, , 1–26 ()

Robust Autonomous Robot Localization 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,*
2 bd Lavoisier, 49045 Angers, France
jaulin@babinet.univ-angers.fr

[3] *HEUDIASYC, CNRS, Université de Technologie de Compiègne,*
B.P. 20529, 60205 Compiègne, France
meizel@hds.univ-compiegne.fr

Editor:

Abstract. This paper deals with the determination of the position and orientation of a mobile robot from distance measurements provided by a belt of onboard ultrasonic sensors. The environment is assumed to be two-dimensional, and a map of its landmarks is available to the robot. In this context, classical localization methods have three main limitations. First, each data point provided by a sensor must be associated with a given landmark. This data-association step turns out to be extremely complex and time-consuming, and its results can usually not be guaranteed. The second limitation is that these methods are based on linearization, which makes them inherently local. The third limitation is their lack of robustness to outliers due, *e.g.*, to sensor malfunctions or outdated maps. By contrast, the method proposed here, based on interval analysis, bypasses the data-association step, handles the problem as nonlinear and in a global way and is (extraordinarily) robust to outliers.

Keywords: Interval Analysis - Identification - State Estimation - Outliers - Bounded Errors - Robotics.

1. Introduction

Robots are articulated mechanical systems employed for tasks that may be dull, repetitive and hazardous or may require skills or strength beyond those of human beings. They first appeared as *manipulating robots* with their base rigidly fixed, performing simple and well defined elementary tasks in a controlled workspace. Since then, much of the research in robotics has been devoted to increasing their autonomy, *e.g.*, by adding sensors, mobility and decision capability. *Mobile robots* may take various forms depending on the task and environment. To be autonomous, they must be able to estimate their present state from available prior information and measurements.

The problem to be considered here is the autonomous localization of a robot such as that described by Figure 1 from distance measurements provided by a belt of

* Corresponding author

onboard exteroceptive sensors. Here, ultrasonic sensors are used, which are known to be cheap but imprecise. Other types of sensors providing range data could 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. No special beacons need to be introduced in the environment to facilitate localization.

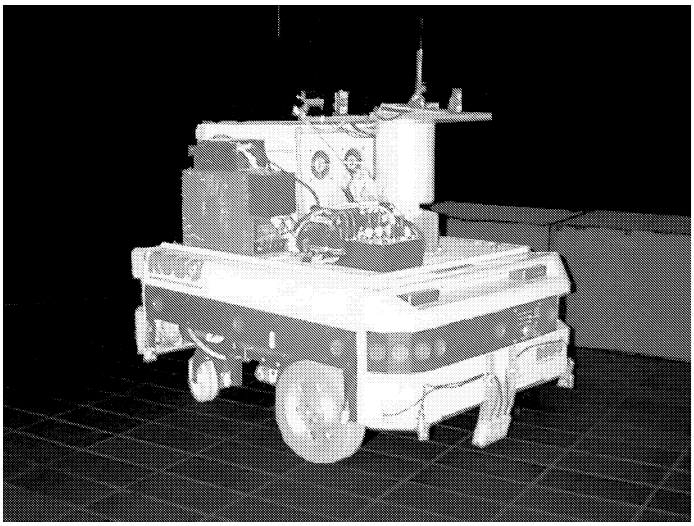

Figure 1. Robuter mobile robot by *Robosoft.*

In this context, classical localization methods [4], [2], [17], [6], [18], [21] and [5] have three main limitations. First, each data point provided by an exteroceptive sensor must be associated with a given landmark. This data-association step turns out to be extremely complex and time-consuming, and its results can usually not be guaranteed. The second limitation is that these methods are based on linearization, which makes them inherently local. The third limitation is their lack of robustness to outliers due, *e.g.*, to sensor malfunctions or outdated maps. By contrast, the method proposed here, which is based on bounded-error set estimation (see, *e.g.*, [27], [22], [23] and [20], and the references therein), bypasses the data-association step, handles the problem as nonlinear and in a global way (see also [19]) and is (extraordinarily) robust to outliers.

This paper is organized as follows. The problem is stated in mathematical terms in Section 2. Section 3 describes the elementary tests that will be used to locate the robot. Extension to intervals and combination of these tests are considered in Section 4. Section 5 describes the algorithm employed to characterize the set of all values of the localization parameters that satisfy the tests chosen. The resulting methodology is illustrated on three tests cases in Section 6, before drawing some conclusions in Section 7. The notation used is summarized in Section 8.

2. Formulation of the problem

Computation will involve two frames, namely the world frame $\mathcal{W}$ and a frame $\mathcal{R}$, of origin $\mathbf{c} = (x_c, y_c)$ in $\mathcal{W}$, tied to the robot. The angle between $\mathcal{R}$ and $\mathcal{W}$, denoted by θ, corresponds to the heading angle of the robot (see Figure 2). Points and their coordinates will be denoted by lower-case letters in $\mathcal{W}$ and by tilded lower-case letters in $\mathcal{R}$. Thus, a point $\tilde{\mathbf{m}}$ with coordinates $(\tilde{x}, \tilde{y})$ in $\mathcal{R}$ will be denoted by $\mathbf{m}$ in $\mathcal{W}$, with

$$\mathbf{m} = \begin{pmatrix} x_c \\ y_c \end{pmatrix} + \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}. \tag{1}$$

Three parameters are to be estimated, namely the coordinates x_c and y_c of the origin of $\mathcal{R}$ in $\mathcal{W}$ and the heading angle θ of the robot. They form the *configuration vector* $\mathbf{p} = (x_c, y_c, \theta)^{\mathrm{T}}$ (Figure 2). Given some (possibly very large) initial search box $[\mathbf{p}_0]$ in configuration space, robot localization can be formulated as the task of characterizing the set $\mathcal{S} = \{\mathbf{p} \in [\mathbf{p}_0] \mid t(\mathbf{p}) \text{ holds true}\}$, where $t(\mathbf{p})$ is a suitable test or combination of tests expressing that the robot configuration is consistent with the measurements and prior information.

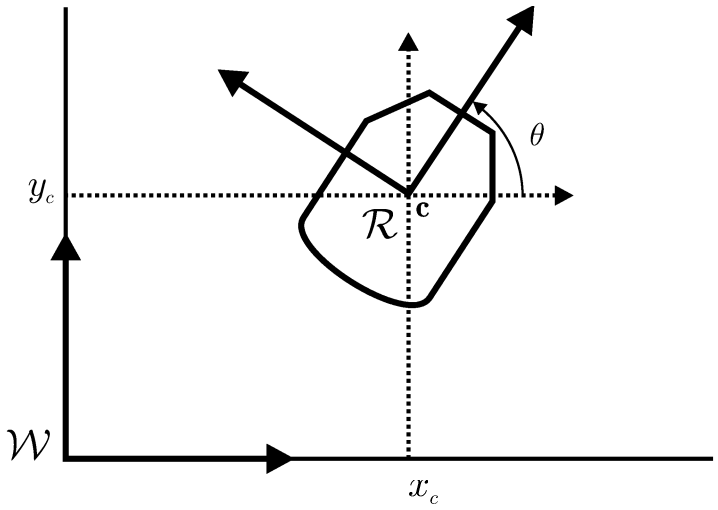

Figure 2. Configuration of the robot.

2.1. Measurements

The robot of Figure 1 is equipped with a belt of n_{s} onboard 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, $\tilde{\gamma}_i = \gamma_i$. This cone

will be denoted by $\mathcal{C}(\tilde{\mathbf{s}}_i, \tilde{\theta}_i, \tilde{\gamma}_i)$. The 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. 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.

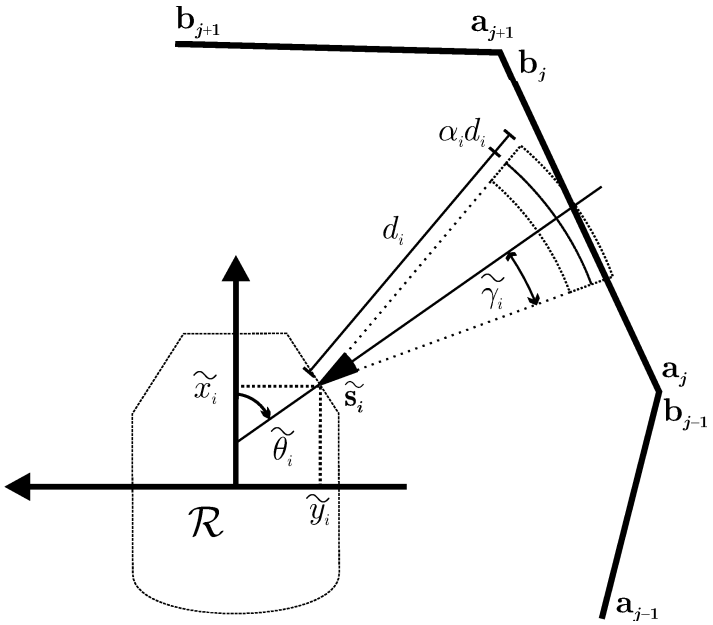

Figure 3. Emission cone.

2.2. Prior information

Two types of prior information will be considered. The first one is a *map* $\mathcal{M} = \{[\mathbf{a}_j, \mathbf{b}_j]\,|j = 1, ..., n_{\mathrm{w}}\}$ of the environment, assumed to consist of n_{w} oriented segments which describe the landmarks (walls, pillars, etc.). By convention, when going from $\mathbf{a}_j$ to $\mathbf{b}_j$, the reflecting face of the segment is on the left. The half-plane $\Delta_{\mathbf{a}_j\mathbf{b}_j}$ situated on the reflecting side of the segment $[\mathbf{a}_j, \mathbf{b}_j]$ is therefore characterized by

$$\Delta_{\mathbf{a}_j\mathbf{b}_j} = \left\{\mathbf{m} \in \mathbb{R}^2 \,\middle|\, \det\left(\overrightarrow{\mathbf{a}_j\mathbf{b}_j}, \overrightarrow{\mathbf{a}_j\mathbf{m}}\right) \geq 0\right\}. \tag{2}$$

The second type of information (optional) is the knowledge of a set described by polygons to which $\mathbf{p}$ is known *a priori* to belong.

3. Localization tests

This section enumerates various elementary tests that will be used to build the global test $t(\mathbf{p})$ employed to define $\mathcal{S}$.

