



Centrum voor Wiskunde en Informatica

REPORTRAPPORT

MAS

Modelling, Analysis and Simulation



Modelling, Analysis and Simulation

Semantics and computability of the evolution of hybrid systems

P.J. Collins

REPORT MAS-R0801 JANUARY 2008

Centrum voor Wiskunde en Informatica (CWI) is the national research institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organisation for Scientific Research (NWO). CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

Copyright © 2008, Stichting Centrum voor Wiskunde en Informatica
P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

Semantics and computability of the evolution of hybrid systems

ABSTRACT

In this paper we consider the semantics for the evolution of hybrid systems, and the computability of the evolution with respect to these semantics. We show that with respect to lower semantics, the finite-time reachable sets are lower-semicomputable, and with respect to upper semantics, the finite-time reachable sets are upper-semicomputable. We use the framework of type-two Turing computability theory and computable analysis, which deal with obtaining approximation results with guaranteed error bounds from approximate data. We show that in general, we cannot find a semantics for which the evolution is both lower- and upper-semicomputable, unless the system is free from tangential and corner contact with the guard sets. We highlight the main points of the theory with simple examples illustrating the subtleties involved.

2000 Mathematics Subject Classification: 93B03; 93-04, 68Q17, 93B40.

Keywords and Phrases: Computable analysis; hybrid system; reachable set.

Note: This research was supported by Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) Vidi grant 639.032.408.

Semantics and Computability of the Evolution of Hybrid Systems

Pieter Collins[†]

[†]Centrum voor Wiskunde en Informatica,
Postbus 94079, 1090 GB Amsterdam, The Netherlands,
Pieter.Collins@cwi.nl *

January 29, 2008

Abstract

In this paper we consider the semantics for the evolution of hybrid systems, and the computability of the evolution with respect to these semantics. We show that with respect to *lower* semantics, the finite-time reachable sets are lower-semicomputable, and with respect to *upper* semantics, the finite-time reachable sets are upper-semicomputable. We use the framework of type-two Turing computability theory and computable analysis, which deal with obtaining approximation results with guaranteed error bounds from approximate data. We show that in general, we cannot find a semantics for which the evolution is both lower- and upper- semicomputable, unless the system is free from *tangential* and *corner* contact with the guard sets. We highlight the main points of the theory with simple examples illustrating the subtleties involved.

Key words. Computable analysis; hybrid system; reachable set.

AMS subject classifications. 93B03; 93-04, 68Q17, 93B40.

1 Introduction

Hybrid systems are dynamic systems in which the evolution has both discrete-time (instantaneous) and continuous-time elements. Hybrid models are becoming increasingly prevalent in industry, and there is a need for tools which can perform reliable simulation and verification analysis of hybrid systems. The interplay between the continuous and discrete dynamics causes difficulties in the analysis of hybrid systems which do not occur in discrete- or continuous-time systems, and which lend hybrid systems a distinctive character.

Many questions about the behaviour of a hybrid system can be framed in the context of reachable sets and the reachability relation. It has long been known that the reachability relation for hybrid systems is undecidable [2], except for the class of timed automata (and slight generalisations), for which reachable sets can be computed exactly [26]. Rather than considering decidability of the reachability relation, it is more natural to consider the computability of the reachable set. For more complicated systems, symbolic computations are infeasible and approximate numerical computations are required. This motivates the study

*This research was supported by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) Vidi grant 639.032.408.

of what is possible to compute using approximations to the exact problem data if it is only necessary to compute the result approximately.

In this paper, we base our computability results on the theory of computable analysis of Weihrauch and co-workers [28], which is equivalent to that of [23] based on oracle machines. All computations are performed using ordinary Turing machines, and hence can be implemented using existing computers. (This is unlike the real-RAM theory of [8], which cannot be effectively implemented.) In order to allow computation on uncountable sets, we allow computations to run indefinitely, writing an output stream which represents successively more accurate approximations to the result. We say a quantity is computable if it can be computed to arbitrary (metric) accuracy, and semicomputable if it is possible to compute convergent approximations from above or below. We note that uncomputability in the framework of computable analysis does not necessarily imply uncomputability in some algebraic framework in which the objects of interest can be specified exactly. The results in this paper extend those of [16], and provide full proofs. Similar results on the computability of reachable sets for discrete-time systems were given in [15].

