

Reachability of uncertain nonlinear systems using a nonlinear hybridization

Nacim Ramdani¹, Nacim Meslem² and Yves Candau²

¹ INRIA Sophia-Antipolis Méditerranée and LIRMM UMR 5506 CNRS UM2,
161 rue Ada, 34392 Montpellier Cedex 5, France

² CERTES, Univ. Paris 12-Val de Marne, 61 av. Gl. de Gaulle, 94000 Créteil, France

Abstract. In this paper, we investigate nonlinear reachability computation in presence of model uncertainty, via guaranteed set integration. We show how this can be done by using the classical Müller's existence theorem. The core idea developed is to no longer deal with whole sets but to derive instead two nonlinear dynamical systems which involve no model uncertainty and which bracket in a guaranteed way the space reachable by the original uncertain system. We give a rule for building the bracketing systems. In the general case, the bracketing systems obtained are only piecewise C^k -continuously differential nonlinear systems and hence can naturally be modeled with hybrid automata. We show how to derive the hybrid model and how to address mode switching. An example is given with a biological process.

1 Introduction

Computing reachable sets for hybrid systems is an important step when one addresses verification or synthesis tasks. A key issue then lays in the calculation of the reachable space for continuous dynamics with nonlinear models. In this paper, we will also emphasize the presence of parameter uncertainty in the nonlinear dynamical models used for characterizing the continuous dynamics.

Consider an *uncertain* dynamical system described by non-autonomous differential equations with the following form:

$$\{ \dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{p}, t), \quad \mathbf{x}(t_0) \in \mathcal{X}_0 \subseteq \mathcal{D}, \quad \mathbf{p} \in \mathcal{P} \} \quad (1)$$

where function $f : \mathcal{D} \times \mathcal{P} \times \mathbb{R}^+ \mapsto \mathbb{R}^n$ is possibly nonlinear, $\mathcal{D} \subseteq \mathbb{R}^n$, \mathcal{X}_0 is the initial domain for state vector \mathbf{x} at time $t_0 \geq 0$ and \mathcal{P} is an uncertainty domain for parameter vector \mathbf{p} . The reachable space of system (1) is then defined as follows

$$\mathcal{R}([t_0, t]; \mathcal{X}_0) = \left\{ \begin{array}{l} \mathbf{x}(\tau), t_0 \leq \tau \leq t \\ (\dot{\mathbf{x}}(\tau) = f(\mathbf{x}, \mathbf{p}, \tau)) \wedge (\mathbf{x}(t_0) \in \mathcal{X}_0) \wedge (\mathbf{p} \in \mathcal{P}) \end{array} \right\} \quad (2)$$

Several methods have been developed recently for the explicit computation of the reachable space, however, most of them do not address the presence of model parameter uncertainty. When the continuous dynamics are linear, these

methods compute over-approximations of the reachable sets by combining time discretization, numerical integration and computational geometry. They use various representations for the reachable sets such as polytopes [1, 3, 6], zonotopes [8] or ellipsoids [5, 13]. Some other methods proceed with hybrid abstractions [9, 7, 14]. When the continuous dynamics are modelled with a nonlinear differential equation, the computation of the reachable set becomes much harder which forms one of the main obstacle in safety verification of hybrid systems [14]. Most computational methods rely on an hybridization of the continuous-time models, i.e. the use of piecewise simpler, possibly affine approximations of the analysed system on cells defined on the state space [2]. Unfortunately, these reachability computations are tractable only for systems where the dimension of the continuous state component is small.

Few authors investigated the computation of reachable set by using guaranteed set integration. In [10], interval Taylor models [16] were used for the verification of hybrid systems, but no parameter uncertainty were considered. They were also used for the simulation of uncertain hybrid systems where the dimension of vectors were small [17]. Nevertheless, it is well-known that in general the size of the reachable space derived with interval Taylor models diverges after few computation steps when the size of initial state domain or parameter uncertainty domain are large. This shortcoming is mainly caused by the *wrapping* effect, i.e. the overestimation of the solution due to the bracketing of any set by an axis-aligned box.

Hence, the contribution of this paper is to show how one can address nonlinear continuous reachability computation in presence of model uncertainty, in a more efficient way by using the classical Müller's theorem [15, 18, 12] allied with interval Taylor models. We will recall the classical Müller's existence theorem and we will indicate how it can be used for guaranteed set integration and hence reachability computation.

The core idea developed in the sequel is to no longer perform set integration with whole domains but to only compute guaranteed bounds for the reachable spaces. To do so, we will first show how the Müller's theorem makes it possible to derive two dynamical systems which enclose the original uncertain dynamical system and thus bound the flow pipe between a minimal solution, i.e. a flow that is always lower than the solution flow pipe, and a maximal solution, i.e. a flow that is always larger. Since the two bounding systems involve no more uncertainty, interval Taylor models can be used for the guaranteed computation of the minimal and maximal solutions. We will show how to build the bracketing systems by analyzing function f partial derivatives signs. Since the latter may change over integration time period, the bounding systems are in general defined by continuous but only piecewise C^k -differentiable functions. We will show how to use hybrid automata to model them and how to address mode switching. In summary, the computation of the reachable set for an uncertain continuous dynamical system boils down to running two hybrid dynamical systems involving no uncertainty in neither model parameters nor initial state.

2 Guaranteed set integration with interval Taylor models

In this section, we will recall how to perform guaranteed set integration with interval Taylor models.

