Anchor-based localization via interval analysis for mobile ad-hoc sensor networks

Farah Mourad, Hichem Snoussi, Fahed Abdallah, Cédric Richard

***Abstract*—Location awareness is a fundamental requirement for many applications of sensor networks. This paper proposes an original technique for self-localization in mobile ad-hoc networks. This method is adapted to the limited computational and memory resources of mobile nodes. The localization problem is solved in an interval analysis framework. The propagation of the estimation errors is based on an interval formulation of a state space model, where observations consist of anchor-based connectivities. The problem is then formulated as a constraint satisfaction problem where a simple Waltz algorithm is applied in order to contract the solution. This technique yields a guaranteed and robust online estimation of the mobile node positions. Observation errors as well as anchor node imperfections are taken into consideration in a simple and computational-consistent way. Multi-hop anchor-based and back-propagated localizations are also made possible in our method. Simulation results on mobile node trajectories corroborate the efficiency of the proposed technique and show that it outperforms the particle filtering methods.**

I. Introduction

The emergence of Mobile Ad-hoc sensor NETworks (MANETs) has recently launched a growing widespread field of research involving theoretical and applicative challenges. The increasing capabilities and the decreasing costs of mobile devices make now mobile wireless networks possible and practical. These networks have significant advantages over traditional wired networks, such as easy deployment, self-management and no requirement for established infrastructure. MANETs are wireless networks defined by a collection of low-cost, tiny and densely distributed mobile sensor nodes, equipped with sensing and computation resources. The arising problems of such embedded devices are their limited memory, bandwidth, computational capabilities and energy reserve. In addition, many applications impose non-renewable nodes batteries and thus a limited lifetime of the network. Having typical low powered nodes, the main challenge of wireless sensor network research is to design collaborative and energy-aware processing tasks in order to prolong the lifetime of the whole network.

Research on mobile sensor networks, where the nodes mobility is uncontrolled, has partly focused on self-localization issue. Data processing and decision-making in mobile sensor networks assume that information provided by the sensor nodes are tightly related to their geographical locations. Therefore, the localization process is an absolute necessity to make sense of the observed data. Many applications such as target tracking or spatial interpolation for region monitoring [1], [2], [3] depend on the knowledge of the locations of sensor nodes. There are several constraints for choosing a localization technique, such as the limited computational power and the restricting environment conditions. A simple solution consists of equipping all nodes with Global Positioning Systems (GPS) [4]. However, this technique is impractical given the high energy consumption, the high cost and the binding size of GPS devices.

Recently, anchor-based localization methods have been proposed. In such methods, few nodes (called *anchors*) are equipped with GPS. The remaining nodes gather information from anchors in order to estimate their positions using efficient self-localization techniques. Some of the existing anchor-based works consist of repetitive static localization algorithms [5], [6], [7]. For instance, in the centroid method [6], each node defines its position as the center of all the observed anchors. In a different scenario, a bounding box localization has been proposed in [8], where nodes use their detection of a moving target to improve their position estimates. However, these methods have a limited performance due to the absence of any mobility modeling in the localization procedure. In [9], a dynamic localization scheme, the Monte-Carlo Localization (MCL), has been proposed. It is a new kind of Bayesian filtering algorithms based on the sequential Monte-Carlo method [10]. Using state space models, it incorporates a mobility propagating model in the localization process. More recently, an enhanced version of MCL, the Monte-Carlo localization Boxed algorithm [11], has been proposed in order to make better use of the gathered information. These Bayesian techniques are implemented by an approximate particle filter method [10] where many samples (particles) are drawn in order to estimate the node position. Nevertheless, particle-based methods need a high number of particles in order to achieve a good localization performance and thus require large amounts of memory.

In our work, we propose a new online anchor-based localization technique allowing the propagation of the position incertitude in an interval form. In order to take advantage of nodes mobility models, a state-space formulation is adopted for the online self-localization of nodes. The central idea of our method is to define the localization problem as a Constraint Satisfaction Problem (CSP). Tools of interval analysis [12] are then used in order to solve the CSP, where the prior mobility model and the observations are both considered as simultaneous constraints. Propagating boxes (multidimen-

F. Mourad, H. Snoussi and Cédric Richard are with the Institut Charles Delaunay (ICD, FRE CNRS 2848), Université de Technologie de Troyes, 12, rue Marie Curie, BP 2060, 10010 Troyes cedex, France, e-mail: {farah.mourad,hichem.snoussi,cedric.richard@utt.fr}@utt.fr.

F. Abdallah is with the Laboratoire HEUDIASYC, Université de Technologie de Compiègne, 60200 Compiègne, France, e-mail: fabdalla@hds.utc.fr.

sional intervals) allows a guaranteed estimation of the node position using only few parameters (endpoints of one box). The proposed boxed localization achieves a substantial gain in both memory and localization computational costs while ensuring the same performance as the particle filter algorithms. Furthermore, a boxed modeling of the anchors positions and of the sensing range incertitude enhances the robustness of the localization technique. Moreover, a back-propagated localization is made possible for applications that do not need a real-time localization. This contribution also proposes a multi-hop based localization technique for low anchor density networks. A large sensor field is covered with very few anchor nodes by flooding the network with anchor messages. With the multi-hop anchors information, every sensor node is able to construct location boundaries where it surely exist.

The rest of the paper is organized as follows. In section II, a brief introduction to interval analysis tools and their use in solving constraint satisfaction problems is proposed. Section III contains the main contribution of this paper: a guaranteed energy/memory-aware self-localization technique for MANETs. In Section IV, numerical results illustrating the efficiency of the proposed technique are discussed. Section V concludes the paper.

II. Interval analysis

The interval analysis represents a rigorous active field in scientific computation. This growing branch of applied mathematics aims to manipulate intervals instead of real numbers. Although the interval analysis is just a new language for inequalities, it is a very powerful framework providing an interesting alternative to punctual approximation. With interval methods, one is able to compute bounds of the possible solutions that correspond to measured quantities, and thus to obtain guaranteed regions that involve the correct solution. With simple operations, the interval analysis allows to consistently deal with problems involving non-punctual (interval) data. Interval computation is a special case of computation on sets, and the set theory provides the foundations for the interval analysis. In the following, the basics of the interval analysis are briefly presented, followed by constraint satisfaction problem tools.

A. Interval arithmetics

The interval analysis approach treats intervals as a new kind of numbers represented by the ordered pair of its endpoints. An interval $[x]$ of $\mathbb{R}$ is a closed bounded set defined as follows,

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

where $\underline{x}$ and $\overline{x}$ are the (finite or infinite) inferior and superior interval endpoints respectively. Interval analysis applies all standard set and arithmetic operations on intervals.

The main classical arithmetic operations, namely addition $(+)$, substraction $(-)$, multiplication $(*)$ and division $(/)$, can be extended to intervals. In fact, let $\diamond$ be any of these operators. It can be extended as an interval operator as follows,

$$[x] \diamond [y] = [\{x \diamond y \in \mathbb{R} | x \in [x], y \in [y]\}]. \quad (2)$$

Operations	Definitions
$\cap$	$[x] \cap [y] = [\max\{\underline{x}, \underline{y}\}, \min\{\overline{x}, \overline{y}\}]$
$\sqcup$	$[x] \sqcup [y] = [\min\{\underline{x}, \underline{y}\}, \max\{\overline{x}, \overline{y}\}]$
$+$	$[x] + [y] = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$
$-$	$[x] - [y] = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$
$*$	$[x] * [y] = [\min\{\underline{x} * \underline{y}, \underline{x} * \overline{y}, \overline{x} * \underline{y}, \overline{x} * \overline{y}\}, \max\{\underline{x} * \underline{y}, \underline{x} * \overline{y}, \overline{x} * \underline{y}, \overline{x} * \overline{y}\}]$

TABLE I
Some definitions of operations used in interval analysis.

Some of the listed operations are defined in Table I.

Set-theoretic operations can be applied to intervals. Consider two intervals $[x]$ and $[y]$. Their intersection is

$$[x] \cap [y] = [\{z \mid z \in [x], z \in [y]\}].$$

Most often, the union of two intervals is not an interval. In the interval framework, the union of two intervals is calculated as the smallest interval containing $[x] \cup [y]$. Thus, the interval union $\sqcup$ is defined as shown in Table I.

A multi-dimensional interval $[\boldsymbol{x}]$ of $\mathbb{R}^n$, called *box*, results from the cartesian product of n real intervals as follows,

$$[\boldsymbol{x}] = [x_1] \times ... \times [x_n] = [\underline{x}_1, \overline{x}_1] \times ... \times [\underline{x}_n, \overline{x}_n]. \quad (3)$$

Its i-th interval component $[\underline{x}_i, \overline{x}_i]$ is the projection of $[\boldsymbol{x}]$ onto the i-th axis. The notions introduced previously for intervals extend straightforwardly to boxes. For example, the intersection of two boxes is obtained by intersecting independently the intervals constituting these boxes.

B. Inclusion functions

Some other interval analysis tools are the inclusion functions. Let $\boldsymbol{f}$ be a function from $\mathbb{R}^n$ to $\mathbb{R}^p$. The image by $\boldsymbol{f}$ of a box $[\boldsymbol{x}]$, denoted $\boldsymbol{f}([\boldsymbol{x}])$, is defined as

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

It is obvious that $\boldsymbol{f}([\boldsymbol{x}])$ is not necessarily a box.

