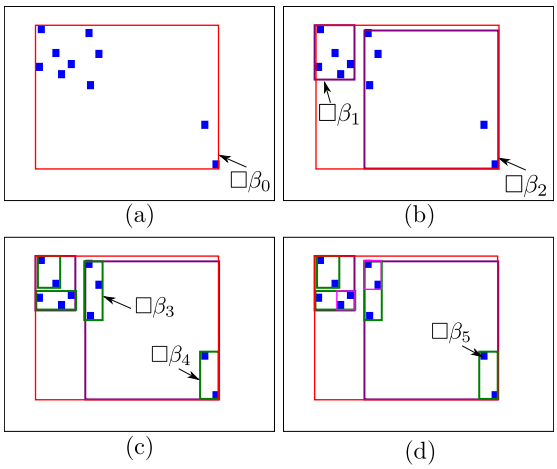

Figure 5: An R-tree is created first to get an efficient contractor

Once constructed, the R-tree allows a logarithmic complexity with respect to ℓ for the constellation contractor. The corresponding algorithm is as follows.

The algorithm is illustrated by Figure 6. First, since $\square\beta_0 \not\subset [\mathbf{x}]$, we explore the two sons of β_0. Since $\square\beta_1 \subset [\mathbf{x}]$ and $\square\beta_2 \not\subset [\mathbf{x}]$, we will only explore the branch corresponding to β_2. Again, since $\square\beta_3 \subset [\mathbf{x}]$ and $\square\beta_4 \not\subset [\mathbf{x}]$, we only explore β_4. Finally, the exploration of the tree corresponds to the following algebraic calculus:

$$\begin{aligned}
\mathcal{C}([\mathbf{x}], \beta) &= [\mathbb{M} \cap \square\beta_0 \cap [\mathbf{x}]] \\
&= \square\beta_1 \sqcup [\mathbb{M} \cap \square\beta_2 \cap [\mathbf{x}]] \\
&= \square\beta_1 \sqcup \square\beta_3 \sqcup [\mathbb{M} \cap \square\beta_4 \cap [\mathbf{x}]] \\
&= \square\beta_1 \sqcup \square\beta_3 \sqcup [\mathbb{M} \cap \square\beta_5 \cap [\mathbf{x}]] \\
&= \square\beta_1 \sqcup \square\beta_3 \sqcup \square\beta_5.
\end{aligned}$$

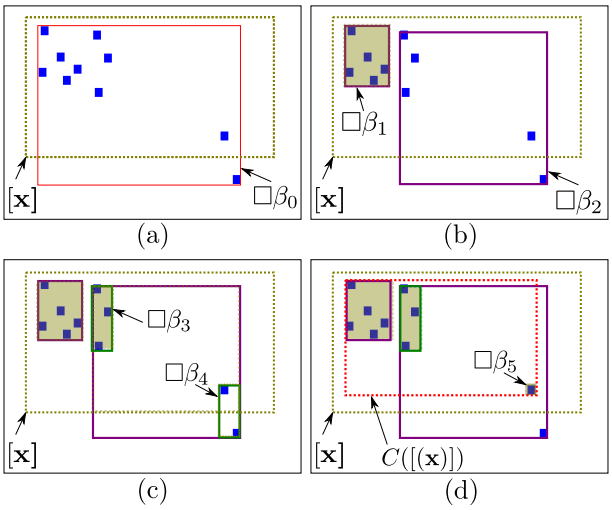

Figure 6: The resulting contractor has a logarithmic complexity

4 Test-case

Consider an AUV starting its mission with a huge position uncertainty. This can happen during a dive in deep water [11] or when, for discretion purpose, a long-range transit phase, underwater, is required to reach the working area.

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

Our robot performs a small mission pattern as depicted in Figure 7. It senses its environment using a forward-looking sonar oriented toward the seabed, the scope of which is represented by the blue pie. Every three seconds, it is able

to measure the distance and bearing between its pose to some landmarks which range between 10 and 70 meters. The positions of the detected landmarks are depicted by green dots. The 90 red segments represent the measurements. Note that only a small number of mapped landmarks have been detected.

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

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

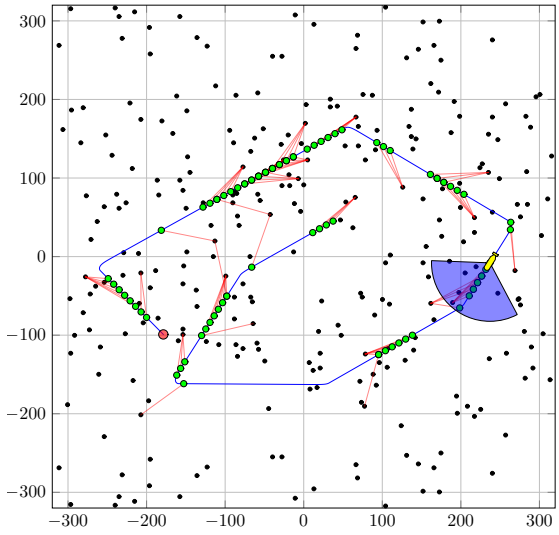

Figure 7: The simulated environment for initial localization. The trajectory of the AUV is depicted by the blue lines. Its starting point is drawn by the red dot. The map is composed of 230 landmarks represented by black dots.

