Hybridization of Monte Carlo and Set-membership Methods for the Global Localization of Underwater Robots

Renata Neuland[1], Jeremy Nicola[2], Renan Maffei[1], Luc Jaulin[2], Edson Prestes[1] and Mariana Kolberg[1]

Abstract— Probabilistic approaches are extensively used to solve high-dimensionality problems in many different fields. The particle filter is a prominent approach in the field of Robotics, due to its adaptability to non-linear models with multi-modal distributions. Nonetheless, its result is strongly dependent on the quality and the number of samples required to cover the space of possible solutions. In contrast, interval analysis deals with high-dimensionality problems by reducing the space enclosing the actual solution. Notwithstanding, it cannot precise where in the resulting subspace the actual solution is. We devised a strategy that combines the best of both worlds. Our approach is illustrated by solving the global localization problem for underwater robots.

I. INTRODUCTION

One of the main goals of robotics is to create autonomous robots able to operate without human supervision. Therefore, several challenges need to be addressed, such as acquiring the true state of the robot in the environment. This information is essential to different kinds of tasks that robots are expected to perform, which are usually repetitive, dangerous or difficult to humans. Examples include tasks like search and rescue, fire control, monitoring and surveillance, reconnaissance of people or other objects in hostile waters and the prevention of algae blooms in drink water reservoirs [1] [2].

Independently of the task or environment, a robot generally needs to know its correct position to perform its task successfully. Such necessity gives rise to the class of problems of *localization*, that consists in estimating the robot pose in an environment using sensor information. Additionally, this class of problems can be partitioned into two subclasses: *local* and *global localization.*

In the local localization problem, also called *position tracking*, the robot aims to actively estimate its pose from a known initial pose. Specifically, in mobile robotics, the robot must determine its most likely world pose using noisy sensor readings.

On the other hand, in the global localization problem, the robot must determine its pose without prior knowledge about its initial localization. This is the commonly called *kidnapped robot problem*. A different variation of global localization is the *wake-up robot problem*, in which a well-localized robot is carried to another position during its navigation without being told. In both problems, it is necessary to handle multiple candidate poses at same time [3].

Localization problems are usually addressed by probabilistic methods [3] or interval analysis [4]. The probabilistic approach provides an efficient solution to the localization problem, however, it is not mathematically guaranteed that the true robot pose is represented by the estimated solution. In contrast, interval methods are always guaranteed to compute a region that contains the true robot pose, considering that the problem is well modeled. Nevertheless, interval methods do not provide a precise robot pose, only a region that contains it, which may be too conservative.

In this paper, we discuss a new approach that combines probabilistic and interval strategies to solve the global localization problem in underwater environments. Through the use of interval analysis, the proposed method reduces the global uncertainty to a specific limited region. Then, knowing this region, a particle filter is performed to obtain enhanced information about the robot localization. With this approach, particles will be distributed in a smaller region, and the result will have boundaries of uncertainty well defined and mathematically guaranteed.

This paper is divided as follows. Section II shows related work. Section III introduces the localization problem and present the probabilistic and interval approaches. Our hybrid method is described in Section IV. Section V shows the results of the experiments performed to evaluate the hybrid method. Finally, Section VI presents the conclusions and discusses some future work.

II. RELATED WORK

Probabilistic and interval approaches for localization problem are easily found in the literature. Corke et al. [5] deal with AUV localization using two different methods, one based on a geometric approach employing acoustic sensor, and other using visual odometry with stereo cameras. Their methods were tested in an unstructured environment that contains sensor nodes. Le Bars et al. [6] use interval analysis to deal with offline state estimation problem, i.e., nonlinear estimation with fleeting data. Nak-Yong et al. [7] proposes an underwater localization approach using particle filter and time difference of arrival (TDOA) of acoustic signals from beacons scattered in the environment.

