



Guaranteed Robust Nonlinear Estimation, with Application to Robot Localization

L. Jaulin¹, M. Kieffer², E. Walter² and D. Meizel³

¹Laboratoire d'Ingénierie des Systèmes Automatisés, ISTIA, 62 avenue Notre Dame du Lac, 49000 Angers, France; luc.jaulin@univ-angers.fr

²Laboratoire des Signaux et Systèmes, CNRS-Supélec, Université Paris-Sud, Plateau de Moulon, 91192 Gif-sur-Yvette, France; {kieffer, walter}@lss.supelec.fr

³HEUDIASYC, UMR CNRS 6599, Université de Technologie de Compiègne B.P. 20529 Compiègne Cédex, France; dominique.meizel@utc.fr

Keywords: bounded-error estimation, interval analysis, nonlinear estimation, outliers, robot localization, robust estimation, set estimation, set inversion.

Suggested running title: Guaranteed robust nonlinear estimation.

Abstract: When reliable prior bounds on the acceptable errors between the data and corresponding model outputs are available, bounded-error estimation techniques make it possible to characterize the set of all acceptable parameter vectors in a guaranteed way, even when the model is nonlinear and the number of data points small. However, when the data may contain outliers, *i.e.*, data points for which these bounds should be violated, this set may turn out to be empty, or at least unrealistically small. The outlier minimal number estimator (OMNE) has been designed to deal with such a situation, by minimizing the number of data points considered as outliers. OMNE has been shown in previous papers to be remarkably robust, even to a majority of outliers. Up to now, it was implemented by random scanning, so its results could not be guaranteed. In this paper, a new algorithm based on set inversion via interval analysis provides a guaranteed OMNE, which is applied to the initial localization of an actual robot in a partially known 2D environment. The difficult problems of associating range data to landmarks of the environment and of detecting potential outliers are solved as by-products of the procedure.

I. INTRODUCTION

This paper deals with estimating the unknown parameters of a model from experimental data. Let $\vec{y} \in \mathbb{R}^{n_y}$ be the vector of all these data. It may consist of system outputs for various values of some independent variables such as time, and multivariable dynamical systems can readily be considered. A set of models is assumed to be available, parameterized by a vector $\vec{p} \in \mathbb{R}^{n_p}$ to be estimated. The simulation of any model $M(\vec{p})$ of this

set generates a vector of model outputs $\vec{f}(\vec{p})$ to be compared with the data vector \vec{y} . In the context of bounded-error estimation, see, *e.g.*, [27]; [3], [7]; [23]; [24]; [20] and the references therein, it is usually assumed that $\vec{f}(\vec{p})$ must belong to some given box (or interval vector) $\vec{Y} \subset \mathbb{R}^{n_y}$ for \vec{p} to be considered as consistent with the data. This box contains \vec{y} and characterizes the set of all output errors that are deemed acceptable, given the approximate nature of the model structure and the imprecision of the measurements. Once the model structure $M(\cdot)$ and the box \vec{Y} have been chosen, the problem to be solved is characterizing the set \mathcal{S} of all values of \vec{p} consistent with the data, *i.e.*,

$$\mathcal{S} = \{\vec{p} \in \mathbb{R}^{n_p} \mid \vec{f}(\vec{p}) \in \vec{Y}\} = \vec{f}^{-1}(\vec{Y}). \quad (1)$$

This can be interpreted in terms of set inversion, and methods based on interval analysis make it possible to enclose \mathcal{S} in a union of boxes with an arbitrary precision [11]. These methods are not, however, robust to a misspecification of the feasible box \vec{Y} . For instance, if the width of the interval Y_i results from an optimistic choice of error bounds or if the measurement y_i is provided by a faulty sensor, \mathcal{S} can be greatly affected and might even become empty. This is why we assume here that some data points may be outliers, *i.e.*, may correspond to errors much larger than originally thought. The corresponding model outputs should then be allowed to violate the error bounds. The set estimator OMNE (for *Outlier Minimal Number Estimator*) has been designed to deal with this situation. It was introduced in [16] and [28] in the context of mathematical modelling in pharmacokinetics and biology. It was shown in these papers that the data could incorporate a very large percentage of outliers and still lead to meaningful parameter estimates. This robustness issue was considered theoretically, in the context of linear estimation, in [25], where it is shown that, under suitable experimental conditions, OMNE achieves a breakdown point of almost 50%, the largest achievable performance. Loosely speaking, the breakdown point of an estimator is the smallest percentage of outliers that needs to be introduced to make the norm of the estimation error tend to infinity, see [26] for more details. OMNE aims at characterizing the *minimum outlier feasible set* \mathcal{S}^* , defined as

$$\mathcal{S}^* = \arg \min_{\vec{p} \in \mathbb{R}^{n_p}} j(\vec{p}), \quad (2)$$

where the value of the cost function $j(\vec{p})$ is the number of model outputs $f_i(\vec{p})$ such that $f_i(\vec{p}) \notin Y_i$, *i.e.*, that do not fall within the feasible ranges defined by the error bounds. The minimal value achieved by this cost function will be denoted by j^* ; it is the minimum number of data points that have to be considered as outliers. This cost function is not continuous, and its gradient is zero wherever j is differentiable. Generally, \mathcal{S}^* is not a singleton and has a nonzero volume. It may be nonconvex and even disconnected. When there are no outliers, \mathcal{S}^* is identical to \mathcal{S} . The algorithm used so far by OMNE was based on random scanning of parameter space, so no guarantee could be provided about

the results. In Section III., a new algorithm is presented to build a set of boxes in the parameter space, the union of which is guaranteed to contain \mathcal{S}^* . This algorithm involves solving a finite sequence of set-inversion problems by calling the algorithm SIVIA (Set Inversion Via Interval Analysis) [11], with suitably modified tests. Section IV. presents some new results on connexity that may help to find which data should be considered as outliers and to understand why disconnected sets are so often involved in the context of bounded-error estimation with outliers. In Section V., the approach is illustrated on a problem of robot localization. Basic notions of interval analysis, an essential ingredient of the procedure to be presented, are briefly recalled in the following section, where the new notions of inclusion and separation degrees are introduced.

II. INTERVAL ANALYSIS

A *box*, or *interval vector* \vec{X} of \mathbb{R}^n is a vector with interval components:

$$\vec{X} = [x_1^-, x_1^+] \times \cdots \times [x_n^-, x_n^+] = X_1 \times \cdots \times X_n = [\vec{x}^-, \vec{x}^+], \quad (3)$$

where $\vec{x}^- = (x_1^-, \dots, x_n^-)^T$ and $\vec{x}^+ = (x_1^+, \dots, x_n^+)^T$. Boxes and intervals will be denoted by capital letters. The *width* $w(\vec{X})$ of the box \vec{X} is the length of its largest side(s). The set of all boxes of \mathbb{R}^n will be denoted by \mathbb{IR}^n . To *bisect* \vec{X} means to split it into two boxes along a symmetry plane normal to a side of maximum length. Let $\vec{f}: \mathbb{R}^n \rightarrow \mathbb{R}^p$ be a vector function, the set-valued function $\vec{F}: \mathbb{IR}^n \rightarrow \mathbb{IR}^p$ is an *inclusion function* of \vec{f} if

$$\vec{f}(\vec{X}) \subset \vec{F}(\vec{X}) \quad (4)$$

for any $\vec{X} \in \mathbb{IR}^n$. This inclusion function is *convergent* if, for any sequence of boxes \vec{X} of \mathbb{IR}^n ,

$$w(\vec{X}) \rightarrow 0 \Rightarrow w(\vec{F}(\vec{X})) \rightarrow 0. \quad (5)$$

