State estimation and fault detection using box particle filtering with stochastic measurements

Joaquim Blesa [1], **Françoise Le Gall**[2], **Carine Jauberthie**[2,3] and **Louise Travé-Massuyès**[2]
[1]Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas, 4-6, 08028 Barcelona, Spain
e-mail: joaquim.blesa@upc.edu
[2]CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
Univ de Toulouse, LAAS, F-31400 Toulouse, France
e-mail: legall,cjaubert,louise@laas.fr
[3]Univ de Toulouse, UPS, LAAS, F-31400 Toulouse

Abstract

In this paper, we propose a box particle filtering algorithm for state estimation in nonlinear systems whose model assumes two types of uncertainties: stochastic noise in the measurements and bounded errors affecting the system dynamics.These assumptions respond to situations frequently encountered in practice. The proposed method includes a new way to weight the box particles as well as a new resampling procedure based on repartitioning the box enclosing the updated state. The proposed box particle filtering algorithm is applied in a fault detection schema illustrated by a sensor network target tracking example.

1 Introduction

For various engineering applications, system state estimation plays a crucial role. Kalman filtering (KF) has been widely used in the case of stochastic linear systems. The Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are KF's extensions for nonlinear systems. These methods assume unimodal, Gaussian distributions. On the other hand, Particle Filtering (PF) is a sequential Monte Carlo Bayesian estimator which can be used in the case of non-Gaussian noise distributions. Particles are punctual states associated with weights whose likelihoods are defined by a statistical model of the observation error. The efficiency and accuracy of PF depend on the number of particles used in the estimation and propagation at each iteration. If the number of required particles is too large, a real implementation is unsuitable and this is the main drawback of PF. Several methods have been proposed to overcome these shortcomings, mainly based on variants of the resampling stage or different ways to weight the particles ([1]).

Recently, a new approach based on *box particles* was proposed by [2; 3]. The Box Particle Filter handles box states and bounded errors. It uses interval analysis in the state update stage and constraint satisfaction techniques to perform measurement update. The set of box particles is interpreted as a mixture of uniform pdf's [4]. Using box particles has been shown to control quite efficiently the number of required particles, hence reducing the computational cost and providing good results in several experiments.

In this paper, we take into account the box particle filtering ideas but consider that measurements are tainted by stochastic noise instead of bounded noise. The errors affecting the system dynamics are kept bounded because this type uncertainty really corresponds to many practical situations, for example tolerances on parameter values. Combining these two types of uncertainties following the seminal ideas of [5] and [6] within a particle filter schema is the main issue driving the paper. This issue is different from the one addressed in [7] in which the focus is put on Bernouilli filters able to deal with data association uncertainty. The proposed method includes a new way to weight the box particles as well as a new resampling procedure based on repartitioning the box enclosing the updated state.

The paper is organized as follows. Section 2 describes the problem formulation. A summary of the Bayesian filtering is presented and the box-particle approach is introduced. The main steps of this approach are developed in section 3. Section 4 and 5 are devoted to the repartitioning of the boxes and the computation of the weight of the box particles in order to control the number of boxes. In section 6 the box particle filter is used for state estimation and fault detection; the results obtained with the proposed method for a target tracking in a sensor network are presented in section 7. Conclusion and future work are overviewed in the last section.

2 Problem formulation

We consider nonlinear dynamic systems represented by discrete time state-space models relating the state $\mathbf{x}(k)$ to the measured variables $\mathbf{y}(k)$

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

$$\mathbf{y}(k) = h(\mathbf{x}(k)) + \mathbf{e}(k), k = 0, 1, \ldots \quad (2)$$

where $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ are nonlinear functions, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ is the system input, $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ is the system output, $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ is the state-space vector, $\mathbf{e}(k) \in \mathbb{R}^{n_y}$ is a stochastic additive error that includes the measurement noise and discretization error and is specified by its known pdf p_e. $\mathbf{v}(k) \in \mathbb{R}^{n_x}$ is the process noise.

In this work the process noise is assumed bounded $|v_i(k)| \leq \sigma_i$ with $i = 1, \ldots, n_x$, i.e $p_v \sim \mathcal{U}([\mathcal{V}])$, where $[\mathcal{V}] = [-\sigma_1, \sigma_1] \times \cdots \times [-\sigma_{n_x}, \sigma_{n_x}]$.

2.1 Bayesian filtering

Given a vector of available measurements at instant k: $\mathbf{Y}(k) = \{\mathbf{y}(i), i = 1, ..., k\}$, $\mathbf{Y}(0) = \mathbf{y}(0)$, the Bayesian

solution to compute the posterior distribution $p(\mathbf{x}(k)|\mathbf{Y}(k))$ of the state vector at instant $k+1$, given past observations $\mathbf{Y}(k)$ is given by (Gustafsson 2002):

$$p(\mathbf{x}(k+1)|\mathbf{Y}(k)) = \int_{\mathbb{R}^{n_x}} p(\mathbf{x}(k+1)|\mathbf{x}(k))p(\mathbf{x}(k)|\mathbf{Y}(k))dx(k) \quad (3)$$

where the posterior distribution $p(\mathbf{x}(k)|\mathbf{Y}(k))$ can be computed by

$$p(\mathbf{x}(k)|\mathbf{Y}(k)) = \frac{1}{\alpha(k)} p(\mathbf{y}(k)|\mathbf{x}(k))p(\mathbf{x}(k)|\mathbf{Y}(k-1)) \quad (4)$$

where $\alpha(k)$ is a normalization constant, $p(\mathbf{y}(k)|\mathbf{x}(k))$ is the likelihood function that can be computed from (2) as:

$$p(\mathbf{y}(k)|\mathbf{x}(k)) = p_e(\mathbf{y}(k) - h(\mathbf{x}(k)) \quad (5)$$

and $p(\mathbf{x}(k)|\mathbf{Y}(k-1))$ is the prior distribution.

