Guaranteed mobile robot tracking using interval analysis

Michel Kieffer, Luc Jaulin, Eric Walter, Dominique Meizel

GUARANTEED MOBILE ROBOT TRACKING USING INTERVAL ANALYSIS

Michel KIEFFER[1], **Luc JAULIN**[1,2], **Éric WALTER**[1] and **Dominique MEIZEL**[3]

[1]*Laboratoire des Signaux et Systèmes, CNRS-Supélec*
Plateau de Moulon, 91192 Gif-sur-Yvette, France
{kieffer, jaulin, walter}@lss.supelec.fr
[2]*on leave from*
Laboratoire d'Ingénierie des Systèmes Automatisés,
Université d'Angers
[3]*HEUDIASYC, UMR CNRS 6599,*
Université de Technologie de Compiègne
B.P. 20529 Compiègne Cédex, France
dominique.meizel@utc.fr

Abstract: The problem considered here is state estimation in the presence of bounded process and measurement noise. A new nonlinear state estimator, based on interval analysis and the notion of set inversion, is applied to robot localization and tracking. This estimator evaluates a set *guaranteed* to contain all values of the state that are consistent with the available observations, given the noise bounds and some possibly very large set containing the initial value of the state. Three situations are considered to illustrate the properties of the estimator.

Keywords: Bounded-error estimation, Interval analysis, Robot localization, Robot tracking, State estimation.

1 Introduction

Much of recent research in robotics has been devoted to increasing autonomy, *e.g.*, by adding sensors, mobility and decision capability. To be autonomous, robots must be able to estimate their present state from available prior information and measurements.

The problem to be considered here is the autonomous localization and tracking of a robot such as that described by Figure 1, using distance measurements provided by a belt of on-board exteroceptive sensors. Ultrasonic sensors are used, but other types of sensors could also be considered, with the same methodology. The environment is assumed to be two-dimensional (although a three-dimensional extension poses no problem in principle), and a map of its landmarks is available to the robot.

In this paper, the methods developed in [11] for static robot localization and in [9] for state estimation are combined. The model of the robot and its environment is presented in Section 2. In Section 3, an interval method recently developed for the guaranteed localization of a robot is briefly recalled. In Section 4, a guaranteed state estimator based on interval analysis and the notion of subpaving is presented. Application of the methodology described in Sections 3 and 4 to robot tracking is reported in Section 5.

2 The robot and its map

The vehicle considered has a single body and its motions are generated by two coaxial independently driven wheels. Its displacement is planned in a 2D environment with respect to a set of

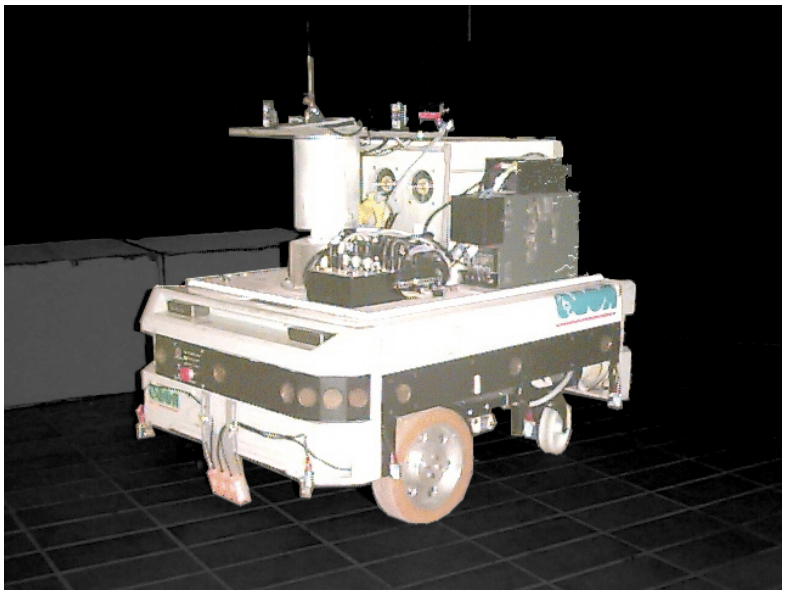

Figure 1: *Robuter* mobile robot by *Robosoft*.

landmarks and obstacles among which the robot has to move. Part of these landmarks define the world reference frame $\mathcal{W}$, in which the mission is defined. Let $\mathcal{R}$ be a reference frame tied to the vehicle (see Figure 2). The *configuration* of the vehicle in the 2D-world is $\mathbf{x} = (x_c, y_c, \theta)^{\mathrm{T}}$, where x_c and y_c are coordinates of a characteristic point $\mathbf{c}$ which defines the origin of $\mathcal{R}$, and θ, the heading angle of the robot, is the angle between $\mathcal{R}$ and $\mathcal{W}$. In what follows, $\mathbf{x}$ will be considered as a state vector, since a kinematic model of motion will be used. Points and their coordinates will be denoted by bold lower-case letters in $\mathcal{W}$ and by tilded bold lower-case letters in $\mathcal{R}$. Thus, for example, a sensor and the coordinates of its emission cone will be denoted by $\mathbf{s}$ in $\mathcal{W}$ and $\tilde{\mathbf{s}}$ in $\mathcal{R}$, with

$$\mathbf{s} = \begin{pmatrix} x_c \\ y_c \end{pmatrix} + \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tilde{\mathbf{s}},$$

where x_c and y_c are the coordinates of $\mathbf{c}$ in $\mathcal{W}$.