The interval function $[\boldsymbol{f}]$ from $\mathbb{R}^n$ to $\mathbb{R}^p$ is an inclusion function for $\boldsymbol{f}$ if it yields a box $[\boldsymbol{f}]([\boldsymbol{x}])$ such that

$$\forall [\boldsymbol{x}] \subset \mathbb{R}^n, \boldsymbol{f}([\boldsymbol{x}]) \subset [\boldsymbol{f}]([\boldsymbol{x}]) \subset \mathbb{R}^p \quad (4)$$

From this definition, it is straightforward to see that for a function $\boldsymbol{f}$ more than one inclusion function may exist. The *minimal* inclusion function of $\boldsymbol{f}$, denoted $[\boldsymbol{f}]^*$, is the smallest box in $\mathbb{R}^p$ that encloses $\boldsymbol{f}([\boldsymbol{x}])$. An illustration of inclusion functions is shown in Fig. 1.

Let $[f_i]$, $i = 1 \cdots p$, be p inclusion functions from $\mathbb{R}^n$ to $\mathbb{R}$ associated with the coordinate functions f_i of $\boldsymbol{f}$. A possible inclusion function of $\boldsymbol{f}$ is then given by

$$[\boldsymbol{f}]([\boldsymbol{x}]) = [f_1]([\boldsymbol{x}]) \times ... \times [f_p]([\boldsymbol{x}]).$$

Thus, a simple approach to build an inclusion function for $\boldsymbol{f}$ is to consider separately each $[f_i]$ and to replace each involved real variable within f_i by an interval variable, and each operator or function by its interval counterpart. The resulting inclusion function $[\boldsymbol{f}]$ is then called the natural inclusion function of $\boldsymbol{f}$.

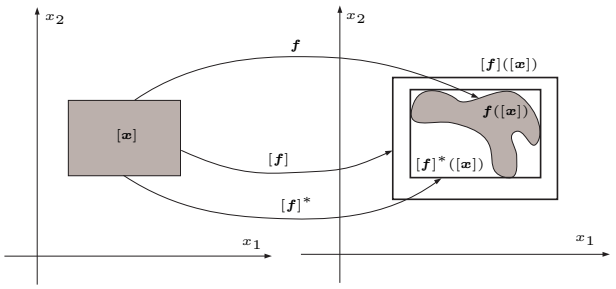


Fig. 1. Illustration of inclusion functions.

There is no guarantee that the natural inclusion function is the *best one* that can be used for f. In fact, the principal lack of interval analysis tools is their incapacity to find the minimal size box that encloses any set of solutions in $\mathbb{R}^n$. In order to overcome this drawback, constraint satisfaction algorithms (also called contractors) are combined to the interval analysis tools. The rationale behind these techniques is to use the redundancy of equations in order to reduce the size of the enclosing box.

C. Satisfaction of constraints : Waltz algorithm

A Constraint Satisfaction Problem (CSP) is defined by a set of m constraints, $f_1, f_2, ..., f_m$, involving n variables $x_1, x_2, ..., x_n$. Each constraint is defined as an equation $f_j(x_1, ..., x_n) = 0$ linking the components of the n-vector $\boldsymbol{x} = (x_1, ..., x_n)$. Each variable x_i has a nonempty domain D_i of possible values. Satisfying the constraints consists of finding the set of solutions $\boldsymbol{S}$, defined as $\boldsymbol{S} = \{\boldsymbol{x} \in \boldsymbol{D} \mid \boldsymbol{f}(\boldsymbol{x}) = 0\}$ where $\boldsymbol{f}$ is the vector of constraints and $\boldsymbol{D}$ an initial prior box enclosing the solution, defined in $\mathbb{R}^n$ by the combination of all the real domains $D_i, i = 1, ..., n$. In general, the solution set is not a box; it is a complete assignment of variables that satisfies all the constraints. Solving the CSP problem in an interval analysis framework consists of finding the smallest box $[\boldsymbol{x}]^* \subset \boldsymbol{D}$ that contains all possible solutions.

A contractor is a tool for the constraint satisfaction problem. It is an operator applied in order to get the solution $[\boldsymbol{x}]^*$, and thus to eventually contract as possible the initial box $\boldsymbol{D}$. In other words, the contractor is an algorithm that uses all constraints in order to reduce as much as possible the area of the box enclosing the exact solution. There are different kinds of methods to develop contractors. Each of these methods may be more suitable to some types of CSP. In this paper, we use the forward-backward propagation technique called Waltz algorithm [13], [14]. This contractor consists of applying iteratively primitive constraints, without any prior order, until the contractor becomes inefficient. A primitive constraint is a function involving arithmetic operations and standard functions (cos, exp, etc.). This strategy involves (i) using constraints equations to define explicitly each variable as a function of all the others, (ii) using interval analysis to express all variables and arithmetic operations in an interval framework, and (iii) initializing the variables with the domains $D_i, i = 1, ..., n$. The final phase (iv) consists of applying the explicit equations repetitively, without any prior order, to contract the variable intervals in the way to satisfy all the constraints. The loop is repeated until none of the variables is no longer contracted. The solution box is therefore defined by the cartesian product of the n contracted intervals. The boxes obtained are not necessarily the smallest boxes enclosing the set of solutions. In other words, the Waltz contractor does not lead certainly to globally minimal boxes; but to local minima. However, it remains a simple low-cost efficient technique. See algorithm 1.

Input: $D_1, D_2, ..., D_n$
Output: $[x_1]^*, [x_2]^*, ..., [x_n]^*$
Generate primitive functions $F_i^j, i = 1, ..., n, j = 1, ..., m$ using the m constraints in order to express explicitly each variable;
while *at least one variable is contracted* **do**
  for $j = 1, ..., m$ **do**
    for $i = 1, ..., n$ **do**
      $[x_i] = [x_i] \cap F_i^j([x_k], k \neq i)$;
    end
  end
end

Algorithm 1: Waltz algorithm.

III. Guaranteed Boxed localization

The proposed Guaranteed Boxed Localization (GBL) technique is based on propagating a set of constraints defined by the prior mobility model of the moving nodes and the information messages communicated by the neighboring anchors (moving or static nodes equipped with positioning systems). In this paper, non-anchor information are not used. Therefore, the problem is reduced to separable self-localization problems where each node uses its individual prior moving model in addition to anchor information. In the following, without loss of generality, we focus on the localization of only one mobile node. The same algorithm is implemented in parallel on the remaining nodes.

A. Definition of the localization problem

Mobile nodes are moving in a 2-dimensional deployment area where changes in direction and speed occur uncontrollably. Many mobility models have been proposed in order to represent the real movements of mobile nodes [17], [18], [19]. The model we are using in our method consists of a state space model where only the maximal velocity of a node is known. In this model, a mobile node moves between two time-steps from its current location to a new location by randomly choosing a direction and a speed for its movement. The speed is bounded by its maximal value, v_{max}, while the direction varies between 0 and 2π. This means that over one time-period, the node moves within the v_{max}-radius disk, centered at the previous position. Let $\boldsymbol{x}(t) = (x_1(t); x_2(t))$ be the coordinates of the mobile node at time t. The dynamic state space model that

we use in the GBL technique is defined by the following equations,

$$\begin{cases} x_1(t) = x_1(t-1) + v(t)\cos\theta(t) \\ x_2(t) = x_2(t-1) + v(t)\sin\theta(t) \end{cases} \quad (5)$$

where, for simplicity, the duration of one time-period is assumed equal to $1s$. Variables $v(t)$ and $\theta(t)$ are respectively the node velocity and the moving direction between instants $t-1$ and t. In our model, where only the maximal velocity of mobile nodes is assumed known, $v(t)$ and $\theta(t)$ are uniformly distributed between 0 and v_{max} and 0 and 2π respectively. This leads to a disk equation given by,

$$(x_1(t) - x_1(t-1))^2 + (x_2(t) - x_2(t-1))^2 \leq v_{max}^2 \ . \quad (6)$$

If we consider a punctual position at $t-1$ defined by $\boldsymbol{x}(t-1) = (a, b)$, the position at time t will be a disk having v_{max} as radius and (a, b) as center. In other words, equation (6) does not yield a punctual solution, but a disk where the real solution certainly exists. Additional prior information on the nodes trajectories could incorporated to refine the general mobility model described above.

Measurements to anchors are then used to further refine the mobile node localization. If a node can communicate with an anchor, its position is restricted to be in some region relative to the anchor. Such constraints are used by the moving node to define local connectivity measurements at each time step. A connectivity measurement relative to an anchor consists of one-bit information set to 1 if the anchor is within the communication range of the mobile node. This anchor is called *one-hop* anchor. We make the assumption of a rotationally symmetric communication range r where each node communicates with neighboring nodes that fall within the disk of radius r centered on the node. Let M be the number of anchors in the deployment area and m be the index of a single anchor such that $m \in \{1, ..., M\}$. The observation equations are defined by

$$y^m(t) = \begin{cases} 1 & \text{if } \ (x_1(t) - a_1^m)^2 + (x_2(t) - a_2^m)^2 \leq r^2 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where a_1^m and a_2^m are the coordinates of the m-th anchor, $m \in \{1, ..., M\}$. As for the mobility equation, each observation equation does not yield a punctual solution, but a disk centered on the corresponding one-hop anchor with r as radius. Satisfying all the observation equations produces the overlapping region of all one-hop anchor disks. The localization problem is defined by the set of the mobility equation and the observation equations where the connectivity measurements are non-zero. The solution of this problem is not punctual, but it consists of the intersection region of the anchor disks with the mobility disk.

