

State estimation using interval analysis and belief function theory: Application to dynamic vehicle localization

Ghalia Nassreddine, Fahed Abdallah and Thierry Denœux
UMR CNRS 6599 Heudiasyc
Université de Technologie de Compiègne
BP 20529 - F-60205 Compiègne cedex - France

October 26, 2009

Abstract

A new approach to non-linear state estimation based on belief function theory and interval analysis is presented. This method uses belief structures composed of a finite number of axis-aligned boxes with associated masses. Such belief structures can represent partial information on model and measurement uncertainties, more accurately than can the bounded error approach alone. Focal sets are propagated in system equations using interval arithmetics and constraint satisfaction techniques, thus generalizing pure interval analysis. This model was used to locate a land vehicle using a dynamic fusion of GPS measurements with dead reckoning sensors. The method has been shown to provide more accurate estimates of vehicle position than does the bounded error method while retaining what is essential: providing guaranteed computations. The performances of our method were also slightly better than those of a particle filter, with comparable running time. These results suggest that our method is a viable alternative to both bounded error and probabilistic Monte-Carlo approaches for vehicle localization applications.

Keywords: Data fusion, State Estimation, Dempster-Shafer theory, Evidence theory, Interval analysis, Bounded error estimation, Localization.

1 Introduction

Estimating the state of a dynamical system based on noisy sensor measurements is a common problem in various areas such as mobile robot navigation or target tracking [2][23]. If several sensors or other information sources are available, then data fusion algorithms have to be used. A critical issue when designing such systems is the representation of uncertainties that pervade both sensor measurements and the state space model itself. Two broad categories of state estimation methods can be distinguished in that respect.

Methods in the first category are based on a *probabilistic description of uncertainty* and assume measurement noise and state perturbations to be realizations of random variables with known statistical properties. This is the mainstream approach. In a linear context, the most popular method is by far the application of the Kalman filter [27], which assumes model and observation noise to be normal. In non-linear cases, the two main methods are based on the Extended Kalman filter (EKF) and the Particle filter (PF). The EKF linearizes the state and measurement equations and then applies the Kalman filter to obtain state estimates [4, 20], assuming process and observation noises to be normal. The state posterior probability distribution is approximated using a Gaussian distribution that is propagated analytically through linearized system equations. However, linearization is inherently local and may fail to produce reliable estimates, especially when the state model is highly non-linear. The statistical interpretation of covariance matrices is also unclear in this approach, as the statistical properties of perturbations though non-linear equations are not well-known. Recently, sequential Monte Carlo methods for recursive Bayesian filtering, or particle filter (PF) methods have emerged as useful tools for problems requiring non-linear dynamic state estimation [7][15][23]. The PF estimator usually provides more accurate information about the state posterior probability distribution than does the EKF, especially if it takes a multimodal shape or if noise distributions are non-Gaussian. The efficiency and accuracy of the PF method depends mostly on the number of particles used in the estimation and on the propagation used to re-allocate weights to these particles at each iteration. To cope with high uncertainty of measurement, a large number of particles have to be used, especially when the number of dimensions of the state vector is high; this may be an issue for real-time implementation of the PF method. Several authors have tried to overcome these shortcomings by combining approaches (see [23] and references therein) or by adapting the size of sample sets during the estimation process [19].

The second group of methods corresponds to *state bounding*. Assuming that all variables belong to known compact sets, these methods attempt to build simple sets, such as ellipsoids or boxes, guaranteed to contain all state vectors consistent with given constraints [26]. In a linear context, this approach has been investigated by several authors since the 1960's (see, e.g., [6, 8, 29, 32, 37, 38, 43] and references therein). In the non-linear context, the methodology is less developed. Non-linear state-bounding methods have been presented in [3][24][26][28]. Recently, a relatively simple and fast method based on constraint propagation and interval analysis was introduced and applied to vehicle localization by [21][22]. The main interest in the bounded-error approach arises from the fact that it allows so-called validated computations: computations with guaranteed accuracy taking into account all possible sources of error,

from data imprecision to rounding errors due to computer calculations. The major drawback of this approach is the difficulty of determining noise bounds. Indeed, if the bounds are too tight, the data may become inconsistent with the system equations, in which case, the method fails to provide a solution. On the other hand, if the bounds are overestimated, the estimated state becomes very imprecise, and the method becomes overly pessimistic. A related problem with this approach is its lack of robustness against outlying observations, which may be a severe limitation in cases where sensor information is unreliable.

In this paper, we propose to replace the set-based representation of uncertainty used in the state bounding approach by a more general formalism based on belief functions. The theory of belief functions, also known as Dempster-Shafer (DS) or Evidence theory, was introduced by Dempster[11] in statistical inference, and was further developed by Shafer [39] and Smets [40] as a theory of epistemic uncertainty. The theory of belief functions makes it possible to model various states of knowledge, ranging from complete ignorance to probabilistic uncertainty [46]. A belief function may be viewed both as a generalized set [17] and as a generalized probability measure: DS theory thus encompasses set-based as well as probabilistic formalisms.

Until now, the application of belief functions to state estimation has been limited. Dempster [12] showed that computations in the Kalman filter can be interpreted in a graphical model in terms of the propagation of Gaussian belief functions (a special kind of belief function characterized by a normal density over linear subspaces). Ristic and Smets [41] have provided another interpretation of the Kalman filter under a belief function paradigm and have presented an approach to joint tracking and classification based on continuous belief functions. Mahler [31] has also discussed various extensions of the Kalman filter that make it possible to handle uncertainties expressed by belief functions. The main motivations of this study have been to cope with situations where the available information about the system under study or about sensor measurements is too weak to be easily handled using a purely probabilistic approach. However, these situations have been confined mostly to linear models; this may be explained by the fact that the manipulation of continuous belief functions is delicate and quickly becomes intractable when applied to more complex models.

In this paper, we propose a different approach — one that generalizes the bounding approach to state estimation. Taking as a starting point the bounded error state estimation method proposed in [22], we present a data filtering and fusion method that uses Dempster-Shafer mass functions to assign belief masses to a *finite* number of focal sets, taking the form of axis-aligned boxes. Such mass functions can be seen as “generalized boxes” composed of a collection of boxes with associated weights. Focal sets are propagated in the system equations using interval arithmetics and constraint satisfaction techniques [26]. Applied to state estimation in dynamical systems, this approach allows us to compute a belief function for the system state at each time step. This approach extends the pure interval approach, making it more robust and accurate, while retaining the validity of computations.

The article is organized as follows. Section 2 first presents the background on interval analysis and bounded error state estimation. The necessary concepts in belief function theory are then recalled in Section 3. Our approach is introduced in Section 4, and in Section 5 we apply our method to dynamic land vehicle localization using GPS, gyrometer and odometer measurements. Finally, we conclude and discuss the main

contributions of the paper in Section 6.

2 Interval Analysis

In this subsection, we briefly recall the main concepts pertaining to interval analysis and constraint satisfaction techniques that will be used in the rest of this work.

2.1 Basic Concepts

Usually, interval analysis is used to model quantities that vary around a central value within certain bounds [26]. A (closed) real interval, denoted $[x]$, is defined as a closed and connected subset of \mathbb{R} :

$$[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}, \quad (1)$$

where \underline{x} and \bar{x} are the minimal and maximal bounds of $[x]$. Set theory operations, such as intersection or union, can be applied to intervals. The intersection of two intervals is always an interval, whereas the union is not. The *interval union* of two intervals $[x]$ and $[y]$ can be defined as the *interval hull* of $[x] \cup [y]$, i.e., the smallest interval containing $[x] \cup [y]$, which is denoted as follows [26]:

$$[x] \sqcup [y] = [[x] \cup [y]]. \quad (2)$$

The four classical arithmetic operations can be extended to intervals. For any such binary operator \diamond , the interval $[x] \diamond [y]$ is defined for any intervals $[x]$ and $[y]$ as [26]

$$[x] \diamond [y] = [\{x \diamond y \in \mathbb{R} \mid x \in [x], y \in [y]\}]. \quad (3)$$

When \diamond is continuous, as is the case for the arithmetic operations, the set $\{x \diamond y \in \mathbb{R} \mid x \in [x], y \in [y]\}$ is an interval; we thus have the following equivalent definition for $[x] \diamond [y]$:

$$[x] \diamond [y] = \{x \diamond y \in \mathbb{R} \mid x \in [x], y \in [y]\}. \quad (4)$$

When \diamond is one of the four arithmetic operations, then the bounds of $[x] \diamond [y]$ can be computed from the bounds of $[x]$ and $[y]$ using simple formulas. The following equations hold [26][33]:

$$[x] + [y] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}] \quad (5)$$

$$[x] - [y] = [\underline{x} - \bar{y}, \bar{x} - \underline{y}] \quad (6)$$

$$[x] \cdot [y] = [\min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}), \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y})]. \quad (7)$$

If the bounds of $[x]$ and $[y]$ are positive, we also have

$$[x]/[y] = [\underline{x}/\bar{y}, \bar{x}/\underline{y}]. \quad (8)$$

Elementary functions such as \exp , \ln , \sin , etc., can also be extended to intervals. If f is a function from \mathbb{R} to \mathbb{R} , its interval counterpart $[f]$ is defined by:

$$[f]([x]) = [\{f(x) \mid x \in [x]\}]. \quad (9)$$

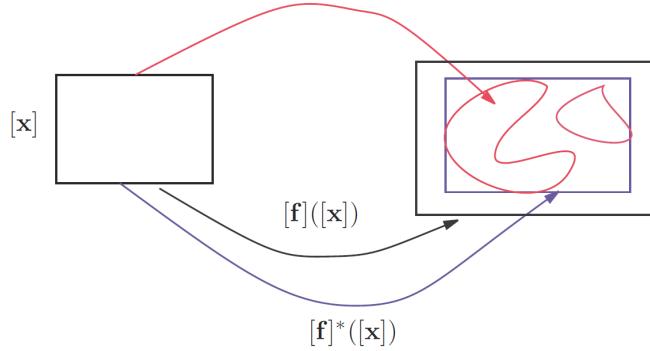


Figure 1: Images of a box $[x]$ by a function f , a pessimistic inclusion function $[f]$ and the minimal inclusion function $[f]^*$.

Again, if f is continuous, then $[f]([x])$ is simply equal to $f([x])$. For instance,

$$[\exp]([x]) = [\exp(\underline{x}), \exp(\bar{x})]. \quad (10)$$

For non-monotonic functions, however, the computation of $[f]([x])$ is usually not as simple.

An interval real vector $[x]$ of \mathbb{R}^n , also called a *box*, is defined as a Cartesian product of n intervals:

$$[x] = [x_1] \times \dots \times [x_n]. \quad (11)$$

Geometrically, it may be figured as an n dimensional, axis-aligned parallelepiped. The set of n -dimensional interval real vectors will be denoted \mathbb{IR}^n . The notions recalled above may be easily extended to boxes. For instance, the interval hull of a subset A of \mathbb{R}^n is the smallest box of \mathbb{R}^n containing A . It is noted $[A]$. The interval union of two boxes $[x]$ and $[y]$ may be computed as:

$$[x] \sqcup [y] = ([x_1] \sqcup [y_1]) \times \dots \times ([x_n] \sqcup [y_n]). \quad (12)$$

Classical operations for real vectors can be extended to interval vectors. For instance,

$$[x] + [y] = ([x_1] + [y_1]) \times \dots \times ([x_n] + [y_n]). \quad (13)$$

Let f be a function from \mathbb{R}^n to \mathbb{R}^m . It is often useful to compute the image $f([x])$ of a box $[x]$ by f . This image is usually not a box (see Figure 1), and its expression may be difficult to obtain. However, it may sometimes be approximated by an *inclusion function*. An interval function $[f]$ from \mathbb{IR}^n to \mathbb{IR}^m is said to be an *inclusion function* for f if

$$f([x]) \subseteq [f]([x]), \quad \forall [x] \in \mathbb{IR}^n. \quad (14)$$

Inclusion functions may be very pessimistic, as shown by Figure 1. The inclusion function $[f]$ is minimal if, for any x , $[f]([x])$ is the interval hull of $f([x])$. The minimal inclusion function for f is unique and may be noted as $[f]^*$.