3.1. Data-association test

To estimate the robot configuration from range data provided by ultrasonic sensors, it is of interest to build a *test* that checks whether a given configuration is consistent with these data, given their imprecision. For this purpose, information available in the robot frame $\mathcal{R}$ will be translated in the world frame $\mathcal{W}$. Consider any ultrasonic sensor of the robot, with emission cone $\mathcal{C}\left(\tilde{\mathbf{s}}, \tilde{\theta}, \tilde{\gamma}\right)$ (in this section, the indices i and j will be omitted to simplify presentation). For any given configuration $\mathbf{p} = (x_c, y_c, \theta)^{\mathrm{T}}$, $\mathcal{C}$ can be equivalently described in $\mathcal{W}$ by its vertex $\mathbf{s}(\mathbf{p})$ and by two unit vectors $\overrightarrow{\mathbf{u}_1}\left(\mathbf{p}, \tilde{\theta}, \tilde{\gamma}\right)$ and $\overrightarrow{\mathbf{u}_2}\left(\mathbf{p}, \tilde{\theta}, \tilde{\gamma}\right)$ corresponding to its edges, given by

$$\overrightarrow{\mathbf{u}_1} = \begin{pmatrix} \cos\left(\theta + \tilde{\theta} - \tilde{\gamma}\right) \\ \sin\left(\theta + \tilde{\theta} - \tilde{\gamma}\right) \end{pmatrix}, \quad \overrightarrow{\mathbf{u}_2} = \begin{pmatrix} \cos\left(\theta + \tilde{\theta} + \tilde{\gamma}\right) \\ \sin\left(\theta + \tilde{\theta} + \tilde{\gamma}\right) \end{pmatrix}. \tag{3}$$

So one may write $\mathcal{C} = \mathcal{C}\left(\mathbf{s}, \overrightarrow{\mathbf{u}_1}, \overrightarrow{\mathbf{u}_2}\right)$ (omitting the dependency in $\mathbf{p}, \tilde{\theta}$ and $\tilde{\gamma}$). By convention, $\overrightarrow{\mathbf{u}_1}$ and $\overrightarrow{\mathbf{u}_2}$ have been indexed so that $\overrightarrow{\mathbf{u}_2}$ is obtained from $\overrightarrow{\mathbf{u}_1}$ by a counterclockwise rotation of $2\tilde{\gamma}$. Since $\tilde{\gamma}$ is always less than $\pi/2$, the condition for any $\mathbf{m} \in \mathbb{R}^2$ to belong to the emission cone is

$$\mathbf{m} \in \mathcal{C}\left(\mathbf{s}, \overrightarrow{\mathbf{u}_1}, \overrightarrow{\mathbf{u}_2}\right) \Leftrightarrow \left(\det\left(\overrightarrow{\mathbf{u}_1}, \overrightarrow{\mathbf{sm}}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{\mathbf{u}_2}, \overrightarrow{\mathbf{sm}}\right) \leq 0\right). \tag{4}$$

The algorithm for testing a given configuration is based on the notion of *remoteness* of a segment from a sensor, which will now be defined. Consider first a single isolated segment $[\mathbf{a}, \mathbf{b}]$. Its remoteness from the sensor $\mathbf{s}$, associated with the cone $\mathcal{C}\left(\mathbf{s}, \overrightarrow{\mathbf{u}_1}, \overrightarrow{\mathbf{u}_2}\right)$, is defined as

$$\begin{aligned} r\left(\mathbf{s}, \overrightarrow{\mathbf{u}_1}, \overrightarrow{\mathbf{u}_2}, \mathbf{a}, \mathbf{b}\right) &= \infty \text{ if } \mathbf{s} \notin \Delta_{\mathbf{ab}} \text{ or if } [\mathbf{a}, \mathbf{b}] \cap \mathcal{C} = \emptyset, \\ &= \min_{\mathbf{m} \in [\mathbf{a}, \mathbf{b}] \cap \mathcal{C}} \|\overrightarrow{\mathbf{sm}}\| \text{ otherwise.} \end{aligned} \tag{5}$$

The remoteness function (5) is evaluated as follows. Equation (2) is used first to check whether $\mathbf{s} \in \Delta_{\mathbf{ab}}$. If this is so, minimization of $\|\overrightarrow{\mathbf{sm}}\|$ over $[\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$ is attempted. This requires taking different situations into account. Let $\mathbf{h}$ be the orthogonal projection of $\mathbf{s}$ onto the line $(\mathbf{a}, \mathbf{b})$. If $\mathbf{h} \in [\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$, then $r\left(\mathbf{s}, \overrightarrow{\mathbf{u}_1}, \overrightarrow{\mathbf{u}_2}, \mathbf{a}, \mathbf{b}\right) = \left\|\overrightarrow{\mathbf{sh}}\right\|$. To check whether $\mathbf{h} \in [\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$, without actually computing it, one may use the following relation:

$$\begin{aligned} \mathbf{h} \in [\mathbf{a}, \mathbf{b}] \cap \mathcal{C} \quad \Leftrightarrow & \left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{sa}} \right\rangle \leq 0\right) \wedge \left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{sb}} \right\rangle \geq 0\right) \\ & \wedge \left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{u}_1} \right\rangle \leq 0\right) \wedge \left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{u}_2} \right\rangle \geq 0\right). \end{aligned} \tag{6}$$

If $\mathbf{h} \notin [\mathbf{a},\mathbf{b}] \cap \mathcal{C}$, the minimum distance is either infinite (if $[\mathbf{a},\mathbf{b}] \cap \mathcal{C} = \emptyset$) or obtained for one of the extremities of the segment $[\mathbf{a},\mathbf{b}] \cap \mathcal{C}$. Let $\mathbf{h}_1$ and $\mathbf{h}_2$ be the intersections of the line $(\mathbf{a},\mathbf{b})$ with the lines $(\mathbf{s},\overrightarrow{\mathbf{u}_1})$ and $(\mathbf{s},\overrightarrow{\mathbf{u}_2})$. The set of possible ends of $[\mathbf{a},\mathbf{b}] \cap \mathcal{C}$ is thus $\mathcal{K} = \{\mathbf{a},\mathbf{b},\mathbf{h}_1,\mathbf{h}_2\}$. Therefore, if $\mathbf{h} \notin [\mathbf{a},\mathbf{b}] \cap \mathcal{C}$, then $r\left(\mathbf{s},\overrightarrow{\mathbf{u}_1},\overrightarrow{\mathbf{u}_2},\mathbf{a},\mathbf{b}\right)$ is either infinite or equal to $\|\overrightarrow{\mathbf{sm}}\|$, for some $\mathbf{m}$ in $\mathcal{K}$. For the example of Figure 4, $r\left(\mathbf{s},\overrightarrow{\mathbf{u}_1},\overrightarrow{\mathbf{u}_2},\mathbf{a},\mathbf{b}\right) = \left\|\overrightarrow{\mathbf{sb}}\right\|$.

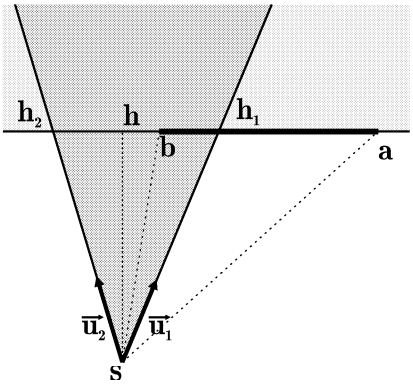

Figure 4. Remoteness of an isolated segment $[\mathbf{a},\mathbf{b}]$ from the sensor $\mathbf{s}$.

A test of whether any element of $\mathcal{K}$ belongs to $[\mathbf{a},\mathbf{b}] \cap \mathcal{C}$ is easily derived from (4). For $\mathbf{v} \in \{\mathbf{a},\mathbf{b}\}$

$$\mathbf{v} \in \mathcal{C} \iff \left(\det\left(\overrightarrow{\mathbf{u}_1},\overrightarrow{\mathbf{sv}}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{\mathbf{u}_2},\overrightarrow{\mathbf{sv}}\right) \leq 0\right). \tag{7}$$

By construction, $\mathbf{h}_i \in \mathcal{C} \cap (\mathbf{a},\mathbf{b})$; one thus has only to check whether $\mathbf{h}_i$ belongs to $[\mathbf{a},\mathbf{b}]$, which is equivalent to proving that $\mathbf{h}_i \in \mathcal{C}\left(\mathbf{s},\frac{\overrightarrow{\mathbf{sa}}}{\|\overrightarrow{\mathbf{sa}}\|},\frac{\overrightarrow{\mathbf{sb}}}{\|\overrightarrow{\mathbf{sb}}\|}\right)$. Thus, for $i = 1,2$,

$$\mathbf{h}_i \in [\mathbf{a},\mathbf{b}] \cap \mathcal{C} \iff \left(\det\left(\overrightarrow{\mathbf{sa}},\overrightarrow{\mathbf{u}_i}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{\mathbf{sb}},\overrightarrow{\mathbf{u}_i}\right) \leq 0\right). \tag{8}$$

Finally, if neither $\mathbf{h}$ nor any element of $\mathcal{K}$ belongs to $[\mathbf{a},\mathbf{b}] \cap \mathcal{C}$, then $[\mathbf{a},\mathbf{b}] \cap \mathcal{C} = \emptyset$, and the remoteness is infinite.

Appendix A presents a function, based on these tests, evaluating $r\left(\mathbf{s},\overrightarrow{\mathbf{u}_1},\overrightarrow{\mathbf{u}_2},\mathbf{a},\mathbf{b}\right)$ for an isolated segment $[\mathbf{a},\mathbf{b}]$.

Remark. This version of remoteness does not take into account the fact that if the incidence angle of the emitted wave is greater than a given angle (depending on the nature of the landmark), no wave will return to the sensor. This could easily be taken care of by modifying the definition of remoteness so as to take the incidence angle into account. Another phenomenon not considered is multiple reflection taking place, for instance, in concave corners. Accounting for multiple reflections would require a more complex definition of remoteness, and is probably

not worthwhile. As will be seen in Section 4.3, a much simpler route is to consider such measurements as outliers. ♢

In the normal situation where n_{w} segments are present, the fact that a given segment may not be detected, because it lies in the shadow of another one closer to the sensor, must be taken into account. Let $r_{ij}(\mathbf{p})$ be the remoteness of the jth segment, taken as isolated, from the ith sensor if the configuration is $\mathbf{p}$. This remoteness is given by

$$r_{ij}(\mathbf{p}) = r\left(\mathbf{s}_i(\mathbf{p}), \overrightarrow{\mathbf{u}_{1i}}\left(\mathbf{p}, \tilde{\theta}_i, \tilde{\gamma}_i\right), \overrightarrow{\mathbf{u}_{2i}}\left(\mathbf{p}, \tilde{\theta}_i, \tilde{\gamma}_i\right), \mathbf{a}_j, \mathbf{b}_j\right). \tag{9}$$

The remoteness of the map from the ith sensor if the configuration is $\mathbf{p}$ is then

$$r_i(\mathbf{p}) = \min_{j=1,\ldots,n_{\mathrm{w}}} r_{ij}(\mathbf{p}). \tag{10}$$

The measurement provided by the ith sensor may be explained by a segment lying at a proper distance if the following test is satisfied:

Test $dat_i(\mathbf{p})$**:** $dat_i(\mathbf{p})$ holds true if and only if $r_i(\mathbf{p}) \in [d_i]$.