Forney et al.[8] use particle filter combined with an active control system to estimate the 2D planar position, velocity, and orientation states of a tagged shark. They use an AUV equipped with a stereo-hydrophone and a receiver system that detects acoustic signals transmitted by the tag. The AUV continually follows the single acoustic tag on the shark and estimates the shark position.

[1]Federal University of Rio Grande do Sul , Informatics Institute, Porto Alegre, Brazil {rcneuland; rqmaffei; prestes; mariana.kolberg} @inf.ufrgs.br
[2]ENSTA Bretagne, Brest, France {jeremy.nicola; luc.jaulin} @ensta-bretagne.fr

Meizel et al. [9] use set-membership estimation to treat localization problems of a vehicle in a polygonal environment. Their method does not require any explicit management of matching hypotheses, being able to deal with ambiguous situations. Their results show robustness to outliers and the ability to deal with nonlinear observation models without any approximation.

Abdallah et al. [10] propose a method called Box Particle Filter (BPF). The method, which was applied to the localization problem of a land vehicle, combines particle filter with interval approach, replacing groups of particles by boxes, called box particles. BPF require a number of box particles smaller than the particles number of a traditional particle filter. In the comparison with the traditional particle filter, the BPF showed a reduction in the running time. On the other hand, it showed no reduction in the error about the robot pose estimation.

III. Solving localization problem

In real world, robot sensors and actuators are not totally reliable, making it difficult to solve problems in robotics. Thus, it is necessary to develop methods that can deal efficiently with uncertainties that may arise from limited perception of sensors or actuators extremely inaccurate [3].

In this section we discuss two prominent approaches to solve the localization problem: the probabilistic approach of particle filters [3], and an interval approach using contractors [4].

The target environment treated in this work is underwater and unstructured, with distinguishable markers spread in the space. In this case, the markers are transponders and their location are supposed to be precisely known. The robot is able to detect the transponders signal and to compute the distance to each marker. Furthermore, the robot needs to extract information about its own motion, such as linear velocity and orientation (euler angles).

A. Particle filter localization

The particle filter, also known as Monte Carlo localization (MCL), is among the most popular approaches to solve the localization problem. The method performs the robot localization using a sample-based representation. From a set of controls u^t and measurements z^t captured by the robot at time-step t, the particle filter computes the belief $bel(x^t)$ about the robot localization using a set of particles, where each particle represents a different estimation of the robot pose [3], [11].

A set of M particles can be denoted by:

$$\chi^t = \{x^t_{[1]}, x^t_{[2]}, ..., x^t_{[M]}\} \qquad (1)$$

Each particle $x^t_{[m]}$ has associated a probability given by:

$$P(x^t|u^{t-1}, z^t) \qquad (2)$$

Algorithm 1 describes the particle filter localization method [3]. The first step is to randomly create a set of particles χ spread over the whole map, in the case of global

localization, or only around a specific pose, when dealing with the tracking problem. At each time-step, particles are propagated according to the input control u^t (line 3) and weighted (line 4). The weight of a particle m is obtained by comparing the current sensor measurements z^t with the observations that the particle should be taking at its current pose x^t_m :

$$P(z^t|x^t_m) \qquad (3)$$

This probability is proportional to the similarity between the information collected by the particle and the information collected by the robot sensors. The smaller the difference between such information the larger will be the particle importance factor.

Next, such factors are used to perform the resampling step, when particles are selected to join the new population χ^t (line 7). This step is usually implemented using the roulette wheel method, where particles with lower importance factors are more likely to be discarded.