Finding inclusion functions that can be evaluated reasonably quickly and such that $[f]([x])$ is not too large for most x is one of the main purposes of interval analysis [26]. Different algorithms have been proposed to reduce the size of boxes enclosing $f([x])$. For the fusion problem considered here, we have chosen to use constraint propagation techniques [26], because of the great redundancy of data and equations.

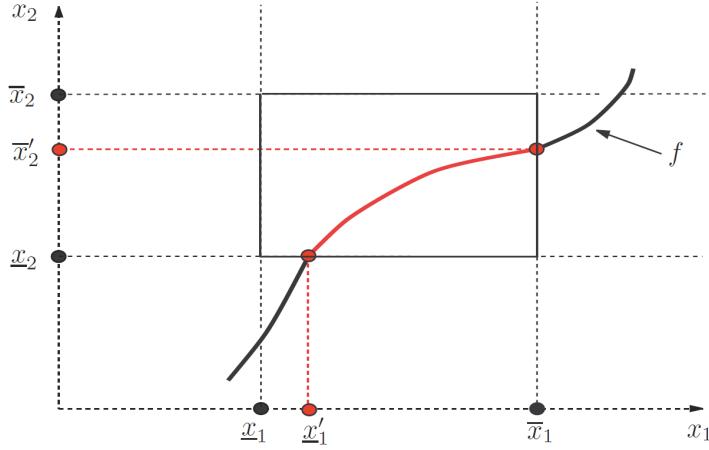


Figure 2: A CSP in \mathbb{R}^2 .

2.2 Constraint Satisfaction Problems

Consider n variables $x_i, i \in \{1, \dots, n\}$ linked by m relations (or constraints) of the form

$$f_j(x_1, \dots, x_n) = 0, \quad j = 1, \dots, m. \quad (15)$$

If we denote by \mathbf{x} the vector $(x_1, x_2, \dots, x_n)^T$ and by \mathbf{f} the function whose coordinate functions are the f_i s: $\mathbf{f} = (f_1, f_2, \dots, f_m)^T$, we can write these m constraints in vector notation as

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}. \quad (16)$$

Let us assume that vector \mathbf{x} is known to belong to some prior domain $[\mathbf{x}]$, and we want to compute the set of all \mathbf{x} in the prior domain verifying the constraints. This defines a constraint satisfaction problem (CSP), which can be denoted as

$$\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}]). \quad (17)$$

The solution set of \mathcal{H} is defined as:

$$\mathbf{S} = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{f}(\mathbf{x}) = \mathbf{0}\}. \quad (18)$$

Note that \mathbf{S} is not necessarily a box. Using interval methods, solving the CSP implies finding a box $[\mathbf{x}]'$ that constitutes an outer approximation of S , i.e., such that $\mathbf{S} \subseteq [\mathbf{x}]' \subseteq [\mathbf{x}]$. Figure 2 illustrates a simple CSP with two variables and a single constraint.

Contracting \mathcal{H} means replacing $[\mathbf{x}]$ by a smaller domain $[\mathbf{x}]'$ such that $\mathbf{S} \subseteq [\mathbf{x}]' \subseteq [\mathbf{x}]$. A *contractor* \mathcal{C} for \mathcal{H} is any operator that can be used to contract \mathcal{H} . Several methods for building contractors are described in [26, Chapter 4], including Gauss elimination, the Gauss-Seidel algorithm and linear programming, among others. Each of these methods may be more suitable for some types of CSP. Although the approach introduced in this paper is not bound to any particular contractor, we will focus on the forward-backward propagation (FBP) technique [42][9][10][26][25]. This method was chosen because it is known to be especially efficient in cases of high redundancy of data and equations — a situation that will be encountered in the application described

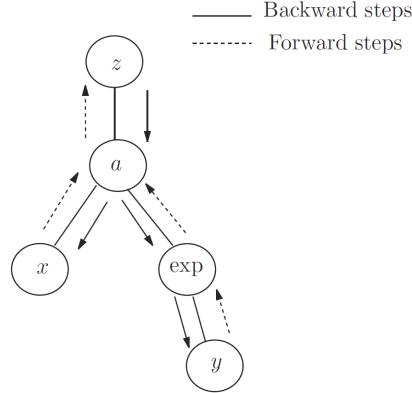


Figure 3: Graph corresponding to the CSP of Example 1.

in Section 5. Furthermore, this method can handle any kind of non-linearity and its implementation is quite simple. However, other contractors could be used as well.

The FBP contractor is based on the propagation of *primitive constraints* for real variables. A constraint is said to be primitive if it involves a single operator (such as $+$, $-$, $*$ or \backslash) or a single function (such as \cos , \sin or \sinh). The method proceeds by contracting \mathcal{H} with respect to each of the primitive constraints until convergence on a minimal domain. The complete description of the FBP algorithm goes beyond the scope of this paper. It can be found in [26, page 77]. Here, it will simply be described through an example.

EXAMPLE 1 Let us consider the constraint $z = x \exp(y)$ and the initial domain $[z] = [0, 3]$, $[x] = [1, 7]$ and $[y] = [0, 1]$. The constraint is not primitive, as it involves one arithmetic operation and one elementary function. However, it can be decomposed into two primitive constraints by introducing an auxiliary variable a :

$$\begin{cases} a = \exp(y) \\ z = a \cdot x. \end{cases} \quad (19)$$

The domain of a may be initialized by $[a] = [0, +\infty)$. The FBP algorithm alternates between two phases, called forward and backward propagation.

In the forward propagation phase, the domains of a and z are contracted using (19). This is achieved in two steps F_1 and F_2 :

- $F_1: [a] \leftarrow [a] \cap [\exp]([y]) = [0, +\infty) \cap [1, e] = [1, e]$
- $F_2: [z] \leftarrow [z] \cap [x] \cdot [a] = [0, 3] \cap [1, 7] \cdot [1, e] = [1, 3]$.

Then, backward propagation is performed, updating the domains associated with all variables. This is achieved in three steps B_3 , B_4 and B_5 :

- $B_3: [x] \leftarrow [x] \cap ([z]/[a]) = [1, 7] \cap [1, 3]/[1, e] = [1, 3]$
- $B_4: [a] \leftarrow [a] \cap ([z]/[x]) = [1, e] \cap [1, 3]/[1, 3] = [1, e]$
- $B_5: [y] \leftarrow [y] \cap [\ln]([a]) = [0, 1] \cap [0, 1] = [0, 1]$.

After one forward-backward propagation cycle, the domains of the variables have thus been reduced to $[z] = [1, 3]$, $[x] = [1, 3]$ and $[y] = [0, 1]$. It can be verified that these domains are no longer changed after another iteration of the procedure. Figure 3 shows a graphical representation of the problem, where each node corresponds to a variable; there is a link between two nodes if the corresponding variables are linked by a primitive constraint. It may be shown that the FPB contractor yields the smallest possible domain if this graph contains no cycle [25].

2.3 Bounded Error Estimation

In this section, we briefly present the bounded error state estimation method (BEE) introduced in [21][22]. Consider a non-linear dynamical system represented in state space by the following discrete-time equations:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \quad (20)$$

$$\mathbf{z}_{k+1} = \mathbf{g}(\mathbf{x}_{k+1}), \quad (21)$$

where $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$ is a possibly non-linear function that relates the state \mathbf{x}_{k+1} at time $k+1$ to the previous state \mathbf{x}_k at time k with the input \mathbf{u}_k . Function $\mathbf{g} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \rightarrow \mathbb{R}^{n_z}$ expresses the relationship between the observed output \mathbf{z}_{k+1} and the state \mathbf{x}_{k+1} . Note that we could easily consider a more complex model including, for instance, additional state and measurement perturbation variables. However, the simple model above will be sufficient to illustrate this approach, as well as for the application we develop in Section 5.

When working with bounded errors, the initial state \mathbf{x}_0 is assumed to belong to some prior box $[\mathbf{x}_0] \subset \mathbb{R}^{n_x}$ and, at each time step k , the input \mathbf{u}_k and the measurement vector \mathbf{z}_k are assumed to belong to boxes $[\mathbf{u}_k]$ and $[\mathbf{z}_k]$, respectively. The problem considered in [22] is to provide at each time step $k+1$ a *guaranteed* estimate of the state \mathbf{x}_{k+1} , in the form of a box $[\mathbf{x}_{k+1}]$ that *certainly* contains the true state vector \mathbf{x}_{k+1} , assuming the system equations (20)-(21) to be correct and taking into account all the information available at time step k . This problem can be seen as a CSP

$$\mathcal{H}_{0:k+1} : (\mathbf{F}(\boldsymbol{\xi}_{0:k+1}) = 0, \boldsymbol{\xi}_{0:k+1} \in [\boldsymbol{\xi}_{0:k+1}]), \quad (22)$$

with $\boldsymbol{\xi}_{0:k+1} = (\mathbf{x}_{k+1}, \dots, \mathbf{x}_0, \mathbf{u}_k, \dots, \mathbf{u}_0, \mathbf{z}_{k+1}, \dots, \mathbf{z}_1)$ and

$$\mathbf{F} : \boldsymbol{\xi}_{0:k+1} \rightarrow (\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \dots, \mathbf{x}_1 - \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0), \mathbf{z}_{k+1} - \mathbf{g}(\mathbf{x}_{k+1}), \dots, \mathbf{z}_1 - \mathbf{g}(\mathbf{x}_1)). \quad (23)$$

The FBP contractor described in the previous section can be applied to this problem after all the constraints in $\mathcal{H}_{0:k+1}$ have been decomposed into primitive ones. The forward and backward propagations are then iterated until convergence.

In practice, however, considering all equations from step 0 to $k+1$ is often not feasible in real-time, as noted in [22]. This problem can be solved by considering only variables and constraints within a limited time window. In the simplest (and fastest) variation considered here, we start from the previous state estimate $[\mathbf{x}_k]$ at step k , and we determine $[\mathbf{x}_{k+1}]$ from $[\mathbf{x}_k]$, $[\mathbf{u}_k]$, $[\mathbf{z}_{k+1}]$ and the constraints (20)-(21) between time steps k and $k+1$. Formally, this corresponds to the following CSP:

$$\mathcal{H}_{k+1} : (\mathbf{F}(\boldsymbol{\xi}_{k+1}) = 0, \boldsymbol{\xi}_{k+1} \in [\boldsymbol{\xi}_{k+1}]), \quad (24)$$

with $\xi_{k+1} = (\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k, \mathbf{z}_{k+1})$ and

$$\mathbf{F} : \xi_{k+1} \rightarrow (\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \mathbf{z}_{k+1} - \mathbf{g}(\mathbf{x}_{k+1})). \quad (25)$$

The method is summarized in Algorithm 1.

