2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

A NEW ESTIMATION METHOD FOR MULTISENSOR FUSION BY USING INTERVAL ANALYSIS AND PARTICLE FILTERING

Amadou Gning , Fahed Abdallah and Philippe Bonnifait

HEUDIASYC, UMR CNRS 6599
Université de Technologie de Compiègne, France
gningelh@hds.utc.fr, fahed.abdallah@hds.utc.fr, philippe.bonnifait@hds.utc.fr

ABSTRACT

This paper presents a new fusion strategy that mixes Interval Analysis techniques and particle filters for data fusion and state estimation purposes occurring in many robotic perception problems. The method requires a small number of "box particles". This, in fact, answers one of major problems when using particle filters techniques. We report the case study of a land vehicle localization problem and we make a comparison based on real data between the performance of a particle filter and the new developed strategy.

Keywords: State filtering and estimation, sensor fusion, particle filter, Kalman filter, interval analysis.

1. INTRODUCTION

In many robotics applications, the perception function is essential to accomplish intelligent tasks by monitoring the environment in the face of uncertainty and variability. Usually, for many problems like obstacle detection, localization or Simultaneous Localization and Map Building (SLAM) [8], the perception system of a mobile robot relies on the fusion of several kinds of sensors like video cameras, lidars, dead-reckoning sensors, etc. The multisensor fusion problem is popularly described by state space equations defining the interesting state, the evolution and observation models. Based on this state space description, the state estimation problem can be formulated as a state tracking problem. To deal with this state observation problem, when uncertainty occurs, the probabilistic Bayesian approaches are the most used in robotics, even if new approaches like the set-membership one (also known as bound errors approach) [5] or Belief theory (also known as Dempster-Shafer) [9] have proved themselves in some applications. The Extended or Unscented Kalman filters have demonstrated their efficiency in many real applications. Recently, Particles Filters (PF) have been extensively studied [3] [4] because of their ability to deal with non linearity, non gaussianity and multimodal density functions. Nevertheless, particle filter methods suffer from some drawbacks. These methods are very sensitive to non consistent measures or large measurement errors. In fact, the efficiency of the filter depends mostly on the number of particles used in the estimation process and on the propagation function used to re-allocate weights to these particles at each iteration. If the imprecision (i.e. bias and noise) of the available information is high, and in order to explore a significant part of the state space, the number of particles has to be very large which induces complexity problems non adapted to a real-time implementation. Several works try to combine approaches in order to overcome these shortcomings (see for example [6] and references therein). Other works use statistical approaches to increase the efficiency of particle filters by adapting the size of sample sets during the estimation process [4].

In this paper, we propose a new state estimation scheme called Box Particle Filter (BPF) which tries to reduce these drawbacks, particulary the particle number, while conserving the particle filters advantages by using "box particles" instead of "discrete particles". The key idea is to use Interval Analysis, constraint propagation techniques and to model noises by bounded errors. This is reinforced by two possible understandings of an interval in one dimension:

1. An interval represents infinity of particles continuously distributed on the interval.
2. An interval represents a particle imprecisely located in the interval.

In a BPF, the PF paradigm is followed except that no random noise is generated as it is usually done in Monte Carlo filtering [6]. There are also similarities between the BPF and nonlinear bayesian estimation using Gaussian sum approximations in that any set can be approximated by a sum of boxes [1].

2. Sketch of particle filtering

Many multisensor fusion problems can be described by a state space representation:

$$\begin{cases} x_{k+1} = & f(x_k, u_k, v_k) \\ y_k = & g(x_k, w_k) \end{cases} \quad (1)$$

where $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \longrightarrow \mathbb{R}^{n_x}$ is a possibly non-linear function defining the state at time $k+1$ from the previous state at time k, the input u_k and an independent identically distributed process noise sequence $v_k, k \in \mathbb{N}$. We note by n_x, n_u and n_v, respectively, the dimensions of the state, the input and process noise vectors. The function $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \longrightarrow \mathbb{R}^{n_y}$ is a possibly non-linear function defining the relation between the state and the measurement at time k, $w_k, k \in \mathbb{N}$ is an i.i.d measurement noise sequence. n_y, n_w are dimensions of the measurement and measurement noise vectors, respectively. The states and the measurements up to time k will be represented by $X_k = \{x_i, i = 1, \cdots, k\}$ and $Y_k = \{y_i, i = 1, \cdots, k\}$, respectively.

