Guaranteed Robust Nonlinear Minimax Estimation

Luc Jaulin and Eric Walter

***Abstract*—Minimax parameter estimation aims at characterizing the set of all values of the parameter vector that minimize the largest absolute deviation between the experimental data and the corresponding model outputs. It is well known, however, to be extremely sensitive to outliers in the data resulting, e.g., of sensor failures. In this paper, a new method is proposed to robustify minimax estimation by allowing a prespecified number of absolute deviations to become arbitrarily large without modifying the estimates. By combining tools of interval analysis and constraint propagation, it becomes possible to compute the corresponding minimax estimates in an approximate but guaranteed way, even when the model output is nonlinear in its parameters. The method is illustrated on a problem where the parameters are not globally identifiable, which demonstrates its ability to deal with the case where the minimax solution is not unique.**

***Index Terms*—Constraint propagation, interval computation, minimax estimation, nonlinear estimation, outliers, robust estimation.**

I. INTRODUCTION

WHEN the vector $\mathbf{p}$ of the parameters of a model has to be estimated from experimental data $\mathbf{y}$, the procedure to be followed depends on the assumptions about the noise. If the components of $\mathbf{y}$ are assumed independently corrupted by an additive noise uniformly distributed over the interval $[-\delta, \delta]$, with δ unknown, then a maximum-likelihood estimate of $\mathbf{p}$ is obtained by minimizing the largest absolute deviation between the data and the corresponding model outputs, which correspond to minimax estimation. The resulting estimate $\widehat{\mathbf{p}}$ belongs to the set of all parameter vectors that are consistent with any value of δ large enough for the set to be nonempty. Moreover, the corresponding largest absolute deviation $\hat{\delta}$ is a lower bound for δ, which provides useful information to anyone interested in bounded-error parameter estimation. (See [6], [15], [18], [20], [21], [25], and the references therein).

Minimax estimation is well known, however, to be extremely sensitive to outliers, as a single of them may suffice to ruin the estimate [22], [3]. Outliers are data that result of events not accounted for by the model, such as sensor failures, transcription errors or erroneous hypotheses on noise distribution. The purpose of this paper is to present a new algorithm for computing guaranteed robust minimax estimates, robust meaning here that a prespecified number of absolute deviations are allowed to become arbitrarily large, and guaranteed meaning that an outer approximation of the set of robust minimax estimates is obtained. The basic idea is akin to that in [16] and [3], but its implementation is radically new. It combines the tools of *interval analysis* and *constraint propagation* in what is known as *interval constraint propagation* (ICP) [7], [8] to provide guaranteed results (contrary to [16]) for nonlinear models (contrary to [3]) in the presence of outliers (contrary to [13]). Estimation methods based on ICP are described in [14].

Some basic notions of ICP are recalled in Section III, and the necessity of extending it to deal with robust minimax estimation is stressed. Section IV describes a rather classical optimization algorithm based on ICP. In order to allow the development of an efficient reduction procedure, able to handle robust minimax estimation, Section V introduces the notion of set polynomials. To the best of our knowledge, this notion is new, at least in this context. A test case is presented in Section V to demonstrate the efficiency of the approach.

II. RELAXED MINIMAX ESTIMATOR

In what follows, the parameter vector $\mathbf{p} \in \mathbb{R}^n$ is assumed to belong to some possibly very large prior axis-aligned search box $[\mathbf{p}_0]$. Let $\mathbf{y} \in \mathbb{R}^m$ be the data vector, $\mathbf{y}_m(\mathbf{p}) \in \mathbb{R}^m$ be the associated model output vector and $\mathbf{f}(\mathbf{p}) \in \mathbb{R}^m$ be the (absolute) error vector defined as $|\mathbf{y} - \mathbf{y}_m(\mathbf{p})|$, where the absolute value is taken componentwise. Denote by q the number of data points where the error is allowed to become arbitrarily large. Define the q-max function from $\mathbb{R}^m \to \mathbb{R}$, where m is an integer with $m \geq q \geq 0$, as the function that associates with $\mathbf{x} = (x_1, \ldots, x_m)^T$ its $(q+1)$th largest entry. By convention for $q = m$, we shall take $q-\max(\mathbf{x}) = -\infty$. For example, if $\mathbf{x} = (3, -4, 3, 5, 0)^T$, then $0-\max(\mathbf{x}) = 5$, $1-\max(\mathbf{x}) = 3$, $2-\max(\mathbf{x}) = 3$, $3-\max(\mathbf{x}) = 0$, $4-\max(\mathbf{x}) = -4$, and $5-\max(\mathbf{x}) = -\infty$. In a robust minimax context, the cost function to be used if q outliers are assumed can be written

$$j_q(\mathbf{p}) = q-\max(\mathbf{f}(\mathbf{p})). \tag{1}$$

With any given q and $\mathbf{y}$, the *relaxed minimax estimator* (RME) associates the set $\widehat{\mathcal{S}}_q = \arg\min_{\mathbf{p}\in[\mathbf{p}_0]} j_q(\mathbf{p})$. Since $j_q(\mathbf{p})$ is a decreasing function of q, the minimum $\widehat{j}_q$ of $j(\mathbf{p})$ over $[\mathbf{p}_0]$ is also a decreasing function of q (i.e., $q_1 \leq q_2 \Leftrightarrow \hat{j}_{q_2} \leq \hat{j}_{q_1}$). The set $\mathcal{S}_q(\delta) \triangleq \{\mathbf{p} \in [\mathbf{p}_0] \mid j_q(\mathbf{p}) \leq \delta\}$ is increasing with q (i.e., $\mathcal{S}_0(\delta) \subset \mathcal{S}_1(\delta) \subset \cdots \subset \mathcal{S}_m(\delta) = [\mathbf{p}_0]$) and with δ (i.e., $\delta_1 \leq \delta_2 \Leftrightarrow \mathcal{S}_q(\delta_1) \subset \mathcal{S}_q(\delta_2)$). Fig. 1 illustrates these properties when the dimension of $\mathbf{p}$ is 1. For readability, the dependency of $\mathcal{S}_q$ in δ is not mentioned. The set $\widehat{\mathcal{S}}_q$ of all global minimizers of $j_q(\mathbf{p})$ over $[\mathbf{p}_0]$ will be computed by the algorithm MINIMIZE, to be presented in Section IV, which is based on interval constraint propagation, briefly recalled in Section III

Manuscript received May 18, 2000; revised May 17, 2002. Recommended by Associate Editor W. M. McEneaney.

L. Jaulin is with the Laboratoire d'Ingénierie des Systèmes Automatisés, Faculté des Sciences, Université d'Angers, 49000 Angers, France (e-mail: luc.jaulin@univ-angers.fr).

E. Walter is with the Laboratoire des Signaux et Systèmes, CNRS-Supélec-Université Paris Sud, 91192 Gif-sur-Yvette, France.

Digital Object Identifier 10.1109/TAC.2002.804479

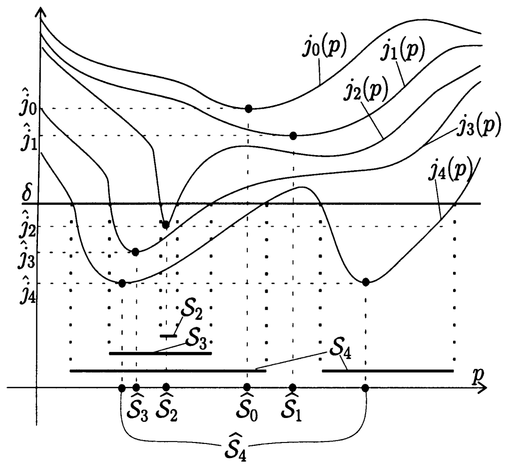

Fig. 1. Whereas $j_q(p)$ and $\hat{j}_q$ are decreasing functions of q, $\mathcal{S}_q(\delta)$ increases with q and δ.

