Nonlinear bounded-error parameter estimation using interval computation
— **Source link** ☒

L. Jaulin, E. Walter

Institutions: University of Paris

Related papers:

- Guaranteed nonlinear estimation using constraint propagation on sets

- Guaranteed linear and nonlinear parameter estimation from bounded-error data: A survey

- Guaranteed recursive non-linear state bounding using interval analysis

- Parameter estimation enhanced by optimal input design in a bounded-error context

- Recursive Robust Minimax Estimation

Nonlinear bounded-error parameter estimation using interval computation

Luc Jaulin, Eric Walter

Nonlinear Bounded-Error
Parameter Estimation
Using Interval Computation

L. Jaulin[⋆][◇] and E. Walter[⋆]

[⋆] Laboratoire des Signaux et Systèmes, CNRS-Supélec-Université de Paris-Sud,

Plateau de Moulon, 91192 Gif-sur-Yvette, France

[◇]On leave from Laboratoire d'Ingénierie des Systèmes Automatisés, Université d'Angers,

Faculté des Sciences, 2 boulevard Lavoisier, 49045 Angers, France

Abstract: This paper deals with the estimation of the parameters of a model from experimental data. The aim of the method presented is to characterize the set $\mathbb{S}$ of all values of the parameter vector that are acceptable in the sense that all errors between the experimental data and corresponding model outputs lie between known lower and upper bounds. This corresponds to what is known as bounded error estimation, or membership set estimation. Most of the methods available to give guaranteed estimates of $\mathbb{S}$ rely on the hypothesis that the model output is linear in its parameters, contrary to the method advocated here, which can deal with nonlinear model. This is made possible by the use of the tools of interval analysis, combined with a branch-and-bound algorithm. The purpose of the present paper is to show that the approach can be cast into the more general framework of granular computing.

1   Introduction

In the exact sciences, in engineering, and increasingly in the human sciences too, mathematical models are used to describe, understand, predict or control the behavior of systems. These mathematical models often involve unknown parameters that should be identified (or calibrated, or estimated) from experimental data and prior knowledge or hypotheses (see, *e.g.*, [WP97]). Let $y_k, k \in \{1, \ldots, k_{\max}\}$, be the data collected on the system to be modeled. They form the data vector $\mathbf{y} \in \mathbb{R}^{k_{\max}}$. Denote the vector of the unknown parameters of the model by $\mathbf{p} = (p_1, p_2, \ldots, p_n)^{\mathrm{T}} \in \mathbb{R}^n$, and the corresponding model by

1

$\mathcal{M}(\mathbf{p})$. For any given $\mathbf{p}$, $\mathcal{M}(\mathbf{p})$ generates a vector model output $\mathbf{y}_m(\mathbf{p})$ homogeneous to the data vector $\mathbf{y}$. Estimating $\mathbf{p}$ from $\mathbf{y}$ is one of the basic tasks of statisticians. This is usually done by minimizing some cost function $j(\mathbf{p})$, for instance a norm of $\mathbf{y} - \mathbf{y}_m(\mathbf{p})$. The Euclidean norm is most commonly used, leading to what is known as least-square estimation. It corresponds to maximum-likelihood estimation of $\mathbf{p}$ under the hypothesis that the data points y_k, $k \in \{1, \ldots, k_{\max}\}$, are independently corrupted by an additive Gaussian measurement noise with zero mean and covariance independent of k. Many other cost functions may be considered, depending on the information available on the noise corrupting the data. The minimization of these cost functions usually lead to a point estimate of $\mathbf{p}$, *i.e.*, a single numerical value for each parameter. Except in a few special cases, the minimization of the cost function is difficult and one can seldom guarantee that a global optimizer of the cost function has been found. Moreover, the characterization of the uncertainty on the estimate of $\mathbf{p}$ is usually performed, if at all, by using asymptotic properties of maximum-likelihood estimators, which is not appropriate when the number $k_{\max}$ of data points is very small, as is often the case in biology for example. An attractive alternative is to resort to what is known as *bounded-error estimation* or *set-membership estimation*. In this context (see, *e.g.,* [Wal90], [Nor94], [Nor95], [MNPLW96] and the references therein), a vector $\mathbf{p}$ is feasible if and only if all errors $e_k(\mathbf{p})$ between the data points y_k and the corresponding model outputs $y_{m,k}(\mathbf{p})$ lie between known lower bounds $\underline{e}_k$ and upper bounds $\bar{e}_k$, which express the confidence in the corresponding measurement. Let $\mathbb{S}$ be the set of all values of $\mathbf{p}$ that are feasible, *i.e.*,