This example aims at providing a practical illustration of how the constraint propagation methods can be used to:

1. find the trajectory the vehicle
2. solve the data association problem without any combinatorial explosion.

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

Table 1: Uncertainties on data used for the application.

Once these two issues are solved, any classical localization method, such as an EKF [5], can be used to get a more accurate estimate of the trajectory. The filter is initialized with a reliable starting point and only updated with observations which are correctly associated.

5 Results

Given $[\mathbf{x}(0)], \ldots, [\mathbf{x}(k_{max})]$ a set of boxes that enclose the state vector of the robot. All are initialized to $\mathbb{R}^2 \times [-\pi, \pi]$. The initial heading is assumed to be known with an accuracy of ± 10 degrees. Table 1 shows bounds used to quantify error on sensor readings. The map $\mathbb{M}$ is composed of 280 landmarks, and 90 observations have been done during the whole mission.

Figure 10 shows the final trajectory obtained after 12 iterations in less than 1 minute on *i7-5600U CPU@2.60GHz*.

In Table 2, for each iteration, the times needed to contract the whole trajectory are given. This time is constant at each iteration. During the constraint propagation process, the thinner the trajectory is, the smaller the number of landmarks contained in $[\mathbf{m}](t_i)$ is, and vice versa. As an indicator, columns 3 (resp. 4) of Table 2 shows the minimal (resp. maximal) number of landmarks included $[\mathbf{m}](t_i)$ among all measurements. The last column corresponds to the number of correct association, *i.e,* when $[\mathbf{m}](t_i)$ contains a single landmark. Figure 8 shows diameters of boxes along the trajectory for different iteration which illustrates the constraint propagation process. The constraint propagation methods is shown to be powerful in situations involving a huge number of possible data association. In comparison, existing method often meet difficulty when both the initial position and the data associations are unknown.

References

[1] B. Desrochers and L. Jaulin. A minimal contractor for the polar equation; application to robot localization. *Engineering Applications of Artificial Intelligence*, 55:83–92, 2016.

[2] B. Desrochers, S. Lacroix, and L. Jaulin. Set-membership approach to the kidnapped robot problem. In *IROS 2015*, 2015.

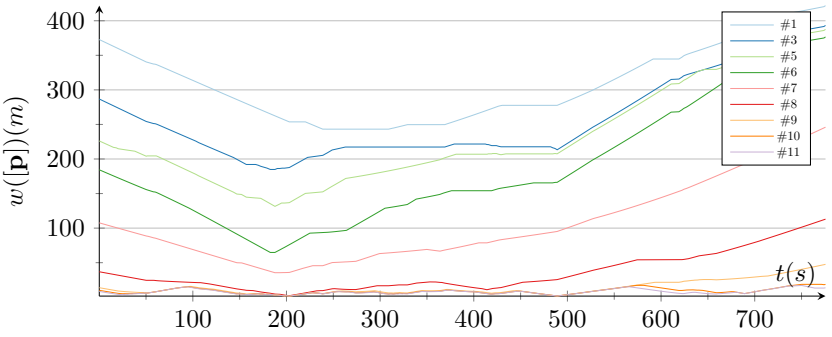

Figure 8: $w([\mathbf{p}])$ with respect to the iteration number.

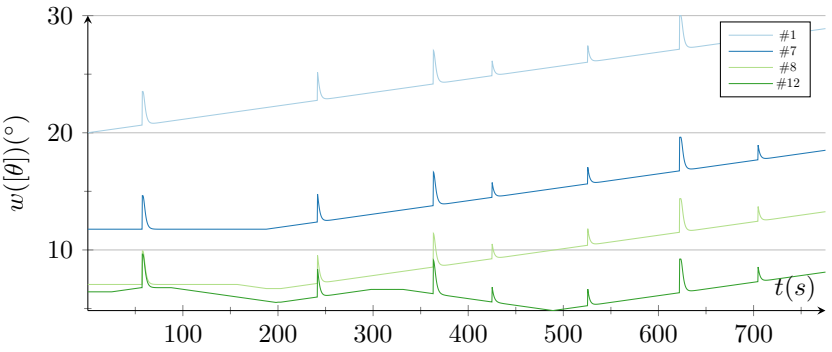

Figure 9: $w([\theta])$ with respect to the iteration number. The peaks in the graph are due to the wrapping effect when the vehicle turns.