The sketch of a particle filter algorithm is as follows. Initially, all particles have equivalent weights attached to them. To progress to the next time instance, two steps are performed in sequence. First, at the prediction step, the state of every particle is updated according to the motion model. An accurate dynamical model is essential for robust properties of the algorithm and for achieving real-time performance. Next, during the measurement step, new information that became available about the system is used to adjust the particle weights. The weight is set to be the likelihood of this particle state describing the true current state of the system, which can be computed, via bayesian inference, to be proportional to the probability of the observed measurements given the particle state (assuming all object states are equiprobable). The sample states are then redistributed to obtain uniform weighting for the next algorithm iteration by resampling them from the computed posterior probability distribution. At any time, some characteristics (position, speed etc.) can be directly computed, if desired, by using the particle set and weights as an approximation of the true probability density function.

3. Interval analysis

We briefly present interval analysis and we describe the constraints propagation technique which is also called consistency technique in some research works.

3.1. Elements of interval analysis

A real interval, denoted $[x]$, is defined as a closed and connected subset of $\mathbb{R}$, and a box $[\mathbf{x}]$ of $\mathbb{R}^{n_x}$ as a cartesian product of n_x intervals: $[\mathbf{x}] = [x_1] \times [x_2] \cdots \times [x_n] = \times_{i=1}^{n_x} [x_i]$. Usually, interval analysis is used to model quantities which vary around a central value within certain bounds.

When working with intervals, one should introduce the inclusion function $[f]$ of a function f, defined such that the image by $[f]$ of an interval $[x]$ is an interval $[f]([x])$ [7]. This function is calculated such that the interval enclosing the image set is optimal. One should also extend all elementary arithmetic operations like +, -,*,/, etc to the bounded error context and extend usual operations between sets of $\mathbb{R}^n$, e.g., $\subset, \supset, \cap, \ldots$

Different algorithms, called contractors, exist in order to reduce the size of boxes enclosing the solutions. For the fusion problem considered, we have chosen to use constraints propagation techniques [7], because of the great redundancy of data and equations.

3.2. Constraints Satisfaction Problem (CSP)

Consider a system of m relations f_m linking variables x_i of a vector x of $\mathbb{R}^{n_x}$ by a set of equations of the forms: $f_j(x_1, \ldots, x_{n_x}) = 0$, $j = 1 \ldots m$, which can be written in a compact way as $f(\mathbf{x}) = \mathbf{0}$, where $\mathbf{f}$ is the cartesian product of the f_j's.

Definition 1 (Constraints Satisfaction Problem)**.** *A Constraints Satisfaction Problem $\mathscr{H}$ is the problem which gathers a vector of variables* $\mathbf{x}$ *from an initial domain* $\mathbf{D}$ *and a set of constraints* $\mathbf{f}$ *linking the variables x_i of* $\mathbf{x}$*.*

Note that under the interval framework, $\mathbf{D} = [\mathbf{x}] = \times_{i=1}^{n_x} [x_i]$. The CSP consists on finding the values of $\mathbf{x}$ which satisfy $f(\mathbf{x}) = \mathbf{0}$. The solution set of the CSP will be defined as $\mathbf{S} = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{f}\{\mathbf{x}\} = \mathbf{0}\}$. Note that $\mathbf{S}$ is not necessary a box. Under the interval framework, solve the CSP is interpreted as *finding the minimal box* $[\mathbf{x}'] \subset [\mathbf{x}]$ such that $\mathbf{S} \subset [\mathbf{x}']$.

3.3. Waltz's Contractor

Definition 2 (Contractor)**.** *A contractor is defined as an operator used to contract the initial domain of the CSP, and thus to provide* a new box $[\mathbf{x}'] \subset [\mathbf{x}]$ *such that* $\mathbf{S} \subset [\mathbf{x}']$.

There are different kinds of methods to develop contractors. Each of these methods may be adapted to some specific CSPs and not to others. The method used in this paper is the Waltz's algorithm [7] which is based on the constraint decomposition on primitive ones and on the use of forward-backward propagation (FBP) technique [7] for each of primitive constraints. A primitive constraint involves only an arithmetic operator or a usual function (cos, exp, etc.). The principle of the Waltz's contractor is to use FBP for each constraint, without any a priori order, until the contractor *becomes inefficient*. The use of this contractor appears to be specially efficient when one has a redundancy of data and equations. In fact, this is the case for the data used in section (5).