2.1 Interval analysis

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 ([11] and the references therein). A real interval $[a] = [\underline{a}, \bar{a}]$ is a connected and closed subset of \mathbb{R} . We have $\text{Inf}[a] = \underline{a}$ and $\text{Sup}[a] = \bar{a}$. The set of all real intervals of \mathbb{R} is denoted by \mathcal{IR} . Real arithmetic operations are extended to intervals. Consider an operator $\circ \in \{+, -, *, \div\}$ and $[a]$ and $[b]$ two intervals. Then:

$$[a] \circ [b] = [\inf_{u \in [a], v \in [b]} u \circ v, \sup_{u \in [a], v \in [b]} u \circ v] \quad (3)$$

An interval vector $[\mathbf{a}]$ is a subset of \mathbb{R}^n that can be defined as the Cartesian product of n intervals. One can write $[\mathbf{a}] = [a_1] \times [a_2] \times \dots \times [a_n]$ where $[a_i] = [\underline{a}_i, \bar{a}_i]$. Consider $g : \mathbb{R}^n \mapsto \mathbb{R}^m$; the range of this function over an interval vector $[\mathbf{a}]$ is given by:

$$g([\mathbf{a}]) = \{g(\mathbf{u}) \mid \mathbf{u} \in [\mathbf{a}]\} \quad (4)$$

where the inclusion $\mathbf{u} \in [\mathbf{a}]$ means that $u_i \in [a_i]$ for all $i = 1, \dots, n$. The interval function $[g] : \mathcal{IR}^n \mapsto \mathcal{IR}^m$ is an inclusion function for g if

$$\forall [\mathbf{a}] \in \mathcal{IR}^n, g([\mathbf{a}]) \subseteq [g]([\mathbf{a}]) \quad (5)$$

An inclusion function for g 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 performances of this inclusion function depend on the formal expression for g .

Given a bounded set \mathcal{E} of complex shape, one usually defines an axis-aligned box or a paving, i.e. a union of non-overlapping boxes, $\bar{\mathcal{E}}$ which contains the set \mathcal{E} : this is known as an *outer* approximation of it. Likewise, one also defines an *inner* approximation $\underline{\mathcal{E}}$ which is contained in the set \mathcal{E} . Hence, we have the following properties

$$\underline{\mathcal{E}} \subseteq \mathcal{E} \subseteq \bar{\mathcal{E}} \quad (6)$$

$$\text{vol}(\underline{\mathcal{E}}) \leq \text{vol}(\mathcal{E}) \leq \text{vol}(\bar{\mathcal{E}}) \quad (7)$$

where $\text{vol}(\cdot)$ is the volume of a set.

2.2 Interval Taylor models

Consider now the differential equation (1) and define a time grid $t_0 < t_1 < t_2 < \dots < t_{n_T}$ which is not necessarily equally spaced. The objective is to compute

interval vectors $[\mathbf{x}_j], j = 1, \dots, n_T$, that are *guaranteed* to contain the solution of (1) at time t_j .

Effective methods for solving such a problem are based on Taylor expansions. These methods are usually one-step methods which proceed with two phases:

1. they first verify existence and uniqueness of the solution using the fixed point theorem and the Picard-Lindelöf operator, compute an *a priori* enclosure $[\tilde{\mathbf{x}}_j]$ such that

$$\forall t \in [t_j, t_{j+1}] \quad \mathbf{x}(t) \in [\tilde{\mathbf{x}}_j] \quad (8)$$

and adapt integration time step size $h_j = t_{j+1} - t_j$ if necessary in order to keep the width of $[\tilde{\mathbf{x}}_j]$ and hence the global truncation error smaller than a given threshold;

2. then they compute a tighter enclosure $[\mathbf{x}_{j+1}]$ of the solution of (1) at t_{j+1} , i.e.

$$[\mathbf{x}_{j+1}] = [\mathbf{x}_j] + \sum_{i=1}^{k-1} h_j^i \mathbf{f}^{[i]}([\mathbf{x}_j], [\mathbf{p}], t_j) + h_j^k \mathbf{f}^{[k]}([\tilde{\mathbf{x}}_j], [\mathbf{p}], [t_j, t_{j+1}]) \quad (9)$$

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 $\mathbf{x}(t)$ which can be computed either numerically by automatic differentiation or analytically via formal methods.

The enclosures thus obtained are said *validated* which 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. Unfortunately, the *wrapping* effect makes the explicit scheme (9) width-increasing and thus not suitable for numerical implementation. To solve such a drawback, one can use mean value forms, matrices preconditioning and linear transforms [16].

Remark 1 *When the size of the initial domain or the parameter vector box is too large, guaranteed numerical integration is often doomed to diverge. In such cases, pessimism might be controlled by bisection, i.e. performing a partition of the initial state vector or parameter vector domains. Nevertheless, such a procedure increases computation times very significantly. Hence, the method introduced in this paper investigates the possibility to achieve numerical integration without employing bisection.*

3 Guaranteed set integration using Müller's existence theorem

In this section, we address set integration by using the classical Müller's existence theorem [15, 18] as reported in [12].

Theorem 1 ([18, 12]). Consider the dynamical system (1), where function f is continuous over a domain \mathcal{T} defined by

$$\mathcal{T} : \begin{cases} \omega(t) \leq \mathbf{x}(t) \leq \Omega(t) \\ \underline{\mathbf{p}} \leq \mathbf{p} \leq \overline{\mathbf{p}} \\ t_0 \leq t \leq t_{n_T} \end{cases} \quad (10)$$

where $\mathbf{a} \leq \mathbf{b}$ means $a_i \leq b_i$ for all i . Assume that functions $\omega_i(t)$ and $\Omega_i(t)$ are continuous over $[t_0, t_{n_T}]$ for all i and satisfy the following properties

