Robust bounded-error tracking in wireless sensor networks

Farah Mourad-Chehade, Hichem Snoussi, Michel Kieffer, Cédric Richard

Robust bounded-error tracking in wireless sensor networks

F. Mourad [*] **H. Snoussi** [*] **M. Kieffer** [**] **C. Richard** [***]

[*] *Institut Charles Delaunay (ICD) - LM2S (STMR UMR CNRS 6279), Université de Technologie de Troyes, 12 rue Marie Curie, 10010 Troyes, France (e-mail: farah.mourad@utt.fr, hichem.snoussi@utt.fr)*
[**] *L2S - CNRS - SUPELEC, Université Paris-Sud, 91192 Gif-sur-Yvette, France, in leave at LTCI - CNRS - Telecom-ParisTech, 75013 Paris, France (e-mail: kieffer@lss.supelec.fr)*
[***] *Laboratoire Fizeau (UMR CNRS 6525), Université de Nice Sophia-Antipolis, Parc de Valrose, 06108 Nice, France (e-mail: cedric.richard@unice.fr)*

Abstract: A wireless sensor network (WSN) consists of spatially distributed sensors connected via a wireless link. Sensors may be designed for pressure, temperature, sound, vibration, motion... This paper considers the problem of target tracking in a WSN. This problem is especially challenging in presence of measurements which are outliers. Two algorithms for target tracking robust to outliers are proposed. They only assume that the maximum number of outliers is known. Based on interval analysis, these algorithms perform a set-membership estimation using either SIVIA or a combinatorial technique. In both cases, sets of boxes guaranteed to contain the actual target location are provided.

Keywords: Interval analysis, mobile target, outliers, set-membership estimation, target tracking.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are networks composed of a large number of wireless devices, having communication, sensing and processing capabilities Akyildiz et al. (2002). The main constraint of WSNs is their limited energy resources. For this reason, one should minimize the energy consumption of sensors in order to increase the lifetime of the whole network. In recent years, WSNs have emerged as a feasible solution for a wide range of applications in military, environment monitoring, healthcare, and so on Czubak and Wojtanowski (2009).

One interesting application of WSNs is target tracking Tran and Yang (2006); Mostafaei et al. (2009). It consists of estimating continuously the position of a moving target. This application is of great importance in surveillance and security domains, especially in military applications. Many algorithms have been proposed in literature for target tracking Ramachandra (2000); Djurić et al. (2008); Teng et al. (2010). Authors of Ramachandra (2000) uses the Kalman filter to estimate the target position. In Djurić et al. (2008), authors present particle filtering-based methods for target tracking using binary sensors, whereas in Teng et al. (2010), a clustering algorithm based on the variational filter is proposed. In a different scenario, authors of Mourad et al. (2011a) propose a target tracking algorithm using interval analysis Jaulin et al. (2001) for controlled mobility sensor networks.

Almost all existing methods are designed for tracking problems where all measurements are consistent with the considered observation model. However, in practical situations, erroneous measurements, or *outliers*, may occur. This paper considers the target tracking problem in presence of outliers in the interval framework. Very few algorithms have been proposed in literature for robust estimation based on intervals. In Jaulin et al. (1996), authors propose an estimator robust to a specific number of outliers. In Leger and Kieffer (2010), a distributed version of the estimation algorithm of Jaulin et al. (1996) is proposed. Assuming that the maximal number of outliers is known, both methods perform a set-membership estimation to compute a specific state vector. In a different scenario, authors of Mourad et al. (2011b) propose a robust localization algorithm based on reliability of measurements. The estimation of mobile sensors positions is thus performed using the belief theory Smets (1993).

This paper proposes a novel approach for single target tracking robust to outliers. Inspired by Jaulin et al. (1996) and Leger and Kieffer (2010), the proposed approach assumes that the maximal number of outliers is known. According to this approach, the estimation problem is defined using connectivity measurements performed between the target and the sensors of the network. Two algorithms are then proposed to solve the tracking problem. Based on interval analysis Jaulin et al. (2001), they both perform a set-membership estimation using either the SIVIA algorithm (Set Inversion Via Interval Analysis) or a combinatorial technique. The estimated positions with both methods are sets of boxes, guaranteed to contain the actual target locations.

The rest of the paper is organized as follows. Section 2 introduces the tracking problem. Section 3 describes the SIVIA-based and the combinatorial algorithms. Section 4 shows simulations results, whereas Section 5 concludes the paper.