III. Interval Arithmetic and Constraint Propagation

The approach to be employed enters into Apt's chaotic iteration framework [1] or into the Saraswat's model [23]. It combines two complementary tools, namely interval analysis [19] and constraint propagation [17], [26], into what is known as *interval constraint propagation* (ICP) [7], [8]. Note that interval analysis is also used for reliable global optimization without constraint propagation, see, e.g., [9], and [27] in a general context, and [13] and [28] in a minimax context. Reliable global optimization based on ICP is often more efficient [24], [29]. Moreover, ICP can handle subsets of $\mathbb{R}$ (or domains) that may not be intervals. Although such domains are less easily manipulated than intervals, they allow more accurate outer approximations of sets.

A. Interval and Domain Arithmetics

A *domain* X of $\mathbb{R}$ is a subset of $\mathbb{R}$. *Domain arithmetic* is a generalization to domains of the classical arithmetic for real numbers. Let X and Y be two domains, $\oplus$ be an operator in $\{+, -, *, /, \hat{}, \max, \min \dots\}$ and f be a real function such as $\sin, \cos, \tan, \mathrm{sqr}, \mathrm{abs} \dots$ By definition

$$\begin{aligned} X \oplus Y =& \{x \oplus y \mid x \in X, y \in Y\} \\ f(X) =& \{f(x) \mid x \in X\}. \end{aligned} \tag{2}$$

When the domains to be handled are intervals, interval analysis can be used to evaluate these quantities. Here, the domains are assumed to consist of finite unions of intervals and interval computation can be extended to computing with such domains [10]. The main advantages of using domains instead of intervals are that the set of domains is closed with respect to the union operator and that domain computation makes it possible to avoid *hull pessimism* when discontinuous or multivalued functions are involved ([10], [13]). For instance, with interval arithmetic $1/[-1,1] =]-\infty, \infty[$, whereas with domain arithmetic $1/[-1,1] =]-\infty,-1] \cup [1,\infty[$.

We shall call *Cartesian domain* of $\mathbb{R}^n$ the Cartesian product of n domains of $\mathbb{R}$, i.e., $\mathbf{X} = X_1 \times \cdots \times X_n$. The notion of Cartesian domain can be interpreted as an extension of that of axis-aligned box (or interval vector). This extension allows a more accurate outer bounding of compact sets with disconnected parts. Note that an axis-aligned box is a Cartesian product of intervals and, thus, a Cartesian domain. The set of all Cartesian domains of $\mathbb{R}^n$ will be denoted by $\mathcal{D}(\mathbb{R}^n)$. Vector calculus can be extended to Cartesian domains using interval arithmetic [19], and the notion of inclusion function. Consider a function $f : \mathbb{R}^n \to \mathbb{R}$. An *inclusion function* of f is a function $F : \mathcal{D}(\mathbb{R}^n) \to \mathcal{D}(\mathbb{R})$ such that

$$\forall \mathbf{X} \in \mathcal{D}(\mathbb{R}^n), \; f(\mathbf{X}) \subset F(\mathbf{X}) \tag{3}$$

where $f(\mathbf{X}) = \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}\}$. Domain arithmetic makes it possible to compute inclusion functions for a very large class of functions f. The principle is to replace each occurrence of a variable x_i in the expression of f by the corresponding domain and each operator or basic function by its domain counterpart, as defined by (2). The following example illustrates computation on domains and demonstrates the pessimism resulting from multiple occurrences of variables in the expression of f.

Example 1: Consider the function $f(\mathbf{x}) = x_1 + x_1 x_2$ and the domain $\mathbf{X} = X_1 \times X_2$, with $X_1 = [1,2]$ and $X_2 = [-3,-2] \cup [3,4]$. A possible inclusion function for f is $F(\mathbf{X}) = X_1 + X_1 X_2$, which is evaluated as follows:

$$\begin{aligned} F(\mathbf{X}) =& [1,2] + ([1,2] * ([-3,-2] \cup [3,4])) \\ =& [1,2] + ([-6,-2] \cup [3,8]) \\ =& [-5,0] \cup [4,10]. \end{aligned}$$

If $f(\mathbf{x})$ is rewritten as $g(\mathbf{x}) = x_1(1 + x_2)$, a new inclusion function is obtained as $G(\mathbf{X}) = X_1(1 + X_2)$, which is evaluated as follows:

$$\begin{aligned} G(\mathbf{X}) =& [1,2] * ([1,1] + ([-3,-2] \cup [3,4])) \\ =& [1,2] * ([-2,-1] \cup [4,5]) \\ =& [-4,-1] \cup [4,10]. \end{aligned}$$

Note that $f(\mathbf{X}) = G(\mathbf{X}) \subset F(\mathbf{X})$, i.e., $G(\mathbf{X})$ provides the exact image of $\mathbf{X}$ by f whereas $F(\mathbf{X})$ provides only an outer approximation (because the two occurrences of x_1 are treated as if they were independent). ■

With domain computation, inclusion functions can thus be obtained for any function f for which an analytical expression is available. We shall see in Section V-D how to obtain an inclusion function for the cost function $j_q(\mathbf{p})$ defined by (1), with the help of the notion of set polynomials introduced in Section V.

B. Interval Constraint Propagation

Primitive constraints are relations involving up to three real variables that can be written in one of the three following forms:

$$\begin{aligned} &\text{(unary constraint)} \quad & z_1 \in& Z \\ &\text{(binary constraint)} \quad & z_1 =& f(z_2) \\ &\text{(ternary constraint)} \quad & z_1 =& z_2 \oplus z_3 \end{aligned}$$

where Z is a domain of $\mathbb{R}$, f is an elementary function such as $\cos, \sin, \exp, \log, \mathrm{sqr}, \mathrm{sqrt}, \dots$ and $\oplus$ is a binary operator such as $+, -, *, /, \hat{}, \max, \min, \dots$.

A constraint is *and-decomposable* if it can be decomposed into a finite set of primitive constraints related by the Boolean operator *and*. For instance, the constraint $(\max(x_1x_2, x_1 - \log(x_2)))^2 + \exp(x_1) \leq 3$ is *and*-decomposable since it admits the following decomposition into eight primitive constraints:

$$\begin{cases} z_1 = x_1x_2 & z_2 = \log(x_2) \\ z_3 = x_1 - z_2 & z_4 = \max(z_1, z_3) \\ z_5 = \mathrm{sqr}\,(z_4) & z_6 = \exp(x_1) \\ z_7 = z_5 + z_6 & z_7 \in]-\infty, 3]. \end{cases} \tag{4}$$

If f is a function from $\mathbb{R}^n$ to $\mathbb{R}$, $\mathbf{X} \in \mathcal{D}(\mathbb{R}^n)$ and $Y \in \mathcal{D}(\mathbb{R})$, ICP makes it possible to obtain, in a very efficient way, an outer approximation of the set

$$\mathcal{S} = \mathbf{X} \cap f^{-1}(Y) \tag{5}$$

by a Cartesian domain, provided that the constraint $\mathbf{x} \in f^{-1}(Y)$ is *and-decomposable* (see, e.g., [10], [5], and [4]). *Contracting* $\mathbf{X}$ with respect to $\mathcal{S}$ means finding a Cartesian domain $\mathbf{R}$ such that $\mathcal{S} \subset \mathbf{R} \subset \mathbf{X}$.