We will see that computability of the reachable set is strongly related to topological properties of the invariants and guard sets, and continuity properties of the continuous and discrete dynamics. In order to separate technical issues relating to the solution of differential equations and differential inclusions from the intrinsic difficulties of hybrid systems, we describe the continuous dynamics directly as a flow, rather than by a differential relation. Upper-semicontinuity of solutions of hybrid systems has been considered in [4, 19]. Lower-semicontinuity of the solutions of Lipschitz differential inclusions and hybrid systems has been studied in [11, 12].

Unlike purely discrete- or continuous-time systems, for which there is a well-defined notion of solution, for hybrid systems we need to use different solution concepts for computing lower- and upper-approximations to the reachable set. The upper solution concept may necessarily impose nondeterministic (multivalued) solutions to an otherwise deterministic system, whereas lower solution concepts may impose blocking. Reliable simulation imposes the need to consider multiple possible evolutions, each of a qualitatively different nature.

There are many other tools available for computing reachable sets of hybrid systems, such as *d/dt* [1], *Hy(per)tech* [22], *VeriShift* [9], *Checkmate* [5] and *Phaver* [18]. However, these tools are mainly restricted to systems with affine dynamics and guard sets (apart from *CheckMate*, which allows nonlinear dynamics), and can only compute over-approximations to the reachable set. To remedy this situation, a software tool *ARIADNE* [6] is being developed to implement the computable operations of this paper. Computation of the solution of hybrid systems using a set-oriented approach using the software package *GAIO* [17], which is particularly applicable to the computation of the operators studied in this paper, has been considered in [21].

The paper is structured as follows. In Section 2 we indicate the difficulties encountered in the study of hybrid systems, and motivate the use of a formal computability theory. In Section 3 we give some technical preliminaries on computability theory for points and sets. In Section 4 we present the ways in which the evolution of a hybrid system may fail to be continuous. In Section 5 we present the main theorems on semicomputability of the evolution. In Section 6 we present some modelling frameworks for hybrid systems, and discuss reliable simulation and implementation issues. Finally, in Section 7 we state some conclusions and give directions for further research.

2 Motivation

2.1 Continuous- and Hybrid-Time Systems

One of the most important results in the theory of continuous-time systems is the existence and uniqueness result for Lipschitz differential equations $\dot{x} = f(x)$. Further, if $\Phi : X \times \mathbb{R} \rightarrow X$ is the solution flow of the differential equation, then Φ is continuous, and can be effectively approximated, in the sense that given the function f , the initial condition x_0 and the integration time t , we can compute $\Phi(x_0, t)$ arbitrarily accurately on a digital computer. In many situations, the data f , x_0 and t may not be known exactly. However, even in this case, given a sufficiently accurate description of f and x_0 , we can still compute the evolution Φ .

Compare the situation for differential equations with that for hybrid systems. If we denote the solution of a hybrid system with initial condition x at time t by $\Psi(t, x)$, we see that there are a number of situations in which the solution may vary discontinuously in x and t .

Time discontinuity at discrete transitions Whenever the state of the system is reset during a discrete transition, a discontinuity in the time evolution occurs.

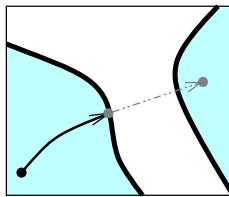


Figure 1: Time discontinuity at a discrete transition.

Spacial discontinuity at tangencies and corner collisions If the continuous evolution touches a guard set tangentially near x , then some points near x undergo a discrete transition, whereas other points undergo further continuous evolution. The same phenomenon may occur if the continuous evolution touches a corner of a guard set.

Spacial discontinuity at switching boundaries If x lies on the boundary of two guard sets, then some points near x undergo one transition, and others undergo the other.

Spacial discontinuity at instantaneous transitions Suppose that after a discrete transition, the state x lies on the boundary of the switching set. Then some points near x immediately undergo a second transition, whereas other points may flow away for the activation region and do not undergo the transition.