Algorithm 1: *Monte Carlo Localization*

  Data: $\chi^{t-1}, map, u^{t-1}, z^t$
  Result: χ^t
1 $\overline{\chi}^t \leftarrow \chi^t \leftarrow \emptyset$
2 **for** $i \leftarrow 0 : M$ **do**
3 $\quad x^t_{[i]} \leftarrow move(x^{t-1}_{[i]}, u^{t-1})$;
4 $\quad w^t_{[i]} \leftarrow weight(x^t_{[i]}, z^t, map)$;
5 $\quad \overline{\chi}^t \leftarrow \overline{\chi}^t + \langle x^t_{[i]}, w^t_{[i]} \rangle$
6 **end**
7 $\chi^t \leftarrow resampling(\overline{\chi}^t)$
8 **return** χ^t

Among the main advantages of the particle filter are its ability to represent multimodal distributions and the low difficult of its implementation. Nevertheless, particle filters are subject to approximation errors that arise from several different reasons [3], such as:

- *The randomness in the resampling phase*. By using a random method in the resampling step, the method can accidentally discard good particles or cause loss of diversity. For instance, when the robot stays in the same pose during consecutive resampling steps, particles are discarded even without new information (z, u). Over time, the result will be a population of particles composed of multiple copies of a few different hypotheses.
- *The difference between proposal and target distributions*. The proposal distribution computed in the particles propagation relies on the robot motion model based on the odometry, while the target distribution computed in the particles weighting is based on the observation model of the robot sensors. When the difference between the accuracy of both models is large, the method might converge to a wrong place.

- *The particle deprivation problem.* When the search space is high-dimensional, some relevant regions with high likelihood to contain the true robot pose may not be covered. This often occurs due to the use of a small number of particles or unlucky random sampling that leaves some regions uncovered.

B. Interval State Estimator

Interval analysis gives a set of mathematics rules to do interval computations [12]. As the use of interval computations is increasing, some libraries have been created to facilitate the development of interval algorithms. Among them is IBEX[1], which was used in this research.

Interval analysis was first introduced to deal with the problem of rounding errors and input data uncertainties in calculations. The main idea is to represent the data as intervals and perform the computations using interval arithmetic. The use of interval analysis appears to be a good choice in robotics, since it deals with uncertainties in both data and computations [4].

A real interval $[x]$ is considered a connected subset of the $\mathbb{R}$, and is composed by a lower bound x^- and an upper bound x^+. A real interval is defined as:

$$[x] = [x^-; x^+] = \{x \in \mathbb{R} \mid x^- \le x \le x^+\}$$

Boxes are used to represent multidimensional data. A box is a subset of $\mathbb{R}^n$ and is defined as the Cartesian product of n intervals [4]:

$$[\mathbf{x}] = [x_1] \times [x_2] \times \cdots \times [x_n], \text{where}$$
$$[x_i] = [x_i^-; x_i^+] \text{ to } i = 1, \ldots, n$$

Each vector component $[x_i]$ is the projection of the $[\mathbf{x}]$ in the ith Cartesian axis.

Interval analysis allows to compute functions with intervals using arithmetic and set-theory operations. An interval image $\mathcal{I}$ of a real function $f([x])$ is defined as:

$$\mathcal{I}(f, [x]) = [\min\{f([x]) \mid x \in [x]\}; \max\{f([x]) \mid x \in [x]\}]$$

Any function f composed by arithmetic operators and elementary functions can be defined as an inclusion function, when [4]:

$$f([\mathbf{x}]) \subset [f]([\mathbf{x}])$$

For instance, considering the function $f(x) = x^2 + 2x + 4$, we can define $[f]([x]) = [x]^2 + 2[x] + 4$ as inclusion function. If $[x] = [-3; 4]$: $f([x]) = [3; 28]$ is subset of $[f]([x]) = [-2; 28]$.

When the result of $[f]([\mathbf{x}])$ is the smallest possible box, the function is called minimal. In most cases, this minimal inclusion function is not available. To circumvent this, we can use contractors [4].