2. PROBLEM STATEMENT

In this paper, the target and all sensors are assumed to be located in a two-dimensional space. Let N_s be the number of sensors composing the network. The position of the i-th sensor is given by $s_i(t) = (s_{i,1}(t), s_{i,2}(t))^T$, $i = 1,...,N_s$, whereas the target position is denoted by $x(t) = (x_1(t), x_2(t))^T$. Note that the sensors could be either fixed or mobile and are aware of their own positions.

The proposed method consists of collecting connectivity measurements to estimate the target position as in Mourad et al. (2009); Teng et al. (2010). Let $y_i(t)$ be the connectivity measurement related to the sensor i at time t. Then, $y_i(t)$ is a one-bit information given as follows,

$$y_i(t) = \begin{cases} 1, & \text{if the sensor } i \text{ detects the target at time } t, \\ 0, & \text{otherwise.} \end{cases}$$
(1)

The generation of such measurements is based on the assumption that the power of a signal decreases monotonically with the increase of the distance traveled by the signal. The target here is assumed to be active, and thus, it keeps on communicating with the sensors. In other words, the target broadcasts regularly signals in the network with the same initial power. These signals are received with different powers, denoted by $\rho_i(t)$, depending on the distances separating the target from the sensors. The generation of measurements of (1) is then performed by comparing each received power to a power threshold ρ_r, corresponding to the sensing range r. If $\rho_i(t) \geq \rho_r$, the sensor i detects the target and $y_i(t)$ is set to 1. Here the target is located at a distance $d_i(x(t))$ less than r. In the other case, where $\rho_i(t) < \rho_r$, the target is located out of the sensing range of the sensor i and here $y_i(t)$ is set to 0. Let $I(t)$ be the set of indices of sensors detecting the target at time t (having $y_i(t) = 1$). Then all sensors refereed in $I(t)$ are located at distances less than r from the target. The problem constraints are thus given by

$$(x_1(t) - s_{i,1}(t))^2 + (x_2(t) - s_{i,2}(t))^2 \leq r^2, \quad i \in I(t). \quad (2)$$

The problem is formulated by a set of disk equations centered on the sensors detecting the target and having r as radii.

In practice, the powers of exchanged signals could be modified due to some additive noise or the inaccuracy of some parameters. Hence, the target could be assumed to be within the sensing range of some sensors while it is not and vice versa, leading to outliers. Let $d_i^*(x(t))$ be the actual distance between the sensor i and the target and let $\rho_i(t)$ be the measured power of the signal. According to the values of $d_i^*(x(t))$ and $\rho_i(t)$, four cases could be encountered:

(1) $\rho_i(t) \geq \rho_r$ and $d_i^*(x(t)) \leq r$. Here, the sensor i assumes that the target is within its sensing range, which is true, leading to a correct constraint.
(2) $\rho_i(t) < \rho_r$ and $d_i^*(x(t)) > r$. Here, the sensor i assumes that the target is outside its sensing range, which is
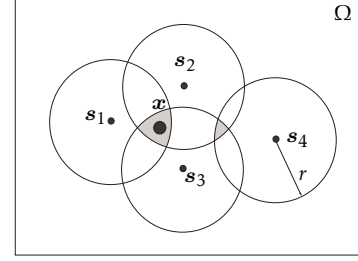

Fig. 1. An example of a tracking problem at a specific time step with one outlier.

also true. No constraint is generated concerning this sensor.

(3) $\rho_i(t) < \rho_r$ and $d_i^*(x(t)) \leq r$. Here, the sensor i assumes that the target is outside its sensing range, which is not true. The constraint concerning the sensor i is withdrawn, leading to a loss in the estimation accuracy.
(4) $\rho_i(t) \geq \rho_r$ and $d_i^*(x(t)) > r$. Here, the sensor i assumes that the target is within its sensing range, which is false. An erroneous constraint is thus generated, leading to an inaccurate estimation of the position of the target. Such constraints are called outliers.

The proposed method takes the outliers into consideration. It only assumes that the maximal number of outliers over all time steps is known, and denoted by q Jaulin et al. (1996). The tracking problem is then defined by the set of connectivity constraints given in (2), with at maximum q outliers. Fig. 1 shows an example of a tracking problem at a specific time step with one outlier. The plot shows in gray the solution area according to the hypothesis of $q = 1$ outlier. Here $q = 0$ leads to an empty solution whereas $q = 3$ for instance leads to the union of all disks.