Equations (5), (4) and (3) can be computed recursively given the initial value of $p(\mathbf{x}(k)|\mathbf{Y}(k-1))$ for $k=0$ denoted as $p(\mathbf{x}(0))$ that represents the prior knowledge about the initial state.

2.2 Objective

Considering the assumptions of our problem, we adopt a particle filtering schema which is well-known for solving numerically complex dynamic estimation problems involving nonlinearities. However, we propose to use box particles and to base our method on the interval framework. Box particle filters have been demonstrated efficient, in particular to reduce the number of particles that must be considered to reach a reasonable level of approximation [2].

Let's consider the current state estimate $\mathcal{X}(k)$ as a set, denoted by $\{\mathcal{X}(k)\}$, that is approximated by N_k disjoint boxes

$$[\mathbf{x}(k)]^i \quad i = 1, \cdots, N_k \quad (6)$$

where $[\mathbf{x}(k)]^i = [\underline{\mathbf{x}(k)^i}, \overline{\mathbf{x}(k)^i}]$, with $\underline{\mathbf{x}(k)^i}, \overline{\mathbf{x}(k)^i} \in \mathbb{R}^{n_x}$. The width of every box is smaller or equal to a given accuracy for every component, i.e

$$\overline{x_j(k)^i} - \underline{x_j(k)^i} \leq \delta_j \quad i = 1, \cdots, N_k, \quad j = 1, \ldots, n_x \quad (7)$$

where δ_j is the predetermined minimum accuracy for every component j.

Moreover, every box $[\mathbf{x}(k)]^i$ is given a prior probability denoted as

$$P([\mathbf{x}(k)]^i|\mathbf{Y}(k-1)) \quad i = 1, \cdots, N_k \quad (8)$$

with

$$\sum_{i=1}^{N_k} P([\mathbf{x}(k)]^i|\mathbf{Y}(k-1)) \geq \gamma \quad (9)$$

where $\gamma \in [0,1]$ is a confidence threshold.

Then, given a new output measurement $\mathbf{y}(k)$, the problem that we consider in this paper is:

- to compute the state estimate $\mathcal{X}(k+1)$,
- to decide about the number N_{k+1} of disjoint boxes of the approximation of $\mathcal{X}(k+1)$, each with accuracy smaller or equal to δ_j,
- to provide the prior probabilities associated to the particles of the new state estimation set

$$P([\mathbf{x}(k+1)]^i|\mathbf{Y}(k)) \quad i = 1, \cdots, N_{k+1} \quad (10)$$

3 Interval Bayesian formulation

This section deals with the evaluation of the Bayesian solution of the state estimation problem considering bounded state boxes (6).

3.1 Measurement update

Whereas each particle is defined as a box by (6), the measurement is tainted with stochastic uncertainty defined by the pdf p_e. The weight $w(k)^i$ associated to a box particle is updated by the posterior probability $P([\mathbf{x}(k)]^i|\mathbf{Y}(k))$:

$$w(k)^i = \frac{1}{\Lambda(k)} P([\mathbf{x}(k)]^i|\mathbf{Y}(k-1)) p_e(\mathbf{y}(k) - h([\mathbf{x}(k)]^i)$$
$$= \frac{1}{\Lambda(k)} P([\mathbf{x}(k)]^i|\mathbf{Y}(k-1)) \int_{x(k)\in[\mathbf{x}(k)]^i} p_e(\mathbf{y}(k) - h(x(k)))\, dx(k) \quad (11)$$
$$i = 1, \ldots, N_k$$

where the normalization constant $\Lambda(k)$ is given by

$$\Lambda(k) = \sum_{i=1}^{N_k} P([\mathbf{x}(k)]^i|\mathbf{Y}(k-1)) \int_{x(k)\in[\mathbf{x}(k)]^i} p_e(\mathbf{y}(k) - h(x(k)))\, dx(k) \quad (12)$$

then

$$\sum_{i=1}^{N_k} w(k)^i = 1 \quad (13)$$

The deduction of the measurement update equation (11) from the particle filtering update equation (4) is detailed in the Appendix for $n_x = 1$, without the loss of generality. The principle of the proof is that the point particles are grouped into particle groups inside boxes, then the posterior probability of a box can be approximated by the sum of posterior probabilities of the point particles when the number of these particles tends to infinity.

3.2 State update

This step is similar to the state update state as in [2] and [3]. Hence, we have:

$$p(\mathbf{x}(k+1)|\mathbf{Y}(k)) \approx \sum_{i=1}^{N_k} w(k)^i \mathcal{U}_{[f]([\mathbf{x}(k)]^i, \mathbf{u}(k), [\mathbf{v}(k)])} \quad (14)$$

The interval boxes $[\mathbf{x}(k+1)|\mathbf{x}(k)]^i$ are computed from (1) using interval analysis as follows,

$$[\mathbf{x}(k+1)|\mathbf{x}(k)]^i \approx [f]([\mathbf{x}(k)]^i, \mathbf{u}(k), [\mathbf{v}(k)]) \quad (15)$$

The update interval boxes inherit the weights $w(k)^i$ of their mother boxes $[\mathbf{x}(k)]^i$ $i = 1, \ldots, N_k$.

