Guaranteed Parameter Set Estimation for Monotone Dynamical Systems Using Hybrid Automata*

Nacim Meslem
CERTES, Université Paris Est, France

Nacim Ramdani
(corresponding author)
INRIA Sophia Antipolis Méditerranée / LIRMM, 161 rue Ada, 34392 Montpellier Cedex 5, France
nacim.ramdani@lirmm.fr

Yves Candau
CERTES, Université Paris Est, France

Abstract

This paper investigates guaranteed methods for estimating feasible parameter sets when the system under study is modeled with ordinary differential equations (ODE). The issue is to find the set of parameters such that the solution of the ODE remains within specified intervals at known time-data points. These intervals correspond usually to measurement uncertainty.

This is a set inversion problem which can be solved in a guaranteed way by using a partitioning algorithm, interval analysis and validated numerical integration methods for IVPs for ODEs.

In order to address high dimensional nonlinear continuous-time systems, a guaranteed but tight enclosure of the solution of the ODE is needed. Here, we introduce a new approach which is capable of bracketing any uncertain nonlinear monotone dynamical system between an upper and a lower deterministic hybrid dynamical system. The solution enclosures thus obtained are tighter than the ones given by classical validated numerical integration methods. The methodology is illustrated with simulated data from an actual parameter identification setup.

Keywords: ordinary differential equations, continuous-time systems, hybrid systems, monotone systems, nonlinear systems, bounded noise, interval analysis, system identification, initial value problems, Taylor series, stability, wrapping effect

*Submitted: October 23, 2007; Revised: July 21, 2009; Accepted: July 22, 2009.

AMS subject classifications: 65G20, 65L05, 34A38

1 Introduction

The identification of continuous-time systems from sampled data is recognized as an inverse problem of great importance in many fields. In the literature, it is usually solved in a stochastic context via probabilistic methods [1, 2].

In fact, probabilistic methods are relevant only when an explicit characterization of the measurement noise is available. If the data sample is too small for instance, it may be difficult to evaluate the effect of non-exact matching of the assumed statistical assumptions on the errors. Furthermore, deterministic modeling or measurement errors cannot adequately be described by random variables. Therefore it is often more natural to assume the perturbations unknown but belonging to a set with known bounds. In this case, set-membership (SM) (also known as unknown-but-bounded-error) approaches allow the characterization of the whole set of parameter vectors that are compatible with the measured data, a model structure and some prior error bounds. One of the advantages of SM approaches is that they are capable of dealing with problems with non-unique solutions and make the uncertainty in the identified parameter vector directly available from the shape and size of the solution set: the projection of this set onto the parameter axes constitutes uncertainty intervals for the identified parameters.

Model validation plays an important role in system identification. When the latter is performed in the stochastic framework, appropriate model validation criteria must be used a posteriori in order to test model quality and correctness [2]. For instance, it is important to check that model residuals satisfy the statistical hypotheses. To the contrary, set membership approaches address model validation in a self-contained manner. Indeed, the emptyness/non-emptyness of the parameter solution set is a validation criterion (see e.g. [3]). Consequently, any model for which the parameter solution set is empty should be invalidated, hence rejected. Nevertheless, classical approaches such as cross-validation, i.e. checking model output error with new data set can still be used with set-membership estimation.

The design of experiment, i.e. the determination of the experimental conditions and system inputs which should provide the best estimation results, is another important aspect in system identification. Because of the self-contained validation capability of SM approaches, and their capabilities to deal with non unique solutions, the classical design of experiment techniques developed for probabilistic methods [2] can also be used for SM estimation.

SM identification of discrete time models is a rather mature topic in the automatic control community, see for instance [4, 5, 6, 7] and the references therein. To the contrary, the case of continuous-time models still needs attention. Hence, this paper deals with the identification of the model parameter vector $\mathbf{p}$ for nonlinear continuous-time systems described by ordinary differential equations (ODE):

$$\left\{ \begin{array}{l} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) \\ \mathbf{y}(t) = \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) \\ \mathbf{x}(t_0) \in \mathbb{X}_0 \\ \mathbf{p} \in \mathbb{P}_0 \end{array} \right\} \tag{1}$$

where $t \in [t_0, T]$, $\mathbf{f} \in C^{k-1}(\mathbb{D} \times \mathbb{U} \times \mathbb{P}_0)$, $\mathbb{D} \times \mathbb{U} \times \mathbb{P}_0 \subseteq \mathbb{R}^{n+n_u+n_p}$ is an open set; n, n_u, m and n_p are the dimension of respectively the state vector $\mathbf{x}$, the input vector $\mathbf{u}$,

the output vector $\mathbf{y}$ and the parameter vector $\mathbf{p}$. The functions $\mathbf{f} : \mathbb{D} \times \mathbb{U} \times \mathbb{P}_0 \to \mathbb{R}^n$ and $\mathbf{h} : \mathbb{D} \times \mathbb{U} \times \mathbb{P}_0 \to \mathbb{R}^m$ are possibly nonlinear. The initial state $\mathbf{x}_0$ is assumed to belong to a prior known set $\mathbb{X}_0$. We assume that measurements $\mathbf{y}_j$ of the output vector are available at sampling times $t_i \in \{t_1, t_2, \ldots, t_n\}$ in $\mathbf{I} = [t_0, t_{n_T}]$. Note that the sampling interval need not be constant. The measurement noise is a discrete time signal assumed additive and bounded with known bounds. Denote by $\mathbb{E}_j$ a feasible domain for output error at time t_j: the feasible domain for model output at time t_j is then given by

$$\mathbb{Y}_j = \mathbf{y}_j + \mathbb{E}_j \tag{2}$$

In this framework, estimating the parameter vector $\mathbf{p}$ consists of determining the set $\mathbb{S}$ of all acceptable parameters

$$\begin{aligned} \mathbb{S} = \{ \quad & \mathbf{p} \in \mathbb{P}_0 \,|\, (\forall t \in [t_0, t_{n_T}],\, \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}, t)) \\ & \wedge (\forall t_j \in \{t_1, t_2, \ldots, t_N\},\, \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{p}, t_j) \in \mathbb{Y}_j)\} \end{aligned} \tag{3}$$

where the set $\mathbb{P}_0$ is the initial search space for the parameters. The characterization of the solution set $\mathbb{S}$ is a *set inversion problem*; a guaranteed approximation of such a set can be provided by using interval analysis only if an inclusion function for the solution of the ODE in (3) can be computed at prescribed time-data points t_j for a whole set of parameter vectors.