Various methods exist for computing convergent inclusion functions associated with any function computable in a finite number of steps, see, *e.g.*, [21]. Most of them are based on the extension to intervals of elementary real operations and functions. Any elementary operator \diamond such as $+$, $-$, $*$, $/$ and any elementary function f such as \exp , \sin , sqr , sqrt operating over real numbers can be extended to intervals as follows:

$$X \diamond Y = \left[\inf_{x \in X, y \in Y} x \diamond y, \sup_{x \in X, y \in Y} x \diamond y \right], \quad (6)$$

$$f(X) = \left[\inf_{x \in X} f(x), \sup_{x \in X} f(x) \right]. \quad (7)$$

For instance,

$$\begin{aligned}
[x^-, x^+] + [y^-, y^+] &= [x^- + y^-, x^+ + y^+], \\
[x^-, x^+] * [y^-, y^+] &= [\min(x^- y^-, x^- y^+, x^+ y^-, x^+ y^+), \\
&\quad \max(x^- y^-, x^- y^+, x^+ y^-, x^+ y^+)] \\
\exp([x^-, x^+]) &= [\exp(x^-), \exp(x^+)], \\
\sqrt{[x^-, x^+]} &= [\sqrt{\max(x^-, 0)}, \sqrt{x^+}] \text{ if } x^+ \geq 0 \text{ and } \emptyset \text{ if } x^+ < 0.
\end{aligned} \tag{8}$$

The following example shows how these elementary interval operations can be used to build an inclusion function.

Example 1 Consider the function

$$f(x_1, x_2) = \frac{1}{\sqrt{2\pi x_2}} e^{-\frac{1}{2} \frac{(t-x_1)^2}{x_2}},$$

where t is a known real number. An inclusion function F for f is given by

$$F(X_1, X_2) = \frac{1}{\sqrt{2\pi X_2}} e^{-\frac{1}{2} \frac{(t-X_1)^2}{X_2}},$$

where the operations involved are those of interval arithmetic as defined by (6) and (7) and where real numbers are viewed as degenerate intervals.

The algorithm presented in Section III. involves an inclusion function $J(\vec{P})$ for the cost function $j(\vec{p})$. This inclusion function will be based on the new notions of separation and inclusion degrees. Let X and Y be two scalar intervals. The *separation degree* $\text{sep}(X, Y)$ between X and Y and the *inclusion degree* $\text{incl}(X, Y)$ of X into Y are defined by

$$\begin{aligned}
\text{sep}(X, Y) &= 1 \quad \text{if } X \cap Y = \emptyset \quad \text{and} \quad \text{sep}(X, Y) = 0 \quad \text{otherwise,} \\
\text{incl}(X, Y) &= 1 \quad \text{if } X \subset Y \quad \quad \quad \text{and} \quad \text{incl}(X, Y) = 0 \quad \text{otherwise.}
\end{aligned} \tag{9}$$

These notions extend to boxes \vec{X} and \vec{Y} of $\mathbb{I}\mathbb{R}^n$ as follows:

$$\begin{aligned}
\text{sep}(\vec{X}, \vec{Y}) &= \sum_{i=1}^n \text{sep}(X_i, Y_i), \\
\text{incl}(\vec{X}, \vec{Y}) &= \sum_{i=1}^n \text{incl}(X_i, Y_i).
\end{aligned} \tag{10}$$

The following lemma will be used in the next section to prove that the interval function $J(\vec{P})$ is an inclusion function. Its proof is trivial.

Lemma 1: Let $\vec{X}, \vec{Y}, \vec{X}_1, \vec{Y}_1$ be four boxes of $\mathbb{I}\mathbb{R}^n$ such that $\vec{X}_1 \subset \vec{X}$ and $\vec{Y}_1 \subset \vec{Y}$ and let \vec{x} be a vector of \mathbb{R}^n , then

$$\begin{aligned}
\text{(i)} \quad &\text{sep}(\vec{X}, \vec{Y}) > 0 \Leftrightarrow \vec{X} \cap \vec{Y} = \emptyset, \\
\text{(ii)} \quad &\text{incl}(\vec{X}, \vec{Y}) = n \Leftrightarrow \vec{X} \subset \vec{Y}, \\
\text{(iii)} \quad &\text{sep}(\vec{x}, \vec{Y}) + \text{incl}(\vec{x}, \vec{Y}) = n, \\
\text{(iv)} \quad &\text{sep}(\vec{X}_1, \vec{Y}_1) \geq \text{sep}(\vec{X}, \vec{Y}), \\
\text{(v)} \quad &\text{incl}(\vec{X}_1, \vec{Y}) \geq \text{incl}(\vec{X}, \vec{Y}_1).
\end{aligned} \tag{11}$$

III. GUARANTEED OMNE

The set of all parameter vectors that are consistent with at least $n_y - q$ data can then be defined as:

$$\mathcal{S}_q = \{\vec{p} \in \mathbb{R}^{n_p} | j(\vec{p}) \leq q\} = j^{-1}([0, q]). \quad (12)$$

Characterizing \mathcal{S}_q has thus been cast into the framework of set inversion, and can therefore be performed by SIVIA [12] for any given value of q . A recursive version of the resulting algorithm using the notions of separation and inclusion degrees is now presented. It relies on the following proposition:

Proposition 1: An inclusion function for $j(\vec{p})$ is

$$J(\vec{P}) = \left[\text{sep}(\vec{F}(\vec{P}), \vec{Y}) \ , \ n_y - \text{incl}(\vec{F}(\vec{P}), \vec{Y}) \right] \quad (13)$$

Proof: Let \vec{P} be a box of \mathbb{R}^{n_p} and \vec{p} be any vector in \vec{P} . We shall prove that $j(\vec{p})$ is inside the interval $J(\vec{P})$ defined by (13). From the definition of j , $j(\vec{p}) = \text{sep}(\vec{f}(\vec{p}), \vec{Y})$. Parts (iii) and (v) of Lemma 1 then imply that $j(\vec{p}) = n_y - \text{incl}(\vec{f}(\vec{p}), \vec{Y}) \leq n_y - \text{incl}(\vec{F}(\vec{P}), \vec{Y})$, and part (iv) of Lemma 1 implies that $j(\vec{p}) = \text{sep}(\vec{f}(\vec{p}), \vec{Y}) \geq \text{sep}(\vec{F}(\vec{P}), \vec{Y})$. ■

Note that if $\text{ub}(J(\vec{P})) \leq q$, then $\vec{P} \subset \mathcal{S}_q$ and if $\text{lb}(J(\vec{P})) > q$, then $\vec{P} \cap \mathcal{S}_q = \emptyset$, where *ub* and *lb* stand for *upper* and *lower bound*, respectively. If the box \vec{P} satisfies neither of the two previous conditions, then \vec{P} is said to be *indeterminate*. In what follows, search will only take place in some box \vec{P}_0 of parameter space, assumed to be large enough to contain \mathcal{S}_q . SIVIA(q) and its subroutine CLASSIFY, presented on Table 1, generate two subpavings (*i.e.*, finite unions of nonoverlapping boxes). The first of them, denoted by $\mathbb{K}^-(q)$, contains all boxes that have been proved to be included in \mathcal{S}_q ; the second one, $\Delta\mathbb{K}(q)$, contains all indeterminate boxes deemed too small to be bisected.

