Estimating the trajectory of low-cost autonomous robots using interval analysis : application to the euRathlon competition

Fabrice Le Bars, Elba Antonio, Jorge Cervantes, Carlos De La Cruz and Luc Jaulin

Abstract In this paper, we describe a method based on interval arithmetic and contractors to compute an envelope containing the trajectory of a robot from usual proprioceptive and exteroceptive data, using a simple state equation model. To illustrate the applicability of the method, data from the euRathlon 2015, a multi-domain robotics competition, will be processed to build an estimation of the trajectory of a low-cost AUV (Autonomous Underwater Vehicle), navigating with the help of acoustic communication and ranging with an ASV (Autonomous Surface Vehicle).

Fabrice Le Bars
ENSTA Bretagne (Ecole Nationale Supérieure de Techniques Avancées Bretagne), Lab-STICC/CID/PRASYS, 2 rue François Verny, 29806 Brest Cédex 09, France, e-mail: fabrice.le_bars@ensta-bretagne.fr

Elba Antonio
CINVESTAV (Center for Research and Advanced Studies of the National Polytechnic Institute), UMI LAFMIA CNRS 3175, 2508 D.F. México City, México, e-mail: eantonio@ctrl.cinvestav.mx

Jorge Cervantes
CINVESTAV (Center for Research and Advanced Studies of the National Polytechnic Institute), UMI LAFMIA CNRS 3175, 2508 D.F. México City, México, e-mail: jcervantes@ctrl.cinvestav.mx

Carlos De La Cruz
CINVESTAV (Center for Research and Advanced Studies of the National Polytechnic Institute), UMI LAFMIA CNRS 3175, 2508 D.F. México City, México, e-mail: cdelacruz@ctrl.cinvestav.mx

Luc Jaulin
ENSTA Bretagne (Ecole Nationale Supérieure de Techniques Avancées Bretagne), Lab-STICC/CID/PRASYS, 2 rue François Verny, 29806 Brest Cédex 09, France, e-mail: lucjaulin@gmail.com

1 Introduction

In a submarine context, estimating the real trajectory of a robot after its mission is a challenging problem (see e.g. [PSSL14] for a review of possible techniques), especially for small low-cost AUVs (Autonomous Underwater Vehicles). Indeed, the GPS is not available underwater and in a small low-cost AUV, some important sensors for localization might not be available or with reduced functionality and accuracy, due to size, costs and energy restrictions:

DVL (Doppler Velocity Log): this sensor can provide the speed of the vehicle w.r.t. the sea floor or surface, which is a very important data for the localization in dead-reckoning if it is accurate enough, see e.g. [HPYZ08], [VM09].

Sonar (SOund Navigation And Ranging): depending on its type and quality, this device can help to the vehicle localization if marks on the sea floor can be detected (see e.g. [Jau09a], [LBSJ10] and [RRNT06] for a context of SLAM), if we are in a known environment (see e.g. [Jau09b], [RNRT06] and [MPM$^+$09]) or using a bathymetry map (see e.g. [FW08]).

USBL (Ultra-Short BaseLine): acoustic ranging and bearing from one or several devices of known positions is a typical way to follow accurately a submarine without error accumulation, see e.g. [RPW06].

INS (Inertial Navigation System)/AHRS (Attitude and Heading Reference System): accurate intertial and magnetic sensors to get the angles of the robot (especially the heading) are necessary to limit the error accumulation on the position estimation in a dead-reckoning context.

Additionally, few ready-to-use software suites able to fuse all those different data to build an estimation of the trajectory of a robot are available. Moreover, since most of the existing localization methods are based on probablities (see e.g. iXBlue DELPH in [GA09] for a Kalman-based fusion between INS, DVL, LBL, USBL, etc.), it is sometimes hard to evaluate the estimation error or detect inconsistencies. Taking into account partial information such as a position deduced by an operator that saw the robot at some time in a know place can be specially difficult to formalize and fuse with the other localization information.

For our underwater robot localization application, we propose a method based on interval arithmetic. Interval analysis has been proven to be efficient for several similar problems in robotics and other related applications:

- Localization of a robot with fugitive data that appear at some given time, see [LSRJ12].
- Mosaicking, see [LJT16].
- SLAM with indistinguishable marks, see [Jau16].
- Getting a guaranteed approximation of the explored area, see [DJ16a].
- Detecting loops in a trajectory, see [ADJ13].
- Improvements in GPS positioning, see [DB16].
- Robot calibration, see [DTDC14].

In our context, we want to estimate the trajectory of a 2D robot knowing its state equations, inputs and heading at all times, with some punctual position measurements (e.g. GPS when surfacing at a waypoint) as well as range measurements with another moving robot of known position (e.g. using acoustic modems with range measurement capability). Cooperation between robots has already demonstrated its efficiency in several situations, compared to the use of a single, usually bigger and more expensive robot: see e.g. [RC10], [APCV11], [GMG+12] or [LC14] for cooperation between 2, 3 or more AUVs, see [CGLP05] for the use of several acoustic buoys to localize underwater robots using a set-membership approach and see e.g. [MM08] and [FPL10] for cooperation between an ASV and an AUV.

In this paper, we will first describe the interval-based data fusion method used, and then show its application to the trajectory estimation of the *SARDINE* AUV during the euRathlon 2015 competition (see [FFD+16] for more information about the euRathlon competition).

2 Set-membership data fusion method for localization

2.1 Interval arithmetic

Interval arithmetic (see e.g. [Moo79], [KK96], [JKDW01] and [JNR15]) is a numerical tool which makes it possible to solve non-linear equations and non-convex minimization problems, in a global and guaranteed way. Now, estimating the state of a robot can be seen as solving simultaneously a set of non-linear equations, for which interval analysis could be useful. This section recalls the basic notions of interval arithmetic.