Algorithm 1 Bounded Error Estimation (BEE) algorithm

Input: $[\mathbf{x}_k], [\mathbf{u}_k], [\mathbf{z}_{k+1}]$

Output: $[\mathbf{x}_{k+1}]$

- 1: Read the input \mathbf{u}_k and its error. Deduce $[\mathbf{u}_k]$.
 - 2: Read the output \mathbf{z}_{k+1} and its error. Deduce $[\mathbf{z}_{k+1}]$.
 - 3: $[\mathbf{x}_{k+1}] \leftarrow (-\infty, +\infty)^n_x$
 - 4: **while** at least one variable is contracted **do**
 - 5: % Forward propagation %
 - 6: $[\mathbf{x}_{k+1}] \leftarrow [\mathbf{x}_{k+1}] \cap [\mathbf{f}]([\mathbf{x}_k], [\mathbf{u}_k])$
 - 7: $[\mathbf{z}_{k+1}] \leftarrow [\mathbf{z}_{k+1}] \cap [\mathbf{g}]([\mathbf{x}_{k+1}])$
 - 8: % Backward propagation %
 - 9: Back-propagate the constraint $\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = 0$
 - 10: Back-propagate the constraint $\mathbf{z}_{k+1} - \mathbf{g}(\mathbf{x}_{k+1}) = 0$
 - 11: **end while**
-

In steps 1 and 2, intervals are determined around the input \mathbf{u}_k and the output \mathbf{z}_{k+1} using error bounds. These bounds depend on the sensors used and are assumed to be known. As noted in [22], they may sometimes be determined from constructor data sheets, or they may be derived from estimated imprecision provided by the sensors in real time.

Initially, nothing is known about \mathbf{x}_{k+1} : $[\mathbf{x}_{k+1}]$ is thus initialized as the largest possible box in step 3. Boxes $[\mathbf{x}_{k+1}], [\mathbf{x}_k], [\mathbf{u}_k]$ and $[\mathbf{z}_{k+1}]$ are then contracted using the FBP algorithm described in the previous section. Step 6 corresponds to a prediction step, as it computes a new set estimate of \mathbf{x}_{k+1} from the current set estimates of \mathbf{x}_k and \mathbf{u}_k . We can remark that an inclusion function $[\mathbf{f}]$ for \mathbf{f} has to be used, as the image of $[\mathbf{x}_k]$ and $[\mathbf{u}_k]$ by \mathbf{f} is usually not a box. Similarly, step 7 computes the intersection between the “predicted output” $[\mathbf{g}]([\mathbf{x}_{k+1}])$ computed from $[\mathbf{x}_{k+1}]$ using an inclusion function $[\mathbf{g}]$ for \mathbf{g} and the measured output. These two steps are illustrated in Figure 4.

Steps 9 and 10 correspond to the backward propagation phase of the FBP algorithm. They consist in contracting all boxes, including $[\mathbf{x}_k]$ and $[\mathbf{u}_k]$, by expressing the corresponding variables as functions of the other variables. As already noted in Section 2.2, the FBP algorithm requires that all constraints be decomposed into primitive ones.

The algorithm stops when no interval is contracted during the most recent iteration. The function computed by this algorithm will be denoted as follows:

$$[\mathbf{x}_{k+1}] = \mathcal{C}(\mathbf{f}, \mathbf{g}, [\mathbf{x}_k], [\mathbf{u}_k], [\mathbf{z}_{k+1}]). \quad (26)$$

As noted in [22], the main difficulty in using this algorithm, and the bounded error approach in general [26, Chapter 6], is the determination of error bounds. If they are overestimated, then the estimation may be overly pessimistic, resulting in

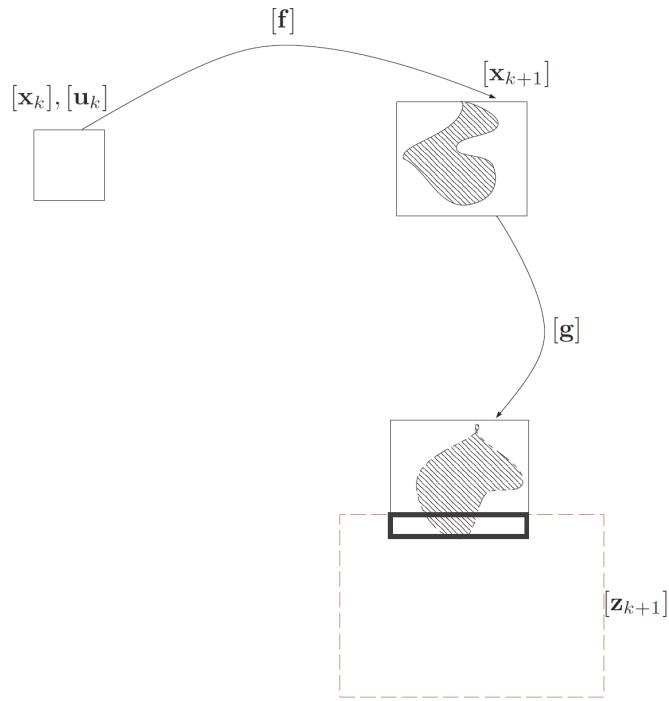


Figure 4: Forward propagation in the BEE algorithm. A set estimate of \mathbf{x}_{k+1} is computed using the current set estimates of \mathbf{x}_k and \mathbf{u}_k . The “predicted output” $[\mathbf{g}](\mathbf{x}_{k+1})$ is then intersected with the measured output, which allows us to contract $[\mathbf{z}_{k+1}]$. The constraints are then back-propagated to contract $[\mathbf{x}_{k+1}], [\mathbf{x}_k]$ and $[\mathbf{u}_k]$.

highly imprecise state estimates. On the other hand, if they are underestimated, then inconsistencies can occur in step 7 of Algorithm 1, as the intersection between the predicted and measured output can become empty, in which case the procedure has to be stopped.

In this paper, we propose to cope with this problem by using a more general formalism to describe the uncertainty of the variables of interest. This formalism, referred to as the theory of belief functions or the Dempster-Shafer theory, can be seen as generalizing both probability theory and set theory. It will allow us to describe errors no longer by single boxes, but by families of boxes with associated belief masses. This formalism is introduced in the next section.

3 Theory of Belief Functions

The Dempster-Shafer (DS) theory of belief functions [39] is a formal tool suitable for representing partial knowledge as well as uncertain and imprecise information. It is based on the formalism of belief functions, which generalize both set-based representations (including interval analysis) and probabilistic representations. In this section, we introduce the main concepts of this theory and some necessary notions that will be used in the proposed approach.

3.1 Basic Definitions

Let ω denote a variable taking values in a domain Ω called the frame of discernment. Let A_1, \dots, A_p be p subsets of Ω . A discrete *mass function* m with focal sets A_1, \dots, A_p is a function from 2^Ω to $[0, 1]$ verifying $m(A_i) = m_i > 0$ for all $i \in \{1, \dots, p\}$, $m(A) = 0$ for all $A \notin \{A_1, \dots, A_p\}$, and:

$$\sum_{i=1}^p m_i = 1. \quad (27)$$

A mass function is said to be *normal* if $m(\emptyset) = 0$, *categorical* if $p = 1$ and *consonant* if the focal sets are nested. In the following, all mass functions will be assumed to be normal, unless otherwise specified.

In most presentations of DS theory, Ω is assumed to be finite. However, the theory remains basically unchanged if Ω is infinite (even uncountable), as long as the number of focal sets remains finite. If $\Omega = \mathbb{R}$, the focal sets are usually assumed to be intervals [18][45][35]. In the multidimensional case where $\Omega = \mathbb{R}^n$, this approach can be extended by assuming focal sets to be n -dimensional boxes. The case of a discrete probability distribution is recovered when each A_i is reduced to a point. We also note that categorical mass functions are in one-to-one correspondence with subsets of Ω : to each subset $A \subseteq \Omega$ corresponds a mass function m_A such that $m_A(A) = 1$. A mass function can thus be seen as a *generalized set* [16]. This observation is the basis of the extension of interval analysis into the realm of belief functions, which will be presented in the next section.

Given a normal mass function m with focal sets A_1, \dots, A_p , the corresponding

belief and *plausibility* functions are defined, respectively, as:

$$bel(A) = \sum_{\{i | A_i \subseteq A\}} m_i \quad (28)$$

$$pl(A) = \sum_{\{i | A_i \cap A \neq \emptyset\}} m_i, \quad (29)$$

for all $A \subseteq \Omega$. In particular, if m is consonant, it may be shown [39] that

$$pl(A \cup B) = \max(pl(A), pl(B)), \quad \forall A, B \subseteq \Omega. \quad (30)$$

In this case, the plausibility function pl associated with m is thus a *possibility measure*, in the sense of possibility theory as introduced by Zadeh [47][17]. The corresponding possibility distribution π is then the function from Ω to $[0, 1]$ defined by $\pi(x) = pl(\{x\})$, for all $x \in \Omega$.

Mass functions may be used to model different situations of uncertainty about the values of variables describing a given problem. In the so-called *multivalued mapping framework* initially introduced by Dempster [11], we have an underlying space $\Psi = \{\psi_1, \dots, \psi_p\}$ and a multivalued mapping $\Gamma : \Psi \rightarrow 2^\Omega$ such that $\Gamma(\psi_i) = A_i$, for $i = 1, \dots, p$, where, for simplicity, the A_i s are assumed to be all distinct. We assume our beliefs on Ψ to be modeled by a probability measure P , with $P(\psi_i) = m_i$. For instance, P might represent some stochastic mechanism that selects an element from Ψ . If ψ_i is selected, all we know about ω is that it is in A_i . Our beliefs concerning Ω may then be represented by the mass function induced by P through the multivalued mapping Γ , so that $m(\Gamma(\psi_i)) = P(\psi_i) = m_i$ for all $i \in \{1, \dots, p\}$. We then have

$$bel(A) = P(\{\psi_i \in \Psi \mid \Gamma(\psi_i) \subseteq A\}) \quad (31)$$

and

$$pl(A) = P(\{\psi_i \in \Psi \mid \Gamma(\psi_i) \cap A \neq \emptyset\}), \quad (32)$$

for all $A \subseteq \Omega$.

The elements of Ψ may be viewed as different ways of explaining or interpreting the evidence that we have for a given problem, or as different mechanisms that may have produced this evidence. In some cases, they may be given a precise physical interpretation. For instance, assume that ω is a measurement error and $\Omega \subseteq \mathbb{R}$. The sensor that delivers the measurement can be in the nominal state (ψ_1) or in a defective state (ψ_2). In the first case, all we know is that $|\omega| \leq b$, whereas in the second case ω can take any value in Ω . The probability that the sensor is in the nominal state is m_1 . This information about ω may be represented as a mass function m such that $m([-b, b]) = m_1$ and $m(\Omega) = 1 - m_1$.

3.2 Extending Interval Analysis to Mass Functions

In [44], Yager proposed a simple scheme for extending arithmetic and other operations to mass functions (see also [36]). This approach will be summarized in this section. Because our goal here is to generalize interval analysis, all mass functions will be assumed to have interval focal sets.

Let us assume that we have two real variables x and y , and our knowledge about these variables is modeled by mass functions denoted by m^x and m^y , respectively. We are interested in the value of $z = x \diamond y$, where \diamond is a binary operator from \mathbb{R}^2 to \mathbb{R} . If m^x and m^y are categorical and such that $m^x([x]) = 1$ and $m^y([y]) = 1$ for some intervals $[x]$ and $[y]$ in \mathbb{IR} , then interval analysis can be applied and our knowledge about z can be represented by a categorical mass function m^z with focal interval $[x] \diamond [y]$, defined by (3). In the more general case, where m^x and m^y are mass functions with arbitrary numbers of focal intervals, then some extension of interval analysis has to be proposed, based on additional assumptions.

Let us assume that m^x and m^y are induced by multivalued mappings Γ^x and Γ^y from underlying spaces Ψ^x and Ψ^y , with $[x_i] = \Gamma^x(\psi_i^x)$ for $i = 1, \dots, p^x$ and $[y_j] = \Gamma^y(\psi_j^y)$ for $j = 1, \dots, p^y$. Let us further assume that our beliefs about Ψ^x and Ψ^y are modeled by probability measures P^x and P^y , and that the joint probability measure $P^{x,y}$ for $\Psi^x \times \Psi^y$ is simply the product of P^x and P^y :

$$P^{x,y}(\psi_i^x, \psi_j^y) = P^x(\psi_i^x)P^y(\psi_j^y) = m_i^x m_j^y, \quad (33)$$

for all i and j . If P^x and P^y correspond to some stochastic mechanisms selecting elements from Ψ^x and Ψ^y , then (33) simply means that these two mechanisms are independent. Under these assumptions, what can we say about z ?

