Robust Hybrid Interval-Probabilistic Approach for the Kidnapped Robot Problem

ARTICLE HISTORY
Compiled January 18, 2019

ABSTRACT

For a mobile robot to operate in its environment it is crucial to determine its position with respect to an external reference frame using noisy sensor readings. A scenario in which the robot is teleported to another position during its operation without being told, known as the kidnapped robot problem, complicates global localisation. In addition to that, sensor malfunction and external influences of the environment can cause not expected errors, called outliers, that negatively affect the localisation process. This paper proposes a method based on the fusion of a particle filter with bound-error localisation, which is able to deal with outliers in the measurement data. The application of our algorithm to simulated data shows an improvement over conventional probabilistic filtering methods.

KEYWORDS
Bayesian Filter; Interval Analysis; Kidnapped Robot Problem; Mobile Robotics.

1. INTRODUCTION

One major challenge in robotics is creating robots capable of performing tasks without human supervision. Acquiring information about the true state of the robot in its environment is essential to carry out autonomous missions. This necessity gives rise to a class of localisation problems that is characterized by the use of sensor information to estimate the robot's position in its environment. The different kinds of localisation problems can be separated into global and local localisation Thrun, Burgard, and Fox (2005).

In local localisation, also known as position tracking, the initial robot position is assumed to be known. The uncertainties associated with the tracking process are local and restricted to the region near to the true robot position. In global localisation, on the other hand, the robot is placed somewhere in its environment before being put to operation and needs to localize itself based on the sensor readings, which forms a more difficult problem than tracking as boundedness of the position error cannot be assumed.

A yet more difficult scenario is described by the kidnapped robot problem, which forms a subclass of global localisation problems Zhang (2010) and describes a situation where a well-localised mobile robot is teleported to an arbitrary location without being told. That is, the robot strongly believes itself to be somewhere else at the time of the kidnapping. Kidnapping may arise from external, environment influences such as drift, water flow, or earth quake Stephant, Charara, and Meizel (2004) Stphant, Charara, and Meizel (2007).

During the localisation process the robot senses its environment to extract relevant information for self-localisation. Various different kinds of sensors, such as cameras, lasers, or sonars can be used for this purpose. The output signals of all these sensors, however, are subject to noise and possibly contain outliers Jaulin, Kieffer, Didrit, and Walter (2001), where the latter is not accounted for in the robot model.

Diverse interval or probabilistic methods are commonly used to solve the localisation problem given noisy sensor data. Also, attempts to fuse both interval techniques and probabilistic filters are found in the literature. In this paper we propose a hybrid approach based on a particle filter and a set-membership method which is robust to outliers to solve the kidnapped robot problem.

This paper is divided as follows. Section 2 presents the state of the art of localisation techniques, followed by an introduction to the basic concepts of probabilistic filtering and interval analysis applied to self-localisation in Section 3. In Section 4 we explain in detail our proposal and in Section 5 we describe the experiments as well as the results. Finally, in section 6 we conclude and present possible future work.

2. RELATED WORK

Current research can be divided into two major areas, that is methods using interval analysis and methods based on Bayesian filtering. Desrochers et al. Desrochers, Lacroix, and Jaulin (2015) presented an interval based algorithm, which exploits geometrical information of the environment in the form of a single image containing depth information, to deal with the kidnapped robot problem. Their technique is suitable to situations where initial models are inaccurate and the number of outliers is large. However, it provides only an initial set of feasible positions but is not able to continuously track a robot's position.

Han et al. Han, Kim, and Myung (2013) proposed a landmark-based particle filter algorithm using a fish-eye system. The algorithm extracts the distance and the angles of a mobile robot with respect to the landmarks. Using this information, the algorithm determines a region including the robot's position, and randomly spreads particles across this region. Their method is able to estimate the robot pose with only two landmarks, but to obtain smoother localisation results, according to Han, odometry should be used as well, which makes the algorithm computationally intense.

The Box Particle Filter developed by Abdallah et al. Abdallah, Gning, and Bonnifait (2008) is a hybrid method that uses GPS, a gyrometer, and an odometer to track a land vehicle. Their method combines particle filtering with interval analysis by replacing groups of particles by boxes, called box particles. Interval computation is used to reduce the number of particles without compromising accuracy. Since the box particle filter only requires a small number of particles it shows a reduction in the running time when compared to the traditional particle filter. However, it showed no reduction in the error of the robot pose estimation.