Contractors are operators used to contract domains according to the proposed constraints, keeping the guarantee that no feasible value of the domain will be discarded. Contractors are often associated to the Constraint Satisfaction

[1]http://www.emn.fr/z-info/ibex/

Problem (CSP). A operator $\mathcal{C}$ is a contractor if, given a constraint c and a domain $[\mathbf{x}]$, it complies with the two following properties:

$$(c \cap [\mathbf{x}]) \subset \mathcal{C}([\mathbf{x}]) \qquad \text{(completeness)}$$
$$\mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}] \qquad \text{(contractance)}$$

There are different kinds of contractors. Here, we highlight the one called *forward-backward*. A constraint can be written like a function $y = f(x)$ or in the inverse form $x = f^{-1}(y)$, forward-backward works in two steps:

1) Forward: Contract y using $[y] \cap [f]([x])$
2) Backward: Contract x using $[x] \cap [f^{-1}]([y])$

For instance, regard the equation $x_3 = x_1 + x_2$, where $[x_1] = [-\infty; 5]$, $[x_2] = [-\infty; 4]$ and $[x_3] = [6; \infty]$.

- $x_3 = x_1 + x_2 \Rightarrow z \in [6; \infty] \cap ([-\infty; 5] + [-\infty; 4]) = [6; \infty] \cap [-\infty; 9] = [6; 9]$
- $x_1 = x_3 - x_2 \Rightarrow x \in [-\infty; 5] \cap ([6; \infty] - [-\infty; 4]) = [-\infty; 5] \cap [2; \infty] = [2; 5]$
- $x_2 = x_3 - x_1 \Rightarrow y \in [-\infty; 4] \cap ([6; \infty] - [-\infty; 5]) = [-\infty; 4] \cap [1; \infty] = [1; 4]$

After contraction the new domains are: $[x_1] = [6; 9]$, $[x_2] = [2; 5]$, $[x_3] = [1; 4]$.

Contractors can be used to deal with the localization problem as shown in Algorithm 2. This interval method is based on the same input data of the probabilistic method, yet, intervals are created considering the uncertainties.

Algorithm 2: *Localization using Contractors*

Data: $[\mathbf{x}], [\mathbf{u}], [\mathbf{z}]$
Result: $\mathbb{S}$

1   $\mathbb{S} = \emptyset$
2   **for** *i=0 : n* **do**
3     $\mathbb{F}$ = constraint set, based on Equations 4 and 5;
4     $[\mathbf{x}] = Contract([\mathbf{x}], \mathbb{F})$;
5     $\mathbb{S} = [\mathbf{x}]$
6     $move([\mathbf{x}], [\mathbf{u}])$; move the box, according Equation 6
7   **end**
8   $return \ \mathbb{S}$

The box $[\mathbf{x}]$ represents the search space. It is initialized to include the whole environment. The control inputs $[\mathbf{u}]$ are defined as the euler angles $[\phi], [\theta], [\psi]$, which are measured from a gyrocompass sensor, and the linear speed vector $[\mathbf{v}]$, measured from a loch-doppler sensor. They are obtained from the raw values $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}, \tilde{\mathbf{v}}$, as:

$$[\phi] = [-k_o \cdot \sigma_\phi + \tilde{\phi}, \tilde{\phi} + k_o \cdot \sigma_\phi]$$
$$[\theta] = [-k_o \cdot \sigma_\theta + \tilde{\theta}, \tilde{\theta} + k_o \cdot \sigma_\theta]$$
$$[\psi] = [-k_o \cdot \sigma_\psi + \tilde{\psi}, \tilde{\psi} + k_o \cdot \sigma_\psi]$$
$$[\mathbf{v}] = [-k_v \cdot \sigma_v + \tilde{\mathbf{v}}, \tilde{\mathbf{v}} + k_v \cdot \sigma_\psi]$$

The observation measurements z are distances $\tilde{d}$, calculated using the distance $\tilde{d}$ between the robot and each observed marker, defined as follows:

$$[z] = [d] = [-k_d \times \sigma_d + \tilde{d}, \tilde{d} + k_d \times \sigma_d]$$

Let $\sigma = (\sigma_\phi, \sigma_\theta, \sigma_\psi, \sigma_v, \sigma_d)$ be the standard deviations of the noise in the sensors, and $k = (k_o, k_v, k_d)$ be a parameter used to determine the width of the confidence interval.