If we know that ψ_i^x and ψ_j^y have been selected, then all we know about x , y and z is that $x \in [x_i]$, $y \in [y_j]$, and $z = x \diamond y \in [x_i] \diamond [y_j]$. As this happens with probability $m_i^x m_j^y$, we can see that our beliefs about z can be represented by the mass function m^z , defined as follows:

$$m^z([z]) = \sum_{\{i,j|[x_i] \diamond [y_j] = [z]\}} m_i^x m_j^y, \quad (34)$$

for all $[z] \in \mathbb{IR}$, which generalizes (3).

This approach can easily be extended to the case of a function of n variables. Let \mathbf{f} be a function from \mathbb{R}^n to \mathbb{R} , $z = \mathbf{f}(x_1, \dots, x_n)$, and $[\mathbf{f}]$ an inclusion function for f . Assume that our knowledge of each variable x_i is represented by a mass function m^{x_i} . If each $m^{x_i}([x_i]) = 1$ for some interval $[x_i]$, then our knowledge of z can be represented by:

$$[z] = [\mathbf{f}](x_1, \dots, x_n). \quad (35)$$

In the general case, using the same line of reasoning as above, we can derive a mass function m^z on z under assumption of independence, using the following formula:

$$m^z([z]) = \sum_{\{i_1, \dots, i_n | [z] = [\mathbf{f}](x_{i_1}, \dots, x_{i_n})\}} m_{i_1}^{x_1} \dots m_{i_n}^{x_n}. \quad (36)$$

EXAMPLE 2 Assume that m^x and m^y have the following expressions:

$$m^x([1, 2]) = 0.7, \quad m^x([0, 3]) = 0.3, \quad (37)$$

$$m^y([0, 1]) = 0.6, \quad m^y([0, 2]) = 0.4. \quad (38)$$

Let z be a variable related to x and y by the equation $z = x \exp(y)$. What are our beliefs regarding z ? Using (36), a mass function m^z on z is then derived as

$$m^z([1, 2] \cdot [\exp([0, 1])]) = m^z([1, 2e]) = 0.7 \times 0.6 = 0.42 \quad (39)$$

$$m^z([1, 2] \cdot [\exp]([0, 2])) = m^z([1, 2e^2]) = 0.7 \times 0.4 = 0.28 \quad (40)$$

$$m^z([0, 3] \cdot [\exp]([0, 1])) = m^z([0, 3e]) = 0.3 \times 0.6 = 0.18 \quad (41)$$

$$m^z([0, 3] \cdot [\exp]([0, 2])) = m^z([0, 3e^2]) = 0.3 \times 0.4 = 0.12. \quad (42)$$

3.3 Summarization

Despite its success as a model of human reasoning under uncertainty, one of the arguments often raised against the theory of belief functions is its relatively high computational complexity. Indeed, the number of focal sets obtained when computing a function of n mass functions using (36) grows exponentially with n , which may lead to serious problems when both the number of steps in the computations and the size of focal sets become large [13].

A natural way to decrease this complexity is therefore to decrease the number of focal sets by grouping similar or unimportant ones. This strategy is at the basis of the summarization method [30], in which the focal sets with the smallest mass are aggregated and the sum of their masses transferred to their union.

More precisely, let m be a mass function on Ω with p focal sets A_1, \dots, A_p , $p > 1$. We note, as before, $m_i = m(A_i)$, $i = 1, \dots, p$. Assume that the focal sets are ordered such that $m_1 \geq m_2 \geq \dots \geq m_p$, and let $1 \leq q < p$. A mass function m' with q focal sets can be constructed from m as follows:

$$m'(A_i) = m_i, \quad i = 1, \dots, q - 1, \quad (43)$$

$$m'(A_0) = \sum_{i=q}^p m_i, \quad (44)$$

where A_0 is defined as $\bigcup_{i=q}^p A_i$, or $\bigsqcup_{i=q}^p A_i$ when the A_i are intervals or boxes.

The summarization method is a simple approximation method making it possible to control the complexity of computations in DS theory by bounding the number of focal sets. More sophisticated approximation schemes have been proposed in, for example, [13] and [14]. Such refined schemes allow a better approximation at the cost of higher computing cost. The simpler and faster summarization method will be adopted in this paper.

3.4 Expectations

Let us assume in this subsection that $\Omega = \mathbb{R}$, and our state of knowledge regarding some variable x is represented by a mass function m . How can the concept of the expected value of x be defined in this context? An answer to this question is more easily found by interpreting m as defining a set of probability measures P such that

$$\text{bel}(A) \leq P(A) \leq \text{pl}(A), \quad (45)$$

for any measurable subset A of \mathbb{R} . A probability measure P verifying the above inequalities is said to be *compatible* with m ; the set of such probability measures will be noted as $\mathcal{P}(m)$.

The lower and upper expectations [11] of x with respect to m may then be defined, respectively, as

$$\mathbb{E}_*(m) = \inf_{P \in \mathcal{P}(m)} \mathbb{E}(P), \quad (46)$$

$$\mathbb{E}^*(m) = \sup_{P \in \mathcal{P}(m)} \mathbb{E}(P). \quad (47)$$

When the focal sets are real intervals $[x_i]$, then the lower-upper expectation interval $[\mathbb{E}](m) = [\mathbb{E}_*(m), \mathbb{E}^*(m)]$ (hereafter referred to as the *interval expectation* of m) may be shown [11] to be equal to

$$[\mathbb{E}](m) = \sum_{i=1}^p m_i \cdot [x_i]. \quad (48)$$

A particular value in this interval may be selected by averaging the center c_i of each focal interval $[x_i]$, which will be denoted:

$$\mathbb{E}(m) = \sum_{i=1}^p m_i c_i. \quad (49)$$

The quantity $\mathbb{E}(m)$ happens to be the expectation with respect to the pignistic probability measure associated with m [40, 35]; it will be referred to as the *pignistic expectation* of m .

Equations (48) and (49) can easily be extended to the multidimensional case where $\Omega = \mathbb{R}^n$. The interval expectation of m is then an n -dimensional box, and its pignistic expectation is a vector of \mathbb{R}^n .

4 Belief State Estimation

As mentioned in Section 2.3, this interval framework guarantees the validity of the calculations under the assumptions that the system equations are correct and measurement errors remain within given bounds. However, this approach may be very conservative, which may result in very imprecise estimates of state variables. As noted in [22], “the major implementation problem of such an approach is to determine correctly the bounds of the noises. Indeed, if these bounds are underestimated, the contractor may lead to no solution. On the contrary, if the bounds are overestimated, the estimated boxes can be very large (the estimates are then very pessimistic)”.

In this section, we propose to solve this problem by replacing intervals by general mass functions with focal intervals. The construction of mass functions using prior knowledge of measurement error will first be addressed in Section 4.1. The bounded error estimation method recalled in Section 2.3 will then be extended to the Dempster-Shafer framework in Subsection 4.2. A simple illustrative example is given in Appendix A.

4.1 Construction of Mass Functions

As recalled in Section 1, classical state estimation techniques such as the Kalman filter and its extensions assume process and observation noises to be normal. In contrast,

the BEE method (Section 2.3) makes the much weaker assumption that perturbations are bounded and that nothing is known except the bounds. While the former approach is very restrictive, the latter may be argued to be too conservative — as noted at the beginning of this section — leading to overly imprecise estimates of the system state. In many cases, however, additional realistic assumptions about error distributions can be made: for instance, we can assume some measure of central tendency such as the mode, the mean or the median to be equal to zero. As will be shown below, the belief function formalism allows us to refine the BEE approach by making use of such additional assumptions.

Let us consider a continuous real random variable X characterized by a probability distribution P_X with known support $[a, b]$ and mode $c \in (a, b)$. Let π denote the triangular possibility distribution

$$\pi(x) = \begin{cases} \frac{x-a}{c-a} & \text{if } a \leq x < c, \\ \frac{b-x}{b-c} & \text{if } c \leq x < b, \\ 0 & \text{otherwise.} \end{cases} \quad (50)$$

As shown in [5], the following inequalities hold for every measurable set A :

$$\Pi(A) \leq P_X(A) \leq 1 - \Pi(\overline{A}), \quad (51)$$

where \overline{A} denotes the complement of A , and Π is the possibility measure associated with π :

$$\Pi(A) = \sup_{x \in A} \pi(x). \quad (52)$$

The possibility distribution π can thus be seen as an approximation of the set of probability measures with support $[a, b]$ and mode c . Such a continuous possibility measure can be itself approximated by a discrete mass function m with p focal intervals defined as follows.

For any $\alpha \in (0, 1]$, the α -cut of π is the set of values x such that $\pi(x) \geq \alpha$. This is the interval $[\pi^\alpha]$ with bounds:

$$\underline{\pi}^\alpha = a + \alpha(c - a) \quad (53)$$

$$\overline{\pi}^\alpha = c + (1 - \alpha)(b - c). \quad (54)$$

By convention, let $[\pi^0] = [a, b]$. Let us consider $p - 1$ distinct values for α : $0 = \alpha_0 < \alpha_1 < \alpha_2 < \dots < \alpha_{p-1} < 1$, and let m be the consonant mass function with focal intervals $[\pi^{\alpha_k}]$, $k = 0, \dots, p - 1$, defined by:

$$m([\pi^0]) = \alpha_1, \quad (55)$$

$$m([\pi^{\alpha_1}]) = \alpha_2 - \alpha_1, \quad (56)$$

$$\vdots \quad (57)$$

$$m([\pi^{\alpha_{p-1}}]) = 1 - \alpha_{p-1}. \quad (58)$$

The principle of this construction is illustrated in Figure 5. In this figure, there are two α -cuts, $[\pi^{\alpha_1}]$ and $[\pi^{\alpha_2}]$, associated with α_1 and α_2 respectively. By considering these α -cuts and $[\pi^0] = [a, b]$, we can construct a mass function with $p = 3$ focal

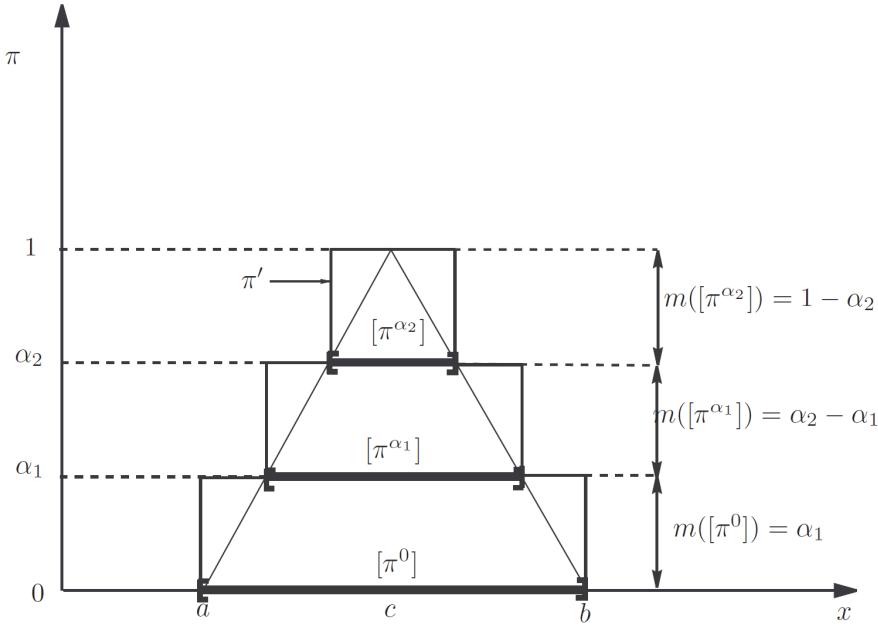


Figure 5: Approximation of a triangular possibility distribution π by a discrete consonant mass function. The focal set of m are defined as the support of $[\pi^0]$ of π , and α -cuts $[\pi^{\alpha_k}]$ for $k = 1, \dots, p - 1$ ($p = 3$ in this figure). The resulting possibility distribution π' dominates π .

elements: $m([\pi^0]) = \alpha_1$, $m([\pi^{\alpha_1}]) = \alpha_2 - \alpha_1$ and $m([\pi^{\alpha_2}]) = 1 - \alpha_2$. It is clear that m corresponds to a possibility distribution π' that approximates π , and such that $\pi' \geq \pi$. Consequently, if bel and pl denote, respectively, the belief and plausibility functions associated with m , the following inequalities hold for every event A :

$$bel(A) \leq P_X(A) \leq pl(A), \quad (59)$$

which means that m can be considered an approximation of the set of probability measures with known support $[a, b]$ and mode c .