SIVIA (in: q ; out: $\Delta\mathbb{K}, \mathbb{K}^-$)
$\Delta\mathbb{K} := \emptyset; \mathbb{K}^- := \emptyset;$ $c_0 := \text{CLASSIFY}(\vec{P}_0);$ if $c_0 := 1, \mathbb{K}^- := \{\vec{P}_0\};$ if $c_0 := [0, 1], \Delta\mathbb{K} := \{\vec{P}_0\};$ return $\Delta\mathbb{K}, \mathbb{K}^-;$
CLASSIFY (in: \vec{P} ; out c)
if $\text{ub}(J(\vec{P})) \leq q, \text{return}(1);$ if $\text{lb}(J(\vec{P})) > q, \text{return}(0);$ if $w(\vec{P}) < \varepsilon, \text{return}([0, 1]);$ bisect \vec{P} to get \vec{P}_1 and $\vec{P}_2;$ $c_1 := \text{CLASSIFY}(\vec{P}_1); c_2 := \text{CLASSIFY}(\vec{P}_2);$ if $c_1 = c_2, \{c = c_1; \text{return}; \}$ if $c_1 = [0, 1], \text{store } \vec{P}_1 \text{ into } \Delta\mathbb{K};$ if $c_2 = [0, 1], \text{store } \vec{P}_2 \text{ into } \Delta\mathbb{K};$ if $c_1 = 1, \text{store } \vec{P}_1 \text{ into } \mathbb{K}^-;$ if $c_2 = 1, \text{store } \vec{P}_2 \text{ into } \mathbb{K}^-;$ $c = \emptyset; \text{return}.$

Table 1: A recursive implementation of SIVIA, and its subroutine CLASSIFY.

If $\text{CLASSIFY}(\vec{P})$ returns 1, \vec{P} is guaranteed to be in \mathcal{S}_q . It can therefore be put in \mathbb{K}^- . If $\text{CLASSIFY}(\vec{P})$ returns 0, \vec{P} is guaranteed to have a void intersection with \mathcal{S}_q and may be discarded. All remaining boxes will be split into subboxes, unless they are smaller than a given required accuracy ε , in which case they will be put in $\Delta\mathbb{K}$. In either case, CLASSIFY refrains from immediately storing \vec{P} in the corresponding subpaving, in the hope that it may be possible to reunite it later with another box, thereby decreasing the total number of boxes to be stored. Note that when $c_1 = 0$ (resp. $c_2 = 0$), the box \vec{P}_1 (resp. \vec{P}_2) is eliminated. Returning \emptyset indicates to the calling program that \vec{P} or its subboxes have all been classified and need no longer be considered.

If $\text{SIVIA}(q)$ returns empty $\mathbb{K}^-(q)$ and $\Delta\mathbb{K}(q)$, there does not exist any vector \vec{p} consistent with at least $n_y - q$ data, so \mathcal{S}_q is empty. Therefore, there are at least $q + 1$ outliers. If $\text{SIVIA}(q)$ returns an empty $\mathbb{K}^-(q)$ and a nonempty $\Delta\mathbb{K}(q)$, then any vector $\vec{p} \in \mathcal{S}_q$ belongs to $\Delta\mathbb{K}(q)$, but it is impossible to know whether \mathcal{S}_q is empty. If the required accuracy ε is small enough, this situation occurs only in atypical situations studied in [12]. Indetermination can therefore generally be removed by re-executing $\text{SIVIA}(q)$ with a smaller accuracy coefficient ε . If $\text{SIVIA}(q)$ returns a nonempty $\mathbb{K}^-(q)$, then \mathcal{S}_q is nonempty and a bracketing of \mathcal{S}_q is given by the inclusions

$$\mathbb{K}^-(q) \subset \mathcal{S}_q \subset \mathbb{K}^-(q) \cup \Delta\mathbb{K}(q). \quad (14)$$

The main algorithm GOMNE (Guaranteed Outlier Minimal Number Estimator), presented in Table 2, uses SIVIA(q) as a subroutine and aims at characterizing the set \mathcal{S}^* . GOMNE starts by calling SIVIA(q) with $q = 0$, which amounts to assuming that there are no outliers, and an adaptive ε . Whenever SIVIA(q) has found \mathcal{S}_q empty, GOMNE increases q by one before calling SIVIA(q) again. When SIVIA(q) returns a nonempty $\mathbb{K}^-(q)$, then $j^* = q$, and a guaranteed characterization for the solution set \mathcal{S}^* is given by the inclusion

$$\mathbb{K}^-(j^*) \subset \mathcal{S}^* \subset \mathbb{K}^-(j^*) \cup \Delta\mathbb{K}(j^*) \quad (15)$$

GOMNE (\vec{P}_0, ε_0)
$q := -1;$ repeat <div style="margin-left: 2em;">$\varepsilon := \varepsilon_0; q := q + 1;$</div> repeat <div style="margin-left: 4em;">$[\mathbb{K}^-, \Delta\mathbb{K}] := \text{SIVIA}(q);$</div> <div style="margin-left: 4em;">$\varepsilon := \varepsilon/2;$</div> <div style="margin-left: 2em;">until ($\mathbb{K}^- \neq \emptyset$ or $\mathbb{K}^- = \Delta\mathbb{K} = \emptyset$);</div> until $\mathbb{K}^- \neq \emptyset$.

Table 2: Guaranteed Outlier Minimal Number Estimator.

Remark 1 *The data considered as outliers may vary from one value of \vec{p} in \mathcal{S}_q to the other. GOMNE does not explore successively all possible combinations of q potential outliers among n_y data, and thus escapes combinatorial explosion.* \diamond

Remark 2 *Undetected outliers may result in the elimination of portions of parameter space that should not have been eliminated, and this may lead to an erroneous conclusion as to the possible values for the parameters. To protect oneself against any given number n_u of such undetected outliers, one may choose to increase the minimum number j^* of outliers by n_u , and characterize the set of all parameter vectors that are consistent with any data set deprived of at least $j^* + n_u$ data points. This only requires one additional call to SIVIA($j^* + n_u$).* \diamond

Remark 3 *In the atypical case where the interior of \mathcal{S}^* is empty, for instance when \mathcal{S}^* is a singleton, the condition ($\mathbb{K}^- \neq \emptyset$ or $\mathbb{K}^- = \Delta\mathbb{K} = \emptyset$) will never be satisfied, and GOMNE will continue halving ε forever. For this reason, and to take into account the fact that the algorithm is implemented with finite-precision arithmetic, this condition is replaced by ($\mathbb{K}^- \neq \emptyset$ or $\mathbb{K}^- = \Delta\mathbb{K} = \emptyset$ or $\varepsilon < \varepsilon_{min}$) where $\varepsilon_{min} > 0$ is a small tuning coefficient to be chosen by the user. When GOMNE indicates that the condition $\varepsilon < \varepsilon_{min}$ has been*

encountered, (14) no longer holds, and one can only guarantee that, upon completion of GOMNE, $\mathcal{S}^* \subset \mathbb{K}^- \cup \Delta\mathbb{K}$. \diamond