B. Localization by interval analysis

The interval framework provides an efficient and consistent methodology to solve the localization problem described by (6) and (7). Instead of manipulating punctual positions, the main idea of our approach is to associate the node position with a multidimensional interval (box). The position box covers a guaranteed rectangular area where all acceptable positions *certainly* exist. The novelty of our work is that we define locations as intervals covering all acceptable solutions with respect to the given constraints instead of using approximated values. Propagating the positions in the interval form guarantees a bounded cumulative error in the localization process. The self-localization problem is then formulated as a CSP problem, where both the mobility and the observation models define the set of constraints.

The boxed localization algorithm is based on two phases at every time-step: (i) the propagation phase and (ii) the contraction phase. The first phase consists of generating the first domain of solutions called the prior region. It uses the mobility equation to propagate the location computed at $t-1$ till the current time t yielding a two-dimensional interval. The contraction step consists of applying the observation and the mobility constraints iteratively in order to minimize the area of the prior box. This can be implemented by the Waltz contractor. Note that the Waltz algorithm does not produce the globally minimal box but a local minimum. This means that the position box generated is not necessarily the optimal solution of the localization problem.

In order to use the interval analysis tools, the localization problem should be modified in the way to fit the interval framework. Let $[x_1](t)$ and $[x_2](t)$ be the coordinate intervals of the mobile node at time t. In the interval form, the motion equation (6) is formulated as follows,

$$[[x_1](t)-[x_1](t-1)]^2+[[x_2](t)-[x_2](t-1)]^2 = [0, v_{max}^2] \quad (8)$$

where v_{max} is the maximal velocity of the node. In order to use interval analysis operators, constraints should be reformulated whereby each coordinate is defined as a function of all other variables. The above constraint can be rewritten as follows,

$$\begin{cases} [x_1](t) = [[[x_1](t-1) - [b_1](t)] \cup [[x_1](t-1) + [b_1](t)]] \\ [x_2](t) = [[[x_2](t-1) - [b_2](t)] \cup [[x_2](t-1) + [b_2](t)]] \end{cases} \quad (9)$$

where $[b_1](t) = [\sqrt{v_{max}^2 - [[x_2](t) - [x_2](t-1)]^2}]$ and $[b_2](t) = [\sqrt{v_{max}^2 - [[x_1](t) - [x_1](t-1)]^2}]$. If the previous position is punctual, the result will be a disk. However, the propagation of a box with a disk equation does not produce a disk. In other words, if the position at the previous time-step is non-punctual, the mobility equation does not yield a disk; but a rounded corner square as shown in Fig. 2. Resolving the equation (8) alone, as in the propagation phase, leads to the minimal inclusion function of the solution, i.e. the minimal box that encloses the rounded corner square. The box could be defined by extending the coordinate intervals obtained at $t-1$ with a v_{max} length in the top, down, right and left directions respectively.

The connectivity constraints could also be formulated in an interval form as follows,

$$[[x_1](t) - a_1^i]^2 + [[x_2](t) - a_2^i]^2 = [0, r^2], \ i \in I \quad (10)$$

where r is the communication range of the node, a_1^i and a_2^i are the punctual coordinates of the i-th anchor and I is the set of all anchors communicating with the mobile node. In other

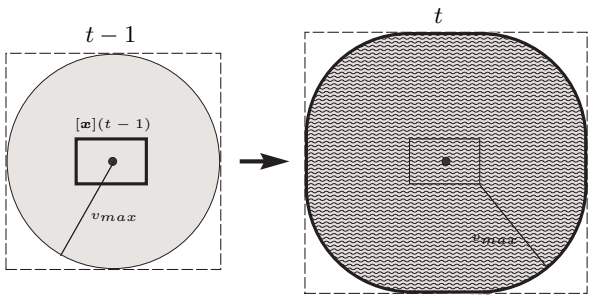


Fig. 2. The result of the propagation of a box with a disk equation.

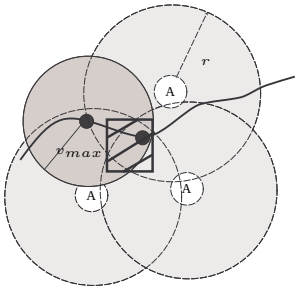


Fig. 3. Mobility and observation models with disk equations.

words, I is the set of indices of all one-hop anchors; those with connectivity measurements equal to 1. Explicit equations are also needed to implement the Waltz algorithm. They are defined as follows,

$$\begin{cases} [x_1](t) = [[a_1^i - [b_1^i](t)] \cup [a_1^i + [b_1^i](t)]] \\ [x_2](t) = [[a_2^i - [b_2^i](t)] \cup [a_2^i + [b_2^i](t)]] \end{cases}, i \in I \quad (11)$$

where $[b_1^i](t) = [\sqrt{r^2 - [[x_2](t) - a_2^i]^2}]$ and $[b_2^i](t) = [\sqrt{r^2 - [[x_1](t) - a_1^i]^2}]$. If we consider punctual anchor positions, the disk equations yield disks centered on the one-hop anchors with r as radius. In the interval form, each disk taken alone is represented by the smallest box that encloses it. It consists of taking a square centered at the corresponding anchor with $2r$ as its side length.

The localization process consists of solving the equation system defined by the mobility and the observation models. The common solution of all equations contains the correct position of the mobile node. It is defined by the intersection region of the observation disks and the mobility rounded corner square. With the interval analysis tools, the computed position consists of the contracted box enclosing the overlapping area set by all the constraints together. Fig. 3 shows the smallest box that encloses the correct solution, if we consider a punctual position for the mobile node at the previous time-step. That is, for simplicity, the rounded corner square is reduced to a disk. In order to reduce the computational cost, we also propose two additional approximating schemes by relaxing the disk equations to square equations. In the first scheme, the prior motion equation is approximated as follows,

$$\begin{cases} [x_1](t) = [x_1](t-1) + [-v_{max}, v_{max}] \\ [x_2](t) = [x_2](t-1) + [-v_{max}, v_{max}] \end{cases}. \quad (12)$$

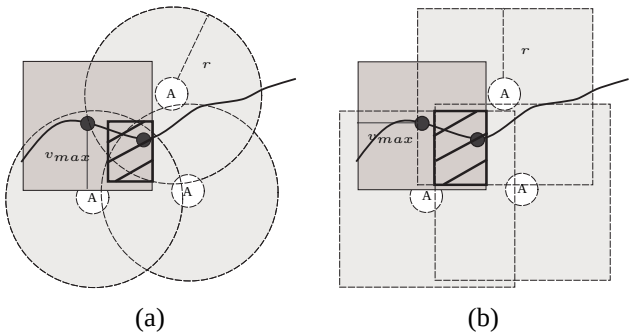


Fig. 4. Approximated schemes using square equations in the mobility model in (a) and both in the mobility and observation models in (b).

Fig. 4(a) shows the resulting box obtained by relaxing the mobility equation. Replacing the correct motion model with the approximated squared model does not affect the propagation phase (i.e. when the mobility equation is used alone). In fact, as shown in Fig. 2, defining the rounded corner square in the interval framework yields an identical box to the one obtained while propagating the previous box with the approximated equation. However, the approximation of the motion model may affect the contraction phase. In the second scheme, the connectivity constraints are also approximated as follows,

$$\begin{cases} [x_1](t) \subseteq [a_1^i - r, a_1^i + r] \\ [x_2](t) \subseteq [a_2^i - r, a_2^i + r] \end{cases}, i \in I. \quad (13)$$

Fig. 4(b) illustrates the box obtained using the approximations of the motion and observation equations. It is worth noting that, in the illustration of the approximated schemes, we have considered a punctual position of the node in the previous time step. As expected, relaxing the constraints leads to larger position intervals enclosing those obtained when using the exact disk constraints.

Algorithm 2 shows the guaranteed boxed localization pseudo-code based on the one-hop anchors observations. It uses the second scheme of the method where only observations are kept as disk equations and the propagation model is approximated by squared constraints. The Waltz algorithm stops when the number of iterations preformed exceeds a certain threshold or when the instant location box is no more contracted.