3. ROBUST TARGET TRACKING

Solving the tracking problem at a given time t consists of finding the set of positions that are consistent with at least $n_q(t) = (|I(t)| - q)$ constraints. This paper uses interval analysis for that purpose Moore (1979); Jaulin et al. (2001). In other words, the computed solution consists of a set of non-overlapping boxes covering the solution set. In this section, a description of the approach is first proposed, then two different algorithms are presented for solving the problem. Note that in the proposed approach, the estimation process is performed at a central processing unit where all measurements are collected. For this purpose, all sensors detecting the target send their measurements and their positions at each time step to the central unit where computation is then processed.

3.1 Description of the approach

The proposed approach uses the set-membership estimation to solve the tracking problem Jaulin et al. (1996, 2001). In other words, it consists of computing the set of locations $\mathbb{X}_q(t)$ that are consistent with at least $n_q(t) = (|I(t)| - q)$ constraints at a given time t. $\mathbb{X}_q(t)$ is defined as follows,

$$\mathbb{X}_q(t) = \{x \in \Omega \mid \exists J \subseteq I(t), |J| \geq n_q(t), \forall j \in J, d_j(x) \leq r\}, \quad (3)$$

where Ω is some two-dimensional region including the surveillance area and $d_j(x) = \|x - s_j(t)\|$ is the Euclidian distance between x and the sensor j. $\mathbb{X}_q(t)$ could also be defined as follows,

$$\mathbb{X}_q(t) = \bigcup_{C \in \boldsymbol{C}_{I(t)}^{n_q(t)}} \left(\bigcap_{\ell \in C} \mathbb{D}_\ell(t) \right), \qquad (4)$$

where $\boldsymbol{C}_{I(t)}^{n_q(t)}$ is the set of all $n_q(t)$-combinations of indices of $I(t)$, C is a set of indices belonging to $\boldsymbol{C}_{I(t)}^{n_q(t)}$ and $\mathbb{D}_\ell(t)$ is the disk centered on the sensor ℓ with radius r,

$$\mathbb{D}_\ell(t) = \{x \in \Omega \mid d_\ell(x) \leqslant r\}. \qquad (5)$$

The total number of combinations to be considered in (4) is equal to $\frac{|I(t)|!}{n_q(t)!q!} = \frac{|I(t)|!}{(|I(t)|-q)!q!}$.

An alternative definition of $\mathbb{X}_q(t)$ not involving any combinatorial is given by

$$\mathbb{X}_q(t) = \{x \in \Omega \mid \Sigma_{i \in I(t)} \lambda(x, s_i(t)) \geqslant n_q(t)\}, \qquad (6)$$

where $\lambda(x, s_i(t)) = \begin{cases} 1 \text{ if } d_i(x) \leqslant r, \\ 0 \text{ otherwise.} \end{cases}$

It is obvious that the definitions (4) and (6) are equivalent.

In order to solve the tracking problem, two algorithms are proposed in the following: Algorithm 1, based on (6), and Algorithm 2, based on (4). Both algorithms employ interval analysis for computation Jaulin et al. (2001). Indeed, instead of computing the solution set $\mathbb{X}_q(t)$, they aim at computing a set of non-overlapping boxes, called *subpaving*, whose union covers $\mathbb{X}_q(t)$. Let $[\mathbb{X}_q](t)$ be the solution subpaving computed at time t, then

$$[\mathbb{X}_q](t) = \bigcup_{j=1}^{N(t)} [\boldsymbol{x}_j](t), \qquad (7)$$

where $[\boldsymbol{x}_j](t)$ is a two-dimensional box of the solution subpaving and $N(t)$ is the total number of boxes in $[\mathbb{X}_q](t)$. The number and the size of the boxes in the solution subpaving depend on the considered algorithm, as shown in what follows.

3.2 Algorithm 1: SIVIA-based algorithm for target tracking

The SIVIA algorithm (for Set Inverter Via Interval Analysis) consists of bisecting and selecting boxes successively, in the way to obtain at last boxes compliant with the problem constraints Jaulin et al. (2001). It aims at computing a solution subpaving $[\mathbb{X}_q](t)$ according to the definition (6). Consider the following test function,

$$\gamma([\boldsymbol{x}]) = \begin{cases} 1 \text{ if } \sum_{i \in I(t)} \lambda([\boldsymbol{x}], s_i) \geqslant n_q(t), \\ -1 \text{ if } \sum_{i \in I(t)} \tilde{\lambda}([\boldsymbol{x}], s_i) > q, \\ 0 \text{ otherwise,} \end{cases} \qquad (8)$$