4 Resampling

Once the updated boxes $[\mathbf{x}(k+1)|\mathbf{x}(k)]^i$ and their associated weights $w(k)^i$ have been computed, the objective is to compute a new set of disjoint boxes. This corresponds to the resampling step of the conventional particle filter.

4.1 Repartitioning

We assume that the new boxes are of the same size, that they cover the whole space defined by the union of the updated boxes $[\mathbf{x}(k+1)|\mathbf{x}(k)]^i\ i=1,\ldots,N_k$, and that their weight is proportional to the weight of the former boxes.

For this purpose, a support box set $\mathcal{Z}$ is computed as the minimum box such that

$$\mathcal{Z} \supseteq \bigcup_{i=1}^{N_k} [\mathbf{x}(k+1)|\mathbf{x}(k)]^i. \tag{16}$$

$\mathcal{Z}$ is partitioned into M disjoint boxes of the same size

$$[\mathbf{z}]^i \quad i=1,\cdots,M \tag{17}$$

where $[\mathbf{z}]^i = [\underline{\mathbf{z}^i}, \overline{\mathbf{z}^i}]$, $\underline{\mathbf{z}^i}, \overline{\mathbf{z}^i} \in \mathbb{R}^{n_x}$, and

$$\overline{z_j^i} - \underline{z_j^i} = \varepsilon_j \quad i=1,\cdots,M \quad j=1,\ldots,n_x. \tag{18}$$

The box component widths are computed as

$$\varepsilon_j = \frac{\overline{\mathcal{Z}_j} - \underline{\mathcal{Z}_j}}{m_j} \quad j=1,\ldots,n_x \tag{19}$$

where m_j is the number of intervals along dimension j computed as

$$m_j = \lceil \frac{\overline{\mathcal{Z}_j} - \underline{\mathcal{Z}_j}}{\delta_j} \rceil \quad j=1,\ldots,n_x \tag{20}$$

where $\lceil . \rceil$ indicates the ceiling function and δ_j the minimum accuracy for every state component j defined in Section 2.2. In this way, we guarantee that

$$\varepsilon_j \leq \delta_j \quad j=1,\ldots,n_x \tag{21}$$

Finally, the number M of boxes of the uniform grid partition is given by

$$M = \prod_{j=1}^{n_x} m_j \tag{22}$$

Once the new boxes $[\mathbf{z}]^i$ have been computed, the weight of the new boxes w_z^i can be computed as

$$w_z^i = \sum_{j=1}^{N_k} \left(\frac{\prod_{l=1}^{n_x} |[x_l(k+1)|\mathbf{x}(k)]^j \bigcap [z_l]^i|}{\prod_{l=1}^{n_x} |[x_l(k+1)|\mathbf{x}(k)]^j|} w(k)^j \right) \tag{23}$$

$$i=1,\ldots,M$$

where $[v_l]^i$ refers to the l-th component of the vector $[\mathbf{v}]^i$ and the interval width $\overline{x_l} - \underline{x_l}$ is denoted by $|[x_l]|$ for more compactness. The new weights fulfill

$$\sum_{i=1}^{M} w_z^i = \sum_{i=1}^{N_k} w(k)^i = 1 \tag{24}$$

The new weights w_z^i in (4.1) can be computed efficiently using Algorithm 1. This algorithm searches the number N_{inter} of boxes of $\mathcal{Z}$ that intersect every $[\mathbf{x}(k+1)|\mathbf{x}(k)]^j$. Then, the weight $w(k)^j$ is distributed proportionally to the volume of the intersection between the updated boxes $[x(k+1)|x(k)]^j$ and each of the N_{inter} boxes of $\mathcal{Z}$ that have a non-empty intersection.

Algorithm 1 Weights of the new boxes.

Algorithm Weights-new-boxes $(\mathcal{Z}, [\mathbf{x}(k+1)|\mathbf{x}(k)]^1, \ldots, [\mathbf{x}(k+1)|\mathbf{x}(k)]^{N_k}, w(k)^1, \ldots w(k)^{N_k})$
  $w_z^i \leftarrow 0\ i=1,\ldots,M$
  for $j=1,\ldots,N_k$ **do**
    $[N_{inter}, \mathbf{V}_{inter}] = intersec([\mathbf{x}(k+1)|\mathbf{x}(k)]^j, \mathcal{Z})$
    for $h=1,\ldots,N_{inter}$ **do**
      $i = \mathbf{V}_{inter}(h)$
      $w_z^i = w_z^i + \frac{\prod_{l=1}^{n_x} |[x_l(k+1)|\mathbf{x}(k)]^j \bigcap [z_l]^i|}{\prod_{l=1}^{n_x} |[x_l(k+1)|\mathbf{x}(k)]^j|} w(k)^j$
    end for
  end for
  Return $(w_z^1, \ldots, w_z^M)$
endAlgorithm