An *interval* is a closed and connected subset of $\mathbb{R}$. $\mathbb{IR}$ is the set of the intervals of $\mathbb{R}$. Consider two intervals $[x]$ and $[y]$ and an operator $\diamond \in \{+,-,.,/\}$, we define $[x]\diamond[y]$ as the smallest interval which contains all feasible values for $x \diamond y$, if $x \in [x]$ and $y \in [y]$. For instance

$$\begin{array}{r} [-3,4]+[2,5]=[-1,9], \\ [-1,4].[2,5]=[-5,20], \\ [-1,3]/[2,5]=[-\frac{1}{2},\frac{3}{2}]. \end{array} \quad (1)$$

If f is an elementary function such as sin,cos,... we define $f([x])$ as the smallest interval which contains all feasible values for $f(x)$, if $x \in [x]$. A *box* or *interval vector* $[\mathbf{x}]$ is a vector whose components are intervals:

$$[\mathbf{x}] = [x_1^-, x_1^+] \times ... \times [x_n^-, x_n^+] = [x_1] \times ... \times [x_n]. \quad (2)$$

x^- is the interval *lower bound* and x^+ its *upper bound*. The *midpoint* (or *center*) of a bounded and non-empty interval is $\mathrm{mid}([x]) = \frac{x^- + x^+}{2}$. The *width* of a non-empty interval is defined by $w([x]) = x^+ - x^-$. By convention, $w(\emptyset) = -\infty$. If $w([x]) = 0$, $[x]$ is *degenerated*. In this case, $[x]$ is a real singleton and will be noted $\{x\}$.

For a box, the width is $w([x]) = \max\{w([x_1]),...,w([x_n])\}$. We define also the volume of a non-empty box as

$$vol([x]) = \prod_{1 \le i \le n} w([x_i]).$$

Most of the time, if $[x]$ is an interval containing the x position of a robot, $\mathrm{mid}([x])$ will be taken as an estimation of x, and $w([x])/2$ will be taken as the estimation error. The volume of the box $[\mathbf{x}]$ representing the 2D or 3D position estimation of the robot could be also used to evaluate the estimation precision.

2.2 Contraction and propagation

ICP (Interval Constraint Propagation) is a technique (or sort of algorithm) that enables to solve efficiently (computing time is in general lower than other methods such as those involving bisections) equations on intervals. It is a combination between 2 more general notions: constraint propagation and interval arithmetic (see e.g. [JW02]).

Consider a constraint $\mathscr{C}$ (*i.e.*, an equation or an inequality), some variables $x_1, x_2, \ldots$ involved in $\mathscr{C}$ and prior interval domains $[x_i]$ that contain all feasible values for the x_i's. Interval arithmetic makes possible to contract the domains $[x_i]$ without removing any feasible values for the x_i. For instance, consider the equation $x_3 = x_1 + x_2$ where the domains for x_1, x_2, x_3 are given by $[x_1] = [-\infty, 5], [x_2] = [-\infty, 4]$ and $[x_3] = [6, \infty]$. These domains can be contracted to $[x_1'] = [2,5], [x_2'] = [1,4]$ and $[x_3'] = [6,9]$. The resulting interval calculus is as follows:

$$\begin{array}{rl} x_3 = x_1 + x_2 \Rightarrow x_3 \in & [6,\infty] \cap ([-\infty,5] + [-\infty,4]) \\ & = [6,\infty] \cap [-\infty,9] = [6,9]. \\ x_1 = x_3 - x_2 \Rightarrow x_1 \in & [-\infty,5] \cap ([6,\infty] - [-\infty,4]) \\ & = [-\infty,5] \cap [2,\infty] = [2,5]. \\ x_2 = x_3 - x_1 \Rightarrow x_2 \in & [-\infty,4] \cap ([6,\infty] - [-\infty,5]) \\ & = [-\infty,4] \cap [1,\infty] = [1,4]. \end{array} \tag{3}$$

A graph of constraints can be used as a simple representation to show which variable will be used to contract the others. For example, if $z^2 = \exp(x) + y$ and $x \in [1,4], y \in [3.1,3.2], z \in [4,8]$, then $x = \ln\left(z^2 - y\right) \Rightarrow x \in [x] \cap \ln\left(z^2 - y\right) \in [2.5,4]$ and $z = \sqrt{\exp(x) + y} \Rightarrow z \in [z] \cap \sqrt{\exp(x) + y} \in [4, 7.6]$. Therefore, the graph for the constraint $z^2 = \exp(x) + y$ would be like in Figure 1.

This contraction procedure can be performed with much more complex constraints. A contraction operator is called a *contractor* (see [CJ09]). When more than one constraint are involved, the contractions are performed sequentially several times, until no more significant contractions can be observed. It can be shown that the box to which the method converges does not depend on the order to which the contractors are applied (see [JKDW01]), but the computing time is highly sensitive

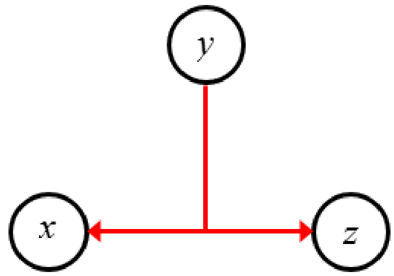

Fig. 1 Graph of constraints.

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 in 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*). Tools such as the IBEX library can be used to manipulate contractors, see [Cha13].

2.3 Context, assumptions and principle of the method used

In this paper, we will only consider a 2D localization problem, where the vertical localization problem is considered as solved or not needed:

- Most of the submarines have accurate depth and/or altimeter sensors compared to x,y localization sensors.
- Due to the submarine depth control at low depth and the fact that the acoustic modem on the surface vehicle is 1 m below the surface, the submarine acoustic modem can be considered in the same plane as the surface robot acoustic modem.
- The robot's trajectory is most of the time in the same plane, the depth changes are typically almost punctual: submerging in the beginning to reach target mission depth, surfacing at the end or to get GPS data.