where $n_q(t) = (|I(t)| - q)$, $[\boldsymbol{x}]$ is a two-dimensional box, $\lambda([\boldsymbol{x}], s)$ is equal to 1 if $\sup(\|[\boldsymbol{x}] - s\|) \leqslant r$ and 0 otherwise while $\tilde{\lambda}([\boldsymbol{x}], s)$ is equal to 1 if $\inf(\|[\boldsymbol{x}] - s\|) > r$ and 0 otherwise. According to (8), $\gamma([\boldsymbol{x}]) = 1$ means that $[\boldsymbol{x}]$ is
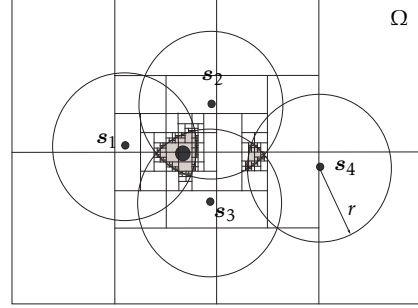

Fig. 2. Subpaving obtained using the Algorithm 1.

entirely compliant with at least $n_q(t)$ constraints and thus it is included in the solution area, $\gamma([\boldsymbol{x}]) = -1$ means that $[\boldsymbol{x}]$ is outside the solution area, whereas $\gamma([\boldsymbol{x}]) = 0$ means that $[\boldsymbol{x}]$ may have an non-empty intersection with $\mathbb{X}_q(t)$.

The SIVIA-based algorithm allows to build recursively $[\mathbb{X}_q](t)$ using the test function (8) as follows:

(1) Put Ω in the list $\mathcal{L}$ of boxes to be treated
(2) Take a box $[\boldsymbol{x}]$ out of $\mathcal{L}$ (initially $[\boldsymbol{x}] = \Omega$)
(3) Compute $\gamma([\boldsymbol{x}])$
    (a) If $\gamma([\boldsymbol{x}]) = 1$, store $[\boldsymbol{x}]$ in $[\mathbb{X}_q](t)$
    (b) If $\gamma([\boldsymbol{x}]) = 0$:
      • If $\omega([\boldsymbol{x}]) < \varepsilon$, store $[\boldsymbol{x}]$ in $[\mathbb{X}_q](t)$
      • Else bisect $[\boldsymbol{x}]$ into $[\boldsymbol{x}_1]$ and $[\boldsymbol{x}_2]$ according to its largest dimension and store them in $\mathcal{L}$
(4) If $\mathcal{L} \neq \emptyset$, go to (2)

ε is some precision parameter and $\omega([\boldsymbol{x}])$ is the width of the box $[\boldsymbol{x}]$. Fig. 2 shows in light and dark gray the solution subpaving related to the problem of Fig. 1 obtained using Algorithm 1 with $q = 1$. It is obvious that the subpaving covers precisely $\mathbb{X}_q(t)$, the number of boxes, and thus the accuracy, depending on ε.

3.3 Algorithm 2: Combinatorial algorithm for target tracking

The combinatorial algorithm employs the definition (4) to solve the tracking problem. Indeed, the solution subpaving $[\mathbb{X}_q](t)$ is computed according to Algorithm 2 as follows:

(1) Compute the set of all $n_q(t)$-combinations of indices of $I(t)$, denoted by $\boldsymbol{C}_{I(t)}^{n_q(t)}$
(2) For each combination C of $\boldsymbol{C}_{I(t)}^{n_q(t)}$:
    (a) Compute the minimal box $[\boldsymbol{x}]$ covering the overlapping region of all constraints denoted in C
    (b) Store $[\boldsymbol{x}]$ in $[\mathbb{X}_q](t)$

The step 2a of the algorithm is detailed in the following. Recall that $n_q(t) = (|I(t)| - q)$.

Having a specific combination C of constraints, the corresponding box $[\boldsymbol{x}]$ should cover all locations $x = (x_1, x_2)^T$ compliant with each constraint of (2) refereed in C. Based on (2), the coordinates of these locations satisfy the following constraints,

$$s_{i,k}(t) - b_{i,k}(t) \leqslant x_k \leqslant s_{i,k}(t) + b_{i,k}(t), \ k = 1, 2, \ i \in C, \qquad (9)$$
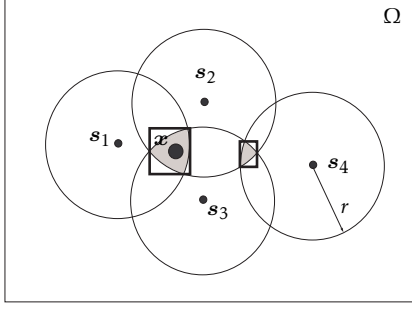
Fig. 3. Subpaving obtained using Algorithm 2.