4.2 Controlling the number of boxes

Once the new disjoint boxes and their associated weights have been computed, the associated weights can be used to select the set of boxes that are worth pushing forward through the next iteration. This is performed by selecting the boxes with highest weights and discarding the others. In order to fulfill the confidence threshold criterium (9) proposed in Section 2.2, Algorithm 2 is proposed. The set W_z of weights w_z^i associated to the boxes $[\mathbf{z}]^i$ is defined as

$$W_z = \{w_z^1, \ldots, w_z^M\}. \tag{25}$$

Given a desired confidence threshold γ, the M disjoint boxes $[\mathbf{z}]^i$ that compose the uniform grid partition of $\mathcal{Z}$ and vector W_z with the associated weights, Algorithm 2 determines the minimum number N_{k+1} of boxes $[\mathbf{z}]^i$ with highest weights w_z^i that fulfill

$$\sum_{i=1}^{N_{k+1}} w_z^i \geq \gamma \tag{26}$$

The new state estimate $\mathcal{X}(k+1)$ is approximated by this set of N_{k+1} boxes and their prior probability by

$$P([\mathbf{x}(k+1)]^i|\mathbf{Y}(k)) \approx W_{k+1}^i \quad i=1,\ldots,N_{k+1}. \tag{27}$$

where W_{k+1}^i are the N_{k+1} highest weights of W_z associated with the disjoint boxes $[\mathbf{x}(k+1)]^i$, $i=1,\cdots,N_{k+1}$, that approximate $\mathcal{X}(k+1)$. W_{k+1}^i can be referred as the *a priori* weights.

Algorithm 2 State update at step $k+1$ with confidence threshold γ.

Algorithm State-update($[\mathbf{z}]^1, \ldots, [\mathbf{z}]^M$,$W_z$,$\gamma$)
  $\gamma_c \leftarrow 0, \{\mathcal{X}(k+1)\} \leftarrow \{\emptyset\}, W_{k+1} \leftarrow \{\emptyset\}, N_{k+1} \leftarrow 0$
  while $\gamma_c < \gamma$ **do**
    $[value, pos] = \max(W_z)$
    $addbox(\mathcal{X}(k+1), [\mathbf{z}]^{pos})$
    $addelement(W_{k+1}, value)$
    $\gamma_c = \gamma_c + value$
    $W_z(pos) \leftarrow 0$
    $N_{k+1} \leftarrow N_{k+1} + 1$
  endwhile
  Return $(\mathcal{X}(k+1), W_{k+1}, N_{k+1})$
endAlgorithm

This algorithm generates a set of state boxes $\{\mathcal{X}(k+1)\}$ a list of weights W^i_{k+1}, a cumulative weight variable γ_c, and a cardinality variable N_{k+1}. At the beginning of the algorithm, the state boxes and weight list are initialized as empty sets and cumulative weight and cardinality variable are initialized to zero. The loop "while" operates as a sorting, eliminating the boxes with smallest weights so that the cumulative sum of the boxes with largest weights is greater or equal to the threshold γ. If the state space is not bounded, the threshold $0 < \gamma < 1$ does not guarantee a bounded number of boxes in a worst-case scenario in which the measurements do not emphasize some particles against others. In this case, a maximum number of particles $Nmax$ should be imposed.

5 State estimation and fault detection

5.1 State estimation

Once the set of N_{k+1} disjoint boxes $[\mathbf{x}(k+1)]^i$, $i = 1, \cdots, N_{k+1}$, that approximate $\mathcal{X}(k+1)$ and their associated *a priori* weights W^i_{k+1} have been computed, their measurement updated weights $w(k+1)^i$ are obtained using (11). Then, according to [2], the state at instant $k+1$ is approximated by

$$\hat{\mathbf{x}}(k+1) = \sum_{i=1}^{N_{k+1}} w(k+1)^i \mathbf{x}^i_0(k+1) \tag{28}$$

where $\mathbf{x}^i_0(k+1)$ is the center of the particle box $[\mathbf{x}(k+1)]^i$.

Algorithm 3 summarizes the whole state estimation procedure.

Algorithm 3 State estimation

Algorithm State estimation
  Initialize $\mathcal{X}(0)$, N_0 and $P([\mathbf{x}(k)]^i|\mathbf{Y}(k-1))_{k=0,i=1\ldots N_0}$
  for $k = 1, \ldots, end$ **do**
    Obtain Input/Output data $\{\mathbf{u}(k), \mathbf{y}(k)\}$
    Measurement update
      compute $\Lambda(k)$ using Eq. (12)
      compute $w(k)^i$ using Eq.(11) $i = 1 \ldots N_0$
    State estimation
      compute $\hat{\mathbf{x}}(k)$ using (28)
    State update
      compute $[\mathbf{x}(k+1)|\mathbf{x}(k)]^i$ $i = 1 \ldots N_0$ using (15)
      compute $\mathcal{Z}$ that fulfils (16)
      compute disjoint boxes $[\mathbf{z}]^i$ $i = 1, \cdots, M$ of (17)
      compute weights w^i_z using Algorithm 1
      compute new state estimation using Algorithm 2
        N_{k+1} disjoint boxes that approximate $\mathcal{X}(k+1)$
        Prior probabilities given by weights W_{k+1}
  end for
endAlgorithm

5.2 Fault detection

In our framework, fault detection can be formulated as detecting inconsistencies based on the state estimation. To do so, we propose the two following indicators:

- Abrupt changes in the state estimation provided by (28) from instant $k-1$ to instant k, i.e. abnormal high values of $\sqrt{(\hat{\mathbf{x}}(k) - \hat{\mathbf{x}}(k-1))(\hat{\mathbf{x}}(k) - \hat{\mathbf{x}}(k-1))^T}$
- Abnormal low sum of the unnormalized posterior probability of all the particles at instant k, which means that all the particles have been penalized by the current measurements. This abnormality can be checked by thresholding $\Lambda(k)$ defined in (12).