The principle of FBP is explained via the following example. Let us consider the constraint $z = x.\exp(y)$. At first, this constraint is decomposed into two primitive constraints, $a = \exp(y)$ and $z = x.a$ (Where a is an auxiliary variable initialized by $[a] = [0, +\infty[$). By using the inclusion functions [exp] and $[(\exp)^{-1}] = [\ln]$, the FBP works as follows: $\mathbf{1-Forward\ propagation}$,we have F_1: $[a] = [a] \cap [\exp]([y])$ and F_2: $[z] = [z] \cap [x].[a]$, $\mathbf{2-Backward\ propagation}$, B_3: $[x] = [x] \cap ([z]/[a])$, B_4: $[a] = [a] \cap ([z]/[x])$ and B_5: $[y] = [y] \cap [\ln]([a])$.

Note that Waltz's algorithm is independent of the non-linearities and provides a locally consistent contractors [7].

4. Box particle strategy

For real data measurements, one usually receives different answers when repeating the same measurement. This variation is due to stochastic error and statistical methods are used to model maximum information from the results. In many applications, the interval framework seems to be a good methodology to deal with non-white and biased measurements specially when these measures vary around a central value within certain bounds. This approach is used in the following to introduce an interval based multisensor data fusion approach.

Instead of point particles and probabilistic models for the errors and for the inputs, the key idea for BPF is to use box particles and a bounded error model.

4.1. Initialization

In order to explore the state space, one can split the state space region under consideration in N boxes $\{[x^{(i)}]\}_{i=1}^{N}$ with empty intersection and associate equivalent weight for each of them. A first advantage expected with this initialization using boxes is the possibility to explore the space with a reduced number of particles.

4.2. Propagation or prediction step

In this step, the state of every box particle is updated according to the evolution model thanks to interval analysis tools. Knowing the box particles $\{[x^{(i)}]\}_{i=1}^{N}$ and the input $\{[u_{(k)}]\}$ at step k, the boxes at step $k+1$ are built using the following propagation equation: $[x_{k+1}^i] = [f]([x_k^i],[u_k])$, where $[f]$ is an inclusion function for f. The interesting propriety one can notice here is that, in order to propagate the box particles, the bounded error method is used without introducing any noise.

4.3. Measurement update

In this step, one uses the new measurement to adjust the particle weights and contract the boxes.

4.3.1. Innovation. Innovation for BPF is a quantity which should indicates the proximity between the real and the predicted measure boxes. In the bounded error framework, this quantity can be evaluated as the intersection between these two boxes. Thus, for all box particles, $i = 1 \cdots N$, we have to predict box measurements using $[z_{k+1}^i] = [g]([x_{k+1}^i])$, where $[g]$ is an inclusion function for g. The innovation here consists on the intersection with the box real measure $[y_{k+1}]$. This intersection is calculated as $[r_{k+1}^i] = [z_{k+1}^i] \cap [y_{k+1}]$.

4.3.2. Likelihood. Under the bounded error framework, it's obvious to conclude that, a box for which the predicted measure box hasn't an intersection with the real measure box should be penalized and a box particle for which the predicted measure is included in the real measure box should be favorite. This lead us to construct a measure of the *box likelihood* as: $A^i = \prod_1^p A^i(j)$ where $A^i(j) = \frac{|[r_{k+1}^i(j)]|}{|[z_{k+1}^i(j)]|}$, p is the dimension of the measure and $|[X]|$ is the width of $[X]$.

4.3.3. Box particles contraction. This step, used only for box particles, doesn't appear in the particle filter algorithm. In fact, in the particle filter algorithm, each particle is propagated without any information about the variance of it's position. Note that the weight of the particle give us only an information about certainty when using this particle. In an opposite manner, after been propagated, the width of each box particle is assumed to take into account the imprecision caused by the model errors and inputs imprecisions. In order to conserve a judicious width of each box, one should use contraction algorithms which eliminate the non consistent part of the box particle with respect to the box measure [5]. This is in fact similar to the correction step of Kalman filtering when the variance-covariance matrix is corrected using the measure [6].

Thus, if the innovation $[r_{k+1}^i]$ is not empty, then we should contract the box particle $[x_{k+1}^i]$ using the intersection box $[r_{k+1}^i]$ and Waltz's algorithm to obtain a new box particle $[x_{k+1}^i]^{new}$. Else, $[x_{k+1}^i]^{new} = [x_{k+1}^i]$ and the box particle stays unchanged.