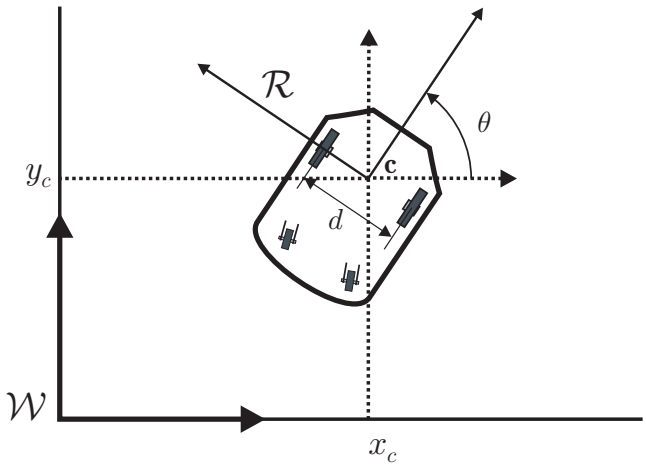

Figure 2: Configuration of the robot.

Given some (possibly very large) initial search box $[\mathbf{x}_0]$ in configuration space, robot localization can be formulated as the task of characterizing the set $\mathcal{X}_0 = \{\mathbf{x} \in [\mathbf{x}_0] \mid t(\mathbf{x}) \text{ holds true}\}$,

where $t(\mathbf{x})$ is some test expressing that the state $\mathbf{x}$ is consistent with the measurements and prior information. This test is built using informations given to the robot, namely distance measurements and a map of the landmarks.

The robot is equipped with a belt of n_s on-board Polaroid ultrasonic sensors (sonars). The position of the ith sensor in the robot frame $\mathcal{R}$ is $\tilde{\mathbf{s}}_i = (\tilde{x}_i, \tilde{y}_i)$. This sensor emits in a cone characterized by its vertex $\tilde{\mathbf{s}}_i$, orientation $\tilde{\theta}_i$ and half-aperture $\tilde{\gamma}_i$ (Figure 3). As $\tilde{\gamma}_i$ is frame-independent, $\tilde{\gamma}_i = \gamma_i$. The emission cone of the ith sensor will be denoted by $\mathcal{C}\left(\tilde{\mathbf{s}}_i, \tilde{\theta}_i, \tilde{\gamma}_i\right)$.

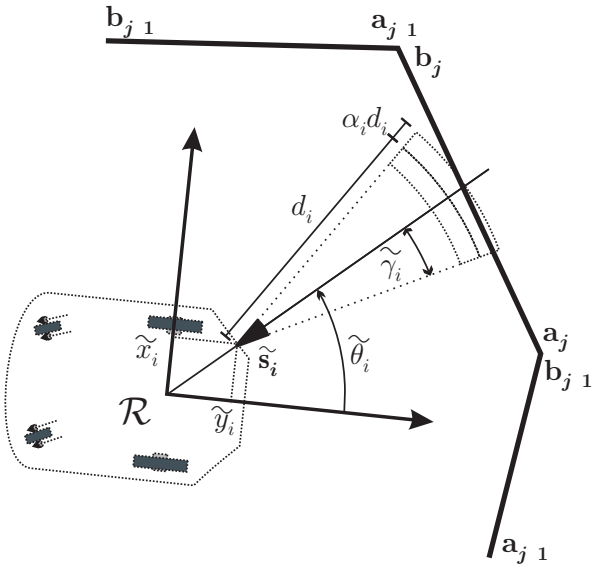

Figure 3: Emission cone.

The ith sensor measures the time-lag between emission and reception of the wave reflected or refracted by some landmark. This time-lag is then converted into a distance d_i to some obstacle, so far unidentified. To take measurement inaccuracy into account, each data point d_i is associated with the interval $[d_i] = [d_i(1-\alpha_i), d_i(1+\alpha_i)]$, where α_i is the known relative measurement accuracy of sensor i. Thus, $[d_i]$ is assumed to contain the actual distance to the closest reflecting landmark intercepting at least part of the ith emission cone.

To localize itself, the robot uses a *map* $\mathcal{M}$ of the environment. This map is assumed to consist of n_w oriented segments which describe the landmarks (walls, pillars, etc.): $\mathcal{M} = \{[\mathbf{a}_j, \mathbf{b}_j] \,|\, j = 1, ..., n_w\}$. By convention, when going from $\mathbf{a}_j$ to $\mathbf{b}_j$, the reflecting face of the segment is on the left-hand side.

3 Static localization

To check whether a given state $\mathbf{x}$ is consistent with the measured outputs $\{[d_i]\}_{i=1}^{n_s}$, the robot evaluates the measurements that its sensors would return if it were in the state $\mathbf{x}$ and compares them with the actual measurements. The test $t(\mathbf{x})$ must hold true if and only if they are deemed compatible; $t(\mathbf{x})$ is based on the notion of *remoteness*. Consider first a sensor i and a single segment j of the map. If the segment does not lie in the emission cone, or if the sensor is on the non-reflecting side of the segment, then the remoteness r_{ij} of i from j is infinite (Figure 4(a) and (b)). Otherwise, r_{ij} is finite and corresponds to the distance between the sensor and the intersection of the segment of the map and the emission cone (Figure 4(c)). When all segments of the map are taken into account, the remoteness r_i of i from the map is given by

$r_i = \min_{j=1,\ldots,n_w} r_{ij}$. It is consistent with the measured output if $r_i \in [d_i]$. The state $\mathbf{x}$ is consistent with all measurements (and thus $t(\mathbf{x}) = 1$) if all r_i's are consistent with the map. More details on the evaluation of remoteness and techniques improving the previous test may be found in [11].