Ashokaraj et. al proposed sensor based robot localisation using an extended Kalman filter Ashokaraj, Tsourdos, Silson, and White (2004b) as well as an unscented Kalman filter Ashokaraj, Tsourdos, Silson, and White (2004c) in combination with interval analysis to bound the estimation error in the presence of landmarks. If the position estimate of the Kalman filter lay outside of the region it was corrected with the geometrically closest point on the boundary. In Ashokaraj, Tsourdos, Silson, White, and Economou (2004) multiple interval robot positions are processed using a fuzzy logic weighted average algorithm to obtain a single robot interval position. The error of an unscented Kalman filter position estimate is then bound by the interval robot position as described above. In Ashokaraj, Tsourdos, Silson, and White (2004a) Ashokaraj et. al used ultrasonic sensor with limited range and SIVIA or IMAGESP. As opposed to the previous works by the authors, when the point estimate of the Kalman filter lay outside the box, the interval robot position was mapped to a point estimate equal to the center of the box and adopted by the unscented Kalman filter. The corresponding covariance was constituted of an ellipse enclosing the box. That is, the major and minor axis radius of the ellipse was used as the covariance matrix values. Their method resulted in a more accurate position estimate.

Neuland et. al proposed a method that combines interval and probabilistic approaches to deal with the global localisation problem Neuland, Nicola, et al. (2014) Neuland, Maffei, Jaulin, Prestes, and Kolberg (2014). The strategy identifies regions of high interest through interval analysis to distribute particles accordingly. Thus, the method provides well-defined error boundaries and higher precision results than those obtained by both strategies applied separately. However, it cannot cope with outliers.

Several others researchers applied hybrid methods to self-localisation, as Kim et al. Kim, Lee, Myung, and Choi (2014) who used a template matching technique and a particle filter to detect artificial landmarks and estimate the pose of a vehicle. Ko et al. Ko, Kim, and Moon (2012) presented a particle-filter based strategy for localisation of an underwater vehicle using acoustic signals from multiple beacons. Forney et al. Forney et al. (2012) proposed a particle filter to track 'tagged' agents, e. g., a shark. Meizel et al. Meizel, Lévêque, Jaulin, and Walter (2002) applied a set-membership estimation to localize a vehicle using range measurements. Guyonneau et. al. Guyonneau, Lagrange, Hardouin, and Lucidarme (2012) modelled localisation as a constraint satisfaction problem, using an interval combination of bisections and contractions techniques.

3. BACKGROUND

This section briefly introduces the underlying concepts about interval analysis and probabilistic Bayesian filtering necessary for the presentation of our localisation algorithm in Section 4.

3.1. *Interval Analysis*

Interval analysis is based on the idea of enclosing real numbers in intervals and real vectors in boxes so as to perform mathematical operations using these structures Jaulin et al. (2001). A real interval $[x]$ can be defined as a connected subset of $\mathbb{R}$, composed by an upper and a lower bound,

$$[x] = [x^-, x^+] = \left\{ x \in \mathbb{R} \,|\, x^- \leq x \leq x^+ \right\}. \tag{1}$$

Multidimensional data is represented using interval vectors, also referred to as boxes. A box $[\boldsymbol{x}]$ is a subset of $\mathbb{R}^n$ defined as the Cartesian product of intervals,

$$[\boldsymbol{x}] = [x_1] \times [x_2] \times \cdots \times [x_n] \tag{2}$$

where n is the dimension.

Basic operations of real computation are naturally extended to intervals. Given the intervals $[x]$, $[y]$ and a binary operator $\diamond \in \{+, -, *, /\}$, the corresponding interval operation is given by

$$[x] \diamond [y] = \big\{x \diamond y \in \mathbb{R} \,|\, x \in [x], y \in [y]\big\}. \tag{3}$$

Interval computations can be applied to arbitrary non-linear problems to generate mathematically guaranteed solutions Jaulin et al. (2001), while at the same time reducing computational cost Abdallah et al. (2008) or increasing the precision of results Neuland, Nicola, et al. (2014). Since interval methods do not discard any feasible solution, when lacking of constraints narrowing the solution set, it may remain large and therefore uninformative. This represents a major limitation of localisation techniques based on interval analysis.

Through interval operations it is possible to treat different problems in robotics. For instance, the localisation problem can be modeled as a set inversion problem described by

$$\mathbb{X} = f^{-1}(\mathbb{Y}) = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, f(\boldsymbol{x}) \in \mathbb{Y}\} \tag{4}$$

where $\mathbb{X}$ is the preimage of $\mathbb{Y}$ under the function f. In the localisation context, the set $\mathbb{X}$ represents all the feasible positions of the robot given a set of observations $\mathbb{Y}$. The observations may be distance measurements between the robot and a set of landmarks, in which case f is the euclidean distance function

$$\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} = d_i \tag{5}$$

where the triple (x_i, y_i, z_i) denotes the position of the i-th landmark.