Remark 1 *In practical cases, actual data may contain some spurious measurements, outlier data corrupted by gross errors exceeding the prior bounds, rendering the data inconsistent with the prior hypotheses on model structure and error bounds, hence leading to an empty set for the parameter solution set* $\mathbb{S}$ *[8, 9, 10]. One way to address this case is to relax the inclusion test in (3) by allowing* q *out of* n *empty intersections* ($q \ll n$)*, hence using what is known as a* $q-$*relaxed set inversion [8, 9]. Now, analyzing data coherence prior to estimation is an open issue. However, it is possible to check data coherence after estimation by using the relaxed inclusion test.* □

We have investigated parameter estimation problem with ODEs using set membership approaches in [11, 12]. It has also been investigated in [13, 14, 15]. Since the model is not an explicit function of the parameters to be identified, the main idea retained was to replace it by its numerical evaluation using a guaranteed integration of the ODE. In [11, 12], this integration was performed with an interval Taylor method implemented with mean-value forms and matrix pre-conditioning [16, 17]; the method was successfully used for the identification of the two-dimensional parameter vector of a Lotka–Volterra predator-prey competition model. However, we found that this interval analysis based approach is not efficient in the general case when the dimension the state space $\mathbb{R}^n$ is large. In fact, the enclosures of the numerical evaluation of the solution of the ODE often become very pessimistic and thus useless when the widths of the initial state or the parameter interval vectors are large, or when one proceeds with numerical integration over a long period of time [18, 19]. Such conservatism leads to an extensive use of bisection and thus huge computation time, since it is well known that the computation time is exponential in the dimension of the parameter vector.

In order to circumvent these shortcomings, the theory of quasimonotone dynamical systems, i.e. order-preserving semiflows (see for instance [20, 21] and the references therein) was used in [13, 18] in order to bracket the whole state flow of the uncertain dynamical system between two lower and upper deterministic dynamical systems, i.e. involving no uncertainty. These bracketing methods have been successfully applied to

parameter identification with systems where the dimension of the state vector was as large as 13.

One difficulty in the application of the theory of monotone dynamical systems resides in the construction of the bracketing systems, since there is no rule for this construction in the general case. Hence, the purpose of this paper is to develop a methodology for obtaining the bracketing systems for any uncertain monotone dynamical system submitted to any kind of input. The case of systems described by continuous-time models of large dimension similar to the ones usually encountered within the context of physical parameter identification is also emphasized. The methodology introduced will use the formalism of hybrid dynamical systems, and the derived bounding systems will be modeled as hybrid nonlinear automata.

In section 2, set inversion via interval analysis is reviewed. Validated integration schemes via interval analysis and via the theory of monotone dynamical systems are reviewed in section 3. Section 4 contains the main contribution of this paper, i.e. the bracketing with hybrid automata. Section 5 contains an application to an actual system with simulated data.

2 Set inversion via intervals

Interval analysis was initially developed to account for the quantification errors introduced by the floating point representation of real numbers with computers and was extended to validated numerics [7]. A real interval $[a] = [\underline{a},\ \bar{a}]$ is a connected and closed subset of $\mathbb{R}$. The set of all real intervals of $\mathbb{R}$ is denoted by $\mathbb{IR}$. Real arithmetic operations are extended to intervals. Consider an operator $\circ \in \{+, -, *, \div\}$ and $[a]$ and $[b]$ two intervals. Then:

$$[a] \circ [b] = \{u \circ v \mid u \in [a], v \in [b]\} \tag{4}$$

Consider $\psi : \mathbb{R}^n \longmapsto \mathbb{R}^m$; the range of this function over an interval vector $[\mathbf{a}]$ is given by:

$$\psi([\mathbf{a}]) = \{\psi(\mathbf{u}) \mid \mathbf{u} \in [\mathbf{a}]\} \tag{5}$$

The interval function $[\psi] : \mathbb{IR}^n \longmapsto \mathbb{IR}^m$ is an inclusion function for ψ if

$$\forall [\mathbf{a}] \in \mathbb{IR}^n,\ \psi([\mathbf{a}]) \subseteq [\psi]([\mathbf{a}]) \tag{6}$$

An inclusion function for ψ can be obtained by replacing each occurrence of a real variable by the corresponding interval and each standard function by its interval counterpart: the resulting function is called the natural inclusion function. The performance, i.e. the tightness of the enclosure provided by this inclusion function depends on the formal expression for ψ.

The computation of the reciprocal image of a set can be computed by the SIVIA algorithm, *Set Inversion Via Interval Analysis* [7]. As soon as the enclosures $[\mathbf{x}_j]$ of the solution of the ODE in (1) at time t_j are available, SIVIA will allow the characterization of the solution set $\mathbb{S}$ defined by (3), by computing two sets $\underline{\mathbb{S}}$ and $\overline{\mathbb{S}}$ such as:

$$\underline{\mathbb{S}} \subseteq \mathbb{S} \subseteq \overline{\mathbb{S}} \tag{7}$$

The set $\underline{\mathbb{S}}$ contains all the boxes proved to be feasible. To prove that a box $[\mathbf{p}]$ is feasible, one should verify that

$$\forall t_j \in \{t_1, t_2, \ldots, t_N\}, \quad \mathbf{h}([\mathbf{x}_j], \mathbf{u}_j, [\mathbf{p}], t_j) \subseteq \mathbb{Y}_j \tag{8}$$

and to prove that a box $[\mathbf{p}]$ is unfeasible, one should prove that

$$\exists t_j \in \{t_1, t_2, \ldots, t_N\}, \quad \mathbf{h}([\mathbf{x}_j], \mathbf{u}_j, [\mathbf{p}], t_j) \cap \mathbb{Y}j = \emptyset \tag{9}$$

The recursive algorithm SIVIA partitions the prior space $\mathbb{P}_0$ into boxes $[\mathbf{p}]$ to be submitted to these tests. Any undetermined box is bisected and tested again, unless its size is less than a precision parameter ϵ to be tuned by the user, which ensures that the algorithm terminates after a finite number of iterations. The outer approximation is then computed as $\overline{\mathbb{S}} = \underline{\mathbb{S}} \cup \Delta\mathbb{S}$, where $\Delta\mathbb{S}$ is the union of all remaining undetermined boxes.

Obtaining enclosures $[\mathbf{x}_j]$, or in other words performing a guaranteed numerical integration of (1), is thus an essential part of the method, for which two approaches can be used: interval Taylor models, and the theory of monotone systems. These methods are to be reviewed in the next section.

3 Guaranteed set integration

3.1 Guaranteed set integration using interval Taylor models

Consider the following differential equation:

$$\begin{array}{c} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}, t), \\ \mathbf{x}(t_0) \in \mathbb{X}_0 \subseteq [\mathbf{x}_0] \subseteq \mathbb{D}, \\ \mathbf{u} \in \mathbb{U},\ \mathbf{p} \in \mathbb{P}_0 \subseteq [\mathbf{p}] \end{array} \tag{10}$$

with $t_0 \geq 0$. The function $\mathbf{f}$, possibly nonlinear, is assumed to be at least $k-$times continuously differentiable in a domain $\mathbb{D} \subseteq \mathbb{R}^n$. The objective is to compute interval vectors $[\mathbf{x}_j]$, $j = 1, \ldots, n_T$, that are guaranteed to contain the solution of (10) at $t_1, t_2, \ldots, t_{n_T}$.