3.2. In_room test

Assume that the map partitions the world into two sets, the *interior*, which the robot should belong to,

$$\mathcal{P}_{\mathrm{int}} = \left\{ \mathbf{m} \in \mathbb{R}^2 \,\middle|\, \sum_{j=1}^{n_{\mathrm{w}}} \arg\left(\overrightarrow{\mathbf{ma}_j}, \overrightarrow{\mathbf{mb}_j}\right) = 2\pi \right\}, \tag{11}$$

and the *exterior*

$$\mathcal{P}_{\mathrm{ext}} = \left\{ \mathbf{m} \in \mathbb{R}^2 \,\middle|\, \sum_{j=1}^{n_{\mathrm{w}}} \arg\left(\overrightarrow{\mathbf{ma}_j}, \overrightarrow{\mathbf{mb}_j}\right) = 0 \right\}, \tag{12}$$

where $\arg\left(\overrightarrow{\mathbf{ma}_j}, \overrightarrow{\mathbf{mb}_j}\right)$, the angle between $\overrightarrow{\mathbf{ma}_j}$ and $\overrightarrow{\mathbf{mb}_j}$, is constrained to belong to $]-\pi, \pi]$. The fact that $-\pi$ is excluded implies that the boundary between $\mathcal{P}_{\mathrm{int}}$ and $\mathcal{P}_{\mathrm{ext}}$ belongs to the interior. Figure 5 illustrates a situation where part of the room is forbidden by suitably oriented internal polygons. For each segment $[\mathbf{a}_j, \mathbf{b}_j]$, the arrow indicates the direction from $\mathbf{a}_j$ to $\mathbf{b}_j$. Recall that the reflecting face is on the left when going from $\mathbf{a}_j$ to $\mathbf{b}_j$.

If $\tilde{\mathbf{m}}$ is any point of the robot with coordinates $(\tilde{x}, \tilde{y})$ in $\mathcal{R}$, then its coordinates $\mathbf{m}$ in $\mathcal{W}$ evaluated according to (1) depend on the robot configuration $\mathbf{p} = (x_c, y_c, \theta)^{\mathrm{T}}$ and the following test will make it possible to eliminate some configurations for which it would not be in $\mathcal{P}_{\mathrm{int}}$.

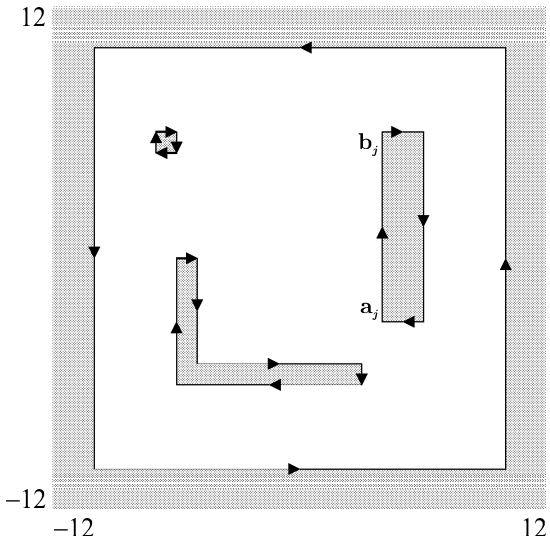

Figure 5. Partition of the world. The interior is in white.

Test *in_room* $(\mathbf{m})$**:**

$$\begin{cases} in\_room\,(\mathbf{m}) = 1 \ \text{ if } \sum_{j=1}^{n_w} \arg\left(\overrightarrow{\mathbf{ma}_j}, \overrightarrow{\mathbf{mb}_j}\right) = 2\pi, \\ in\_room\,(\mathbf{m}) = 0 \ \text{ otherwise.} \end{cases}$$

When $\mathbf{m}$ is the projection of $\mathbf{p}$ onto the $x \times y$ plane, *in_room* $(\mathbf{m})$ will be rewritten as *in_room* $(\mathbf{p})$.

As shown in Section 6, this test will contribute to eliminating configurations more efficiently than the data-association test alone, on purely geometrical grounds and in the absence of any measurements. However, it is reliable only when the map and the partition it induces are reliable. Even when this is not the case, this test remains of interest, as it forms the basis for the *leg_in* test presented below and still applicable.

Remark. Fictitious nonreflecting segments may be needed to define $\mathcal{P}_{\text{int}}$ and $\mathcal{P}_{\text{ext}}$. They may be transparent (open doors and windows), or absorbing. The reflectivity of each of these segments could be taken into account with a more elaborate definition of remoteness. With the definition adopted here, such segments may lead to outliers, see Section 4.3. ♢

3.3. Leg_in test

Consider a robot configuration $\mathbf{p} = (x_c, y_c, \theta)^{\mathrm{T}}$, the ith robot sensor $\mathbf{s}_i$, with coordinates $(\tilde{x}_i, \tilde{y}_i)$ in $\mathcal{R}$, and its associated interval measurement $[d_i]$. Let $\mathbf{c}_i$ be the

point at a distance equal to the lower bound $\underline{d_i}$ of $[d_i]$ from $\mathbf{s}_i$ in the direction of emission $\tilde{\theta}_i$. The coordinates of $\mathbf{c}_i$ in $\mathcal{W}$ satisfy

$$\mathbf{c}_i = \begin{pmatrix} x_c \\ y_c \end{pmatrix} + \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \tilde{x}_i + \cos\left(\tilde{\theta}_i\right) \underline{d_i} \\ \tilde{y}_i + \sin\left(\tilde{\theta}_i\right) \underline{d_i} \end{pmatrix}. \tag{13}$$

Assuming, as for *in_room*, that the world is partitionned into $\mathcal{P}_{\text{int}}$ and $\mathcal{P}_{\text{ext}}$, one can define

Test $leg\_in_i(\mathbf{p})$**:** $\quad leg\_in_i(\mathbf{p}) = \overline{in\_room(\mathbf{s}_i(\mathbf{p}))} \vee in\_room(\mathbf{c}_i)$.

The following result explains why this test can be used in conjunction with dat_i to eliminate configurations.

PROPOSITION 1 $\quad leg\_in_i(\mathbf{p}) = 0 \;\Rightarrow dat_i(\mathbf{p}) = 0.$ ◊

Proof: $leg\_in_i(\mathbf{p}) = 0$ implies that $\mathbf{s}_i$ is in $\mathcal{P}_{\text{int}}$ and $\mathbf{c}_i$ in $\mathcal{P}_{\text{ext}}$ (see Figure 6). Then there exists j such that $[\mathbf{s}_i, \mathbf{c}_i] \cap [\mathbf{a}_j, \mathbf{b}_j] \neq \emptyset$ and $\mathbf{s}_i \in \Delta_{\mathbf{a}_j\mathbf{b}_j}$. Let $\mathbf{m}_{ij} = [\mathbf{s}_i, \mathbf{c}_i] \cap [\mathbf{a}_j, \mathbf{b}_j]$. The ith cone intersects $[\mathbf{a}_j, \mathbf{b}_j]$ at least at $\mathbf{m}_{ij}$. So the remoteness of $[\mathbf{a}_j, \mathbf{b}_j]$ from $\mathbf{s}_i$ is less than or equal to $\|\overrightarrow{\mathbf{s}_i\mathbf{m}_{ij}}\|$. As $\mathbf{m}_{ij} \in [\mathbf{s}_i, \mathbf{c}_i[$, $\|\overrightarrow{\mathbf{s}_i\mathbf{m}_{ij}}\| < \|\overrightarrow{\mathbf{s}_i, \mathbf{c}_i}\| = \underline{d_i}$, and the remoteness of $[\mathbf{a}_j, \mathbf{b}_j]$ from $\mathbf{s}_i$ is therefore incompatible with $[d_i]$, so $\overline{dat_i}(\mathbf{p}) = 0$. ■

The test $leg\_in_i(\mathbf{p})$ thus provides a necessary condition for $\mathbf{p}$ to be consistent with the ith measurement. As this condition is not sufficient, $leg\_in_i(\mathbf{p})$ may hold true even when $dat_i(\mathbf{p})$ holds false. It will only be useful to eliminate some unfeasible configurations more quickly.

4. Interval tests

The tests presented in the preceding section for point configurations, should now be extended to interval configurations. The notion of *Boolean intervals* will be used to take the possible ambiguity of test results into account. It will then be possible to give interval counterparts of the localization tests, which will be associated to increase their efficiency.

4.1. Boolean intervals and inclusion tests

A Boolean interval is an element of $\mathbb{IB} = \{0, [0,1], 1\}$, where 0 stands for *false*, 1 for *true* and $[0,1]$ for *indeterminate*. It is a convenient object for implementing three-valued logic.

Table 1 specifies the `AND` ($\wedge$) and `OR` ($\vee$) operations between two Boolean intervals. As Boolean intervals are sets, standard set operators such as $\cup$ and $\cap$ also apply. They should not be confused with the logical operators $\vee$ and $\wedge$. For instance, $[0,1] \wedge 1 = [0,1]$ but $[0,1] \cap 1 = 1$.

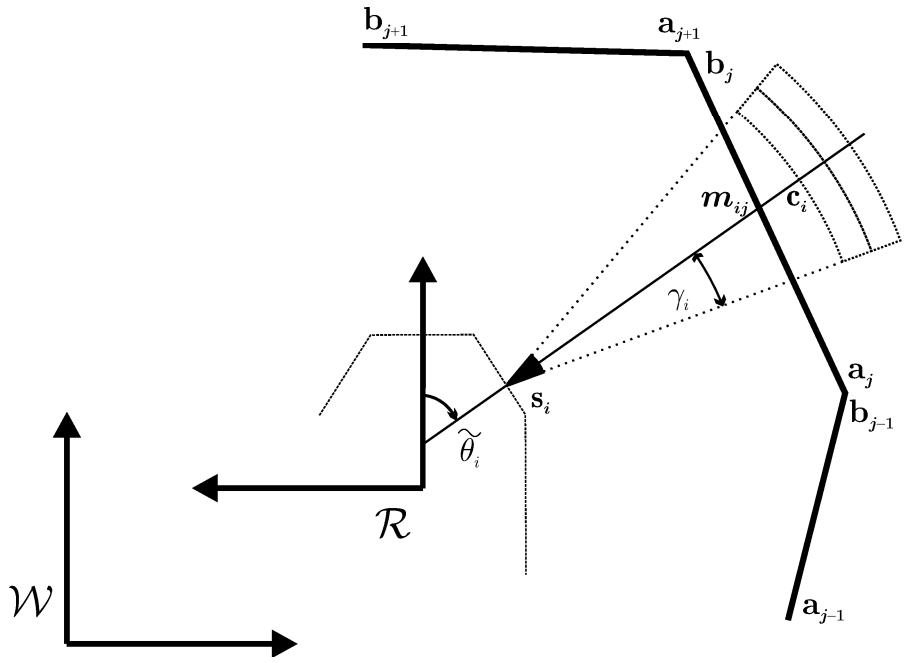

Figure 6. The test leg_in_i holds false.

