Box Particle Filtering for Nonlinear State Estimation using Interval Analysis

Fahed Abdallah, Amadou Gning, Philippe Bonnifait

HEUDIASYC, UMR CNRS 6599
Université de Technologie de Compiègne
B.P. 20529, 60205 Compiègne Cedex, FRANCE

Abstract

In recent years particle filters have been applied to a variety of state estimation problems. A particle filter is a sequential Monte Carlo Bayesian estimator of the posterior density of the state using weighted particles. The efficiency and accuracy of the filter depend mostly on the number of particles used in the estimation and on the propagation function used to re-allocate weights to these particles at each iteration. If the imprecision, i.e. bias and noise, in the available information is high, the number of particles needs to be very large in order to obtain good performances. This may give rise to complexity problems for a real-time implementation. This kind of imprecision can easily be represented by interval data if the maximum error is known. Handling interval data is a new approach successfully applied to different real applications. In this paper, we propose an extension of the particle filter algorithm able to handle interval data and using interval analysis and constraint satisfaction techniques. In standard particle filtering, particles are punctual states associated with weights whose likelihoods are defined by a statistical model of the observation error. In the *box particle filter*, particles are boxes associated with weights whose likelihood is defined by a bounded model of the observation error. Experiments using actual data for global localization of a vehicle show the usefulness and the efficiency of the proposed approach.

Key words: State filtering and estimation; sensor fusion; particle filter; Kalman filter; interval analysis.

1 INTRODUCTION

In many application areas it is necessary to estimate the state of a dynamic system using a sequence of noisy sensor measurements. The Extended Kalman Filter (EKF) is used in sensor fusion for nonlinear systems [**?**]. This approach is based on applying the Kalman Filter (KF) algorithm on the linearization of the possibly nonlinear state and measurement functions of the state model using a first-order Taylor series expansion. The state distribution, or the posterior, can be then approximated by a Gaussian random variable which is propagated analytically through the first-order linearization of the nonlinear system. Instead of linearizing using Jacobian matrices, the Unscented Kalman Filter (UKF) uses a deterministic sampling strategy to capture the mean and covariance with a small set of carefully selected points known as *sigma points* [**?**]. Both the EKF and the UKF assume unimodal and Gaussian distributions. Recently, the particle filter (PF) has emerged as a useful tool for problems requiring dynamic state estimation. A particle filter is a sequential Monte Carlo Bayesian estimator which is expected to provide more valuable information about the posterior, especially if it has a multimodal shape or if noise distributions are non-Gaussian [**?**] [**?**] [**?**]. Nevertheless, particle filter methods suffer from some drawbacks. These methods are very sensitive to inconsistent measures or high measurement errors. In fact, the efficiency and accuracy of the filter depend mostly on the number of particles used in the estimation, and on the propagation function used to re-allocate weights to these particles at each iteration. If there is a high degree of imprecision as a result of bias and noise then the number of particles should be very large in order to explore a significant part of the state space. This will entail a level of complexity unsuitable for a real-time implementation. Several works like Unscented Particle Filters or Extended Kalman Particle Filter try to combine approaches in order to overcome these shortcomings (see for example [**?**] and references therein). Other works, like Particle Filters with Kullback-Leibler distance, use statistical approaches to increase the efficiency of particle filters by adapting the size of sample sets during the estimation process [**?**].

Email addresses: fahed.abdallah@hds.utc.fr (Fahed Abdallah), gningelh@hds.utc.fr (Amadou Gning), philippe.bonnifait@hds.utc.fr (Philippe Bonnifait).

In many applications, the interval framework seems to be a good methodology to deal with non-white and biased measurements. In this paper, we present a particle filter strategy for mobile localization involving interval data. Using interval data is more efficient in that it requires a significantly smaller set of particles than the normal PF algorithm, thereby reducing the computational cost. The pivotal notion in this work is that there are two different ways of looking at an interval in one dimension:

(1) An interval represents infinitely many particles continuously distributed throughout the interval.
(2) An interval represents a particle imprecisely located in the interval.