Remark 4 GOMNE could be extended to include prior information. For instance, assume that the parameter vector \vec{p} should belong to the set $\mathcal{G} = \{\vec{p} \in \mathbb{R}^{n_p} \mid g(\vec{p}) \leq 0\}$, where g is a possibly nonlinear function. Then, the set of all parameter vectors that are consistent with at least $n_y - q$ data becomes

$$\mathcal{S}_q = \{\vec{p} \in \mathbb{R}^{n_p} \mid g(\vec{p}) \leq 0 \text{ and } j(\vec{p}) \leq q\} = g^{-1}(]-\infty, 0]) \cap j^{-1}([0, q])$$

and the minimum outlier feasible set \mathcal{S}^* is equal to \mathcal{S}_{q^*} where

$$q^* = \min \{q \in \{0, \dots, n_y\} \text{ such that } \mathcal{S}_q \neq \emptyset\}$$

Assume that an inclusion function G is available for g , then if we replace the two statements

$$\begin{aligned} \text{if } ub(J(\vec{P})) \leq q, & \quad \text{return}(1); \\ \text{if } lb(J(\vec{P})) > q, & \quad \text{return}(0); \end{aligned}$$

of CLASSIFY by

$$\begin{aligned} \text{if } \left(ub(J(\vec{P})) \leq q \right) \text{ and } \left(ub(G(\vec{P})) \leq 0 \right), & \quad \text{return}(1); \\ \text{if } \left(lb(J(\vec{P})) > q \right) \text{ or } \left(lb(G(\vec{P})) > 0 \right), & \quad \text{return}(0); \end{aligned}$$

then GOMNE will provide a guaranteed characterization of \mathcal{S}^* . \diamond

Remark 5 The complexity of SIVIA is exponential in the number of parameters [12]. Therefore, GOMNE has also an exponential complexity and is thus limited to small dimensional problems. Note that the exponential ratio can be reduced by using contractors (see, e.g., Chapter 5 of [11]). Contractors are operators used to contract boxes \vec{P} without losing any feasible parameter vector. \diamond

IV. CONNEXITY ANALYSIS

This section presents some new results on connexity that may help to give an interpretation of each connected components of the sets \mathcal{S}^* and \mathcal{S}_q . These results will make it possible to decide which data points have been considered as outliers by the algorithm GOMNE.

Assume that the vector function \vec{f} is continuous. Let $\phi(\vec{p})$ be the set of all i such that $f_i(\vec{p}) \notin Y_i$.

Lemma 2: Any vector \vec{p} of \mathbb{R}^{n_p} belongs to an open set \mathcal{O} such that $\forall \vec{p}_1 \in \mathcal{O}, \phi(\vec{p}) \subset \phi(\vec{p}_1)$. \diamond

Proof: Let i be any integer in $\phi(\vec{p})$. Since f is continuous, $\mathcal{O}_i \triangleq f_i^{-1}([-\infty, y_i^-[\cup]y_i^+, \infty[)$ is an open set of \mathbb{R}^{n_p} that contains \vec{p} . So is $\mathcal{O} \triangleq \bigcap_{i \in \phi(\vec{p})} (\mathcal{O}_i)$. If $\vec{p}_1 \in \mathcal{O}, \forall i \in \phi(\vec{p}), f_i(\vec{p}_1) \notin Y_i$ i.e. $i \in \phi(\vec{p}_1)$. Therefore $\phi(\vec{p}) \subset \phi(\vec{p}_1)$. \blacksquare

Lemma 3: Let \vec{p}_1 and \vec{p}_2 be two vectors in $j^{-1}(q)$ such that $\phi(\vec{p}_1) \neq \phi(\vec{p}_2)$. The two sets $\mathcal{P}_1 = \{\vec{p} \in \mathbb{R}^{n_p} | \phi(\vec{p}) = \phi(\vec{p}_1)\}$ and $\mathcal{P}_2 = \{\vec{p} \in \mathbb{R}^{n_p} | \phi(\vec{p}) = \phi(\vec{p}_2)\}$ are not connected. \diamond

Proof: Since $\phi(\vec{p}_1) \neq \phi(\vec{p}_2), \mathcal{P}_1 \cap \mathcal{P}_2 = \emptyset$. The fact that the intersection of two sets is empty does, however, not imply that they are disconnected. Connection may still take place in $\text{clo}(\mathcal{P}_1) \cap \text{clo}(\mathcal{P}_2)$, as exemplified by the sets $[2, 3[$ and $[3, 4]$, such that $\text{clo}([2, 3[) \cap \text{clo}([3, 4]) = 3$. The proof of Lemma 3 is by contradiction. Since $\phi(\vec{p}_1)$ and $\phi(\vec{p}_2)$ both have q elements, i.e., $\#\phi(\vec{p}_1) = \#\phi(\vec{p}_2) = q$, and since $\phi(\vec{p}_1) \neq \phi(\vec{p}_2)$, we have

$$\#(\phi(\vec{p}_1) \cap \phi(\vec{p}_2)) < q. \quad (16)$$

Assume that \mathcal{P}_1 and \mathcal{P}_2 are connected, then there exists $\vec{p} \in \mathcal{P}_1 \cup \mathcal{P}_2$ such that $\vec{p} \in \text{clo}(\mathcal{P}_1) \cap \text{clo}(\mathcal{P}_2)$. Since $\vec{p} \in \text{clo}(\mathcal{P}_1)$ and because of Lemma 2, $\exists \vec{r}_1 \in \mathcal{P}_1$ such that $\phi(\vec{p}) \subset \phi(\vec{r}_1) = \phi(\vec{p}_1)$. For the same reasons, $\phi(\vec{p}) \subset \phi(\vec{p}_2)$. This amounts to writing that $\phi(\vec{p}) \subset (\phi(\vec{p}_1) \cap \phi(\vec{p}_2))$, which would imply that

$$\#\phi(\vec{p}) \leq \#(\phi(\vec{p}_1) \cap \phi(\vec{p}_2)). \quad (17)$$

Now, since $\vec{p} \in \mathcal{P}_1 \cup \mathcal{P}_2 \subset j^{-1}(q)$,

$$j(\vec{p}) = \#\phi(\vec{p}) = q. \quad (18)$$

(16), (17) and (18) are incompatible. \blacksquare

Proposition 2: In any given connected component of $j^{-1}(q)$, the q data considered as outliers are the same. \diamond

Proof: Lemma 3 implies that $\phi(\vec{p})$ cannot vary when \vec{p} describes a given connected component of $j^{-1}(q)$. \blacksquare

Since $\mathcal{S}^* = j^{-1}(j^*)$, the set $\phi(\mathcal{S}^*) \triangleq \{\phi(\vec{p}) | \vec{p} \in \mathcal{S}^*\}$ contains all possible ways of discarding exactly j^* data points to obtain a nonempty feasible set for \vec{p} . A consequence of this proposition is that if \mathcal{S}^* is connected, then this choice is unique, and obtained by computing $\phi(\vec{p})$ at any \vec{p} in \mathcal{S}^* . More generally, any given connected component of \mathcal{S}^* corresponds to a single possible choice of the data considered as outliers. If \mathcal{S}^* turns out to be disconnected, $\phi(\mathcal{S}^*)$ can be computed by evaluating $\phi(\vec{p})$ at only one value of \vec{p} in each connected component of \mathcal{S}^* . If the value of $\phi(\mathcal{S}^*)$ is the same on each of them, then the fact that \mathcal{S}^* is disconnected must have other explanations than an exchange in the data classified as outliers, e.g., an identifiability problem.