Each observation is transformed into an interval to include sensor uncertainties in the model. This can be done by inflating the measurement vector $\boldsymbol{d}$ with the sensor error,

$$[\boldsymbol{d}] = [\boldsymbol{d} - 3\sigma, \boldsymbol{d} + 3\sigma], \tag{6}$$

where σ denotes the standard deviation of the sensor. An error boundary at a distance of 3σ with respect to the measurement ensures that 99.73% of the measurements lie inside the box. Replacing the position (x, y, z) in Equation 5 with its interval counterpart as well gives us a set of $i \in \{1, \ldots, n\}$ constraints

$$\sqrt{([x] - x_i)^2 + ([y] - y_i)^2 + ([z] - z_i)^2} = [d_i] \tag{7}$$

This constraint satisfaction problem can be solved using the SIVIA-algorithm (Set Inversion Via Interval Analysis), a non-linear bounded-error estimator introduced by Jaulin and Walter Jaulin and Walter (1993). Its main idea is bisecting and testing the search space narrowing down the set of all feasible solutions. Given an initial search space modeled as a box $[\boldsymbol{x}]$ the SIVIA algorithm works as follows:

- If $f([\boldsymbol{x}])$ does not intersect with $\mathbb{Y}$, $[\boldsymbol{x}]$ is discarded.
- If $f([\boldsymbol{x}])$ is contained in $\mathbb{Y}$, $[\boldsymbol{x}]$ is considered part of the solution.
- If $f([\boldsymbol{x}])$ intersects with $\mathbb{Y}$, but is not contained in $\mathbb{Y}$, $[\boldsymbol{x}]$ is bisected given the width of its largest component is bigger than a predefined limit ϵ. If not, $[\boldsymbol{x}]$ is considered as part of the solution.

In real world problems, observations often contain outliers, that is the observation exceeds the predetermined maximum range given by Equation 6. In such a case the constraints should be relaxed, i.e., a box can be considered part of the solution even if it violates a certain number of constraints.

[Figure 1 near here]

A possible scenario corrupted by an outlier is presented in Fig. 1, where the robot observes four markers at the same time in an unidimensional environment. Table 1 shows the marker positions and the measured distance between the robot and each of the markers, respectively, while the bold measurement in the last row shall be the outlier. Note how the measurement is already infalted to an interval.

[Table 1 near here]

Based on the respective measurement interval, the robot can determine an interval position, given in the last column of Table 1. The solution set is then computed as the intersection of all four intervals in the last column. If we disregard the possibility of outliers the solution set is the empty set and therefore the constrained satisfaction problem does not have a solution. However, if we allow one outlier, a solution exists and is defined by the intersection $m_1 \cap m_2 \cap m_3 = [40, 55]$, i.e. the robot position is enclosed by the interval $[40, 55]$.

The SIVIA algorithm was extended to the RSIVIA algorithm by Jaulin (2009) to handle this new scenario. The difference of relaxed SIVIA is that $[\boldsymbol{x}]$ will be part of the solution if $f([\boldsymbol{x}])$ intersects with at least k intervals of the set $\mathbb{Y}$, where k is the number of measurements minus the number of outliers.

3.2. Probabilistic Bayesian Filtering

As opposed to the interval methods presented above, in probabilistic bayesian filtering uncertainty is represented by probability distributions over the space of hypotheses. Therefore, instead of simply including or excluding points of the state space in the solution set, the degree of uncertainty is captured. Problems involving perception and action in the real world are some of the applications of probabilistic algorithms, since the methods are scalable to complex and unstructured environments, and usually robust in the face of sensor limitations and environment dynamics. However, computational inefficiency is one of the disadvantages frequently associated with probabilistic methods Thrun et al. (2005).

One of the most popular methods in the context of self-localisation is Monte Carlo Localisation (MCL) proposed by Dellaert et al. Dellaert, Fox, Burgard, and Thrun (1999). MCL has mainly gained its popularity due to the fact that it works well with different localisation problems, it is able to represent multi-modal distributions, and it is easy to implement Thrun et al. (2005) Han et al. (2013).

MCL uses a set of M particles, each of which representing the possible robot posistion at time step t,

$$\mathcal{X}_t = \{\boldsymbol{x}_t^{[1]}, \boldsymbol{x}_t^{[2]}, \ldots, \boldsymbol{x}_t^{[M]}\}. \tag{8}$$

An important concept of probabilistic estimation is the *belief*. The belief $bel(x_t)$ represents the internal knowledge of the robot in relation to the state of the environment and is an abbreviation of the Bayes filter posterior

$$bel(x_t) = p(x_t \,|\, z_{1:t}, u_{1:t},). \tag{9}$$

where $z_{1:t}$ and $u_{1:t}$ denotes the sequence of all measurements and control inputs up to and including time t, respectively. It can be constructed recursively from the predicted belief

$$\overline{bel}(x_t) = p(x_t \,|\, z_{1:t-1}, u_{1:t},). \tag{10}$$

This probability distribution describes the state x_t, conditioned to the current robot controls $u_{1:t}$ and the measurements up to the previous time step, denoted by $z_{1:t-1}$.