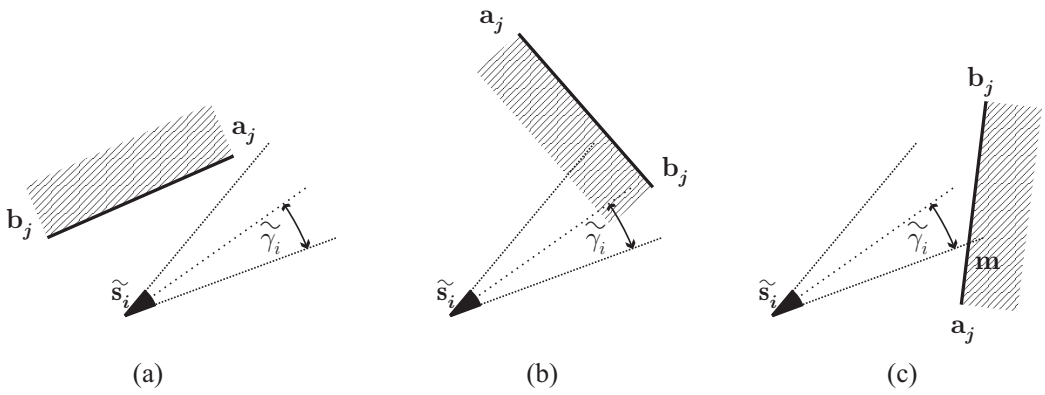

Figure 4: Evaluation of remoteness r_{ij}: (a) and (b) $r_{ij} = \infty$, (c) $r_{ij} = \|\tilde{\mathbf{s}}_i \mathbf{m}\|$.

When the robot is motionless, the set of all configurations in the prior box that are consistent with all measurements can be written as

$$\mathcal{X}_0 = \{\mathbf{x} \in [\mathbf{x}_0] \,|\, t(\mathbf{x}) \text{ holds true}\},$$

where $t(\mathbf{x})$ holds true if and only if $(r_i(\mathbf{x}) \in [d_i])_{i=1}^{n_s}$. Equivalently, $\mathcal{X}_0 = \mathbf{r}^{-1}_{[\mathbf{x}_0]}([\mathbf{d}])$. Except in very particular cases, it is not possible to evaluate the set $\mathcal{X}_0$, but it is possible to get a guaranteed outer approximation of $\mathcal{X}_0$, as accurate as desired, using *subpavings* [9] and the SIVIA (Set Inversion Via Interval Analysis) algorithm ([7], [8]).

A subpaving is a finite set of non-overlapping boxes that are all included in some *root* box. It is called *regular* when all its boxes result from successive bisections of the root box according to some canonical bisection law. Any compact subset $\mathcal{X}$ can be approximated with any desired precision using a regular subpaving $\widehat{\mathcal{X}}$. Obviously, a subpaving may be empty or consist of its entire root box.

Characterizing $\mathcal{X}_0$ belongs to the class of *set-inversion* problems, formulated as follows: given two sets $\mathcal{X} \subset \mathbb{R}^n$, $\mathcal{Y} \subset \mathbb{R}^m$ and a function $\mathbf{f}: \mathbb{R}^n \to \mathbb{R}^m$, characterize the set $\mathbf{f}^{-1}_{\mathcal{X}}(\mathcal{Y}) = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{f}(\mathbf{x}) \in \mathcal{Y}\}$. SIVIA characterizes $\mathbf{f}^{-1}_{\mathcal{X}}(\mathcal{Y})$ by bracketing it between inner and outer subpavings. Its convergence has been studied in [8]. In [9], a recursive version of SIVIA, evaluating the outer subpaving only, is presented. It returns a subpaving $\widehat{\mathcal{S}}_\epsilon = \text{SIVIA}(\widehat{\mathcal{X}}, \mathbf{f}_{[]}, \widehat{\mathcal{Y}}, \epsilon)$ containing $\mathbf{f}^{-1}_{\widehat{\mathcal{X}}}\left(\widehat{\mathcal{Y}}\right)$, where $\mathbf{f}_{[]}$ is an inclusion function for $\mathbf{f}$ and ϵ a precision parameter.

So $\mathcal{X}_0$ can easily be approximated by $\widehat{\mathcal{X}_0} = \text{SIVIA}([\mathbf{x}_0], \mathbf{r}^{-1}_{[]}, [\mathbf{d}], \epsilon)$, which encloses all states consistent with measurements.

In the dynamical case, this procedure could be repeated periodically, but this approach would have two drawbacks. Firstly it would be uselessly time-consuming, because no information about the previously evaluated configurations would be taken into account. Secondly, it would often lead to a more pessimistic characterization, because past measurements would be neglected. Using a model of the dynamics of the robot should help predict the configuration of the robot after a move and thus reduce the prior feasible domain in state space in which consistency with the new measurements should be studied. This policy is based on the recursive state-estimation algorithm presented in [9] and briefly recalled in Section 4.

4 Guaranteed state estimation

Consider the nonlinear and possibly time-varying discrete time-system defined by

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k), \\ \mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{w}_k, \end{cases} \quad k = 0, 1, ..., \tag{2}$$

where $\mathbf{u}_k \in \mathbb{R}^m, \mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{y}_k \in \mathbb{R}^p$ are respectively the input, state, and output vectors. The initial state $\mathbf{x}_0$ is assumed to be included in some prior compact set $\mathcal{X}_0 \subset \mathbb{R}^n$. $\{\mathbf{v}_k\}$ and $\{\mathbf{w}_k\}$ are unknown state and measurement noise sequences, respectively assumed to belong to the known intervals sequences $\{[\mathbf{v}]_k\}$ and $\{[\mathbf{w}]_k\}$. $\mathbf{f}_k$ and $\mathbf{h}_k$ are known functions (or finite algorithms) evaluating $\mathbf{x}_{k+1}$ and $\mathbf{y}_k$ at each step k. Let $\mathcal{X}_l$ be the smallest set guaranteed to contain all values of $\mathbf{x}_l$ compatible with the information available at time l, *i.e.*, with

$$\mathcal{I}_l = \left\{\mathcal{X}_0, \{\mathbf{u}_k, \mathbf{y}_k, [\mathbf{v}]_k, [\mathbf{w}]_k\}_{k=0}^{l}\right\}. \tag{3}$$

We shall first present an idealized algorithm to estimate $\mathcal{X}_l$ recursively, and then describe an actual algorithm for computing an outer approximation of $\mathcal{X}_l$. Let $\mathcal{X}_{l+}$ be the set of all values of the state that are reachable from some $\mathbf{x}_l$ in $\mathcal{X}_l$ with input $\mathbf{u}_l$ for some state noise $\mathbf{v}_l \in [\mathbf{v}]_l$:

$$\mathcal{X}_{l+} = \mathbf{f}_l(\mathcal{X}_l, \mathbf{u}_l, [\mathbf{v}]_l) = \{\mathbf{f}_l(\mathbf{x}, \mathbf{u}_l, \mathbf{v}_l) \mid \mathbf{x} \in \mathcal{X}_l, \mathbf{v}_l \in [\mathbf{v}]_l\}.$$