The Equations 4 and 5 show the constraints of the problem. The constraints used by the method are associated to observed transponders and search space (line 3).

$$S = \bigcap_{n=1}^{N} \left\{ (x,y,z) \in \mathbb{R} | \sqrt{(x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2} \in [d]_n \right\} \tag{4}$$

$$S \cap [\mathbf{x}] \tag{5}$$

After defining the constraints, the search space is contracted (line 4). The contracted box is updated according to the Equation 6, enabling the box to aggregate the information about robot movements (line 6). This new box is used as the search space to the next iteration.

$$[\mathbf{x}]_{t+1} = [\mathbf{f}]([\mathbf{x}]_t) = [\mathbf{x}]_t + [\mathbf{R}]_{([\phi],[\theta],[\psi])} \times [\mathbf{v}] \tag{6}$$

IV. SOLVING UNDERWATER LOCALIZATION PROBLEM WITH HYBRID METHOD

In this section, we describe our approach to deal with the underwater localization problem. The proposed method is hybrid: it inherits interval and probabilistic characteristics, using contractors and particle filter.

The chosen interval approach uses contractors to reduce the uncertainty about the robot localization to a box. Since this box is calculated using interval rules, we can assume that the true robot pose is contained in the box, provided that a correct modeling of the process is made (sensors and motion models).

However, this box representation may be large and little informative. To extract more information from the box is used a probabilistic approach, such as the particle filter.

The method works as described in Algorithm 3. At each iteration, the method use contractors and particle filter to find a solution through a sequence of steps. In the first step, the constraints are defined (line 3) and the search space is contracted into a smaller box (line 4).

In the first iteration, a population with m particles will be created inside of the defined search space (line 6). These particles are randomly spread in the space considering the box bounds. In the remaining iterations, the particles are evaluated and discarded, if situated outside the search space (line 11). For each particle discarded, a new particle is randomly created in the search space and added to current population (line 12).

The next steps are the weighting (line 16) and the resampling (line 17) of the population. The weighting is based on Equation 3 and the resampling uses the roulette wheel method. After these steps, the particles average pose will be computed and defined as the current robot localization (line 18). The current population and the box that represents the current search space are propagated according to Equations 2 and 6 respectively.

Some expected hybrid approach advantages are:

Algorithm 3: *Contractor + Particle Filter*

Data: $[\mathbf{x}], u, z, [\mathbf{u}], [\mathbf{z}]$
Result: $\mathbb{S}$

1 $\mathbb{S} = \emptyset$
2 **for** $i=0 : n$ **do**
3      $\mathbb{F}$ = constraint set based on Equations 4 and 5;
4      $[\mathbf{x}] = Contract([\mathbf{x}], \mathbb{F})$;
5   **if** $!\ \mathbb{P}$ **then**
6          $\mathbb{P} = createParticle(m, [\mathbf{x}])$;
7   **end**
8   **else**
9   **for** $j=0 : m$ **do**
10   **if** $p^j \cap [\mathbf{x}] = \emptyset$ **then**
11                 $\mathbb{P} = \mathbb{P} \setminus p^j$;
12                 $\mathbb{P} = \mathbb{P} \cup createParticle(1, [\mathbf{x}])$;
13   **end**
14   **end**
15   **end**
16     $weight(\mathbb{P})$;
17     $resampling(\mathbb{P})$;
18     $\mathbb{S} = \mathbb{S} \cup avgParticle(\mathbb{P})$;
19     $move([\mathbf{x}], [\mathbf{u}])$;
20     $move(\mathbb{P}, u)$;
21 **end**
22 $return\ \mathbb{S}$

- *High coverage of the uncertainty region.* Given the reduction of search space, the particles may be better used. The interval method bounds all possible poses of the robot, thus the particles are forced to remain inside this limit. Without increasing the number of particles, it is possible to increase the coverage of the real interesting region.