If enough representative fault free data are available, the indicators defined above can be determined by means of thresholds computed with these data. For example, the threshold that defines the abnormal abrupt change in state estimation can be computed as

$$\Delta\hat{\mathbf{x}}^{max} = \beta_1 \max_{i=2,\cdots,L} \sqrt{(\hat{\mathbf{x}}(i) - \hat{\mathbf{x}}(i-1))(\hat{\mathbf{x}}(i) - \hat{\mathbf{x}}(i-1))^T} \tag{29}$$

where L is the length of the fault free scenario and $\beta_1 > 1$ a tuning parameter. Then the fault detection test consists in checking at each instant k if

$$\sqrt{(\hat{\mathbf{x}}(k) - \hat{\mathbf{x}}(k-1))(\hat{\mathbf{x}}(k) - \hat{\mathbf{x}}(k-1))^T} > \Delta\hat{\mathbf{x}}^{max} \tag{30}$$

In a similar way, threshold Λ^{min} that defines the minimum expected unnormalized posterior probability can be computed as

$$\Lambda^{min} = \beta_2 \min_{i=2,\cdots,L} (\Lambda(i)) \tag{31}$$

where $\Lambda(i)$ is determined using (12) and $0 < \beta_2 < 1$ is a tuning parameter. Then the fault detection test consists in checking at each instant k if

$$\Lambda(k) < \Lambda^{min} \tag{32}$$

6 Application example

In this section a target tracking in a sensor network example presented in [8] is used to illustrated the state estimation method presented above. The problem consists of three sensors and one target moving in the horizontal plane. Each sensor can measure distance to the target, and by combining these a position fix can be computed. Fig. 1 depicts a scenario with a trajectory and a certain combination of sensor locations (S_1, S_2 and S_3).

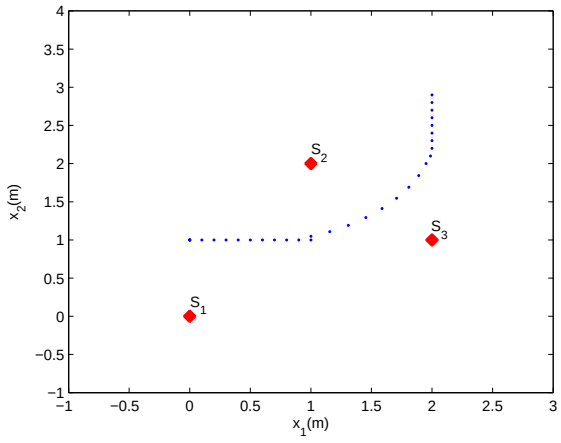

Figure 1: Target true trajectory and sensor positions in the bounded horizontal plane

The behaviour of the system can be described by the following discrete time state-space model:

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + T_s \begin{pmatrix} v_1(k) \\ v_2(k) \end{pmatrix} \quad (33)$$

$$\begin{pmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{pmatrix} = \begin{pmatrix} \sqrt{(x_1(k)-S_{1,1})^2+(x_2(k)-S_{1,2})^2} \\ \sqrt{(x_1(k)-S_{2,1})^2+(x_2(k)-S_{2,2})^2} \\ \sqrt{(x_1(k)-S_{3,1})^2+(x_2(k)-S_{3,2})^2} \end{pmatrix} + \begin{pmatrix} e_1(k) \\ e_2(k) \\ e_3(k) \end{pmatrix} \quad (34)$$

where $x_1(k)$ and $x_2(k)$ are the object coordinates bounded by $-1 \leq x_1(k) \leq 3$ and $-1 \leq x_2(k) \leq 4 \; \forall k \geq 0$. $T_s = 0.5s$ is the sampling time, $v_1(k)$ and $v_2(k)$ are the speed components of the target that are unknown but considered bounded by the maximum speed $\sigma_v = 0.4m/s$ ($|v_1(k)| \leq \sigma_v$ and $|v_2(k)| \leq \sigma_v$). $y_1(k)$, $y_2(k)$ and $y_3(k)$ are the distances measured by the sensors. $S_{i,j}$ denotes the component j of the location of sensor i. $e_1(k)$, $e_2(k)$ and $e_3(k)$ are the the stochastic measurement additive errors $p_{e_i} \sim N(0, \sigma_i)$ with $\sigma_1 = \sigma_2 = \sigma_3 = \sqrt{0.05}m$.

Fig. 2 shows the evolution of the real sensor distances and measurements in the target trajectory scenario depicted in Fig. 1.