1. $\omega(t_0) = \underline{\mathbf{x}}_0$ and $\Omega(t_0) = \overline{\mathbf{x}}_0$
2. the lower Dini derivatives $D^-\omega_i(t)$ and $D^-\Omega_i(t)$ and the upper Dini derivatives $D^+\omega_i(t)$ and $D^+\Omega_i(t)$ of $\omega_i(t)$ and $\Omega_i(t)$ are such that

$$\forall i, D^{\pm}\omega_i(t) \leq \min_{\underline{\mathcal{T}}_i(t)} f_i(\mathbf{x}, \mathbf{p}, t) \quad (11)$$

$$\forall i, D^{\pm}\Omega_i(t) \geq \max_{\overline{\mathcal{T}}_i(t)} f_i(\mathbf{x}, \mathbf{p}, t) \quad (12)$$

where $\underline{\mathcal{T}}_i(t)$ is the subset of $\mathcal{T}(t)$ defined by

$$\underline{\mathcal{T}}_i : \{x_i = \omega_i(t), \quad \omega_j(t) \leq x_j \leq \Omega_j(t), j \neq i, \quad \underline{\mathbf{p}} \leq \mathbf{p} \leq \overline{\mathbf{p}}\} \quad (13)$$

and where $\overline{\mathcal{T}}_i(t)$ is the subset of $\mathcal{T}(t)$ defined by

$$\overline{\mathcal{T}}_i : \{x_i = \Omega_i(t), \quad \omega_j(t) \leq x_j \leq \Omega_j(t), j \neq i, \quad \underline{\mathbf{p}} \leq \mathbf{p} \leq \overline{\mathbf{p}}\} \quad (14)$$

Then for all $\mathbf{x}_0 \in [\underline{\mathbf{x}}_0, \overline{\mathbf{x}}_0]$, $\mathbf{p} \in [\underline{\mathbf{p}}, \overline{\mathbf{p}}]$, system (1) admits a solution $\mathbf{x}(t)$ that stays in the domain

$$\mathcal{X} : \begin{cases} t_0 \leq t \leq t_{n_T} \\ \omega(t) \leq \mathbf{x}(t) \leq \Omega(t) \end{cases} \quad (15)$$

and takes the value \mathbf{x}_0 at t_0 . If, in addition, for all $\mathbf{p} \in [\underline{\mathbf{p}}, \overline{\mathbf{p}}]$, function $f(\mathbf{x}, \mathbf{p}, t)$ is Lipschitzian with respect to \mathbf{x} over \mathcal{D} then this solution is unique for any given \mathbf{p} .

Finally, an enclosure for the solution of (1) is given by

$$\forall t \in [t_0, t_{n_T}], \quad [\mathbf{x}](t) = [\omega(t), \Omega(t)] \quad (16)$$

Denote $[\tilde{\omega}_j]$ and $[\tilde{\Omega}_j]$ a priori solutions for bracketing systems (11-12). It is easy to prove that the enclosures

$$[\tilde{\mathbf{x}}_j] = \left[\text{Inf}([\tilde{\omega}_j]), \text{Sup}([\tilde{\Omega}_j]) \right], \quad j = 1, \dots, n_T - 1 \quad (17)$$

satisfy (8) and hence are a priori solutions for (1).

The main difficulty now, is to obtain suitable bracketing functions $\omega(t)$ and $\Omega(t)$ in the general case. However, when the components of f are monotonic with respect to each parameter and each state vector component, it is quite easy to define these systems [12].

Rule 1 [Use of monotonicity property – Analysis of the partial derivatives signs] Here we adapt the idea introduced in [12]. Let's assume that the sign of the partial derivatives $\frac{\partial f_i}{\partial p_k}$ and $\frac{\partial f_i}{\partial x_j}$ is constant over the time period considered. Define $\bar{\delta}^i(p_k)$ as follows

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

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

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

and $\underline{\delta}^i(\mathbf{p}) = [\underline{\delta}^i(p_1), \dots, \underline{\delta}^i(p_k), \dots]^T$. Now define $\bar{\gamma}^i(x_j)$ as follows

$$\bar{\gamma}^i(x_j) = \begin{cases} \Omega_i & \text{if } i = j \\ \Omega_j & \text{if } (i \neq j) \wedge \frac{\partial f_i}{\partial x_j} \geq 0 \\ \omega_j & \text{if } (i \neq j) \wedge \frac{\partial f_i}{\partial x_j} < 0 \end{cases} \quad (20)$$

and $\bar{\gamma}^i(\mathbf{x}) = [\bar{\gamma}^i(x_1), \dots, \bar{\gamma}^i(x_j), \dots]^T$. In a similar way, define $\underline{\gamma}^i(x_j)$ as follows

$$\underline{\gamma}^i(x_j) = \begin{cases} \omega_i & \text{if } i = j \\ \omega_j & \text{if } (i \neq j) \wedge \frac{\partial f_i}{\partial x_j} \geq 0 \\ \Omega_j & \text{if } (i \neq j) \wedge \frac{\partial f_i}{\partial x_j} < 0 \end{cases} \quad (21)$$

and $\underline{\gamma}^i(\mathbf{x}) = [\underline{\gamma}^i(x_1), \dots, \underline{\gamma}^i(x_k), \dots]^T$. Now the components of the differential equations which make it possible to compute the upper and lower solutions are obtained as follows

$$i = 1, \dots, n, \quad \begin{cases} \dot{\omega}_i(t) = f_i(\underline{\gamma}^i(\mathbf{x}), \underline{\delta}^i(\mathbf{p}), t) \\ \dot{\Omega}_i(t) = f_i(\bar{\gamma}^i(\mathbf{x}), \bar{\delta}^i(\mathbf{p}), t) \end{cases} \quad (22)$$

Denote

$$\underline{f}_i(\omega, \Omega, \underline{\mathbf{p}}, \bar{\mathbf{p}}, t) = f_i(\underline{\gamma}^i(\mathbf{x}), \underline{\delta}^i(\mathbf{p}), t) \quad (23)$$

$$\bar{f}_i(\omega, \Omega, \underline{\mathbf{p}}, \bar{\mathbf{p}}, t) = f_i(\bar{\gamma}^i(\mathbf{x}), \bar{\delta}^i(\mathbf{p}), t) \quad (24)$$

then obviously $\omega(t)$ and $\Omega(t)$ are in general, solutions of a system of coupled differential equations, i.e.

$$\begin{cases} \dot{\omega}(t) = \underline{f}(\omega, \Omega, \underline{\mathbf{p}}, \bar{\mathbf{p}}, t), & \omega(t_0) = \underline{\mathbf{x}}_0 \\ \dot{\Omega}(t) = \bar{f}(\omega, \Omega, \underline{\mathbf{p}}, \bar{\mathbf{p}}, t), & \Omega(t_0) = \bar{\mathbf{x}}_0 \end{cases} \quad (25)$$

which involve no uncertain quantity. Therefore interval Taylor models such as the one introduced in the previous section can be used for efficiently solving (25). Indeed when these methods are used for solving differential equations with no uncertainty, they are usually able to curb the pessimism induced by the wrapping effect, even over long integration time.