Example 2: Consider the set $\mathcal{S}$ defined by (5), where $f(\mathbf{x}) \triangleq x_1^2x_2 + x_3x_1$, $\mathbf{X} = [1, 10]^3$ *and* $Y = [-4, 4]$. To contract $\mathbf{X}$, first decompose the constraint $f(\mathbf{x}) \in Y$ into the following set of primitive constraints:

$$\begin{array}{ll} \text{(C1)} & z_1 = x_1^2 \\ \text{(C2)} & z_2 = x_2z_1 \\ \text{(C3)} & z_3 = x_3x_1 \\ \text{(C4)} & y = z_2 + z_3 \\ \text{(C5)} & y \in [-4, 4]. \end{array}$$

The prior domains for the variables x_1, x_2, x_3, z_1, z_2, z_3 and y are taken as $[x_1] = [x_2] = [x_3] = [1, 10]$, $[z_1] = [z_2] = [z_3] = [y] =]-\infty, \infty[$. For simplicity, these domains have been chosen as intervals, but this is not required by the method. By propagating these four constraints as long as contraction takes place, one gets

$$\begin{aligned}
(C5) &\to [y] := [-4, 4] \\
(C1) &\to [z_1] := [z_1] \cap ([x_1])^2 = [1,100] \\
(C2) &\to [z_2] := [z_2] \cap ([x_2] * [z_1]) = [1,1000] \\
(C3) &\to [z_3] := [z_3] \cap ([x_3] * [x_1]) = [1,100] \\
(C4) &\to [y] := [y] \cap ([z_2] + [z_3]) = [2, 4] \\
(C4) &\to [z_2] := [z_2] \cap ([y] - [z_3]) = [1, 3] \\
(C4) &\to [z_3] := [z_3] \cap ([y] - [z_2]) = [1, 3] \\
(C3) &\to [x_1] := [x_1] \cap \left(\frac{[z_3]}{[x_3]}\right) = [1, 3] \\
(C3) &\to [x_3] := [x_3] \cap \left(\frac{[z_3]}{[x_1]}\right) = [1, 3] \\
(C2) &\to [z_1] := [z_1] \cap \left(\frac{[z_2]}{[x_2]}\right) = [1, 3] \\
(C2) &\to [x_2] := [x_2] \cap \left(\frac{[z_2]}{[z_1]}\right) = [1, 3] \\
(C1) &\to [x_1] := [x_1] \cap \sqrt{[z_1]} = [1, \sqrt{3}].
\end{aligned}$$

Thus, $\mathcal{S} = \left\{\mathbf{x} \in [1, 10]^3 \mid x_1^2x_2 + x_3x_1 \in [-4, 4]\right\}$ is included in the box $\left[1, \sqrt{3}\right] \times [1, 3]^2$. ■

TABLE I
CLASSICAL ICP-BASED ALGORITHM FOR RELIABLE MINIMIZATION

	MINIMIZE(in: $[\mathbf{p}_0]$, $j(.)$; out: $[\widehat{j}]$, $\mathcal{S}^+$)
1	$Q = \{[\mathbf{p}_0]\}$; $\mathcal{S}^+ = \emptyset$; $j^+ = \infty$; $[\widehat{j}] = \emptyset$;
2	while $Q \neq \emptyset$,
3	put first element of Q into $[\mathbf{p}]$;
4	$j^+ = \text{GODOWN}(\text{center}([\mathbf{p}]), j(.))$;
5	find a box $[\mathbf{r}]$ such that $[\mathbf{p}] \cap j^{-1}(]-\infty, j^+]) \subset [\mathbf{r}] \subset [\mathbf{p}]$;
6	if $[\mathbf{r}] = \emptyset$, then go to 2;
7	if $(\text{width}([\mathbf{r}]) \leq \varepsilon)$, then $\{\mathcal{S}^+ = \mathcal{S}^+ \cup \{[\mathbf{r}]\}$; go to 2;$\}$
8	bisect $[\mathbf{r}]$ and append the two resulting boxes to Q;
9	end while;
10	remove all $[\mathbf{p}]$ in $\mathcal{S}^+$ such that $\mathrm{lb}([j]([\mathbf{p}])) > j^+$;
11	for all $[\mathbf{p}]$ in $\mathcal{S}^+$, $[\widehat{j}] = [\widehat{j}] \cup [j]([\mathbf{p}])$;
12	$[\widehat{j}] = [\widehat{j}] \cap]-\infty, j^+]$.

When the constraints encountered are not *and-decomposable*, the classical ICP approach, as presented above, does not apply directly. For instance, the constraint $j_q(\mathbf{p}) \leq j^+$ involved in the solution of the robust minimax estimation problem is not *and-decomposable*, as illustrated by the following example.

Example 3: If $j(p) = 1 - \max(p, p^2, \sin(p))$, the constraint $j(p) \leq j^+$ can be expanded as

$$\begin{gathered} (z_1 = p^2) \text{ and } (z_2 = \sin(p)) \text{ and} \\ \{((z_3 = \max(p, z_1)) \text{ and } (z_3 \in]-\infty, j^+])) \\ \text{or } ((z_4 = \max(p, z_2)) \text{ and } (z_4 \in]-\infty, j^+])) \\ \text{or } ((z_5 = \max(z_1, z_2)) \text{ and } (z_5 \in]-\infty, j^+]))\} \end{gathered}$$

and cannot be expressed as a set of primitive constraints related by *and*. ■

An efficient adaptation of ICP to such *and-or-decomposable* problems will be made possible by the introduction of the new notion of set polynomials in Section V. To implement RME, we also need a reliable procedure for global optimization. A trivial adaptation of a classical optimization algorithm [29], [24] will be used. This is presented in the following section. The resulting algorithm is particularly well suited to the use of ICP.

IV. MINIMIZE ALGORITHM

The problem to be solved in this section is the global minimization of a cost function $j(\mathbf{p})$ over a box $[\mathbf{p}_0]$. Let $\widehat{\mathcal{S}}$ be the set of all global minimizers of j over $[\mathbf{p}_0]$ and $[j]([\mathbf{p}])$ be an inclusion function for the cost function $j(\mathbf{p})$. The algorithm MINIMIZE, presented in Table I, computes a list of boxes $\mathcal{S}^+$, the union of which contains $\widehat{\mathcal{S}}$, and an interval containing the global minimum $\widehat{j}$. A local minimization procedure GODOWN, similar to that presented in [12], is used at Step 4 to decrease the upper bound j^+ for the minimum $\hat{j}$. Interval analysis is involved at Steps 10 and 11 to compute an inclusion function $[j]([\mathbf{p}])$ in order to enclose the range of j over the box $[\mathbf{p}]$. By taking advantage of the availability of the upper bound j^+, ICP is involved at Step 5 to replace $[\mathbf{p}]$ by a smaller box $[\mathbf{r}]$ such that any global minimizer in $[\mathbf{p}]$ is also in $[\mathbf{r}]$. Q is a First-In–First-Out list of boxes containing the part of search space that has not yet been studied. Any point of Q is still a potential candidate for being a global minimizer. The real number $\varepsilon > 0$ is the width below which boxes will not be bisected. The interval $[\hat{j}]$ contains the global minimum $\hat{j}$. It is computed by interval evaluation of j over all boxes of $\mathcal{S}^+$ at Step 11. The lower bound of the interval $[j]([\mathbf{p}]$ is denoted by $\mathrm{lb}([j]([\mathbf{p}]))$.

In the context of robust minimax estimation, two problems remain to be solved, namely getting an inclusion function for $j_q(\mathbf{p})$ as defined by (1) and adapting an ICP-based procedure to the contraction of a box $[\mathbf{p}]$ under the constraint $j_q(\mathbf{p}) \leq j^+$, as required by Step 5 of MINIMIZE. These two operations will be based upon of a new type of object presented in Section V.

V. Set Polynomials

The problem considered in this section is the contraction of the Cartesian domain $\mathbf{P}$ associated with the vector $\mathbf{p}$ subject to the constraint $j(\mathbf{p}) \in Y$ when this constraint is *and-or*-decomposable. This problem has to be solved to implement Step 5 of MINIMIZE where the constraint to be taken into account is $j_q(\mathbf{p}) \in]-\infty, j^+]$. The notions developed in this section will also be used to obtain an inclusion function for the cost function $j_q(\mathbf{p})$, as needed at Steps 10 and 11 of MINIMIZE. A procedure to contract $\mathbf{P}$ could in principle consist of three steps.