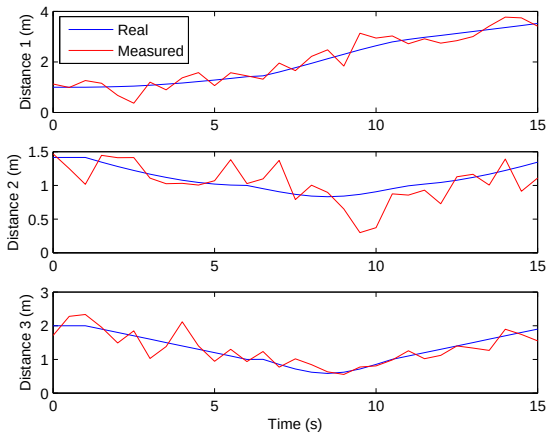

Figure 2: Real and measured distances from the target to the sensors

In order to apply the state estimation methodology presented above, a minimum accuracy $\delta_1 = \delta_2 = \delta = 0.2m$ has been selected for both components. No a priori information has been used in the initial state. Then, a uniform grid of disjoint boxes with the same weights and component widths $\varepsilon_1 = \varepsilon_2 = \delta$ that covers all the bounded coordinates $-1 \leq x_1 \leq 3$ and $-1 \leq x_2 \leq 4$ has been chosen as initial state $\mathcal{X}(0)$. Posterior probabilities of the boxes have been approximated by weights $w(k)^i$ computed using the new sensor distances measurements in (4.1). State update has been computed considering speed bounds in (33). The new boxes have been rearranged considering the minimum accuracy δ and their associated weights have been computed using (4.1). Finally, Algorithm 2 with threshold $\gamma = 1$ has been applied to reduce the number of boxes.

Figs. 3 and 4 depict the box weights and their contours using measurement $y_1(1)$ (up) and all the measurements at instant $k = 1$ ($y_1(1), y_2(1)$ and $y_3(1)$) (down). Fig. 5 depicts the box weights and their contours using the measurements at hand at instant $k = 2$.

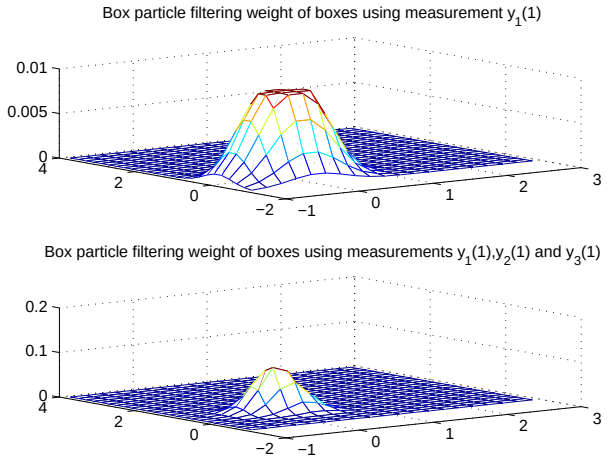

Figure 3: Box weights using measurement $y_1(k)$ (up) and measurements $(y_1(k), y_2(k), y_3(k))^T$ (down)

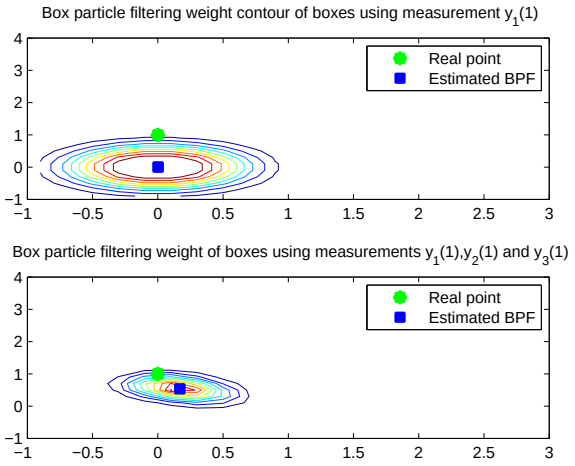

Figure 4: Box weight contours using measurement $y_1(k)$ (up) and measurements $(y_1(k), y_2(k), y_3(k))^T$ (down)

The real trajectory and the one estimated using (28) are shown in Fig. 6.

Finally, different additive sensor faults have been simulated and satisfactory results of the fault detection tests (30) and (32) have been obtained for faults bigger than $0.5m$ using thresholds $\Delta\hat{\mathbf{x}}^{max}$ and Λ^{min} computed with (29) and (31)with $L = 3200$, $\beta_1 = 1.1$ and $\beta_2 = 0.9$.

Fig. 7 shows the real trajectory and the one estimated using (28) when an additive fault of $+0.5m$ affects sensor S_1 at time $k = 22$. The behaviour of fault detection tests (30) and (32) is depicted in Fig. 8. As seen in this figure, both thresholds are violated at time instant $k = 22$ and therefore the fault is detected at this time instant.