To construct the belief from its prediction, the weight $w_t^{[m]}$ of each particle $x_t^{[m]}$ is evaluated, given by

$$w_t^{[m]} = p(z_t \,|\, x_t^{[m]}). \tag{11}$$

In the following we sum up the individual steps of the MCL method:

- Create a set of particles distributed over the whole search space.
- Move each particle according to the control u_t.
- Weight each particle according to the measurements z_t.
- Resample the current set of particles.

Traditional MCL as presented may suffer from particle deprivation Thrun et al. (2005) which is the lack of particles in relevant regions of the search space. When the robot is kidnapped and brought to a region without particles, it cannot determine its position and therefore MCL is not able to recover from the failure of localisation. To mitigate this shortcoming, Thrun et al. (2005) proposed the use of a simple heuristic which adds new random particles in the whole search space. However, it is desirable to merely add these random particles in regions of high likelihood.

4. PROPOSED ROBUST HYBRIDIZATION

We propose a robust hybrid interval-probabilistic approach for the kidnapped robot problem which is based on the method proposed in Neuland, Nicola, et al. (2014) Neuland, Maffei, et al. (2014) that combines the MCL and the SIVIA algorithm in order to achieve improved self-localisation. The main contributions are:

- *Dealing with kidnapped robot problems*: The problem addressed by this research is a more difficult variant of the global localisation problem. During its operation the robot may believe to be in a position that does not coincide with the true position. Then, the robot needs to detect this and it needs to recover from the global localisation failure.

- *Dealing with outliers*: Our method is suitable to deal with datasets containing outliers, which are unavoidable when using real sensor data.
- *Strong integration of the approaches during the resampling step*: The resampling step is necessary to define the particles that will survive to compose the new set of particles. In our approach we consider particles as punctual boxes and use the current constraints based on interval analysis to identify the particles that will survive to next generation.

The key idea of our method is to only perform MCL over a delimited region of the search space obtained by interval analysis. Thus, the particles cover only high probability regions and no particles will be wasted in areas that are not feasible.

Algorithm 1 presents how the regions of interest are obtained. It computes the region of interest based on the search space $[\boldsymbol{x}]$ and the set of measurements $[\boldsymbol{z}]$. Where the initial dimensions of $[\boldsymbol{x}]$ represent the same dimensions of the environment and each element of $[\boldsymbol{z}]$ represents an interval measurement observed by the robot. The algorithm returns a set of boxes, $\mathbb{S}$, that cover the space of feasible solutions.

First, we initialize the number of outliers q to be zero (line 3), since we hope to find a solution with the lowest possible number of outliers. We also define the solution set $\mathbb{S}$ as empty (line 4). The core of the Algorithm is a loop (lines 5 to 8) that is executed until the solution set is non-empty. At each execution of the loop, $\mathbb{S}$ is sought by RSIVIA method using q to relax the constraints, as shown in the example (Table 1) of Section 3.1. While $\mathbb{S}$ is empty, the RSIVIA algorithm is executed once again considering $q + 1$ outliers.

Algorithm 1 setRegion

1: **Data:** $[\boldsymbol{x}], [\boldsymbol{z}]$
2: **Result:** $\mathbb{S}$
3: $q = 0$
4: $\mathbb{S} = \emptyset$
5: **while** $\mathbb{S} == \emptyset$ **do**
6: $\quad \mathbb{S} = rsivia([\boldsymbol{x}], [\boldsymbol{z}], q)$
7: $\quad q = q + 1;$
8: **end while**
9: *return* $\mathbb{S}$

The solution set $\mathbb{S}$ represents all the feasible robot positions, while all positions in $\mathbb{S}$ are equiprobable, independent of its size. Then, we use MCL to obtain a more precise estimation of the robot position. Particles are only spread within the boxes in $\mathbb{S}$, which bounds the error of the probabilistic position estimate. The modified MCL Algorithm is depicted in Algorithm 2. It starts with information about the initial search space modeled by a box $[\boldsymbol{x}]$ (line 1). So, the current robot observations $\boldsymbol{z}_1$ are collected and transformed into intervals $[\boldsymbol{z}_1]$ (line 2), as given by Equation 6. After that, a region containing all feasible robot positions is represented by the set $\mathbb{S}_0$ (line 3), which was generated from the interval approach described in Algorithm 1. Now, we spread particles uniformly over the space confined by $\mathbb{S}_0$ (line 4).