Effective methods for solving such problems are based on Taylor expansions, see [22, 23] and the references therein. These methods are usually one-step methods which proceed with two phases:

1. they first verify existence and uniqueness of the solution using a fixed point theorem and the Picard-Lindelöf operator [22, 24, 25], compute an a priori enclosure $[\tilde{\mathbf{x}}_j]$ such that $\mathbf{x}(t) \in [\tilde{\mathbf{x}}_j]$ for all $t \in [t_j, t_{j+1}]$ and adapt the integration step size h_j if necessary in order to keep the relative width of the solution smaller than a given threshold;
2. they then compute a tighter enclosure $[\mathbf{x}_{j+1}]$ of the solution for (10) at t_{j+1} as

$$\begin{array}{rl} [\mathbf{x}_{j+1}] = [\mathbf{x}_j] & + \sum_{i=1}^{k-1} h_j^i \mathbf{f}^{[i]}([\mathbf{x}_j], [\mathbf{u}_j], [\mathbf{p}], t_j) \\ & + h_j^k \mathbf{f}^{[k]}([\tilde{\mathbf{x}}_j], [\tilde{\mathbf{u}}_j], [\mathbf{p}], [t_j, t_{j+1}]), \end{array} \tag{11}$$

which corresponds to a Taylor expansion of order k, where $[\tilde{\mathbf{x}}_j]$ is used to compute the remainder term. The coefficients $\mathbf{f}^{[i]}$ are the Taylor coefficients of the solution $\mathbf{x}(t)$, and can be computed either numerically by automatic differentiation or analytically via formal methods. In (11), the interval vector $[\tilde{\mathbf{u}}_j]$ is such that $\mathbf{u}(t) \in [\tilde{\mathbf{u}}_j]$ for all $t \in [t_j, t_{j+1}]$.

The enclosures thus obtained are said to be *validated*. This is in contrast with conventional numerical integration techniques, which derive approximations with unknown global error, and where the accumulation of both truncation and round-off errors may cause the computed solution to deviate widely from the real one. When using interval Taylor models, it is then possible to control the global truncation error, since it is directly connected to the width of the solution enclosure.

Unfortunately, the *wrapping* effect, i.e. the overestimation due to the bracketing of a set of any shape by an axis-aligned box makes the explicit scheme (11) width-increasing and thus not suitable for numerical implementation. To remove this drawback, one usually uses mean value forms, matrix preconditioning and linear transformations [22, 16, 17, 26, 27, 28, 23, 25, 29, 30].

A good compromise between CPU time required and achieved accuracy can be obtained when the order of the Taylor expansion series is taken between 15 and 20 [31].

In [27, 29], a Taylor series expansion with respect to initial state has also been used in order to curb the pessimism introduced by wrapping effect. An implicit interval method has been introduced in [32], where a tighter enclosure of the true solution is computed via a Taylor expansion model, where the desired enclosure appears implicitly. A more general scheme has been developed in [31, 23, 25], where the interval method is founded on the Hermite-Obreshkoff expansion series where the sought enclosure appears both implicitly and explicitly. In [33], an alternative technique has been introduced where constraint propagation techniques are used in connection with a guaranteed relaxation of the ODE in order to build a pruning step.

In practice, apart for some particular cases such as affine uncertain stable systems, the above techniques derive useful enclosures only if the ODE under study involves no uncertain variable. Indeed, when the widths of the initial state or the parameter interval vectors are large, or when one proceeds with numerical integration over a long period of time, the enclosure $[\mathbf{x}_{j+1}]$ usually becomes very pessimistic and thus useless, notwithstanding all the techniques used to circumvent the wrapping effect in interval computations. In the next subsection, we indicate how to solve this problem when the system under study is a monotone dynamical system.

3.2 Guaranteed set integration with the theory of monotone dynamical systems

A monotone dynamical system is just a dynamical system on an ordered metric space which has the property that ordered initial states lead to ordered subsequent states. The theory of monotone dynamical systems has mainly been developed by Hirsch after the seminal work of Müller, Kamke and Krasnoselskij (see [34, 20, 21] and the references therein). The application of monotone methods and comparison arguments in differential equations started in the early 1920s.

Monotone dynamical systems is a broad class of dynamical systems, to which many chemical, physical and economic models belong, and is one of the most important classes of systems in theoretical biology (see [21, 35, 36], and the references therein). Furthermore, it is not difficult to find a change of orthants which transforms a monotone dynamical system into a cooperative one [35].

Property 1 *If a system of ODE is cooperative, a property to be defined later, then the dynamical system is monotone, and it is possible to compute an inclusion function for the solution of the ODE.*

Definition 1 *The dynamical system is* cooperative *over $\mathbb{D}$, if all the off-diagonal terms of its Jacobian matrix are non negative over $\mathbb{D}$, i.e.*

$$\forall i \neq j,\ t \geq 0,\ \forall \mathbf{p} \in [\mathbf{p}],\ \forall \mathbf{x} \in \mathbb{D},\ \forall \mathbf{u} \in \mathbb{U},\ \frac{\partial f_i(\mathbf{x}, \mathbf{p}, \mathbf{u}, t)}{\partial x_j} \geq 0. \tag{12}$$

Theorem 1 *[20, 13]: Let us consider two cooperative systems*

$$\dot{\overline{\mathbf{z}}}(t) = \overline{\mathbf{f}}(\overline{\mathbf{z}}, \overline{\mathbf{p}}, \underline{\mathbf{p}}, \mathbf{u}, t) \tag{13}$$

$$\dot{\underline{\mathbf{z}}}(t) = \underline{\mathbf{f}}(\underline{\mathbf{z}}, \overline{\mathbf{p}}, \underline{\mathbf{p}}, \mathbf{u}, t) \tag{14}$$

which satisfy the condition

$$\begin{array}{l} \forall \mathbf{p} \in [\underline{\mathbf{p}}, \overline{\mathbf{p}}],\ \forall \mathbf{x} \in \mathbb{D},\ \mathbf{u} \in \mathbb{U},\ \forall t \geq t_0, \\ \underline{\mathbf{f}}(\underline{\mathbf{z}}, \overline{\mathbf{p}}, \underline{\mathbf{p}}, \mathbf{u}, t) \leq \mathbf{f}(\mathbf{x}, \mathbf{p}, \mathbf{u}, t) \leq \overline{\mathbf{f}}(\overline{\mathbf{z}}, \overline{\mathbf{p}}, \underline{\mathbf{p}}, \mathbf{u}, t).Z \end{array} \tag{15}$$

Moreover, if there exists two initial conditions such that

$$\forall \mathbf{p} \in [\underline{\mathbf{p}}, \overline{\mathbf{p}}],\ \underline{\mathbf{z}}(\underline{\mathbf{p}}, \overline{\mathbf{p}}, t_0) \leq \mathbf{x}(\mathbf{p}, t_0) \leq \overline{\mathbf{z}}(\underline{\mathbf{p}}, \overline{\mathbf{p}}, t_0), \tag{16}$$

then the solution of (10) satisfies

$$\forall t \geq t_0,\ [\mathbf{x}(t)] \subseteq [\underline{\mathbf{z}}(t), \overline{\mathbf{z}}(t)]. \tag{17}$$

Now, the bracketing systems (13)–(14) involve *degenerate* intervals only, so interval Taylor models can be used for the guaranteed numerical evaluation of $\underline{\mathbf{z}}(t)$ and $\overline{\mathbf{z}}(t)$, since the *wrapping effect* in interval computations can be efficiently controlled by the methods we reviewed in the previous section.