The article is organized as follows. Section **??** presents the principle of Bayesian filtering for nonlinear models. When certain constraints do not hold, this optimal solution is intractable. Sequential Monte Carlo methods, that is to say particle filters introduced in Section **??** are among several different strategies to approximate the optimal solution. In Section **??**, we briefly present interval analysis and we introduce some relevant interval operations usually used in the area of bounded error approaches. The main contribution of this paper is presented in Section **??**. We propose the Box Particle Filter (BPF) which involves handling box states, inputs and observations and using constraint satisfaction techniques. In Section **??**, we apply the proposed algorithm to a dynamic localization using a GPS receiver, a gyro and an odometer. Finally, in Section **??**, we present our conclusion regarding the BPF.

2 Bayesian filtering

2.1 Introduction

Consider the following nonlinear system:

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

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

From a Bayesian point of view, given the measurements Y_k, everything worth knowing about the state at time k is given by the conditional probability density $p(X_k|Y_k)$, known as the posterior density. The posterior constitutes the complete solution to the sequential estimation problem. In many real applications, only the filtering density $p(x_k|Y_k)$ which is a marginal of the posterior density $p(X_k|Y_k)$, needs to be estimated. The filtering density can be seen as a measure of the belief in the state x_k at time k, taking different values, given the measurements $Y_k = \{y_i, i = 1, \cdots, k\}$. Knowing $p(x_k|Y_k)$, various estimates of the system's state including means, modes and confidence intervals can easily be calculated.

2.2 Bayesian solution

If we consider additive noise sequences, the Bayesian solution to problem (**??**) is given by [**?**]:

$$\begin{cases} p(x_k|Y_k) = \frac{1}{\alpha_k} p_w(y_k - g(x_k)) p(x_k|Y_{k-1}) \\ p(x_{k+1}|Y_k) = \int p_v(x_{k+1} - f(x_k, u_k)) p(x_k|Y_k) dx_k \end{cases} \tag{2}$$

where α_k is a normalization factor given by $\alpha_k = \int p_w(y_k - g(x_k)) p(x_k|Y_{k-1}) dx_k$. The recursion has to be initialized with $p(x_0|Y_{-1}) = p(x_0)$, where $p(x_0)$ is a representation of prior knowledge, e.g. a uniform distribution over some region of the state space. The first line in (**??**) is known as the measurement update, and the second as the time update. In the case of linear, Gaussian environments, the Bayesian solution is the Kalman filter [**?**].

2.3 Particle filters

Statistical and stochastic techniques have been developed to estimate the state for systems whose parameters are distributed according to a probability distribution. Among these methods, sequential Monte Carlo Methods for Bayesian filtering, or particle filter methods, are the class of simulation filters which recursively approximate the filtering density $p(x_k|Y_k)$ as the cloud of N discrete *particles* with a probability mass, or weight, assigned to each of them. Hence, a possibly continuous probability density function is approximated by a discrete one.

2.4 Particle filtering schema

The outline of the particle filtering algorithm is as follows [**?**]. Initially, all particles have equal 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 dynamic model is essential for robust properties of the algorithm. Next, during the measurement step, new information that has become available about the system is used to adjust the

particle weights. The weight corresponds to the likelihood of each particle state describing the true current state of the system. This 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 following iteration by resampling them from the computed posterior probability distribution. At any time, certain 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 constraint propagation technique which is also known as *consistency technique* in the literature.

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: $[\mathbf{x}] = [x_1] \times [x_2] \cdots \times [x_n] = \times_{i=1}^{n_x} [x_i]$.

When working with intervals it is necessary to introduce the inclusion function $[f]$ of a function f, defined such that the image by $[f]$ of an interval $[x]$ is an interval $[f]([x])$ [**?**]. This function has to be calculated such that the interval enclosing the image set is optimal. Elementary arithmetical operations including +, -,* and /, and standard operations between sets of $\mathbb{R}^n$, e.g., $\subset$, $\supset$, $\cup$ and $\cap$, also need to be extended to the bounded error context.