Table 1. Operations between Boolean intervals

$\wedge$	$\mathbf{1}$	$\mathbf{0}$	$[\mathbf{0},\mathbf{1}]$
$\mathbf{1}$	1	0	$[0,1]$
$\mathbf{0}$	0	0	0
$[\mathbf{0},\mathbf{1}]$	$[0,1]$	0	$[0,1]$

$\vee$	$\mathbf{1}$	$\mathbf{0}$	$[\mathbf{0},\mathbf{1}]$
$\mathbf{1}$	1	1	1
$\mathbf{0}$	1	0	$[0,1]$
$[\mathbf{0},\mathbf{1}]$	1	$[0,1]$	$[0,1]$

The following Boolean function will be useful in Section 4.3 to define tests to deal with abnormal data resulting from sensor failures or erroneous maps. Let q and m be positive integers, with $q < m$. By definition, the *q-relaxed and* test

$$\bigoplus\nolimits_q(t_1,\ldots,t_m) = \bigoplus_{i=1}^{m}\nolimits_q(t_i) \tag{14}$$

holds true if and only if at least $m-q$ of the Booleans t_i $(i=1,\ldots,m)$ are true. When $q=0$, $\bigoplus_q$ is equivalent to the operator $\wedge$. When $q=m-1$, $\bigoplus_q$ becomes equivalent to the operator $\vee$. Let s be the sum, *in the usual real sense*, of the values of the t_i's. To evaluate $\bigoplus_q$, it suffices to check whether $s \geq m-q$.

Let $\mathbb{IR}^n$ be the set of all n-dimensional real boxes (or vectors of real intervals). An *inclusion test* for the *test* $t : \mathbb{R}^n \to \{0,1\}$ is a function $t_{[]} : \mathbb{IR}^n \to \mathbb{IB}$ such that

for any $[\mathbf{x}]$, $t([\mathbf{x}]) \subset t_{[]}([\mathbf{x}])$, *i.e.*,

$$\begin{array}{l} t_{[]}([\mathbf{x}]) = 1 \;\Rightarrow\; \forall \mathbf{x} \in [\mathbf{x}], t(\mathbf{x}) = 1, \\ t_{[]}([\mathbf{x}]) = 0 \;\Rightarrow\; \forall \mathbf{x} \in [\mathbf{x}], t(\mathbf{x}) = 0. \end{array} \tag{15}$$

EXAMPLE: An inclusion test $t_{[]}$ for $t(\mathbf{x}) \Leftrightarrow \mathbf{x} \in \mathcal{Y}$, where $\mathcal{Y}$ is some predefined set, is

$$\left\{ \begin{array}{ll} t_{[]}([\mathbf{x}]) = 1 & \text{if } [\mathbf{x}] \subset \mathcal{Y}, \\ t_{[]}([\mathbf{x}]) = 0 & \text{if } [\mathbf{x}] \cap \mathcal{Y} = \emptyset, \\ t_{[]}([\mathbf{x}]) = [0,1] & \text{otherwise.} \end{array} \right. \tag{16}$$

□

EXAMPLE: To obtain an interval counterpart for $\bigoplus_q$, it suffices to evaluate the sum of the interval values of the $t_{i[]}$'s and to compare the result with $m - q$. For instance, $\bigoplus_{q[]}(1, [0,1], 0, 1)$ is equal to 0 if $q = 0$, to $[0,1]$ if $q = 1$ and to 1 if $q = 2$.

□

Let $t_{[]1}$ and $t_{[]2}$ be two inclusion tests associated with the same test t. $t_{[]1}$ will be said to be *more powerful* than $t_{[]2}$ if for any $[\mathbf{x}]$, $t_{[]1}([\mathbf{x}]) \subset t_{[]2}([\mathbf{x}])$. The intersection of two inclusion tests associated with the same point test is more powerful than any of them. The following theorem will be useful to define more powerful tests.

THEOREM 1 *Let $t_{[]}$ be an inclusion test for t and $u_{[]}$ be an inclusion test for u, such that if $t(\mathbf{x})$ holds true then $u(\mathbf{x})$ does. Then $t'_{[]} = \left([0,1] \wedge u_{[]}\right) \cap t_{[]}$ is an inclusion test for t, which is more powerful than $t_{[]}$.* ♢

Proof: If $u_{[]}([\mathbf{x}]) \in \{[0,1], 1\}$ then $[0,1] \wedge u_{[]}([\mathbf{x}]) = [0,1]$ and $t'_{[]}([\mathbf{x}]) = [0,1] \cap t_{[]}([\mathbf{x}]) = t_{[]}([\mathbf{x}])$. If $u_{[]}([\mathbf{x}]) = 0$, then (15) holds and $\forall \mathbf{x} \in [\mathbf{x}]$, $u(\mathbf{x}) = 0$. Therefore $\forall \mathbf{x} \in [\mathbf{x}]$, $t(\mathbf{x}) = 0$ (if there existed $\mathbf{x}_0 \in [\mathbf{x}]$ such that $t(\mathbf{x}_0) = 1$, then $u(\mathbf{x}_0)$ would be equal to 1). As $\forall \mathbf{x} \in [\mathbf{x}]$, $t(\mathbf{x}) = 0$, $t_{[]}([\mathbf{x}])$ is either 0 or $[0,1]$. Thus $t'_{[]}([\mathbf{x}]) = ([0,1] \wedge 0) \cap t_{[]}([\mathbf{x}]) = 0 \cap t_{[]}([\mathbf{x}]) = 0$, so $t'_{[]}([\mathbf{x}]) \subset t_{[]}([\mathbf{x}])$. Thus, $t'_{[]}([\mathbf{x}])$ is an inclusion test for t, and is more powerful than $t_{[]}([\mathbf{x}])$. ■

Consider a test t obtained by performing logical operations on the results of elementary tests. A possible way to obtain an inclusion test associated with t is to replace each operator by its interval counterpart and each elementary test by an associated inclusion test. The result will be called a *natural interval extension* of t.

4.2. Interval extensions for the localization tests

A natural interval extension of each elementary data-association test dat_i is built as in Example

$$\left\{ \begin{array}{ll} dat_{i[]}([\mathbf{p}]) = 1 & \text{if } r_{i[]}([\mathbf{p}]) \subset [d_i], \\ dat_{i[]}([\mathbf{p}]) = 0 & \text{if } r_{i[]}([\mathbf{p}]) \cap [d_i] = \emptyset, \\ dat_{i[]}([\mathbf{p}]) = [0,1] & \text{otherwise.} \end{array} \right. \tag{17}$$

This test is based on the evaluation of remoteness, which involves a number of conditional branchings, and it remains to be decided which branches should be executed. The function presented below and derived from Kearfott's Chi function [12] is a possible way of getting rid of the problem. If t is the Boolean result of a test and y and z are two real numbers, then

$$\chi(t,y,z) = \begin{cases} y & \text{if } t = 1, \\ z & \text{if } t = 0. \end{cases} \tag{18}$$

The interval counterpart of $\chi(t,y,z)$ is given by

$$\chi_{[]}([t],[y],[z]) = \begin{cases} [y] & \text{if } [t] = 1, \\ [z] & \text{if } [t] = 0, \\ \text{convex hull of } [y] \text{ and } [z] & \text{if } [t] = [0,1]. \end{cases} \tag{19}$$

The result of the evaluation of a test based on $\chi_{[]}$ is therefore always an interval. For more details on the interval extension of remoteness, see Appendix B.

A natural interval extension of *in_room* might be very pessimistic, because of the accumulation of uncertainty over a sum of angles. Instead, the following interval version of *in_room* will be used, where $[\mathbf{m}]$ is a box enclosing the set $\mathbf{m}([\mathbf{p}])$ for a given interval configuration $[\mathbf{p}]$ and $\mathbf{c}_{[\mathbf{m}]}$ is the center of $[\mathbf{m}]$.

Interval test $in\_room_{[]}([\mathbf{m}])$**:**

$$\begin{cases} in\_room_{[]}([\mathbf{m}]) = 1 & \begin{cases} \text{if } [\mathbf{a}_j, \mathbf{b}_j] \cap [\mathbf{m}] = \emptyset, \text{ for } j = 1, ..., n_{\mathrm{w}}, \\ \text{and } in\_room\left(\mathbf{c}_{[\mathbf{m}]}\right) = 1, \end{cases} \\ in\_room_{[]}([\mathbf{m}]) = 0 & \begin{cases} \text{if } [\mathbf{a}_j, \mathbf{b}_j] \cap [\mathbf{m}] = \emptyset, \text{ for } j = 1, ..., n_{\mathrm{w}}, \\ \text{and } in\_room\left(\mathbf{c}_{[\mathbf{m}]}\right) = 0, \end{cases} \\ in\_room_{[]}([\mathbf{m}]) = [0,1] & \text{otherwise.} \end{cases} \tag{20}$$

If $[\mathbf{m}]$ does not intersect any segment of the map, it is either in $\mathcal{P}_{\mathrm{int}}$ or in $\mathcal{P}_{\mathrm{ext}}$. To decide which of them $[\mathbf{m}]$ is included in, it suffices to check one point (here $\mathbf{c}_{[\mathbf{m}]}$). As in Section 3, when $[\mathbf{m}]$ is the projection of $[\mathbf{p}]$ onto the $x \times y$ space, $in\_room_{[]}([\mathbf{m}])$ is written as $in\_room_{[]}([\mathbf{p}])$.

The natural interval extension of leg_in_i is obtained by substituting $in\_room_{[]}$ for *in_room*.

4.3. Combining localization tests

The three elementary tests defined in Section 3 should now be combined into a global test $t(\mathbf{p})$. In the ideal case where the map is correct and no error bound is violated, this global test can be written as $t_{\mathrm{ideal}}(\mathbf{p}) = in\_room(\mathbf{p}) \wedge (\bigwedge_{i=1}^{n_{\mathrm{w}}} dat_i(\mathbf{p}))$. A necessary condition for $dat_i(\mathbf{p})$ to hold true is that $leg\_in_i(\mathbf{p})$ does. As this condition is not sufficient, leg_in_i can only be used in conjunction with dat_i in

order to facilitate elimination of inconsistent configurations in an interval context. The resulting interval test

$$t_{\text{ideal}[]}([\mathbf{p}]) = in\_room_{[]}([\mathbf{p}]) \wedge \left(\bigwedge_{i=1}^{n_{\text{w}}} \left(\left(leg\_in_{i[]}([\mathbf{p}]) \wedge [0,1] \right) \cap dat_{i[]}([\mathbf{p}]) \right) \right) \tag{21}$$

is more powerful than the natural interval extension of t_{ideal}, according to Theorem 1.