Step 1) Decompose the constraint $j(\mathbf{p}) \in Y$ into a set of m *and-decomposable* constraints related by the Boolean operator *or*. For instance, the constraint considered in Example 3 is decomposed as follows:

$$1 - \max(p, p^2, \sin(p)) \leq j^+ \Leftrightarrow \begin{cases} \max(p, p^2) \leq j^+ \\ \text{or} \quad \max(p^2, \sin(p)) \leq j^+ \\ \text{or} \quad \max(p, \sin(p)) \leq j^+. \end{cases} \tag{6}$$

Step 2) Contract $\mathbf{P}$ with respect to each of the m constraints taken independently. Thus, m Cartesian domains $\mathbf{P}(1), \ldots, \mathbf{P}(m)$ are obtained.

Step 3) Compute $\mathbf{P}(1) \cup \cdots \cup \mathbf{P}(m)$.

The disjunctive decomposition at Step 1), however, would lead to a combinatorial explosion in our context. To avoid this, the disjunctive form at Step 1) will be replaced by a specific decomposition and the unions at Step 3) will be replaced by a specific set algorithm. The theoretical background needed to understand this adaptation is presented in Sections V-A–V-C, via the notion of set polynomials. In Section V-D, a new procedure is given for contracting $\mathbf{P}$ under the constraint $j_q(\mathbf{p}) \in]-\infty, j^+]$ while avoiding combinatorial explosion.

A. Definitions

Let $\mathcal{P}(\mathbb{R}^n)$ be the set of all subsets of $\mathbb{R}^n$. The set function $\mathbf{F} : \mathcal{P}(\mathbb{R}^n) \times \cdots \times \mathcal{P}(\mathbb{R}^n) \to \mathcal{P}(\mathbb{R}^n)$ is *inclusion monotonic* if

$$\mathbf{X}(1) \subset \mathbf{Y}(1), \ldots, \mathbf{X}(m) \subset \mathbf{Y}(m) \Rightarrow \mathbf{F}(\mathbf{X}(1), \ldots, \mathbf{X}(m)) \subset \mathbf{F}(\mathbf{Y}(1), \ldots, \mathbf{Y}(m)).$$

For instance, $\mathbf{F}(\mathbf{X}(1), \mathbf{X}(2), \mathbf{X}(3)) \triangleq (\mathbf{X}(1) \cap \mathbf{X}(2)) \cup (\mathbf{X}(1) \cap \mathbf{X}(3))$ is inclusion monotonic, contrary to $\mathbf{G}(\mathbf{X}) \triangleq \bar{\mathbf{X}}$, where $\bar{\mathbf{X}}$ is the complementary of $\mathbf{X}$ in $\mathbb{R}^n$. Let $\mathbf{X}(i) \in \mathcal{P}(\mathbb{R}^n)$, $i = 1, \ldots, m$ be m set indeterminates. The construction of the set $\mathcal{B}[\mathbf{X}(1), \ldots, \mathbf{X}(m)]$ of all polynomials in these indeterminates with coefficients in the set of Boolean numbers $\mathcal{B} \triangleq \{0, 1\}$ is in principle illicit, because $(\mathcal{P}(\mathbb{R}^n), \cup, \cap)$ is not a ring but only a semiring, since $(\mathcal{P}(\mathbb{R}^n), \cup)$ is only a monoid and not a group (see [11, p. 116]). By an abuse of notation commonly committed, e.g., in the $(\max, +)$ community [2], we shall nevertheless speak of $\mathcal{B}[\mathbf{X}(1), \ldots, \mathbf{X}(m)]$ as a set of polynomials. One should keep in mind, however, that some classical operations allowed for ring-based polynomials are no longer valid. Any element of $\mathcal{B}[\mathbf{X}(1), \ldots, \mathbf{X}(m)]$ will be called a *set polynomial*. An example of a set polynomial is $(\mathbf{X}(1) \cap \mathbf{X}(2)) \cup (\mathbf{X}(1) \cap \mathbf{X}(1)) \cup \mathbf{X}(2)$, which is an element of $\mathcal{B}[\mathbf{X}(1), \mathbf{X}(2)]$. When there is no ambiguity, $\mathcal{A} \cup \mathcal{B}$ and $\mathcal{A} \cap \mathcal{B}$ will be denoted more concisely by $\mathcal{A} + \mathcal{B}$ and $\mathcal{A}.\mathcal{B}$, respectively. Set polynomials are obviously inclusion monotonic.

To obtain an outer approximation of a set $\mathcal{S} = \mathbf{F}(\mathbf{X}(1), \ldots, \mathbf{X}(m))$ where $\mathbf{F}$ is inclusion monotonic, it suffices to enclose each $\mathbf{X}(k)$ into a box $\mathbf{Y}(k)$ (or more generally a Cartesian domain) and then to compute $\mathbf{F}(\mathbf{Y}(1), \ldots, \mathbf{Y}(m))$. A method for evaluating $\mathbf{F}(\mathbf{Y}(1), \ldots, \mathbf{Y}(m))$ is proposed in the following section.

B. Evaluation Over Cartesian Domains

Lemma 4 (Cartesian Expansion): Let $\mathbf{X}$ and $\mathbf{Y}$ be Cartesian domains. Then

$$(X_1 \times \cdots \times X_n) \cap (Y_1 \times \cdots \times Y_n) = (X_1 \cap Y_1) \times \cdots \times (X_n \cap Y_n) \tag{7}$$

and

$$(X_1 \times \cdots \times X_n) \cup (Y_1 \times \cdots \times Y_n) \subset (X_1 \cup Y_1) \times \cdots \times (X_n \cup Y_n). \tag{8}$$

Lemma 5 (Cartesian Decomposition): If $\mathbf{X}(1), \ldots, \mathbf{X}(m)$ are Cartesian domains of $\mathbb{R}^n$, and $\mathbf{F}$ is a set polynomial, then $\mathbf{F}(\mathbf{X}(1), \ldots, \mathbf{X}(m))$ is a subset of

$$F(X_1(1), \ldots, X_n(1)) \times \cdots \times F(X_1(m), \ldots, X_n(m)).$$

This lemma is a direct consequence of Lemma 4. We shall only give a sketch of its proof on an academic example, where $n = 2$.

Example 6: Assume that

$$\mathbf{F}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \mathbf{X}.\mathbf{Y} + \mathbf{Y}.\mathbf{Z} = (X_1 \times X_2).(Y_1 \times Y_2) + (Y_1 \times Y_2).(Z_1 \times Z_2).$$

(7) then implies that $\mathbf{F}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = ((X_1.Y_1) \times (X_2.Y_2)) + ((Y_1.Z_1) \times (Y_2.Z_2))$ and (8) implies that

$$\mathbf{F}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \subset (X_1.Y_1 + Y_1.Z_1) \times (X_2.Y_2 + Y_2.Z_2) = F(X_1, Y_1, Z_1) \times F(X_2, Y_2, Z_2).$$

■

When $\mathbf{X}(1), \ldots, \mathbf{X}(m)$ are Cartesian domains and $\mathbf{F}$ is a set polynomial, the smallest Cartesian domain containing $\mathbf{F}(\mathbf{X}(1), \ldots, \mathbf{X}(m))$ can be computed exactly by rewriting $\mathbf{F}$ in disjunctive form. However, an exact procedure to evaluate it via the computation of a disjunctive form may become too complex when the number m of sets increases, because this disjunctive form is usually longer than the initial form. This will be avoided by using Lemma 5.

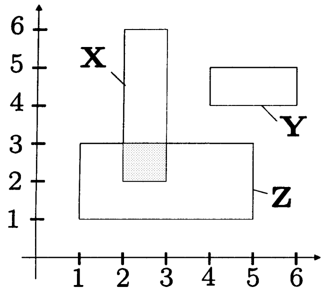

Fig. 2. The grey box is the value of the set polynomial $\mathbf{F}(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ of Example 7.