Interval analysis has two main drawbacks, both related to the necessity of using interval arithmetic. The first is that standard computers are not equipped with dedicated interval-arithmetic capabilities, although many tools are well known to the interval community. The second is that the boxes are not particularly suitable for enclosing any set of solutions of $\mathbb{R}^n$. However, improvements have been made to reduce this limitation [**?**] [**?**] [**?**].

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 constraint propagation techniques [**?**], because of the the high degree of 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 equations of the form $f_j(x_1, \ldots, x_{n_x}) = 0, \quad j = 1 \ldots m$, which can be written more succinctly as $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, where $\mathbf{f}$ is the Cartesian product of the f_j's.

Definition 1 (Constraints Satisfaction Problem) *A Constraints Satisfaction Problem (CSP) $\mathcal{H}$ 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 $\mathbf{x}$.*

Under the interval framework, the imprecision or the uncertainty on the variable x_i of $\mathbb{R}$ is modelled by an interval $[x_i]$ which usually corresponds to prior knowledge about the domain of x_i. The a priori knowledge on the domain of $\mathbf{x}$ will then be extended to the region in $\mathbb{R}^{n_x}$ defined by: $[\mathbf{x}] = \times_{i=1}^{n_x} [x_i]$. The CSP consists of finding the values of $\mathbf{x} \in [\mathbf{x}]$ which satisfy the equality constraints $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. 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 necessarily a box. Under the interval framework, solving the CSP is equivalent to *finding the minimal box* $[\mathbf{x}'] \subset [\mathbf{x}]$ such that $\mathbf{S} \subset [\mathbf{x}']$.

3.3 Global and local Consistency

In the literature concerning CSPs, there are two key notions regarding the consistency of a solution. We now define these notions.

Definition 2 (Global consistency) *A scalar value x_i belonging to the i^{th} component of the box $[\mathbf{x}]$ is* globally consistent *with a CSP $\mathcal{H}$ if it is possible to find at least one vector within $[S]$ having x_i as the i^{th} coordinate. Thus x_i is* globally consistent *if* $\exists \{x_1 \in [x_1], \ldots, x_{i-1} \in [x_{i-1}], x_{i+1} \in [x_{i+1}], \ldots, x_{n_x} \in [x_{n_x}]\} / \mathbf{f}(\mathbf{x}) = \mathbf{0}$.

Definition 3 (Local consistency) *A scalar value x_i belonging to the i^{th} component of the box $[\mathbf{x}]$ is* locally consistent *with a CSP $\mathcal{H}$ if it is possible to find at least one vector within all $[S_j]_{j=1..m}$ taken separately having x_i as the i^{th} coordinate. Thus x_i is* locally consistent *if* $\forall\ f_j\ /\ j = 1 \ldots m,\ \exists \{x_1 \in [x_1],\ \ldots,\ x_{i-1} \in [x_{i-1}],\ x_{i+1} \in [x_{i+1}],\ \ldots,\ x_{n_x} \in [x_{n_x}]\} / f_j(\mathbf{x}) = 0$.

3.4 Waltz's Contractor

Definition 4 (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* $\mathbf{S} \subset [\mathbf{x}']$.

Different kinds of methods exist for developing contractors. The method used in this paper is an adaptation to real intervals of Waltz's algorithm [**?**] which is based on the primitive constraints propagation. A primitive constraint involves only an arithmetic operator or a standard function (cos, exp, etc.). The principle of the

Waltz's contractor is to contract each constraint, without any a priori order, until the contractor *becomes inefficient*. The use of this contractor appear to be especially efficient when there is redundancy of data and equations. In fact, this is the case for the data used in Section **??**. Note that this method is independent of the nonlinearities and provides locally-consistent contractors [**?**]. The principle can be explained using 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 $[a] = [0, +\infty[$. Each primitive constraint has been obtained by isolating one of the variables in the initial constraint. Consider an initial domain defined by $[z] = [0, 3]$, $[x] = [1, 7]$ and $[y] = [0, 1]$. Using the inclusion functions [exp] and $[(\exp)^{-1}] = [\ln]$, constraint propagation works as follows:

- $[a] = [a] \cap [\exp]([y]) = [1, e]$
- $[z] = [z] \cap [x].[a] = [1, 3]$
- $[x] = [x] \cap ([z]/[a]) = [1/e, 3]$
- $[a] = [a] \cap ([z]/[x]) = [1/3, 3e]$
- $[y] = [y] \cap [\ln]([a]) = [0, 1]$

and thus the new domain of the variables will be reduced to $[z] = [1, 3]$, $[x] = [1/e, 3]$ and $[y] = [0, 1]$.

4 From particle to box particle filters

When making real data measurements, different results are usually obtained when the same measurement is repeated. 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 would appear to be a good methodology for dealing with non-white, biased measurements, particularly when these measures vary around a central value within certain bounds. This approach will be used in this paper 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 in BPF is to use box particles and a bounded error model. Below we present the analogy between different steps in the particle filter algorithm and the corresponding steps in the box particle filter. Please notice that in the following sections *particles* refers to PF and *box particles* to BPF.

4.1 Box particle initialization

For the particle filter, this stage consists in generating a set of N point particles $\{x^{(i)}\}_{i=1}^{N}$ in a limited region of the state space. This region is chosen in order to explore the state space. Generally, all particles have identical weights attached to them. It is straightforward to extend this process of initialization to boxes. So, instead of point particles, the state space region in question can

S[c][c]State at time k-1 C[c][c]Constraints satisfaction
T[c][c]techniques B[c][c]Weighted Boxes W[c][c]after contraction
R[c][c]Resampling using weights D[c][c]Subdivision
P[c][c]Propagation (instant k) St[c][c]State at time k M[c][c]Measure

Fig. 1. Scenarios for the Box Particle Filter.

be divided into N boxes $\{[x^{(i)}]\}_{i=1}^{N}$ with empty intersection and equivalent weights can be associated with each of them. One advantage of this initialization using boxes is that the number of particles can be reduced.

4.2 Propagation or prediction step

In this step, the state of every particle is updated according to the evolution model using different realizations of the noise. Knowing the particles $\{x^{(i)}\}_{i=1}^{N}$ and the input $\{u_{(k)}\}$ at step k, the particles at step $k+1$ are built using the following propagation equation: $x_{k+1}^{i} = f(x_k^i, u_k, v_k^i)$, where v_k^i is the noise realization corresponding to the particle x_k^i. In the case of box particles, thanks to interval analysis tools, it is also possible to propagate boxes using the same propagation equations. 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 difference between operations on particles and operations on box particles in this step is the use of noise when propagating particles, and the bounded error form used for propagating box particles without noise. For particle filters, the noise allows us to take into account an error that may occur in the model and the measured inputs. Fortunately, in the case of box particles, the model and input errors are directly taken into account by the bounded error approach.

4.3 Measurement update

In this step, the new measurement is used to adujst the particle weights and contract the boxes.

4.3.1 Innovation

The innovation for particle filters consists in a quantity which depends on the difference between the real and the predicted measurements for each particle. Using the prediction $z_{k+1}^{i} = g(x_{k+1}^{i})$ and the real measurement y_{k+1}, the innovation corresponding to the i^{th} of the N particles will be $r_{k+1}^{i} = y_{k+1} - z_{k+1}^{i}$. The main difference when using box particles consists in predicting a value which can be compared to a real box measurement. The innovation should indicate the proximity between the real and the predicted values. Thus, in the bounded error framework, it can be evaluated as the intersection between the two boxes. 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 corresponds to the intersection with the real box measurement $[y_{k+1}]$. Thus, we calculate the innovation as $[r^i_{k+1}] = [z^i_{k+1}] \cap [y_{k+1}]$.