- *Fast detection of the wrong coverage.* If the method converges to a wrong place, due to flawed propagation models, the interval approach produces an empty solution set. This situation is an indicative of the occurrence of errors.

- *The solution has the uncertainty mathematically guaranteed.* The interval approach does not discard any feasible solution. Thus, the interval solution is mathematically guaranteed, given that the problem was well-modeled. Additionally, the results of the particle filter process are included in the interval limits, therefore the particle uncertainty is also defined by these limits.

- *More information can be extracted using the particles, in case the interval result is not enough informative.* As said earlier, the interval method does not discard any feasible solution, so the result can be a box of large size, that provides little information. By using a particle filter inside this region, it is possible to extract probabilistic information about the robot localization in the box interior.

V. EXPERIMENTS

We performed simulated experiments to evaluate the proposed method.

The results obtained by our hybrid method are compared to the results of the traditional particle filter. The test environments used for the data extraction were simulated using MORSE Simulator [13]. In the simulations, we defined transponders as markers. Also, the robot was equipped with gyrocompass sensor and loch-doppler sensor to extract the euler angles of its orientation and its linear velocity. The robot used in the simulations was the generic submarine provided by the MORSE.

We proposed 3 test scenarios, whose main difference is the amount of information available to the localization process. All scenarios are 3-dimensional and have a size of $400m \times 400m \times 400m$, considering the initial search space as $[\mathbf{x}] = [-200; 200][-200; 200][-400; 0]$. The environments 1, 2 and 3 have respectively 2, 4 and 8 transponders . Their positions (in meters) are distributed as follows:

- **Environment 1**: (-112; -10; -20), (180; 30; -300);
- **Environment 2**: (-150; 130; -40), (-25; -115; -130), (180; 30; -300), (90; -75; -60);
- **Environment 3**: (-150; 130; -40), (-38; 15; -200), (-112; -10; -20), (-25; -115; -130), (40; 100; -100), (180; 30; -300), (90; -75; -60), (130; -190; -255);

Table I shows the sensors parameters used by the methods during the experiments.

TABLE I

PARAMETERS OF THE SIMULATIONS

Measurement	Error inserted (σ=standard deviation)	k
Loch-Doppler Sensor	0.04 m/s	k_v= 3
Gyrocompass Sensor	0.02°	k_o= 3
Transponders Distance	0.3 m	k_d= 3

The robot made 2 different paths for each environment. Each configuration was tested 10 times, totalizing 60 tests with each method. In all tests we used 1000 particles as the particle filter population.

The tests using the environments 2 and 3 presented very similar results, therefore, for concision reasons, we decided to show only the results referent to the environment 3, compared to the results of environment 1.

As we observed in our tests, there is a limit to insert transponders and see improvements in the results. However, the use of an insufficient number of transponders harms the full potential of the method. This number varies according to the environment size and transponders range. Nonetheless, when comparing the graphs of environment 1 with the graphs of environment 3, we can see that, by increasing the number of transponders, the results significantly improve for both the traditional particle filter and our hybrid method.

The graphs depicted in Figures 1 to 8 show the error in meters between the real robot position and the computed position by the methods. The vertical axes represent the error in meters, while the horizontal axes represent the methods iterations. The black line is the average among the results for each iteration (considering the ten runs) and the grey lines are the standard deviation of the average.

Comparing the graphs of the Figures 1 with 2 and 3 with 4 we can see that the hybrid approach obtained better results in these tests. Moreover, the standard deviation is significantly reduced. This occurs because the box defined by interval analysis limits the spread of the particles.
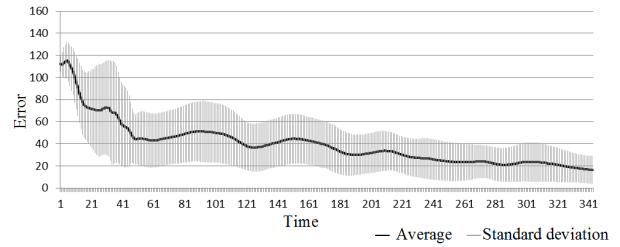