Moreover, let $\mathcal{Y}_{l+1}$ be the set of all admissible values of the output, when its measured value is $\mathbf{y}_{l+1}$

$$\mathcal{Y}_{l+1} = \mathbf{y}_{l+1} - [\mathbf{w}_{l+1}] = \{\mathbf{y}_{l+1} - \mathbf{w}_{l+1} \mid \mathbf{w}_{l+1} \in [\mathbf{w}]_{l+1}\},$$

and let $\mathcal{X}_{l+1}^o$ be the set of all values of $\mathbf{x}$ which could have led to an observation $\mathbf{y} \in \mathcal{Y}_{l+1}$

$$\mathcal{X}_{l+1}^o = \mathbf{h}_{l+1}^{-1}(\mathcal{Y}_{l+1}) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{h}_{l+1}(\mathbf{x}) \in \mathcal{Y}_{l+1}\}.$$

A set containing all values of $\mathbf{x}_l$ compatible with $\mathcal{I}_{l+1}$ is then given by

$$\mathcal{X}_{l+1} = \mathcal{X}_{l+} \cap \mathcal{X}_{l+1}^o.$$

As $\mathcal{X}_0$ contains all possible values of the initial state vector $\mathbf{x}_0$, one may thus, at least theoretically, recursively evaluate $\mathcal{X}_l$. These ideas are summarized in the following idealized algorithm, which parallels a Kalman filter in its structure.

Algorithm 1

For $l = 0$ to L, do

(a) Prediction: $\mathcal{X}_{l+} = \mathbf{f}_l(\mathcal{X}_l, \mathbf{u}_l, [\mathbf{v}]_l)$.
(b) Correction: $\mathcal{X}_{l+1} = \mathbf{h}_{l+1}^{-1}(\mathcal{Y}_{l+1}) \cap \mathcal{X}_{l+}$.

Proposition 1 *[9] $\mathcal{X}_l$, as computed by Algorithm 1, is the smallest set guaranteed to contain $\mathbf{x}_l$ that can be computed from $\mathcal{I}_l$.*

As in the case of static localization, it will usually not be possible to evaluate the sets $\mathcal{X}_{l+}$ and $\mathcal{X}_{l+1}$ exactly, but one may obtain a guaranteed outer approximation of $\mathcal{X}_{l+1}$, as accurate as desired.

The *correction step* requires characterizing $\mathcal{X}_{l+1} = \{\mathbf{x} \in \mathcal{X}_{l+} \mid \mathbf{h}_{l+1}(\mathbf{x}) \in \mathcal{Y}_{l+1}\}$. This task is again performed in a approximated way using SIVIA:

$$\widehat{\mathcal{X}_{l+1}} = \text{SIVIA}(\widehat{\mathcal{X}_{l+}}, \mathbf{h}_{l+1[]}, \widehat{\mathcal{Y}_{l+1}}, \epsilon).$$

The task performed by the *prediction step* can be included in the more general problem of direct image evaluation, which is at the core of interval arithmetic: given two compact sets

$\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{S}_0 \subset \mathbb{R}^m$ and a function $\mathbf{f}:\mathbb{R}^n \to \mathbb{R}^m$, characterize the set $\mathcal{S} \subset \mathcal{S}_0$ such that $\mathcal{S} = \{\mathbf{f}(\mathbf{x}) \in \mathcal{S}_0 \mid \mathbf{x} \in \mathcal{X}\}$. Two procedures, depending on whether $\mathbf{f}$ can be inverted, have been presented in [9]. They are shortly recalled here.

When $\mathbf{f}$ is invertible, prediction can again be cast in the formalism of set inversion, as the problem of finding $\mathcal{S} = \left\{\mathbf{x} \in \mathcal{S}_0 \middle| \mathbf{f}^{-1}(\mathbf{x}) \in \mathcal{X}\right\}$. The prior search set $\mathcal{S}_0$ should be taken large enough to be guaranteed to contain the set of interest. If $\widehat{\mathcal{S}_0}$ and $\widehat{\mathcal{X}}$ are subpavings enclosing $\mathcal{S}_0$ and $\mathcal{X}$, then $\mathcal{S}$ can be approximated by

$$\widehat{\mathcal{S}}_\epsilon = \textsc{Sivia}(\widehat{\mathcal{S}_0},\ \mathbf{f}_{[]}^{-1},\ \widehat{\mathcal{X}},\ \epsilon),$$

provided that an inclusion function $\mathbf{f}_{[]}^{-1}$ is available for $\mathbf{f}^{-1}$. Assuming that $\mathbf{f}$ is invertible may seem rather strong, but in many physical cases inverting the dynamics only means inverting time, so the inverse dynamics is rather simple to obtain from the direct one.

When $\mathbf{f}$ is not invertible, a specific and computationally more demanding procedure is needed. The basic idea of the direct Image SubPaving evaluation procedure (ImageSp) is to describe the initial set $\mathcal{X}$ using a subpaving consisting of p boxes $[\mathbf{x}]_i$ whose widths are less than ϵ. Then ImageSp evaluates the image of each of these p boxes using an inclusion function $\mathbf{f}_{[]}$ of $\mathbf{f}$ and stores these images in a list. One thus gets p *image* boxes, each of which contains the true image set of the associated initial box. The image set $\mathcal{S}$ is therefore included in the union of all of them. At last, ImageSp merges all these image boxes into a subpaving to allow further processing. The convergence of this algorithm has been studied in [10]. ImageSp returns the image subpaving $\widehat{\mathcal{S}}_\epsilon = \textsc{ImageSp}([\mathbf{s}]_0, \mathbf{f}_{[]},\ \widehat{\mathcal{X}},\ \epsilon)$ of the subpaving $\widehat{\mathcal{X}}$ by the function $\mathbf{f}$ (more exactly by one of its inclusion function $\mathbf{f}_{[]}$), or rather the part of it that is included in some prior search box $[\mathbf{s}]_0$. Again, ϵ is a precision parameter.