We will also assume that there are no outliers in the data: they were manually removed, e.g. obvious temporary bad GPS fixes at some surfacing points (GPS fixes will be considered as punctual: for each fix, only 1 position at a specific time has been taken into account), and bad range measurements or data communications were automatically rejected. If outliers cannot be removed manually, a method using relaxed intersection is proposed in [Jau09b].

The method used in this paper is inspired from [Jau09a] and [LBSJ10]. In our set-membership context, we can describe our problem as follows

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x},\mathbf{u}) \quad \text{(evolution equation)} \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}) \quad \text{(observation equation)} \\ r_i &= r(\mathbf{x},\mathbf{m}) \quad \text{(ranging equation)} \end{cases} \tag{4}$$

where $\mathbf{x}(t)$ is the state vector of the submarine, $\mathbf{u}(t)$ is its input vector, $\mathbf{y}(t)$ is its output vector, $\mathbf{m}$ is the location of the surface robot and $\mathbf{r}_i$ is the i^{th} range measurement between the 2 robots. We consider that

- At some time t, we have a box $[\mathbf{x}]\,(t)$ containing the state vector:

$$\mathbf{x}(t) \in [\mathbf{x}]\,(t) \tag{5}$$

- For all $t \in [t_0, t_f]$, we have boxes enclosing the $\mathbf{u}(t)$ and $\mathbf{y}(t)$:

$$\forall t \in [t_0, t_f], \mathbf{u}(t) \in [\mathbf{u}]\,(t) \text{ and } \mathbf{y}(t) \in [\mathbf{y}]\,(t) \tag{6}$$

- We have a finite subset $\mathscr{M} \subset [t_0, t_f]$ such that

$$r_i(t) \in [r_i]\,(t). \tag{7}$$

  If $t \in \mathscr{M}$, the ranging measurement i has been made at time t and we have a box enclosing $\mathbf{m}(t)$.

For simplicity, we assume that the submarine motion can be described by the following evolution equation (inspired from [Jau09b]):

$$\begin{cases} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= u_2 - u_1 \\ \dot{v} &= (u_1 + u_2) - v \end{cases} \tag{8}$$

where (x, y) are the 2D coordinates of the robot, θ is its heading, v its speed, u_1 the input for the right thruster, u_2 the input for the left thruster (see Figure 2).

This equation is similar to the classical tank/wheelchair model (see e.g. [Jau15]), with very simple adaptations to simulate the fluid friction effects. We have chosen here a normalized model with the friction coefficient (in front of v) equal to 1, but in practice, the coefficient is different and needs to be estimated (e.g. empirically).

The observation equation is

$$\mathbf{y} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \tag{9}$$

because (x, y) can be measured sometimes by the GPS when the robot is near the surface and θ is measured by the AHRS at all times.

The ranging equation for the i^{th} communication and range measurement is

$$r_i = \sqrt{(x - m_x)^2 + (y - m_y)^2} \tag{10}$$

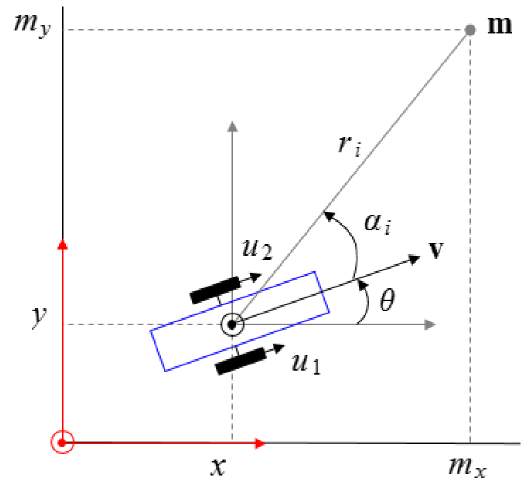

Fig. 2 Description of the submarine variables.

where r_i is the distance between the submarine and the surface robot (measured by the internal ranging function of the acoustic modems) and $\mathbf{m} = (m_x, m_y)$ is the position of the surface robot (measured by its GPS and sent to the submarine through the acoustic modems). It is worth noting that to be accurate in practice, due to the communication and ranging time, we should consider the fact that the position of the robots might have changed between the moment when the range measurement is made, the moment when the position of the surface vehicle is measured, and the moment when it is received by the submarine. We could also have clock synchronization errors between robots (see e.g. [BJ14] for methods to handle this). In our case, those uncertainties will be included inside the uncertainty for $\mathbf{m}$ and r_i.

As a consequence, our problem can be written as follows:

$$\begin{cases} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= u_2 - u_1 \\ \dot{v} &= (u_1 + u_2) - v \\ r_i &= \sqrt{(x-m_x)^2 + (y-m_y)^2} \end{cases} . \tag{11}$$

To solve this problem, we will use the ICP techniques described previously:

- First we need to use the observation equation to contract the known GPS positions of the robot, see Figure 3.
- Then, a forward-backward propagation w.r.t. time on the differential equations discretized using e.g. Euler method can be used to get a first evaluation of the trajectory, see Figure 4.
- Contracting the ranging equation should improve the position estimation of the robot at the specific ranging times, see Figure 5.

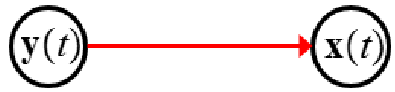

Fig. 3 Graph of constraints for the observation equation.

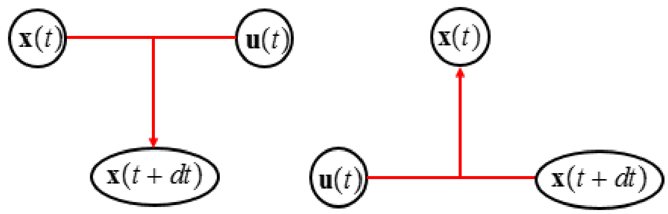

Fig. 4 Graph of constraints for the evolution equation (left : forward with time, right : backward with time).