Remark 2 *Althoug interval Taylor models can be used for solving in an efficient way the system (25), there is no guaranty that the size of the enclosure $[\omega(t), \Omega(t)]$ will not diverge.*

In practice, when rule 1 is used with functions the monotonicity of which changes along the time interval $[t_0, t_{n_T}]$, the obtained bracketing functions $\{\underline{f}, \overline{f}\}$ are not continuously differentiable. Therefore interval Taylor models cannot be used directly for numerical integration. In the sequel, we will show how we can overcome this difficulty by using hybrid automata as bracketing systems. But first, let us recall how to compute the reachable sets with set integration.

4 Computing reachable sets with set integration

For $j = 0, \dots, n_T - 1$ and $t \in [t_j, t_{j+1}]$ define

$$[\mathbf{x}](t) = [\mathbf{x}_j] + \sum_{i=1}^{k-1} (t - t_j)^i \mathbf{f}^{[i]}([\mathbf{x}_j], [\mathbf{p}], t_j) + (t - t_j)^k \mathbf{f}^{[k]}([\boldsymbol{\psi}_j], [\mathbf{p}], [t_j, t_{j+1}]) \quad (26)$$

Proposition 1

If $[\boldsymbol{\psi}_j] \supseteq [\tilde{\mathbf{x}}_j]$ then $\forall t \in [t_j, t_{j+1}], \mathbf{x}(t) \in [\mathbf{x}](t), \quad j = 0, \dots, n_T - 1 \quad (27)$

Proof 1 *It suffices to write a Taylor series expansion at time t_j and use $[\tilde{\mathbf{x}}_j]$ for evaluating the remainder term (see [16]).*

Define $\overline{\mathcal{R}}$ as an over-approximation of a reachable space \mathcal{R} , as follows

$$\forall t, t' \in [t_0, t_{n_T}], \overline{\mathcal{R}}([t, t']; [\mathbf{x}](t)) \supseteq \mathcal{R}([t, t']; [\mathbf{x}](t)) \quad (28)$$

Proposition 2 *A conservative over-approximation of (28) is given by*

$$\forall t \in [t_j, t_{j+1}], \overline{\mathcal{R}}([t_j, t]; [\mathbf{x}_j]) = \cup_{\tau \in [t_j, t]} [\mathbf{x}](\tau), \quad j = 0, \dots, n_T - 1 \quad (29)$$

and satisfies

$$\forall t \in [t_j, t_{j+1}], \overline{\mathcal{R}}([t_j, t]; [\mathbf{x}_j]) \subseteq [\tilde{\mathbf{x}}_j], \quad j = 0, \dots, n_T - 1 \quad (30)$$

Proof 2 *Obvious from (27) and (8).*

Define $\overline{\mathcal{R}}([t_0, t_0]; [\mathbf{x}_0]) = [\mathbf{x}_0]$.

Proposition 3 *An over-approximation of the reachable space 2 is given by*

$$\forall t \in [t_j, t_{j+1}], \overline{\mathcal{R}}([t_0, t]; [\mathbf{x}_0]) = \overline{\mathcal{R}}([t_0, t_j]; [\mathbf{x}_0]) \cup \overline{\mathcal{R}}([t_j, t]; [\mathbf{x}_j]), \quad j = 1, \dots, n_T - 1 \quad (31)$$

and satisfies

$$\forall t \in [t_j, t_{j+1}], \overline{\mathcal{R}}([t_0, t]; [\mathbf{x}_0]) \subseteq \cup_{i \in \{0, j\}} [\tilde{\mathbf{x}}_i], \quad j = 1, \dots, n_T - 1 \quad (32)$$

Proof 3 *Obvious from (27) and proposition (2).*

As a conclusion, it is clear that thanks to (26) and (31), one can derive explicit formulas which characterize the boundaries of the reachable space. In practice however, one can use instead of (31) the over-approximation (32) obtained by using the *a priori* solutions $[\tilde{\mathbf{x}}_j]$ only.

5 Computing a reachable set by using hybrid automata as bounding systems

In this section, we introduce a new approach for enclosing the reachable space of uncertain dynamical systems, for which the signs of the partial derivatives $\partial f_i/\partial x_j$ and $\partial f_i/\partial p_k$ change along the integration time interval $[t_0, t_{n_T}]$. In such a case, the Müller's theorem and rule 1 make it possible to build system (25) over each time interval where functions f_i are monotonic with respect to both variables x_j and p_k . When system (1) is analysed over the whole time interval $[t_0, t_{n_T}]$, the bounding systems given by rule 1 are only piecewise C^k -times continuously differentiable. System (25) can then be regarded as a hybrid dynamical system, and thus be modelled by the following hybrid automaton

$$\mathcal{H} = (\mathcal{Q}, \mathcal{E}, \mathcal{D}, \mathcal{P}, \mathcal{F}, \mathcal{T}, \mathcal{J}) \quad (33)$$

where:

1. \mathcal{Q} is a finite set of modes. For each mode corresponds a continuous-time system which provides the maximal and minimal solution of (1). These systems are built using rule 1.
2. $\mathcal{E} \subseteq \mathcal{Q} \times \mathcal{Q}$ is the set of the transitions. It contains all the possible commutations between the continuous systems which bracket (1).
3. \mathcal{D} is the state space of (1).
4. $\mathcal{P} = [\mathbf{p}] = [\underline{\mathbf{p}}, \overline{\mathbf{p}}]$ represents a feasible domain for model parameters for (1).
5. $\mathcal{F} = \{(\underline{f}_q, \overline{f}_q), q \in \mathcal{Q}\}$ is the collection of bracketing systems obtained with rule 1

$$\forall q \in \mathcal{Q}, \quad \begin{cases} \underline{f}_q : \mathcal{D}^2 \times \mathcal{P}^2 \longrightarrow \mathcal{D} \\ \overline{f}_q : \mathcal{D}^2 \times \mathcal{P}^2 \longrightarrow \mathcal{D} \end{cases} \quad (34)$$

6. $\mathcal{T} = \{t_e, e \in \mathcal{E}\}$ is the collection of switching time instants. Define $g_{i,r}(\cdot) = \frac{\partial f_i}{\partial p_r}(\cdot)$ and $h_{i,l}(\cdot) = \frac{\partial f_i}{\partial x_l}(\cdot)$ with $i \in \{1, \dots, n\}$, $r \in \{1, \dots, n_p\}$ and $l \in \{1, \dots, n\}$. The set \mathcal{T} is defined as

$$\mathcal{T} = \left\{ t_e \in [t_0, t_{n_T}] \mid \exists i, \exists l, \exists r, \exists \mathbf{p} \in [\mathbf{p}], \exists \mathbf{x} \in [\mathbf{x}](t_e) \right. \\ \left. ((g_{i,r}(\mathbf{x}, \mathbf{p}, t_e) = 0) \vee (h_{i,l}(\mathbf{x}, \mathbf{p}, t_e) = 0)) \right\} \quad (35)$$

That is to say that if the monotonicity of f with respect to a parameter or state vector component changes at t_e , a transition $e = (q, q') \in \mathcal{E}$ occurs and the bracketing systems changes too.

7. $\mathcal{J} = \{\mathcal{J}_e, e \in \mathcal{E}\}$ 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 $e = (q, q')$: $\{\underline{\mathbf{x}}_{q'}(t_e), \overline{\mathbf{x}}_{q'}(t_e)\} = \mathcal{J}_e(\underline{\mathbf{x}}_q(t_e), \overline{\mathbf{x}}_q(t_e))$.

Now, in order to use rule 1, 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 as in the preceding sections.

Denote \mathcal{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 of f of (1) are monotonic with respect to each parameter and state vector

$$\mathcal{I}_M = \{[t_j, t_{j+1}] \subseteq [t_0, t_{n_T}] | \forall e \in \mathcal{E}, t_e \notin [t_j, t_{j+1}]\} \quad (36)$$

Next proposition shows how to compute an inner approximation $\underline{\mathcal{I}}_M$ for \mathcal{I}_M , i.e. a set which satisfies the property

$$[t, t'] \in \underline{\mathcal{I}}_M \Rightarrow [t, t'] \in \mathcal{I}_M \quad (37)$$

Proposition 4 (Inner approximation of \mathcal{I}_M) *An inner approximation $\underline{\mathcal{I}}_M \subseteq \mathcal{I}_M$ is given by*

$$\underline{\mathcal{I}}_M \equiv \left\{ \left([t_j, t_{j+1}] \subseteq [t_0, t_{n_T}] | \forall i, \forall l, \forall k, \right. \right. \\ \left. \left. ((0 \notin [g]_{i,k}([\tilde{\mathbf{x}}_j], [\mathbf{p}]), [t_j, t_{j+1}]) \wedge (0 \notin [h]_{i,l}([\tilde{\mathbf{x}}_j], [\mathbf{p}]), [t_j, t_{j+1}])) \right) \right\} \quad (38)$$

Proof 4 *Since the a priori solution $[\tilde{\mathbf{x}}_j]$ of (1) as given by (8) or (17) encloses the whole state trajectory over $[t_j, t_{j+1}]$, we can write*

$$\forall i, \forall j, \forall \mathbf{x}(t_j) \in [\mathbf{x}_j], \forall \mathbf{p} \in [\mathbf{p}], \forall t \in [t_j, t_{j+1}] \\ g_{i,j}(\mathbf{x}, \mathbf{p}, t) \in [g]_{i,j}([\tilde{\mathbf{x}}], [\mathbf{p}], [t_j, t_{j+1}]) \quad (39)$$

Consequently

$$0 \notin [g]_{i,j}([\tilde{\mathbf{x}}], [\mathbf{p}], [t_j, t_{j+1}]) \Rightarrow \forall t \in [t_j, t_{j+1}], g_{i,j}(\mathbf{x}, \mathbf{p}, t) \neq 0 \quad (40)$$

We have similar results for function $h_{i,l}$. This ends the proof.

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

$$\mathcal{I}_S = \{[t_j, t_{j+1}] \subset [t_0, t_{n_T}] | \exists e \in \mathcal{E}, t_e \in [t_j, t_{j+1}]\} \quad (41)$$

Since we have

$$[t_0, t_{n_T}] = \mathcal{I}_M \cup \mathcal{I}_S \quad (42)$$

then an outer approximation $\overline{\mathcal{I}}_S$ of \mathcal{I}_S , i.e. a set which satisfies the property

$$[t, t'] \in \mathcal{I}_S \Rightarrow [t, t'] \in \overline{\mathcal{I}}_S \quad (43)$$

can be obtained as follows

$$\overline{\mathcal{I}}_S = [t_0, t_{n_T}] \setminus \underline{\mathcal{I}}_M \quad (44)$$

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

$$\forall [I_m] \in \underline{\mathcal{I}}_M, \forall m \in \mathcal{Q}, \forall \mathbf{p} \in [\mathbf{p}], \forall \mathbf{x} \in \mathcal{D}, \\ \forall t \in [I_m], \underline{f}_m(\omega, \Omega, \underline{\mathbf{p}}, \overline{\mathbf{p}}, t) \leq \mathbf{f}(\mathbf{x}, \mathbf{p}, t) \leq \overline{f}_m(\omega, \Omega, \underline{\mathbf{p}}, \overline{\mathbf{p}}, t) \quad (45)$$

where $(\underline{f}_m, \overline{f}_m) \in \mathcal{F}$.