Remark. Elementary tests are performed from the left to the right, thus starting by the simplest methods available to eliminate a given configuration box. For the actual implementation, advantage is also taken of the fact that $leg\_in_{i[]}([\mathbf{p}])$ evaluates faster than $dat_{i[]}([\mathbf{p}])$, so all $leg\_in_{i[]}([\mathbf{p}])$ are evaluated before all $dat_{i[]}([\mathbf{p}])$.♢

Assume now that the part of the map involved in the definition of $\mathcal{P}_{\text{int}}$ is still correct but that outliers are present. Outliers are data points for which the hypotheses made on the bounds of the measurement errors are violated. In the context of robot localization, they are almost unavoidable. They may correspond, for instance, to multiple reflections, to the presence of persons or pieces of furniture, to sensor failures, etc. In the presence of such outliers, the set $\mathcal{S}$, as defined by t_{ideal}, may turn out to be empty. Using the *q-relaxed and* operator $\bigoplus_q$ introduced in Section 4.1, $t_{\text{ideal}[]}$ can be modified into

$$t_{\text{outliers}[]}([\mathbf{p}], q) = in\_room_{[]}([\mathbf{p}]) \wedge \left(\bigoplus_{i=1}^{n_{\text{w}}}{}_{q[]} \left(\left(leg\_in_{i[]}([\mathbf{p}]) \wedge [0,1] \right) \cap dat_{i[]}([\mathbf{p}]) \right) \right), \tag{22}$$

to tolerate up to q outliers. A possible policy is to start with $q = 0$, which corresponds to using t_{ideal}, and to increase q by one whenever the set of possible configurations is found to be empty. More details on this technique and the stopping criterion can be found in [10]. It corresponds to a guaranteed implementation of the Outlier Minimal Number Estimator (OMNE) ([16], [26] and [24]).

When no reliable $\mathcal{P}_{\text{int}}$ and $\mathcal{P}_{\text{ext}}$ are available, the test $in\_room_{[]}$ can be dropped from $t_{\text{ideal}[]}$ or $t_{\text{outliers}[]}$, depending on the reliability of the remaining data. Another option, not considered further in what follows, would be to give the same confidence to $in\_room_{[]}$ as to $dat_{i[]}$ and write

$$t_{\text{robust}[]}([\mathbf{p}], q) = \bigoplus_{i=0}^{n_{\text{w}}}{}_{q[]} \left(t_{i[]} \right), \tag{23}$$

where

$$\begin{aligned} t_{0[]} &= in\_room_{[]}([\mathbf{p}]), \\ t_{i[]} &= \left(leg\_in_{i[]}([\mathbf{p}]) \wedge [0,1] \right) \cap dat_{i[]}([\mathbf{p}]), \quad i = 1, \ldots, n_{\text{w}}. \end{aligned} \tag{24}$$

The purpose of the next section is to show how the set of all possible configurations can be characterized in a systematic way, once a suitable test $t(\mathbf{p})$ has been defined.

5. Recursive set inversion

The set $\mathcal{S} = \{\mathbf{p} \in [\mathbf{p}_0] \mid t(\mathbf{p}) = 1\}$ can also be written as $t^{-1}_{[\mathbf{p}_0]}(1)$. Characterizing $\mathcal{S}$ can therefore be viewed as a problem of *set inversion*, which can be solved in an approximated but guaranteed way using the SIVIA (Set Inversion Via Interval Analysis) algorithm [7], [8], [9]. Here, a recursive version of SIVIA will be used, which will make it possible to reduce the amount of testing required to enclose $\mathcal{S}$ in an outer subpaving $\widehat{\mathcal{S}}$ (*i.e.*, a union of boxes in configuration space), with the help of the notion of masked tests.

If $t_{[]}([\mathbf{p}_0]) = 1$, $[\mathbf{p}_0]$ is in the solution set $\mathcal{S}$ and is stored in $\widehat{\mathcal{S}}$. If $t_{[]}([\mathbf{p}_0]) = 0$, then $[\mathbf{p}_0]$ has a void intersection with $\mathcal{S}$ and is dropped altogether from further consideration. If $t_{[]}([\mathbf{p}_0]) = [0,1]$ and if the width of $[\mathbf{p}_0]$ is larger than some prespecified precision parameter ϵ, then $[\mathbf{p}_0]$ is bisected, leading to two child subboxes $L[\mathbf{p}]$ and $R[\mathbf{p}]$, and the test $t_{[]}(.)$ is recursively applied to each of them. Any box with width less than ϵ is considered small enough and incorporated in $\widehat{\mathcal{S}}$. This algorithm is finite. Its complexity has been studied in [9]. Upon completion, $\widehat{\mathcal{S}}$ is guaranteed to enclose $\mathcal{S}$.

5.1. Masked tests

If the value of an elementary inclusion test over a box $[\mathbf{p}]$ is either true or false, this result remains valid for any subbox of $[\mathbf{p}]$. It is thus no longer necessary to evaluate it again over its children. Only elementary tests with uncertain values have to be tested again. This is the principle of masked tests, which may be found for example in [25], but had not so far been implemented in SIVIA. Consider a test t obtained by Boolean combination of p elementary tests t_i. In the context of interval evaluation, interval extensions $t_{i[]}$ of these elementary tests are used. The associated mask for a given value of $[\mathbf{p}]$ is the function $\mu_{[]}(.) : \mathbb{IR}^3 \to \mathbb{IB}^p$ defined by

$$\mu_{[]}([\mathbf{p}]) = \left(t_{1[]}([\mathbf{p}]), \ldots, t_{p[]}([\mathbf{p}])\right)^{\mathrm{T}}. \tag{25}$$

Except when $[\mathbf{p}] = [\mathbf{p}_0]$, whenever t is to be evaluated over a box $[\mathbf{p}]$, the results of the elementary tests $t_{i[]}$ have already been evaluated over at least one parent box. Provided that these results have been stored in a mask $[\mu]$ attached to this parent box, it is no longer necessary to evaluate tests which have already received unambiguous answers. The resulting masked test, which is also in charge of updating $[\mu]$, will be denoted by $t_{[]}([\mathbf{p}], [\mu])$.

5.2. Masked Sivia

Masked tests are incorporated into SIVIA with the help of the recursive function CLASSIFY (see Table 2). This function makes it possible to store boxes in the outer

approximation $\widehat{\mathcal{S}}$ of the solution set, according to the results of the evaluation of the masked interval test $t_{[]}([\mathbf{p}],[\mu])$. In an effort to store boxes as large as possible in $\widehat{\mathcal{S}}$, whenever the two children of the same parent box turn out to have to been stored in $\widehat{\mathcal{S}}$, either because t holds true or because the value of t is indeterminate and they are small enough, these two children are merged into their parent box. The process is iterated as long as possible before storing the result into $\widehat{\mathcal{S}}$.

Table 2. Recursive function called by MaskSivia.

Classify
Inputs: $[\mathbf{p}],[\mu],\widehat{\mathcal{S}},\epsilon$;
Outputs: $[t],\widehat{\mathcal{S}}$;
$\lfloor t \rfloor = t_{[]}(\lfloor \mathbf{p} \rfloor, \lfloor \mu \rfloor)$;
if $(\lfloor t \rfloor \neq \lfloor 0,1 \rfloor)$ return$(\lfloor t \rfloor, \widehat{\mathcal{S}})$;
if $(w([\mathbf{p}]) < \epsilon)$ return$([0,1],\widehat{\mathcal{S}})$;
else bisect $[\mathbf{p}]$ into $L[\mathbf{p}]$ and $R[\mathbf{p}]$;
$\left([t_L],\widehat{\mathcal{S}}\right) = \text{Classify}\left(L[\mathbf{p}],[\mu],\widehat{\mathcal{S}},\epsilon\right)$;
$\left([t_R],\widehat{\mathcal{S}}\right) = \text{Classify}\left(R[\mathbf{p}],[\mu],\widehat{\mathcal{S}},\epsilon\right)$;
if $([t_L] \wedge [t_R] \neq 0)$ return$([t_L] \wedge [t_R],\widehat{\mathcal{S}})$;
if $([t_L] \neq 0)$ store $L[\mathbf{p}]$ into $\widehat{\mathcal{S}}$;
if $([t_R] \neq 0)$ store $R[\mathbf{p}]$ into $\widehat{\mathcal{S}}$;
return $\left(0,\widehat{\mathcal{S}}\right)$.

Classify is first called by MaskSivia described by Table 3. If the value $[t_0]$ returned by Classify to MaskSivia differs from 0, then the whole initial search box $[\mathbf{p}_0]$ must be included in $\widehat{\mathcal{S}}$. Else, the outer approximation $\widehat{\mathcal{S}}$ has been built recursively by Classify.

6. Test cases

Interval-based localization will now be illustrated on three test cases. Although based on simulations, these test cases are realistic and the characteristics of the robot (size, sensors location and performances) are those of the robot of Figure 1.

Table 3. Recursive MaskSivia.

MaskSivia
Inputs: $[\mathbf{p}_0],\epsilon$;
Outputs: $\widehat{\mathcal{S}}$;
Initialisation: $\widehat{\mathcal{S}} = \emptyset; [\mu_0] = [0,1]^p$;
$\left([t_0],\widehat{\mathcal{S}}\right) = \text{Classify}\left([\mathbf{p}_0],[\mu_0],\widehat{\mathcal{S}},\epsilon\right)$;
if $([t_0] \neq 0)$ $\widehat{\mathcal{S}} = \{\lfloor \mathbf{p}_0 \rfloor\}$;
return $\left(\widehat{\mathcal{S}}\right)$.

This robot is equipped with $n_s = 24$ ultrasonic sensors on its periphery. Each of them has been found to have an emission angle $\tilde{\gamma}$ of 0.2 *rad* and a distance relative inaccuracy α of 2% within its operating range.

In each of the test cases treated, the initial search domain in configuration space is $[-12m, 12m] \times [-12m, 12m] \times [0, 2\pi]$, and the precision parameter ϵ is taken equal to 0.04. All computations were performed on a P233MMX personal computer, using a C++ implementation of MaskSivia.

6.1. First test case

This test case illustrates the potential contribution of the various accelerating tools proposed in this paper under ideal conditions. The robot is located in the room described by Figure 7, and the map available to the robot matches this environment exactly. Figure 8 describes the emission diagram of the 24 sensors. It is such that an obstacle should lie at least in part between the two arcs associated with any given sensor.