An approximate but guaranteed version of Algorithm 1 is then as follows:

Algorithm 2

For $l = 0$ to L, do

(a) Guaranteed prediction. *Compute the set estimate $\widehat{\mathcal{X}_{l+}}$ for the state at step $l+1$ before measurement either by*

$$\widehat{\mathcal{X}_{l+}} = \textsc{Sivia}(\widehat{\mathcal{S}}, \mathbf{f}_{l[]}^{-1}, \widehat{\mathcal{X}}_l \times \{\mathbf{u}_l\} \times [\mathbf{v}]_l, \epsilon);$$

if $\mathbf{f}_l$ is invertible, where $\widehat{\mathcal{S}} = \{[\mathbf{s}]\}$ is the search subpaving consisting of a possibly very large box in which all states are assumed to stay, or by

$$\widehat{\mathcal{X}_{l+}} = \textsc{ImageSp}([\mathbf{s}], \mathbf{f}_{l[]}, \widehat{\mathcal{X}}_l \times \{\mathbf{u}_l\} \times [\mathbf{v}]_l, \epsilon);$$

(b) Guaranteed correction. *From $\widehat{\mathcal{X}_{l+}}$, select all elements that are compatible with measurements at step $l+1$*

$$\widehat{\mathcal{X}_{l+1}} = \textsc{Sivia}(\widehat{\mathcal{X}_{l+}},\ \mathbf{h}_{l+1[]}, \widehat{\mathcal{Y}_{l+1}}, \epsilon);$$

5 Application to robot tracking

In order to use this state-estimation algorithm for robot tracking, the two functions $\mathbf{f}$ and $\mathbf{h}$ required in (2) have to be built. The observation equation is directly deduced from the remoteness function. To predict the configuration of the robot, a model of its dynamics based on the kinematic equations of the motion is used. This model is reliable only when slow motions are considered. Moreover, the wheels are supposed to roll without sliding. The sampling time is T, both driving wheels have the same radius R, and their angular rotation speeds are ω_p (for port side) and ω_s (for starboard side), assumed to be constant between two samples. The origin

$\mathbf{c}$ of $\mathcal{R}$ is taken as the middle of the segment joining the two independently controlled steering wheels (the two other wheels are free). When the state is $\mathbf{x}_l = ((x_c)_l, (y_c)_l, (\theta)_l)^{\mathrm{T}}$ at step l, one may easily show that the state at step $l+1$ is given by

$$\begin{pmatrix} (x_c)_{l+1} \\ (y_c)_{l+1} \\ (\theta)_{l+1} \end{pmatrix} = \begin{pmatrix} (x_c)_l + 2d\frac{\omega_s+\omega_p}{\omega_s-\omega_p} \cos\left((\theta)_l + T\frac{R(\omega_s-\omega_p)}{4d}\right) \sin\left(T\frac{R(\omega_s-\omega_p)}{4d}\right) \\ (y_c)_l + 2d\frac{\omega_s+\omega_p}{\omega_s-\omega_p} \sin\left((\theta)_l + T\frac{R(\omega_s-\omega_p)}{4d}\right) \sin\left(T\frac{R(\omega_s-\omega_p)}{4d}\right) \\ (\theta)_l + T\frac{R(\omega_s-\omega_p)}{2d} \text{ modulo } 2\pi \end{pmatrix}$$

if $\omega_s \neq \omega_p$, and

$$\begin{pmatrix} (x_c)_{l+1} \\ (y_c)_{l+1} \\ (\theta)_{l+1} \end{pmatrix} = \begin{pmatrix} (x_c)_l + T\frac{R(\omega_s+\omega_p)}{2} \cos(\theta)_l \\ (y_c)_l + T\frac{R(\omega_s+\omega_p)}{2} \sin(\theta)_l \\ (\theta)_l \end{pmatrix}$$

if $\omega_s = \omega_p$. No state noise has been considered here, but it could easily be introduced to take sliding or speed variation between samples into account.

Interval-based localization and tracking will now be illustrated on three fairly realistic simulated test cases. The characteristics of the robot are those of that of Figure 1. This robot is equipped with $n_s = 24$ ultrasonic sensors located on its periphery. As a result of experimentation on the actual robot, their emission angle $\tilde{\gamma}$ is taken as 0.2 rad and the distance relative inaccuracy α in the operating range is taken as 2%. In all test cases, the initial search domain in state space is $[-12 \text{ m}, 12 \text{ m}] \times [-12 \text{ m}, 12 \text{ m}] \times [0 \text{ rad}, 2\pi \text{ rad}]$; the precision parameter ϵ is taken as 0.1 m (for x_c and y_c) and 0.1 rad (for θ) and the sampling time as $T = 1$ s. The robot is assumed to be located in the room described by Figure 5, and the map available to the robot is supposed to match the environment. All computations were performed on a P233MMX personal computer, using a C++ implementation.