4.3.2 Likelihood

Using probabilistic models p_w for the measurement noise w, particle filters calculate the likelihood for each particle as: $p(y_{k+1}|x^i_{k+1}) = p_w(y_{k+1} - z^i_{k+1}) = p_w(r^i_{k+1})$. With the bounded error approach, it is quite obvious that a box particle for which the predicted box measurement has no intersection with the real box measurement should be penalized, and a box particle for which the predicted value is included in the real box measurement should be favored. This lead us to construct a measure of the *box likelihood* as $A^i = \prod_1^p A^i(j)$, where $A^i(j) = \frac{|[r^i_{k+1}(j)]|}{|[z^i_{k+1}(j)]|}$, p is the dimension of the measurement and $|[X]|$ is the width of $[X]$. In the following sections we will use only the word *likelihood* when we speak about the *box likelihood* quantity A.

4.3.3 Box particles contraction

This step, used only for box particles, does not feature in the particle filter algorithm. In fact, in the particle filter algorithm, each particle is propagated without any information about the variance of its position. Note that the weight of the particle provides no more than an indication about certainty when using this particle. In contrast, the width of each box particle after propagation is assumed to take into account the imprecision caused by model errors and input imprecisions. In order to conserve a judicious width for each box, contraction algorithms should be used in order to eliminate the non-consistent part of the box particle with respect to the box measurement [**?**], (see Figure **??**). This is in fact similar to the correction step of the Kalman filter when the variance-covariance matrix is corrected using the measurement [**?**]. Thus, if the innovation $[r^i_{k+1}]$ is not empty, the box particle $[x^i_{k+1}]$ is contracted using the intersection box $[r^i_{k+1}]$ and Waltz's algorithm to obtain a new box particle $[x^i_{k+1}]^{new}$. Else, $[x^i_{k+1}]^{new} = [x^i_{k+1}]$ and the box particle stays unchanged.

4.3.4 Weights update

In the particle filter case the weight for each particle is updated as $\omega^i_{k+1} = p(y_{k+1}|x^i_{k+1})\omega^i_k = p_w(r^i_{k+1})\omega^i_k$. In the same manner, one can construct an update to the weights of the box particle by multiplying the previous weight by each *box likelihood* as $\omega^i_{k+1} = (\prod_1^p A^i(j))\omega^i_k = A^i\omega^i_k$.

4.4 Normalization

This step is used in order to handle normalized weights so that their sum is equal to one: $\omega^i_{k+1} \longleftarrow \frac{\omega^i_{k+1}}{\sum_{j=1}^N \omega^j_{k+1}}$.

4.5 Estimation

At the *kth* step, the state is usually approximated empirically, using the weighted particles, as $\hat{x}_k = \sum_{i=1}^N \omega^i_k x^i_k$, [**?**] [**?**]. The confidence of this estimation is characterized by an estimation of the variance covariance matrix given by $\hat{P}_k = \sum_{i=1}^N \omega^i_k(\hat{x}_k - x^i_k)(\hat{x}_k - x^i_k)^T$, where A^T corresponds to the transpose of a matrix A. In the case of box particles, the state can be estimated as $\hat{x}_k = \sum_{i=1}^N \omega^i_k C^i_k$, where C^i_k is the center of the box particle i. One might also use a maximum weight estimate, i.e where the state estimate is the center of the box particle with the larger weight. A pessimistic confidence in the estimation will be a very well determined area consisting of a box containing all the possible weighted boxes. This can therefore be termed an *enclosing box*. Given that the BPF estimation $\hat{x}_k$ is calculated using N vectors C^i_k, another confidence in the estimation based on the confidence of each C^i_k can be calculated with $\hat{P}_k = \sum_{i=1}^N \omega^i_k P^i_k$, where P^i_k is the partial confidence generated when using each box particle center C^i_k. In practice, P^i_k can be taken as the half width of each box particle. Thus, $\hat{P}_k = \sum_{i=1}^N \omega^i_k \frac{|[x^i_k]|}{2}$.