The main difficulty is to obtain suitable bracketing systems in the general case. However, when the components of $\mathbf{f}$, $f_i(\mathbf{x}, \mathbf{p}, \mathbf{u}, t)$ $(i = 1, ..., n)$ are monotonic with respect to each parameter p_k, it is quite easy to derive upper and lower bounds for $f_i(\mathbf{x}, \mathbf{p}, \mathbf{u}, t)$ and to define the bracketing systems [37], while avoiding possible divergence that may occur when both upper and lower bounds of a parameter appear simultaneously in the same expression of the components of the bracketing systems [18].

Rule 1 Use of monotonicity property*: Here we adapt the idea introduced in [37]. Define $\overline{\delta}^i(p_k)$ as follows.*

$$\overline{\delta}^i(p_k) = \begin{cases} \overline{p}_k & \text{if} \quad \frac{\partial f_i}{\partial p_k} \geq 0, \\ \underline{p}_k & \text{if} \quad \frac{\partial f_i}{\partial p_k} < 0, \end{cases} \tag{18}$$

and $\overline{\delta}^i(\mathbf{p}) = [\overline{\delta}^i(p_1), ..., \overline{\delta}^i(p_k), ...]^T$. In a similar way, define $\underline{\delta}^i(p_k)$ as follows.

$$\underline{\delta}^i(p_k) = \begin{cases} \underline{p}_k & \text{if} \quad \frac{\partial f_i}{\partial p_k} \geq 0, \\ \overline{p}_k & \text{if} \quad \frac{\partial f_i}{\partial p_k} < 0, \end{cases} \tag{19}$$

and $\underline{\delta}^i(\mathbf{p}) = [\underline{\delta}^i(p_1), ..., \underline{\delta}^i(p_k), ...]^T$.

If system (10) is cooperative over $\mathbb{D}$ then the components of the enclosing systems (13)–(14) can be obtained as follows.

$$\begin{array}{ll} \dot{\overline{z}}_i(t) = f_i(\overline{\mathbf{z}}, \overline{\delta}^i(\mathbf{p}), \mathbf{u}, t); & \overline{z}_i(t_0) = \overline{x}_{0,i}, \\ \dot{\underline{z}}_i(t) = f_i(\underline{\mathbf{z}}, \underline{\delta}^i(\mathbf{p}), \mathbf{u}, t); & \underline{z}_i(t_0) = \underline{x}_{0,i}. \end{array} \tag{20}$$

4 Bracketing uncertain monotone dynamical systems with hybrid automata

In this section, we address the case of uncertain monotone dynamical systems, for which the signs of the partial derivatives $\partial f_i/\partial p_k$ change along the integration time interval $[t_0, t_{n_T}]$. In such cases, the uncertain system (10) admits an enclosure over each time interval where the functions f_i are monotonic with respect to variables p_k. Therefore both upper and lower bounding systems are defined by piecewise nonlinear ODEs, and can thus be regarded as hybrid dynamical systems. They can be modeled by a *hybrid automaton*, where the hybrid state encompasses both a *discrete time* component and *continuous time* state variables associated with it [38]. The hybrid automaton which will model the systems which bracket (10) is defined by :

$$H = (\mathbb{Q}, \mathbb{E}, \mathbb{D}, \mathbb{U}, \mathbb{F}, \mathbb{T}, \mathbb{R}), \tag{21}$$

where:

1. $\mathbb{Q}$ is a finite set of the discrete components of the hybrid states, called "modes" or "locations." To each location corresponds two continuous-time systems which provide the locally upper and lower solutions of (10). These systems are built using the monotonicity property, i.e. rule 1 and hence equation (20).
2. $\mathbb{E} \subseteq \mathbb{Q} \times \mathbb{Q}$ is the set of the transitions. It contains all the possible commutations between the locally upper (resp. lower) continuous systems which brackets (10).
3. $\mathbb{D}$ is the state space of (10).
4. $\mathbb{U}$ represents the definition domain for the input of (10).
5. $\mathbb{F} = \overline{\mathbb{F}} \cup \underline{\mathbb{F}}$ where $\overline{\mathbb{F}} = \{\overline{\mathbf{f}}_q, q \in \mathbb{Q}\}$ and $\underline{\mathbb{F}} = \{\underline{\mathbf{f}}_q, q \in \mathbb{Q}\}$ are the collections of the field vectors defined by the upper and lower systems which enclose locally the state flow generated by (10).

$$\forall q \in \mathbb{Q}, \quad \overline{\mathbf{f}}_q \quad : \quad \mathbb{D} \times \mathbb{U} \longrightarrow \mathbb{R}^n \tag{22}$$

$$\forall q \in \mathbb{Q}, \quad \underline{\mathbf{f}}_q \quad : \quad \mathbb{D} \times \mathbb{U} \longrightarrow \mathbb{R}^n \tag{23}$$

6. $\mathbb{T} = \{t_e, e \in \mathbb{E}\}$ is the collection of switching time instants.
   Define $g_{i,k}(.) = \frac{\partial f_i}{\partial p_k}(.)$. The set $\mathbb{T}$ is defined as

$$\mathbb{T} = \left\{ \begin{array}{l} t_e \in [t_0, t_{n_T}] \mid \\ \exists\, k = 1, ..., n_p,\ \exists i = 1, ..., n, \\ \exists \mathbf{p} \in [\mathbf{p}] \,|\, g_{i,k}(\mathbf{x}, \mathbf{u}, \mathbf{p}, t_e) = 0 \end{array} \right\}. \tag{24}$$

   That is to say, if the monotonicity of $\mathbf{f}$ with respect to one of the parameters changes at t_e, a transition $e = (q, q') \in \mathbb{E}$ occurs and the bracketing systems change too from $\{\overline{\mathbf{f}}_q, \underline{\mathbf{f}}_q\}$ to $\{\overline{\mathbf{f}}_{q'}, \underline{\mathbf{f}}_{q'}\}$.
7. $\mathbb{R} = \{\mathbf{R}_e, e \in \mathbb{E}\}$ is the collection of reset functions. They initialize the field vectors $\overline{\mathbf{f}}_{q'}$ (resp.$\underline{\mathbf{f}}_{q'}$) after the activation of a transition $e = (q, q')$: $\overline{\mathbf{x}}_{q'}(t_0) = \mathbf{R}_e(\overline{\mathbf{x}}_q(t_e))$ and $\underline{\mathbf{x}}_{q'}(t_0) = \mathbf{R}_e(\underline{\mathbf{x}}_q(t_e))$.