All these situations may occur generically, which means that they persist under small perturbations of the parameters defining the system. From the viewpoint of dynamics, it is these discontinuities which distinguish hybrid systems from purely discrete-time or continuous-time systems. In many cases, the spacial discontinuities only occur for a “small” (measure zero) set of initial conditions, and might therefore be considered not to be of physical interest. However, spacial discontinuities may still occur on a (locally) dense set of initial conditions. If the exact solution passes very near a discontinuity point at time t , then the presence of even a small numerical error may cause the computed solution after time t to

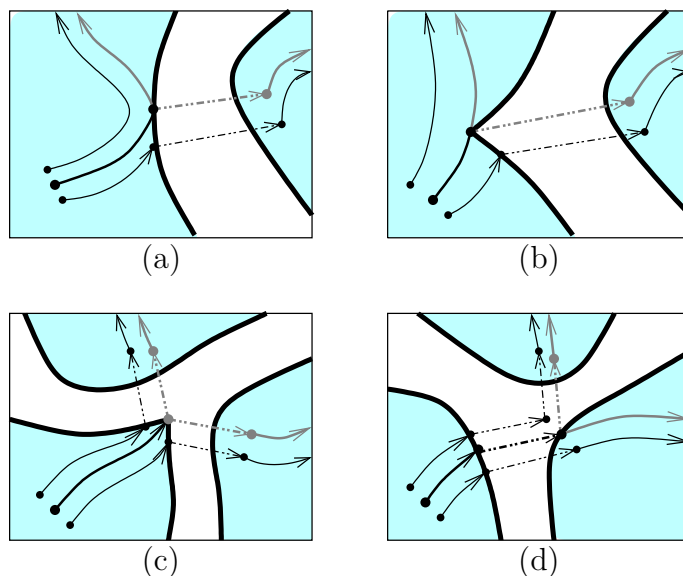


Figure 2: Spatial discontinuities. (a) At a tangency; (b) at a corner collision, (c) at a switching boundary, and (d) at an instantaneous transition.

differ drastically from the exact solution. As we shall see, it is important to handle these situations correctly in the development of a sound numerical theory of hybrid systems.

2.2 Computability theory

We have seen that hybrid systems may exhibit discontinuities in the evolution, and intuitively we expect that the presence of discontinuities will cause difficulties in computing the system evolution, even to the extent that it is impossible to compute the evolution to arbitrary accuracy. However, to actually prove that a certain computational task is impossible, we need a formal theory of computation, which requires specifying a computational model, and also the input and output data that the computational model works with. We compare this motivation with Turing's motivation for introducing his computing machines, which was to prove the impossibility of an algorithmic solution of Hilbert's *Entschuldigungsverfahren* problem. Since we are interested in algorithmic solutions to problems concerning hybrid systems, if our original problem turns out to be unsolvable in general, we want to know to what extent our problems are solvable, or find related problems which are completely solvable.

In this paper, we use the theory of *computable analysis* as developed by Weihrauch [28] and co-workers. In this theory, computation is performed by ordinary Turing machines acting on streams of data. The data stream encodes a sequence of approximations to some quantity, such as a subset of the state space, or a function describing a system. A function or operator is computable if given a data stream encoding a sequence of approximations converging to the input, it is possible to calculate a data stream encoding a sequence of approximations converging to the output. In practice, finite computations can be obtained by terminating whenever a given accuracy criterion is met. However, it is theoretically very convenient to consider the computations to be infinite, since we can talk about computing the mathematical objects themselves. Two encodings or *representations* of the same class of mathematical object are equivalent if each can be transformed into the other by a Turing

machine; this makes it possible to relate results on representations which are easy to work with theoretically to representations which are efficient to work with in implementations.

The representations used in computable analysis are each related to a topology on the set of objects under consideration, and so give a clean link between approximability, continuity and formal computability. The fundamental theorem is that only continuous functions with respect to a given topology can be computable with respect to representations based on that topology. Hence if we can prove that a function is discontinuous, then it is uncomputable. For “naturally” defined functions the converse is typically also true, that continuous functions are computable. It is worth emphasising that a function which is uncomputable with respect to one representation may be computable with respect to a representation based on a different topology. This corresponds to giving more information in the input, or requiring less information in the output. We shall see later that the use of the correct topology/representation is vital when considering computability for hybrid systems.

Since objects are described by sequences of symbols, we can represent sets of continuum cardinality. This includes points in Euclidean space, open, closed and compact subsets, continuous functions and semicontinuous multivalued functions, but not arbitrary subsets of space or arbitrary discontinuous functions. It is also possible to represent Borel probability measures and measurable functions, though in this article, we only consider computations involving points and sets. In particular, we will require the data describing our systems to be in terms of open/closed sets and (semi)continuous functions.