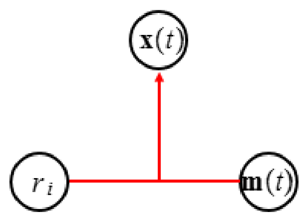

Fig. 5 Graph of constraints for the ranging equation.

- Repeating all the previous contraction operations until no more significant improvement on the trajectory estimation should propagate the punctual contractions on the whole trajectory.

Note that if needed, we could take into account more general exteroceptive data such as bearing measurements, see e.g. [DJ16b] or [IJG14] for related work.

3 Application

3.1 Description of the experiment

The euRathlon 2015 competition (see [FFD$^+$16]) proposed a situation where marine and submarine robots had to collaborate with ground and aerial robots to analyze the state of an area after a possible earthquake and determine which pipeline from a mock nuclear plant should be stopped to stem a leak while letting other pipelines cool the reactor. The submarine data presented in this paper corresponds to the AUV

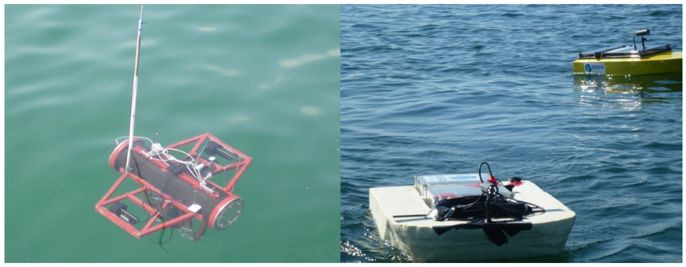

Fig. 6 AUV and ASV.

navigating through the pipelines structures area, and searching for the right one (2 groups of pipelines separated by 50 m and at a depth of around 5 m were to be inspected, with colored objects indicating their state). Meanwhile, aerial and ground robots were inspecting the damages around and inside the mock nuclear plant.

We must emphasize that these data were taken during a competition, therefore they have not been specifically done to illustrate the method described here. This implies that they are not perfect, for example: compasses of the robots were not calibrated correctly, the competition area was probably not deep enough to get a good acoustic communication, the acoustic modem on the ASV only started in the middle of the experiment, etc. This is to show that the approach can be efficient on real-world data.

The *SARDINE* AUV was equipped with a pressure sensor to get his depth, a mid-cost AHRS to get its heading, a GPS to get its position near the surface, a rotating sonar (not used for this part of the experiment), 2 cameras (not used for this part of the experiment) and a low-cost acoustic modem with ranging capability. In particular, it did not have a DVL or other sensor to evaluate its speed. It is controlled in the plane using 2 horizontal thrusters, and a central vertical thruster controls his depth (it is statically equilibrated to have an almost neutral buoyancy without roll or pitch), see Figure 6. The ASV was controlled using 2 thrusters and was equipped with a GPS, a low-cost AHRS, a camera (not used for this part of the experiment) and the same acoustic modem as the AUV, installed at 1 m below the hull to avoid immediate reflexion of acoustic waves on it.

During the experiment, the robots had their own embedded position estimation algorithms. The principle of the algorithm to estimate the position used inside the submarine was similar to what is described in this paper, but with the following differences:

- No backward propagation w.r.t. time.
- Different coefficients were used in the state equations (few time was spent to adjust them during the experiment).
- In case of inconsistencies between data, specific non-optimal fallback choices were made.

It is also important to note that this estimation was not always used by the embedded control part, see Figure 7:

- In the beginning yes.
- However for the ”spirals”, no, an open loop was used.

Here is a description of the trajectory (in Figure 7) evaluated by the embedded algorithm and retrieved just after the experiment, without further processing (we will refer to this estimation as the ”realtime estimation” in what follows):

- The ASV was always staying around the same waypoint ”HV2”.
- The AUV submerged at ”Surface begin” point and started to follow the waypoints, surfacing at each one (”WP1”, ”WP2”, ”WP3”, ”WP4”). During this phase, we clearly see the ”jumps” in the trajectory estimation due to the GPS when surfacing: between 2 waypoints, the trajectory estimation appears to be perfectly in the direction of the target waypoints, however this estimation is corrected when the robot surfaces and gets the GPS. When at the surface, we also see additional ”jumps” due to the fact that the GPS provides very noisy data when it is at the limit to not be able to see the satellites.
- After ”WP4”, it started to make a spiral around, using relative control commands (i.e. without trying to use its absolute position estimation). The trajectory estimation has some ”jumps” at this point also due to the corrections made with the range measurements.
- Then, it moved to the East and made another spiral.
- Finally it surfaced at the end of the mission. The total distance covered by the robot was approximately 500 m.

3.2 Results

A program available on

`www.ensta-bretagne.fr/lebars/sardine2015/`

has been made to make the offline processing of the data, see Figure 8.

All the state variables are first initialized with intervals centered on the realtime estimation and with an infinite width, while the inputs are loaded as degenerated intervals. After loading all the data as intervals, the first operation that should be done is to contract the data w.r.t. the measured punctual GPS positions (using the observation equation). To propagate the effect of those punctual known positions, we need then to use the evolution equation of the submarine to contract forward in time (see Figure 9), and then backward (see Figure 10).