Example 7: If $\mathbf{F}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = (\mathbf{X} \cap \mathbf{Y}) \cup (\mathbf{Y} \cap \mathbf{Z}) \cup (\mathbf{X} \cap \mathbf{Z})$, for the situation of Fig. 2, $\mathbf{F}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \emptyset \cup \emptyset \cup (\mathbf{X} \cap \mathbf{Z}) = [2,3] \times [2,3]$ is the box in grey. Since $\mathbf{F}(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ is already in disjunctive form (a very special case), the complexity of using the Cartesian decomposition remains the same, but the result is now pessimistic

$$\begin{aligned} & F(X_1, Y_1, Z_1) \times F(X_2, Y_2, Z_2) \\ &= ((X_1 \cap Y_1) \cup (Y_1 \cap Z_1) \cup (X_1 \cap Z_1)) \\ &\quad \times ((X_2 \cap Y_2) \cup (Y_2 \cap Z_2) \cup (X_2 \cap Z_2)) \\ &= (\emptyset \cup [4,5] \cup [2,3]) \times ([4,5] \cup \emptyset \cup [2,3]) \\ &= ([2,3] \cup [4,5]) \times ([2,3] \cup [4,5]). \end{aligned}$$

The resulting Cartesian domain thus consists of four boxes, and is an outer approximation of the actual solution $\mathbf{F}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = [2,3] \times [2,3]$. ■

C. Elementary Set Polynomials

We now focus attention on constraints of the form $j_q(\mathbf{p}) \leq j^+$, as needed by Step 5 of MINIMIZE. We first define symmetric set polynomials and show how they govern the relations between the *and*-decomposable constraints of the robust minimax estimation problem. We then propose a recursive definition of symmetric polynomials, in order to derive an efficient algorithm for their outer evaluation over Cartesian domains.

Definitions: A set polynomial $\mathbf{F}(\mathbf{X}(1), \ldots, \mathbf{X}(m))$ is *symmetric* if it is invariant under permutation. For instance, $\mathbf{X}(1)\mathbf{X}(2) + \mathbf{X}(1)\mathbf{X}(3) + \mathbf{X}(2)\mathbf{X}(3) + \mathbf{X}(1)\mathbf{X}(2)\mathbf{X}(3)$ is symmetric. By analogy with [11, p. 133], the *elementary symmetric set polynomials* are defined as

$$\begin{aligned} \Phi_0(m) &= \prod_{i=1}^{m} \mathbf{X}(i) \\ \Phi_1(m) &= \prod_{i<j}^{m} \mathbf{X}(i) + \mathbf{X}(j) \\ \Phi_2(m) &= \prod_{i<j<k}^{m} \mathbf{X}(i) + \mathbf{X}(j) + \mathbf{X}(k) \\ &\vdots \quad \vdots \quad \vdots \\ \Phi_{m-2}(m) &= \sum_{i<j}^{m} \mathbf{X}(i)\mathbf{X}(j) \\ \Phi_{m-1}(m) &= \sum_{i=1}^{m} \mathbf{X}(i). \end{aligned}$$

By convention, $\Phi_m(m) = \mathbb{R}^n$. We shall call $\Phi_q(m)$ the *q-intersection* of the m sets $\mathbf{X}(k)$, $k \in \{1, \ldots, m\}$, as it is the set of all $\mathbf{x}$'s that belong to at least $m - q$ of these sets. Since

$$j_q(\mathbf{p}) \leq j^+ \Leftrightarrow \begin{cases} \mathbf{p} \in \Phi_q(m) \\ \text{with } \mathbf{X}(i) = f_i^{-1}(]-\infty, j^+]), i \in \{1, \ldots, m\} \end{cases} \tag{9}$$

an enclosure of $\mathcal{S}_q(j^+) \triangleq \{\mathbf{p} \in [\mathbf{p}] \mid j_q(\mathbf{p}) \leq j^+\}$ could be obtained by expanding Φ_q into its disjunctive form. Unfortunately, this expansion gives rise to a combinatorial explosion. For instance, if $m = 10$ and $q = 4$, which corresponds to ten measurements with at most four outliers, Φ_q is the sum of 210 monomials. This combinatorial explosion can be avoided by using the next theorem, which provides a new way to evaluate $\Phi_r(m)$, $0 \leq r \leq q$, efficiently and recursively over m.

Theorem 8: Assume that $\Phi_0(m-1), \ldots, \Phi_q(m-1)$ are available and that a new set $\mathbf{X}(m)$ has to be taken into account. Then, $\Phi_0(m), \ldots, \Phi_q(m)$ can be obtained recursively as follows:

$$\begin{pmatrix} \Phi_0(m) \\ \Phi_1(m) \\ \vdots \\ \Phi_q(m) \end{pmatrix} = \begin{pmatrix} \Phi_0(m-1)\mathbf{X}(m) \\ \Phi_1(m-1)\mathbf{X}(m) \cup \Phi_0(m-1) \\ \vdots \\ \Phi_q(m-1)\mathbf{X}(m) \cup \Phi_{q-1}(m-1) \end{pmatrix} \tag{10}$$

where $\Phi_k(0) = \mathbb{R}^n$ and $\Phi_k(k) = \mathbb{R}^n$ if $k \in \{0, \ldots, q\}$. ■

In a $(\cup, \cap)$ algebra, (10) can be interpreted as a linear discrete-time state equation where the state vector is $\mathbf{\Phi}(m) = (\Phi_0(m) \ \ldots \ \Phi_q(m))^{\mathrm{T}}$ and the input is $\mathbf{X}(m)$.

Proof: The proof is a direct application of Horner's scheme. To avoid introducing a new notation and tedious manipulations of indexes, we shall restrict ourselves to checking (10) for $m = 3$

$$\begin{aligned} \begin{pmatrix} \Phi_0(3) \\ \Phi_1(3) \\ \Phi_2(3) \end{pmatrix} &\triangleq \begin{pmatrix} \mathbf{X}(1)\mathbf{X}(2)\mathbf{X}(3) \\ \mathbf{X}(1)\mathbf{X}(2) + \mathbf{X}(1)\mathbf{X}(3) + \mathbf{X}(2)\mathbf{X}(3) \\ \mathbf{X}(1) + \mathbf{X}(2) + \mathbf{X}(3) \end{pmatrix} \\ &= \begin{pmatrix} (\mathbf{X}(1)\mathbf{X}(2))\mathbf{X}(3) \\ (\mathbf{X}(1) + \mathbf{X}(2))\mathbf{X}(3) + \mathbf{X}(1)\mathbf{X}(2) \\ \mathbb{R}^n\mathbf{X}(3) + (\mathbf{X}(1) + \mathbf{X}(2)) \end{pmatrix} \\ &= \begin{pmatrix} \Phi_0(2)\mathbf{X}(3) \\ \Phi_1(2)\mathbf{X}(3) + \Phi_0(2) \\ \Phi_2(2)\mathbf{X}(3) + \Phi_1(2) \end{pmatrix}. \end{aligned}$$

The following theorem gives three basic properties of the elementary set polynomials.

Theorem 9: The following properties hold true:

i) $\Phi_0(m) \subset \Phi_1(m) \subset \cdots \subset \Phi_q(m)$;

ii)

$$\begin{aligned} \Phi_q(m) \subset \Phi_q(m-1) \subset \cdots \subset \Phi_q(q) &= \Phi_q(q-1) \\ &= \cdots = \Phi_q(0) = \mathbb{R}^n; \end{aligned}$$

iii)

$$\begin{aligned} \mathbf{X}(k) \cap \Phi_q(m) = \emptyset \Rightarrow \Phi_q(m) &= \Phi_{q-1}(\mathbf{X}(1), \ldots, \mathbf{X}(k-1), \\ &\qquad \mathbf{X}(k+1), \ldots, \mathbf{X}(m)). \end{aligned}$$

Proof of i): $\Phi_k(m)$ is the sum of all monomials of the form $\mathbf{X}(i_1)\mathbf{X}(i_2)\mathbf{X}(i_3) \ldots \mathbf{X}(i_{m-k})$. Now, $\mathbf{X}(i_1)\mathbf{X}(i_2)\mathbf{X}(i_3) \ldots \mathbf{X}(i_{m-k-1})$ is a monomial of $\Phi_{k+1}(m)$.