4.6 Resampling

After some iterations, only a few box particles may be likely, and the rest may have weights close to or identical to zero. As for the resampling step in the particle filtering, the box particles are resampled 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 [**?**]. The problem with resampling is that the resulting samples are dependent since there is a high likelihood that the samples will be drawn from a small number of ancestors. In the case of the particle filter algorithm, instead of representing the smooth probability density as they should, particles will be clustered into groups. Therefore, some artificial noise should be added to the resampled particles in order to lessen the dependency. This step prevents the particle filter from breaking 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, small boxes around regions with high likelihoods, it is obvious that we can

divide each box by the corresponding number of realizations after sampling, i.e. if a box is resampled n times, we suggest to replace it by dividing it into n different subbox. Nevertheless, in the bounded error area, the choice of the number of divisions for each dimension remains a subject of research [**?**]. 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 introduced in [**?**] and is given by $N_{eff} = \frac{1}{\sum_{i=1}^{N}(\omega_k^i)^2}$. 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)
(2) 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 $\frac{1}{N}$
(3) *FOR* $i = 1 \cdots N$
(4) Propagation or prediction
$[x_{k+1}^i] = [f]([x_k^i], [u_k])$.
(5) 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.
(6) 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*
(7) State estimation
$\hat{x}_k = \sum_{i=1}^{N} \omega_k^i C_k^i$. $\hat{P}_k = \sum_{i=1}^{N} \omega_k^i \frac{|[x_k^i]|}{2}$.
(8) 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.
(9) $k = k + 1$, Goto 2 Until $k = k^{end}$

Fig. 2. BPF algorithm.

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

Let consider the localization problem of a land vehicle. The mobile frame origin is selected as the center of the rear axle. The elementary rotation and displacement between two samples can be obtained fairly precisely using only a fiber optic gyrometer and the two rear wheels' ABS sensors. Between two sampling instants, elementary rotations of the two rear wheels are integrated by counters. These values allow the distance travelled by the rear wheels between two samples to be calculated. Thus, at instant k, using the mobile frame, the elementary displacement covered by M, denoted $\delta_{S,k}$, and the elementary rotation, denoted $\delta_{\theta,k}$, are given by $\delta_{S,k} = \frac{\delta_{RR,k}+\delta_{RL,k}}{2}$ and $\delta_{\theta,k} = \delta_{\theta,k}^{gyro}$, where $\delta_{RR,k}$ and $\delta_{RL,k}$ denote the measured variables with values obtained 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 that the covered distance error between two instants t_{k-1} and t_k is less than the covered distance corresponding to one increment of the ABS sensor counter (denoted δ_{ABS}) assuming that the vehicle is not subject to slipping. For gyro interval measurement, thanks to specific static tests, we estimate the maximum of the error, which is $\delta_{\theta,k}^{gyro} = 3.10^{-3}$ degrees for our experiments. The position and heading angle of the vehicle which, at time k, is $[X_k] = [x_k] \times [y_k] \times [\theta_k]$ are calculated in time by using linear and angular velocities thanks to the following discrete representation:

$$\begin{cases} x_{k+1} = x_k + \delta_{S,k}\cos(\theta_k + \frac{\delta_{\theta,k}}{2}) \\ y_{k+1} = y_k + \delta_{S,k}\sin(\theta_k + \frac{\delta_{\theta,k}}{2}) \\ \theta_{k+1} = \theta_k + \delta_{\theta,k} \end{cases} \quad (3)$$