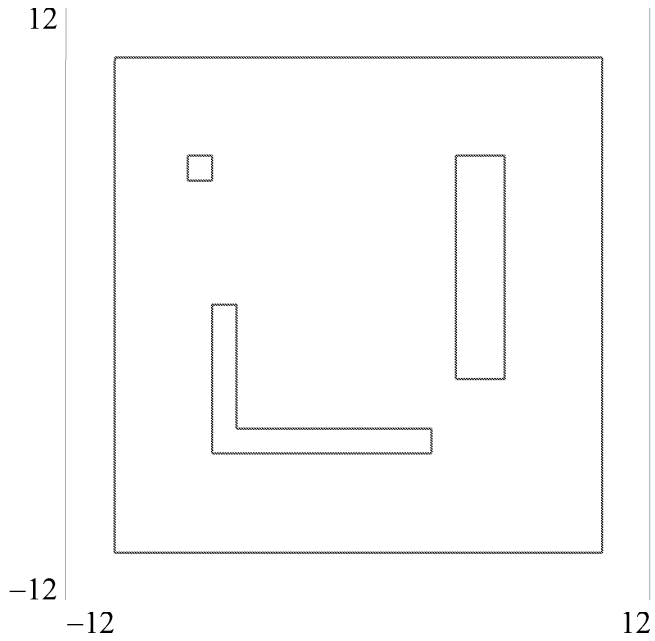

Figure 7. Map used by the robot for Test Cases 1 to 3. The projection of the initial search box onto the $x \times y$ space is the external square.

This diagram was obtained by computing the remoteness of each sensor from the map according to (10) for an actual configuration given by $(x_c, y_c, \theta) = \left(-2, 3, \frac{9\pi}{32}\right)$. Obviously, this actual configuration is not transmitted to the localization algorithm. Table 4 indicates computing time for various combinations of the tests proposed.

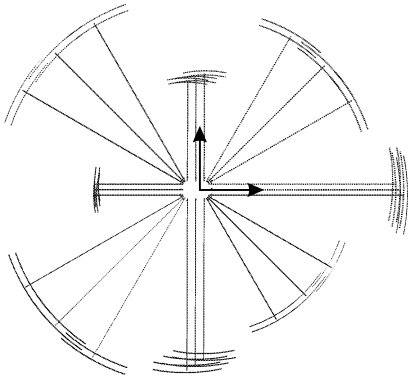

Figure 8. Emission diagram (Test Case 1).

Table 4. Computing times for Test Case 1.

Tests	**Mask**	**Time (sec.)**
$t_{\mathrm{dat}[]}$	no	97
$t_{\mathrm{dat}[]}$	yes	29
$t_{\mathrm{room}[]}$	no	91
$t_{\mathrm{room}[]}$	yes	27
$t_{\mathrm{leg}[]}$	no	48
$t_{\mathrm{leg}[]}$	yes	11
$t_{\mathrm{ideal}[]}$	no	49
$t_{\mathrm{ideal}[]}$	yes	11

The test $t_{\mathrm{dat}[]}$ only involves the elementary tests $dat_{i[]}$, $i = 1, \ldots, n_{\mathrm{s}}$. The test $t_{\mathrm{room}[]}$ combines $in\_room_{[]}$ and $t_{\mathrm{dat}[]}$. The test $t_{\mathrm{leg}[]}$ uses $leg\_in_{i[]}$, $i = 1, \ldots, n_{\mathrm{s}}$ to reinforce $t_{\mathrm{dat}[]}$. Finally, $t_{\mathrm{ideal}[]}$ combines all these tests as described by (21). In all cases, the resulting solution boxes turn out to be very similar, and Figure 9 presents those obtained with the complete algorithm. The union of these boxes is guaranteed to contain all configurations consistent with the map and measurements. The actual robot configuration is indicated in black.

On this example, the masked version of SIVIA using $t_{\mathrm{leg}[]}$ or $t_{\mathrm{ideal}[]}$ is about ten times quicker than a basic SIVIA using only $t_{\mathrm{dat}[]}$. The mask appears responsible for most of the improvement, followed by $leg\_in_{i[]}$ and $in\_room_{[]}$. When the mask and $leg\_in_{i[]}$ are implemented, $in\_room_{[]}$ leads to no improvement, but remember that $leg\_in_{i[]}$ is based on $in\_room_{[]}$.

The next two examples will illustrate more difficult but quite realistic situations.

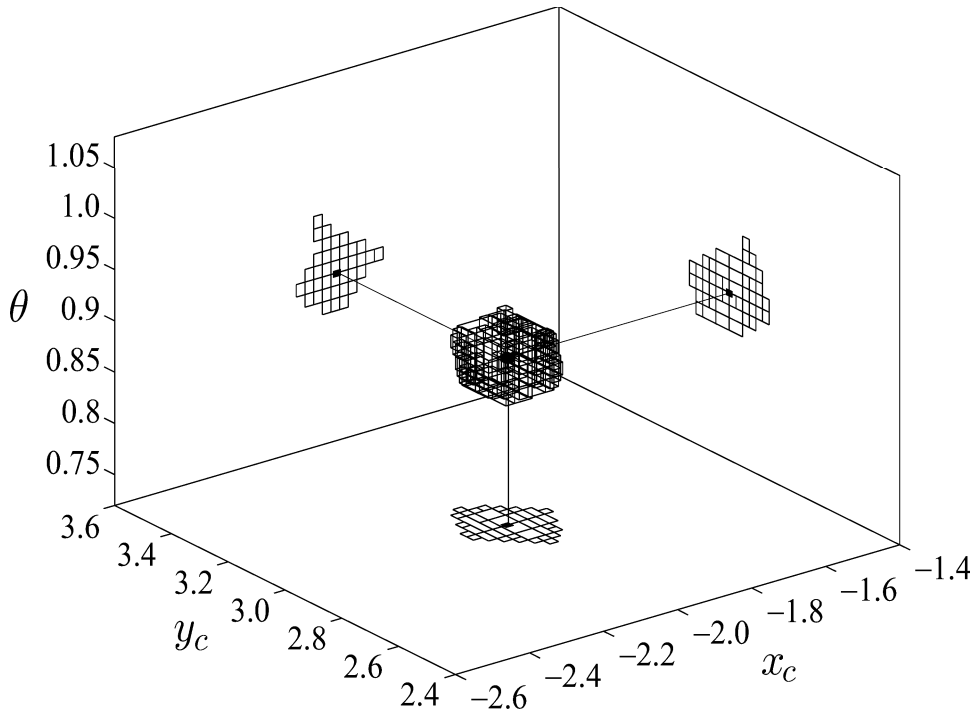

Figure 9. Outer approximation of the set of all possible configurations and its 2D projections (Test Case 1). The actual configuration is indicated in black.

6.2. Second test case

In this test case, the room and map remain identical to those of Test Case 1, but the actual (unknown) configuration is now $(x_c, y_c, \theta) = (1, -7.5, \pi)$, and the emission diagram is given by Figure 10. In *19* seconds, MASKSIVIA using $t_{\text{ideal}[]}$ finds the set

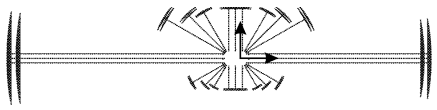

Figure 10. Emission diagram (Test Case 2).

of boxes described by Figure 11. This set consists of two disconnected subsets, one of which contains the actual configuration of the robot. Figure 12 illustrates the fact that, due to local symmetries, there are indeed two radically different types of possible configurations, each of which corresponds to a different association of segments of the map with distances measured by the sensors. Note that this data association is a by-product of the algorithm, and does not need to be performed by

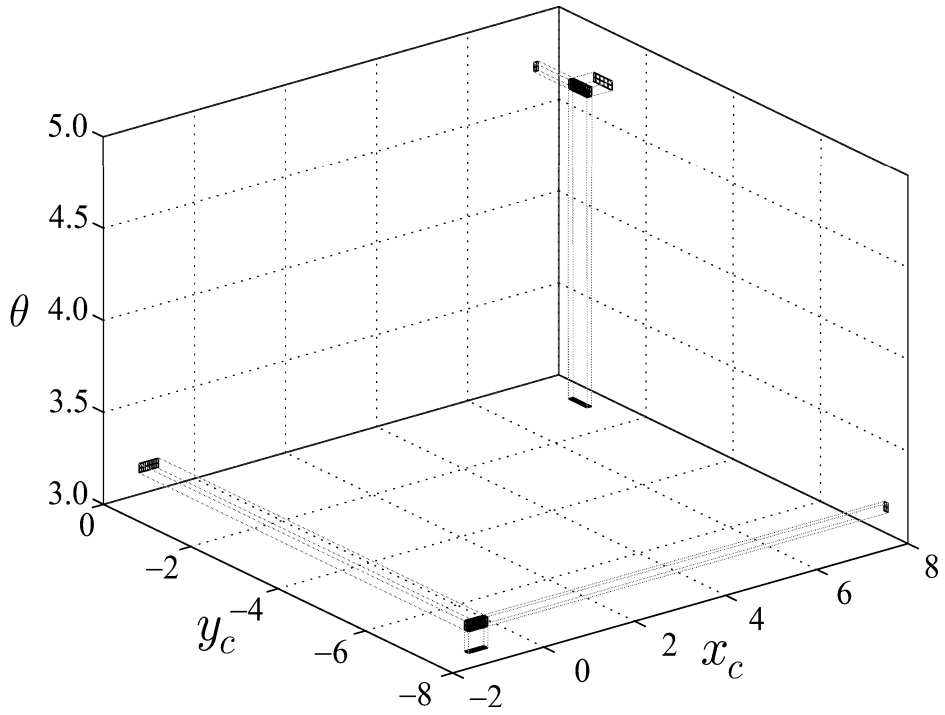

Figure 11. Outer approximation of the set of all possible configurations (Test Case 2) and its 2D projections.

a preprocessor as in the usual localization methods. Given that data association is one of the bottlenecks of automated localization, this is no minor advantage.

6.3. Third test case

The additional difficulties created by outliers and an outdated map will now be taken into account. The map provided to the robot is the same as in the previous test cases, but it is now partly incorrect. The actual environment is that of Figure 13.

The previous pillar has been moved, and a second one added. Moreover, two out of the 24 distances have been taken equal to twice their actual values. The actual (unknown) configuration is the same as in the first test case. Any of the modifications considered here (*i.e.*, the incorrect map or the outliers) is enough to make the set found by the original algorithm empty. Note that the map can no longer be assumed to be correct, so $in\_room_{[]}$ will not be employed. The value of q is increased until the set of boxes found using $t_{\text{outliers}[]}$ without $in\_room_{[]}$ becomes nonempty, which takes place when $q = 6$. The set of possible configurations thus found is slightly larger than that on Figure 9, but similar and will not be

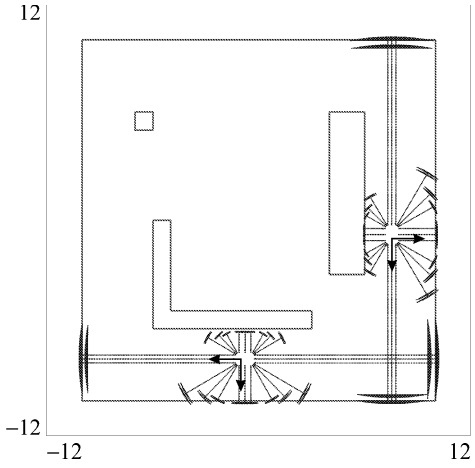

Figure 12. Two possible configurations (Test Case 2).