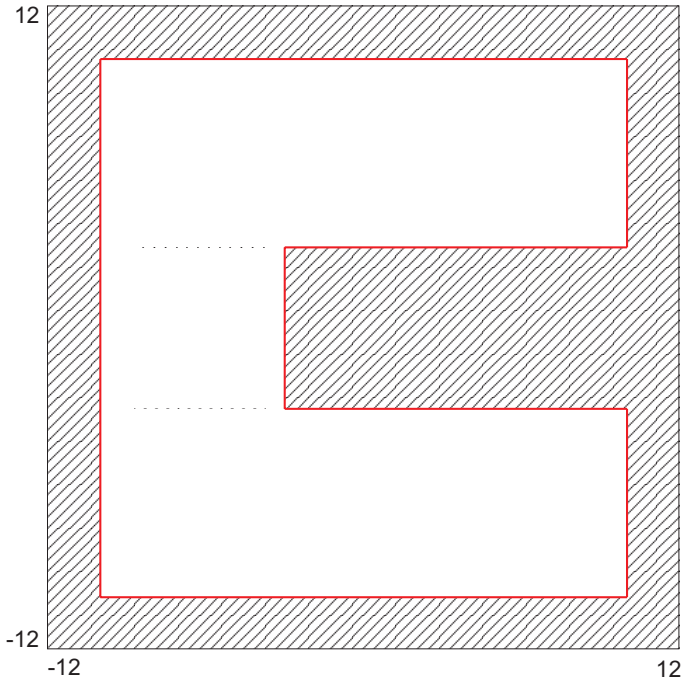

Figure 5: Map used by the robot for the test-cases.

The *first test case* illustrates the properties of the initial localization algorithm and of the prediction step alone: no correction step is applied to take into account new information on the state of the robot. So the state at step l is obtained from initial measurements by cascading l predictions.

The emission diagram of the 24 sensors during these initial measurements is represented on Figure 6. Two arcs are associated with each sensors, between which some obstacle(s) should lie at least in part.

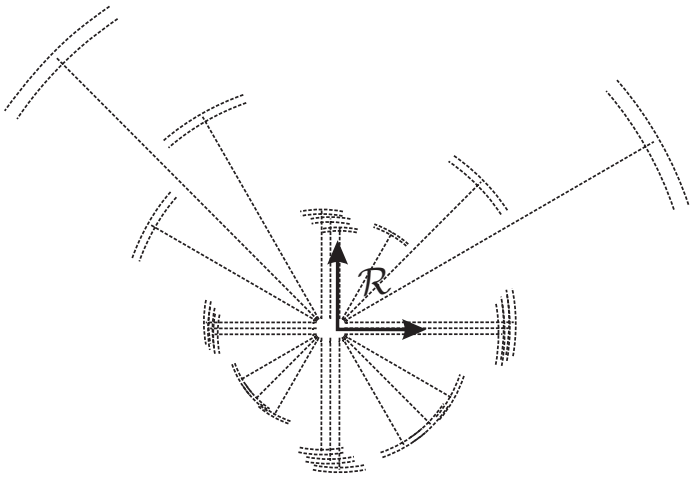

Figure 6: Emission diagram during initial measurements (Test Case 1); the scale is as in Figure 5.

This diagram was obtained using the algorithm described in Section 3 for the actual initial state $(x_c, y_c, \theta)_0 = (-5 \text{ m}, 6 \text{ m}, 4 \text{ rad})$. This information was obviously not made available to the tracking algorithm. First, a static localization is performed, which corresponds to the correction step with a search space equal to the prior feasible configuration space. A first subpaving enclosed in the box $[-5.09 \text{ m}, -4.89 \text{ m}] \times [5.93 \text{ m}, 6.07 \text{ m}] \times [3.99 \text{ rad}, 4.01 \text{ rad}]$, and guaranteed to contain the actual initial state, is found in less than 8 s. Then, the robot starts moving with linear wheel speeds $v_p = R\omega_p = 1.2 \text{ m.s}^{-1}$ and $v_s = R\omega_s = 1.6 \text{ m.s}^{-1}$, so the robot trajectory should be an arc of circle in (x, y)-space. Prediction is applied for $lT \in [0 \text{ s}, 6 \text{ s}]$. At each step l, the predicted subpaving $\widehat{\mathcal{X}_l^+}$ is evaluated using IMAGESP, and displayed on Figures 7 and 8. No correcting step is performed, so $\widehat{\mathcal{X}_{l+1}}$ is taken equal to $\widehat{\mathcal{X}_l^+}$.

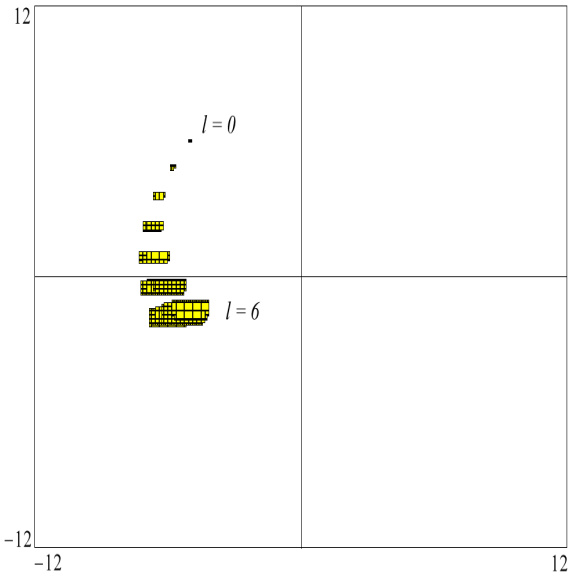

Figure 7: First test case. Projection onto the (x, y)-plane.

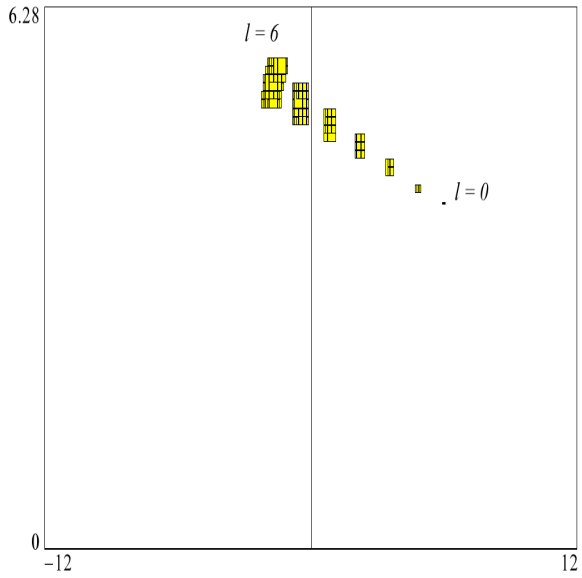

Figure 8: First test case. Projection onto the (y, θ)-plane.