4.3.4. Weights update. The update of the weights is constructed by multiplying the previous weight by each *box likelihood* as: $\omega_{k+1}^i = (\prod_1^p A^i(j))\omega_k^i = A^i\omega_k^i$

4.4. Normalization

This step is necessary in order to use normalized weights so that their sum is equal to one: $\omega_{k+1}^i \longleftarrow \frac{\omega_{k+1}^i}{\sum_{j=1}^N \omega_{k+1}^j}$

4.5. Estimation

The state can be estimated by the center means of the weighted box particles as: $\hat{x}_k = \sum_{i=1}^N \omega_k^i C_k^i$, where C_k^i is the center of the box particle i. One can use also a maximum weight estimate, i.e the state estimate will be the center of the box particle with the larger weight. A pessimist confidence in the estimation will be a very well determined area consisting in a box which contain all the possible weighted boxes. Hence, one can assign for this box the noun of *enclosing box*. Note that the estimation $\hat{x}_k$ is calculated by using N vectors C_k^i. Thus, another confidence in the estimation based on the confidence of each C_k^i can be calculated with the expression: $\hat{P}_k = \sum_{i=1}^N \omega_k^i P_k^i$, where P_k^i is the partial confidence generated when using each box particle center C_k^i. In practice, P_k^i can be taken as the half width of each box particle. Thus, $\hat{P}_k = \sum_{i=1}^N \omega_k^i \frac{|[x_k^i]|}{2}$.

4.6. Resampling

After some iterations, only few box particles may be likely and the rest may have weights close, or even exactly equal to zero. Thus, one has to sample box particles according to their weights. Box particles that have high weights are more likely to survive, whereas those with lower weights are less likely. The resampling can be efficiently implemented using a classical algorithm for sampling N ordered independent identically distributed variables [6]. The problem of the resampling is that the resulting samples are dependent since there is a *big chance* that the samples will be drawn from a few number of ancestors. In the case of particle filter algorithm, instead of representing the smooth probability density as they should, particles are clustered into groups. Therefore, some artificial noise should be added to the resampled particles in order to lessen the dependency. This step avoids the particle filter to fall down. One can use the same strategy for box particles by adding an artificial noise to the bounds of the box. Moreover, regarding the possibilities given by boxes properties, other techniques of resampling can be considered. For example, in order to obtain independent and small boxes around regions with high likelihoods, it's easily perceived that we can divide each box by the correspondent number of realization after sampling. Nevertheless, in the bounded error area, the choice of the number of divisions that one have to do for each dimension constitutes an open problem under study [2]. After the resampling step, we have to assign the same weight for all box particles.

Note that an estimation of the effective sample size N_{eff} is given by [6]: $N_{eff} = \frac{1}{\sum_{i=1}^{N}(\omega_k^i)^2}$. The resampling step can be performed if the effective number of samples is less than some threshold N_{th} which is determined experimentally.

A summary of the BPF algorithm is given in Figure 1.

5. Application to dynamic localization using GPS, a gyro and an odometer

Let consider the localization problem of a land vehicle. The vehicle frame origin M is chosen at the middle of rear axle. The elementary rotation and displacement between two samples can be obtained with good precision uniquely using a fiber optic gyrometer and the two rears wheels ABS sensors. Between two sampling instants, elementary rotations of the two rear wheels are integrated by counters. These values allow calculating the distances travelled between two samples by the rear wheels. Thus, one can obtain at instant k, the elementary displacement covered by M, $\delta_{S,k} = \frac{\delta_{RR,k}+\delta_{RL,k}}{2}$ and its elementary rotation $\delta_{\theta,k} = \delta_{\theta,k}^{gyro}$ where $\delta_{RR,k}$ and $\delta_{RL,k}$ denote the measured variables with valued counted between two samples, and $\delta_{\theta,k}^{gyro}$ is a measure of the elementary rotation given by the gyro. To compute the odometer intervals ($[\delta_{RR,k}]$ and $[\delta_{RL,k}]$), we suppose