We may remark that, in the above formalism, all values outside the interval $[a, b]$ are considered to be absolutely impossible. In many cases, such an assumption may be argued to be too strong. For instance, if X denotes a measurement error, the bounds a and b may be derived from knowledge of the measurement process, which may be valid most of the time except in rare situations (for instance, in case of sensor failure). To account for the fact that values outside $[a, b]$ are not impossible but have very small plausibilities, we may *discount* m by a small discount rate ϵ [39, page 251]. Discounting m means transferring to the frame Ω a fraction ϵ of the mass $m(A)$ given to all proper subsets A of Ω . The resulting mass function ϵm is thus defined as

$$\epsilon m(A) = \begin{cases} (1 - \epsilon)m(A) & \text{if } A \subset \Omega, \\ (1 - \epsilon)m(\Omega) + \epsilon & \text{if } A = \Omega. \end{cases} \quad (60)$$

We note that ${}^\epsilon bel(A) = (1 - \epsilon)bel(A) \leq bel(A)$ and ${}^\epsilon pl(A) = (1 - \epsilon)pl(A) + \epsilon \geq pl(A)$ for all strict subsets A of Ω . Consequently, discounting has the effect of enlarging the intervals $[bel(A), pl(A)]$, and (59) remains valid.

EXAMPLE 3 Let U be a real random variable modeling the observation noise in some measurement process. Assume that expectation of U is zero, and its standard deviation σ is given. In bounded error approaches, it is common to take the interval $[-3\sigma, +3\sigma]$ as the support of U . If we assume that the mode of U is zero, then we can better approximate its probability distribution by a symmetrical triangular possibility distribution, which can itself be approximated by the following discrete mass function with $p = 3$ focal intervals:

$$m([-3\sigma, 3\sigma]) = 1/3, \quad (61)$$

$$m([-2\sigma, 2\sigma]) = 1/3, \quad (62)$$

$$m([- \sigma, \sigma]) = 1/3. \quad (63)$$

Obviously, a better approximation of the continuous triangular possibility distribution can be obtained by increasing the number p of focal intervals, at the expense of higher computational complexity. Discounting m at rate ϵ and assuming $\Omega = \mathbb{R}$ yields the following discounted mass function:

$${}^\epsilon m([-3\sigma, 3\sigma]) = (1 - \epsilon)/3, \quad (64)$$

$${}^\epsilon m([-2\sigma, 2\sigma]) = (1 - \epsilon)/3, \quad (65)$$

$${}^\epsilon m([- \sigma, \sigma]) = (1 - \epsilon)/3 \quad (66)$$

$${}^\epsilon m(\mathbb{R}) = \epsilon. \quad (67)$$

4.2 Extending the BEE Approach

Let us consider again the dynamical system described by Equations (20)-(21). However, in contrast with the interval analysis performed in Section 2.3, we now assume the initial state \mathbf{x}_0 as well as the input vector \mathbf{u}_k and the observation vector \mathbf{z}_k at each time step k to be described by mass functions with interval focal sets. To denote a mass function, the variable will be indicated as a superscript, and the time step as a subscript. For instance, $m_k^{\mathbf{x}}$ will denote the mass function regarding the state \mathbf{x}_k at time step k .

The BEE algorithm can be generalized using the extension of interval analysis to belief functions, as recalled from Section 3.2. We recall that Algorithm 1 computes a box estimate $[\mathbf{x}_{k+1}]$ of the state at each time step $k+1$ from the imprecise input $[\mathbf{u}_k]$ and the output $[\mathbf{z}_{k+1}]$, using (26). The theory recalled from Section 3.2 allows us to extend function \mathcal{C} from intervals to mass functions.

Let $p_k^{\mathbf{x}}$, $p_k^{\mathbf{u}}$ and $p_k^{\mathbf{z}}$ denote, respectively, the number of focal sets of $m_k^{\mathbf{x}}$, $m_k^{\mathbf{u}}$, $m_k^{\mathbf{z}}$ (without loss of generality, we assume the mass functions on \mathbf{u} and \mathbf{z} to be computed using the procedure described in Section 4.1, with a constant number of focal intervals). The i -th focal element of $m_k^{\mathbf{x}}$ will be noted as $[\mathbf{x}_k^i]$, and similarly for the other mass functions. Using (36), Equation (26) can be generalized to

$$m_{k+1}^{\mathbf{x}}([\mathbf{x}]) = \sum_{\{i,j,\ell | [x] = \mathcal{C}(\mathbf{f}, \mathbf{g}, [\mathbf{x}_k^i], [\mathbf{u}_k^j], [\mathbf{z}_{k+1}^{\ell}])\}} m_k^{\mathbf{x}}([\mathbf{x}_k^i]) \cdot m_k^{\mathbf{u}}([\mathbf{u}_k^j]) \cdot m_{k+1}^{\mathbf{z}}([\mathbf{z}_{k+1}^{\ell}]). \quad (68)$$

The detailed procedure is described in Algorithm 2. It can be summarized as follows. The algorithm depends on three parameters: the number of focal elements

Algorithm 2 Belief State Estimation (BSE) algorithm.

Input: $m_k^{\mathbf{x}}, m_k^{\mathbf{u}}, m_{k+1}^{\mathbf{z}}, p^{\mathbf{u}}, p^{\mathbf{z}}, q$

Output: $m_{k+1}^{\mathbf{x}}, [\hat{\mathbf{x}}_{k+1}], \hat{\mathbf{x}}_{k+1}$

- 1: Read the input \mathbf{u}_k and its error. Deduce $m_k^{\mathbf{u}}$ with $p^{\mathbf{u}}$ focal elements
 - 2: Read the output \mathbf{z}_{k+1} and its error. Deduce $m_{k+1}^{\mathbf{z}}$ with $p^{\mathbf{z}}$ focal elements
 - 3: $r \leftarrow 0$
 - 4: **for** $i = 1$ to $p_k^{\mathbf{x}}$ **do**
 - 5: **for** $j = 1$ to $p^{\mathbf{u}}$ **do**
 - 6: **for** $\ell = 1$ to $p^{\mathbf{z}}$ **do**
 - 7: $r \leftarrow r + 1$
 - 8: $[\mathbf{x}_{k+1}^r] \leftarrow \mathcal{C}(\mathbf{f}, \mathbf{g}, [\mathbf{x}_k^i], [\mathbf{u}_k^j], [\mathbf{z}_{k+1}^\ell]).$
 - 9: $m_{k+1}^{\mathbf{x}}([\mathbf{x}_{k+1}^r]) \leftarrow m_k^{\mathbf{x}}([\mathbf{x}_k^i]) \cdot m_k^{\mathbf{u}}([\mathbf{u}_k^j]) \cdot m_{k+1}^{\mathbf{z}}([\mathbf{z}_{k+1}^\ell])$
 - 10: **end for**
 - 11: **end for**
 - 12: **end for**
 - 13: Normalize $m_{k+1}^{\mathbf{x}}$
 - 14: Summarize $m_{k+1}^{\mathbf{x}}$ to keep at most q focal elements.
 - 15: *% Computation of interval and point estimates %*
 - 16: $[\hat{\mathbf{x}}_{k+1}] \leftarrow [\mathbb{E}(m_{k+1}^{\mathbf{x}})]$
 - 17: $\hat{\mathbf{x}}_{k+1} \leftarrow \mathbb{E}(m_{k+1}^{\mathbf{x}})$
-

$p^{\mathbf{u}}$ and $p^{\mathbf{z}}$ for the input and the output, respectively, and the maximum number q of focal elements for $m_{x+1}^{\mathbf{x}}$ used by the summarization algorithm recalled from Section 3.3. In steps 1 and 2, the mass functions on the input at time k and on the output at time $k+1$ are constructed using the method described in Section 4.1. The BEE algorithm (Algorithm 1) is then applied for each choice of focal element of $[\mathbf{x}_k^i]$ of $m_k^{\mathbf{x}}$, focal interval $[\mathbf{u}_k^j]$ of $m_k^{\mathbf{u}}$ and focal element $[\mathbf{z}_{k+1}^\ell]$ of $m_{k+1}^{\mathbf{z}}$ (steps 4-12). The belief mass assigned to the r -th focal element $[\mathbf{x}_{k+1}^r]$ of $m_{k+1}^{\mathbf{x}}$ is the product of the masses assigned to $[\mathbf{x}_k^i]$, $[\mathbf{u}_k^j]$ and $[\mathbf{z}_{k+1}^\ell]$ (line 9). Once the whole mass function $m_{k+1}^{\mathbf{x}}$ has been computed, it is normalized (step 13) and summarized (step 14) using the summarization algorithm described in Section 3.3 with a maximum number q of focal elements. Normalization consists in dividing each mass assigned to a non empty focal set by $1 - m_{k+1}^{\mathbf{x}}(\emptyset)$ and setting $m_{k+1}^{\mathbf{x}}(\emptyset)$ to zero. Finally, interval and pignistic expectations are computed using (48) and (49).

A simple illustrative example is given in Appendix A.

This algorithm, hereafter referred to as the Belief State Estimation (BSE) algorithm, thus amounts to running the BEE algorithm several times (Algorithm 1). The maximum number of runs of BEE at each time step is $p^{\mathbf{x}}p^{\mathbf{z}}q$: the choice of these three parameters is thus crucial for controlling the execution time of the algorithm. The sensitivity to these three parameters of the results will be studied in the experiments presented in Section 5.

As explained in Section 3.2, Equation (68) results from the assumption of independence between mass functions $m_k^{\mathbf{x}}([\mathbf{x}_k^i])$, $m_k^{\mathbf{u}}([\mathbf{u}_k^j])$ and $m_{k+1}^{\mathbf{z}}$, in the precise sense outlined in Section 3.2. This hypothesis is the counterpart of the statistical independence assumptions that are commonly accepted when models are designed with

random perturbations.

As a final remark, we note that the above algorithm may fail if each call to the BEE procedure in step 8 returns the empty set, in which case the normalization operation performed in step 13 is undefined. Such a situation may arise if the input \mathbf{u}_k or the output \mathbf{z}_{k+1} are outliers and are inconsistent with the current state estimate encoded in mass function $m_k^{\mathbf{x}}$. To avoid such a situation, we may discount $m_k^{\mathbf{u}}$ and/or $m_{k+1}^{\mathbf{x}}$ by a small rate ϵ using (60), making the estimation procedure robust against outliers.