Now, in order to build $\{\overline{\mathbf{f}}_q$ and $\underline{\mathbf{f}}_q\}$ using rule 1 and hence (20), we will split the experiment time period $[t_0, t_{n_T}]$ into a succession of integration time intervals $[t_j, t_{j+1}]$, where $t_{j+1} = t_j + h_j$ and where integration time steps h_j are either chosen a priori or adapted on-line.

Denote by $\mathbb{I}_M$, the set of time intervals $[t_j,\, t_{j+1}]$ over which no switching occurs, i.e., all the components of the field vectors $\mathbf{f}$ of (10) are monotonic with respect to each parameter:

$$\mathbb{I}_M = \{[t_j,\, t_{j+1}] \subset [0, t_{n_T}] \,|\, \forall e \in \mathbb{E},\, t_e \notin [t_j,\, t_{j+1}]\}\,. \tag{25}$$

Since the a priori solution $[\tilde{\mathbf{x}}_j]$ encloses the whole state trajectory over $[t_j,\, t_{j+1}]$, an inner approximation of the set (25) can also be defined without loss of guarantee as follows

$$\underline{\mathbb{I}_M} = \left\{ \begin{array}{l} [t_j,\, t_{j+1}] \subset [0, t_{n_T}] \,| \\ \forall i = 1, ..., n,\, \forall k = 1, ..., n_p, \\ 0 \notin [g]_{i,k}([\tilde{\mathbf{x}}_j], \mathbf{u}, [\mathbf{p}], [t_j, t_{j+1}]) \end{array} \right\}. \tag{26}$$

Similarly, define the set $\mathbb{I}_S$ of intervals where a switching occurs, i.e.,

$$\mathbb{I}_S = \{[t_j,\, t_{j+1}] \subset [0, t_{n_T}] \,|\, \exists e \in \mathbb{E},\, t_e \in [t_j,\, t_{j+1}]\}\,. \tag{27}$$

Since we have

$$[t_0, t_{n_T}] = \mathbb{I}_M \cup \mathbb{I}_S, \tag{28}$$

we can write without loss of guarantee

$$\overline{\mathbb{I}_S} = [t_0,\, T] \setminus \underline{\mathbb{I}_M}. \tag{29}$$

Now, we can use rule (1) and (20) over each time interval $[I_m] \in \mathbb{I}_M$ in order to derive $\underline{\mathbf{f}}_m$ and $\overline{\mathbf{f}}_m$ and to bracket all the possible solutions of the uncertain system (10):

$$\begin{array}{c} \forall [I_m] \in \mathbb{I}_M,\, \forall m \in \mathbb{Q},\, \forall \mathbf{p} \in [\mathbf{p}],\, \forall \mathbf{x} \in \mathbb{D},\, \forall \mathbf{u} \in \mathbb{U}, \\ \forall t \in [I_m],\;\; \underline{\mathbf{f}}_m(\underline{\mathbf{x}}, \underline{\mathbf{p}}, \overline{\mathbf{p}}, \mathbf{u}, t) \leq \mathbf{f}(\mathbf{x}, \mathbf{p}, \mathbf{u}, t) \leq \overline{\mathbf{f}}_m(\overline{\mathbf{x}}, \underline{\mathbf{p}}, \overline{\mathbf{p}}, \mathbf{u}, t), \end{array} \tag{30}$$

where $\underline{\mathbf{f}}_m \in \underline{\mathbb{F}}$ and $\overline{\mathbf{f}}_m \in \overline{\mathbb{F}}$.

One difficulty remains, since the actual time instant, i.e., t_e in (24), when the upper (*resp.* lower) hybrid system reaches one of its switching time instant is unknown a priori. By using a validated interval Taylor model integration method, we will be able to solve this problem in an efficient and guaranteed way. It suffices to apply a validated integration method over each time interval $[I_s] \in \mathbb{I}_S$ to the system (10) in order to cross the switching time instant. By doing so, we keep the guarantee property for the enclosures without having to derive the actual time instant where the commutation occurs. Note that since the widths of the intervals $[I_s]$ are equal to an integration time step (i.e., h_j) the *wrapping* effect will be very small in the validated integration method. Eventually, the time intervals $[I_s]$ might also be reduced using root solving algorithms, for instance [39]. By doing so, one reduces the size of the time step where switching occurs, and hence ensures that the actual length of the step size h_j has minor impact onto the quality of the state trajectory.

Example 1 (Illustrative example) *Let's consider the scalar dynamical system with two uncertain parameters:* p_1 *and* p_2

$$\left\{ \begin{array}{l} \dot{x}(t) = f(x, p_1, p_2, t), \\ \quad x_0 \in [\underline{x}_0, \overline{x}_0], \\ p_i \in [\underline{p}_i, \overline{p}_i]\;\; i = 1, 2. \end{array} \right. \tag{31}$$

Figure 1 depicts a possible time history for $g_1 = \frac{\partial f}{\partial p_1}$ *and* $g_2 = \frac{\partial f}{\partial p_2}$. *There are 4 cases where the signs of the two partial derivatives can be determined with no ambiguity : during these time periods bounding systems are, as time goes forward:*

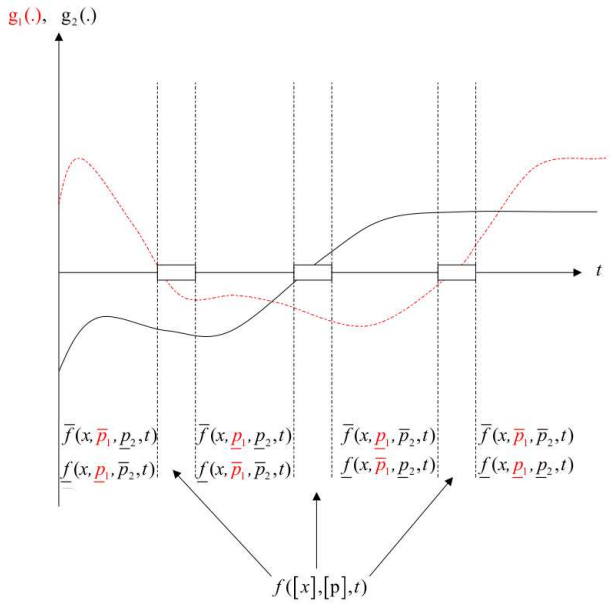

Figure 1: Time history for the partial derivative g_i of example (31).