$$\mathbb{S} = \{\mathbf{p} \in \mathbb{R}^n \mid \text{for all } k \in \{1, \ldots, k_{\max}\}, \ y_k - y_{m,k}(\mathbf{p}) \in [\underline{e}_k, \bar{e}_k]\} \tag{1}$$

Some methods only look for a value of $\mathbf{p}$ in $\mathbb{S}$, but then the size and shape of $\mathbb{S}$, which provide useful information about the uncertainty on $\mathbf{p}$ that results from the uncertainty in the data, remain unknown. This is why one should rather try to characterize $\mathbb{S}$. When $\mathbf{y}_m(\mathbf{p})$ is linear, $\mathbb{S}$ is a convex polytope, which can be characterized exactly [WPL89]. In the nonlinear case, the situation is far more complicated if one is looking for a *guaranteed* characterization of $\mathbb{S}$. The algorithm SIVIA (for Set Inverter Via Interval Analysis) [JW93a], [JW93c], [JW93b] nevertheless makes it possible to compute guaranteed estimates of $\mathbb{S}$ in many situations of practical interest, by combining a branch-and-bound algorithm with techniques of interval computation. The purpose of this chapter is to present the resulting methodology in the framework of granular computing. The chapter

is organized as follows. In Section 2, the very few notions of interval analysis required to understand SIVIA are recalled. Section 3 presents this algorithm in the context of granular computing. Its application to nonlinear bounded-error estimation is illustrated on a simple example in Section 4.

2 Interval computation

Interval computation was initially developed (see [Moo79]) to quantify the uncertainty of results calculated with a computer using a floating point number representation, by bracketing any real number to be computed between two numbers that could be represented exactly. It has found many others applications, such as global optimization and guaranteed solution of sets of nonlinear equations and/or inequalities [Han92], [HDM97]. Here, we shall only use interval computation to test whether a given box in parameter space is inside or outside $\mathbb{S}$.

An interval $[x] = [\underline{x}, \overline{x}]$ is a bounded compact subset of $\mathbb{R}$. The set of all intervals of $\mathbb{R}$ will be denoted by $\mathbb{IR}$. A vector interval (or box) $[\mathbf{x}]$ of $\mathbb{R}^n$ is the Cartesian product of n intervals. The set of all boxes of $\mathbb{R}^n$ will be denoted by $\mathbb{IR}^n$. The basic operations on intervals are defined as follows

$$
\begin{aligned}
[x] + [y] &= [\underline{x} + \underline{y}, \overline{x} + \overline{y}], \\
[x] - [y] &= [\underline{x} - \overline{y}, \overline{x} - \underline{y}], \\
[x] * [y] &= [\min\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}, \max\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}], \\
1/[y] &= [1/\overline{y}, 1/\underline{y}] \quad (\text{provided that } 0 \notin [y]), \\
[x]/[y] &= [x] * (1/[y]) \quad (0 \notin [y]).
\end{aligned}
$$

All continuous basic functions such as sin, cos, exp, sqr... extend easily to intervals by defining $f([x])$ as $\{f(x)|x \in [x]\}$. As an example, $\sin([0, \pi/2]) * [-1, 3] + [-1, 3] = [0, 1] * [-1, 3] + [-1, 3] = [-1, 3] + [-1, 3] = [-2, 6]$. Note that some properties true in $\mathbb{R}$ are no longer true in $\mathbb{IR}$. For instance, the property $x - x = 0$ translates into $[x] - [x] \ni 0$, and the property $x^2 = x * x$ translates into $[x]^2 \subset [x] * [x]$.

Example 1 *If* $[x] = [-1, 3]$, $[x] - [x] = [-4, 4]$, $[x]^2 = [0, 9]$ *and* $[x] * [x] = [-3, 9]$.