Example 2 As an illustration, assume that the i th model output is given by

$$f_i(\vec{p}) = \frac{1}{\sqrt{2\pi p_2}} \exp\left(-\frac{(t_i - p_1)^2}{2p_2}\right),$$

which corresponds to the value of a Gaussian probability density function at t_i . Assume that uncertain measurements of the values taken by the probability density function for some values of t_i are available under the form of the feasible intervals Y_i , as indicated in the following table

t_i	1	2	3	4	5	6
Y_i	[0; 0.2]	[0.3; 2]	[0.3; 2]	[0.1; 0.2]	[0.4; 2]	[-1; 0.1]

In this context $j^{-1}(q)$ corresponds to the set of all \vec{p} that are consistent with exactly $6 - q$ of the interval data Y_i . SIVIA has been used to characterize $j^{-1}(q)$ for $q = 0, \dots, 6$. In less than 0.01 second on a Pentium 233 MMX, it shows that $j^{-1}(0) = j^{-1}(6) = \emptyset$, which means that there exists no Gaussian function that goes through all six data bars and no Gaussian function that avoids all six bars. Note that this result, could not be obtained by random search or gridding. Figure 1 (left) represents the sets $j^{-1}(q)$, for $q \in \{1, 2, 3, 4, 5\}$. Each of them have been obtained by SIVIA in less than 2 seconds. The frame box is $\vec{P} = [-2, 8] \times [0, 3]$. The dark gray boxes have been proven to be inside $j^{-1}(q)$ whereas the light gray boxes have been proven to be outside $j^{-1}(q)$. The sets obtained by SIVIA are consistent with the fact that the sets $j^{-1}(0), j^{-1}(1), \dots, j^{-1}(6)$ should form a partition of \mathbb{R}^2 , i.e., $j^{-1}(q_1) \cap j^{-1}(q_2) = \emptyset$ if $q_1 \neq q_2$ and $\bigcup_{q=0}^{q=6} j^{-1}(q) = \mathbb{R}^2$. Figure 1 (right) represents the superposition of all Gaussian functions whose mean p_1 and variance p_2 are in $j^{-1}(q)$. The frame box is $[-1, 8] \times [-2, 4]$. For $q = 1$, SIVIA concludes that $j^{-1}(q)$ is connected, and that for all \vec{p} in $j^{-1}(q)$, $\phi(\vec{p}) = \{1, 2, 3, 4, 6\}$. The fact that the same datum (the 5th) is considered as an outlier for all $\vec{p} \in j^{-1}(1)$ could have been forecasted from Proposition 2. For $q = 2$, SIVIA shows that $j^{-1}(q)$ consists of four connected components and that $\phi(j^{-1}(q)) = \{\{1, 2, 3, 4\}, \{1, 2, 3, 6\}, \{1, 3, 4, 6\}, \{1, 4, 5, 6\}\}$. This means that there exists 4 different ways for a Gaussian function to cross 4 of the 6 bars, each of which corresponding to a connected component of $j^{-1}(2)$ (see Proposition 2).

Remark 6 In some applications, it might be useful to assign some measure of reliability to each data point. A possible such measure, proposed in [10], is the safety of the i th interval data, defined as the ratio of the volume of \mathcal{S}^* to that of the feasible set obtained when the i th interval data is discarded. Safety is always between 0 and 1, and the larger it is, the more the information provided by a given data interval is confirmed by the others. \diamond

The next section shows how GOMNE can be applied to an actual estimation problem where outliers are unavoidable.

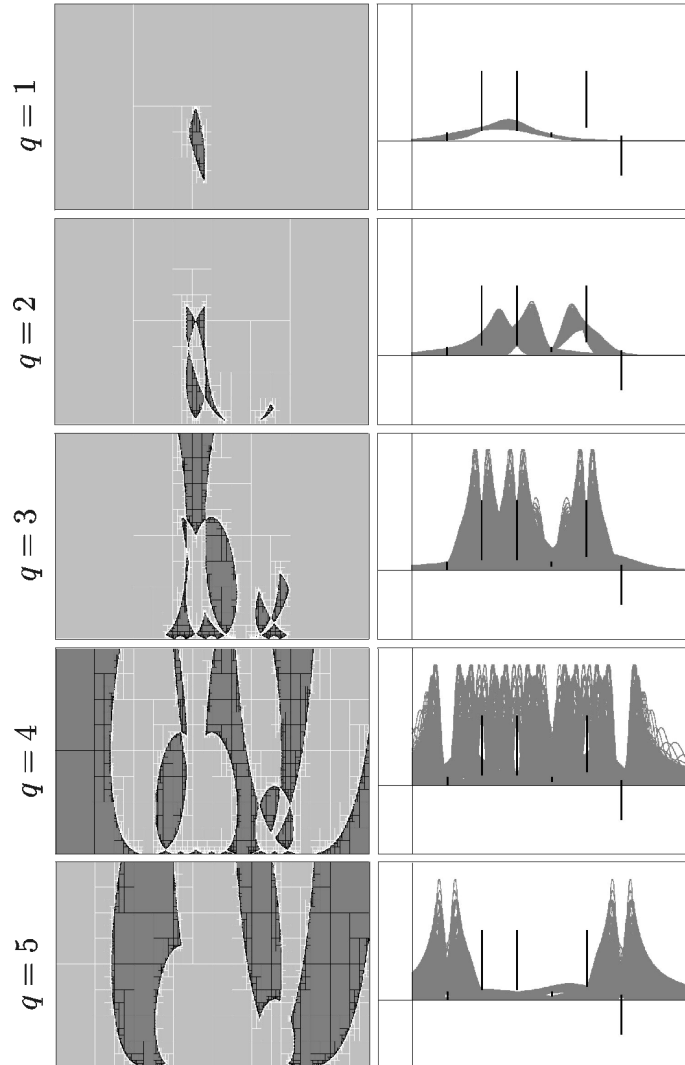


Figure 1: Left: sets $j^{-1}(q)$ of all \vec{p} consistent with exactly $6 - q$ bars; right: associated Gaussian probability density functions

V. ROBOT LOCALIZATION

The autonomous localization of a robot in a partially known environment is a key problem of mobile robotics. A variety of sensors may be used, each of them providing uncertain measurements that must be combined, and this localization is archetypal of problems of data fusion [4], [2]. To localize itself dynamically, a robot must first estimate its initial configuration. This problem has been the subject of a renewed interest during the last four years. Crowley et. al. [5] used pattern recognition to match dense measurements with a set of landmarks via a principal component representation of the measurements. Markovian approaches [8], [15] consider localization as the computation of a probability distribution over the free configuration space. Methods with adaptative step sizes have been proposed, based on the use of octrees [1] or on Monte-Carlo sampling [6]. The approach presented in [22] has strong similarities with that presented in this paper. It partitions the free space iteratively with boxes and tests some characteristic points inside each box. If the test is negative, the whole box is discarded, else if its width is larger than a given precision, it is split into subboxes. But, since only a finite number of points are tested in each box, this method is not guaranteed.