Algorithm 2 Proposed robust method

1: **Data:** $[\boldsymbol{x}]$
2: $[\boldsymbol{z}_1] = toInterval(\boldsymbol{z}_1)$;
3: $\mathbb{S}_0 = setRegion([\boldsymbol{x}_0], [\boldsymbol{z}_1])$;
4: $\boldsymbol{\mathcal{X}_0} = initializeParticles(\mathbb{S}_0)$;
5: **for** t = 1 : n **do**
6: $\quad [\boldsymbol{u}_t] = toInterval(\boldsymbol{u}_t)$;
7: $\quad [\boldsymbol{z}_t] = toInterval(\boldsymbol{z}_t)$;
8: $\quad [\boldsymbol{s}_t] = moveRegion(\mathbb{S}_{t-1}, [\boldsymbol{u}_t])$;
9: $\quad moveParticles(\boldsymbol{\mathcal{X}}_{t-1}, \boldsymbol{u}_t)$;
10: $\quad \mathbb{S}_t = setRegion([\boldsymbol{s}_t], [\boldsymbol{z}_t])$;
11: $\quad weighting(\boldsymbol{\mathcal{X}}_t, \boldsymbol{z}_t)$;
12: $\quad resampling(\boldsymbol{\mathcal{X}}_t, \mathbb{S}_t)$;
13: $\quad showLocalisation(\boldsymbol{\mathcal{X}_t})$;
14: **end for**

The algorithm has a loop (lines 5 to 14) to be run at each new robot motion or sensing. Inside the loop (lines 6 and 7) the sensing information is obtained and modeled as intervals. We compute the robot motion first by moving $\mathbb{S}_{t-1}$ (line 8). To simplify the motion of $\mathbb{S}_{t-1}$, the set is converted into a box $[\boldsymbol{s}_{t-1}]$, by the interval hull operator

$$[\boldsymbol{s}_{t-1}] = hull(\mathbb{S}_{t-1}), \tag{12}$$

and then propagated according to the robot controls $[\boldsymbol{u}_t]$ generating $[\boldsymbol{s}_t]$. Then we move the set of particles $\boldsymbol{\mathcal{X}}_{t-1}$ accordingly to $\boldsymbol{u}_t$ (line 9). The current region defined by $\mathbb{S}_t$ is updated (line 10) using the measurement $[\boldsymbol{z}_t]$. The next step is the evaluation of the particle weight (line 11).

During the resampling process (line 12), particles outside $\mathbb{S}_t$ are discarded. In the case all particles are outside the set $\mathbb{S}_t$, all will be discarded and it means that a localization failure or a kidnap happened. Treating each particle as a punctual box, allows to use simple operations of set theory to determine if a particle is inside $\mathbb{S}_t$.

[Figure 2 near here]

For instance, as shown in Fig. 2, the robot observes a marker m (represented by *) in a distance of $[1, 3]$ meters. We need to define if a particle p_1 (represented by $\diamond$) and p_2 (represented by $\circ$) are in the set $\mathbb{S}$. Information about the position of the environment objects are given by Table 2.

[Table 2 near here]

The computation can be done as follows:

$$\begin{aligned}
&d = \sqrt{(p_x - m_x)^2 + (p_y - m_y)^2} \\
&d = \sqrt{(1-0)^2 + (2-0)^2} \\
&d = 2.236 \\
&d \in [1, 3]
\end{aligned}$$

where d is the distance between the particle and the mark. We have a constraint $d \in [1, 3]$ in accordance with the robot measurement. p_1 does not violate the constraint, thus, p_1 is kept. Now, considering a particle p_2, we have $d = 4$ so $d \notin [1, 3]$, consequently, p_2 is discarded.

Since the number of particles does not change over time, each discarded particle in the resampling process is randomly repositioned over the set $\mathbb{S}_t$ to keep the same initial size of $\boldsymbol{\mathcal{X}_0}$. Spreading particles randomly in the feasible region is a known technique to recover from failures that can cause a wrong convergence of the particle set, as mentioned in section 3.2. Then, the roulette wheel method is applied on the current set of particles as in traditional MCL.

Finally, we show the robot pose given by the average of the particles of $\boldsymbol{\mathcal{X}_t}$ (line 13). We validate our method applied to kidnapped robot problem performing some experiments presented in the next section.

5. EXPERIMENTS AND RESULTS

In our experiments we compared our method against the conventional Monte Carlo Localisation in two and three dimensions, using data simulated with the MORSE simulator Echeverria, Lassabe, Degroote, and Lemaignan (2011). The robot was equipped with a three-dimensional linear velocity sensor with a standard deviation of 0.05 meters and performed global localisation measuring its distance to multiple distinguishable markers. The markers were spread randomly in the environment and its positions are known a priori. The maximum measurable distance for each sensor was 100 meters, so that it was possible that the robot did not see all markers all the time. The robot velocity sensor has a standard deviation of 0.05 meters and the orientation angles obtained from the three-dimensional gyroscope have a standard deviation of 0.005 rad. Since the farther the marker is located from the robot the noisier the measurement will be, we assumed an increase in standard deviation of 0.05m per meter increase in distance. Besides, the measurements are corrupted by outliers, too. We assumed that in 30% of the sensing 0% to 20% of the measurements of that sensing are outliers.