Moore has shown [Moo66] that if a sequence of basic operations on real numbers is replaced

by the same sequence on intervals containing these real numbers, then the interval result contains the corresponding real result. ■

Example 2 *Since* $[0, 2] + ([2, 3] * [4, 5]) - [2, 3] = [0, 2] + [8, 15] - [2, 3] = [8, 17] + [-3, 2] = [5, 19]$, *the result obtained by performing the same computation on real numbers belonging to these intervals is guaranteed to belong to* $[5, 19]$. *For instance* $1 + (2 * 5) - 2 = 9 \in [5, 19]$. ■

For any function f from $\mathbb{R}^n$ to $\mathbb{R}$ which can be evaluated by a succession of elementary operations, it is thus possible to compute an interval that contains the range of f over a box of $\mathbb{IR}^n$. Depending on the formal expression of the function the interval obtained may differ, but it is always guaranteed to contain the actual range. This is illustrated by the following example for $n = 1$.

Example 3 *The function defined by* $f(x) = x^2 - x$ *can equivalently be defined by* $f(x) = (x - 1/2)^2 - 1/4$. *The range of* f *over* $[x] = [-1, 3]$ *can thus be evaluated in the two following ways.*

$$
\begin{aligned}
[x]^2 - [x] &= [-1, 3]^2 - [-1, 3] = [0, 9] + [-3, 1] = [-3, 10], \\
\left([x] - \tfrac{1}{2}\right)^2 - \tfrac{1}{4} &= \left[-\tfrac{3}{2}, \tfrac{5}{2}\right]^2 - \tfrac{1}{4} = \left[0, \tfrac{25}{4}\right] - \tfrac{1}{4} = \left[-\tfrac{1}{4}, 6\right].
\end{aligned}
\tag{2}
$$

The first result is a pessimistic approximation of the range, whereas the second one gives the actual range. This is due to the fact that in the first expression $[x]$ *appears twice, and that the actual value of* x *in the two occurrences are assumed to vary independently within* $[x]$. *It is thus advisable to write the functions in such a way as to minimize the number of occurrences of each variable.* ■

Let f be a function from $\mathbb{R}^n$ to $\mathbb{R}$ and $[a, b]$ an interval. Interval computation will provide *sufficient* conditions to guarantee either that

$$
\forall \mathbf{x} \in [\mathbf{x}] \in \mathbb{IR}^n, \ f(\mathbf{x}) \in [a, b],
\tag{3}
$$

or that

$$
\forall \mathbf{x} \in [\mathbf{x}] \in \mathbb{IR}^n, \ f(\mathbf{x}) \notin [a, b].
\tag{4}
$$

4

Of course, when part of $f([\mathbf{x}])$ belongs to $[a, b]$ and part does not, no conclusion can be reached using these conditions. Although the number of $\mathbf{x}$ in $[\mathbf{x}]$ is not even countable, these conditions will be tested in a finite number of steps. For this purpose, an enclosure $[f]$ of the range $f([\mathbf{x}])$ will be computed using interval computation.

- if $[f] \subset [a, b]$ then all $\mathbf{x}$ in $[\mathbf{x}]$ satisfy (3),

- if $[f] \cap [a, b] = \emptyset$, then no $\mathbf{x}$ in $[\mathbf{x}]$ satisfies (3), or equivalently all of them satisfy (4).

Note that if these two tests take the value *false*, then no conclusion can be drawn.

These basic principles will be used to test whether a given box in parameter space $[\mathbf{p}]$ is either inside or outside $\mathbb{S}$, where $\mathbb{S}$ is given by (1). It suffices to compute an enclosure $[e_k]$ of $y_k - y_{m,k}([\mathbf{p}])$ for all $k \in \{1, \ldots, k_{\max}\}$, using interval computation.

- if for all k in $\{1, \ldots, k_{\max}\}$, $[e_k] \subset [\underline{e}_k, \bar{e}_k]$, then $[\mathbf{p}] \subset \mathbb{S}$.

- if there exists k in $\{1, \ldots, k_{\max}\}$, such that $[e_k] \cap [\underline{e}_k, \bar{e}_k] = \emptyset$, then, $[\mathbf{p}] \cap \mathbb{S} = \emptyset$.