TABLE II
EVALUATION OF AN ENCLOSURE OF THE SET POLYNOMIAL Φ_q OVER m CARTESIAN DOMAINS

```
q-INTERSECT(in: X(1), ..., X(m); out: [Φ_q(m)])
    for k ∈ {0, ..., m}, [Φ_{-1}](k) = ∅;
    for ℓ ∈ {0, ..., q}, [Φ_ℓ](0) = ℝ^n;
    for k = 1 up to m
        for ℓ = 0 up to q
            [Φ_ℓ](k) = ([Φ_ℓ](k − 1) ∩ X(k))
                        ∪ [Φ_{ℓ−1}](k − 1);
```

Thus, any monomial of $\Phi_k(m)$ is included in $\Phi_{k+1}(m)$. Therefore, $\Phi_k(m) \subset \Phi_{k+1}(m)$.

Proof of ii): $\Phi_q(m) = \Phi_q(m-1)\mathbf{X}(m) \cup \Phi_{q-1}(m-1)$. Now, from i), $\Phi_{q-1}(m-1) \subset \Phi_q(m-1)$, so $\Phi_q(m) \subset \Phi_q(m-1)\mathbf{X}(m) \cup \Phi_q(m-1) = \Phi_q(m-1)$.

Proof of iii): Set $\Phi_q^{\overset{k}{}}(m) = \Phi_q(\mathbf{X}(1), \ldots, \mathbf{X}(k-1), \mathbf{X}(k+1), \ldots, \mathbf{X}(m))$. Factor $\Phi_q(m)$ with respect to $\mathbf{X}(k)$ to get $\Phi_q(m) = \Phi_q^{\overset{k}{}}(m)\mathbf{X}(k) \cup \Phi_{q-1}^{\overset{k}{}}(m)$. Intersect both sides of this equation with $\Phi_q(m)$ to get $\Phi_q(m) = \Phi_q^{\overset{k}{}}(m)\mathbf{X}(k)\Phi_q(m) \cup \Phi_{q-1}^{\overset{k}{}}(m)\Phi_q(m)$. Now, by assumption, $\mathbf{X}(k) \cap \Phi_q(m) = \emptyset$. Therefore, $\Phi_q(m) = \emptyset \cup \Phi_{q-1}^{\overset{k}{}}(m)\Phi_q(m) = \Phi_{q-1}^{\overset{k}{}}(m)$. ∎

In the context of bounded-error estimation, Theorem 9 can be interpreted as follows: $\mathbf{p} \in \mathbf{X}(k)$ means that $\mathbf{p}$ is consistent with the kth datum; i) if $\mathbf{p}$ is consistent with at least $m-q$ data, then it is also consistent with at least $m-q-1$ of them; ii) if $\mathbf{p}$ is consistent with at least $m-q$ of the first m data, then it is also consistent with at least $m-q-1$ of the first $m-1$ data; iii) if there exists no $\mathbf{p}$ consistent with at least $m-q$ of the data and with the kth datum, then the kth datum (which is interpreted as an outlier) can be removed from the data set and q can be replaced by $q-1$ to get a simpler definition of $\Phi_q(m)$.

Evaluation Over Cartesian Domains: To implement (10), one should use sets on which unions and intersections can be computed or at least enclosed in computable outer approximations, such as Cartesian domains, boxes, ellipsoids or polytopes. Unfortunately, outer approximation with such sets introduces pessimism and the equality (10) becomes an inclusion. The q-INTERSECTION algorithm (given in Table II) computes a Cartesian domain that encloses the set $\Phi_\ell(\mathbf{X}(1), \ldots, \mathbf{X}(m))$, $\ell \in \{1, \ldots, q\}$, where the sets $\mathbf{X}(k)$ are Cartesian domains. In the computer, $[\Phi_\ell](k)$, $\ell \in \{-1, \ldots, q\}$, $k \in \{0, \ldots, m\}$ is represented by a $((q+2) \times (m+1))$-matrix, the entries of which are Cartesian domains.

This algorithm will now be illustrated on the situation described by Fig. 3, where the Cartesian domains to be considered are boxes. For $q = 0, 1, 2$, q-INTERSECTION, respectively, yields $[\Phi_0]([\mathbf{x}](1), \ldots, [\mathbf{x}](9)) = \emptyset$, $[\Phi_1]([\mathbf{x}](1), \ldots, [\mathbf{x}](9)) = \emptyset$ and $[\Phi_2]([\mathbf{x}](1), \ldots, [\mathbf{x}](9)) \subset [6,7] \times [5,6]$ (represented by the black box in Fig. 3). The results obtained here are exact, but in general they are pessimistic. To limit such a pessimism, a possible improvement based Theorem 9 *iii)* can be used. This improvement remains to be implemented.

D. Application to Robust Minimax Estimation

In this paragraph, we show how set polynomials can be used to contract $[\mathbf{p}]$ under the constraint $j_q(\mathbf{p}) \le j^+$, as required by Step 5 of MINIMIZE in the context of robust minimax estimation, and to obtain an inclusion function for j_q, as required by Steps 10 and 11. The following algorithm contracts $[\mathbf{p}]$, by taking advantage of the fact that

$$[\mathbf{p}] \cap j_q^{-1}(]-\infty, j^+]) = \Phi_q\left([\mathbf{p}] \cap f_1^{-1}(]-\infty, j^+]), \ldots, [\mathbf{p}] \cap f_m^{-1}(]-\infty, j^+])\right)$$

see (9).

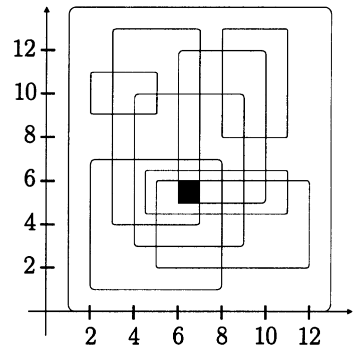

Fig. 3. The black box is the 2-intersection of 9 boxes.

```
CONTRACT (in: [p], j⁺; out: [r]);
for k = 1 to m, compute a Cartesian domain P(k)
    such that ([p] ∩ f_k^{-1}(]−∞, j⁺])) ⊂ P(k) ⊂ [p];
R = q-INTERSECTION(P(1), ..., P(m));
return [r], the smallest box that contains [p] ∩ R.
```

Since the constraint $f_k(\mathbf{p}) \le j^+$ is *and*-decomposable, $\mathbf{P}(k)$ can be obtained using interval constraint propagation as explained in Section III-B. To get an inclusion function for $j_q(\mathbf{p})$, consider the semiring $(\mathbb{R}, \min, \max)$. Denote the qth elementary symmetric polynomial in m indeterminates by $\Phi_q(x_1, \ldots, x_m)$. For instance

$$\begin{aligned}
\Phi_0(x_1, x_2, x_3) &= \max(x_1, x_2, x_3)\\
\Phi_1(x_1, x_2, x_3) &= \min(\max(x_1, x_2), \max(x_2, x_3)\\
&\qquad \max(x_1, x_3))\\
\Phi_2(x_1, x_2, x_3) &= \min(x_1, x_2, x_3).
\end{aligned}$$