One difficulty remains as the actual time instant, i.e., t_e in (35) when the 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 *on-the-fly* in an efficient and guaranteed way. By doing so, we keep the guarantee property for the enclosures without having to derive the actual time instant where the commutation occurs. Let us use *mode 0* to denote the original *uncertain* dynamical system and *modes* $q \neq 0$ to denote coupled bounding systems. The following propositions will make it possible to detect on-the-fly the switching between modes, i.e. $q \neq 0 \mapsto q' = 0$ and $q = 0 \mapsto q' \neq 0$ and to instantiate the new mode.

Proposition 5 (Switching $q \neq 0 \mapsto q' = 0$)

$$\begin{aligned} & \text{If } (q \neq 0) \wedge \\ & \quad (\exists i, \exists l, \exists k, (0 \in [g]_{i,k}([\tilde{\mathbf{x}}_j], [\mathbf{p}], [t_j, t_{j+1}])) \vee (0 \in [h]_{i,l}([\tilde{\mathbf{x}}_j], [\mathbf{p}], [t_j, t_{j+1}]))) \\ & \Rightarrow ((e = (q, q') \wedge (q' = 0)) \wedge ([\tilde{\mathbf{x}}_j] \text{ must be re-computed via (8)}) \end{aligned} \quad (46)$$

Proof 5 When mode $q \neq 0$ and one of the partial derivatives $g(\cdot)$ or $h(\cdot)$ changes sign at $t_e \in [t_j, t_{j+1}]$ then a transition occurs and the new mode is necessarily $q' = 0$. Indeed in this case, the sign of the partial derivative cannot be ascertained for all t in $[t_j, t_{j+1}]$. Now, recall that $[\tilde{\mathbf{x}}_j]$ is computed via (17). But, since solutions $\omega(t)$ and $\Omega(t)$ computed with the bounding systems derived for mode q are valid only over $[t_j, t_e]$, $[\tilde{\mathbf{x}}_j]$ does not contain $[\mathbf{x}](t)$ for $t \in]t_e, t_{j+1}]$. $[\tilde{\mathbf{x}}_j]$ must be re-computed with the original uncertain system.

Proposition 6 (Switching $q = 0 \mapsto q' \neq 0$)

$$\begin{aligned} & \text{If } (q = 0) \wedge \\ & \quad (\forall i, \forall l, \forall k, (0 \notin [g]_{i,k}([\tilde{\mathbf{x}}_j], [\mathbf{p}], [t_j, t_{j+1}])) \wedge (0 \notin [h]_{i,l}([\tilde{\mathbf{x}}_j], [\mathbf{p}], [t_j, t_{j+1}]))) \\ & \Rightarrow (e = (q, q') \wedge (q' \neq 0)) \end{aligned} \quad (47)$$

Proof 6 When mode $q = 0$ and it becomes possible to ascertain the sign of all the partial derivatives $g(\cdot)$ and $h(\cdot)$ for all t in $[t_j, t_{j+1}]$ which is done by using the inclusion functions, then a transition occurs and the new mode is necessarily $q' \neq 0$. $[\tilde{\mathbf{x}}_j]$ is computed with interval Taylor models and is always valid. Numerical integration can then be taken forward from t_{j+1} .

Finally, the algorithm for computing the reachable space of (1) is as follows

Algorithm Hybrid-Bounding

(in: $t_0, t_{n_T}, \mathbf{f}, \mathbf{F}, [\mathbf{x}_0], [\mathbf{p}]$; out: $[\tilde{\mathbf{x}}_0], [\tilde{\mathbf{x}}_1], \dots, [\tilde{\mathbf{x}}_{n_T}], [\mathbf{x}_1], \dots, [\mathbf{x}_{n_T}]$)

1. $j := 0$;
2. $q := \text{Initialize}(f, [\mathbf{x}_0], [\mathbf{p}])$;
3. while ($j < n_T$) do
4. $\{h_j, [\mathbf{x}_{j+1}], [\tilde{\mathbf{x}}_j]\} := \text{Integrate-one-step-ahead}(q, \{f\}, \mathcal{F}, t_j, [\mathbf{x}_j], [\mathbf{p}])$;

5. $\{_jump_, q'\} := \mathbf{Check-Switching}(q, \{f\}, [\tilde{\mathbf{x}}_j]);$
6. if ($_jump_$) then
7. if ($q = 0$) then
8. $q := q'; \quad j := j + 1;$
9. else
10. $q := 0;$
11. endif
12. else
13. $j := j + 1;$
14. endif
15. end

where algorithm **Integrate-one-step-ahead** computes the one-step ahead solution for an uncertain differential equation. It is summarized in the following algorithm

- Algorithm Integrate-one-step-ahead*
(in : $q, \{f\}, \mathcal{F}, t_j, [\mathbf{x}_j], [\mathbf{p}];$ out : $h_j, [\mathbf{x}_{j+1}], [\tilde{\mathbf{x}}_j]$)
1. if $q := 0$ then
 2. $\{h_j, [\mathbf{x}_{j+1}], [\tilde{\mathbf{x}}_j]\} := \mathbf{Interval-Integrate}(f, [\mathbf{x}_j], [\mathbf{p}], t_j);$
 3. else
 4. $(\underline{f}_q, \bar{f}_q) := \mathbf{Select-Boundings}(q, \mathcal{F});$
 5. $[\omega_j] := [\underline{\mathbf{x}}_j];$
 6. $[\Omega_j] := [\bar{\mathbf{x}}_j];$
 7. $\{h_j, [\omega_{j+1}], [\Omega_{j+1}], [\tilde{\omega}_j], [\tilde{\Omega}_j]\} :=$
 $\mathbf{Interval-Integrate}(\underline{f}_q, \bar{f}_q, [\omega_j], [\Omega_j], [\mathbf{p}], [\bar{\mathbf{p}}], t_j);$
 8. $[\tilde{\mathbf{x}}_j] := [\mathit{Inf}([\tilde{\omega}_j]), \mathit{Sup}([\tilde{\Omega}_j])];$
 9. $[\mathbf{x}_{j+1}] := [\mathit{Inf}([\omega_{j+1}]), \mathit{Sup}([\Omega_{j+1}])];$
 10. end