- $(\underline{f}(x, \underline{p}_1, \overline{p}_2, t), \overline{f}(x, \overline{p}_1, \underline{p}_2, t))$;
- $(\underline{f}(x, \overline{p}_1, \overline{p}_2, t), \overline{f}(x, \underline{p}_1, \underline{p}_2, t))$;
- $(\underline{f}(x, \overline{p}_1, \underline{p}_2, t), \overline{f}(x, \underline{p}_1, \overline{p}_2, t))$;
- $(\underline{f}(x, \underline{p}_1, \underline{p}_2, t), \overline{f}(x, \overline{p}_1, \overline{p}_2, t))$.

To the contrary, there are time intervals where the signs of the partial derivatives cannot be ascertained. During these time intervals we will handle the uncertain *system (31) via interval Taylor models, i.e. we will perform numerical integration with intervals of significant widths. As a conclusion, the hybrid automaton which characterizes the bounding systems will contain 1+4 modes:* mode *0 refers to the use of interval Taylor models,* mode *1 to 4 refer to the 4 cases where bounding systems can be tuned for (31). The automaton driving the upper bracketing system is depicted in Figure 2. The one driving the lower bracketing system can be obtained in the same way.* □

In the sequel, *mode* 0 will denote the original *uncertain* dynamical system and *modes* $q \neq 0$ will denote the case where the bounding systems method is used.

Finally, the methodology used for computing the upper (resp. lower) bracketing system is summarized in the following algorithm:

Algorithm Hybrid-Bound-Solution
$(in : t_0, t_{n_T}, \mathbf{f}, \mathbb{F}, [\mathbf{x}](t_0), [\mathbf{p}]$;
$(out : \mathbf{Bnd}([\mathbf{x}](t_1), \ldots, \mathbf{Bnd}([\mathbf{x}](t_n)$

1. $t := t_0$;
2. $q :=$ **Initialize**$(\mathbf{f}, [\mathbf{x}](t_0), [\mathbf{p}])$;

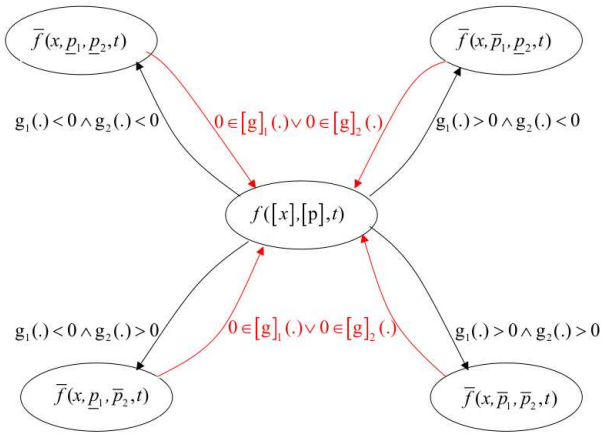

Figure 2: The automaton used for computing the upper bracketing hybrid system for (31).

3. $\mathbf{f}_q :=$ **Select-ODE**(**Bnd**$(\mathbb{F}), q)$;
4. *while* $(t < t_{n_T})$ *do*
5. $\{h, [\mathbf{x}](t+h), [\tilde{\mathbf{x}}(t)]\} :=$ **Integrate**$(\mathbf{f}_q,$ **Bnd**$([\mathbf{x}](t)], t)$;
6. $\{transition, q'\} :=$ **Guard-Cond**$([\tilde{\mathbf{x}}(t)], \mathbf{f}_q)$;
7. *if* $(transition)$ *then*
8. $q := q'$;
9. $[\mathbf{x}](t+h) :=$ **Interval-Integrate**$(\mathbf{f},$ **Bnd**$([\mathbf{x}](t)), [\mathbf{p}], t, h)$;
10. $\mathbf{f}_q :=$ **Select-ODE**$(\mathbb{F}, q)$;
11. *endif*
12. $t := t + h$;
13. end

The upper and lower solutions of (1) can be computed by the same algorithm **Hybrid-Bound-Solution**. It suffices to set algorithm **Bnd** to return either the upper or the lower bound of the state vector, accordingly.

Algorithm **Hybrid-Bound-Solution** finds the initial discrete mode q at line 2 (algorithm **Initialize**), then selects the ODE $\mathbf{f}_q$ which corresponds to this initial discrete state at line 3 (algorithm **Select-ODE**). The latter only implements the monotonicity property as detailed in rule 1. In the *while* loop, the ODE $\mathbf{f}_q$ is integrated until a transition occurs, which is detected at line 6 by algorithm **Guard-Cond**. If this is the case (boolean *transition* is *true*), algorithm **Guard-Cond** also returns the new discrete state q'. In order to cross the guard condition with guarantee, one integration step over $[I_s] = [t, t+h]$ is performed for the original uncertain ODE $\mathbf{f}$ with a full interval validated method (algorithm **Interval-Integrate** at line 9).

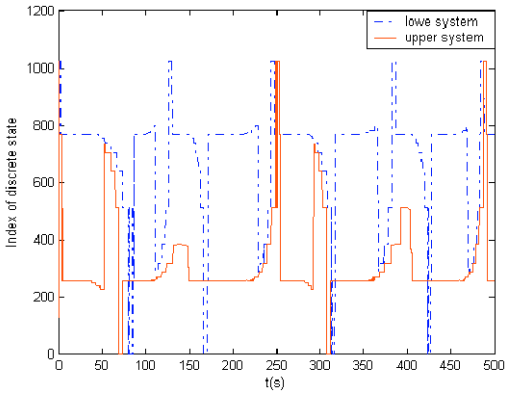

Figure 3: Discrete mode evolution for the automata modeling the bounding systems of (32), with $\mathbb{X}_{0_i} = [9, 11]^o C$ and $\mathbb{P}_0 = [0.73, 1.23]s^{-1} \times [0.23, 0.64]mW^{-1}K$. (continuous line: upper system, dash-dot: lower system)