with $b_{i,1}(t) = \sqrt{r^2 - (x_2 - s_{i,2}(t))^2}$ and $b_{i,2}(t) = \sqrt{r^2 - (x_1 - s_{i,1}(t))^2}$. The box $[\boldsymbol{x}] = [x_1] \times [x_2]$ covering all these locations is then obtained by contracting the initial domain Ω using all constraints of (9), implemented in the *forward-backward* algorithm Jaulin et al. (2001). The computation of $[\boldsymbol{x}]$ of step 2a of Algorithm 2 is performed according to the following:

  (i) Initialize $[\boldsymbol{x}]$ to Ω
  (ii) For each index i of C:
    • Compute $b_{i,1}(t) = \sup(\sqrt{r^2 - ([x_2] - s_{i,2}(t))^2})$
    • $[x_1] \leftarrow [x_1] \cap [s_{i,1}(t) - b_{i,1}(t), s_{i,1}(t) + b_{i,1}(t)]$
    • Compute $b_{i,2}(t) = \sup(\sqrt{r^2 - ([x_1] - s_{i,1}(t))^2})$
    • $[x_2] \leftarrow [x_2] \cap [s_{i,2}(t) - b_{i,2}(t), s_{i,2}(t) + b_{i,2}(t)]$
  (iii) If $[\boldsymbol{x}] \neq \emptyset$ and $[\boldsymbol{x}]$ is contracted, go to (ii)

Fig. 3 shows in thick black line the solution subpaving related to the problem of Fig. 1 obtained using Algorithm 2 with $q = 1$. It is obvious that this subpaving is less precise than the one obtained using Algorithm 1, but it is also less consuming in terms of memory resources since it has less boxes.

The computed boxes of $[\mathbb{X}_q](t)$ might have non-empty intersections. Once the computation of $[\mathbb{X}_q](t)$ is performed using Algorithm 2, and in order to obtain non-overlapping boxes, one could do the following steps:

  (a) Sort the boxes of $[\mathbb{X}_q](t)$ according to their decreasing areas
  (b) Initialize an empty set $[\mathbb{B}]$ to the top box of $[\mathbb{X}_q](t)$
  (c) Select the following box of $[\mathbb{X}_q](t)$
  (d) Deprive it from the boxes of $[\mathbb{B}]$ and add the result to $[\mathbb{B}]$
  (e) Go to step (c) until all boxes of $[\mathbb{X}_q](t)$ are considered
  (f) Set $[\mathbb{X}_q](t) = [\mathbb{B}]$

Recall that depriving a box $[\boldsymbol{x}]$ from a box $[\boldsymbol{y}]$ leads to a set of non-overlapping boxes, whose union covers all points of $[\boldsymbol{x}]$ not included in $[\boldsymbol{y}]$.

4. SIMULATIONS

This section compares Algorithm 1, using SIVIA, to Algorithm 2, based on combinatorial. For this aim, a single target is considered in a network of sensors deployed in a 100m×100m square area. The target is assumed to move over 100s with an estimation period of 1s, whereas the sensors are static and randomly deployed in the square area. In order to generate measurements, the distances

between the target and the sensors are used with the Okumura-Hata model Nadir et al. (2008), given by

$$\rho_i(t) = \rho_0 - 10 n_P \log_{10} \frac{d_i(x(t))}{d_0} + \varepsilon_i(t), \quad (10)$$

where $\rho_i(t)$ is the power (in dBm) of the signal emitted by the target and received by the sensor i at time t, ρ_0 is the power measured (in dBm) at a reference distance d_0 from the target, $d_i(x(t))$ is the Euclidian distance between the target and the sensor i at time t, n_P is the path-loss exponent and $\varepsilon_i(t)$ is the measurement noise. In this paper, ρ_0, d_0 and n_P are set to 100dBm, 1m and 4 respectively and $\varepsilon_i(t)$ is assumed to be a zero-mean Gaussian noise having a variance σ^2. The sensing range r is set to 10m leading to $\rho_r = 60$dBm with a noiseless version of model (10). The noisy powers are then compared to ρ_r leading to sets of measurements including outliers. All simulations are performed on an Intel Core i5-2520M CPU at 2.5GHz and 4GB RAM using Matlab 7.10 (R2010a).