Let us stress that the initialization phase is more complex problem than tracking proper. Indeed, once a reasonably accurate initialization has been performed, it is often possible to use a well established local tracking technique such as Kalman filtering [17] or its bounded-error counterpart [9], [19].

This section shows how GOMNE behaves for the static localization [18] of the robot presented on Figure 2 from measurements provided by onboard sensors in a partially known 2D environment. The aim is to estimate the *configuration* of the robot $\vec{p} = (p_1, p_2, p_3)^T$, where (p_1, p_2) is its position in the world frame and p_3 its heading angle with respect to some reference direction. In what follows, p_1 and p_2 are expressed in meters and p_3 in radians.

This configuration is assumed to belong to some box \vec{P}_0 in configuration space, large enough to contain all configurations of interest. The landmarks of the environment are assumed to be j_{\max} oriented segments, the collection of which constitutes the *map*. The two extreme points of the j th segment are denoted by \vec{a}_j and \vec{b}_j , with the convention that when going from \vec{a}_j to \vec{b}_j , the reflecting face of the segment is on the left. The robot is equipped with $i_{\max} = 24$ on-board Polaroid ultrasonic sensors, represented by crosses on Figure 3. The i th sensor is installed on the vehicle at coordinates $(\tilde{x}_i, \tilde{y}_i)$ in the robot frame. The i th segment gives the orientation $\tilde{\theta}_i$ of the emission axis. Figure 3 illustrates these notations.

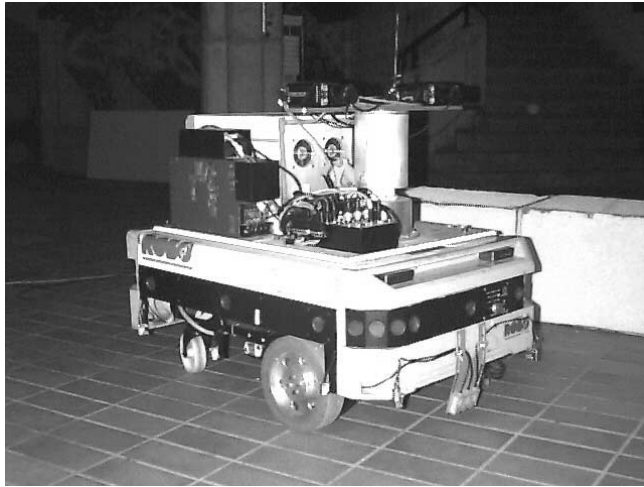


Figure 2: Robot actually used in experimentation

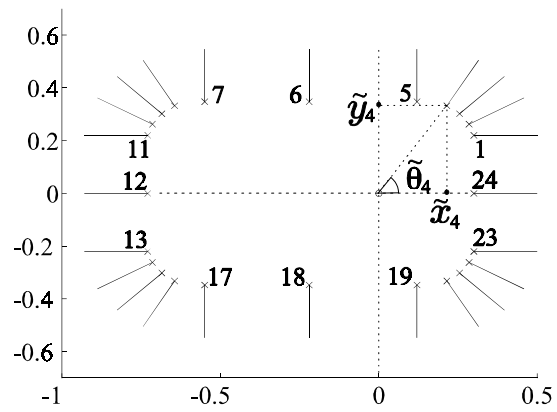


Figure 3: Locations and orientations of the sensors in the robot frame

The i th sensor is expected to return a measurement y_i that corresponds to the distance to some unknown landmark at least partly located in its emission cone. To take measurement inaccuracy into account, with each data point y_i is associated a feasible interval $Y_i = [y_i(1 - \alpha), y_i(1 + \alpha)]$, where α , the relative precision of the measurement, is assumed to be known and, for simplicity, independent of the sensor considered. With each data point y_i , a thick uncertainty arc can then be associated, resulting from the intersection of the corresponding emission cone with the ring centered on the i th sensor with interval radius Y_i . A configuration \vec{p} is consistent with Y_i if the i th thick arc intersects a landmark and no other landmark is located between this arc and the i th ultrasonic sensor.

	$\vec{f}(\vec{p})$
<p>input: $\vec{p} = (p_1, p_2, p_3)^T$;</p> <p>for $i = 1$ to i_{\max}</p> $\vec{s} := \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} \cos(p_3) & -\sin(p_3) \\ \sin(p_3) & \cos(p_3) \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \end{bmatrix};$ $\vec{u} := \begin{bmatrix} \cos(p_3 + \tilde{\theta}_i - \gamma) \\ \sin(p_3 + \tilde{\theta}_i - \gamma) \end{bmatrix}; \vec{v} := \begin{bmatrix} \cos(p_3 + \tilde{\theta}_i + \gamma) \\ \sin(p_3 + \tilde{\theta}_i + \gamma) \end{bmatrix};$ <p>for $j = 1$ to j_{\max}</p> $\vec{a} := \vec{a}_j; \vec{b} := \vec{b}_j; \vec{w} := \vec{b} - \vec{a}; \vec{m} := \vec{a} - \vec{s}; \vec{n} := \vec{b} - \vec{s};$ $r_h := \infty; r_{h_1} := \infty; r_{h_2} := \infty; r_a := \infty; r_b := \infty; r_j := \infty;$ <p>if $\vec{w} \cdot \vec{m} \leq 0 \wedge \vec{w} \cdot \vec{n} \geq 0 \wedge \vec{u} \cdot \vec{w} \leq 0 \wedge \vec{v} \cdot \vec{w} \geq 0$, $r_h = \frac{\det(\vec{m}, \vec{w})}{\ \vec{w}\ }$;</p> <p>if $\det(\vec{u}, \vec{m}) \leq 0 \wedge \det(\vec{u}, \vec{n}) \geq 0$, $r_{h_1} := \left \frac{\det(\vec{w}, \vec{m})}{\det(\vec{w}, \vec{u})} \right$;</p> <p>if $\det(\vec{v}, \vec{m}) \leq 0 \wedge \det(\vec{v}, \vec{n}) \geq 0$, $r_{h_2} := \left \frac{\det(\vec{w}, \vec{m})}{\det(\vec{w}, \vec{v})} \right$;</p> <p>if $\det(\vec{u}, \vec{m}) \geq 0 \wedge \det(\vec{v}, \vec{m}) \leq 0$, $r_a := \ \vec{m}\$;</p> <p>if $\det(\vec{u}, \vec{n}) \geq 0 \wedge \det(\vec{v}, \vec{n}) \leq 0$, $r_b := \ \vec{n}\$;</p> <p>if $\det(\vec{m}, \vec{w}) \geq 0 \wedge (\det(\vec{u}, \vec{w}) \geq 0 \vee \det(\vec{v}, \vec{w}) \geq 0)$;</p> $r_j := \min\{r_h, r_{h_1}, r_{h_2}, r_a, r_b\};$ <p>next j;</p> $f_i := \min\{r_1, \dots, r_{j_{\max}}\};$ <p>next i;</p> <p>output: $\vec{f} = (f_1, \dots, f_{i_{\max}})^T$</p>	

Table 3: Algorithm for generating simulated measurements.

This algorithm is to be inverted by SIVIA.