The cost function $j_q(\mathbf{p})$ of (1) can then be expressed as $j_q(\mathbf{p}) = j_q(m, \mathbf{p}) = \Phi_q(f_1(\mathbf{p}), \ldots, f_m(\mathbf{p}))$ and (10) can be adapted to this context to compute $j_q(\mathbf{p})$ as

$$\begin{pmatrix} j_0(m, \mathbf{p})\\ j_1(m, \mathbf{p})\\ \vdots\\ j_q(m, \mathbf{p}) \end{pmatrix} = \begin{pmatrix} \max(j_0(m-1, \mathbf{p}), f_m(\mathbf{p}))\\ \min(\max(j_1(m-1, \mathbf{p}), f_m(\mathbf{p})), j_0(m-1, \mathbf{p}))\\ \vdots\\ \min(\max(j_q(m-1, \mathbf{p}), f_m(\mathbf{p})), j_{q-1}(m-1, \mathbf{p})) \end{pmatrix} \tag{11}$$

where $j_\ell(k, \mathbf{p}) = \Phi_\ell(f_1(\mathbf{p}), \ldots, f_k(\mathbf{p}))$, $k \in 1, \ldots, m$; $k > q$, $j_k(0, \mathbf{p}) = -\infty$, and $j_k(k, \mathbf{p}) = -\infty$. An inclusion function

TABLE III
FIRST PART OF THE RESULTS OBTAINED BY THE ROBUST MINIMAX ESTIMATOR FOR THREE DATA SETS WITH 0, 1 AND 2 OUTLIERS, RESPECTIVELY, AND FOR THREE VALUES OF q

data	q	#split	time	$\#\mathcal{S}_q^+$	$[\hat{\jmath}_q]$	$j_q(\mathbf{p}^*)$
$\mathbf{y}^0$	0	323	21	42	$[0.441, 0.458]$	1.067
$\mathbf{y}^0$	1	845	70	18	$[0.371, 0.375]$	0.694
$\mathbf{y}^0$	2	6044	770	21	$[0.198, 0.214]$	0.465
$\mathbf{y}^1$	0	1581	84	230	$[14.05, 14.10]$	33.07
$\mathbf{y}^1$	1	341	28	70	$[0.442, 0.459]$	1.067
$\mathbf{y}^1$	2	906	104	10	$[0.372, 0.375]$	0.694
$\mathbf{y}^2$	0	298	15	70	$[17.60, 17.66]$	33.07
$\mathbf{y}^2$	1	2779	220	246	$[14.05, 14.10]$	29.59
$\mathbf{y}^2$	2	343	42	59	$[0.441, 0.458]$	1.067

for $j_q(\mathbf{p})$, can, thus, be derived from (11) by applying the rules of interval computation and by returning an interval enclosure of $j_q(m, [\mathbf{p}])$.

VI. TEST CASE

Consider a two-exponential model where the relation between the parameter vector $\mathbf{p}$ and the model output is given by

$$y_m(\mathbf{p}, t) = p_1 \exp(-p_2 t) + p_3 \exp(-p_4 t). \tag{12}$$

Since a permutation of p_1 with p_3 and of p_2 with p_4 does not affect the model output y_m, the model is not globally identifiable. Therefore, any reliable identification method should lead to symmetrical solutions, if the search domain is large enough. Ten data points $y(1), \ldots, y(10)$ have been generated as follows. First, a noise-free data vector $\mathbf{y}^*$ was computed. Its ten components were obtained by evaluating $y_m(\mathbf{p}^*, t_k)$ as given by (12) for $\mathbf{p}^* = (20, 0.8, -10, 0.2)^T$ and $t_k = 1/4k^2$, $k \in \{1, \ldots, 10\}$. Noisy data were then obtained by adding to each component y_k^* of $\mathbf{y}^*$ the noise $n_k = (|y_k^*| + 5) * \eta_k$, where η_k is a random noise with a uniform distribution in $[-0.1, 0.1]$. The resulting data vector $\mathbf{y}^0$ is the *regular* data vector. A second data vector $\mathbf{y}^1$ was obtained by replacing y_3 by 30 in $\mathbf{y}^0$, and a third data vector $\mathbf{y}^2$ by replacing y_8 by -30 in $\mathbf{y}^1$. For $\varepsilon = 0.05$ and $[\mathbf{p}_0] = [-40, 40] \times [0, 1] \times [-40, 40] \times [0, 1]$, the results obtained by using RME as described in Section II are summarized in Table III, where $\#\mathcal{S}_q^+$ is the number of boxes of $\mathcal{S}_q^+$ and #split is the number of bisections performed by MINIMIZE. All computing times (in s) are for a Pentium 133MHz. As expected, for a given data vector $\mathbf{y}^i$, $j_q(\mathbf{p}^*)$ and $\hat{\jmath}_q(\mathbf{p})$ are decreasing when q increases. At each run, $\mathcal{S}_q^+$ turns out to consist of two connected components. One of them, denoted by $\mathcal{S}_q^+(1)$, belongs to the half-space where $p_1 > 0$ and the second one $\mathcal{S}_q^+(2)$ belongs to the half-space where $p_1 < 0$). The smallest boxes guaranteed to contain $\mathcal{S}_q^+(1)$ (*i.e.*, the set associated with $\mathbf{p}^*$) are given in Table IV, which evidences the fact that reasonable estimates are obtained only provided that q is equal to or greater than the actual number of outliers q^*.

Various strategies can be thought of for the choice of q, the maximum number of tolerated outliers. A reasonable guideline is to iterate the minimization of $j_q(\mathbf{p})$ over $[\mathbf{p}_0]$ for $q = 0, 1, 2, \ldots$ until the results obtained lead one to believe that q is greater than the actual number of outliers q^*. No systematic procedure exists for the detection of q^*. Nevertheless, as illustrated by the example in this section, if the optimizers are on the boundary of $[\mathbf{p}_0]$ or if the value of $\hat{\jmath}_q$ is too large, it can be suspected that there are at least $q + 1$ outliers. Note that $\hat{\jmath}_q$ corresponds to one of the output errors and some bound is often available on the largest regular error one is prepared to accept. Moreover, if $\hat{\jmath}_q - \hat{\jmath}_{q+1}$ is large then the reliability of the datum y_i that satisfies $\hat{\jmath}_q = |y_i - y_m, i(\hat{\mathbf{p}})|$ for one $\hat{\mathbf{p}} \in \widehat{\mathcal{S}}_q$ is questionable since its presence strongly modifies the estimation results. These comments suggest to take the smallest q such that: i) $\hat{\jmath}_q$ is acceptably small, ii) $\hat{\jmath}_q$ is close to $\hat{\jmath}_{q+1}$, and iii) $\widehat{\mathcal{S}}_q$ is not on the boundary of the search box $[\mathbf{p}_0]$.

TABLE IV
SECOND PART OF THE RESULTS OBTAINED BY THE ROBUST MINIMAX ESTIMATOR FOR THREE DATA SETS WITH 0, 1 AND 2 OUTLIERS, RESPECTIVELY, AND FOR THREE VALUES OF q

data	q	box guaranteed to contain $\mathcal{S}_q^+(1)$
$\mathbf{y}^0$	0	$[18.3, 18.7] \times [0.904, 0.925] \times [-7.63, -7.40] \times [0.160, 0.165]$
$\mathbf{y}^0$	1	$[17.7, 18.0] \times [0.995, 1.000] \times [-6.63, -6.55] \times [0.158, 0.159]$
$\mathbf{y}^0$	2	$[19.1, 19.4] \times [0.992, 1.000] \times [-7.68, -7.57] \times [0.189, 0.192]$
$\mathbf{y}^1$	0	$[39.4, 40.0] \times [0.300, 0.305] \times [-40.0, -39.4] \times [0.988, 1.000]$
$\mathbf{y}^1$	1	$[18.4, 18.7] \times [0.904, 0.926] \times [-7.61, -7.47] \times [0.162, 0.165]$
$\mathbf{y}^1$	2	$[17.7, 18.2] \times [0.992, 1.000] \times [-6.74, -6.56] \times [0.158, 0.161]$
$\mathbf{y}^2$	0	$[39.9, 40.0] \times [0.136, 0.152] \times [-17.1, -16.1] \times [0.001, 0.002]$
$\mathbf{y}^2$	1	$[39.3, 40.0] \times [0.300, 0.305] \times [-40.0, -39.4] \times [0.987, 1.000]$
$\mathbf{y}^2$	2	$[18.3, 18.7] \times [0.903, 0.928] \times [-7.66, -7.45] \times [0.161, 0.166]$