In algorithm **Hybrid-Bounding**, line 2 initializes the initial mode, i.e. at time t_0 . While integration time is smaller than t_{n_T} , algorithm integrates one step ahead from t to $t + h$ (line 4), then checks if a mode switching occurs during the time interval $[t, t + h]$ (line 5). This is done by checking if the signs of the partial derivatives $g_{i,r}$ and $h_{i,l}$ have changed. If this is the case, variable $_jump_$ is set to *true*, otherwise it is set to *false*. If there is a switching, then action will depend on the current mode. If the current mode is $q = 0$ then it suffices to switch to the new mode $q' \neq 0$ and carry on integration (lines 8-9) according to proposition 6. To the contrary, if current mode is not 0, then algorithm has to re-do computation for current time step with the *uncertain* model in order to cross the switching condition in a guaranteed way (lines 11) according to proposition 5.

In algorithm **Integrate-one-step-ahead**, numerical integration is done via interval Taylor models with the original uncertain system when $q = 0$ (line 2). When $q \neq 0$, the bounding systems are selected at line 4 and bounding solutions $\omega(t_j)$ and $\Omega(t_j)$ are set at line 5 and 6. The numerical integration is performed

at line 7. In order to have guaranteed results, we have chosen to use the same interval Taylor model method as in line 2 for solving the coupled system (25), but with intervals of zero width.

6 Example

We consider the Haldane model to simulate the biotechnological process in a stirred reactor. The model is taken from [4] but addresses the existence of one specy on a chemostat with a single substrate. Consider the following equations:

$$\begin{cases} \dot{x} = f_x(x, s) = (\mu_0 \frac{s}{s+k_s+s^2/k_i} - \alpha d)x \\ \dot{s} = f_s(x, s) = -k\mu_0 \frac{s}{s+k_s+s^2/k_i}x + (s_{in} - s)d \end{cases} \quad (48)$$

where x designates the biomass density, s the substrate concentration, d the dilution rate of the chemostat, s_{in} the concentration of input substrate. The coefficients k , k_s , k_i and α are positive constans which are defined as follows $k = 42.14$, $k_s = 9.28 \text{mmol/l}$, $k_i = 256 \text{mmol/l}$ and $\alpha = 0.5$. $s_{in}(t) = s_{in}^0 + 15\cos(1/5t)$ and $d = 2$. The coefficients μ_0 and s_{in}^0 are assumed uncertain : $\mu_0 = 0.75$ with relative uncertainty $\pm 1\%$ and $s_{in}^0 = 65$ with relative uncertainty $\pm 1.5\%$. Initial state is taken uncertain and is defined as follows $x(t_0) \times s(t_0) = [9.5, 10.5] \times [36, 44]$.

It easy to check that the signs of the partial derivatives needed to apply rule 1 are as follows

$$\text{sign}(\partial f_x / \partial s) = \text{sign}(k_s k_i - s^2) \quad (49)$$

$$\forall t > t_0, (\partial f_x / \partial \mu_0 > 0) \wedge (\partial f_s / \partial x < 0) \wedge (\partial f_s / \partial \mu_0 < 0) \wedge (\partial f_s / \partial s_{in}^0 > 0) \quad (50)$$

Hence, the automaton (33) which must be used with algorithm **Hybrid-Bounding** contains only 3 modes :

- mode $q = 0$ corresponds to the original system (48) ;
- mode $q = 1$ is active when $s > \sqrt{k_s k_2}$, i.e. $\partial f_x / \partial s < 0$ and system (25) writes

$$\begin{cases} \dot{\underline{x}} = \underline{\mu}_0 \frac{\underline{s}}{\underline{s}+k_s+\underline{s}^2/k_i} \underline{x} - \alpha \underline{u} \underline{x} \\ \dot{\underline{s}} = -k \underline{\mu}_0 \frac{\underline{s}}{\underline{s}+k_s+\underline{s}^2/k_i} \underline{x} + u(\underline{s}_{in} - \underline{s}) \\ \dot{\bar{x}} = \bar{\mu}_0 \frac{\bar{s}}{\bar{s}+k_s+\bar{s}^2/k_i} \bar{x} - \alpha \bar{u} \bar{x} \\ \dot{\bar{s}} = -k \bar{\mu}_0 \frac{\bar{s}}{\bar{s}+k_s+\bar{s}^2/k_i} \bar{x} + u(\bar{s}_{in} - \bar{s}) \end{cases} \quad (51)$$

- mode $q = 2$ is active when $s < \sqrt{k_s k_2}$ and system (25) writes

$$\begin{cases} \dot{\underline{x}} = \underline{\mu}_0 \frac{\underline{s}}{\underline{s}+k_s+\underline{s}^2/k_i} \underline{x} - \alpha \underline{u} \underline{x} \\ \dot{\underline{s}} = -k \underline{\mu}_0 \frac{\underline{s}}{\underline{s}+k_s+\underline{s}^2/k_i} \underline{x} + u(\underline{s}_{in} - \underline{s}) \\ \dot{\bar{x}} = \bar{\mu}_0 \frac{\bar{s}}{\bar{s}+k_s+\bar{s}^2/k_i} \bar{x} - \alpha \bar{u} \bar{x} \\ \dot{\bar{s}} = -k \bar{\mu}_0 \frac{\bar{s}}{\bar{s}+k_s+\bar{s}^2/k_i} \bar{x} + u(\bar{s}_{in} - \bar{s}) \end{cases} \quad (52)$$