The interval-based approach also provides a convenient framework to deal with environment imperfections such as anchors positions or the communication range imprecisions. For instance, the GPS does not provide exact anchor positions. Moreover, converting the received signal strength to distances is imprecise. The conversion errors could affect the localization results when the real distance from the mobile node to an anchor is close to the communication range value. In these circumstances, the mobile node could be supposed outside the connectivity disk of an anchor while it is not and vice-versa. In the GBL technique, the uncertainty about the measurements can be incorporated by the use of intervals for anchors positions and the communication range instead of approximated values. With the non scalar communication range, the border of the connectivity disk is not a circle but a

Input: v_{max}, r, anchors coordinates, observations
Initialization: $[x_1](0)$ and $[x_2](0)$;
for $t \leq T$ **do**
  $[x_1](t) = [x_1](t-1) + [-v_{max}, v_{max}]$;
  $[x_2](t) = [x_2](t-1) + [-v_{max}, v_{max}]$;
  $I =$ Indices of 1-hop anchors;
  while *contraction is positive* **do**
    for $i \in I$ **do**
      $[b_1^i](t) = [\sqrt{r^2 - [[x_2](t) - a_2^i]^2}]$;
      $[x_1](t) =$
      $[[x_1](t) \cap [[a_1^i - [b_1^i](t)] \cup [a_1^i + [b_1^i](t)]]]$;
      $[b_2^i](t) = [\sqrt{r^2 - [[x_1](t) - a_1^i]^2}]$;
      $[x_2](t) =$
      $[[x_2](t) \cap [[a_2^i - [b_2^i](t)] \cup [a_2^i + [b_2^i](t)]]]$;
    end
  end
end

Algorithm 2: Guaranteed boxed localization algorithm using approximated propagation equations.

ring. The constraints are modified as follows,

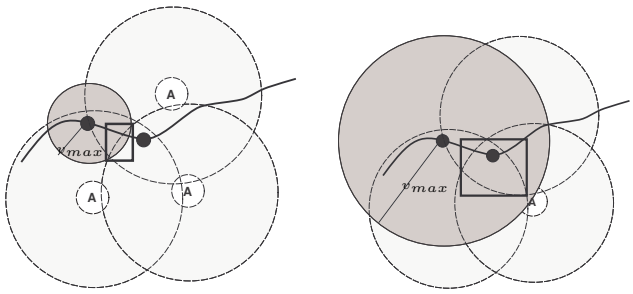
$$[[x_1](t) - [a_1^i]]^2 + [[x_2](t) - [a_2^i]]^2 = [0, \max[r]^2],\ i \in I \quad (14)$$

where $[a_1^i]$ and $[a_2^i]$ are the coordinate intervals of the i-th anchor and $[r]$ is the communication range box.
The explicit equations are then defined as follows,

$$\begin{cases} [x_1](t) = [[[a_1^i] - [b_1^i](t)] \cup [[a_1^i] + [b_1^i](t)]] \\ [x_2](t) = [[[a_2^i] - [b_2^i](t)] \cup [[a_2^i] + [b_2^i](t)]] \end{cases}, i \in I \quad (15)$$

where $[b_1^i](t) = [\sqrt{\max[r]^2 - [[x_2](t) - [a_2^i]]^2}]$ and $[b_2^i](t) = [\sqrt{\max[r]^2 - [[x_1](t) - [a_1^i]]^2}]$.

Remark 1: The algorithm we propose uses constraints based on a state space model. The contraction is thus made on an initial box until it encloses the intersection area of a propagation disk and the connectivity regions provided by the detected anchors. Using the general prior model where only the maximal velocity of a node is fixed, we define the propagation constraint as a disk equation determined by the maximal velocity. An erroneous value of the maximal velocity parameter may affect the localization performance leading to an inaccurate intersection region. Indeed, as shown in Fig. 5, a small velocity may cause an empty overlapping region, whereas a larger velocity than the real value causes less contraction of the solution box. Note that with a large value, the propagation disk may cover the intersection of the sensed anchors regions. Applying the localization in this case is equivalent to repetitive static methods that do not consider the state space model. This situation also occurs when we use the correct value of v_{max} but for some time instants, the real velocity is much smaller than v_{max}. In this case, the connectivity region formed by anchors will be included in the v_{max} disk; and thus the motion model will not contribute to the contraction at these time instants. In case of empty intersection, the mobility model constraint is relaxed and the Waltz algorithm is only applied to the observation constraints. A larger box that yields less accurate position is thus obtained,



(a) (b)

Fig. 5. Different values of v_{max} (lower in (a) and larger in (b) than its real value).

as shown in Fig. 5(b). Note that the choice of the maximal velocity has the same impact on all localization methods using the state space model with a fixed velocity. For instance, the particle filtering technique suffers, in the same way, from the dependence on the value of the maximal velocity.

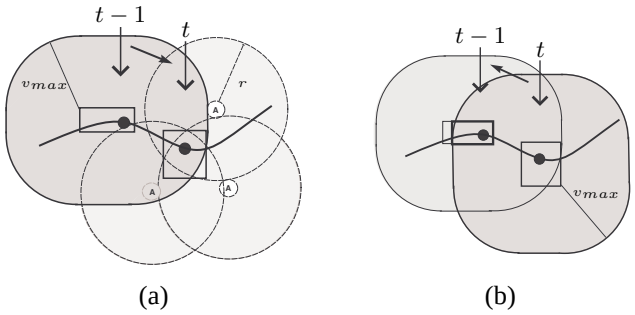
C. Back-propagated localization

In this section, we propose an extension of the GBL technique for situations where the realtime constraint is relaxed. In online applications, observations collected at a current instant should be processed to generate a solution before receiving observations at the next time instant. However, for non-realtime applications, collected information at a current time-step could be used to correct boxes obtained at previous time-steps. In fact, the position box obtained at time t using only observations from the current instant is used to correct boxes obtained at instants $t-1$, $t-2$, $t-3$,.... Since the position coordinates coming from two consecutive instants are related with the mobility model, this equation is used to correct past boxes. Let the position box corresponding to time $t-1$ denoted by the *previous box*. The observations collected at time t are used to generate a current box $[\boldsymbol{x}](t)$. This box is then propagated in a backward way using the prior equation defined by (8). The back-propagation of $[\boldsymbol{x}](t)$ yields an additional constraint for the *previous box*. Using this constraint yields a more accurate box for the instant $t-1$. The resulting box $[\boldsymbol{x}](t-1)$ is then similarly used to correct the box corresponding to $t-2$, and so on. These steps could be repeated until the initial instant or until a fixed number of backward time-steps is performed. The equations used to define the constraints for the *previous boxes* are shown in the following,

$$\begin{cases} [x_1](k-1) = [[[x_1](k) - [b_1](k)] \cup [[x_1](k) + [b_1](k)]] \\ [x_2](k-1) = [[[x_2](k) - [b_2](k)] \cup [[x_2](k) + [b_2](k)]] \end{cases} \quad (16)$$

where k decreases from the current instant t down to some fixed instant greater than 2, $[b_1](k) = [\sqrt{v_{max}^2 - [[x_2](k-1) - [x_2](k)]^2}]$ and $[b_2](k) = [\sqrt{v_{max}^2 - [[x_1](k-1) - [x_1](k)]^2}]$. These equations are defined using equation (9) and going back in time starting with the current instant t.

The first phase of the algorithm defined above consists of going back in order to correct *previous boxes*. The following



(a) (b)

Fig. 6. Back-propagated localization : first forward step in (a) and backward step in (b).

phase consists of propagating the resulting *previous boxes* forwardly till the current time-step in order to update again all available position boxes. The forward-propagation applies the same iterations of the backward correction but in the other direction. That is, the *previous boxes* are propagated forwardly in order to provide additional constraints for the following instants. The forward propagation is given by the following equations,

$$\begin{cases} [x_1](k) = [[[x_1](k-1) - [b_1](k)] \cup [[x_1](k-1) + [b_1](k)]] \\ [x_2](k) = [[[x_2](k-1) - [b_2](k)] \cup [[x_2](k-1) + [b_2](k)]] \end{cases} \quad (17)$$

where k increases from the instant 2 up to the current instant t, $[b_1](k) = [\sqrt{v_{max}^2 - [[x_2](k) - [x_2](k-1)]^2}]$ and $[b_2](k) = [\sqrt{v_{max}^2 - [[x_1](k) - [x_1](k-1)]^2}]$. Fig. 6(a) shows the forward localization while Fig. 6(b) shows the effect of the backward phase on the previous boxes.

This technique takes advantage of all available information since it makes use of old, current and future observations. Using more information, it outperforms the simple online localization introduced in Section III-B. However, since this method uses additional constraints at every time step, it requires higher computational capacities than the online method.

D. Boxed localization using multi-hop information

In the guaranteed boxed localization technique, anchors flood the sensor field with beacon messages periodically. Each unknown position sensor node receives messages from the anchors within its communication range and makes use of this information to construct location boundaries. The accuracy of the position estimation is thus related to the number of one-hop anchors neighboring each GPS-less node.

To enhance the localizations performance in networks where anchors density is low, a hop counting technique based on received signal strength can be employed. The anchor i is ranged as a k-hop anchor with respect to a certain node if it is located within the *ring* formed by the two concentric R_{k-1}-circle and R_k-circle around the node with $R_k = kr$ and r is the communication range. The estimated distance between an anchor and the mobile node is obtained by converting the strength of the signal sent by the anchor and received by the node. The additional constraint is given by the following equation,

$$(k-1)r < \sqrt{(x_1(t) - a_1^i)^2 + (x_2(t) - a_2^i)^2} \leq kr \quad (18)$$

where $k \geq 1$, $x_1(t)$ and $x_2(t)$ are the real coordinates of the node and a_1^i and a_2^i are the coordinates of the i-th anchor.

Other hop-counting techniques might be used, for instance the DV-hop method [15], [16]. It is a hop-by-hop localization algorithm that works as an extension of both distance vector routing and GPS positioning. This method aims to compute average hop distances from each node to all anchors. Every anchor overwhelms the network with messages composed of five fields: the ID of the anchor, its two coordinates in the deployment area coordinate system, the sequence number (consistent with the time step) and a hop counter that increases every time the message passes by a node. Each unknown-position node has an anchor table updated every time the node receives an anchor message with a new sequence number or a lower hop-count. An anchor that receives another anchor message uses the hop-count, the coordinates of the anchor and its own coordinates to compute an elementary distance corresponding to one hop-count. Every anchor sends the average elementary distance as a supplementary field in its following-sequence messages or as an intermediate correction message. The average hop equivalent size is therefore used by the unknown position nodes in addition to hop-counts to compute there average distances to anchors. In order to apply this technique in our method, every node compares its estimated distances from anchors to thresholds that are multiple of the communication range and therefore, it defines the ring that encloses every anchor. This means that we are adapting the DV-hop method in the way that we are bounding the estimated distances to affect hop-counts to anchors. Since the computation of the distances to anchors are imprecise, using thresholds and thus hop-counts instead of the estimated distances keeps the guaranteed aspect of the method.