---

1. Initialization
   Set $k = 0$ and generate N boxes $\{x^{(i)}(k)\}_{i=1}^{N}$ with empty intersection and with same width and weights equal to $\frac{1}{N}$
2. *FOR* $i = 1 \cdots N$
3. Propagation or prediction
   $[x_{k+1}^i] = [f]([x_k^i],[u_k])$.
4. Measurement update
   - Predicted measurement: $[z_{k+1}^i] = [g]([x_{k+1}^i])$.
   - Innovation: $[r_{k+1}^i] = [z_{k+1}^i] \cap [y_{k+1}]$.
   - likelihood: $A^i = \prod_1^p A^i(j)$, where $A^i(j) = \frac{|[r_{k+1}^i(j)]|}{|[z_{k+1}^i(j)]|}$.
   - Box particle contraction: IF $[r_{k+1}^i] \neq \emptyset$, THEN, contract $[x_{k+1}^i]$ using $[r_{k+1}^i]$ and Waltz algorithm to obtain $[x_{k+1}^i]^{new}$, ELSE, $[x_{k+1}^i]^{new} = [x_{k+1}^i]$, ENDIF.
   - Weights update: $\omega_{k+1}^i = (\prod_1^p A^i(j))\omega_k^i = A^i\omega_k^i$

   ENDFOR.
5. Weights normalization
   FOR $i = 1 \cdots N$, $\omega_{k+1}^i \longleftarrow \frac{\omega_{k+1}^i}{\sum_{j=1}^{N}\omega_{k+1}^j}$, *ENDFOR*
6. State estimation
   $\hat{x}_k = \sum_{i=1}^{N}\omega_k^i C_k^i$. $\hat{P}_k = \sum_{i=1}^{N}\omega_k^i \frac{|[x_k^i]|}{2}$.
7. Resampling
   $N_{eff} = \frac{1}{\sum_{i=1}^{N}(\omega_k^i)^2}$. IF $N_{eff} < N_{th}$, THEN resample to create N new particle boxes with the same weights.
8. $k = k+1$, Goto 2 Until $k = k^{end}$

Fig. 1. BPF algorithm.

that the covered distance error between two instants t_{k-1} and t_k is less than the covered distance corresponding to one top of the ABS sensor counter (denoted δ_{ABS}) with the assumption that the vehicle rolls without slipping. For gyro interval measurement, thanks to specific static tests, we estimate the maximum of the error which is for the experiments $\delta_{\theta,k}^{gyro} = 3.10^{-3}$ degrees. The position and heading angle of the vehicle which is at time k, $[X_k] = [x_k] \times [y_k] \times [\theta_k]$ are calculated in time by using linear and angular velocities thanks to the following discrete representation:

$$\begin{cases} x_{k+1} = & x_k + \delta_{S,k}\cos(\theta_k + \frac{\delta_{\theta,k}}{2}) \\ y_{k+1} = & y_k + \delta_{S,k}\sin(\theta_k + \frac{\delta_{\theta,k}}{2}) \\ \theta_{k+1} = & \theta_k + \delta_{\theta,k} \end{cases} \quad (2)$$

where (x_k, y_k) and θ_k represent respectively the vehicle position and heading angle at time t_k. Note here that the width of each interval should guarantee maximum variation of the variables between two instants. The measurement of the position at time consists here in a Global Position System

(GPS) which is (x_{GPS}, y_{GPS}). The "longitude, latitude" estimated point of the GPS is converted in a Cartesian local frame and the GPS bounded error measurement is obtained thanks to the GST NMEA sentence [5]. The width of the GPS measure box can be quantified using the standard deviation σ_{GPS} estimated in real time by the GPS receiver (GST frame). Thus,

$$\begin{cases} [x_{GPS}] = & [x_{GPS} - 3\sigma_{GPS}, x_{GPS} + 3\sigma_{GPS}] \\ [y_{GPS}] = & [y_{GPS} - 3\sigma_{GPS}, y_{GPS} + 3\sigma_{GPS}] \end{cases} \quad (3)$$

The GPS measurement $([x_{GPS}], [y_{GPS}])$ is used to initialize the box state position $([x_1], [y_1])$ at instant t_1. Note that we haven't a direct measure of the heading angle, so the heading state of the vehicle should be initialized as $[\theta_1] = [-\infty, +\infty]$.

In order to be able to compute estimation errors, we have used a Thales Navigation GPS receiver used in a Post-Processed Kinematic mode working with a local base (a Trimble 7400). This system was able to give reference positions at 1 Hz sampling rate. Since the constellation of the satellites was good enough during all the trials, all the kinematics ambiguities were fixed. Therefore, a few centimeters accuracy was reached. The synchronization between this reference and the outputs of the dynamic localizers (BPF and PF) has been made thanks to the GPS timestamps. We have also taken into account the position offsets between the antennas of the two GPS receivers and the origin of the mobile frame.