The number of markers and the number of particles vary among the tests. We simulated scenarios with 12, 20, and 40 markers in the 2D environment, while in the 3D environment we simulated scenarios with 20 and 40 markers. Experiments in each scenario were carried out using 2000, 8000 and 10000 particles. The duration of one trajectory was 83 min in 2D and 33 min in 3D, containing one kidnap event each.

[Figure 3 near here]

Figure 3 depicts a boxplot of the localisation error of the 3D trajectory using 40 markers, where A, C, and E are MCL results and B, D, and F are results of our approach. A and B performed localisation using 2000 particles, C and D used 8000 particles and E and F used 10000. Considering the median error of localisation using 2000, 8000 and 10000 particles, MCL errors are 179.01, 175.54 and 177.25 meters, while our method errors are 2.72, 1.72 and 1.51 meters, respectively.

[Figure 4 near here]

Figure 4 shows a boxplot of the localisation error for the experiments using 10000 particles in the environments with 12, 20 and 40 markers and a 2D trajectory. In this graphic we can see the effects of increasing the number of markers in the localisation result. A, C and E show the errors of MCL and B, D and F show the errors of our method. The mean error in meters to 12, 20 and 40 markers are 179.54, 170.37 and 168.48 for MCL and 19.82, 3.10 and 1.38 for our approach, respectively. When more information is available to be used during the localisation process, the results are better.

[Figure 5 near here]

Figure 5 shows the localisation error at each time step. This test was performed using 10000 particles in an environment with 40 markers and 2D trajectory. Analyzing the graphic it is easy to identify the moment of the kidnapping at about 4000 sec characterized by the jump in the MCL mean error. The mean error in meters ten interactions before the kidnap is 38.63 with MCL and 1.71 with our method and the mean error in meters ten timesteps after the kidnap is 178.75 with MCL and 0.6 with our method. Using the hybrid approach the robot localisation could quickly recover after the kidnapping thanks to the delimited region obtained by the interval constraints.

As shown by all graphics presented so far our method obtains more precise results for the robot localisation than MCL. However, the improvement causes an increase in computational cost. Table 3 shows the percentage of extra time consumed by the hybrid approach in relation to MCL during the tests with the 2D and 3D trajectories and using 12, 20, and 40 markers. The first quartile, median and second quartile of the extra time percentage are presented, as well as the median extra time in seconds, also presented to give a better idea of the time running. The experiments were performed in an Intel I7 with 16GB RAM. The interval part of our hybrid approach creates one constraint based on each observed marker, thus, when the number of markers is increased the computational time required to deal with this information is also increased.

[**Table 3 near here**]

We also tested the MCL in the 3D environment with 40 markers using 15000 particles, this test is equivalent in relation to time running to our approach in the same scenario using 10000 particles. However, our method keeps getting better results, the test with MCL presented localisation errors of the 129.26, 174.41 and 207.93 meters to first quartile, median and third quartile, respectively. While our method by using the same time running presented localisation errors of the 0.09, 1.51 and 3.41 meters to first quartile, median and third quartile, respectively.

6. CONCLUSION

In this paper we presented a robust interval-probabilistic approach to deal with the kidnapped robot problem. Through interval computations the method is able to reduce the search space and spread particles only in regions of high probability. Therefore, we obtain a better probability distribution and consequently more precise results than when using MCL alone. The main contributions of this work are related to the strategy to overcome challenges of the kidnapped robot problem, the robustness against outliers and the integration during the resampling process with interval constraints to decide the particles survival by treating them as punctual boxes.

Although the experiments showed that our approach provides more precise localisation, the hybrid method needs more time to compute a solution than conventional MCL. However, as also shown by the experiments, using the additional computational intensity in MCL, that is increasing the number of particles, is not enough to obtain similar localisation precision.

In the future, we intend to improve the method so that it can be applied to environments with indistinguishable markers as well.

References

Abdallah, F., Gning, A., & Bonnifait, P. (2008). Box particle filtering for nonlinear state estimation using interval analysis. *Automatica*, *44*(3), 807–815.

Ashokaraj, I., Tsourdos, A., Silson, P., & White, B. (2004a). Mobile robot localisation and navigation using multi-sensor fusion via interval analysis and ukf. *Proceedings of the 2004 Towards Autonomous Robotic Systems (TAROS), University of Essex, Colchester, UK*.

Ashokaraj, I., Tsourdos, A., Silson, P., & White, B. (2004b, July). Sensor based robot localisation and navigation: using interval analysis and extended kalman filter. In *2004 5th asian control conference (ieee cat. no.04ex904)* (Vol. 2, p. 1086-1093 Vol.2).

Ashokaraj, I., Tsourdos, A., Silson, P., White, B., & Economou, J. (2004, June). A fuzzy logic approach in feature based robot navigation using interval analysis and ukf. In *Ieee annual meeting of the fuzzy information, 2004. processing nafips '04.* (Vol. 2, p. 808-813 Vol.2).