In our work, we adapt the multi-hop information as additional constraints to be involved in the nodes localization problem. This technique outperforms the one-hop based method in both accuracy and computational efficiency in low anchors density networks. While the propagation equation remains unchanged, the observations constraints are modified according to the following equation,

$$[[x_1](t) - a_1^i]^2 + [[x_2](t) - a_2^i]^2 = [(k-1)^2 r^2, k^2 r^2] \quad (19)$$

where $i \in I_k$, I_k is the indices set of all k-hop anchors detected by the mobile node and $k \geq 1$. The algorithm 3 refers to the multi-hop based localization method. It considers the second scheme of our method that uses the approximated propagation equations.

The hop-counting techniques presented above could generate erroneous measurements due to environment imperfections. The strengths of the signals sent by the anchors decrease with the increase of the traveled distance. The relation between the signal strength and the distance is defined by a parameterized pathloss model. The conversion of the received signal strength to distance information is imprecise due to the imprecision of the pathloss parameters. Moreover, the signal strength could be modified in a noisy channel. In parallel, the DV-hop technique yields estimated distances that could lead to imprecise observations in an noisy environment. The distance obtained in the two-hop counting techniques could be lower or

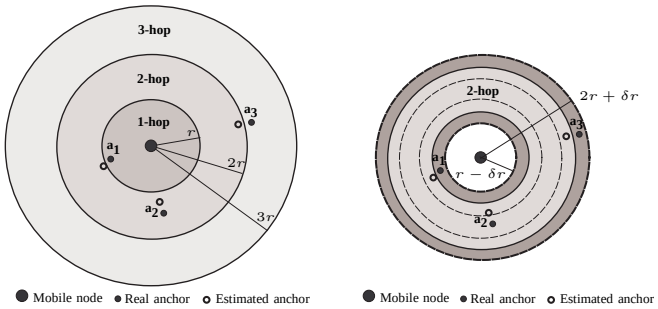
Input: v_{max}, r, anchors coordinates, observations
Initialization: $[x_1](0)$ and $[x_2](0)$;
for $t \leq T$ **do**
  $[x_1](t) = [x_1](t-1) + [-v_{max}, v_{max}]$;
  $[x_2](t) = [x_2](t-1) + [-v_{max}, v_{max}]$;
  $K =$ maximal hop-count;
  for $k \in \{1, ..., K\}$ **do**
    $I_k =$ Indices set of k-hop anchors
  end
  while *contraction is positive* **do**
    for $k \in \{1, ..., K\}$ **do**
      for $i \in I_k$ **do**
        $[b_1^i](t) = [\sqrt{k^2r^2 - [[x_2](t) - a_2^i]^2}]$;
        $[x_1](t) =$
        $[[x_1](t) \cap [[a_1^i - [b_1^i](t)] \cup [a_1^i + [b_1^i](t)]]]$;
        $[b_2^i](t) = [\sqrt{k^2r^2 - [[x_1](t) - a_1^i]^2}]$;
        $[x_2](t) =$
        $[[x_2](t) \cap [[a_2^i - [b_2^i](t)] \cup [a_2^i + [b_2^i](t)]]]$;
      end
    end
  end
end

Algorithm 3: Multi-hop guaranteed boxed localization algorithm.

larger than its true value. A comparison between the distance and the thresholds is performed in order to locate an anchor within a ring and thus to generate measurements needed in our multi-hop approach. A small modification of the distance has no impact when the true position of the anchor is equidistant to the circles defining the ring; but it could produce significant erroneous measurements when the anchor is too close to one of the ring bounds. In Fig. 7(a), we consider a 1-hop anchor a_1, a 2-hop anchor a_2 and a 3-hop anchor a_3. a_2 and a_3 are located within the rings centered on the mobile node and having r and $2r$ and $2r$ and $3r$ as inner and outer radii respectively, while a_1 is located within the disk of radius r. With the hop-counting techniques, the distance from the mobile node to the anchors are modified in the way that the three anchors are considered as 2-hop anchors, which leads to erroneous observations. Using inaccurate measurements to locate the mobile node leads to incorrect boxes. In order to take the ranging errors into considerations, we changed the scalar thresholds that are multiple of the communication range. We replaced them with intervals centered on their values with a fixed width. In other words, we used larger rings as illustrated in Fig. 7(b). A threshold kr is replaced by the interval $[kr-\delta r, kr+\delta r]$. The use of boxed thresholds induces the equation given below

$$\begin{array}{c}[[x_1](t) - [a_1^i]]^2 + [[x_2](t) - [a_2^i]]^2 = \\ [\max\{0, (k-1)r - \delta r\}^2, (kr + \delta r)^2].\end{array} \quad (20)$$

Fig. 7(b) shows in a dashed line the circles obtained by replacing the thresholds r and $2r$ with intervals. The new ring corresponding to the 2-hop anchors is thus located between the circles of radii $r - \delta r$ and $2r + \delta r$. In our localization process, an anchor considered as a 2-hop anchor is thus located at a distance from the mobile node varying between $r - \delta r$ and $2r + \delta r$. The correct distances between the anchors and the mobile node are thus covered with the extended ring. Therefore, choosing a convenient value of the interval width over the thresholds ensures robustness against ranging errors.



(a) (b)

Fig. 7. The impact of the ranging errors over the observation information in (a) and the result of using the boxed thresholds in (b).

IV. Simulations

The performance of the proposed GBL localization technique in MANETs is tested using the reference point group mobility model given in [17], [18], [19]. According to this mobility model, the mobile nodes follow the same reference trajectory with small independent stochastic deviations. We consider 300 nodes deployed in a square $100m \times 100m$ region. The communication and sensing range are set to $10m$. The density of anchors is set in such a way that each node has at least 3 anchors in its vicinity. The reference trajectory is composed of 2 sinusoids with an abrupt change in order to test the capacity of the algorithm to track the position of nodes in a difficult case (see Fig. 8). As the localization technique is only based on known anchor positions, the performance can be illustrated by following only one mobile node.

The maximal true velocity is $v^* = 2.035m$ per second. The three proposed boxed localization techniques (with different prior and observations constraints types), are denoted as follows: Case 1 (square-square), Case 2 (square-disk) and Case 3 (disk-disk). In order to compare the three cases, the maximal velocity parameter v_{max} is set to its maximal real value v^*. The computation times needed to accomplish the localization for the three cases are respectively $0.3280s$, $0.6550s$ and $0.7640s$. The average errors, that are the distances between the real and the estimated positions, are respectively $1.9830m$, $1.7775m$ and $1.7775m$, whereas the corresponding variances are $1.5960m^2$, $1.3504m^2$ and $1.3504m^2$, respectively. Fig. 8 shows the estimated boxes obtained in the first and second cases. The boxes obtained in Case 2 are included in those of Case 1 during the whole observation period.

In Fig. 9, we show the ratios of the boxes areas obtained in the three cases. As expected, using the correct disk equations for observations contracts more the boxes. However, it is worth noting that relaxing the disk constraint to a square constraint for the prior mobility does not affect the obtained boxes in our example (constant ratio equal to 1). In fact, the approximation

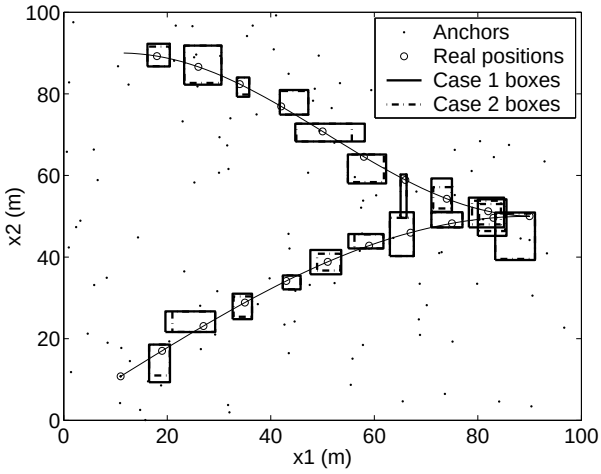


Fig. 8. Estimated boxes shown every $5s$ in Case 1 (square-square) and Case 2 (square-disk).

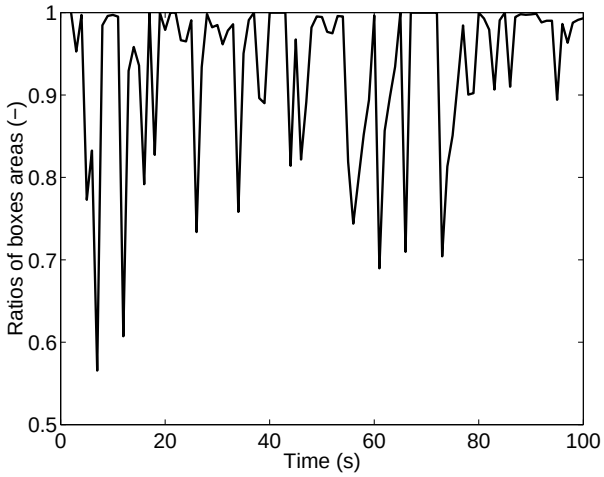


Fig. 10. Ratios of boxes areas obtained with the Waltz loop over the boxes areas obtained without the Waltz loop, in Case 2.