5 Application

In this section, we apply the belief state estimation (BSE) algorithm introduced above to dynamic localization of a land vehicle. The application and the data are identical to those used in [22] and [1]¹. We will first introduce the problem and describe the experimental settings and then present the simulation results.

5.1 Problem Description

As noted in [22], dynamic localization of a land vehicle in outdoor environments is a key issue in many applications. It consists of computing, at each time step k , an estimation of the vehicle position based on sensor measurements. Usually, two kinds of sensors are used: dead reckoning sensors (such as odometers, gyroimeters, etc.) and absolute sensors, such as telemeters, goniometers, vision or Global Positioning System (GPS), among others. A major issue is to exploit the complementarity and redundancy between these sensors in order to achieve higher precision, availability and integrity. For instance, the quality of GPS positioning depends on the configuration of visible satellites, and GPS information can even become unavailable because of masking effects occurring in forests, tunnels, cities, etc. Fusing GPS information with dead reckoning sensor measurements makes it possible to filter the GPS estimates, thus increasing the performance of the localizer.

The conventions and notations used in this section are illustrated in Figure 6. The mobile frame origin M is chosen as the middle of the rear axle. Let (x_k, y_k) be the position of the vehicle and θ_k its heading angle at time step k . Let $\mathbf{x}_k = (x_k, y_k, \theta_k)^T$ be the state of the vehicle. As shown in [22], its evolution can be described by the following state equations:

$$\begin{cases} x_{k+1} = x_k + \delta_{S,k} \cos(\theta_k + \frac{\delta_{\theta,k}}{2}) \\ y_{k+1} = y_k + \delta_{S,k} \sin(\theta_k + \frac{\delta_{\theta,k}}{2}) \\ \theta_{k+1} = \theta_k + \delta_{\theta,k}, \end{cases} \quad (69)$$

where $\delta_{S,k}$ and $\delta_{\theta,k}$ denote, respectively, the elementary displacement and rotation of the vehicle at time step k . These quantities can be obtained with good precision with two rear-wheel ABS sensors and a fiber optic gyrometer, respectively. They are considered as the input to the system. With previous notations, we can thus denote $\mathbf{u}_k = (\delta_{S,k}, \delta_{\theta,k})^T$.

¹We thank Philippe Bonnifait and his group for providing and allowing us to use these data.

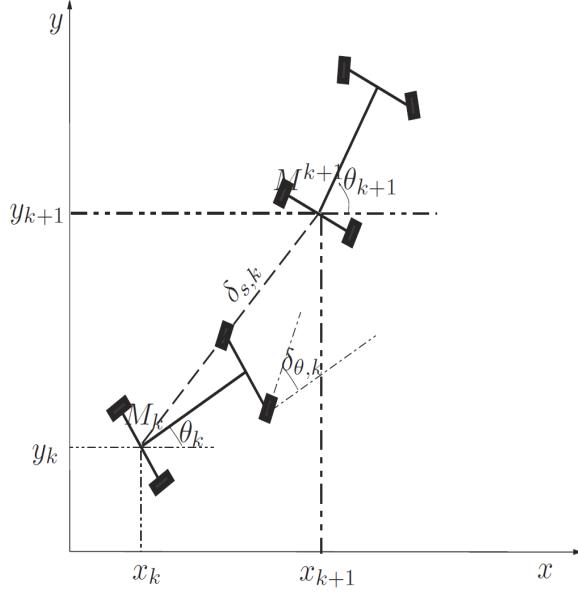


Figure 6: Vehicle representation at time steps k and $k + 1$.

The measure of the position $\mathbf{z}_k = (x_{GPS}, y_{GPS})$ is obtained by a GPS receiver, after converting each estimated point (longitude, latitude) in a Cartesian local frame [21]. We note that the heading angle θ_k is not observed.

5.2 Experimental Settings

The results reported here were obtained using real sensor measurements collected using an experimental car (see [22] for details about the sensors and data processing). The vehicle was equipped with two GPS receivers: a Trimble AgGPS 132 to estimate its position in real time, and a high-precision Thales Navigation GPS receiver to compute estimation errors. The Thales receiver was used in post-processed kinematic (PPK) mode with a local base (a Trimble 7400). This system was able to provide reference positions at a sampling rate of 1 Hz. Because the constellation of the satellites was satisfactory during all the trials, kinematic ambiguities were resolved, and an accuracy of a few centimeter was achieved. To avoid confusion, this reference data will be referred to as “PPK”, while the term “GPS” will be reserved for data from the Trimble AgGPS 132 GPS receiver.

Experiments were carried out on a test track in Versailles (France). We report hereafter the analysis of a 4.7 km trajectory with a mean speed of 50 km/h (see the left-hand side of Figure 7). The zoomed part of the trajectory (right-hand side of Figure 7) shows the GPS and estimated positions.

As in [22] and [1], the GPS error bounds were taken to be plus or minus three times the estimated standard deviation computed in real time by the receiver. Assuming the mode on the GPS error to be zero, mass functions $m_0^{\mathbf{x}}$ and $m_k^{\mathbf{z}}$ were constructed to approximate triangular possibility distributions, as explained in Section 4.1. The heading angle θ was initialized as $[\theta_0] = [0, 2\pi]$. The error bounds of the input vector \mathbf{u}_k were computed from characteristics of the ABS sensor and the gyrometer (± 3

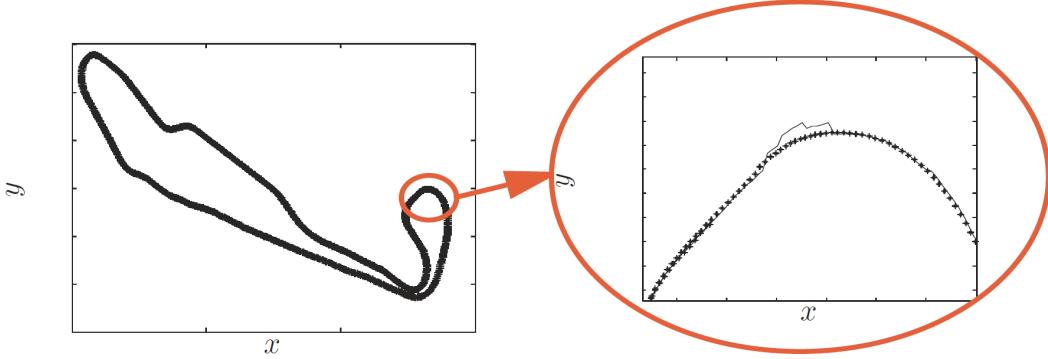


Figure 7: Test trajectory (left), with a zoom on part of the trajectory (right) showing the GPS positions (solid line) and the estimated positions (*).

degrees for $\delta_{\theta,k}$).

The three parameters of the BSE algorithm are p^z , p^u and q . In the results presented in the next section, the default values of these parameters are $p^z = 4$, $p^u = 1$ and $q = 20$. A sensitivity analysis of the influence of these parameters will also be presented.

5.3 Results

Figure 8 shows the focal sets of m_k^x at time steps $k = 10$ and 100 . In this figure, the vehicle trajectory estimated by the GPS alone is plotted as a solid line and the estimated positions given by the BSE method are plotted as black (*) points. The interval expectation $[E](m_k^x)$ is represented in bold. For the sake of clarity, only the focal sets with mass greater than 0.05 are shown. The summarization process is illustrated by Figure 9, which shows the focal sets of m_k^x before and after summarization.

Figures 10 and 11 show the interval errors on x and y for the GPS as well as the BEE and BSE methods. We can see that the BSE method provides narrower intervals that still contain the true positions along both coordinates. The BSE method is also much less affected than is the BEE method by degradation of the GPS signal due to masking effects, as occurred around time steps 150, 210 and 400.

Figures 12 and 13 show the distance (in meters) between estimated and actual positions of the vehicle for the following estimation methods: GPS, BEE and BSE (Figure 12); GPS, particle filter (PF) and BSE (Figure 13). Implementation and parameterization of the PF method were the same as reported in [1]. The PF was run with 1000 particles. We can see that the BSE method is clearly much more accurate than BEE. The difference is smaller between PF and BSE, with a small advantage exhibited by BSE most of the time.

These results are confirmed by Table 1, which reports the mean squared errors (MSE) on both coordinates x and y as well as the mean one-step running time (on a PC with Matlab) for the GPS, the PF (with 1000, 2000 and 3000 particles), bounded error estimator (BEE) and the belief state estimator (BSE) with the default parameter values ($p^z = 4$, $p^u = 1$ and $q = 20$). We can see that the BSE method significantly outperforms the BEE method, and is also slightly more accurate than the PF method,

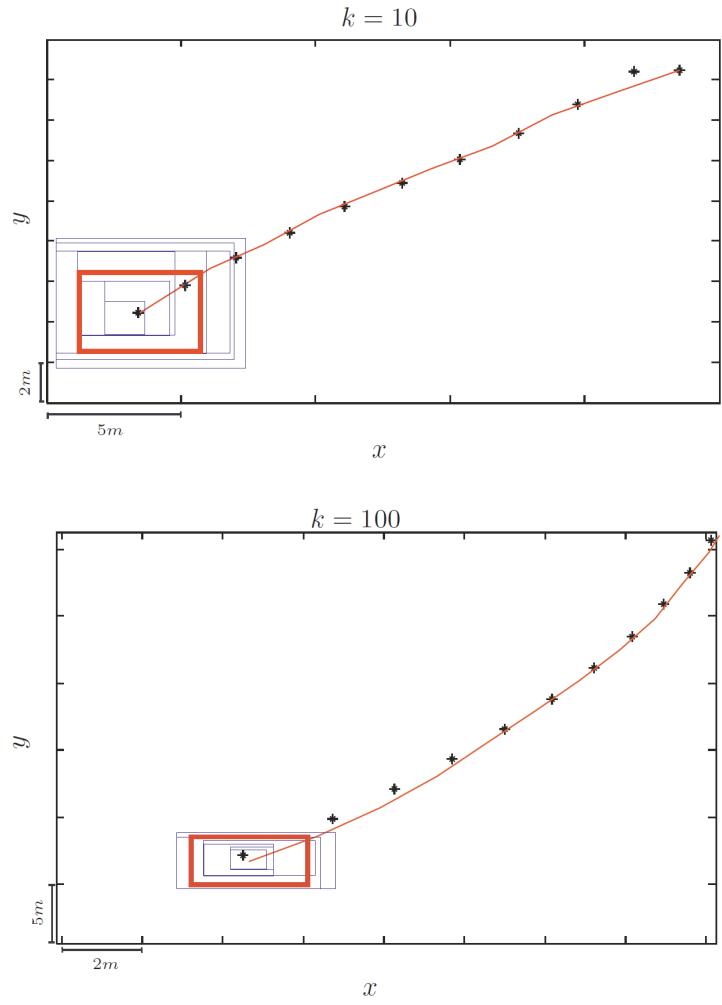


Figure 8: Focal sets of the mass function for \mathbf{x}_k , with mass greater than 0.05, at time steps $k = 10$ (up) and $k = 100$ (down). The GPS and estimated positions are shown as a solid lines and * symbols, respectively. The interval expectation $\mathbb{E}(m_k^x)$ is represented in bold. The vehicle is moving downwards.

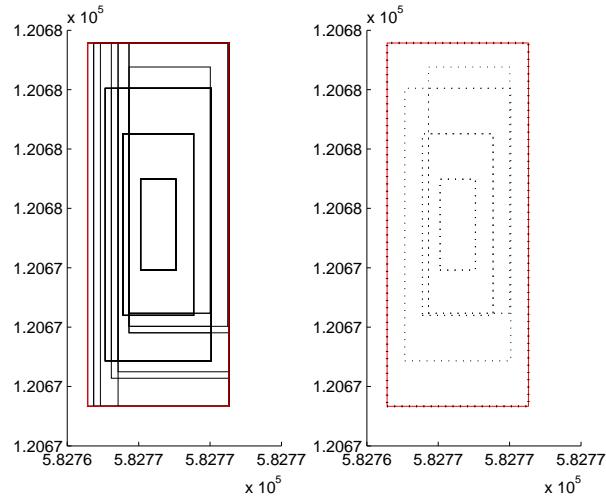


Figure 9: Focal sets of the mass function on x_k before (left) and after (right) summarization. The line width is proportional to the mass.