Again, the fact that boxes are considered instead of vectors will allow the exploration of the whole space of interest in a guaranteed way in a finite number of steps, as opposed to Monte-Carlo methods that only sample a finite number of points and can thus not guarantee their results. When no conclusion can be reached for the box $[\mathbf{p}]$, it may be split into subboxes on which the tests will be reiterated, as in the next section.

3 SIVIA

The principle of the *Set Inverter Via Interval Analysis* algorithm is to split the initial problem of characterizing $\mathbb{S}$ into a sequence of more manageable tasks, each of which is solved using interval computation. SIVIA thus pertains to the framework of granular computing.

The parameter vector $\mathbf{p}$ is assumed to belong to some (possibly very large) search domain $[\mathbf{p}_0]$. Let ε be a positive number to be chosen by the user; ε will be called the *level of*

information granularity. A *layer* associated with $\mathbb{S}$ at level ε is a list $\mathbb{G}(\varepsilon)$ of nonoverlapping colored boxes $[\mathbf{p}]$ that cover $[\mathbf{p}_0]$, where[1],

1. $[\mathbf{p}]$ is *black* if it is known that $[\mathbf{p}] \subset \mathbb{S}$,

2. $[\mathbf{p}]$ is *grey* if it is known that $[\mathbf{p}] \cap \mathbb{S} = \emptyset$,

3. $[\mathbf{p}]$ is *white* in all other cases.

Note that some white boxes may actually satisfy either $[\mathbf{p}] \subset \mathbb{S}$ or $[\mathbf{p}] \cap \mathbb{S} = \emptyset$, because of the pessimism of interval computation. Each box of $\mathbb{G}(\varepsilon)$ is called a *granule*. The knowledge required to color it may have been obtained by using the techniques explained in Section 2. Moreover, all white boxes in $\mathbb{G}(\varepsilon)$ are such that their widths are smaller than or equal to ε.

The principle of a procedure to build a layer $\mathbb{G}(\varepsilon_0)$ associated with the set $\mathbb{S}$ is described by the following algorithm. As most granular algorithms, it generates a sequence of layers $\mathbb{G}(\varepsilon)$ indexed by the level of information granularity ε for decreasing values of ε. Starting from a very coarse representation associated with a large ε corresponding to the width of the initial search domain, it stops when ε becomes smaller than or equal to ε_0. In what follows, $w([\mathbf{p}])$ is the *width* of $[\mathbf{p}]$, *i.e.*, the length of its largest side, and splitting a box means cutting it into two subboxes, perpendicularly to its largest side.

---

[1]These colors are, of course, arbitrary. They have been chosen to facilitate the visualization of the boundary of $\mathbb{S}$.

$\text{S{\sc ivia}}(\text{in: } [\mathbf{p}_0], \varepsilon_0; \text{ out: } \mathbb{G}));$

1   $\varepsilon := w([\mathbf{p}_0]);$ Paint $[\mathbf{p}_0]$ white; $\mathbb{G} := \{[\mathbf{p}_0]\};$

2   while $\varepsilon \geq \varepsilon_0$, do

3       $\varepsilon := \varepsilon/2;$

4   while there exists a white $[\mathbf{p}]$ of $\mathbb{G}$ such that $w([\mathbf{p}]) \geqslant \varepsilon;$

5   split $[\mathbf{p}]$ into $[\mathbf{p}_1]$ and $[\mathbf{p}_2];$

6   if attempt to prove that $[\mathbf{p}_1] \subset \mathbb{S}$ is a success, then paint $[\mathbf{p}_1]$ black;

7   if attempt to prove that $[\mathbf{p}_1] \cap \mathbb{S} = \emptyset$ is a success, then paint $[\mathbf{p}_1]$ grey;

8   otherwise, paint $[\mathbf{p}_1]$ white;

9   if attempt to prove that $[\mathbf{p}_2] \subset \mathbb{S}$ is a success, then paint $[\mathbf{p}_2]$ black;

10   if attempt to prove that $[\mathbf{p}_2] \cap \mathbb{S} = \emptyset$ is a success, then paint $[\mathbf{p}_2]$ grey;

11   otherwise, paint $[\mathbf{p}_2]$ white;

12   endwhile;

13 endwhile;

14 return $\mathbb{G}$.