For any given configuration \vec{p} , the algorithm described in Table 3 computes the vector $\vec{f}(\vec{p})$ of all measurements that would be reported by the ultrasonic sensors under idealized conditions. In this algorithm, γ denotes the half aperture of the emission cone. The configuration \vec{p} is consistent with the measurement reported by the i th sensor if $f_i(\vec{p})$

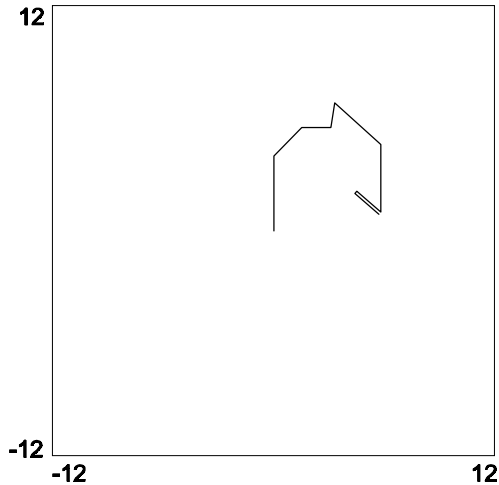


Figure 4: Map of the room in the world frame

belongs to Y_i . The methodology followed to obtain $\vec{f}(\vec{p})$ is not described here for the sake of brevity, and can be found in [13]. Although fairly realistic, this model of measurements remains simple and does not take into account many effects difficult or even impossible to incorporate. Examples of such effects are multiple nonspecular reflections, sensor failures, beam interceptions by pieces of furniture or passersby, parasitic ultrasonic signals emitted by machinery and outdated maps. Each of them can generate outliers. A robust estimation method is therefore particularly needed.

To use GOMNE to estimate the configuration of the robot, it is necessary to obtain an inclusion function for $\vec{f}(\vec{p})$. Recall that if \vec{f} is a finite algorithm which does not contain conditional statement, an inclusion function can be obtained by replacing all operations over real numbers in \vec{f} by their interval counterparts as in Example 1. Here, the conditional statements in $\vec{f}(\vec{p})$ could pose specific problems, but can be eliminated with the help of the notion of χ function developed by Kearfott [14], see [13] for more details. The use of a programming language that allows operator overloading (such as C++, FORTRAN 90 or ADA) then makes it possible to use exactly the same code to compute $\vec{F}(\vec{P})$ and $\vec{f}(\vec{p})$.

The robot represented on Figures 2 and 3 is in a room described by the map of Figure 4.

The collected measurements are represented on the emission diagram of Figure 5. The length of the dotted segments characterizes the measurement y_i ; except for outliers, some obstacle should lie at least in part between the two arcs associated with the corresponding sensor.

For a half aperture of the emission cone $\gamma = 11.3^\circ$, a relative precision of the measurements $\alpha = 0.02$, a required accuracy $\varepsilon_0 = 0.002$ for the characterization of \mathcal{S}^* and a search box

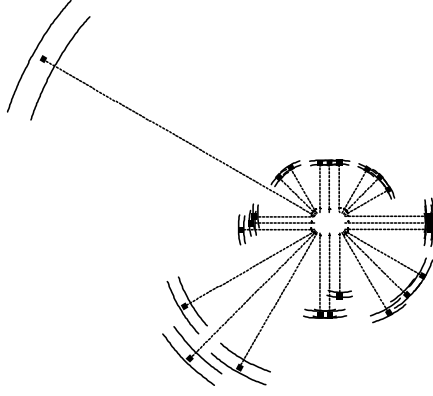


Figure 5: Representation of the data collected by the 24 sensors, with their associated uncertainties

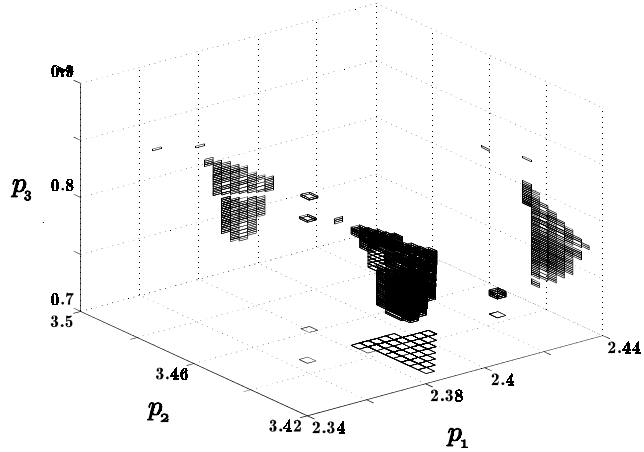


Figure 6: Subpaving generated by GOMNE, which encloses \mathcal{S}^* , and its two-dimensional projections

$\vec{P}_0 = [-12, 12] \times [-12, 12] \times [0, 2\pi]$, GOMNE detects a minimum of seven outliers. This means that it has proved that there does not exist a configuration vector \vec{p} which is consistent with more than 17 data points; we are not aware of any other method which would be able to guarantee this result. The outer subpaving $\mathbb{K}^+ = \mathbb{K}^- \cup \Delta\mathbb{K}$ for \mathcal{S}^* is represented on Figure 6, together with its 2D projections.

Computing time on a Pentium 233 MMX, is indicated on Table 4, where t_q is the computing time for SIVIA(q) and $t_q^{\text{cumul}} = \sum_{i=0}^q t_i$.

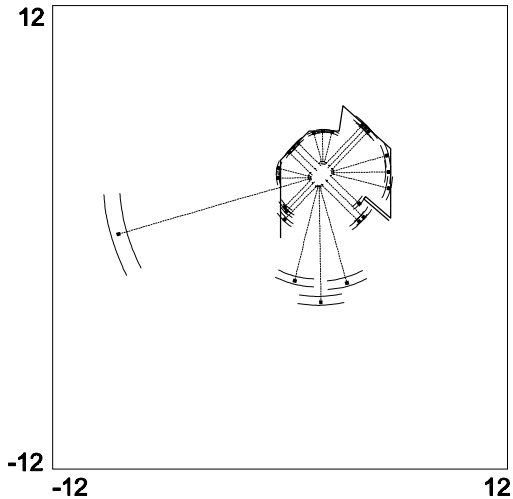


Figure 7: Point estimate of the configuration of the robot

q	$\mathbb{K}^+(q)$	$t_q(\text{sec.})$	$t_q^{\text{cumul}}(\text{sec.})$
0	\emptyset	2	2
1	\emptyset	5.5	7.5
2	\emptyset	9	16.5
3	\emptyset	15	31.5
4	\emptyset	22	53.5
5	\emptyset	30.5	84
6	\emptyset	43	127
7	See Figure 6	61	188

Table 4: Computing times.

The smallest box that contains $\mathbb{K}^+ = \mathbb{K}^+(7)$ is $\vec{P}_1 = [2.364, 2.429] \times [3.425, 3.463] \times [0.716, 0.826]$. The configuration associated with its center is represented on Figure 7.