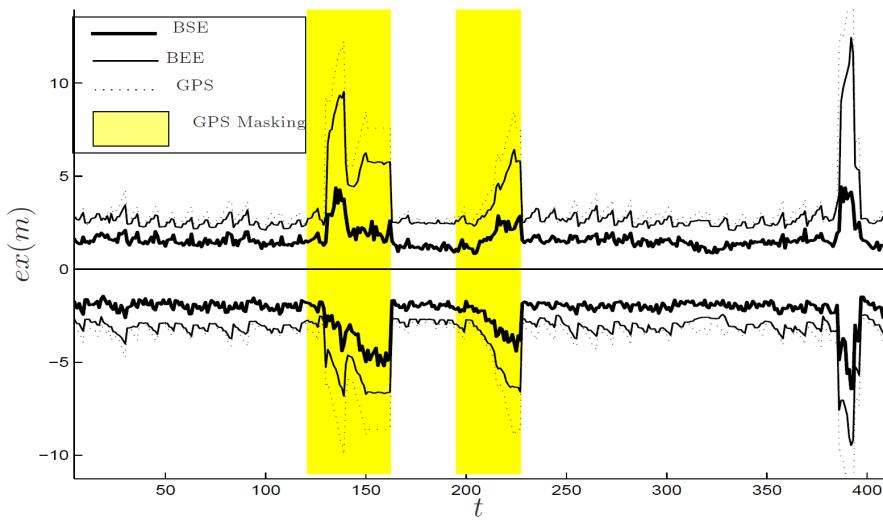


Figure 10: Interval errors on x for GPS (dashed line), BSE (bold line) and BEE (thin line).

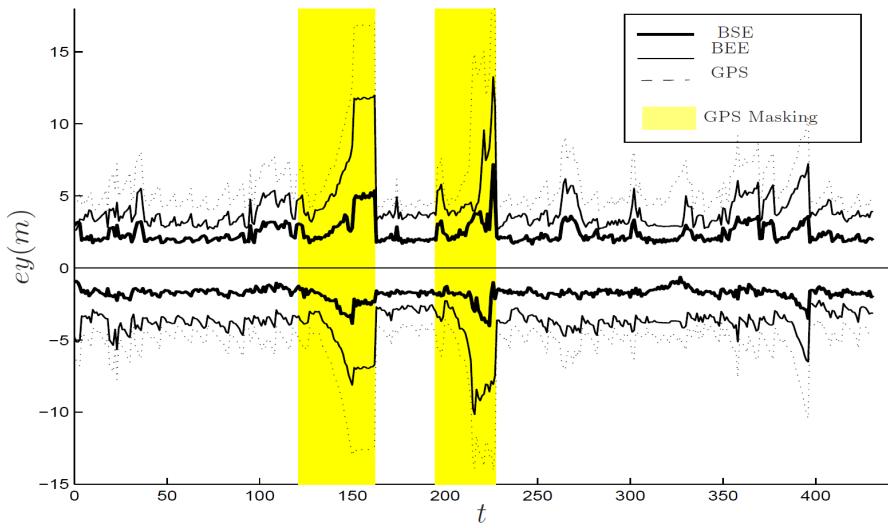


Figure 11: Interval errors on y for GPS (dashed line), BSE (bold line) and BEE (thin line).

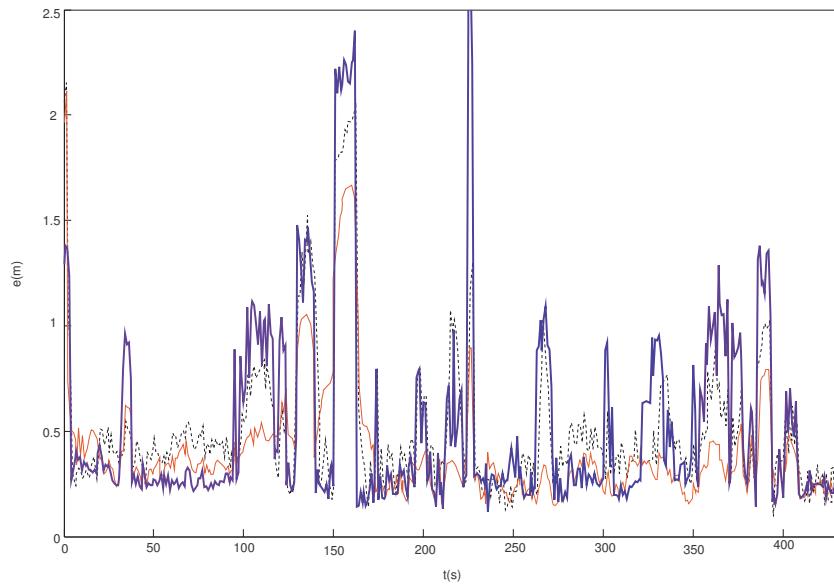


Figure 12: Distance in meters between the estimated and actual vehicle positions for the GPS (bold line), BEE (dotted line) and BSE (thin line).

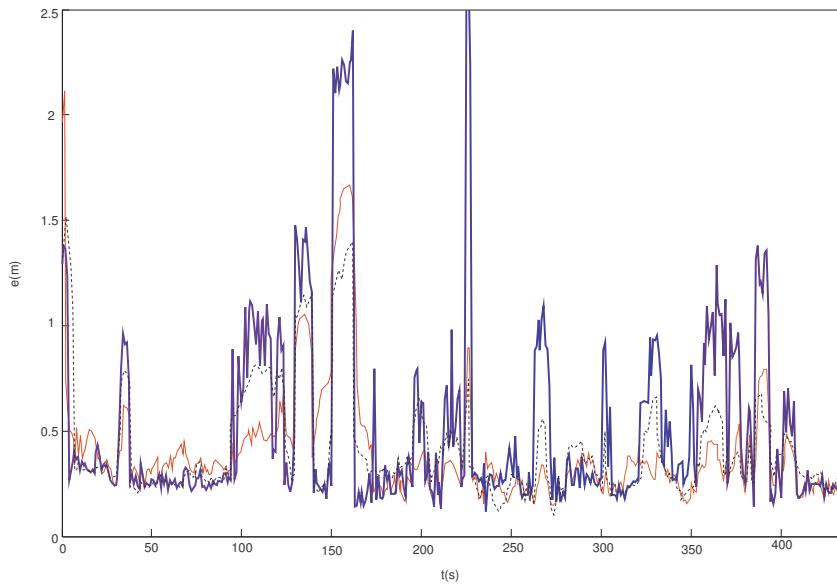


Figure 13: Distance in meters between the estimated and actual vehicle positions for the GPS (bold line), PF (dotted line) and BSE (thin line).

Table 1: Mean squared errors (MSE) on x and y , and one-step running times for the GPS without filtering, and the following methods: particle filter (PF) with 3000, 2000, 1000 particles, bounded error estimation (BEE) and belief state estimation (BSE) with $q = 15$ and 20.

	GPS	PF(3000)	PF(2000)	PF(1000)	BEE	BSE (20)
MSE on x (m^2)	0.134	0.119	0.121	0.125	0.123	0.118
MSE on y (m^2)	0.374	0.215	0.232	0.243	0.249	0.199
Running time (ms)	-	639	526	401	136	409

Table 2: Mean squared errors (MSE) on x and y (in m^2), and average one-step running times (in ms) for the belief state estimation (BSE).

p^z	p^u	q	running time	MSE on x	MSE on y
1	1	1	136	0.123	0.249
2	1	10	158	0.121	0.240
3	1	10	198	0.120	0.236
4	1	10	241	0.120	0.229
2	2	10	199	0.121	0.236
2	4	10	236	0.120	0.229
2	6	10	316	0.120	0.221
4	1	15	386	0.120	0.225
4	1	20	409	0.118	0.199
4	1	22	426	0.118	0.197
4	1	25	452	0.116	0.197

with comparable running time. Increasing the number of particles in the PF improves its performance, at a higher computational cost.

Table 2 displays the average one-step-ahead running times as well as the mean squared errors on both spatial coordinates for the BSE algorithm with different choices for the three parameters p^z , p^u , q . The setting $p^z = p^u = q = 1$ corresponds to the BEE method. Parameter q seems to have a critical impact on both computational cost and performance. However, increasing it beyond 20 does not significantly improve estimation accuracy.

We can remark that simulations were done in Matlab on a PC computer and that the code was not optimized. Consequently, the reported running times in Tables 1 and 2 can only be used to compare the methods. We can conclude from these first results that our approach is no more costly, but slightly more accurate than the PF approach for the kind of application considered here. Although more costly than the BEE, our algorithm remains compatible with a real-time implementation.

6 Conclusion

This paper has proposed a new approach to non-linear state estimation based on belief function theory and interval analysis. This method uses mass functions composed of a finite number of axis-aligned boxes with associated masses. Such mass functions can encode partial information for model and measurement uncertainties more accurately than the bounded error approach alone. Focal sets are propagated in the system equations using interval arithmetics and constraint satisfaction techniques, thus generalizing pure interval analysis. Applied to state estimation in dynamical systems, this approach makes it possible to compute a belief function for the system state at each time step.

This method was applied to the localization of a land vehicle based on the dynamic fusion of GPS measurements with ABS sensors and a gyrometer. The method was shown to provide more accurate estimates of vehicle position than did the bounded error method, while retaining the essential property of interval analysis of providing guaranteed computations (as the box provided by the BEE method is always one of the focal sets computed by the BSE method). The performance of the BSE approach is also slightly better than that of the particle filter method, with comparable running time. This suggests that our method might be a viable alternative to both bounded error and probabilistic Monte-Carlo approaches to non-linear state estimation, at least for the kind of applications considered here. The preliminary results reported in this paper will obviously have to be confirmed by further experiments related to more complex estimation problems.

A further advantage of the belief function approach is that it makes possible combining computed belief structures with additional information, such as digital road network data, which can also be represented in the belief function framework. A belief-function based map matching method that constructs a belief structure from a set of candidate roads extracted from an existing local map has already been presented in [34]. Further results in this direction will be reported in future publications.

References

- [1] F. Abdallah, A. Gning, and Ph. Bonnifait. Box particle filtering for non linear state estimation using interval analysis. *Automatica*, 44:807–815, 2008.
- [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*, 50(2):174–188, 2002.
- [3] I. Ashokaraj, A. Tsourdos, P. Silson, and B. White. Sensor based robot localisation and navigation: Using interval analysis and extended Kalman filter. In *5th Asian Control Conference (ASCC 2004)*, Melbourne, Australia, 2004.
- [4] Y. Bar-Shalom and W. D. Blair. *Multitarget-Multisensor Tracking: Applications and Advances, vol. III*. Artech House, 2000.
- [5] C. Baudrit and D. Dubois. Practical representations of incomplete probabilistic knowledge. *Computational Statistics and Data Analysis*, 51(1):86–108, 2006.