5 An Application

The problem under investigation is the simultaneous identification of thermo-physical properties of material samples, taken from [18]. The state vector $\mathbf{x} \in \mathbb{R}^{13}$ stands for the node temperatures and the state equation. It is as follows:

$$\begin{cases} \dot{x}_1(t) &= \alpha_1(x_2 - 2x_1 + u_1), \\ \dot{x}_2(t) &= 2\alpha_1(x_1 - (1 + \frac{\rho_1}{\rho_2})x_2 + \frac{\rho_1}{\rho_2}x_3), \\ \dot{x}_3(t) &= 2p_1(x_4 - x_3 + p_2\frac{\delta_2}{\rho_2}(x_2 - x_3)), \\ \dot{x}_i(t) &= p_1(x_{i+1} - 2x_i + x_{i-1}) \quad i = 4, \ldots, 9, \\ \dot{x}_{10}(t) &= 2p_1(x_9 - x_{10} + p_2\frac{\delta_2}{\rho_2}(x_{11} - x_{10})), \\ \dot{x}_{11}(t) &= 2\alpha_2(x_{12} - (1 + \frac{\rho_3}{\rho_2})x_{11} + \frac{\rho_3}{\rho_2}x_{10}), \\ \dot{x}_{12}(t) &= \alpha_3(x_{13} - 2x_{12} + x_{11}), \\ \dot{x}_{13}(t) &= 2\alpha_3(x_{12} - (1 + \frac{\rho_3}{\rho_4})x_{13} + \frac{\rho_3}{\rho_4}u_2), \end{cases} \tag{32}$$

where the unknown parameter vector to be identified is $\mathbf{p} = [p_1\ p_2]^T$. The system input u_1 is taken as a multi-harmonic signal with mean value $T_0 = 10^\circ C$ and $u_2 = T_0$. The model output is taken as $y(\mathbf{p}, t) = x_{12}(t)$.

5.1 Direct image : Solution enclosure

Using full interval Taylor models. Here we test the three integration schemes (explicit and implicit Taylor and IHO, with order $k = 11$) with a varying integration step size strategy as implemented in the VNODE package [31]. As expected, the enclosure of the solution of the ODE becomes very pessimistic (it diverges, actually) as soon as the width of the box used for the parameter vector is greater than 0.0025. Nevertheless, for this limiting box, the use of validated integration schemes is successful. However, if used within SIVIA, set inversion would only be achieved after an unreasonable computation time.

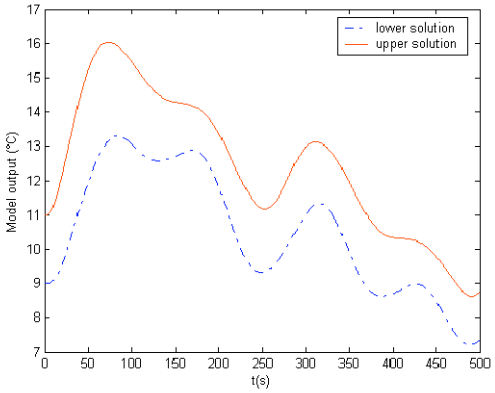

Figure 4: Bounding solutions for the output of model (32). (continuous line: upper solution, dash-dot: lower solution). CPU time = 67.8h (PIV 2.4 GHz)

Using the hybrid bounding method. There are 10 guard conditions, since parameters p_1 and p_2 appear in f_3 and f_{10}, and parameter p_1 appears in f_i, $i = 4 \ldots 9$. Therefore, the set $\mathbb{Q}$ of discrete modes contains 2^{10} elements, but not all of them may be activated. Figure 3 shows the evolution of the discrete modes as obtained for the upper and lower hybrid automata used for bracketing the solutions for the output temperatures of (32) as generated by the algorithms **Hybrid-Bound-Solution**, when both initial state vector and parameter vector are taken uncertain with large uncertainty. Note that the numerical integration of the bounding systems is also achieved using VNODE package. Figure 4 shows the evolution of the upper and lower solutions for the original continuous time model output. Obviously, even for very large parameter boxes, the hybrid bracketing method does not diverge. In Figure 5, the solution enclosures as obtained with the hybrid bounding method, and by IHO as

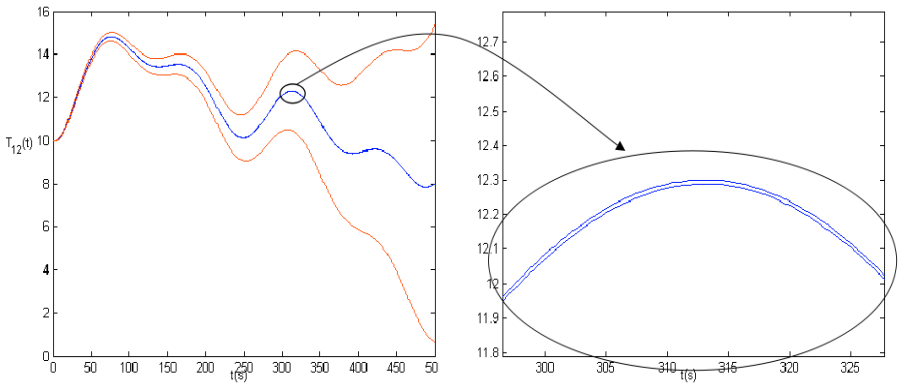

Figure 5: Solution enclosure for the output of (32). Blue curves obtained with the hybrid method; Red curves obtained with interval Hermite–Obreshkoff method.

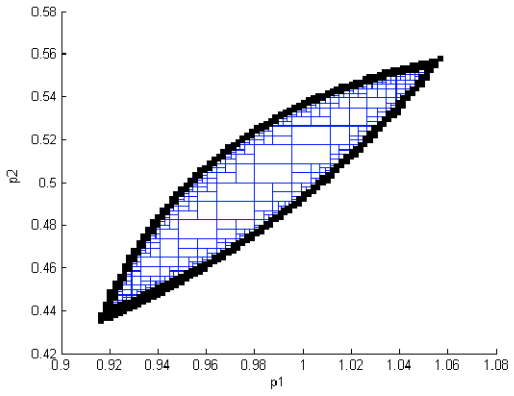

Figure 6: Inner approximation (white boxes) of the solution set (3) with model (32). Uncertainty layer $\mathbf{\Delta}\mathbb{S} = \overline{\mathbb{S}} \setminus \underline{\mathbb{S}}$ (black boxes). CPU time = 24h (PIV 2.4 GHz)

given by VNODE package, are plotted when the size of the parameter vector interval corresponds to the case where full interval Taylor models derive effective results. It is clear that the size of the enclosures derived with the hybrid bounding approach are tighter than the ones obtained by full interval Taylor models.

5.2 Set inversion with hybrid bounding

The initial state vector is taken as a point vector. Pseudo-actual data are constructed by adding to model simulation a random uniform noise. The prior feasible domain for the model output error is taken as $\mathbb{E}_i = [-0.05, 0.05]^o C$. The sampling time step is taken constant $\Delta t = 5s$ and the duration of the experiment is 500s. The initial search space for parameter vector is taken to be $\mathbb{P}_0 = [0.5, 1.5]s^{-1} \times [0.25, 1]mW^{-1}K$. Used with SIVIA, the hybrid bracketing approach derives the inner and outer approximations of the solution set $\mathbb{S}$ as defined in (3). This set is plotted in Figure 6.