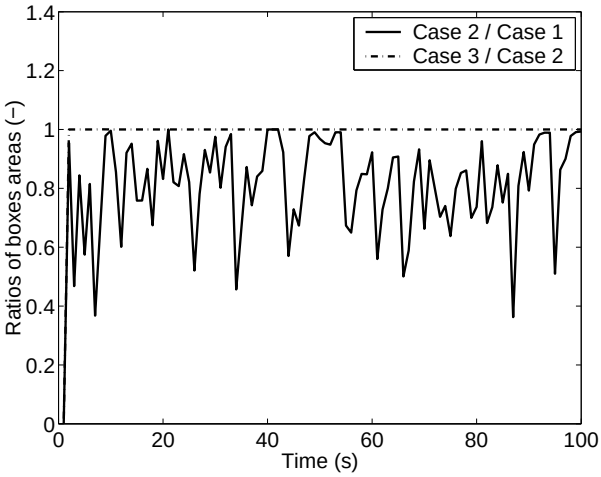


Fig. 9. Ratios of boxes areas for the considered three cases.

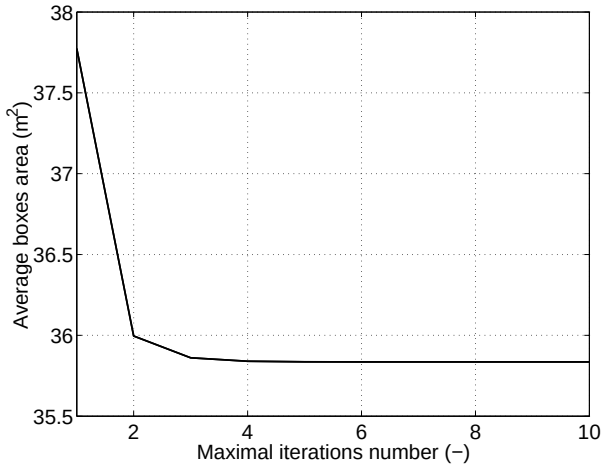


Fig. 11. Average boxes areas vs. the maximal iterations number in the Waltz algorithm in Case 2.

of the mobility model has no effect in the propagation phase as shown in Section III-B. This approximation may however have an impact on the algorithm performances when used in the Waltz algorithm (iterating the propagation and the contraction phases). The use of the correct mobility equation may lead to smaller location boxes when the observations area intersects with one of the rounded corners of the square, which was not the case in our example.

A. Efficiency of the Waltz algorithm

The Waltz algorithm is a contractor developed as a solution for constraint satisfaction problems. It propagates the constraints set by the model equations. The principle of the Waltz algorithm is to loop over the constraints, without any prior order, until the contraction is no more possible. In order to identify its effectiveness, in this section, we compare the localization results obtained by intersecting constraints equations with and without the Waltz loop over the constraints. In Fig. 10, we show the ratios of resulting boxes areas. This figure illustrates the contraction capacity of the Waltz algorithm. Fig. 11 reports the decreasing average boxes area over the maximal iterations number tolerated by the Waltz loop. Note that only a few number of iterations is needed to optimally contract the position boxes.

B. Variation of the maximal velocity parameter

The estimated boxes result from the intersection of the prior and the observations model regions. Here, we show the dependance of the localization outputs on the maximal velocity parameter v_{max}. In Fig. 12, we report the average error as function of the ratio of v_{max} to its real value (from 0.5 to 6). This figure shows that the best results are obtained at the real maximal velocity value. After a certain value of v_{max}, the average error becomes constant. In fact, the prior region encloses the intersection of observation regions and has then no effect on the contraction of the boxes. This situation is equivalent to the absence of the prior model information. Fig. 13 shows boxes obtained with v_{max} lower, equal and larger than its real value. With a lower parameter, boxes are reduced but do not contain the true positions at most of the time steps. A larger parameter leads to larger boxes and thus

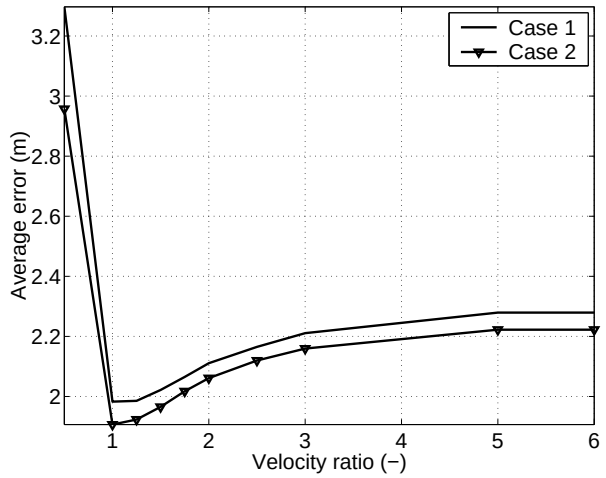


Fig. 12. Average error vs. the velocity ratio in Case 1 and Case 2.

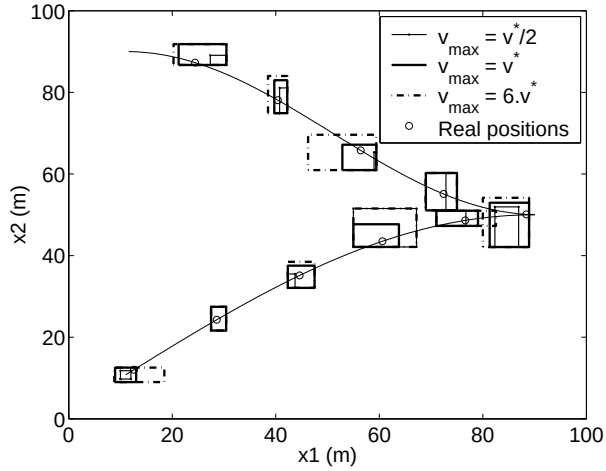


Fig. 13. Estimated boxes for three different values of v_{max} in Case 1.

to a loss in estimation accuracy.

C. Comparison to the Monte-Carlo boxed localization

The Monte-Carlo Boxed (MCB) localization algorithm [11] consists of two steps: (i) The prediction of particles inside the intersection of the mobility square and the observations approximated squares, and (ii) the filtering of particles by accepting only those respecting the disks constraints. These two steps are repeated until a fixed number of particles is kept. The estimated position is the mean of the particles. This method requires saving all the particles in the memory at every instant in order that can be used in the next time step.

In order to compare the proposed Guaranteed Boxed Localization (GBL) to the MCB method, we use the same simulation conditions as above. The number of particles is set to 50 particles. In Fig. 14, we plot the estimated boxes obtained by the GBL technique. We plot also the 50 particles obtained by the MCB method. Fig. 14 clearly illustrates that the real position can not be efficiently covered with 50 particles. The average error is equal to $2.1865m$ with a variance equal to $1.7596m^2$ using the MCB method while they are up to $1.7775m$ and $1.3504m^2$ respectively with the GBL method

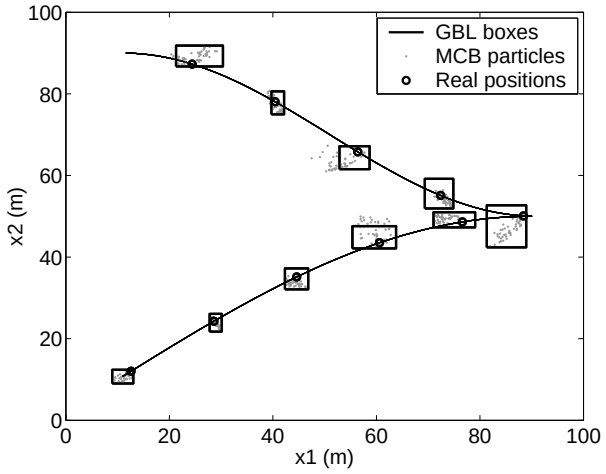


Fig. 14. Comparison of the GBL method (Case 2) to the MCB method using 50 particles at each step. The estimated boxes with GBL are shown in bold line. The computed particles with MCB and the real positions are illustrated by dots and circles respectively.

(with Case 2). The time needed to accomplish the localization algorithms is around $0.5930s$ for the GBL method and $2.7300s$ for the MCB method. Besides the gain in computation time, MCB requires the storage of at least 50 particles every time step while GBL only needs to save the resulting endpoints coordinates describing one estimated box.

In order to find out the particles number needed for a total coverage of the real position, we use a variable number of particles in the MCB localization algorithm. Experiments show that the minimal number of particles needed is around 500 particles. With this particle distribution, the average error obtained in MCB is $2.0296m$ with a variance equal to $1.6200m^2$ while the computation time is up to $25.4960s$. These results demonstrate the high performance of our method compared to the MCB localization, in terms of computation time, memory requirement and accuracy. In Fig. 15, we show the results of the localization by MCB with 500 particles compared to the GBL technique.

