Robust bounded-error tracking in wireless sensor networks
Farah Mourad, Hichem Snoussi, Michel Kieffer, Cédric Richard

To cite this version:

HAL Id: hal-00727519
https://hal-supelec.archives-ouvertes.fr/hal-00727519
Submitted on 3 Sep 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 et al. (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 $s_i(t) = (s_{i1}(t), s_{i2}(t))^T$, $i = 1, ..., N_s$, 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,

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

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 $\rho_r$, 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

$$(x_1(t) - s_{i1}(t))^2 + (x_2(t) - s_{i2}(t))^2 \leq r^2, \quad i \in I(t). \quad (2)$$

The proposed method uses the set-membership estimation 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. 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 $X_q(t)$ that are consistent with at least $n_q(t) = |[I(t)] - q]$ constraints at a given time $t$. $X_q(t)$ is defined as follows,

$$X_q(t) = \{ x \in \Omega \ | \ \exists J \subseteq I(t), |J| \geq n_q(t), \forall j \in J, d_j(x) \leq r \}. \quad (3)$$
where $\Omega$ is some two-dimensional region including the surveillance area and $d_j(x) = \|x - s_j(t)\|$ is the Euclidean distance between $x$ and the sensor $j$. $X_q(t)$ could also be defined as follows,

$$X_q(t) = \bigcup_{C \in C_{n_q(t)}^q} \left( \bigwedge_{\ell \in C} D_\ell(t) \right),$$  \hspace{1cm} (4)

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

$$D_\ell(t) = \{x \in \Omega \mid d_\ell(x) \leq r\}.$$ \hspace{1cm} (5)

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

An alternative definition of $X_q(t)$ not involving any combinatorial is given by

$$X_q(t) = \{x \in \Omega \mid \sum_{\ell \in I(t)} \lambda(x, s_i(t)) \geq n_q(t),$$ \hspace{1cm} (6)\]

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

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 $X_q(t)$, they aim at computing a set of non-overlapping boxes, called subpaving, whose union covers $X_q(t)$. Let $[X_q(t)]$ be the solution subpaving computed at time $t$, then

$$[X_q(t)] = \bigcup_{j=1}^{N(t)} [x_j](t),$$ \hspace{1cm} (7)

where $[x_j](t)$ is a two-dimensional box of the solution subpaving and $N(t)$ is the total number of boxes in $[X_q(t)]$. 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 $[X_q(t)]$ according to the definition (6). Consider the following test function,

$$\gamma([x]) = \begin{cases} 1 & \text{if } \sum_{\ell \in I(t)} \lambda([x], s_i(t)) \geq n_q(t), \\ -1 & \text{if } \sum_{\ell \in I(t)} \lambda([x], s_i(t)) > q, \\ 0 & \text{otherwise}, \end{cases}$$ \hspace{1cm} (8)

where $n_q(t) = (||I(t)|| - q)$, $[x]$ is a two-dimensional box, $\lambda([x], s)$ is equal to 1 if $\sup(||x - s||) \leq r$ and 0 otherwise while $\lambda([x], s)$ is equal to 1 if $\inf(||x - s||) > r$ and 0 otherwise. According to (8), $\gamma([x]) = 1$ means that $[x]$ is entirely compliant with at least $n_q(t)$ constraints and thus it is included in the solution area, $\gamma([x]) = -1$ means that $[x]$ is outside the solution area, whereas $\gamma([x]) = 0$ means that $[x]$ may have an non-empty intersection with $X_q(t)$.

The SIVIA-based algorithm allows to build recursively $[X_q(t)]$ using the test function (8) as follows:

1. Put $\Omega$ in the list $\mathcal{L}$ of boxes to be treated
2. Take a box $[x]$ out of $\mathcal{L}$ (initially $[x] = \Omega$)
3. Compute $\gamma([x])$
   a. If $\gamma([x]) = 1$, store $[x]$ in $[X_q(t)]$.
   b. If $\gamma([x]) = 0$:
      - If $\omega([x]) < \varepsilon$, store $[x]$ in $[X_q(t)]$.
      - Else bisect $[x]$ into $[x_1]$ and $[x_2]$ according to its largest dimension and store them in $\mathcal{L}$
4. If $\mathcal{L} = \emptyset$, go to (2)

$\varepsilon$ is some precision parameter and $\omega([x])$ is the width of the box $[x]$. 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 $X_q(t)$, the number of boxes, and thus the accuracy, depending on $\varepsilon$.

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 $[X_q(t)]$ 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 $C_{n_q(t)}^q$.
2. For each combination $C$ of $C_{n_q(t)}^q$:
   a. Compute the minimal box $[x]$ covering the overlapping region of all constraints denoted in $C$.
   b. Store $[x]$ in $[X_q(t)]$.

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 $[x]$ should cover all locations $x = (x_1, x_2)^T$ compliant with each constraint of (2) referred to in $C$. Based on (2), the coordinates of these locations satisfy the following constraints,

$$s_{i,k}(t) - b_{i,k}(t) \leq x_k \leq s_{i,k}(t) + b_{i,k}(t), \hspace{0.5cm} k = 1, 2, \hspace{0.5cm} i \in C, \hspace{0.5cm} (9)$$
Deprive it from the boxes of $\Omega$. Recall that depriving a box $x$ of a set of non-overlapping boxes, whose union covers all $\Omega$, is a useful tool for reducing the number of boxes.

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$^2$ 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

$$
\rho_i(t) = \rho_0 - 10n_p \log_{10} \frac{d_i(x(t))}{d_0} + \epsilon_i(t),
$$

where $\rho_i(t)$ is the power (in dBm) of the signal emitted by the target and received by the sensor $i$ at time $t$, $\rho_0$ 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 $\epsilon_i(t)$ is the measurement noise. In this paper, $\rho_0$, $d_0$ and $n_p$ are set to 100dBm, 1m and 4 respectively and $\epsilon_i(t)$ is assumed to be a zero-mean Gaussian noise having a variance $\sigma^2$.

Consider the vector $\rho$ of power measurements. Determining from $\rho$ 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 $\rho$ and $\nu \in [0, 1]$ is some tuning parameter. One has

$$
\Pr(Q \leq q|\rho) = \sum_{k=0}^{q} \Pr(Q = k|\rho) = \sum_{k=0}^{q} \sum_{u_1, \ldots, u_k} \Pr(Q = k, u|\rho),
$$

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

$$
\Pr(Q = k, u|\rho) = \Pr(Q = k|u, \rho) \Pr(u|\rho).
$$

Let $p_i$ and $p_i^*$ 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 $p_i < p_i^*$, since in this case, the sensor $i$ is not detected and will thus not provide any outlier. Otherwise, $p_i$ is given as follows,

$$
p_i = \Pr(p_i^* < p_i|p_i \geq p_i^*) = \Pr(p_i^* > p_i - p_i^*) = \frac{1}{2} \left( 1 - \operatorname{erf} \left( \frac{p_i - p_i^*}{\sqrt{2\sigma^2}} \right) \right),
$$

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

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

Now, since

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

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,
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, $\nu$ is set to 0.01. In Algorithm 1, $\epsilon$ 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 $\epsilon$. Indeed, $\epsilon = 5m$ 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.
Now the sensors density is set to 0.025 and the standard deviation \( \sigma \) 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 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 \( \sigma \) 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 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 \( \sigma \) 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 the bottom plot as functions of the variance of the additive noise.

### 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 algorithms 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