For the forward contraction, the x and y errors (see Figure 11) increase linearly over the time. Between e.g. the second and the third waypoint, we clearly see 3 different inclinations of the x and y error curves:

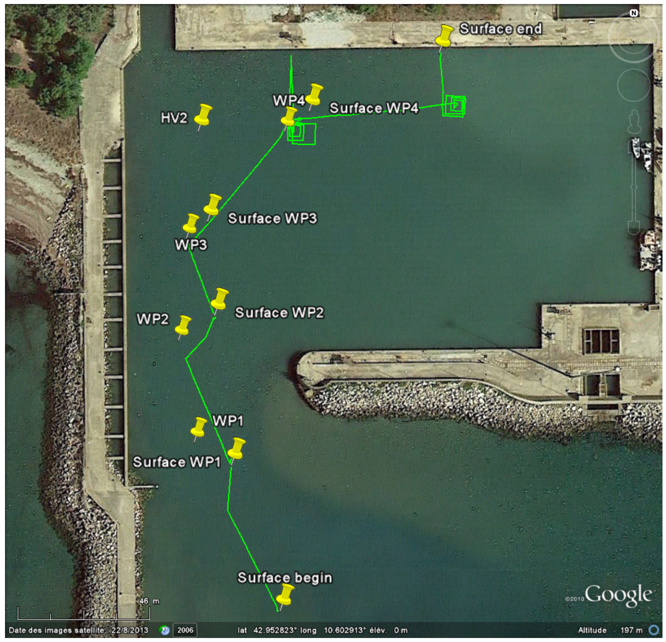

Fig. 7 Realtime trajectory estimation retrieved from the robot just after the experiment. Note that we do not have the ground truth for this experiment and that in the beginning, the robot is following waypoints ("WP1"..."WP4"), while in the end, it is making 2 spirals without following specific waypoints.

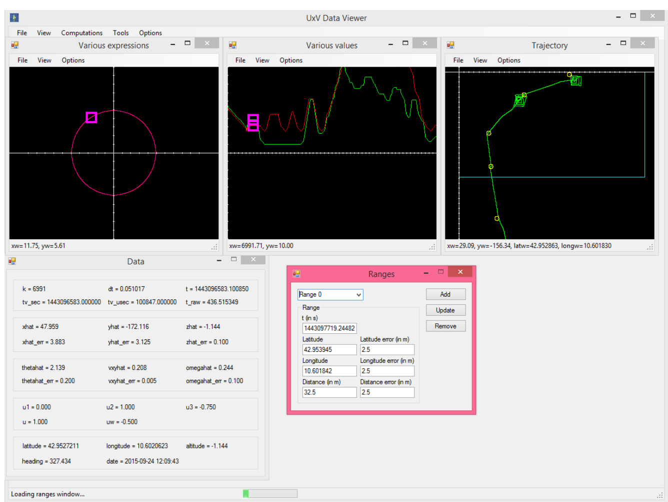

Fig. 8 UxV Data Viewer, a software made to post-process the data of robots using interval methods.

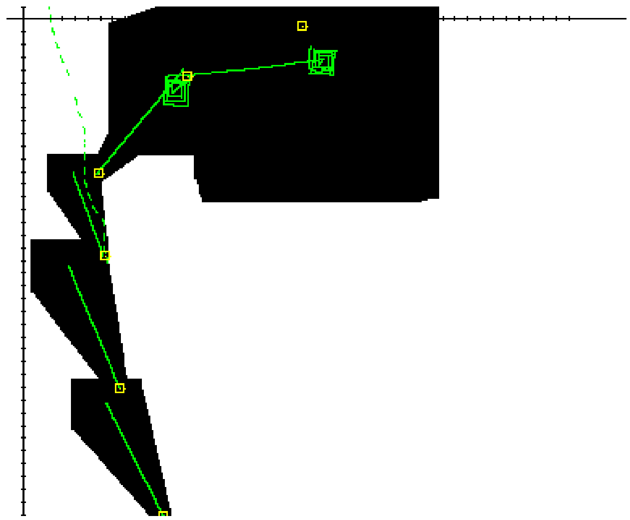

Fig. 9 Trajectory envelope (black squares) and center (green dots) after the GPS position contraction (the yellow squares represent the GPS fixes) and forward contraction.

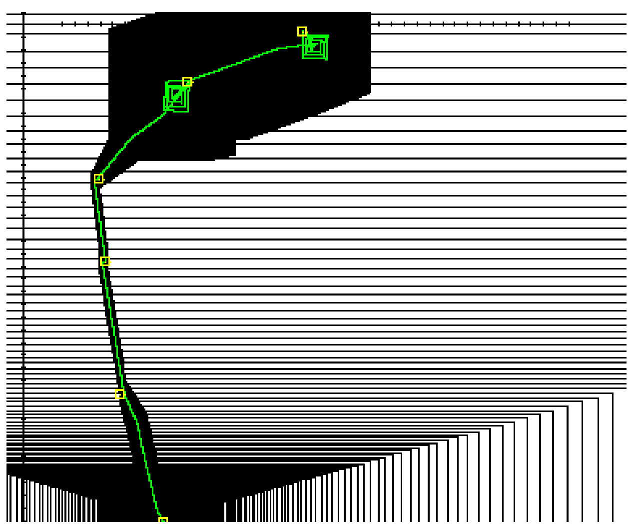

Fig. 10 Trajectory envelope (black squares) and center (green dots) after the GPS position contraction (the yellow squares represent the GPS fixes), forward contraction and backward contraction.