- [6] D. P. Bertsekas and I. B. Rhodes. Recursive state estimation for a set-membership description of uncertainty. *IEEE Transactions on Automatic Control*, 16(2):117–128, 1971.
- [7] W.-S. Chen. *Bayesian Estimation by Sequential Monte-Carlo Sampling*. PhD thesis, Ohio State University, 2004.
- [8] F. L. Chernousko. *State Estimation for Dynamic Systems*. CRC Press, Boca Raton, FL, 1994.
- [9] J. G. Cleary. Logical arithmetic. *Future computing systems*, 2:125–149, 1987.
- [10] E. Davis. Constraint propagation with interval labels. *Artificial Intelligence*, 32:281–331, 1987.
- [11] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.
- [12] A. P. Dempster. Normal belief functions and the Kalman filter. In A. K. Md. E. Saleh, editor, *Data Analysis from Statistical Foundations*, pages 65–68. Nova Science Publishers, 2001.
- [13] T. Denœux. Inner and outer approximation of belief structures using a hierarchical clustering approach. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(4):437–460, 2001.
- [14] T. Denœux and A. Ben Yaghane. Approximating the combination of belief functions using the fast moebius transform in a coarsened frame. *International Journal of Approximate Reasoning*, 31(1-2):77–101, 2002.
- [15] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte-Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10:197–208, 2000.
- [16] D. Dubois and H. Prade. A set-theoretic view of belief functions: logical operations and approximations by fuzzy sets. *International Journal of General Systems*, 12(3):193–226, 1986.
- [17] D. Dubois and H. Prade. *Possibility Theory: An approach to computerized processing of uncertainty*. Plenum Press, New-York, 1988.
- [18] S. Ferson, V. Kreinovich, L. Ginzburg, D. S. Myers, and K. Sentz. Constructing probability boxes and Dempster-Shafer structures. Technical Report SAND2002-4015, Sandia National Laboratories, Albuquerque, NM, 2003.
- [19] D. Fox. Adapting the sample size in particle filters through KLD-sampling. *The International Journal of Robotics Research*, 22(12):985–1004, 2003.
- [20] A. Gelb. *Applied Optimal Estimation*. MIT Press, 1974.
- [21] A. Gning and Ph. Bonnifait. Dynamic vehicle localization using constraints propagation techniques on intervals. a comparison with Kalman filtering. In *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

- [22] A. Gning and Ph. Bonnifait. Constraints propagation techniques on intervals for a guaranteed localization using redundant data. *Automatica*, 42(7):1167–1175, 2006.
- [23] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nord-lund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50:425–435, 2002.
- [24] L. Jaulin. Nonlinear bounded-error state estimation of continuous-time systems. *Automatica*, 38:1079–1082, 2002.
- [25] L. Jaulin, M. Kieffer, I. Braems, and E. Walter. Guaranteed nonlinear estimation using constraint propagation on sets. *International Journal of Control*, 74(18):1772–1782, 2001.
- [26] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, London, 2001.
- [27] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE, Part D, Journal of Basic Engineering*, 82:35–45, 1960.
- [28] M. Kieffer, L. Jaulin, and E. Walter. Guaranteed recursive nonlinear state estimation using interval analysis. In *Proceedings of the 37th IEEE Conference on Decision & Control, Florida, USA*, 1998.
- [29] A. Kurzhanski and I. Valyi. *Ellipsoidal Calculus for Estimation and Control*. Birkhäuser, Boston, MA, 1997.
- [30] J. D. Lowrance, T. D. Garvey, and T. M. Strat. A framework for evidential-reasoning systems. In T. Kehler et al., editor, *Proceedings of AAAI'86*, volume 2, pages 896–903, Philadelphia, August 1986. AAAI.
- [31] R. Mahler. Can the Bayesian and Dempster-Shafer approaches be reconciled? Yes. In *8th International Conference on Information Fusion*, pages 25–28, 2005.
- [32] D. Maksarov and J. P. Norton. State bounding with ellipsoidal set description of the uncertainty. *International Journal of Control*, 65(5):847–866, 1996.
- [33] R.E. Moore. *Methods and Applications of Interval Analysis*. Society for Industrial Mathematics (SIAM), 1979.
- [34] G. Nassreddine, F. Abdallah, and T. Denœux. Map matching algorithm using belief function theory. In *Proceedings of the 11th Int. Conf. on Information Fusion (FUSION '08)*, pages 995–1002, 2008.
- [35] S. Petit-Renaud and T. Denœux. Nonparametric regression analysis of uncertain and imprecise data using belief functions. *International Journal of Approximate Reasoning*, 35(1):1–28, 2004.
- [36] H. M. Regan, S. Ferson, and D. Berleant. Equivalence of methods for uncertainty propagation of real-valued random variables. *International Journal of Approximate Reasoning*, 36(1):1–30, 2004.

- [37] F. Schweiß. Recursive state estimation: unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, 13:22–28, 1968.
- [38] L. Servi and Y.C. Ho. Recursive estimation in the presence of uniformly distributed measurement noise. *IEEE Transactions on Automatic Control*, 26:563–565, 1981.
- [39] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, Princeton, N.J., 1976.
- [40] Ph. Smets and R. Kennes. The Transferable Belief Model. *Artificial Intelligence*, 66:191–243, 1994.
- [41] Ph. Smets and B. Ristic. Kalman filter and joint tracking and classification based on belief functions in the TBM framework. *Information fusion*, 8:16–27, 2007.
- [42] D. Waltz. Generating semantic descriptions from drawings of scenes with shadows. In P. H. Winston, editor, *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, New York, 1975.
- [43] H. S. Witsenhausen. Sets of possible states of linear systems given perturbed observations. *IEEE Transactions on Automatic Control*, 13(5):556–558, 1968.
- [44] R. R. Yager. Arithmetic and other operations on Dempster-Shafer structures. *Int. J. Man-Machines Studies*, 25:357–366, 1986.
- [45] R. R. Yager. Cumulative distribution functions from Dempster-Shafer belief structures. *IEEE Trans. on Systems, Man and Cybernetics B*, 34(5):2080–2087, 2004.
- [46] R. R. Yager and L. Liu, editors. *Classic Works of the Dempster-Shafer Theory of Belief Functions*. Springer, Heidelberg, 2008.
- [47] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

A Illustrative Example

To illustrate the computations performed in the BSE algorithm, let us consider as a very simple illustrative example the linear motion of a vehicle on an axis x . The vehicle position x_k at each time step k is assumed to be governed by the following equations:

$$\begin{cases} x_{k+1} = x_k + u_k \cdot T \\ z_{k+1} = x_{k+1}, \end{cases} \quad (70)$$

where u_k is the vehicle speed at time step k measured by an odometer sensor with an uncertainty of ± 3 m/s, z_k is the measured position of the vehicle given by a GPS receiver and T is the time sampling taken, equal to 1 s.

At each time step k , we receive mass functions m_k^u on the vehicle speed and m_k^z on its position. Furthermore, an initial estimate of the position x_0 at time $k = 0$ is given

in the form of a mass function m_0^x . Assume that m_0^x , m_0^u and m_1^z have the following expressions:

$$m_0^x([-6, 6]) = 1/2, \quad m_0^x([-3, 3]) = 1/2 \quad (71)$$

$$m_0^u([17, 23]) = 1, \quad (72)$$

$$m_1^z([27, 33]) = 1/2, \quad m_1^z([24, 39]) = 1/2. \quad (73)$$

We will detail the calculations needed to compute the mass function m_1^x regarding the position x_1 of the vehicle at time step $k = 1$. Because m_0^x and m_1^z have two focal intervals each and m^u has only one, the FBP algorithm has to be run four times, corresponding to the four different ways of choosing one focal interval for each of the three BSs. Calculations are detailed below.

1. $[x_0^1] = [-6, 6]$, $[z_1^1] = [27, 33]$. The forward steps of the FBP algorithm are as follows:

$$[x_1] \leftarrow (-\infty, +\infty) \cap ([-6, 6] + [17, 23]) = [11, 29]. \quad (74)$$

$$[z_1] \leftarrow [27, 33] \cap [11, 29] = [27, 29], \quad (75)$$

$$[x_1^1] \leftarrow [27, 29]. \quad (76)$$

We can verify that this interval is no longer contracted by further steps of the FBP algorithm. The corresponding mass is:

$$m_1^x([27, 29]) = 1/2 \cdot 1/2 = 1/4. \quad (77)$$

2. $[x_0^1] = [-6, 6]$, $[z_1^2] = [24, 39]$. The forward steps of the FBP algorithm are:

$$[x_1] \leftarrow (-\infty, +\infty) \cap ([-6, 6] + [17, 23]) = [11, 29]. \quad (78)$$

$$[z_1] \leftarrow [24, 39] \cap [11, 29] = [24, 29], \quad (79)$$

$$[x_1^2] \leftarrow [24, 29]. \quad (80)$$

Again, this interval is not contracted by further steps of the FBP algorithm. The corresponding mass is:

$$m_1^x([24, 29]) = 1/2 \cdot 1/2 = 1/4. \quad (81)$$

3. $[x_0^2] = [-3, 3]$, $[z_1^1] = [27, 33]$. The forward steps of the FBP algorithm are:

$$[x_1] \leftarrow (-\infty, +\infty) \cap ([-3, 3] + [17, 23]) = [14, 26]. \quad (82)$$

$$[z_1] \leftarrow [27, 33] \cap [14, 26] = \emptyset, \quad (83)$$

$$[x_1^3] \leftarrow \emptyset. \quad (84)$$

We thus have

$$[x_1^3] = \emptyset, \quad (85)$$

and

$$m_1^x(\emptyset) = 1/2 \cdot 1/2 = 1/4. \quad (86)$$

4. $[x_0^2] = [-3, 3]$, $[z_1^2] = [24, 39]$. The predicted state at $k = 1$ is the same as above: $[x_1] = [14, 26]$, and the intersection with $[z_1^2]$ yields:

$$[z_1] \leftarrow [24, 39] \cap [14, 26] = [24, 26]. \quad (87)$$

We thus have $[x_1^4] = [24, 26]$. Again, this interval is not contracted by the FBP algorithm. The corresponding mass is:

$$m_1^x([24, 26]) = 1/2 \cdot 1/2 = 1/4. \quad (88)$$

We thus obtain a non normal mass function with 4 focal intervals. After normalization, we get:

$$m_1^x([27, 29]) = 1/3, \quad m_1^x([24, 29]) = 1/3, \quad m_1^x([24, 26]) = 1/3. \quad (89)$$

The interval expectation of x_1 is

$$[\mathbb{E}](m_1^x) = ([27, 29] + [24, 29] + [24, 26])/3 = [25, 28], \quad (90)$$

and its pignistic expectation is

$$\mathbb{E}(m_1^x) = (28 + 26.5 + 25)/3 = 26.5. \quad (91)$$

Biographies

Ghalia Nassreddine was born in Makenh, Lebanon, in 1984. She received the Engineer diploma from the Lebanese university, Tripoli, Lebanon, in 2005, and M.S. degree from the Université de Technologie de Compiègne (UTC), France, in 2006. Since 2006, she has been a Ph.D student at UTC.

Fahed Abdallah was born in Lebanon, on August 18, 1976. He received the Dipl. Ing and the M.S. degrees in 1999 and 2000 from the Lebanese University, Beirut, Lebanon, and the Ph.D degree in 2004 from the Université de Technologie de Troyes, France, all in electrical and computer engineering. Since 2005, he has been an Associate Professor with the HEUDIASYC laboratory at the Université de Technologie de Compiègne, France. His current research interests involve state estimation for dynamic models based on multisensor fusion, statistical estimation and decision theories, pattern recognition and belief functions theory.

Thierry Denœux graduated in 1985 as an engineer from the Ecole Nationale des Ponts et Chaussées in Paris, and received a doctorate from the same institution in 1989. Currently, he is Full Professor with the Department of Information Processing Engineering at the Université de Technologie de Compiègne, France. His research interests concern belief functions theory, fuzzy data analysis and, more generally, the management of imprecision and uncertainty in data analysis, pattern recognition and information fusion. He is the Editor-in-Chief of the *International Journal of Approximate Reasoning*, and a member of the editorial board of *Fuzzy Sets and Systems*.