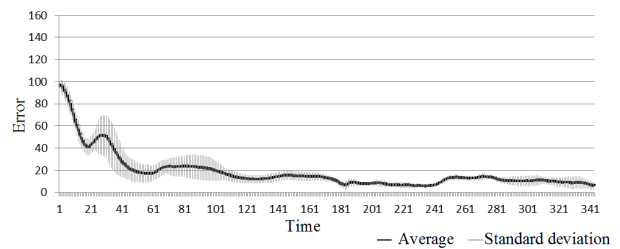Fig. 1.   Traditional particle filter - trajectory 1 environment 1

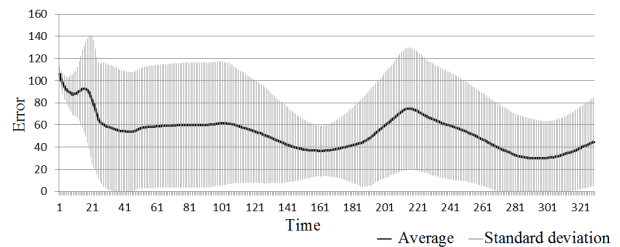Fig. 2.   Hybrid method - trajectory 1 environment 1

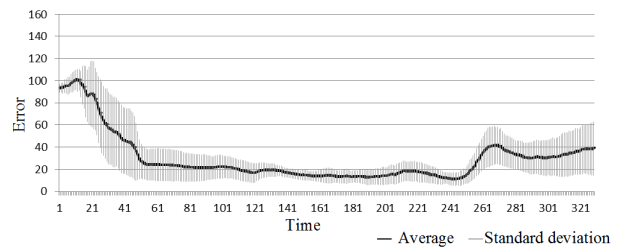Fig. 3.   Traditional particle filter - trajectory 2 environment 1

Fig. 4.   Hybrid method - trajectory 2 environment 1

Such improvements are obtained using only two transponders, when using eight transponders the improvements are more significant. Comparing the graphs of the Figures 5 with 6 and 7 with 8, paying attention to the scale on the vertical axis, we see that the improvement regarding accuracy is very significant. The traditional particle filter generates results with errors above 20 meters, contrasting with our hybrid

approach that generates results with errors lower than 1.5 meters. The peaks in the Figures 6 and 7 occur due to the box growth, caused by the increase of the uncertainty that cannot be reduced by contractors.
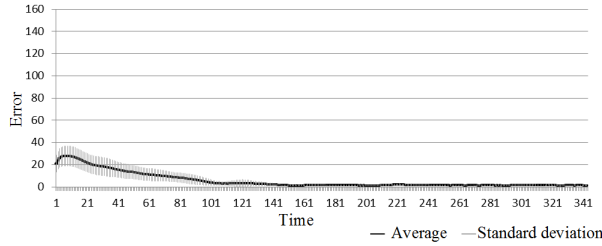

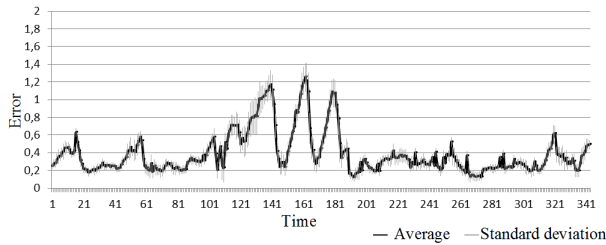Fig. 5. Traditional particle filter - trajectory 1 environment 3

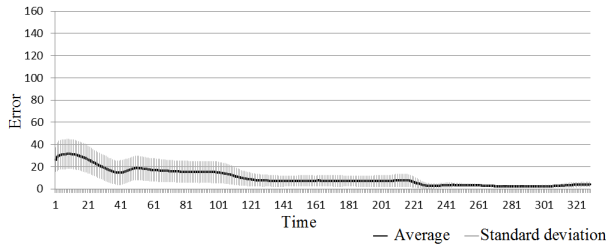Fig. 6. Hybrid method - trajectory 1 environment 3