The representations used must allow information about the objects they describe to be obtained from a finite amount of data. Consider a computation whose result is some real number x . In traditional numerical analysis, it is usual to compute a sequence of floating-point or rational approximations x_n converging to x . Often some order of convergence is given, such as $|x - x_n| = O(1/n^k)$ for some integer k . Unfortunately, in this model, knowing some particular approximation x_n gives in theory *no information* on the value of x . To gain information about x , we also need to know an error bound ϵ_n for the approximation, such that $|x - x_n| < \epsilon_n$. If $\epsilon_n \rightarrow 0$, then we can compute an approximation to x with arbitrary *known* accuracy. We say that x_n *converges effectively* to x . In some problems, especially optimisation problems, we merely seek a sequence of approximations x_n converging to x from above (or below). In this case, we cannot give metric bounds on x , but can still deduce properties of x , such as $x > x_n$.

In theoretical work, especially when making a link between computation and topology, it is more convenient to work with *properties* of objects. For example, if (a, b) is an open interval, then $x \in (a, b)$ is a property of x . Further, such properties should be *robust*, in the sense that if some property holds for x , then it holds for all y near x . Topologically, this means that a property corresponds to membership of an open set.

To describe arbitrary objects in some space, we first choose a countable collection $\sigma = \{I_1, I_2, \dots\}$ of *basic* open sets (properties) such that x is determined uniquely by its properties. For example, if we take σ to be the collection of all open intervals (a, b) with rational endpoints, then determining whether $x \in (a, b)$ for all $(a, b) \in \sigma$ is sufficient to determine x . Usually, we only need to know a subset of properties to determine x and all of its properties uniquely. For example, if we can enumerate a sequence of rational intervals (a_n, b_n) such that $x \in (a_n, b_n)$ for all n and $\lim_{n \rightarrow \infty} b_n - a_n = 0$, then we can determine all other intervals (a, b) such that $x \in (a, b)$. Notice that the information given by approximations is equivalent to the information given by properties. For if we know $x \in (a, b)$, then $(a + b)/2$ is an approximation to x with error $\epsilon = (b - a)/2$.

In practice, we cannot determine all properties of x , or compute an infinite sequence

of approximations to x . Instead, we are usually content to compute sufficient information about x to be able to approximate x to some desired accuracy (which can be checked a-posteriori). However, it is useful to know that x can be approximated to *any* desired accuracy. Further, by describing x by a list of *all* its properties, we can often conceptually work with the object x itself rather than with approximations to x , a considerable simplification.

While the model of computation, being based on Turing machines, subsumes ordinary, finite computation, the main purpose of computable analysis theory is to deal with approximations. In particular, the data describing the systems is interpreted as being approximate. This can drastically change the computability properties. Consider the following simple example:

Example 2.1. Consider the differential equation $\dot{x} = x^2 + \epsilon$ with initial condition $x(0) = -1$. We wish to determine whether the solution remains bounded. If ϵ is taken to be a rational number which is described exactly, the problem is always solvable; the solution is bounded if and only if $\epsilon \leq 0$. However, if the *only* information we have about ϵ is approximate (possibly ϵ is an experimental parameter) then if $\epsilon = 0$, then no matter how accurate the approximation to ϵ , we cannot eliminate the possibility that $\epsilon > 0$ and that the solution is unbounded.

To summarise, boundedness of the solution in the case $\epsilon = 0$ is undecidable when using approximate data, but decidable using exact data. Further, if ϵ is very close to 0, we need a very accurate approximation to ϵ in order to determine boundedness. Even in the exact model, if $\epsilon = 0$, then a very small amount of noise in the system will destroy boundedness.

The interested reader is strongly advised to read [28] for more details.

3 Technical preliminaries

3.1 Multivalued Dynamical Systems

In many applications, it is convenient to represent systems by nondeterministic models defined by multivalued functions. Further, as we shall see, nondeterminism is unavoidable if we are to give a framework for hybrid systems under which we can compute the evolution.

We say F is a *multivalued function* from X to Y , denoted $F : X \rightrightarrows Y$, if F associates to each $x \in X$, a subset $F(x)$ of Y . If $A \subset X$, we define $F(A) = \bigcup_{x \in A} F(x)$, and if $F : X \rightrightarrows Y$ and $G : Y \rightrightarrows Z$, we define $G \circ F : X \rightrightarrows Z$ by $G \circ F(x) := G(F(x)) = \bigcup_{y \in F(x)} G(y)$. The *preimage* $F^{-1} : Y \rightrightarrows X$ of a multivalued function $F : X \rightrightarrows Y$ is defined by $F^{-1}(B) = \{x \in X \mid F(x) \cap B \neq \emptyset\}$. We say F is *lower-semicontinuous* if $F^{-1}(V)$ is open whenever V is open, and *upper-semicontinuous* if $F^{-1}(B)$ is closed whenever B is closed. F is *continuous* if it is both lower- and upper-semicontinuous. If $F : X \rightrightarrows Y$ is closed-valued lower-semicontinuous and C is compact, then $F(C)$ need not be closed, but for any set A , $F(\bar{A}) \subset \text{cl}(F(A))$.