The measurement of the position at time k here makes use of a Global Position System (GPS) which is (x_{GPS}, y_{GPS}). The "longitude, latitude" estimated point of the GPS is converted to a Cartesian local frame and the GPS bounded error measurement is obtained thanks to the GST NMEA sentence [**?**]. The width of the GPS box measurement can be quantified using the standard deviation σ_{GPS} estimated in real time by the GPS receiver (GST frame). Thus, $[x_{GPS}] = [x_{GPS} - 3\sigma_{GPS}, x_{GPS} + 3\sigma_{GPS}]$ and $[y_{GPS}] = [y_{GPS} - 3\sigma_{GPS}, y_{GPS} + 3\sigma_{GPS}]$. The GPS measurement $([x_{GPS}], [y_{GPS}])$ is used to initialize the box state position $([x_1], [y_1])$ at time t_1. Note that we do not have a direct measurement 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 with a 1 Hz sampling rate. Since the constellation of the satellites was sufficiently good throughout the trials, all the kinematics ambiguities were fixed, and an accuracy to within a few centimeters was attained. The synchronization between this reference and the outputs of the dynamic localizers (BPF and PF) was obtained using the GPS timestamps. We also took into account the position offsets between the antennas of the two GPS receivers and the origin of the mobile frame.

Experiments were carried out on a test track in Versailles (France) with the Laboratory's experimental ve-

hicle. The results presented in this paper corresponds to a drive which lasted about four minutes. The data from the sensors were time stamped and logged during several tests. Below we present the analysis of a 4.7-kilometer trajectory 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).

In this section, we use the available data in order to compare the BPF and the PF. We compare both the accuracy of the estimation and the guarantee. In addition, we make a comparison between the calculation time and the number of particles and box particles for the two algorithms.

For the resampling step in the particle filter method, a basic deterministic algorithm was used [**?**]. As stated in section (**??**), by taking into account the specificities of interval data, different strategies may be used for the resampling step in the case of BPF. In this paper, we will present a strategy, named *subdivision resampling* that appears to us to be more efficient than others. A comparison between PF and BPF, based on the actual experimental data, will be given.

The idea behind the subdivision resampling derives from the fact that the manipulation of interval data always yields a very pessimistic solution as a result of the basic rules of the interval arithmetic and the wrapping effect when boxes are propagated via models [**?**]. Thus, in order to obtain a more selective and precise solution, one can divide the pessimistic box into several sub-boxes, thus making it possible to refine the solution for the following steps. Consequently, the idea consists first in sampling the box particles according to their weights using for example a classical deterministic algorithm, and secondly in dividing each resampled box into as many sub-boxes as there are realizations resulting from the resampling algorithm. This type of resampling allows us to refine the solution around regions with high likelihood and to eliminate boxes with low weights. Nevertheless, as stated in section (**??**), one needs to determine the number of divisions for each dimension. For example, for the state considered in the case of the model (**??**), which is a three-variable state, the box particles will be in $\mathbb{R}^3$. If after the resampling step we conclude that we have to divide a box particle into four sub-boxes, this will not be a straightforward exercise since there are different ways of doing it. In our case, we suggest that is preferable to bisect boxes' heading angles $[\theta]$ since we do not have a direct measurement of this variable, but only the elementary rotation δ_θ of the mobile. This division is performed until the width of the interval on θ of the resampled box is less than a fixed quantity (two degrees for example). For the choice between the subdivision of intervals on x or intervals on y, we favor intervals which are wider.

Table **??** shows the mean square errors on x and y for GPS, PF and BPF. As a conclusion, the BPF and the PF give equivalent filtering performances. Nevertheless, for the running BPF, we use only 10 box particles comparing with 3000 particles for the PF. The small number of box particles explains the slow convergence of the BPF in the first seconds. The number of box particles is very encouraging inasmuch as the number of particles can be significantly reduced (for this application, the factor is about 300). Table **??** gives also the mean of the execution time of one step for each algorithm. Since the output frequency of each filter is 5 HZ, the execution time for BPF satisfies real time constraints despite the use of interval arithmetic programs under Matlab and without code optimization. This is not the case with PF. Figure **??** shows the interval error for x and y estimated for GPS (dashed black), BPF (bold black) and PF (solid blue). The colored rectangles (yellow) indicate the DGPS corrections lost. 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 an interval error that contains "0" indicates that this interval contains the PPK's point.

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

Table 1

Comparison of PF and BPF. The table shows the mean square error for GPS, PF and BPF. The particle and box particle numbers are given for PF and BPF. We also give the mean of the running time of one step for each algorithm.