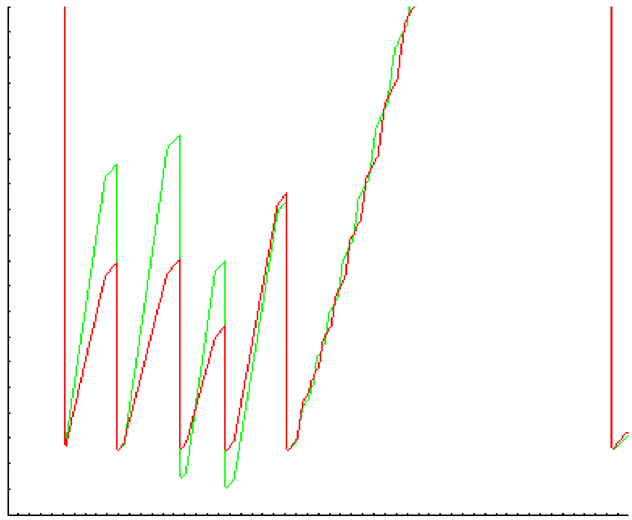

Fig. 11 Trajectory error w.r.t. time (x: green, y: red) after the GPS position contraction and forward contraction. Axis are $[0,3600]\text{s}\times[0,20]\text{m}$. The graphic is cut on the top to be able to see in the same time the general trends and the specific parts discussed in this section.

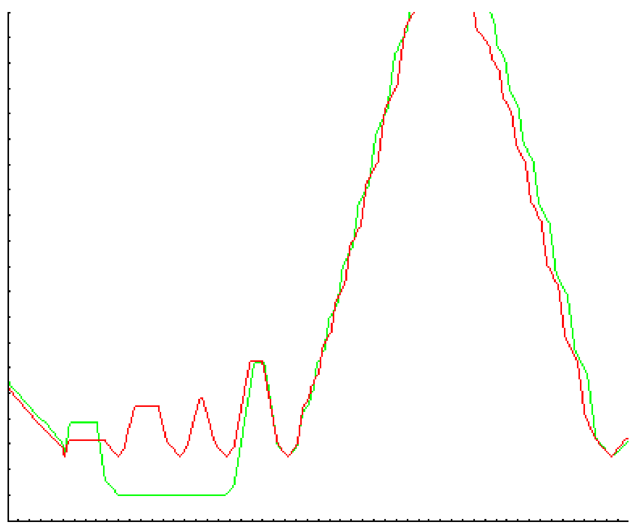

Fig. 12 Trajectory error w.r.t. time (x: green, y: red) after the GPS position contraction, forward contraction and backward contraction. Axis are $[0,3600]\text{s}\times[0,20]\text{m}$. The graphic is cut on the top to be able to see in the same time the general trends and the specific parts discussed in this section.

- The first is when the robot is stopped at the surface or going down after the GPS fix, but is not trying to move in the horizontal plane.
- The second when the robot is moving from one waypoint to the next.
- The third is similar to the first: the robot is surfacing but is not trying to move in the horizontal plane.

If we were only doing the backward contraction (i.e. without forward contraction before), the errors would decrease linearly.

After the forward-backward contractions, we notice a step in the middle of the y error graphics, between e.g. the first and the second waypoint (see Figure 12). This is due to the fact that the robot seems to have a bias (bad magnetic calibration, currents, non symmetric parts on the robot could explain this) and the GPS position measured at the surface was a surprise w.r.t. the expected surfacing position. This is consistent with the small width of the boxes (i.e. small width of $[x]$) at this part of the trajectory: the GPS is about to be inconsistent with the state equations on the x coordinate, but it is still consistent enough to not generate an empty set. To be more clear, we can take a simple example where a car travels between 2 cities in a straight line (1 dimension problem) at a limited speed in a known time. If:

- The speed limit is 110 km/h (the car can be slower, but not faster),
- The time of travel was 3h10,
- The distance between the cities is 330 km,

We can estimate the position of the car at all times (except at the very beginning and the very end where we know in which city we are) with the same error, which corresponds to the distance covered in 10 min at 110 km/h, i.e. 18.33 km.

If we continue the analysis of Figure 12, there is another case where there is almost no step (triangular curve shape): between the second and the third waypoint. In this case, the robot was always in the middle of the box, there was no surprise at the end. In Figure 10, we can also mention that there are very big boxes in the beginning before the first GPS fix, and the dotted green part of the trajectory visible in Figure 9 corresponds also to the center of some of them before the backward contraction.

If we now use the ranging equation to contract the trajectory, and then make again a forward and backward contraction to propagate the effect of the punctual range measurements, we get the Figures 13 and 14. Since the range measurements were made mainly when the submarine was far on the right of the surface robot, only the x coordinate estimation is significantly improved.

The final post-processed trajectory estimation is compared with the realtime estimation in Figure 15. We clearly see that for the final ”spiral”, the robot was not aware that it drifted closer to the North wall than expected. Indeed, the operators following the robot during the competition (a small buoy at the surface was attached to the submarine as a safety mark) confirmed that the robot was very likely to be close to the wall.

It is important to note that in the context of such a submarine competition, it is very difficult (for example necessary sensors would be expensive, would perturbate other sensors or would not fit on all the robots), to get a ground truth. Most

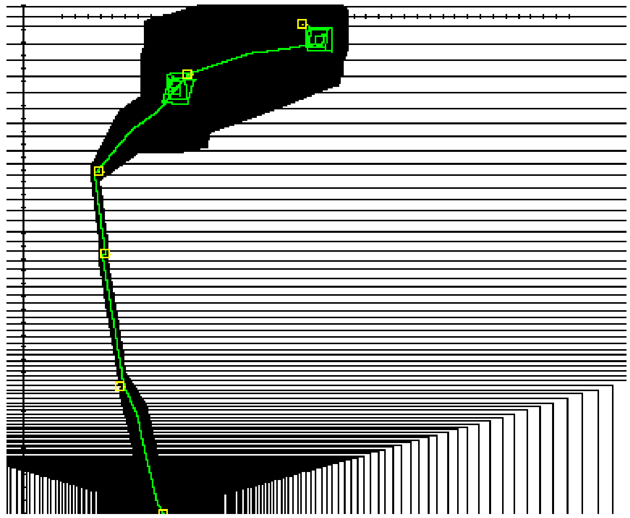

Fig. 13 Final trajectory envelope (black squares) and center (green dots), with GPS fixes (yellow squares).