In control theory, lower-semicontinuous functions are often appropriate to model control input, and upper-semicontinuous functions are appropriate to model disturbances. For hybrid automata (without inputs), lower-semicontinuous functions are used if we want to be sure that a trajectory with some property *exists*, whereas upper-semicontinuous functions are used if we want to be sure that *all* trajectories have some property.

We write $f : \subset X \rightarrow Y$ if f is a partial function from X to Y . Let $C(\mathbb{R}^+ \dashrightarrow X)$ be the set of continuous partial functions $\eta : \subset \mathbb{R}^+ \rightarrow X$ such that $\text{dom}(\eta)$ is a nonempty

connected interval $[0, t]$ or $[0, t[$. A *multivalued flow* is a subset Φ of $C(\mathbb{R}^+ \dashrightarrow X)$ with the following properties:

1. If $\eta_1, \eta_2 \in \Phi$ and $\eta_1(s) = \eta_2(0)$, then the catenation $\eta = \eta_1 \cdot \eta_2$ given by $\eta(t) = \eta_1(t)$ for $t \leq s$ and $\eta(t) = \eta_2(t - s)$ for $t \geq s$, $t - s \in \text{dom}(\eta_2)$ is in Φ .
2. If $\eta \in \Phi$, then the shift $\sigma_s \eta$ given by $\sigma_s \eta(t) = \eta(t + s)$ for $t + s \in \text{dom} \eta$ is in Φ .
3. If $\eta \in \Phi$, then the restriction of η to an initial subdomain is in Φ .

Given a flow $\Phi \subset C(\mathbb{R}^+ \dashrightarrow X)$, we can define

- a multivalued function $X \rightrightarrows C(\mathbb{R}^+ \dashrightarrow X)$ by $x \mapsto \{\eta \in \Phi \mid \eta(0) = x\}$, and
- a multivalued function $X \times \mathbb{R}^+ \rightrightarrows X$ by $(x, t) \mapsto \{y \in X \mid \exists \eta \in \Phi \text{ s.t. } \eta(0) = x \text{ and } \eta(t) = y\}$.

We will use Φ interchangeably to denote the flow as a subset of $C(\mathbb{R}^+ \dashrightarrow X)$, as the multivalued function $X \rightrightarrows C(\mathbb{R}^+ \dashrightarrow X)$ or as the multivalued functions $X \times \mathbb{R}^+ \rightrightarrows X$ defined above. The usage will be clear from the context.

We say a flow is *upper-semicontinuous* if $\text{dom}(\eta)$ is closed for all $\eta \in \Phi$, and the map $\Phi : X \rightrightarrows C(\mathbb{R}^+ \dashrightarrow X)$ is upper-semicontinuous, and *lower-semicontinuous* if $\text{dom}(\eta) = [0, t[$ is half-open for all $\eta \in \Phi$, and $\Phi : X \rightrightarrows C(\mathbb{R}^+ \dashrightarrow X)$ is lower-semicontinuous.

A *differential inclusion* on a manifold X is a continuous-time evolution equation of the form $\dot{x} \in F(x)$ where $F : x \in X \rightrightarrows T_x X$. A *solution* to the differential inclusion $\dot{x} \in F(x)$ is an absolutely continuous function $\xi : [0, T) \rightarrow X$ such that $\dot{\xi}(t) \in F(\xi(t))$ for almost all $t \in [0, T)$. The *flow* of a differential inclusion is set of all solutions. In this paper we work directly with flows and their semicontinuity properties, but lower- There is considerable work in the literature on semicontinuity properties of the solutions of a differential inclusion, see [3] for an overview.

3.2 Hybrid Systems

A minimal definition of a hybrid system is

Definition 3.1. A hybrid system is a triple $H = (X, \Phi, R)$ where X is the state space, $\Phi \subset C(\mathbb{R}^+ \dashrightarrow X)$ is a *dynamic* satisfying the flow conditions and $R \subset X \times X$ is the *reset relation*.