As could be expected, the predicted set grows at each step, due to the accumulation of prediction errors (IMAGESP gives only an outer approximation of the image of a subpaving, and these approximations accumulate). Thus, at step $l = 6$, the subpaving containing the possible state is enclosed in $[-6.85 \text{ m}, -4.12 \text{ m}] \times [-2.25 \text{ m}, -1.03 \text{ m}] \times [5.10 \text{ rad}, 5.70 \text{ rad}]$, the volume of the subpaving has been multiplied by more than 2000 in six steps. This first example illustrates the difficulty of performing realistic tracking without taking observations into account.

In the *second test case*, the conditions (initial configuration, wheel speeds) are the same as in the first one, but now, at each step l, prediction is updated using observations. The tracking algorithm is applied for $lT \in [0 \text{ s}, 13 \text{ s}]$. At each step l, the predicted subpaving $\widehat{\mathcal{X}_l^+}$ is evaluated using IMAGESP, and the corrected subpaving $\widehat{\mathcal{X}_{l+1}}$ is evaluated using SIVIA. The corrected subpavings are presented on Figures 9 and 10.

The evaluation of the 13 steps takes 10 s. In this test case, the observations allow the uncertainty on the robot configuration to be kept within reasonable bounds. It would be possible to improve these bounds, at a higher computing time cost, by reducing the precision factor ϵ. On Figure 10, the state seems to jump, this is only due to the fact that θ remains into $[0, 2\pi]$.

The *third test case* illustrates the ability of the tracking algorithm to estimate the state of a robot the configuration of which is only known to belong to one of possibly many disconnected sets in configuration space. The room that the robot is supposed to belong to is the same as before. Actual initial configuration is $(x_c, y_c, \theta)_0 = (6, 6.5, \pi)$. Figure 11 represents two possible states belonging to the subpaving $\widehat{\mathcal{X}_0}$, which consists of two disconnected subsets because of the symmetry of the room, and is not presented here for the sake of brevity.

The tracking algorithm is applied, now with $v_p = R\omega_p = 0.6 \text{ m.s}^{-1}$ and $v_s = R\omega_s = 0.6 \text{ m.s}^{-1}$. Figure 12 represents the projection in the (x, y)-space of the subpavings guaranteed to contain the actual configuration at each step.

Both disconnected feasible state sets are tracked until information is available to prove that one of them should become empty. Only corrected subpavings are presented. Figure 13 illustrates the measurements available to the robot for step $l = 9$, when the lower configuration set is eliminated.

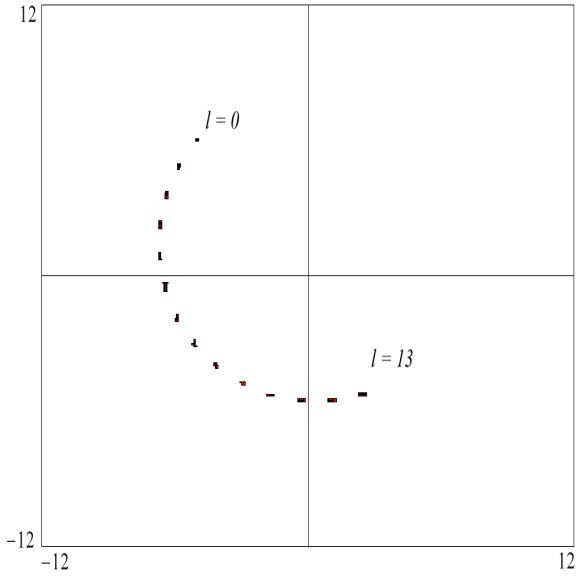

Figure 9: Second test case. Projection onto the (x, y)-plane.

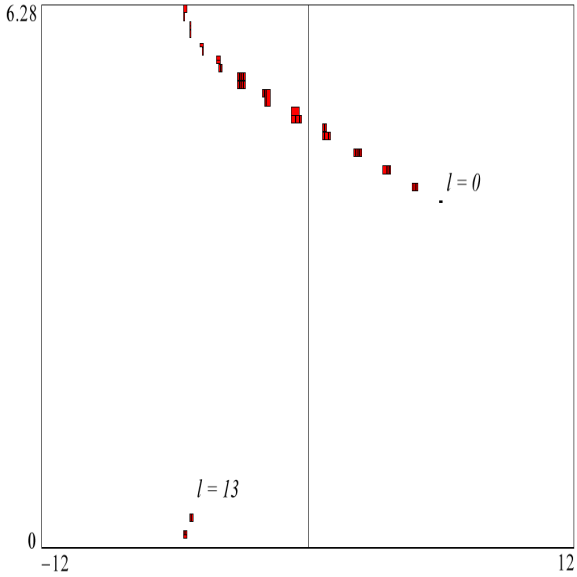

Figure 10: Second test case. Projection onto the (y, θ)-plane.

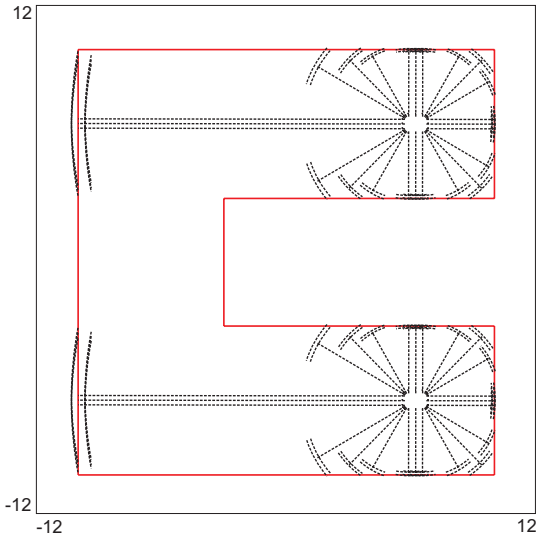

Figure 11: Third test case. Two possible initial states belonging to $\mathcal{X}_0$.