Denote the union of all black boxes by $\mathbb{S}^-$ and the union of all black and white boxes by $\mathbb{S}^+$. At each iteration of the external while loop, one then has

$$\mathbb{S}^- \subset \mathbb{S} \subset \mathbb{S}^+. \tag{5}$$

Example 4 *Consider the set $\mathbb{S}$ described by*

$$\mathbb{S} = \{\mathbf{p} \in \mathbb{R}^2 | p_1^2 + p_2^2 \in [1, 2]\}.$$

For $[\mathbf{p}_0] = [-3, 3] \times [-3, 3]$ and $\varepsilon_0 = 0.04$, S{\sc ivia} generates the sequence of layers described by Figure 1. Each of these layers corresponds to the state of $\mathbb{G}$ at Step 13. The hierarchical properties of these layers are schematically portrayed by the information pyramid of Figure 2.

4   Application to bounded-error estimation

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

$$y_m(\mathbf{p}, t) = \sin(p_1 * (t + p_2)). \tag{6}$$

We choose a two-parameter model to facilitate illustration, but note that the method applies to higher dimensions without modification. This specific example was chosen to show that the methodology advocated in this paper was able to handle unidentifiable models without requiring an identifiability analysis. We shall therefore first present the results obtained with SIVIA on simulated data, before interpreting them in the light of the notion of identifiability.

4.1 Bounded error estimation

Ten simulated measurements were generated as follows. First ten noise-free measurements y_k^* were computed for $k = 1, 2, \ldots, 10$, at a "true" value of the parameter vector $\mathbf{p}^* = (1, 2)^{\mathrm{T}}$. Of course, this true value is not communicated to the estimation procedure. Noisy data y_k were then obtained by adding to y_k^* realizations of a random noise uniformly distributed in $[-1, 1]$. The resulting data are presented in Figure 3, where the vertical bars indicate all possible values of the noise-free data when the $\underline{e}_k$'s are taken equal to -1 and the $\bar{e}_k$'s to 1. The feasible parameter set $\mathbb{S}$ is thus the set of all $\mathbf{p}$'s that satisfy

$$\begin{cases} \sin(p_1 * (1 + p_2)) & \in & [y_1 - 1, y_1 + 1], \\ \sin(p_1 * (2 + p_2)) & \in & [y_2 - 1, y_2 + 1], \\ \vdots & \vdots & \vdots \\ \sin(p_1 * (10 + p_2)) & \in & [y_{10} - 1, y_{10} + 1]. \end{cases} \tag{7}$$

SIVIA was used to characterize the part of $\mathbb{S}$ located in three search domains $[\mathbf{p}_0]$ of increasing size, namely $[0, 2] \times [-\pi, \pi]$, $[0, 4] \times [-2\pi, 2\pi]$ and $[0, 8] \times [-4\pi, 4\pi]$. In all cases, the accuracy parameter ε_0 was taken equal to 0.008. The results, obtained in less than 10 seconds on a Pentium 133 MHz computer, are summarized by Figure 4, where the sizes of the frames are those of the search domains.

4.2 Identifiability analysis

As is readily apparent from Figure 4, $\mathbb{S}$ consists of many disconnected components. Part of this fact may be explained by an identifiability analysis [Wal82]. Roughly speaking, a model is said to be uniquely (or globally) identifiable if there is a single value of the parameter vector associated to any given behavior of the model output. From the equation

8

of the model (6), it is clear that any two vectors $\mathbf{p}$ and $\mathbf{q}$ of $\mathbb{R}^2$ such that $p_1(t + p_2) = q_1(t + q_2) + 2\ell\pi$, $\ell \in \mathbb{Z}$, will lead to exactly the same behavior for all t's. Then any $\mathbf{q}$ such that $q_1 = p_1$ and $q_2 = \frac{p_2 - 2\ell\pi}{p_1}$ will lead to the same behavior as $\mathbf{p}$. The model considered is therefore not uniquely identifiable, and if $\mathbf{p} \in \mathbb{S}$, then all vectors $\mathbf{q}$ of the form

$$\mathbf{q}(\ell) = (p_1, \frac{p_2 - 2\ell\pi}{p_1}), \ell \in \mathbb{Z}, \tag{8}$$

are also in $\mathbb{S}$. This is why all the components of $\mathbb{S}$ are piled up with a pseudo periodicity given by $2\pi/p_1$. Note that this identifiability analysis was not required for the estimation of the parameters by the method described in this chapter. Moreover, this estimation method allow us to obtain all models with similarly acceptable behaviors, and not just those that have *exactly* the same behavior.