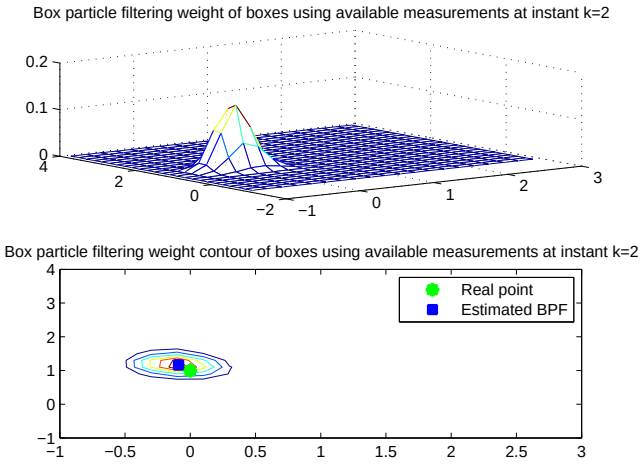

Figure 5: Box weights (up) and Box weights contours (down) at instant $k = 2$

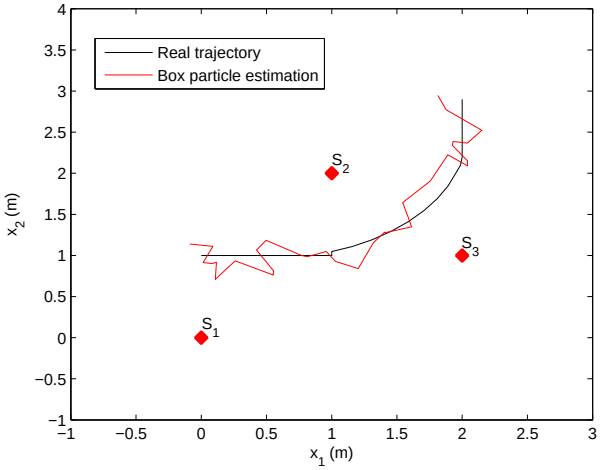

Figure 6: Trajectories

7 Conclusion and perspectives

A Box particle algorithm has been proposed for estimation and fault detection in the case of nonlinear systems with stochatic and bounded uncertainties. Using this method in the case of a target tracking sensor networks illustrates its feasibility. It has been shown how the measurement update state for the box particle is derived from the particle case. However convergence and stability of this filter have to be proved. Resampling unfortunatly drops information and waives guaranteed results that characterize interval analysis based solutions. However without resampling the particle filter suffers from sample depletion. This is the reason why resampling is a critical issue in particle filtering (Gustafsson 2002). This approach has to be compared to other PF variants which reduce the number of particles [2] and further investigations concerning resampling are required, in particular if we want to take better benefit of the interval based approach.

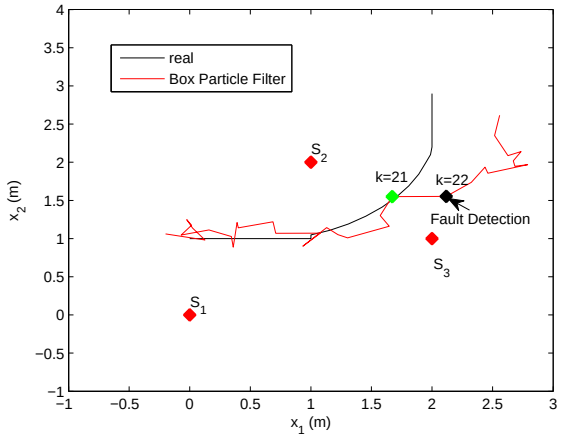

Figure 7: Trajectories in fault scenario

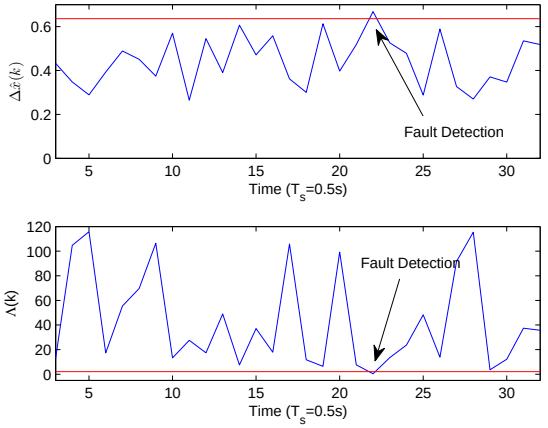

Figure 8: Fault indicators and thresholds in the fault scenario

Acknowledgments

This work has been partially funded by the Spanish Ministry of Science and Technology through the Project ECOCIS (Ref. DPI2013-48243-C2-1-R) and Project HARCRICS (Ref. DPI2014-58104-R).

A Demonstration of Measurement update: "From particles to boxes"

A.1 Particle filtering

Consider the particles $\{x(k)^j\}_{j=1}^N$ uniformly distributed in $x(k)^j \in [\underline{x(k)}, \overline{x(k)}] \ \ \forall j = 1, \ldots, N$ where $\underline{x(k)}, \overline{x(k)} \in \mathbb{R}$. Then according to [1] the relative posterior probability for each particle is approximated by

$$P(\mathbf{x}(k)^j|\mathbf{Y}(k)) \approx \frac{1}{c(k)} P(\mathbf{x}(k)^j|\mathbf{Y}(k-1)) p_e(y(k) - h(x(k)^j)) \tag{35}$$

with

$$c(k) = \sum_{j=1}^{N} P(\mathbf{x}(k)^j|\mathbf{Y}(k)) \tag{36}$$

A.2 Grouping particles

If we group the N particles in N_g groups of ΔN elements

$$\{x(k)^j\}_{j=1}^{N} = \bigcup_{i=1}^{N_g} \{x(k)^l\}_{l=1+(i-1)\Delta N}^{i\Delta N} \tag{37}$$

with $N_g = \frac{N}{\Delta N}$

If we select the groups of points in such a way that

$$\{x(k)^l\}_{l=1+(i-1)\Delta N}^{i\Delta N} \in [x(k)]^i \quad \forall i = 1, \ldots, N_g \tag{38}$$