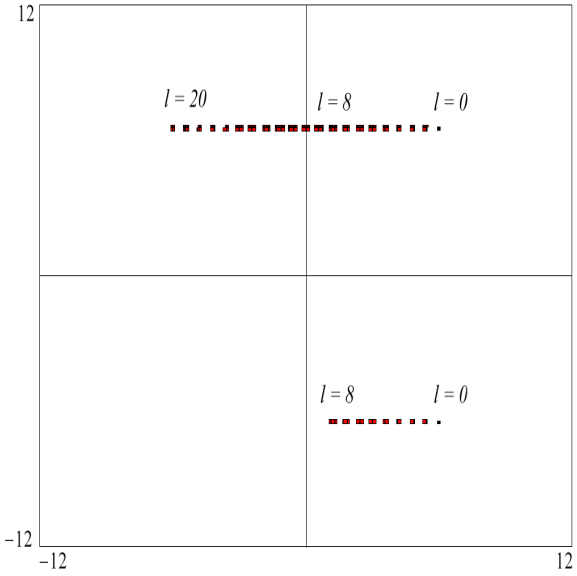

Figure 12: Third test-case. Projection onto the (x, y)-plane. One subset disappears at step 9.

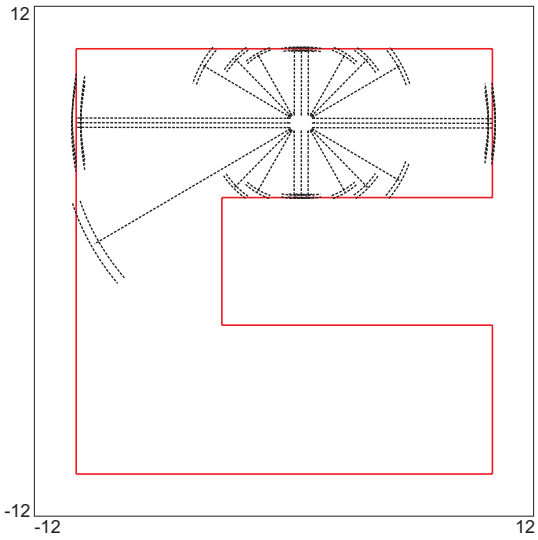

Figure 13: Third test case. Step 9 eliminates ambiguity.

The algorithm takes 17 s for 20 steps. It could thus be implemented for real-time tracking. As long as the volume of the subpaving remains small (the configuration is fairly well known), prediction and correction are fast.

6 Conclusions

Some properties of an interval-based recursive non-linear state estimation algorithm have been illustrated on a non-trivial robot tracking example. At each estimation step, the estimator has the important property of delivering a set guaranteed to contain all states that are consistent with the measurements returned by the sensors to the robot. Even when disconnected sets of configurations have to be considered, each of them is tracked until information is available to eliminate it.

The localization procedure used in the tracking algorithm does not suffer some of the traditional drawbacks of previous localization methods. It delivers a guaranteed solution, unlike the methods based on the extended Kalman filter (see [2], [12] and [13]) or bounded-error techniques requiring a linearization, such as described in [1], [16], [14] and [4]. It does not need a separate matching algorithm to recognize the environment prior to the localization, as in [3] and [5]. It directly manages multiple hypothesis contrary to [6]. It does not request any separate initialization procedure ([15] and [12]).

The ultrasonic sensors are known to be inaccurate; moreover, they often return outliers. Thus, before actual real-time implementation, a robust version of the algorithm has to be tested. The prediction step will remain unchanged, the correction step needs some modifications to tolerate measurements that do not satisfy the initial hypothesis made on the noise. This is currently under investigation.

References

[1] D. P. Bertsekas and I. B. Rhodes. Recursive state estimation for a set-membership description of uncertainty. *IEEE Trans. on Automatic Control*, 16:117–128, 1971.

[2] J. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Proc. IEEE International Conference on Robotics and Automation*, pages 674–680, Scottsdale, Arizona, 1989.

[3] M. Drumheller. Mobile robot localization using sonar. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(2):325–332, 1987.

[4] C. Durieu, B. Polyak, and E. Walter. Trace versus determinant in ellipsoidal outer bounding with application to state estimation. In *Proc. 13th IFAC World Congress*, volume I, pages 43–48, San Francisco, 1996.

[5] W. E. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(4):469–482, 1987.

[6] E. Halbwachs and D. Meizel. Multiple hypothesis management for mobile vehicule localization. In *CD Rom of the European Control Conference*, Louvain, 1997.

[7] L. Jaulin. *Solution globale et garantie de problèmes ensemblistes ; application à l'estimation non linéaire et à la commande robuste.* Phd dissertation, Université Paris-Sud, Orsay, 1994.

[8] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.

[9] M. Kieffer, L. Jaulin, and E. Walter. Guaranteed recursive nonlinear state estimation using interval analysis. To appear in *Proc. 37th IEEE Conference on Decision and Control*, 1998.

[10] M. Kieffer, L. Jaulin, and E. Walter. Guaranteed recursive nonlinear state estimation using interval analysis. Internal report (long version of this paper), Laboratoire des Signaux et Systèmes, July 1998.

[11] M. Kieffer, L. Jaulin, E. Walter, and D. Meizel. Robust autonomous robot localization using interval analysis. Internal report SYS98-10, Laboratoire des Signaux et Systèmes, September 1998.

[12] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics and Automation*, 7(3):376–382, 1991.

[13] J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation.* Kluwer Academic Publishers, Boston, 1992.

[14] D. Maksarov and J. P. Norton. State bounding with ellipsoidal set description of the uncertainty. *Int. J. of Control*, 65(5):847–866, 1996.

[15] J. Neira, J. Horn, J. D. Tardoz, and G. Schmidt. Multisensor mobile robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 673–679, Mineapolis, USA, 1996.

[16] F. C. Schweppe. *Uncertain Dynamic Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1973.