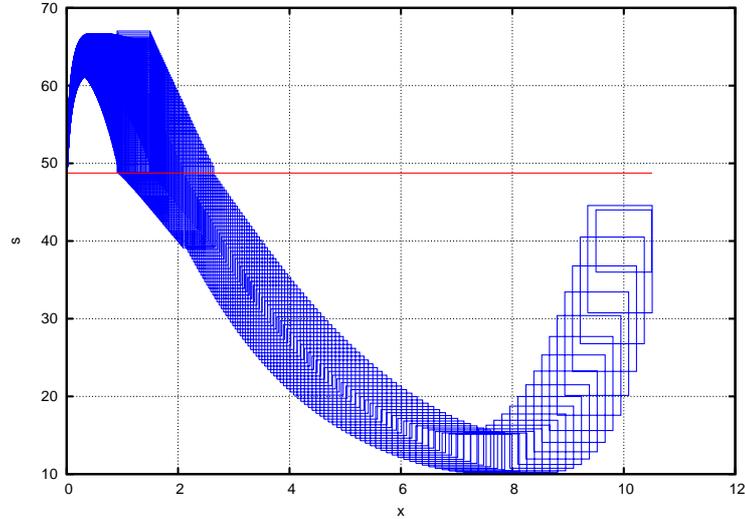


Fig. 1. Reachable space of (48) for $[t_0 = 0\text{s}, t_{n_T} = 20\text{s}]$. CPU time = 16.75s PIV 2GHz

Algorithm **Interval-Integrate** is implemented with the extended mean value algorithm [16] with a constant integration time step $h = 0.03\text{s}$. Profil/BIAS C++ class library is used for interval computations and FADBAD++ package is used for computing the Taylor coefficients. The reachable space as obtained, in 16.75s CPU time, by algorithm **Hybrid-Bouding** for the integration time interval $[t_0 = 0\text{s}, t_{n_T} = 20\text{s}]$ is plotted in figure 1. Note also the switching hyperplane defined by $s = \sqrt{k_s k_2}$. To the contrary, the reachable space as obtained by a state-of-the-art interval Taylor model based method diverges after few steps only.

7 Conclusion

In this paper we have addressed the issue of computing the reachable space for non-autonomous uncertain nonlinear continuous dynamical systems by guaranteed set integration by employing the Müller's existence theorem and hybrid automata as bounding systems. We have shown that this *hybrid bounding* method is capable of computing the reachable space for non-linear systems with fairly large uncertainty in both parameter and state vectors. Used with state-of-the-art hybrid system verification tools, it should make it easier to solve hybrid reachability issues when the continuous dynamical systems are described via nonlinear differential equations. Further work will address error and convergence issues, i.e. how to ensure that the reachable space for a stable nonlinear system is tractable with the introduced method. Also we will study how to optimize the performance of the algorithms introduced when using a bisection strategy for crossing switching hyperplanes. Constraint propagation shall then be of great help.

References

1. 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.
2. E. Asarin, T. Dang, and A. Girard. Hybridization methods for the analysis of non-linear systems. *Acta Informatica*, 43:451–476, 2007.
3. E. Asarin, O. Maler, and A. Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138:35–65, 1995.
4. O. Bernard and J.-L. Gouzé. Closed loop observers bundle for uncertain biotechnological models. *Journal of Process Control*, 14:765–774, 2004.
5. O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In *Hybrid Systems: Computation and Control*, vol. 1790 in LNCS, pages 73–88. Springer, 2000.
6. A. Chutinan and B. H. Krogh. Computational techniques for hybrid systems verification. *IEEE T. Automatic Control*, 48, 2003.
7. L. Doyen, T.A. Henzinger, and J.F. Raskin. Automatic rectangular refinement of affine hybrid systems. In *FORMATS’05*, vol. 3829 in LNCS, pages 144–161, 2005.
8. A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, vol. 3414 in LNCS, pages 291–305, 2005.
9. H. Guéguen and J. Zaytoon. On the formal verification of hybrid systems. *Control Engineering Practice*, 12:1253–1267, 2004.
10. T.A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi. Beyond hytech: Hybrids systems analysis using interval numerical methods. volume 1790, pages 130–144, 2000.
11. L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis: with examples in parameter and state estimation, robust control and robotics*. Springer-Verlag, London, 2001.
12. M. Kieffer, E. Walter, and I. Simeonov. Guaranteed nonlinear parameter estimation for continuous-time dynamical models. In *Proceedings 14th IFAC Symposium on System Identification*, pages 843–848, Newcastle, Aus, 2006.
13. A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for hybrid dynamics: the reachability problem. In W.P. Dayawansa, A. Lindquist, and Y. Zhou, editors, *New Directions and Applications in Control Theory, Lecture Notes in Control and Information Sciences*, volume 321, pages 193–205. Springer, 2005.
14. M.-A. Lefebvre and H. Guéguen. Hybrid abstractions of affine systems. *Nonlinear Analysis*, 65(6):1150–1167, 2006.
15. M. Müller. Über das fundamentaltheorem in der theorie der gewöhnlichen differentialgleichungen. *Mathematische Zeitschrift*, 26:619–645, 1927.
16. 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.
17. A. Rauh, M. Kletting, H. Aschemann, and E.P Hofer. Interval methods for simulation of dynamical systems with state-dependent switching characteristics. In *Proceedings of the 2006 IEEE International Conference on Control Applications*, pages 355–360, Munich, 2006.
18. W. Walter. Differential inequalities and maximum principles: Theory, new methods and applications. *Nonlinear Analysis, Theory, Methods & Applications*, 30(8):4695–4711, 1997.