where

$$[x(k)]^i = [\underline{x(k)} + (i-1)\Delta L, \underline{x(k)} + i\Delta L] \tag{39}$$

with

$$\Delta L = \frac{\overline{x(k)} - \underline{x(k)}}{N_g} \tag{40}$$

If the number of particles $N \to \infty$ and therefore $\Delta N \to \infty$

$$P([\mathbf{x}(k)]^i|\mathbf{Y}(k)) \approx \sum_{j=1+(i-1)\Delta N}^{i\Delta N} P(\mathbf{x}(k)^j|\mathbf{Y}(k)) \tag{41}$$

according to (35)

$$P([\mathbf{x}(k)]^i|\mathbf{Y}(k)) \approx \frac{\sum_{j=1+(i-1)\Delta N}^{i\Delta N} P(\mathbf{x}(k)^j|\mathbf{Y}(k-1)) p_e(y(k) - h(x(k)^j))}{\sum_{l=1}^{N_g} \sum_{j=1+(l-1)\Delta N}^{l\Delta N} P(\mathbf{x}(k)^j|\mathbf{Y}(k-1)) p_e(y(k) - h(x(k)^j))} \tag{42}$$

If we consider the particles in the same group i have the same prior probabilities, then:

$$p(\mathbf{x}(k)^j|\mathbf{Y}(k-1)) = \frac{P([\mathbf{x}(k)]^i|\mathbf{Y}(k-1))}{\Delta N} \quad \forall j = 1 + (i-1)\Delta N, \ldots, i\Delta N \tag{43}$$

and (42) leads to

$$P([\mathbf{x}(k)]^i|\mathbf{Y}(k)) \approx \frac{P([\mathbf{x}(k)]^i|\mathbf{Y}(k-1)) \sum_{j=1+(i-1)\Delta N}^{i\Delta N} p_e(y(k) - h(x(k)^j))}{\sum_{l=1}^{N_g} (P([\mathbf{x}(k)]^l|\mathbf{Y}(k-1)) \sum_{j=1+(l-1)\Delta N}^{l\Delta N} p_e(y(k) - h(x(k)^j)))} \tag{44}$$

If the N particles are uniformly distributed in the interval $[\underline{x(k)}, \overline{x(k)}]$, i.e

$$x(k)^j - x(k)^{j-1} = \Delta x(k) \quad \forall j = 2, \ldots, N \tag{45}$$

where

$$\Delta x(k) = \frac{\overline{x(k)} - \underline{x(k)}}{N} = \frac{\Delta L}{\Delta N} \tag{46}$$

Then

$$\sum_{j=1+(i-1)\Delta N}^{i\Delta N} p_e(y(k) - h(x(k)^j))\Delta x(k) \approx \int_{(1+(i-1)\Delta N)\Delta x(k)}^{(i\Delta N)\Delta x(k)} p_e(y(k) - h(x(k)))dx(k) \approx \int_{x(k)\in[x(k)]^i} p_e(y(k) - h(x(k)))dx(k) \tag{47}$$

Finally, multiplying the numerator and denominator of equation (44) by Δx, we obtain the particle box measurement update equation

$$P([\mathbf{x}(k)]^i|\mathbf{Y}(k)) \approx \frac{P([\mathbf{x}(k)]^i|\mathbf{Y}(k-1)) \int_{x(k)\in[x(k)]^i} p_e(y(k) - h(x(k)))dx(k)}{\sum_{l=1}^{N_g} (P([\mathbf{x}(k)]^l|\mathbf{Y}(k-1)) \int_{x(k)\in[x(k)]^l} p_e(y(k) - h(x(k)))dx(k))} \tag{48}$$

that corresponds to the equation (11) with

$$\Lambda(k) = \sum_{l=1}^{N_g} (P([\mathbf{x}(k)]^l|\mathbf{Y}(k-1)) \int_{x(k)\in[x(k)]^l} p_e(y(k) - h(x(k)))dx(k)) \tag{49}$$

References

[1] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.J. Nordlund. Particle filters for positioning, navigation, and tracking. *Signal Processing, IEEE Transactions on*, 50(2):425–437, 2002.

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

[3] A. Doucet, N. De Freitas, and N. Gordon. *An introduction to sequential Monte Carlo methods*. Springer, 2001.

[4] A. Gning, L. Mihaylova, and F. Abdallah. Mixture of uniform probability density functions for non linear state estimation using interval analysis. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8. IEEE, 2010.

[5] R.M. Fernández-Cantí, S. Tornil-Sin, J. Blesa, and V. Puig. Nonlinear set-membership identification and fault detection using a bayesian framework: Application to the wind turbine benchmark. In *Proceedings of the IEEE Conference on Decision and Control*, pages 496–501, 2013.

[6] J. Xiong, C. Jauberthie, L. Travé-Massuyès, and F. Le Gall. Fault detection using interval kalman filtering enhanced by constraint propagation. In *Proceedings of the IEEE Conference on Decision and Control*, pages 490–495, 2013.

[7] A. Gning, B. Ristic, and L. Mihaylova. Bernoulli particle/box-particle filters for detection and tracking in the presence of triple measurement uncertainty. *IEEE Transactions on Signal Processing*, 60(5):2138–2151, 2012.

[8] F. Gustafsson. *Statistical sensor fusion*. Studentlitteratur, Lund, 2010.