6 Conclusion

This paper investigates guaranteed methods for estimating feasible parameter sets when the system under study is modeled with ODE. The issue is to find the set of parameters such that the solution of the ODE remains within specified intervals at known time-data points. When the system under study is a monotone dynamical system, we introduce an effective method for computing enclosures for the solution of the ODE which are tighter than the one usually obtained even with efficient validated numerical integration software. The bracketing systems are modeled using hybrid automata. Used within a set inversion algorithm, it was successfully tested for parameter set estimation on a problem of fairly high dimension. In the future, the approach will be tested on monotone systems of larger dimension. The case of uncertain non linear hybrid dynamic systems will also be addressed.

References

[1] P. Eykhoff. *System Identification*. John Wiley and Sons, New York, 1979.

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

[3] I. Braems, N. Ramdani, M. Kieffer, L. Jaulin, E. Walter, and Y. Candau. Guaranteed characterization of thermal conductivity and diffusivity in presence of model uncertainty. *Inverse Problems in Science and Engineering*, 15(8):895–910, 2007.

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

[5] J. P. Norton editor. Special issue on bounded-error estimation. *Int. J. Adaptive Control Signal Processing*, 9(1):1–132, 1995.

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

[7] 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.

[8] M. Kieffer, L. Jaulin, E. Walter, and D. Meizel. Robust autonomous robot localization using interval analysis. *Reliable Computing*, 6:337–362, 2000.

[9] L. Jaulin. Robust set-membership state estimation; application to underwater robotics. *Automatica*, 45(1):202–206, 2009.

[10] N. Ramdani and P. Poignet. Robust dynamic experimental identification of robots with set membership uncertainty. *IEEE/ASME Transactions on Mechatronics*, 10(2):253–256, 2005.

[11] T. Raïssi, N. Ramdani, and Y. Candau. Parameter estimation for non linear continuous-time systems in a bounded error context. In *Proceedings 42th IEEE CDC*, pages 2240–2245, 2003.

[12] T. Raïssi, N. Ramdani, and Y. Candau. Set membership state and parameter estimation for systems described by nonlinear differential equations. *Automatica*, 40(10):1771–1777, 2004.

[13] E. Walter and M. Kieffer. Interval analysis for guaranteed nonlinear parameter estimation. In *Proceedings 13th IFAC SYSID*, pages 259–270, Rotterdam, 2003.

[14] E. Walter and M. Kieffer. Guaranteed nonlinear parameter estimation in knowledge-based models. *Journal of Computational and Applied Mathematics*, 199:277–285, 2007.

[15] T. Johnson and W. Tucker. Rigorous parameter reconstruction for differential equations with noisy data. *Automatica*, 44:2422–2426, 2008.

[16] R. J. Lohner. Enclosing the solutions of ordinary initial and boundary value problems. In Wiley, editor, *Computer Arithmetic: Scientific Computation and Programming Languages*, pages 255–286, Stuttgart, 1987.

[17] R. Rihm. Interval methods for initial value problems in ODEs. In Elsevier Studies in Computational Mathematics, editor, *Validated computations: proceedings of the IMACS-GAMM International Workshop on Validated Computations*, pages 173–208, Amsterdam, 1994. J. Herzberger.

[18] N. Ramdani, N. Meslem, T. Raïssi, and Y. Candau. Set-membership identification of continuous-time systems. In *Proceedings 14th IFAC Symposium on System Identification*, pages 446–451, Newcastle, Aus, 2006.

[19] T. Raïssi, N. Ramdani, and Y. Candau. Robust nonlinear continuous-time state estimation using interval Taylor models. In *Proceedings 6th IFAC Symposium on Robust Control*, Toulouse, 2006.

[20] H.L. Smith. *Monotone dynamical systems: An introduction to the theory of competitive and cooperative systems*, volume 41 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, 1995.

[21] M. Hirsch and H.L. Smith. Monotone dynamical systems. In A. Canada, P. Drabek, and A. Fonda, editors, *Handbook of Differential Equations, Ordinary Differential Equations*, volume 2, chapter 4. Elsevier, 2005.

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

[23] N. S. Nedialkov and K. R. Jackson. An interval Hermite–Obreschkoff method for computing rigorous bounds on the solution of an initial value problem for an ordinary differential equation. *Reliable Computing*, 5:289–310, 1999.

[24] G.F. Corliss and R. Rihm. Validating an a priori enclosure using high-order Taylor series. In G. Alefeld and A. Formmer, editors, *Scientific computing, Computer Arithmetic, and Validated Numerics*, pages 228–238. Akademie Velag, Berlin, 1996.

[25] N.S. Nedialkov, K. Jackson, and J. Pryce. An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE. *Reliable Computing*, 7(6):449–465, 2001.

[26] W. Kühn. Rigorously computed orbits of dynamical systems of ordinary differential equations. *Computing*, 61:47–67, 1998.

[27] M. Berz and K. Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order taylor models. *Reliable Computing*, 4:361–369, 1998.

[28] N.S. Nedialkov, K.R. Jackson, and G.F. Corliss. validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation*, 105:21–68, 1999.

[29] N. Revol, K. Makino, and M. Berz. Taylor models and floating point arithmetic: proof that arithmetic operations are bounded in COSY. *The Journal of Logic and Algebraic Programming*, 64:135–154, 2005.

[30] Y. Lin and M. A. Stadtherr. Validated solutions of initial value problems for parametric ODEs. *Applied Numerical Mathematics*, 0:000–000, 2007.

[31] N.S. Nedialkov. *Computing rigorous bounds on the solution of an initial value problem for an ordinary differential equation*. PhD University of Toronto, 1999.

[32] R. Rihm. Implicit methods for enclosing solutions of ODEs. *Journal of Universal Computer Science*, 42(6):202–209, 1998.

[33] M. Janssen, P.V. Hentenryck, and Y. Deville. A constraint satisfaction approach for enclosing solutions to parametric ordinary differential equations. *SIAM J. Numerical Analysis*, 40:1896–1939, 2002.

[34] J.-L. Gouzé and K.P. Hadeler. Monotone flows and order intervals. *Nonlinear World*, 1:23–34, 1994.

[35] D. Angeli and E.D Sontag. Monotone control systems. *IEEE Transaction on Automatic Control*, 48(10):1684–1698, 2003.

[36] E.D. Sontag. Molecular systems biology and control. *European Journal of Control*, 11:396–435, 2005.

[37] M. Kieffer and E. Walter. Guaranteed nonlinear state estimation for continuous-time dynamical models from discrete-time measurements. In *Proceedings 6th IFAC Symposium on Robust Control*, Toulouse, 2006.

[38] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

[39] N.S. Nedialkov and M. von Mohrenschildt. Rigorous simulation of hybrid dynamic systems with symbolic and interval methods. In *Proceedings of the American Control Conference*, volume 1, pages 140–147, 2002.