Experiments have been carried out on a test track in Versailles (France) with our experimental vehicle. The data of the sensors have been time stamped and logged during several tests. We report hereafter the analysis of a 4.7 Km path with a mean speed of 50 Km/h using a 3GHz Pentium 4 and a Matlab implementation. The two filters provide outputs at the frequency of the GPS (5Hz). We use the available data in order to compare the BPF and the PF. This comparison is based simultaneously on the accuracy of the estimation and the guarantee. In addition, a comparison between the number of particles and box particles for the two algorithms will be given.

The resampling method used for the BPF consists on a subdivision strategy. The idea behind the subdivision resampling becomes from the fact that the manipulation of interval data gives always a very pessimistic solution caused by the basic rules of the interval arithmetics and the phenomenon of wrapping effect when propagate boxes via models [7]. Thus, in order to obtain a more selective and precise solution, one can divide the pessimistic box to several ones which give us the possibility to refine the solution for the next steps. Consequently, the idea consists first in sampling box particles according to their weights using for example a classical deterministic algorithm, and second in dividing each resampled box to several boxes with a number equals to the realizations of this box resulting from the resampling algorithm. This type of resampling refines the solution around regions with high likelihood and eliminates boxes with low weights. Nevertheless, as stated in section (4.6), one has to determine the number of divisions to do for each dimension. For example, for the state considered in the case of the model (2), which is a three variables state, the box particles will be in $\mathbb{R}^3$. If after the resample step we conclude that we have to divide a box particle to four sub-boxes, this will not be a straightforward job since we can do this by different manners. In our case, we suggest to give the preference to bisect boxes heading angle $[\theta]$ since we haven't a direct measure on this variable, but only the elementary rotation δ_θ of the mobile. This division is firstly performed until the width of the interval on θ constructing the resampled box is more than a fixed quantity (two degrees for example). For the choice between the subdivision of intervals on x or intervals on y, we give the preference to intervals with larger width.

Figure 2 shows the absolute error for GPS (bold black), BPF (solid blue) and PF (dashed green). As a conclusion, BPF and PF give equivalent filtering performances. This is also concluded from the mean square error given in Table 1. Nevertheless, for the BPF running, we use only 10 box particles comparing with 3000 particles for the PF. The few number of box particles explain the slow convergence of the BPF in the first seconds. The box particles number is very encouraging for using BPF since one can reduce significatively the particles number (for this application, the factor is about 300). Table 1 gives the mean of the running time of one step for each algorithm. Since the output frequency of each filter is 5 HZ, the running time for BPF satisfy real time constraints despite the use of interval arithmetic programs under Matlab and without code optimization. This is not the case of the PF. Figure 3 shows the interval error for x and y estimated for GPS (dashed black), BPF (bold black) and PF (solid blue). For PF, the interval error is calculated by using 3σ errors bounds around the point estimate. It can be seen that for this nonlinear problem, the two filters are consistent. Note that if the interval error contain "0" then the interval contains the PPK point.

Figure 4 plots the heading estimated error and the interval errors, in degrees, for BPF (bold black) and PF (solid blue). The errors on the heading estimation angles provided by BPF and PF are of the same magnitude. One can conclude that BPF is able to reconstruct a non directly measured variable. Note that the reference heading angle was built manually from the PPK measurements.

6. CONCLUSION AND FUTURE WORKS

A new algorithm for localization based simultaneously on particle filters and interval data has been proposed. The main idea consists on dealing with interval framework

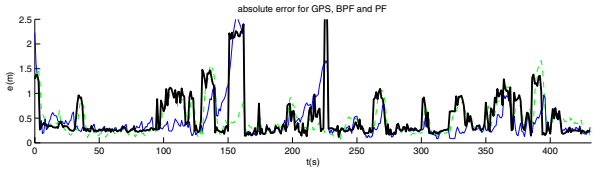

Fig. 2. Absolute error for GPS (bold black), BPF (solid blue) and PF (dashed green).