Figure **??** plots the estimated heading error and the interval errors, in degrees, for BPF (bold black) and PF (solid blue). The errors on the heading estimation angles provided by the BPF and the PF are of the same magnitude. One can conclude that the BPF is able to reconstruct a non-directly measured variable. Note that the reference heading angle was built manually from the PPK measurements. A movie illustrating the behavior of the filter is accessible on line at the address http://www.hds.utc.fr/ bonnif/mov/.

Fig. 3. The figures show the interval error for x and y estimated for GPS (dashed black), BPF (bold black) and PF (solid blue).

Fig. 4. The figures show the estimated heading error and the interval errors, in degrees, for BPF (bold black) and PF (solid blue).

6 CONCLUSIONS AND FUTURE WORKS

A new algorithm for localization based simultaneously on particle filters and interval data has been proposed. The method is based on the interval framework which seems to be a good methodology to use in the case of non-white and biased measurements. The experiments on real data show the feasibility and the effectiveness of the method compared to PF. In addition, the new algorithm seems to be well suited to real time applications, which is not the case for the particle filter algorithm as shown in Table **??**. As part of future work, we intend to study other resampling strategies using properties of the interval framework. We also plan to compare this approach to other PF alternatives which try to reduce the number of particles. Another perspective of this research is to adapt the BPF for Map Matching problems which use map data with rectangular roads. Indeed, in this case, the boxes are adapted to calculate intersections with the map and to manage the multiple hypotheses due to junctions or parallel roads.

References

[1] Martin Adams, Sen Zhang and Lihua Xie. Particle Filter Based Outdoor Robot Localization Using Natural Features Extracted from Laser Scanners. *ICRA 04*. New Orleans, April 2004.

[2] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. on Signal Processing*, Vol.50, No.2, pp 174-188, 2002.

[3] F. Benhamou, F. Goualard, L. Granvilliers, and J.F. Puget. Revising hull and box consistency. *Proceedings of the International Conference on Logic Programming*, pp.230-244, Las Cruces, NM, 1999.

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

[5] Wen-Shiang Chen. *Bayesian estimation by sequential Monte Carlo sampling*. Dissertation, The Ohio State University, 2004.

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

[7] 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.

[8] A. Gning, Ph. Bonnifait. "Dynamic Vehicle Localization using Constraints Propagation Techniques on Intervals. A comparison with Kalman Filtering". *IEEE International Conference on Robotics and Automation (ICRA 05)*. Barcelona, April 2005.

[9] A. Gning, Ph. Bonnifait. Constraints Propagation Techniques on Intervals for a Guaranteed Localization using Redundant Data. *Automatica*. Volume 42, Issue 7, pages 1167-1175, July 2006.

[10] 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.

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

[12] Jaulin L., M. Kieffer, O. Didrit and E. Walter. Guaranteed nonlinear estimation using constraint propagation on sets. *International Journal of Control*. volume 74, no 18, 1772-1782, 2001.

[13] L. Jaulin. Nonlinear bounded-error state estimation of continuous-time systems. *Automatica*, 38, 1079-1082, 2002.

[14] L. Jaulin. Computing minimal-volume credible sets using interval analysis; Application to Bayesian estimation. *IEEE Trans. on Signal Processing,* pages 3632-3636. Volume 54, Issue 9, 2006.

[15] A. Kong, J. S. Liu, and W. H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

[16] R. van der Merwe, J. F. G. de Freitas, A. Doucet, and E. A. Wan. *The Unscented Particle Filter*. Technical report, Dept. of Engineering, University of Cambridge, 2000.

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

[18] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge Univ. Press, Cambridge, 1990.

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

[20] Sebastian Thrun, Wolfram Burgard and Dieter Fox. *Probabilistic Robotics*, The MIT press, September 2005.

[21] D. Waltz. "Generating semantic descriptions from drawings of scenes with shadows", *The psychology of computer Vision*, New York, NY pp. 19-91, 1975.