5   Conclusions

Bounded-error estimation is an attractive alternative to more conventional approaches to parameter estimation based on a probabilistic description of uncertainty. When bounds are available on the maximum acceptable error between the experimental data and the corresponding model output, it is possible to characterize the set of all acceptable parameter vectors by bracketing it between inner and outer approximations. Most of the results available in the literature require the model output to be linear in the unknown parameters to be estimated and only compute an outer approximation. The method described in this chapter is one of the very few that can be used in the nonlinear case. It provides both inner and outer approximations, which is important because the distance between these approximations is a precious indication about the quality of the description that they provide. We hope to have shown that the resulting methodology falls naturally into the framework of granular computing. The quality of the approximation obtained depends of the level of granularity, and a compromise must of course be struck between accuracy of description and complexity of representation.

Many important points could not be covered in this introductory material and still form the subject of ongoing research. They include the extension of the methodology to the estimation of the state vector of a dynamical system (or to the tracking of time-varying parameters) [KJW98], and the robustification of the estimator against outliers, *i.e.*, against

data points for which the error should be much larger than originally thought, because, *e.g.*, of sensor failure [JWD96], [KJWM99].

References

[Han92]   E. R. Hansen. *Global Optimization using Interval Analysis*. Marcel Dekker, New York, 1992.

[HDM97]   P. Van Hentenryck, Y. Deville, and L. Michel. *Numerica. A Modeling Language for Global Optimization*. MIT Press, 1997.

[JW93a]   L. Jaulin and E. Walter. Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis. *Math. and Comput. in Simulation*, 35:1923–1937, 1993.

[JW93b]   L. Jaulin and E. Walter. Guaranteed nonlinear parameter estimation via interval computations. *Interval Computation*, pages 61–75, 1993.

[JW93c]   L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.

[JWD96]   L. Jaulin, E. Walter, and O. Didrit. Guaranteed robust nonlinear parameter bounding. In *Proc. CESA'96 IMACS Multiconference (Symposium on Modelling, Analysis and Simulation)*, pages 1156–1161, Lille, July 9-12, 1996.

[KJW98]   M. Kieffer, L. Jaulin, and E. Walter. Guaranteed recursive nonlinear state estimation using interval analysis. In *Proc. 37th IEEE Conference on Decision and Control*, pages 3966–3971, Tampa, December 16-18, 1998.

[KJWM99]   M. Kieffer, L. Jaulin, E. Walter, and D. Meizel. Nonlinear identification based on unreliable priors and data, with application to robot localization. In A. Garulli, A. Tesi, and A. Vicino, editors, *Robustness in Identification and Control*, pages 190–203, LNCIS 245, London, 1999. Springer.

[MNPLW96] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter (Eds). *Bounding Approaches to System Identification*. Plenum Press, New York, 1996.

[Moo66]   R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[Moo79]   R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM Publ., Philadelphia, 1979.

[Nor94]   J. P. Norton (Ed.). Special issue on bounded-error estimation: Issue 1. *Int. J. of Adaptive Control and Signal Processing*, 8(1):1–118, 1994.

[Nor95]   J. P. Norton (Ed.). Special issue on bounded-error estimation: Issue 2. *Int. J. of Adaptive Control and Signal Processing*, 9(1):1–132, 1995.

[Wal82]   E. Walter. *Identifiability of state space models*. Springer-Verlag, Berlin, 1982.

[Wal90]   E. Walter (Ed.). Special issue on parameter identifications with error bounds. *Mathematics and Computers in Simulation*, 32(5&6):447–607, 1990.

[WP97]   E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer-Verlag, London, 1997.