VI. CONCLUSIONS

One of the most attractive features of bounded-error estimation is the fact that its results can be guaranteed even for nonlinear models and finite data sets, provided that a few explicit hypotheses on the size of the acceptable errors are satisfied. Unfortunately, it is extremely difficult to guarantee that this condition will always be satisfied, and that no outliers will creep among the data. The purpose of the outlier minimal number estimator (OMNE) is to make parameter bounding robust to such outliers, and this estimator has been shown in the literature to exhibit remarkable robustness properties, even when outliers are introduced on purpose to fool it. The implementations of OMNE available

so far, however, could not guarantee their results, because they involved random search, and the advantage of bounded-error estimation previously mentioned was thus lost. The new algorithm GOMNE presented in this paper now makes it possible to characterize, in a guaranteed way, the set of all parameter vectors that are consistent with the largest possible number of data points and associated error bounds. It is important to note that the algorithm does *not* involve trying all possible ways to eliminate a given number of data points considered as candidate outliers from the data base, a task that would become tremendously complicated when the numbers of data points and potential outliers are large. The data considered as outliers are obtained as a by-product of the procedure, and may vary within the set estimated, although Proposition 2 indicates that they remain the same on any given connected component of \mathcal{S}^* .

Robot localization from measurements by onboard range sensors is an ideal application field for GOMNE for several reasons. Firstly, the number of the parameters to be estimated is small enough for the exponential complexity of SIVIA to remain manageable. Secondly, bounds are easy to obtain for the errors committed by sensors in normal operating conditions, and more realistic than the usual Gaussian assumption. Thirdly, and for a number of reasons, it is unrealistic to assume that all sensors will report correctly, and outliers cannot be avoided. Lastly, the association of measurements with the landmarks of the environment, classically considered as a crucial step of robot localization and usually handled in a very complex and heuristic way, is trivially solved as a by-product of the procedure. The problem considered here was that of static localization, which is important whenever the robot is put in some new environment or has lost track of its configuration. An automated procedure such as the one advocated here makes it possible to avoid the need for a human intervention, thereby increasing the robot autonomy.

The source code in C++ corresponding to the example of Section IV. and all associated libraries can be downloaded at <http://www.istia.univ-angers.fr/~jaulin/gausscpp.zip>.

REFERENCES

- [1] A. Burgard, D. Derr, D. Fox, and A. B. Cremers, *Integrating global position estimation and position tracking for mobile robots: the dynamic markov localization approach*, IROS Conference (Victoria, Canada), 1998, pp. 730–735.
- [2] J. A. Castellanos, J.M.M. Monteil, J. Neira, and J.D. Tardos, *The SMAP: A probabilistic framework for simultaneous localization and map building*, IEEE Transactions on Robotics and Automation **15** (1999), no. 5, 948–952.
- [3] P. L. Combettes, *The foundations of set theoretic estimation*, Proceedings of the IEEE **81** (1993), no. 2, 182–208.
- [4] J. Crowley, *World modeling and position estimation for a mobile robot using ultrasonic ranging*, Proceedings of the IEEE International Conference on Robotics and Automation (Scottsdale, AZ), 1989, pp. 674–680.
- [5] J. L. Crowley, F. Wallner, and B. Schiele, *Position estimation using principal components of range data*, Proceedings of the IEEE International Conference on Robotics and Automation (Leuven, Belgium), 1998, pp. 3121–3128.
- [6] F. Dellaert, D. Fox, A. Burgard, and S. Thrun, *Monte-Carlo localization for mobile robots*, Proceedings of the IEEE International Conference on Robotics and Automation (Detroit, Michigan), 1999, pp. 1322–1328.
- [7] J.R. Deller, M. Nayeri, and S.H. Odeh, *Least-square identification with error bounds for real-time signal processing and control*, Proceedings of the IEEE **81** (1993), no. 6, 815–849.
- [8] J.S. Gutmann, A. Burgard, D. Fox, and K. Konolige, *An experimental comparison of localization methods*, IROS conference (Victoria, Canada), 1998, pp. 736–743.
- [9] U.D. Hanebeck and G. Schmidt, *Set theoretic localization of fast mobile robots using an angle measurement technique*, Proceedings of the IEEE International Conference on Robotics and Automation (Minneapolis, MN), april 1996, pp. 1387–1394.
- [10] L. Jaulin, J. L. Godet, E. Walter, A. Elliasmine, and Y. Leduff, *Light scattering data analysis via set inversion*, Journal of Physics A: Mathematical and General (1997), 7733–7738.
- [11] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag, London, 2001.

- [12] L. Jaulin and E. Walter, *Set inversion via interval analysis for nonlinear bounded-error estimation*, *Automatica* **29** (1993), no. 4, 1053–1064.
- [13] L. Jaulin, E. Walter, O. Lévêque, and D. Meizel, *Set inversion for χ -algorithms, with application to guaranteed robot localization*, *Mathematics and Computers in Simulation* **52** (2000), no. 3-4, 197–210.
- [14] R. B. Kearfott, *A Fortran 90 environment for research and prototyping of enclosure algorithms for nonlinear and global optimization*, *ACM Trans. Math. Software* **21** (1995), no. 1, 63–78.
- [15] K. Konolige and K. Chou, *Markov localization using correlation*, *International Joint Conference on Artificial Intelligence*, 1999, pp. 1154–1159.
- [16] H. Lahanier, E. Walter, and R. Gomeni, *OMNE: a new robust membership-set estimator for the parameters of nonlinear models*, *Journal of Pharmacokinetics and Biopharmaceutics* **15** (1987), 203–219.
- [17] J. J. Leonard and H. F. Durrant-Whyte, *Mobile robot localization by tracking geometric beacons*, *IEEE Transactions on Robotics and Automation* **7** (1991), no. 3, 376–382.
- [18] O. Lévêque, L. Jaulin, D. Meizel, and E. Walter, *Vehicule localization from inaccurate telemetric data: a set inversion approach*, *Proceedings of 5th IFAC Symposium on Robot Control SY.RO.CO.'97 (Nantes, France)*, vol. 1, 1997, pp. 179–186.
- [19] D. Meizel, A. Preciado-Ruiz, and E. Halbwachs, *Estimation of mobile robot localization: geometric approaches*, *Bounding Approaches to System Identification (M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, eds.)*, Plenum Press, New York, NY, 1996, pp. 463–489.
- [20] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter (eds.), *Bounding Approaches to System Identification*, Plenum Press, New York, NY, 1996.
- [21] R. E. Moore, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, PA, 1979.
- [22] E. M. Mouaddib and B. Marhic, *Geometrical matching for mobile robot localization*, *IEEE Transactions on Robotics and Automation* **16** (2000), no. 5, 542–552.
- [23] J. P. Norton (ed.), *Special Issue on Bounded-Error Estimation: Issue 1, 1994*, *International Journal of Adaptive Control and Signal Processing* **8**(1):1–118.
- [24] J. P. Norton (ed.), *Special Issue on Bounded-Error Estimation: Issue 2, 1995*, *International Journal of Adaptive Control and Signal Processing* **9**(1):1–132.

- [25] L. Pronzato and E. Walter, *Robustness to outliers of bounded-error estimators and consequences on experiment design*, Bounding Approaches to System Identification (New York) (M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, eds.), Plenum, 1996, pp. 199–212.
- [26] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, Wiley, New York, 1987.
- [27] E. Walter (ed.), Special Issue on Parameter Identification with Error Bounds, 1990, *Mathematics and Computers in Simulation* **32**(5-6):447–607.
- [28] E. Walter and H. Piet-Lahanier, *Estimation of the parameter uncertainty resulting from bounded-error data*, *Mathematical Biosciences* **92** (1988), 55–74.