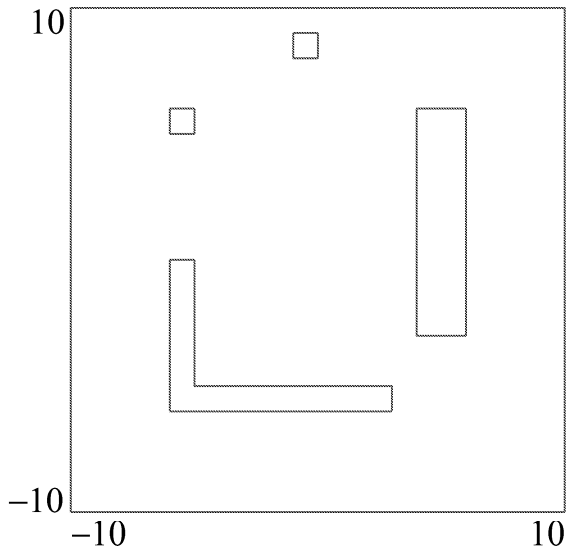

Figure 13. Room of Test Case 3.

repeated. It still contains the actual robot configuration. Figure 14 presents a typical configuration of this set, where the data that could not be associated are indicated by numbers. Emission cones labelled 1 and 6 are inconsistent with the map, because of the presence of obstacles that are closer to the sensors considered. Emission cones labelled 2 to 5 correspond to the two misspecified pillars. Table 5 indicates computing time and various properties of $\widehat{\mathcal{S}}$ as functions of q. Note that the set of

boxes obtained for a given value of q is only guaranteed to contain the actual robot configuration if there are no more than q actual outliers. One may protect oneself against a larger number of outliers by increasing q. The sets obtained here for $q > 6$ are quite close to that obtained for $q = 6$, the actual number of outliers. The result of this robust localization procedure thus turns out to be rather insensitive to the choice made for q.

Table 5. Characteristics of $\widehat{\mathcal{S}}$ and cumulated computing time as functions of q for Test Case 3.

q	Set volume	Bounding box (outward rounded)	Time
0	0	$\emptyset$	7 s.
1	0	$\emptyset$	21 s.
2	0	$\emptyset$	41 s.
3	0	$\emptyset$	71 s.
4	0	$\emptyset$	113 s.
5	0	$\emptyset$	166 s.
6	2.68×10^{-3}	$[-2.14, -1.87][2.85, 3.15][0.83, 0.95]$	249 s.
7	3.09×10^{-3}	$[-2.14, -1.87][2.85, 3.15][0.83, 0.95]$	366 s.
8	4.25×10^{-3}	$[-2.16, -1.82][2.83, 3.17][0.83, 0.95]$	519 s.
9	5.88×10^{-3}	$[-2.18, -1.82][2.83, 3.19][0.83, 0.96]$	776 s.
10	8.05×10^{-3}	$[-2.21, -1.80][2.81, 3.19][0.82, 0.97]$	1126 s.

Computing time is seen to increase with q, because it becomes increasingly difficult to eliminate a box.

Contrary to what would be the case with traditional methods involving a phase of data association, no combinatorics is involved in deciding which q measurements have to be considered as outliers, and this is again a tremendous simplification.

7. Conclusions and perspectives

Autonomous robot localization is particularly well amenable to solution via interval analysis, because the number of parameters to be estimated is small. In this context, the method advocated here has definite advantages over conventional numerical methods. It is not necessary to enumerate all possible associations between sensor data and landmarks, nor is it necessary to consider all possible choices of q outliers among n_s data points. As a result, combinatorial explosion is avoided. The results obtained are global, and no configuration compatible with prior information and measurements can be missed. These results are extremely robust, and the estimator used can even handle a majority of outliers. Provided that the number of actual outliers is less than or equal to the value chosen for q, the results are still guaranteed. The present computing times seem already acceptable for a *static* localization with such remarkable properties.

The method is flexible, and additional information on the physics of the problem could readily be incorporated. One could, for instance, take into account the fact that the operational range of ultrasonic sensors is limited, or that the incidence angle should be small enough for the reflected or refracted wave to be picked up

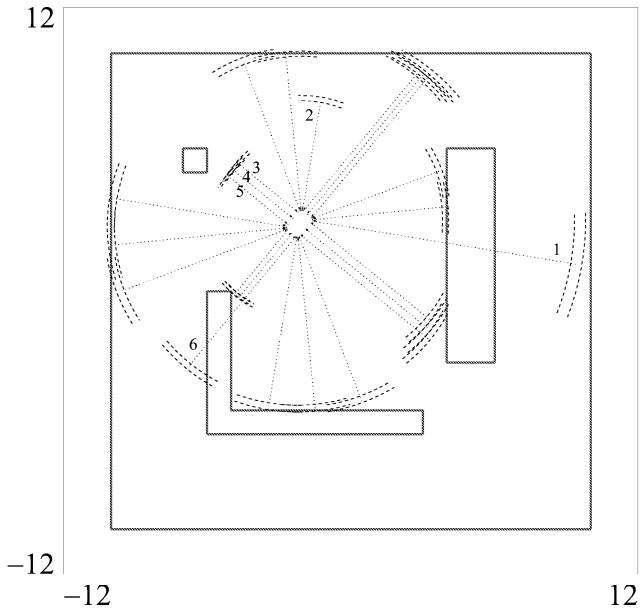

Figure 14. Possible configuration for Test Case 3.

by the sensor. Other types of sensors such as rotating laser range finders (see, *e.g.*, [1], [3]), as well as multi-sensor data fusion [11] should form the subject of future studies in the context of interval methods such as those advocated in this paper.

In this paper, localization was static; a natural extension of the present work is to consider the tracking of the set of possible configurations of a moving robot. This can be done [15], [13], using a recently developed bounded-error state estimation techniques for nonlinear models [14]. The fact that the initial search domain in configuration space is much smaller at any given time instant reduces drastically the computational effort and makes it compatible with real time.

The methodology described obviously applies to many other fields, where feasibility is also defined in terms of possibly nonlinear inequalities. The case where some of these inequalities may not be meaningful could be handled directly by treating them as outliers.

8. Notation

Vectors are in bold with an arrow on top: $\overrightarrow{\mathbf{u}}$. Points are in bold: $\mathbf{a}, \mathbf{b}, \mathbf{c}$. Coordinates for two-dimensional vectors $\overrightarrow{\mathbf{u}}$ and points $\mathbf{a}$ are denoted by x_u, y_u and x_a, y_a.

$\langle \overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}} \rangle$	:	scalar product of $\overrightarrow{\mathbf{u}}$ and $\overrightarrow{\mathbf{v}}$,
$(\mathbf{a}, \mathbf{b})$	:	line supported by $\mathbf{a}$ and $\mathbf{b}$,
$[\mathbf{a}, \mathbf{b}]$	:	line segment between $\mathbf{a}$ and $\mathbf{b}$,
$\overrightarrow{\mathbf{ab}}$	:	vector with extreme points $\mathbf{a}$ and $\mathbf{b}$,
$(\mathbf{s}, \overrightarrow{\mathbf{u}})$	:	line supported by $\mathbf{s}$ with direction vector $\overrightarrow{\mathbf{u}}$,
$\tilde{\gamma}$	:	half aperture of the emission cone,
i	:	index for sensors,
j	:	index for segments,
$d(\mathbf{s}, (\mathbf{a}, \mathbf{b}))$	:	distance from $\mathbf{s}$ to $(\mathbf{a}, \mathbf{b})$,
$d_{\overrightarrow{\mathbf{u}}}(\mathbf{s}, (\mathbf{ab}))$	:	distance from $\mathbf{s}$ to $(\mathbf{a}, \mathbf{b})$ along $\overrightarrow{\mathbf{u}}$,
n_{s}	:	number of sensors,
n_{w}	:	number of segments,
$\mathbf{p} = (x_c, y_c, \theta)^{\mathrm{T}}$	:	robot configuration,
$\mathcal{S}$	:	set of all feasible robot configurations,
$\mathbb{IR}$	:	set of real intervals,
$\mathbb{IB}$	:	set of Boolean intervals,
$\wedge$	:	logical AND,
$\vee$	:	logical OR,
$w([\mathbf{p}])$	:	width of $[\mathbf{p}]$.

Appendix A
Real evaluation of remoteness

Table A.1 presents the implementation of the real evaluation of remoteness, based on Section 3.1.

The distance $d(\mathbf{s}, (\mathbf{a}, \mathbf{b}))$ from $\mathbf{s}$ to the line $(\mathbf{a}, \mathbf{b})$ (Figure A.1) is given by

$$d(\mathbf{s}, (\mathbf{a}, \mathbf{b})) = \left\| \overrightarrow{\mathbf{ah}} \right\| = \frac{\det\left(\overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{as}}\right)}{\left\| \overrightarrow{\mathbf{ab}} \right\|}, \tag{A.1}$$

and the distance $d_{\overrightarrow{\mathbf{u}}}(\mathbf{s}, (\mathbf{a}, \mathbf{b}))$ from $\mathbf{s}$ to the line $(\mathbf{a}, \mathbf{b})$ along the unit vector $\overrightarrow{\mathbf{u}}$ by

$$d_{\overrightarrow{\mathbf{u}}}(\mathbf{s}, (\mathbf{a}, \mathbf{b})) = \|\overrightarrow{\mathbf{am}}\| = \frac{\left\| \overrightarrow{\mathbf{ah}} \right\|}{|\sin\theta|} = \frac{\det\left(\overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{as}}\right)}{\left\| \overrightarrow{\mathbf{ab}} \right\| |\sin\theta|} = \frac{\left|\det\left(\overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{as}}\right)\right|}{\left|\det\left(\overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{u}}\right)\right|}. \tag{A.2}$$

Appendix B
Interval evaluation of remoteness

The interval counterpart of Table A.1 is given by Table B.1.