[WPL89]   E. Walter and H. Piet-Lahanier. Exact recursive polyhedral description of the feasible parameter set for bounded-error models. *IEEE Transactions on Automatic Control*, 34(8):911–915, 1989.
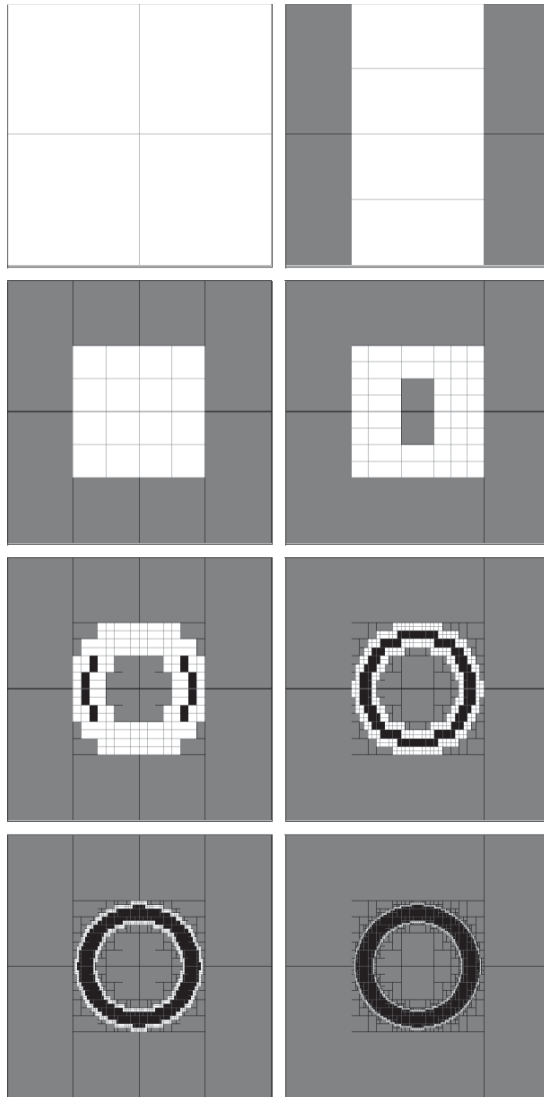
Figure 1: Sequence of layers generated by SIVIA in Example 4;
the black granules are inside $\mathbb{S}$, the grey granules are outside $\mathbb{S}$,
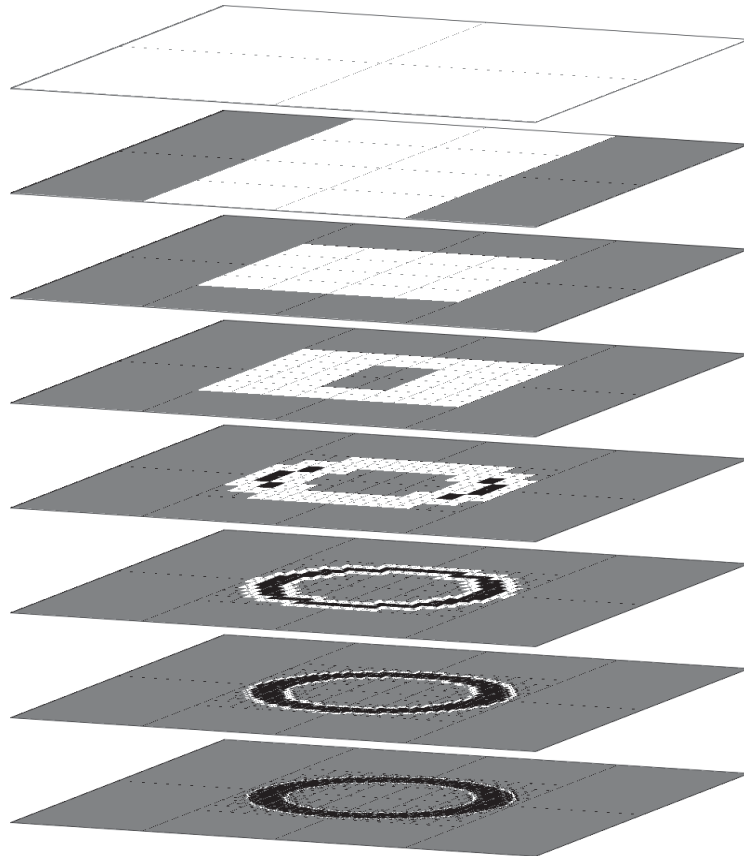nothing is known about the white granules.