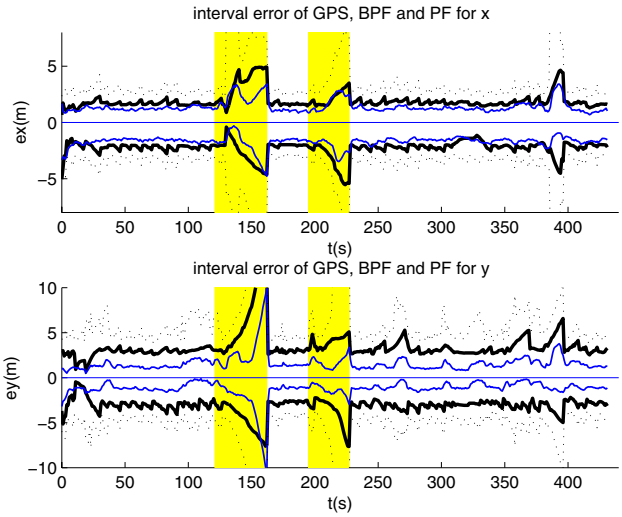

Fig. 3. Interval errors for x and y for GPS (dashed black), BPF (bold black) and PF (solid blue).

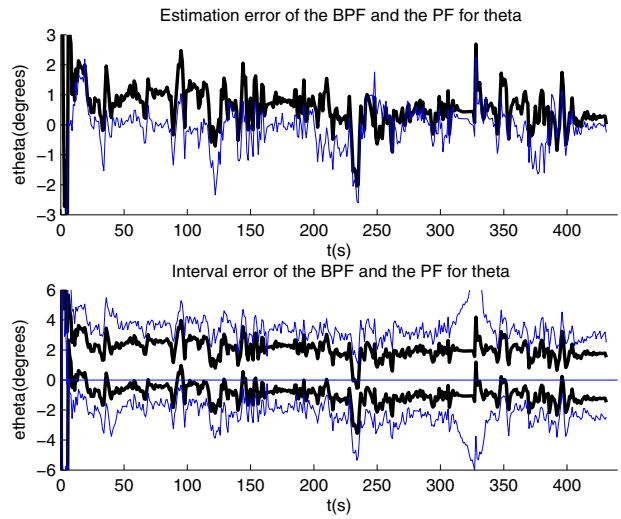

Fig. 4. Heading error and interval errors for BPF (bold black) and PF (solid blue).

	GPS	PF	BPF
mean square error for x	0.134	0.129	0.119
mean square error for y	0.374	0.217	0.242
particle number	-	3000	10
one step running time	-	666 ms	149 ms

Table 1. Comparison of PF and BPF.

which seems to be a good methodology to use with non-white and biased measurements. The experiments on real data show the feasibility and the effectiveness of the method compared with PF. In addition, the new algorithm seems to be well adapted to real time applications which is not the case for PF. One of the perspectives of this research is to create other resampling strategies by using properties of the interval framework. In our opinion, BPF is particularly adapted when using map data with rectangular roads. Indeed, in this case the boxes are adapted to calculate an intersection with the map. Thus, future works will consist on applying the BPF algorithm to Map Matching problems.

7. References

[1] Alspach, D. L. and Sorenson, H. W.. Nonlinear Bayesian Estimation Using Gaussian Sum Approximations.*IEEE Transactions on Automatic Control* 17(4), 439-448, 1972.

[2] Cendes, T. and Ratz, D. Subdivision direction selection in interval methods for global optimization. *SIAM Journal of numerical Analysis*, 34:922-938, 1997.

[3] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian fltering. *Statistics and computing*, 10:197-208, 2000.

[4] D. Fox. adapting the Sample Size in Particle Filters Through KLD-Sampling. *The international Journal of robotics research*, vol. 22, 12. pp. 985-1004, Dec 2003.

[5] A. Gning and Ph. Bonnifait. "Dynamic Vehicle Localization using Constraints Propagation Techniques on Intervals. A comparison with Kalman Filtering". *ICRA 05*. Barcelona, April 2005.

[6] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P. Nord-lund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, vol. 50, pp. 425- 435, Feb. 2002.

[7] Jaulin L., M. Kieffer, O. Didrit and E. Walter. *Applied Interval Analysis*. Springer-Verlag, 2001.

[8] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM. A factored solution to the simultaneous localization and mapping problem. *In AAAI*, 2002.

[9] Branko Ristic and Philippe Smets. Kalman filters for tracking and classification and the transferable belief model. *FUSION04*. Stockholm, Sweden, 2004.