Ashokaraj, I., Tsourdos, A., Silson, P., & White, B. A. (2004c, Sept). Sensor based robot localisation and navigation: using interval analysis and unscented kalman filter. In *2004 ieee/rsj international conference on intelligent robots and systems (iros) (ieee cat. no.04ch37566)* (Vol. 1, p. 7-12 vol.1).

Dellaert, F., Fox, D., Burgard, W., & Thrun, S. (1999). Monte carlo localization for mobile robots. In *Robotics and automation, 1999. proceedings. 1999 ieee international conference on* (Vol. 2, pp. 1322–1328).

Desrochers, B., Lacroix, S., & Jaulin, L. (2015). Set-membership approach to the kidnapped robot problem. In *Intelligent robots and systems (iros), 2015 ieee/rsj international conference on* (pp. 3715–3720).

Echeverria, G., Lassabe, N., Degroote, A., & Lemaignan, S. (2011, May). Modular open robots simulation engine: Morse. In *2011 ieee international conference on robotics and automation* (p. 46-51).

Forney, C., Manii, E., Farris, M., Moline, M. A., Lowe, C. G., & Clark, C. M. (2012, May). Tracking of a tagged leopard shark with an auv: Sensor calibration and state estimation. In *Robotics and automation (icra), 2012 ieee international conference on* (p. 5315-5321).

Guyonneau, R., Lagrange, S., Hardouin, L., & Lucidarme, P. (2012). The kidnapping problem of mobile robots: A set membership approach. In *7th national conference on control architectures of robots.*

Han, S.-B., Kim, J.-H., & Myung, H. (2013). Landmark-based particle localization algorithm for mobile robots with a fish-eye vision system. *Mechatronics, IEEE/ASME Transactions on*, *18*(6), 1745–1756.

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

Jaulin, L., Kieffer, M., Didrit, O., & Walter, E. (2001). *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics.* Springer.

Jaulin, L., & Walter, E. (1993). Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, *29*(4), 1053–1064.

Kim, D., Lee, D., Myung, H., & Choi, H.-T. (2014). Artificial landmark-based underwater localization for auvs using weighted template matching. *Intelligent Service Robotics*, 1-10. Retrieved from `http://dx.doi.org/10.1007/s11370-014-0153-y`

Ko, N.-Y., Kim, T. G., & Moon, Y. S. (2012, May). Particle filter approach for localization of an underwater robot using time difference of arrival. In *Oceans, 2012 - yeosu* (p. 1-7).

Meizel, D., Lévêque, O., Jaulin, L., & Walter, E. (2002). Initial localization by set inversion. *Robotics and Automation, IEEE Transactions on*, *18*(6), 966-971.

Neuland, R., Maffei, R., Jaulin, L., Prestes, E., & Kolberg, M. (2014). Improving the precision of auvs localization in a hybrid interval-probabilistic approach using a set-inversion strategy. *Unmanned Systems*, *2*(04), 361–375.

Neuland, R., Nicola, J., Maffei, R., Jaulin, L., Prestes, E., & Kolberg, M. (2014). Hybridization of monte carlo and set-membership methods for the global localization of underwater robots. In *Intelligent robots and systems (iros 2014), 2014 ieee/rsj international conference on* (pp.

199–204).
Stephant, J., Charara, A., & Meizel, D. (2004, April). Virtual sensor: application to vehicle sideslip angle and transversal forces. *IEEE Transactions on Industrial Electronics*, *51*(2), 278-289.
Stphant, J., Charara, A., & Meizel, D. (2007). Evaluation of a sliding mode observer for vehicle sideslip angle. *Control Engineering Practice*, *15*(7), 803 - 812. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0967066106000803` (Special Issue on Award Winning Applications2005 {IFAC} World Congress)
Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics (intelligent robotics and autonomous agents series)*. Cambridge, MA, USA: MIT Press.
Zhang, L. (2010). *Self-adaptive Markov Localization for Single-Robot and Multi-Robot Systems* (Theses, Université Montpellier II - Sciences et Techniques du Languedoc). Retrieved from `https://tel.archives-ouvertes.fr/tel-00491010`

Table 1. Markers position and observation

Marker	Position	Measured distance in m	Robot position in m constraints
m_1	10	$[30, 50]$	$[40, 60]$
m_2	100	$[40, 60]$	$[40, 60]$
m_3	110	$[55, 75]$	$[35, 55]$
m_4	120	$[\mathbf{25}, \mathbf{35}]$	$[85, 95]$

Table 2. Objects position of Fig. 2

	Represented by	Position (x,y)
m	$*$	$(0, 0)$
p_1	$\diamond$	$(1, 2)$
p_2	$\circ$	$(4, 0)$

Table 3. Extra time consumed by the hybrid approach.