4.1 Estimation of q

Consider the vector ρ of power measurements. Determining from ρ the maximal number of outliers q, which have to be tolerated, may be done by choosing q such that $\Pr(Q \leq q | \rho) > 1 - \nu$, where $\Pr(Q \leq q | \rho)$ is the probability that q or less outliers have occurred knowing ρ and $\nu \in [0,1]$ is some tuning parameter. One has

$$\Pr(Q \leq q | \rho) = \sum_{k=0}^{q} \Pr(Q = k | \rho) = \sum_{k=0}^{q} \sum_{(\mathbf{u} \in \{0,1\}^{N_s}, \sum_i u_i = k)} \Pr(Q = k, \mathbf{u} | \rho),$$
(11)

where $\mathbf{u}$ is some pattern indicating whether the sensor $i = 1, \dots, N_s$ is providing an outlier ($u_i = 1$) or a reliable measurement ($u_i = 0$). Now

$$\Pr(Q = k, \mathbf{u} | \rho) = \Pr(Q = k | \mathbf{u}, \rho) . \Pr(\mathbf{u} | \rho). \quad (12)$$

Let ρ_i and ρ_i^* be the noisy and noiseless received power measurements provided by the i-th sensor. The probability p_i that the i-th sensor provides an outlier is null ($p_i = 0$) if $\rho_i < \rho_r$ since in this case, the sensor i is not detected and will thus not provide any outlier. Otherwise, p_i is given as follows,

$$
\begin{aligned}
p_i &= \Pr\left(\rho_i^* < \rho_r | \rho_i \geq \rho_r\right) = \Pr\left(\rho_i - \rho_i^* > \rho_i - \rho_r\right) \\
&= \Pr(\varepsilon_i > \rho_i - \rho_r) = \frac{1}{2}\left(1 - \text{erf}\left(\frac{\rho_i - \rho_r}{\sqrt{2\sigma^2}}\right)\right),
\end{aligned}
\tag{13}
$$

where ε_i is the i-th measurement noise. Then, assuming that all measurement noise samples are independent

$$\Pr(\mathbf{u} | \rho) = \prod_{i=1}^{N_s} (p_i u_i + (1 - p_i)(1 - u_i)). \quad (14)$$

Now, since

$$\Pr(Q = k | \mathbf{u}, \rho) = \Pr(Q = k | \mathbf{u}) = \begin{cases} 1, & \text{if } \sum_i u_i = k \\ 0, & \text{otherwise} \end{cases}, \quad (15)$$

one is able to evaluate $\Pr(Q \leq q | \rho)$ and to choose q.

4.2 Comparison of Algorithms 1 and 2

This section compares Algorithm 1 to Algorithm 2 for a density of sensors equal to 0.015, leading to 150 sensors,
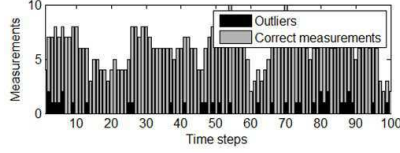
Fig. 4. Numbers of outliers and correct measurements with 150 sensors and $\sigma = 1$dBm.
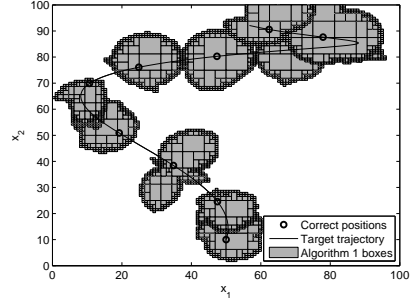

Fig. 5. Subpavings obtained using Algorithm 1.
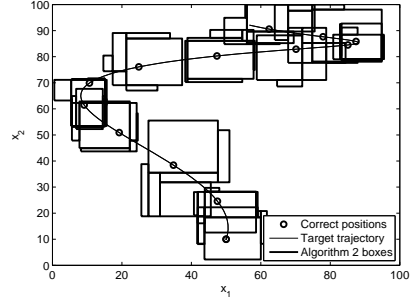

Fig. 6. Subpavings obtained using Algorithm 2.