VII. CONCLUSION

When the noise corrupting the data can be assumed to belong to a sequence of random variables that are independently uniformly distributed over the interval $[-\delta, \delta]$, with δ unknown, minimax estimation is a standard approach for the identification of the model parameters, because the resulting estimated parameter vector belongs to the set of all maximum-likelihood estimates for any value of δ such that this set is not empty. Minimax estimation is however seldom used in practice, because of its well-known sensitivity to outliers. The procedure described in this paper makes minimax estimation robust to a prespecified number of data points that can take arbitrary values. It does so in a guaranteed way, by enclosing the set of all such robust minimax estimates in a union of boxes in parameter space, even in the case where the model is nonlinear in its parameters and the estimates are not unique. This has been illustrated by an example. To the best of our knowledge, there is no other method available in the literature to deal with this type of problem.

There are many reasons why combinatorial complexity looms over any attempt at solving such a problem, and several measures were taken to limit it as much as possible. First, it should be stressed that the various combinations of up to q outliers among m data points are not considered in isolation, but collectively. The other measures for limiting complexity while preserving guaranteedness result from the combination of three tools: *interval analysis* which provides guaranteed results, *constraint propagation* to efficiently eliminate large parts of the search space without requiring bisections, and *set polynomials* to deal with Boolean connections between the constraints.

The approach presented in this paper opens up new possibilities of research in this relatively young field, which should also lead to improvements in the algorithms for guaranteed nonlinear estimation in the presence of outliers.

In this paper, we have assumed that the ith error equation $|y_i - y_{m,i}(\mathbf{p})| = j^+$ was decomposable into primitive constraints. Now, when the model output $y_m(\mathbf{p})$ is described by an algorithm containing if statements, or when it is defined as the solution of a set of differential equations, this decomposition is generally impossible. However, the main limitation of the approach is due to the exponential complexity of MINIMIZE with respect to the dimension of $\mathbf{p}$. As a result, only low-dimensional problems can be considered, and an important challenge is to push the complexity barrier as much as possible.

REFERENCES

[1] K. Apt, "The essence of constraint propagation," *Theoret. Comput. Sci.*, vol. 221, no. 1, 1998.

[2] F. Baccelli, G. Cohen, G. J. Olsder, and J. P. Quadrat, *Synchronization and Linearity. An Algebra for Discrete Event Systems*. New York: Wiley, 1992.

[3] E. W. Bai and H. Cho, "Minimization with few violated constraints and its application in set-membership identification," in *Proc. IFAC Word Congr.*, vol. H, Beijing, China, 1999, pp. 343–348.

[4] F. Benhamou and L. Granvilliers, "Automatic generation of numerical redundancies for nonlinear constraint solving," *Rel. Comput.*, vol. 3, no. 3, pp. 335–344, 1997.

[5] F. Benhamou and W. Older, "Applying interval arithmetic to real, integer and boolean constraints," *J. Logic Program.*, pp. 1–24, 1997.

[6] F. L. Chernousko, *State Estimation for Dynamic Systems*. Boca Raton, FL: CRC Press, 1994.

[7] J. G. Cleary, "Logical arithmetic," *Future Comput. Syst.*, vol. 2, no. 2, pp. 125–149, 1987.

[8] E. Davis, "Constraint propagation with interval labels," *Art. Intell.*, vol. 32, no. 3, pp. 281–331, 1987.

[9] E. R. Hansen, *Global Optimization using Interval Analysis*. New York: Marcel Dekker, 1992.

[10] E. Hyvönen, "Constraint reasoning based on interval arithmetic; The tolerance propagation approach," *Art. Intell.*, vol. 58, no. 1-3, pp. 71–112, 1992.

[11] N. Jacobson, *Basic Algebra*. San Francisco, CA: Freeman and Company, 1974.

[12] L. Jaulin, "Interval constraint propagation with application to bounded-error estimation," *Automatica*, vol. 36, no. 10, pp. 1547–1552, 2000.

[13] ——, "Reliable minimax parameter estimation," *Rel. Comput.*, vol. 7, no. 3, pp. 231–246, 2001.

[14] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. London, U.K.: Springer-Verlag, 2001.

[15] A. V. Kurzhanski and I. Valyi, *Ellipsoidal Calculus for Estimation and Control*. Boston, MA: Birkhäuser, 1997.

[16] H. Lahanier, E. Walter, and R. Gomeni, "OMNE: A new robust membership-set estimator for the parameters of nonlinear models," *J. Pharmacokinetics Biopharmaceutics*, vol. 15, pp. 203–219, 1987.

[17] A. K. Mackworth, "Consistency in networks of relations," *Art. Intell.*, vol. 8, no. 1, pp. 99–118, 1977.

[18] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, Eds., *Bounding Approaches to System Identification*. New York: Plenum, 1996.

[19] R. E. Moore, *Methods and Applications of Interval Analysis*. Philadelphia, PA: SIAM, 1979.

[20] "Special issue on bounded-error estimation: Issue 1," *Int. J. Adapt. Control Signal Processing*, vol. 8, no. 1, pp. 1–118, 1994.

[21] "Special issue on bounded-error estimation: Issue 2," *Int. J. Adapt. Control Signal Processing*, vol. 9, no. 1, pp. 1–132, 1995.

[22] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley, 1987.

[23] V. Saraswat, "Concurrent constraint programming," in *Proc. ACM Symp. Principles Programming Languages*, 1990, pp. 232–245.

[24] P. van Hentenryck, Y. Deville, and L. Michel, *Numerica: A Modeling Language for Global Optimization*. Boston, MA: MIT Press, 1997.

[25] "Special issue on parameter identification with error bounds," *Math. Comput. Simulat.*, vol. 32, no. 5-6, pp. 447–607, 1990.

[26] D. L. Waltz, "Generating semantic descriptions from drawings of scenes with shadows," in *The Psychology of Computer Vision*, P. H. Winston, Ed. New York: McGraw-Hill, 1975, pp. 19–91.

[27] M. A. Wolfe, "Interval methods for global optimization," *Appl. Math. Comput.*, vol. 75, pp. 179–206, 1996.

[28] ——, "On discrete minimax problems in the set of real numbers using interval arithmetic," *Rel. Comput.*, vol. 5, no. 4, pp. 371–383, 1999.

[29] J. Zhou, "A permutation-based approach for solving the job-shop problem," *Constraints*, vol. 1, pp. 1–30, 1996.

Luc Jaulin was born in Nevers, France, in 1967. He received the Ph.D. degree in automatic control from the University of Orsay, France, in 1993.

Since 1993, he has been Associate Professor of Physics at the University of Angers, France. His research interests include robust estimation and control using interval methods and constraint propagation.

Eric Walter was born in Saint Mandé, France, in 1950. He received the Doctorat d'État degree in control theory from the University of Paris Sud, France, in 1980.

Currently, he is Directeur de Recherche at CNRS (the French national center for scientific research), and is Director of the Laboratoire des Signaux et Systèmes.. His research interests revolve around parameter estimation and its application to chemical engineering, chemistry, control, image processing, medicine pharmacokinetics, and robotics. He is the author or coauthor of *Identifiability of State-Space Models* (Berlin, Germany: Springer-Verlag,1982), *Identification of Parametric Models from Experimental Data* (London, U.K.: Springer-Verlag, 1997) and *Applied Interval Analysis* (London, U.K.: Springer-Verlag, 2002).