D. Impact of the ranging errors on the localization process

The GBL method uses RSSI measurements in order to define anchor information. Fixing a threshold P_0, an anchor is considered within the sensing range of the node if its corresponding RSSI is larger than P_0. Errors over the RSSI values may yield erroneous measurements and thus affect the localization process. Another equivalent problem is the accuracy of the channel pathloss model relating the signal strength to the distance traveled by the signal. Errors over the pathloss parameters lead to an incorrect sensing range value corresponding to P_0. Let r be the computed value of the sensing range and r_0 its correct value. An erroneous value of r leads to an inaccurate localization even with correct observations. A value larger than r_0 yields larger boxes and thus less precision in the localization results; while a lower value leads to smaller boxes that may not contain the real positions or it could lead to empty intersections, as well. In this case, the estimation error is more important. Moreover,

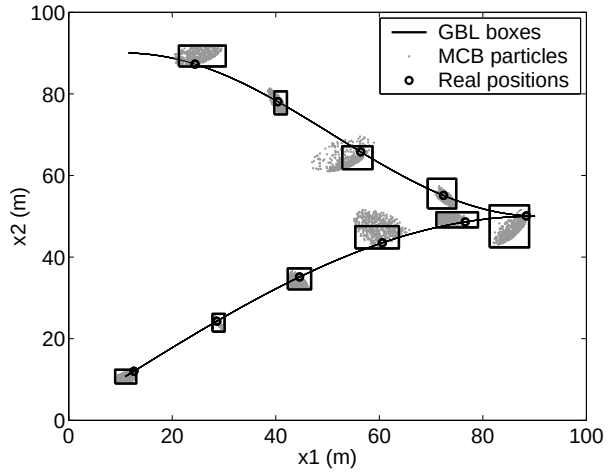


Fig. 15. Comparison of the GBL method (Case 2) to the MCB method using 500 particles at each step. The estimated boxes with GBL are shown in bold line. The computed particles with MCB and the real positions are illustrated by dots and circles respectively.

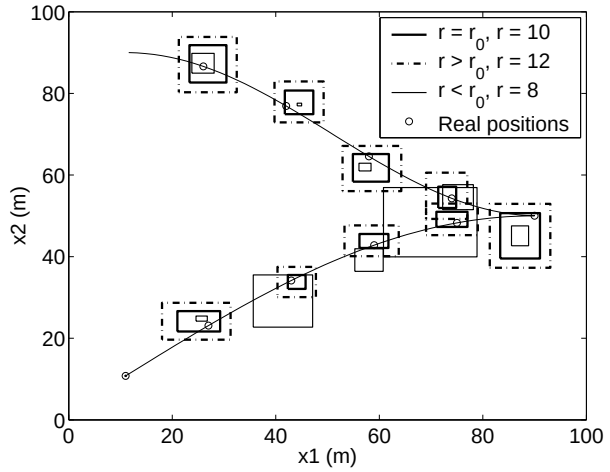


Fig. 16. Estimated boxes obtained using different values of the sensing range.

this may cause the loss of the guaranteed aspect of the method. Fig. 16 shows the boxes obtained using different values of r. As can be expected, the boxes obtained using $r > r_0$ are wider than the ones obtained using r_0 but they still contain the real positions; while the boxes obtained with smaller ranges do not contain the exact position at almost all the time-steps. Fig. 17 shows the average estimation error as a function of the sensing range value. r_0 is set to $10m$. We varied r between $6m$ and $14m$. As expected, the minimal error is obtained while using the correct value of the sensing range. When r is greater than r_0, the error increases slightly with the increase of r. On the other hand, when r is lower than r_0, the increase of the estimation error is more important with the decrease of r.

E. Boxed anchors positions and communication range

In a network where only few nodes (anchors) are equipped with GPS, unknown-position nodes estimate their position using anchors information. Thus, anchor-based localization techniques, where nodes periodically communicate with anchors, require accurate anchor positions and communication

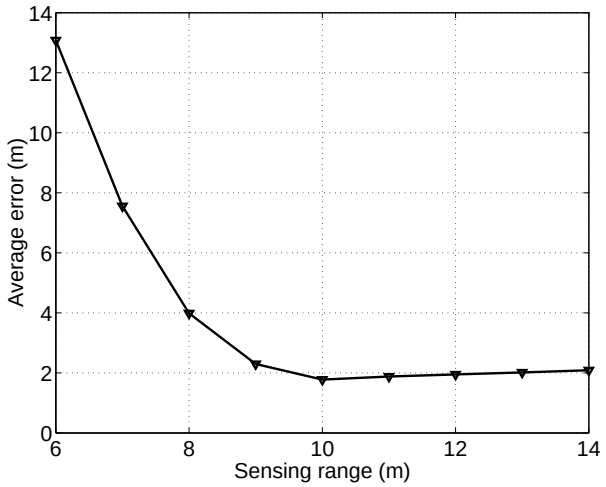


Fig. 17. Average error obtained using different values of the sensing range.

range information. Uncertainty about these information may lead to inaccurate localization results as shown in Section IV-D. In GBL method, where boxes might be used instead of approximated values for these parameters, one can enhance the robustness of the localization technique. An illustration of this robustness is shown in this section where we compare the GBL (Case 2) to MCB method based on uncertain anchors positions and communication range. For this purpose, we put a $2m \times 2m$ box around the true anchors positions and a $2m$ interval on the communication range whose exact value is $10m$. The average errors obtained are equal to $1.9712m$ and $3.0203m$ for GBL and MCB respectively, showing the capacity of the GBL technique to efficiently deal with uncertainty about the input parameters.

F. Back-propagated localization

The back-propagated localization uses the current result to update previous boxes, and then propagates again forwardly the updated boxes to correct the up-to-time one. In order to show the performance of our technique in offline applications, we use the mobility model composed of two sinusoids as above. The maximal velocity of the node is $v^* = 2.035m$ per second. We run the offline GBL localization and compare it to the online localization where current measurements are used for contracting only current boxes. Simulations show that the backward propagation yields a reduction of *previous boxes*, while the following forward propagation of *previous boxes* has no contracting impact. This means that only one-way update (backward way) is sufficient.

In our technique, we limited the number K of backward time-steps allowed since it increases the computational complexity. In order to determine the maximal K after which contractions are no more efficient, we plot the average boxes area as a function of the number of the backward time-steps allowed as reported in Fig. 18. It shows that the backward propagation remains efficient until 5 time-steps back. Fig. 19 shows the boxes obtained in the online localization and the back-propagated localization with 5 back time-steps. We can see, in Fig. 19, that with back-propagated localization, boxes

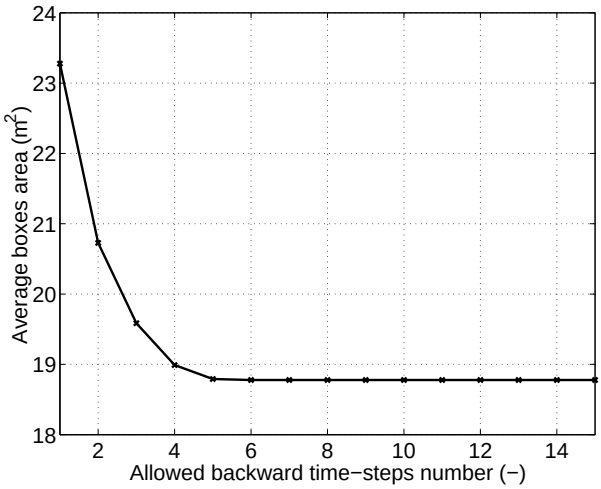


Fig. 18. Average boxes areas vs. the number of backward time-steps allowed in Case 2.

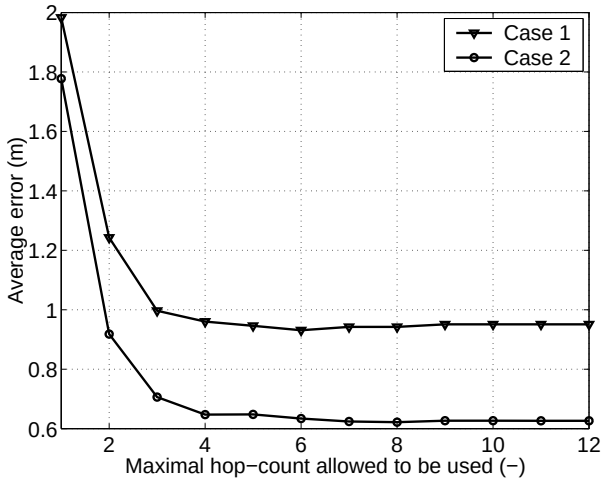


Fig. 20. Average error vs. the hop-count involved in Case 1 and Case 2 of our method.

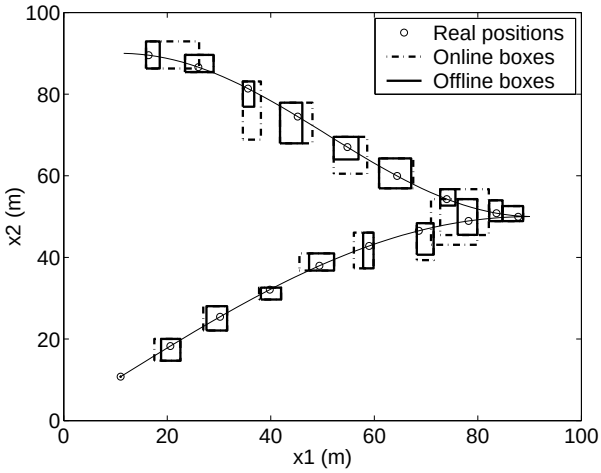


Fig. 19. Estimated boxes with online and back-propagated (offline) GBL (Case 2).