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

Table 2: Time needed to contract the whole trajectory for each iteration. #min (resp. #max) is the smallest (resp. greatest) number of elements of $[\mathbf{m}](t_i)$ among all measurements. #ok denotes the number of good associations. The computing time is nearly constant.

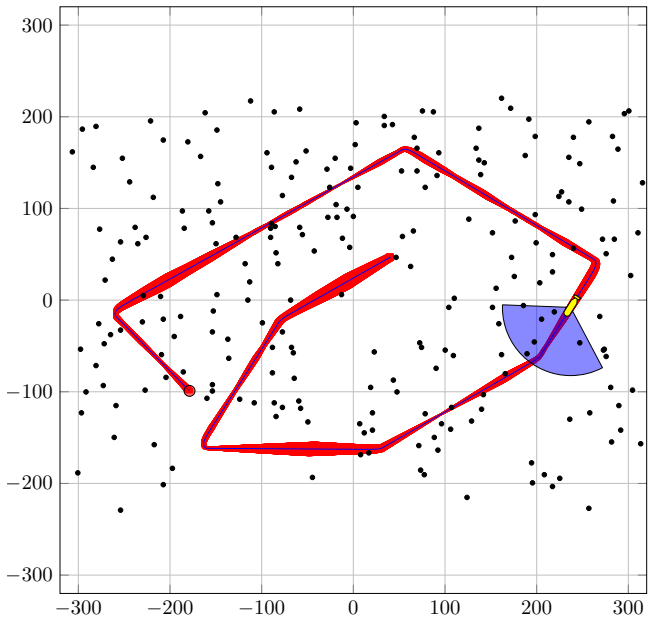

Figure 10: Final trajectory with correct estimation of the initial position and landmarks association. The true trajectory, in blue, belongs the tube in red.

[3] Maurice F. Fallon, John Folkesson, Hunter McClelland, and John J. Leonard. Relocating underwater features autonomously using sonar-based SLAM. *IEEE Journal of Oceanic Engineering*, 38(3):500–513, 2013.

[4] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *International conference on management data*, pages 47–57. ACM, 1984.

[5] L. Jaulin. *Mobile Robotics*. ISTE editions, 2015.

[6] L. Jaulin. Pure range-only slam with indistinguishable marks. *Constraints*, 21(4):557–576, 2016.

[7] L. Jaulin and B. Desrochers. Robust localisation using separators. In *COPROD 2014*, 2014.

[8] R. B. Kearfott and V. Kreinovich, editors. *Applications of Interval Computations*. Kluwer, Dordrecht, the Netherlands, 1996.

[9] V. Kreinovich, L. Longpré, P. Patangay, S. Ferson, and L. Ginzburg. Outlier detection under interval uncertainty: Algorithmic solvability and computational complexity. In I. Lirkov, S. Margenov, J. Wasniewski, and P. Yalamov, editors, *Large-Scale Scientific Computing*, Proceedings of the 4th International Conference LSSC'2003, 2003.

[10] Sergey I. Kumkov and Yuliya V. Mikushina. Interval approach to identification of catalytic process parameters. *Reliable Computing*, 19(2):197–214, 2013.

[11] Stephen McPhail. Autosub6000: A Deep Diving Long Range AUV. *Journal of Bionic Engineering*, 6(1):55–62, mar 2009.

[12] R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, PA, 1979.

[13] T. Raissi, N. Ramdani, and Y. Candau. Set membership state and parameter estimation for systems described by nonlinear differential equations. *Automatica*, 40:1771–1777, 2004.

[14] A. Rauh, E. Auer, T. Dotschel, and H. Aschemann. Verified stability analysis of continuous-time control systems with bounded parameter uncertainties and stochastic disturbances. *Computing*, 92(2):345–356, 2012.

[15] O. Reynet, L. Jaulin, and G. Chabert. Robust tdoa passive location using interval analysis and contractor programming. In *Radar*, Bordeaux, France, 2009.

[16] S. Rohou, L. Jaulin, M. Mihaylova, F. Le Bars, and S. Veres. Guaranteed Computation of Robots Trajectories. *Robotics and Autonomous Systems*, 93:76–84, 2017.

[17] J. Sliwka, F. Le Bars, O. Reynet, and L. Jaulin. Using interval methods in the context of robust localization of underwater robots. In *NAFIPS 2011*, El Paso, USA, 2011.

[18] S. Thrun, W. Bugard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, M.A., 2005.

[19] E. Walter and H. Piet-Lahanier. Estimation of the parameter uncertainty resulting from bounded-error data. *Math. Biosci.*, 92:55–74, 1988.