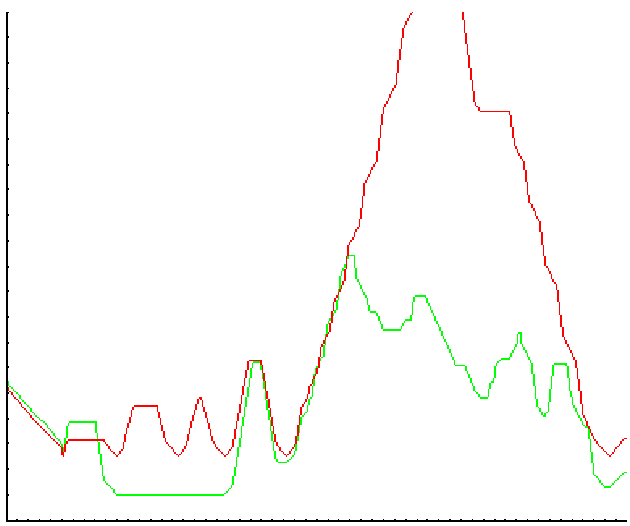

Fig. 14 Final trajectory error w.r.t. time (x: green, y: red). Axis are $[0,3600]\mathrm{s} \times [0,20]\mathrm{m}$. The graphic is cut on the top to be able to see in the same time the general trends and the specific parts discussed in this section.

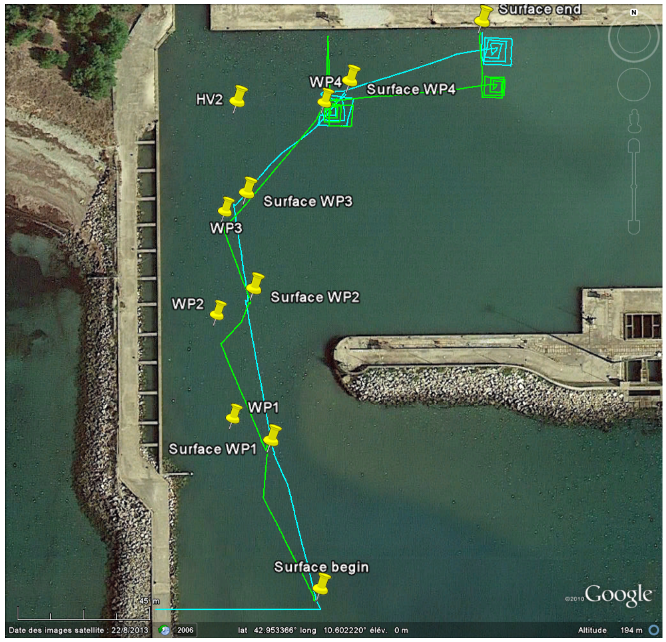

Fig. 15 Comparison between realtime estimation (in green) and post-processed estimation (in blue).

of the time the robot is not even visible from the operators point of view when the robot is navigating in the sea. In those circumstancies, a guaranteed set-membership method enables to quickly obtain an estimation of the robot trajectory despite the non-linearities of the equations and if all the assumptions made are valid, it guaranties that the robot is inside the trajectory envelope, otherwise it helps detecting unknown events generating inconsistencies that would have been missed otherwise if we were using probabilistic methods.

4 Conclusion

In this paper, we described a set-membership method used to take into account different types of proprioceptive and exteroceptive data to compute the envelope of the trajectory of a mobile robot. The method has been applied to the localization of a submarine with puntual GPS positions when at the surface and punctual range measurements with a surface vehicle of known position. A post-processing software has been made to demonstrate the simple use of interval methods in this kind of localization problem, and a similar algorithm was run in realtime on the submarine (with some limitations such as the absence of backward propagation in time because the robot does not know its future GPS fixes). The method is general enough to be also used for other kind of robots.

References

[ADJ13] C. Aubry, R. Desmare, and L. Jaulin. Loop detection of mobile robots using interval analysis. *Automatica*, 2013.

[APCV11] B. Allotta, L. Pugi, R. Costanzi, and G. Vettori. Localization algorithm for a fleet of three AUVs by INS, DVL and range measurements. In *15th International Conference on Advanced Robotics (ICAR)*, 2011.

[BJ14] A. Bethencourt and L. Jaulin. Solving non-linear constraint satisfaction problems involving time-dependant functions. *Mathematics in Computer Science*, 8(3), 2014.

[CGLP05] A. Caiti, A. Garulli, F. Livide, and D. Prattichizzo. Localization of autonomous underwater vehicles by floating acoustic buoys: a set-membership approach. *IEEE Journal of Oceanic Engineering*, 30(1):140–152, 2005.

[Cha13] G. Chabert. *IBEX 2.0, available at* `http://www.emn.fr/z-info/ibex/`. Ecole des Mines de Nantes, 2013.

[CJ09] G. Chabert and L. Jaulin. QUIMPER, A Language for Quick Interval Modelling and Programming in a Bounded-Error Context. *Artificial Intelligence*, 173:1079–1100, 2009.

[DB16] V. Drevelle and P. Bonnifait. Interval-based fast fault detection and identification applied to radio-navigation multipath. *International Journal of Adaptive Control and Signal Processing*, 30:154–172, January 2016.

[DJ16a] B. Desrochers and L. Jaulin. Computing a guaranteed approximation the zone explored by a robot. *IEEE Transaction on Automatic Control*, 2016.

[DJ16b] B. Desrochers and L. Jaulin. A minimal contractor for the polar equation; application to robot localization. *Engineering Applications of Artificial Intelligence*, 2016.

[DTDC14] J. A. Dit Sandretto, Gilles Trombettoni, David Daney, and Gilles Chabert. Certified Calibration of a Cable-Driven Robot Using Interval Contractor Programming. In Federico Thomas and Alba Perez Gracia, editors, *Computational Kinematics*, volume 15 of *Mechanisms and Machine Science*, pages 209–217. Springer Netherlands, 2014.