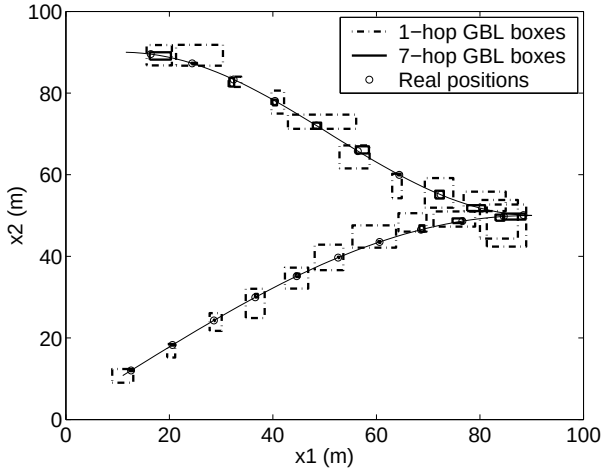


Fig. 21. Estimated boxes with 1-hop and up to 7-hop based GBL (Case 2).

are much more contracted than with online localization. The computation time of the offline localization algorithm is up to $0.7650s$ with an average error decreasing to $1.0759m$, compared to a computation time of $0.6450s$ and an average error of $1.7775m$ for online localization; whereas the error variance decreases from $1.5960m^2$ for the online method to $0.3899m^2$ for the back-propagated technique.

G. Boxed localization using multi-hop anchors

The multi-hop based localization is an efficient technique in low anchors density networks. It uses near and far anchors information to localize each moving node. In order to evaluate this technique, we keep the same anchors density as above and consequently, the number of anchors involved in the localization process at each time step will increase. Both the received strength based and the DV-hop based method can be used to define at which hop each anchor belongs at every time-step. In Fig. 20, we plot the average error as a function of the maximal hop-count allowed in the total approximated scheme (Case 1) and the approximated prior model based scheme (Case 2) of our method. One can note that with 1 to 7-hop anchors in Case 2 (6-hop anchors in Case 1), the mobile node is optimally localized. Fig. 21 shows the boxes obtained with 1-hop based and up to 7-hop based methods in the second case. With the increase of the number of involved anchors, the average error decreases from $1.7775m$ to $0.6241m$ with variances decreasing from $1.3504m^2$ to $0.3494m^2$, at the cost of the computational time which increases from $0.5920s$ to $15.1930s$. Note that this technique could be used to reduce the number of anchors in the network while maintaining a good localization performance.

V. Conclusion

In this contribution, we introduced the Guaranteed Boxed Localization, a novel approach based on interval analysis for self-localization in MANETs. The use of a state space model provides an accurate dynamic process instead of a repeated static localization. By associating boxes to estimated positions, we have covered bounded areas where solutions surely exist. The propagation of position boxes maintains a bounded estimation cumulative error. Compared to Monte

Carlo-based algorithms, the computation time and the needed memory are highly reduced while the total error decreases. Our simulation experiments reveal that the interval-based technic provides accurate results even when the anchors information and the sensing range are uncertain. With the multi-hop-based approach, we are able to provide accurate localization in low-anchor density networks; whereas in offline applications, current measurements together with past and future observations can be used to improve the performance of our method. Many issues remain to be explored in future work, particularly in the presence of environment imperfections. The use of belief functions represents a solution to handle more consistently erroneous information communicated between nodes.

REFERENCES

[1] *Collaborative Information Processing*, IEEE Signal Processing Magazine, 2002, vol. 19, no. 2.

[2] *Special Issue on Self-Organizing Distributed Collaborative Sensor Networks*, IEEE Journal on Selected Areas in Communications, 2005, vol. 23, no. 4.

[3] *Distributed Signal Processing In Sensor Networks*, IEEE Signal Processing Magazine, 2006, vol. 16.

[4] B. Hoffman-Wellenhof, H. Lichteneeger and J. Collins, *Global Positioning System: Theory and Practice*, 4th ed. New York, USA: Springer-Verlag, 1997.

[5] L. Lazos, R. Poovendran and S. Capkun, *ROPE: Robust Position Estimation in Wireless Sensor Networks*, Los Angeles, USA: Proc. IPSN, article No.43, 2005.

[6] N. Bulusu, J. Heidemann and D. Estrin, *GPS-less Low Cost Outdour Localization for Very Small Devices*, IEEE personal communications magazine, vol. 7, No. 5, pp. 28-34, October 2000.

[7] V. Vivekanandan and V. W.S. Wong, *Concentric Anchor Beacon Localization Algorithm for Wireless Sensor Networks*, IEEE transactions on vehicular technology, vol. 56, No. 5, September 2007.

[8] A. Galstyan, B. Krishnamachari, S. Pattem and K. Lerman, *Distributed Online Localization in Sensor Networks Using a Moving Target*, Berkeley, CA, USA : Proceedings of IEEE/ACM 3rd International Symposium on Information Processing in Sensor Networks (IPSN), pp. 61-70, April 2004.

[9] L. Hu and D. Evans, *Localization for Mobile Sensor Networks*, Philadelphia, USA: Tenth annual international conference on mobile computing and networking (MobiCom), 2004.

[10] A. Doucet, S. Godsill and C. Andrieu, *On Sequential Monte Carlo Sampling Methods for Bayesian Filtering*, Statistics and Computing, vol. 10, pp. 197-208, 2000.

[11] A. Baggio and K. Langendoen, *Monte-Carlo Localization for Mobile Wireless Sensor Networks*, Hond Kong, China: Mobile Ad-hoc and Sensor Networks, second international conference, MSN, 2006.

[12] R. E. Moore, *Methods and applications of interval analysis*, Philadelphia, USA: Applied mathematics, 1979.

[13] D. Waltz, *Generating semantic descriptions from drawings pf scenes with shadows*, New York, USA: The psychology of computer vision, pp. 19-91, 1975.

[14] F. Abdallah, A. Gning and Ph. Bonnifait, *Box particle filtering for non linear state estimation using interval analysis*, Automatica, vol. 44, pp. 807-815, 2008.

[15] D. Niculescu and B. Nath, *DV Based Positioning in Ad Hoc Networks*, Telecommunication Systems, vol. 22, No. 1-4, pp. 267-280, 2003.

[16] J. Yi, S. Yang and H. Cha, *Multi-hop-based Monte Carlo Localization for Mobile Sensor Networks*, San Diego, CA, USA: 4th annual IEEE Communication Society Conference on Sensor, Mesh and Ad hoc Communications and Networks, pp. 162-171, 2007.

[17] T. Camp, J. Boleng and V. Davies, *A survey of mobility models for ad hoc network research*, Wireless Communications and Mobile Computing: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp. 483-502, 2002.

[18] J.-Y. Le Boudec and M. Vojnovic, *The random trip mobility, part I: Stability*, Tech. Rep., 2005.

[19] X. Hong, M. Gerla, G. Pei and C.-C. Chiang, *A group mobility model for ad hoc wireless networks*, in MSWiM'99: Proceedings of the 2nd ACM international workshop on modeling, analysis and simulation of wireless and mobile systems, 1999.

Farah Mourad was born in Tripoli, Lebanon, in 1984. She received the diploma degree in electrical engineering from the Lebanese University, Faculty of Engineering, Tripoli, Lebanon, in 2006. She also received the Master degree in Systems Optimization and Safety from the University of Technology of Troyes (UTT), France, in 2007. Since October 2007, she is a PhD. student at the UTT. Her research concerns the localization of mobile sensors in mobile ad hoc sensor networks.

Hichem Snoussi was born in Bizerta, Tunisia, in 1976. He received the diploma degree in electrical engineering from the Ecole Superieure d'Electricite (Supelec), Gif-sur-Yvette, France, in 2000. He also received the DEA degree and the Ph.D. in signal processing from the University of Paris-Sud, Orsay, France, in 2000 and 2003 respectively. Between 2003 and 2004, he was postdoctoral researcher at IRCCyN, Institut de Recherches en Communications et Cybernétiques de Nantes. He has spent short periods as visiting scientist at the Brain Science Institute, RIKEN, Japan and Olin Neuropsychiatry Research Center at the Institute of Living in USA. Since 2005, he is associate professor at the University of Technology of Troyes, France. Since January 2008, he is leading research group "Surveillance" of LM2S laboratory. He is in charge of the regional research program S3 (System Security and Safety) of the CPER 2007-2013 and the CapSec plateform (wireless embedded sensors for security). He is the principal investigator of an ANR-Blanc project (mv-EMD), a CRCA project (new partnership and new technologies) and a GDR-ISIS young researcher project. He is partner of many ANR projects, GIS, strategic UTT programs. He obtained the national doctoral and research supervising award PEDR 2008-2012.

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 Troyes University of Technology, Troyes, France, all in electrical and computer engineering. From 2005, he has been an Associate Professor with the HEUDIASYC laboratory at Compiègne University of Technology, 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.

Cédric Richard was born January 24, 1970 in Sarrebourg, France. He received the Dipl.-Ing. and the M.S. degrees in 1994 and the Ph.D. degree in 1998 from Compiègne University of Technology, France, all in electrical and computer engineering. From 1999 to 2003, he was an Associate Professor at Troyes University of Technology, France. Since 2003, he is a Professor at the Systems Modeling and Dependability Laboratory, Troyes University of Technology. His current research interests include statistical signal processing and machine learning. Dr. Richard is the author of over 70 papers. He is the General Chair of the XXIth francophone conference GRETSI on Signal and Image Processing to be held in Troyes, France, in 2007. In 2005, he was offered the position of chairman of a pattern recognition section of the federative CNRS research group ISIS on Information, Signal, Images and Vision. He is also in charge of the PhD students network of this group. Cédric Richard is a member of GRETSI association board, and of the EURASIP and IEEE-SP societies. He serves also as an associate editor of the IEEE Transactions on Signal Processing.