Fig. 7. Subpavings obtained at time steps 33 in (a) and (b) and 63 in (c) and (d).

and an error variance of 1dBm^2. The sensors are randomly deployed in the surveillance area. Fig. 4 shows the number of correct measurements and outliers obtained at each time step. According to the plot, the maximal number of outliers is equal to 2. Using the estimation method of Section 4.1, one obtains $\Pr(q = 0) = 0.532$, $\Pr(q = 1) = 0.929$ and $\Pr(q = 2) = 0.998$. With $\nu = 0.01$, one gets $q = 2$, whereas with $\nu = 0.1$ leads to $q = 1$. In the following, ν is set to 0.01. In Algorithm 1, ε is set to 1m. Fig. 5 and Fig. 6 show the subpavings obtained using Algorithm 1 and Algorithm 2 respectively. Algorithm 1 leads to more precise subpavings compared to Algorithm 2, with an average ratio of subpavings areas (Algorithm 1 / Algorithm 2) equal to 0.85. The computation times are equal to 0.53s per time step and 0.043s per time step respectively whereas the average number of boxes in a subpaving are equal to 154 and 6 boxes respectively. For this example, Algorithm 2 is less consuming in terms of memory resources and computation time than Algorithm 1. This is mainly due to the small numbers of outliers considered in the problem. In fact, the computation time of Algorithm 2 highly depends on the number of sensors and the value of q. A study of this dependance is shown in the following section.

The number of boxes, the accuracy and the computation time of Algorithm 1 vary with the variation of the algorithm precision ε. Indeed, $\varepsilon = 5$m for instance leads to a computation time equal to 0.13s per time step, with an average number of boxes of 33. However, with this precision, Algorithm 1 leads to larger subpavings with an average ratio of subpaving areas with respect to Algorithm 2 (Algorithm 1 / Algorithm 2) equal to 1.18. In the following, the precision of Algorithm 1 is set to 1m.

The area of a subpaving depends as well on the number and the positions of the sensors detecting the target. Fig. 7 shows the subpavings obtained at time step 33 in (a) and (b) with Algorithm 1 and Algorithm 2 respectively and at time step 63 in (c) and (d) respectively. It also shows in ∗ the sensors detecting the target at these time steps. At time step 33, seven sensors detect the target whereas at time step 63, only three sensors is detecting it. The subpavings obtained at time step 33 are more accurate that the ones obtained at time step 63. Indeed, with the hypothesis of $q = 2$ and according to the definition of the approach, the solution remains at the union of the sets obtained by the intersection of any 4 constraints over 7 at time step 33, whereas it is at the union of all disks at time step 63. In other words, entire disks are considered at time step 63 whereas more disks are considered at time step 33, which leads to larger subpavings at time step 63.

4.3 Impact of sensors density and q value

This section shows the impact of the sensors density and the variance of the error on the performances of both algorithms. For this aim, the sensors density is first varied from 0.01 to 0.05 with a step of 0.01, leading to a number of sensors going from 100 to 500 randomly deployed in the surveillance area. Here the variance of the error is set to 1dBm^2. Fig. 8 shows the average number of sensors detecting the target and the maximal number of outliers in the top plot, the average subpavings areas in semi-log scale in the middle plot and the average computation times per time step in semi-log scale in the bottom plot. As expected, the number of sensors detecting the target highly increases going from 3 to 14 with a slight increase of q, from 2 to 3. The subpavings areas decrease for both algorithms due to the increase of the number of the sensors involved in the problem definition. Moreover, the computation time of Algorithm 1 remains almost constant whereas it increases for Algorithm 2. This is due to the increase of the number of sensors involved, leading to an increase of the considered combinations.
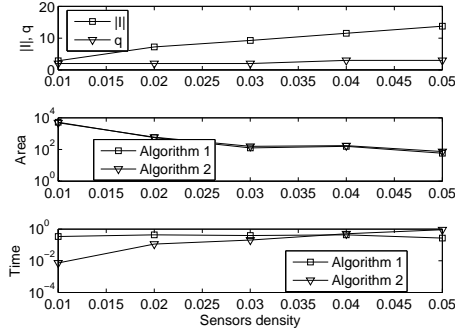
Fig. 8. Average number of sensors detecting the target $|I|$ and the maximal number of outliers q in the top plot, average subpavings areas in the middle plot and average computation times per time step in the bottom plot as functions of the sensors density.
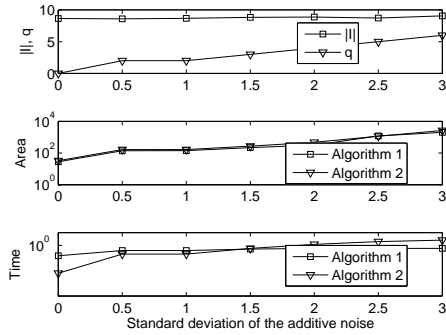

Fig. 9. Average number of sensors detecting the target $|I|$ and the maximal number of outliers q in the top plot, average subpavings areas in the middle plot and average computation times per time step in the bottom plot as functions of the variance of the additive noise.