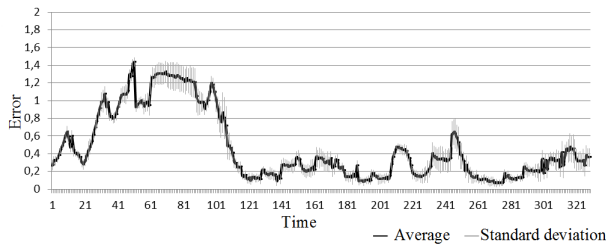Fig. 7. Traditional particle filter - trajectory 2 environment 3

Fig. 8. Hybrid method - trajectory 2 environment 3

Regarding cost, the hybrid method has extra computational burden, though is only a small percentage of the traditional particle filter cost. Table II shows the percentage of the extra time to running the hybrid method compared to the traditional particle filter in these experiments.

VI. CONCLUSIONS

After the experiments analysis, we can draw some conclusions. The main one is that the hybrid method obtains

Environment	1		3	
Trajectory	1	2	1	2
Extra time	4.3%	4.5%	9.1%	9.6%

better results in this kind of environment, despite the small increase in the running time. Using interval analysis helps to reduce the uncertainty area, making it possible to obtain a better particles distribution. Inconsistencies in the data of the hybrid method can be more easily detected than in the traditional particle filter, since such errors usually generate empty sets in interval methods.

As future work, we intend to improve the method for applications with indistinguishable markers. Other possibility is the extension for the SLAM problem.

ACKNOWLEDGMENT

REFERENCES

[1] S. Shatara, X. Tan, E. Mbemmo, N. Gingery, and S. Henneberger, "Experimental investigation on underwater acoustic ranging for small robotic fish," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 712–717.
[2] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus, "Data muling over underwater wireless sensor networks using an autonomous underwater vehicle," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2091–2098.
[3] S. Thrun, W. Burgard, D. Fox, *et al.*, *Probabilistic robotics*. MIT press Cambridge, 2005, vol. 1.
[4] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*. Springer, 2001.
[5] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, and I. Vasilescu, "Experiments with underwater robot localization and tracking," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 4556–4561.
[6] F. Le Bars, J. Sliwka, L. Jaulin, and O. Reynet, "Set-membership state estimation with fleeting data," *Automatica*, vol. 48, no. 2, pp. 381–387, 2012.
[7] N.-Y. Ko, T. G. Kim, and Y. S. Moon, "Particle filter approach for localization of an underwater robot using time difference of arrival," in *OCEANS, 2012 - Yeosu*, 2012, pp. 1–7.
[8] C. Forney, E. Manii, M. Farris, M. A. Moline, C. G. Lowe, and C. M. Clark, "Tracking of a tagged leopard shark with an auv: Sensor calibration and state estimation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5315–5321.
[9] D. Meizel, O. Lévêque, L. Jaulin, and E. Walter, "Initial localization by set inversion," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 6, pp. 966–971, 2002.
[10] F. Abdallah, A. Gning, and P. Bonnifait, "Adapting particle filter on interval data for dynamic state estimation," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2. IEEE, april 2007, pp. II–1153 –II–1156.
[11] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1322–1328.
[12] E. Seignez and A. Lambert, "Complexity study of guaranteed state estimation applied to robot localization," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*. IEEE, 2008, pp. 398–405.
[13] G. Echeverria, N. Lassabe, A. Degroote, and S. Lemaignan, "Modular open robots simulation engine: Morse." in *ICRA*. IEEE, 2011, pp. 46–51. [Online]. Available: http://dblp.uni-trier.de/db/conf/icra/icra2011.htmlEcheverriaLDL11