[FFD+16] G. Ferri, F. Ferreira, V. Djapic, Y. Petillot, M. Palau, and A. Winfield. The euRathlon 2015 Grand Challenge: The First Outdoor Multi-domain Search and Rescue Robotics Competition – A Marine Perspective. *Marine Technology Society Journal*, 50(4):81–97, 2016.

[FPL10] M. F. Fallon, G. Papadopoulos, and J. J. Leonard. *Cooperative AUV Navigation Using a Single Surface Craft*, pages 331–340. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[FW08] N. Fairfield and D. Wettergreen. Active localization on the ocean floor with multibeam sonar. In *OCEANS 2008*, 2008.

[GA09] R. Girault and P. Alain. Advances in geophysical sensor data acquisition. In *12th ASF Congress*, Rennes, France, 2009.

[GMG+12] J. González, I. Masmitjà, S. Gomáriza, E. Molino, J. del Río, A. Mànuel, J. Busquets, A. Guerrero, F. López, M. Carreras, D. Ribas, A. Carrera, C. Candela, P. Ridao, J. Sousa, P. Calado, J. Pinto, A. Sousa, R. Martins, D. Borrajo, A. Olaya, B. Garau, I. González, S. Torres, K. Rajan, M. McCann, and J. Gilabert. AUV Based Multi-vehicle Collaboration: Salinity Studies in Mar Menor Coastal Lagoon. *IFAC Proceedings Volumes*, 45(5):287 – 292, 2012.

[HPYZ08] S. Hou, S. Peng, Z. Yan, and W. Zhang. Research on the error model of INS/DVL system for autonomous underwater vehicle. In *IEEE International Conference on Automation and Logistics*, China, 2008.

[IJG14] M. S. Ibn Seddik, L. Jaulin, and J. Grimsdale. Phase based localization for underwater vehicles using interval analysis. *Mathematics in Computer Science*, 8(3), 2014.

[Jau09a] L. Jaulin. A Nonlinear Set-membership Approach for the Localization and Map Building of an Underwater Robot using Interval Constraint Propagation. *IEEE Transaction on Robotics*, 25(1):88–98, 2009.

[Jau09b] L. Jaulin. Robust set-membership state estimation; application to underwater robotics. *Automatica*, 45(1):202–206, 2009.

[Jau15] L. Jaulin. *Mobile Robotics*. ISTE editions, 2015.

[Jau16] L. Jaulin. Pure range-only SLAM with indistinguishable marks. *Constraints*, 00(0):00–00, 2016.

[JKDW01] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.

[JNR15] L. Jaulin, J. Ninin, and O. Reynet. IAMOOC : A MOOC on interval analysis with applications to parameter estimation and robot localization. `http://iamooc.ensta-bretagne.fr/`, 2015.

[JW02] L. Jaulin and E. Walter. Guaranteed robust nonlinear minimax estimation. *IEEE Transaction on Automatic Control*, 47(11):1857–1864, 2002.

[KK96] R. B. Kearfott and V. Kreinovich, editors. *Applications of Interval Computations*. Kluwer, Dordrecht, the Netherlands, 1996.

[LBSJ10] F. Le Bars, A. Bertholom, J. Sliwka, and L. Jaulin. Interval SLAM for underwater robots; a new experiment. In *NOLCOS 2010*, Bologna, Italy, 2010.

[LC14] M. L'Hour and V. Creuze. French Archaeology's Long March to the Deep-The Lune Project: Building the Underwater Archaeology of the Future. In M. Ani Hsieh, Oussama Khatib, and Vijay Kumar, editors, *ISER: International Symposium on Experimental Robotics*, volume 109 of *The 14th International Symposium on Experimental Robotics*, pages 911–927, Marrakech/Essaouira, Morocco, June 2014. Springer International Publishing.

[LJT16] M. Laranjeira, L. Jaulin, and S. Tauvry. Building Underwater Mosaics Using Navigation Data and Feature Extraction. *Reliable Computing*, 2016.

[LSRJ12] F. Le Bars, J. Sliwka, O. Reynet, and L. Jaulin. State estimation with fleeting data. *Automatica*, 48(2):381–387, 2012.

[MM08] J. Melo and A. Matos. Guidance and control of an ASV in AUV tracking operations. In *OCEANS 2008*, 2008.

[Moo79] R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, PA, 1979.

[MPM$^+$09] F. Maurelli, Y. Petillot, A. Mallios, P. Ridao, and S. Krupinski. Sonar-based AUV localization using an improved particle filter approach. In *OCEANS 2009*, 2009.

[PSSL14] L. Paull, S. Saeedi, M. Seto, and H. Li. AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, 39(1):131–149, 2014.

[RC10] G. Rui and M. Chitre. Cooperative positioning using range-only measurements between two AUVs. In *OCEANS 2010*, 2010.

[RNRT06] D. Ribas, J. Neira, P. Ridao, and J.D. Tardos. AUV localization in structured underwater environments using an a priori map. In *7th IFAC Conference on Manoeuvring and Control of Marine Crafts*, Lisboa, Portugal, 2006.

[RPW06] P. Rigby, O. Pizarro, and S. B. Williams. Towards Geo-Referenced AUV Navigation through Fusion of USBL and DVL Measurement. In *The Oceans Journal*, pages 1–6, 2006.

[RRNT06] D. Ribas, P. Ridao, J. Neira, and J.D. Tardos. SLAM using an imaging sonar for partially structured underwater environments. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[VM09] C. Vrignaud and J. Meyrat. Use of a DVL in an autonomous underwater vehicle for a rapid environmental assessment. In *ADCP in Action*, San Diego, 2009.