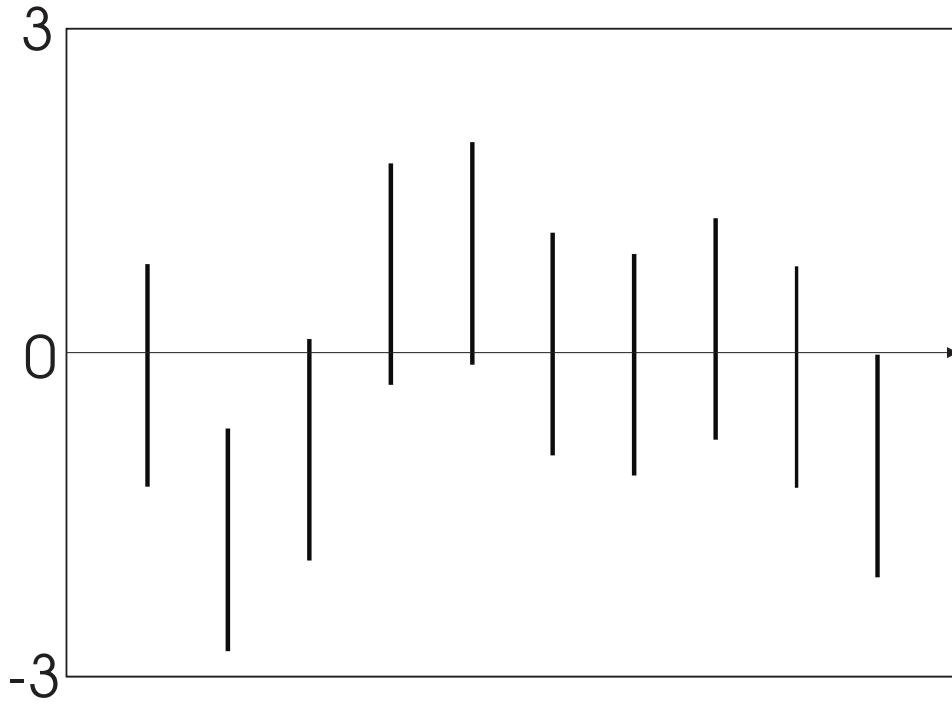Figure 2: Information pyramid associated with Example 4

14

Figure 3: Data bars for the parameter estimation example;
each **p** in $\mathbb{S}$ is such that the corresponding model output crosses all these bars
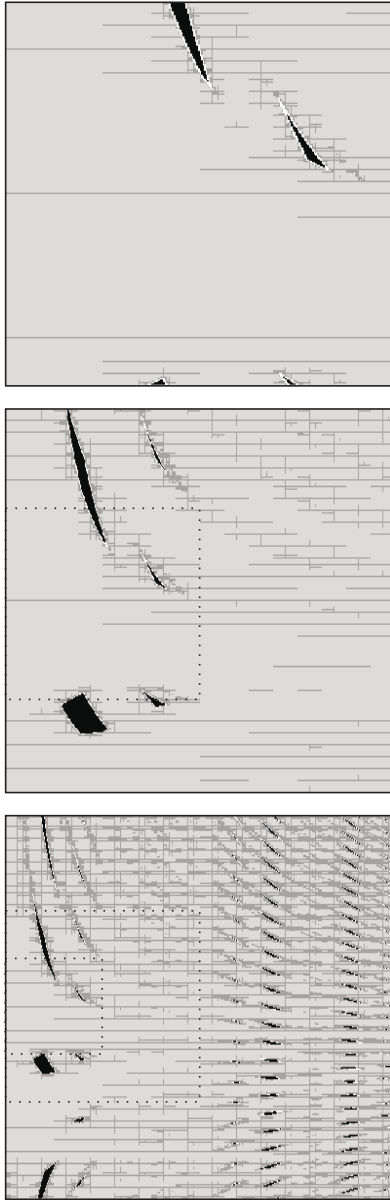
Figure 4: Characterizations of the part of $\mathbb{S}$ in $[\mathbf{p}_0]$
obtained by SIVIA for increasing sizes of $[\mathbf{p}_0]$;
dotted lines in the bottom two subfigures indicate
the correspondence between the frames of the subfigures

16