In this table, $\overrightarrow{[\mathbf{s}]\,\mathbf{a}}$ stands for the set of all vectors with origin in the box $[\mathbf{s}]$ and extremity at $\mathbf{a}$. The box $[\mathbf{s}]$, guaranteed to contain the location of the sensor $\mathbf{s}$ for

Table A.1. Evaluation of remoteness.

$$r\left(\mathbf{s}, \overrightarrow{\mathbf{u}_1}, \overrightarrow{\mathbf{u}_2}, \mathbf{a}, \mathbf{b}\right)$$

if $\left(\det\left(\overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{as}}\right) \geq 0\right)$
  return $(+\infty)$;

if $\left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{sa}} \right\rangle \leq 0\right) \wedge \left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{sb}} \right\rangle \geq 0\right) \wedge \left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{u}_1} \right\rangle \leq 0\right) \wedge \left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{u}_2} \right\rangle \geq 0\right)$
  then $r_{\mathbf{h}} = d\left(\mathbf{s}, (\mathbf{a}, \mathbf{b})\right)$, else $r_{\mathbf{h}} = +\infty$;

if $\left(\det\left(\overrightarrow{\mathbf{u}_1}, \overrightarrow{\mathbf{sa}}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{\mathbf{u}_2}, \overrightarrow{\mathbf{sa}}\right) \leq 0\right)$
  then $r_{\mathbf{a}} = \|\mathbf{sa}\|$, else $r_{\mathbf{a}} = +\infty$;

if $\left(\det\left(\overrightarrow{\mathbf{u}_1}, \overrightarrow{\mathbf{sb}}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{\mathbf{u}_2}, \overrightarrow{\mathbf{sb}}\right) \leq 0\right)$
  then $r_{\mathbf{b}} = \left\|\overrightarrow{\mathbf{sb}}\right\|$, else $r_{\mathbf{b}} = +\infty$;

for $i = 1$ to 2
  if $\left(\det\left(\overrightarrow{\mathbf{sa}}, \overrightarrow{\mathbf{u}_i}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{\mathbf{sb}}, \overrightarrow{\mathbf{u}_i}\right) \leq 0\right)$
  then $r_{\mathbf{h}_i} = d_{\overrightarrow{\mathbf{u}_i}}\left(\mathbf{s}, (\mathbf{a}, \mathbf{b})\right)$, else $r_{\mathbf{h}_i} = +\infty$;

return $\left(\min\left(r_{\mathbf{h}}, r_{\mathbf{a}}, r_{\mathbf{b}}, r_{\mathbf{h}_1}, r_{\mathbf{h}_2}\right)\right)$.

Table B.1. Inclusion function for remoteness.

$$r_{[]}\left([\mathbf{s}], \overrightarrow{[\mathbf{u}_1]}, \overrightarrow{[\mathbf{u}_2]}, \mathbf{a}, \mathbf{b}\right)$$

$[t_1] = \det\left(\overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{a}\,[\mathbf{s}]}\right)$;
`if` $\left([t_1] \geq 0\right)$
  `return` $(+\infty)$;

$[t_{\mathbf{h}}] = \left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{[\mathbf{s}]\,\mathbf{a}} \right\rangle \leq 0\right) \wedge \left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{[\mathbf{s}]\,\mathbf{b}} \right\rangle \geq 0\right) \wedge \left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{[\mathbf{u}_1]} \right\rangle \leq 0\right) \wedge \left(\left\langle \overrightarrow{\mathbf{ab}}, \overrightarrow{[\mathbf{u}_2]} \right\rangle \geq 0\right)$;
$[r_{\mathbf{h}}] = \chi\left([t_{\mathbf{h}}], d_{[]}\left([\mathbf{s}], (\mathbf{a}, \mathbf{b})\right), +\infty\right)$;

$[t_{\mathbf{a}}] = \left(\det\left(\overrightarrow{[\mathbf{u}_1]}, \overrightarrow{[\mathbf{s}]\,\mathbf{a}}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{[\mathbf{u}_2]}, \overrightarrow{[\mathbf{s}]\,\mathbf{a}}\right) \leq 0\right)$;
$[r_{\mathbf{a}}] = \chi\left([t_{\mathbf{a}}], \left\|\overrightarrow{[\mathbf{s}]\,\mathbf{a}}\right\|, +\infty\right)$;

$[t_{\mathbf{b}}] = \left(\det\left(\overrightarrow{[\mathbf{u}_1]}, \overrightarrow{[\mathbf{s}]\,\mathbf{b}}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{[\mathbf{u}_2]}, \overrightarrow{[\mathbf{s}]\,\mathbf{b}}\right) \leq 0\right)$;
$[r_{\mathbf{b}}] = \chi\left([t_{\mathbf{b}}], \left\|\overrightarrow{[\mathbf{s}]\,\mathbf{b}}\right\|, +\infty\right)$;

`for` $i = 1$ `to` 2
  $[t_{\mathbf{h_i}}] = \left(\det\left(\overrightarrow{[\mathbf{s}]\,\mathbf{a}}, \overrightarrow{[\mathbf{u}_i]}\right) \geq 0\right) \wedge \left(\det\left(\overrightarrow{[\mathbf{s}]\,\mathbf{b}}, \overrightarrow{[\mathbf{u}_i]}\right) \leq 0\right)$;
  $[r_{\mathbf{h}_i}] = \chi\left([t_{\mathbf{h_i}}], d_{\overrightarrow{[\mathbf{u}_i]}}\left([\mathbf{s}], (\mathbf{a}, \mathbf{b})\right), +\infty\right)$;

`return` $\left(\min\left([r_{\mathbf{h}}], [r_{\mathbf{a}}], [r_{\mathbf{b}}], \lfloor r_{\mathbf{h}_1} \rfloor, \lfloor r_{\mathbf{h}_2} \rfloor\right)\right)$;

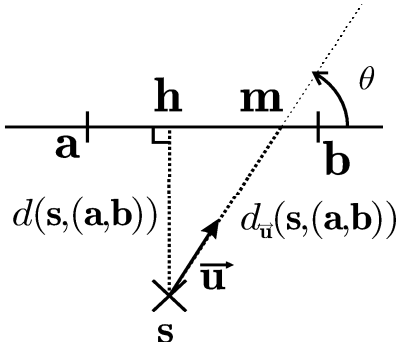

Figure A.1. Distances from the point $\mathbf{s}$ to the line $(\mathbf{a}, \mathbf{b})$.

any configuration in $[\mathbf{p}] = ([x_c], [y_c], [\theta])^{\mathrm{T}}$, is evaluated by replacing all occurrences of the real variables in (1) by their interval counterparts. Similarly, the characteristics of the cone (3) are evaluated as $[\mathcal{C}] = \mathcal{C}\left([\mathbf{s}], [\overrightarrow{\mathbf{u}_1}], [\overrightarrow{\mathbf{u}_2}]\right)$. Finally, the minimum of two intervals is defined as follows

$$\min([a], [b]) = \left[\min(\underline{a}, \underline{b}), \min(\overline{a}, \overline{b})\right],$$

the extension to more intervals being straightforward.

References

1. J. Borenstein, H. Everett, and L. Feng. *Navigating Mobile Robots.* A. K. Peters Ltd., Wellesley, Massachusetts, 1996.
2. J. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 674–680, Scottsdale, Arizona, 1989.
3. J. L. Crowley, F. Wallner, and B. Schiele. Position estimation using principal components of range data. In *IEEE Int. Conf. on Robotics and Automation*, pages 3121–3128, Leuven, 1998.
4. M. Drumheller. Mobile robot localization using sonar. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(2):325–332, 1987.
5. E. Halbwachs and D. Meizel. Multiple hypothesis management for mobile vehicule localization. In *CD Rom of the European Control Conference*, Louvain, 1997.
6. A. A. Holenstein, M. A. Müller, and E. Badreddin. Mobile robot localization in a structured environment cluttered with obstacles. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2576–2581, Nice, 1992.
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. Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis. *Math. and Comput. in Simulation*, 35:1923–1937, 1993.
9. L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.
10. L. Jaulin, E. Walter, and O. Didrit. Guaranteed robust nonlinear parameter bounding. In *Proc. IMACS—IEEE–SMC CESA'96 Symposium on Modelling and Simulation*, volume 2, pages 1156–1161, Lille, 1996.

11. M. Kam, X. Zhu, and P. Kalata. Sensor fusion for mobile robot navigation. *Proceeedings of the IEEE*, 85(1):108–119, 1997.
12. R. B. Kearfott. Interval extensions of non-smooth functions for global optimization and nonlinear system solvers. *Computing*, 57(2):149–162, 1996.
13. M. Kieffer. *Estimation ensembliste par analyse par intervalles, application à la localisation d'un véhicule.* PhD dissertation, Université Paris-Sud, Orsay, 1999.
14. M. Kieffer, L. Jaulin, and E. Walter. Guaranteed recursive nonlinear state estimation using interval analysis. In *Proc. 37th IEEE Conference on Decision and Control*, pages 3966–3971, Tampa, Florida, 16-18 December 1998.
15. M. Kieffer, L. Jaulin, E. Walter, and D. Meizel. Guaranteed mobile robot tracking using interval analysis. In *Proc. MISC'99 Workshop on Application of Interval Analysis to System and Control*, pages 347–359, Girone, 24-26 February 1999.
16. H. Lahanier, E. Walter, and R. Gomeni. OMNE: a new robust membership-set estimator for the parameters of nonlinear models. *J. of Pharmacokinetics and Biopharmaceutics*, 15:203–219, 1987.
17. 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.
18. J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation.* Kluwer Academic Publishers, Boston, 1992.
19. O. Lévêque, L. Jaulin, D. Meizel, and E. Walter. Vehicule localization from inaccurate telemetric data: a set inversion approach. In *Proc. 5th IFAC Symposium on Robot Control SY.RO.CO.'97*, volume 1, pages 179–186, Nantes, France, 1997.
20. M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter (Eds). *Bounding Approaches to System Identification.* Plenum Press, New York, 1996.
21. J. Neira, J. Horn, J. D. Tardoz, and G. Schmidt. Multisensor mobile robot localization. In *Proc. IEEE International Conference on Robotics and Automation*, pages 673–679, Mineapolis, USA, 1996.
22. J. P. Norton (Ed.). Special issue on bounded-error estimation: Issue 1. *Int. J. of Adaptive Control and Signal Processing*, 8(1):1–118, 1994.
23. J. P. Norton (Ed.). Special issue on bounded-error estimation: Issue 2. *Int. J. of Adaptive Control and Signal Processing*, 9(1):1–132, 1995.
24. L. Pronzato and E. Walter. Robustness to outliers of bounded-error estimators and consequences on experiment design. In M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, editors, *Bounding Approaches to System Identification*, pages 199–212, New York, 1996. Plenum Press.
25. H. Ratschek and J. Rokne. *New Computer Methods for Global Optimization.* Wiley, New York, 1988.
26. E. Walter and H. Piet-Lahanier. Estimation of the parameter uncertainty resulting from bounded-error data. *Mathematical Biosciences*, 92:55–74, 1988.
27. E. Walter (Ed.). Special issue on parameter identifications with error bounds. *Mathematics and Computers in Simulation*, 32(5&6):447–607, 1990.