Traj	**Markers**	**Q1 (%)**	**Median (%)**	**Q3(%)**	**Median (sec)**
2D	12	7.4	8.2	9.1	0.011
2D	20	12.3	13.4	14.5	0.018
2D	40	24.5	26.6	28.8	0.038
3D	20	16.2	17.5	19.1	0.024
3D	40	28.6	30.5	33.5	0.044

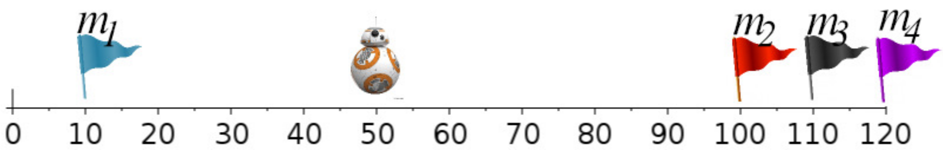

Figure 1. Robot measurements in an unidimensional environment.

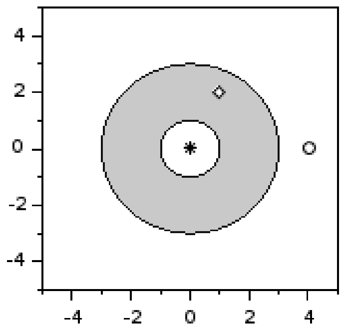

Figure 2. Defining particles survival according to interval constraints.

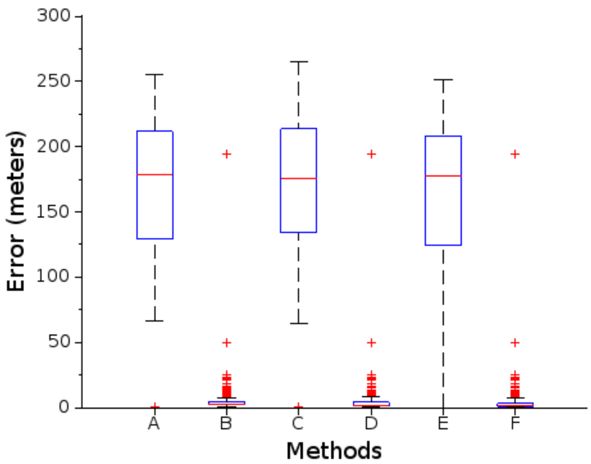

Figure 3. Benefits of increasing the number of particles. All data are from 3D trajectory using 40 markers. A (MCL) and B (Our) use 2000 particles. C (MCL) and D (Our) use 8000 particles. A (MCL) and B (Our) use 10000 particles.

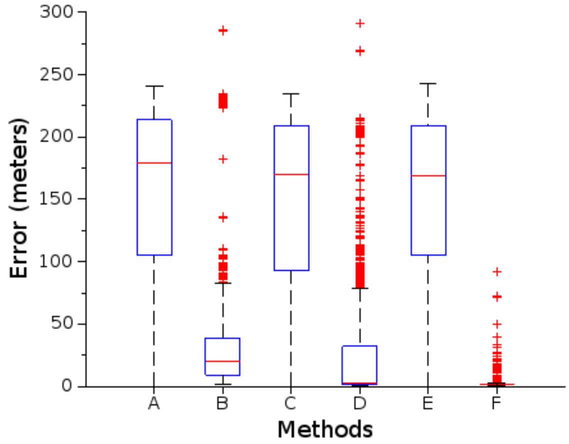

Figure 4. Benefits of increasing the number of markers. All data are from 2D and the tests used 10000 particles. A (MCL) and B (Our) use 12 markers. C (MCL) and D (Our) use 20 markers. A (MCL) and B (Our) use 40 markers.

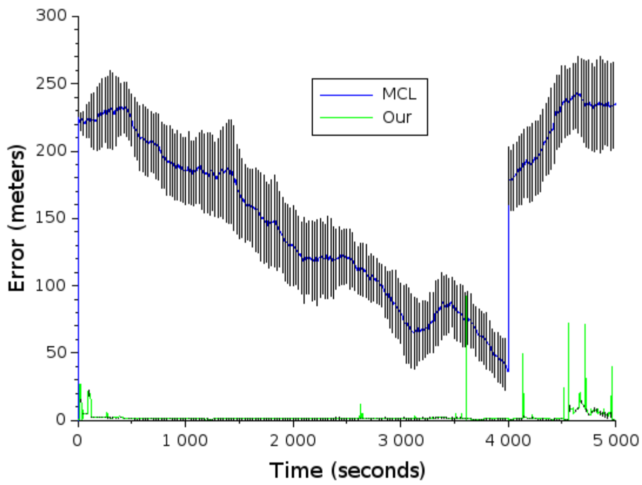

Figure 5. Mean and standard deviation of the localisation error in the 2D trajectory with 40 markers, 10000 particles are used.