Now the sensors density is set to 0.025 and the standard deviation σ of the noise is varied from 0dBm to 3dBm with a step of 0.5dBm. Fig. 9 shows the average number of sensors detecting the target and the maximal number of outliers in the top plot, the average subpavings areas in semi-log scale in the middle plot and the average computation times per time step in semi-log scale in the bottom plot. As expected, the maximal number of outliers increases, going from 0 to 6. The number of sensors detecting the target is almost constant even with the increase of q, which is mainly due to the loss of correct measurements because of the increasing error. The subpavings areas increase with a higher precision with Algorithm 1 compared to Algorithm 2. In terms of computation time, it is almost constant for Algorithm 1 whereas it highly increases with the increase of q for Algorithm 2 since here more combinations are considered.

5. CONCLUSION

This paper compares two algorithms for target tracking robust to outliers. Based on connectivity measurements, the tracking problem is defined by a set of connectivity disks centered on sensors detecting the target. Both algo-rithms perform a set-membership estimation using either the SIVIA algorithm or a combinatorial technique to solve the problem. Using interval analysis, the solution at each time step is a set of non-overlapping boxes guaranteed to contain the correct position of the target, provided that the number of tolerated outliers is larger than the actual number of outliers. A comparison of both algorithms is performed using simulated data. According to the simulation results, the SIVIA-based algorithm leads to a larger number of boxes than the combinatorial one, but it leads to a more precise estimate. In future works, one could imagine to solve the tracking problem in a distributed manner, where computation is performed on more than one sensor. An alternative way to provide a robust estimator would also take the reliability of measurements into consideration.

REFERENCES

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, 40, 102–114.

Czubak, A. and Wojtanowski, J. (2009). On applications of wireless sensor networks. In S.B.. Heidelberg (ed.), *Internet - Technical Development and Applications*, volume 64/2009, 91–99.

Djurić, P.M., Vemula, M., and Bugallo, M.F. (2008). Target tracking by particle filtering in binary sensor networks. *IEEE Transactions on Signal Processing*.

Jaulin, L., Kieffer, M., Didrit, O., and Walter, E. (2001). *Applied interval analysis*. Springer.

Jaulin, L., Walter, E., and Didrit, O. (1996). Guaranteed robust nonlinear parameter bounding. In *Proceedings of CESA'96 IMACS Multiconference*, 1156–1161. Lille, France.

Leger, J. and Kieffer, M. (2010). Guaranteed robust distributed estimation in a network of sensors. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 3378–3381.

Moore, R.E. (1979). *Methods and applications of interval analysis*. Siam, Philadelphia, USA.

Mostafaei, R., Habiboghli, A., and Meybodi, M.R. (2009). Target tracking in sensor networks: A distributed constraint satisfaction approach. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 55, 97–101. Oslo, Norway.

Mourad, F., Chehade, H., Snoussi, H., Yalaoui, F., Amodeo, L., and Richard, C. (2011a). Controlled mobility sensor networks for target tracking using ant colony optimization. *IEEE Transactions on Mobile Computing*.

Mourad, F., Snoussi, H., Abdallah, F., and Richard, C. (2009). Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *IEEE Transactions on Signal Processing*, 57(8), 3226–3239.

Mourad, F., Snoussi, H., Abdallah, F., and Richard, C. (2011b). A robust localization algorithm for mobile sensors using belief functions and interval theory. *IEEE Transactions on Vehicular Technology*, 60(4), 1799–1811.

Nadir, Z., Elfadhil, N., and Touati, F. (2008). Pathloss determination using okumura-hata model and spline interpolation for missing data for oman. In *Proceedings of the World Congress on Engineering*, volume 1. London, UK.

Ramachandra, K.V. (2000). *Kalman Filtering Techniques for Radar Tracking*. Marcel Dekker.

Smets, P. (1993). Belief functions: The disjunctive rule of combination and the generalized bayesian theorem. *International Journal of Approximate Reasonning*, 9, 1–35.

Teng, J., Snoussi, H., and Richard, C. (2010). Decentralized variational filtering for target tracking in binary sensor networks. *IEEE Transactions on Mobile Computing*.

Tran, S.P.M. and Yang, T.A. (2006). Evaluations of target tracking in wireless sensor networks. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, 97–101. Houston, Texas, USA.