We will typically restrict to single-valued flows in the examples. To represent a trajectory of a hybrid system, we need to take into account the possibility that more than one discrete event occurs at a given time. To capture the intermediate states, we use the following definition of *hybrid time domain* [14, 20], which is based on work of [25]:

Definition 3.2 (Hybrid trajectory). Let $(t_i)_{i < \infty}$ be an increasing sequence in $\mathbb{R}^+ \cup \{\infty\}$ with $t_0 = 0$. Then the t_i define a *hybrid time domain* $\mathcal{T} \subset \mathbb{R}^+ \times \mathbb{Z}^+$ by

$$\mathcal{T} = \{(t, n) \in \mathbb{R}^+ \times \mathbb{Z}^+ \mid t_n \leq t \leq t_{n+1}\} = \bigcup_{n=0}^{\infty} [t_n, t_{n+1}] \times \{n\}.$$

A *hybrid trajectory* is a continuous function $\xi : \mathcal{T} \rightarrow X$ for some hybrid time domain. This is equivalent to requiring that $t \mapsto \xi(t, n)$ is continuous for $t \in [t_n, t_{n+1}]$.

The trajectory ξ is *Zeno* if $\lim_{n \rightarrow \infty} t_n < \infty$, and has finitely many events if there exists n such that $t_m = \infty$ for all $m > n$.

The evolution of a hybrid system consists of continuous flow interspersed with discrete transitions.

Definition 3.3 (Solution of a hybrid system). A hybrid trajectory is a *solution* or *execution* of the hybrid system $H = (X, \Phi, R)$ if

1. $\xi(\cdot, n) \in \Phi$, and
2. $(\xi(t_n, n-1), \xi(t_n, n)) \in R$ for all n .

The *evolution* Ψ of a hybrid system H is the function $\Psi : X \times \mathbb{R}^+ \times \mathbb{Z}^+ \rightrightarrows X$ by

$$\Psi(x, t, n) = \{y \mid \exists \text{ solution } \xi \text{ of } H \text{ s.t. } \xi(0, 0) = x \text{ and } \xi(n, t) = y\}. \quad (1)$$

In later sections, we will consider hybrid systems defined using invariant domains, activation regions and guard sets.

3.3 Computable analysis for sets and functions

We now outline how to describe objects such as points, sets and functions in the framework of computable analysis. The material in this section can be found in [28, 15].

Formally, a *representation* of some set X is a partial surjective function $\delta : \subset \Sigma^\omega \rightarrow X$ for some finite alphabet Σ . We say $w \in \Sigma^\omega$ is a δ -*name* of $x \in X$ if $\delta(w) = x$.

Let X be a topological space whose topology τ is generated by a countable collection of open sets $\sigma = \{I_0, I_1, \dots\}$. Then δ is the *standard representation* of (X, τ, σ, ν) if a δ -name of $x \in X$ is a binary encoding of an enumeration of $\{I \in \sigma \mid x \in I\}$. Informally, we say that a δ -name of x encodes a list of all $I \in \sigma$ such that $x \in I$.

We say that a function $f : X_1 \times \dots \times X_k \rightarrow X_0$ is $(\delta_1, \dots, \delta_k; \delta_0)$ -computable if there is a Turing-computable partial function $\mathcal{M} : \subset \Sigma^\omega \times \dots \times \Sigma^\omega \rightarrow \Sigma^\omega$ such that $\delta_0(\mathcal{M}(w_1, \dots, w_k)) = f(\delta_1(w_1), \dots, \delta_k(w_k))$ whenever the right-hand side is defined.

We will restrict to hybrid systems such that the state space X is a locally-compact second-countable Hausdorff space, and let β be a base for X . For Euclidean space $X = \mathbb{R}^n$, we take β to be the collection of all open bounded boxes with rational endpoints; $(a_1, b_1) \times (a_2, b_2) \times \dots \times (a_n, b_n)$ with $a_i, b_i \in \mathbb{Q}$ for $i = 1, 2, \dots, n$.

We have the following representations of points and sets.

- A ρ -name of $x \in X$ encodes a list of all $I \in \beta$ such that $x \in I$.
- A $\theta_{<}$ -name of open $U \subset X$ encodes a list of all $I \in \beta$ such that $\bar{I} \subset U$.
- A $\psi_{<}$ -name of closed $A \subset X$ encodes a list of all $I \in \beta$ such that $I \cap A \neq \emptyset$.
- A $\psi_{>}$ -name of closed $A \subset X$ encodes a list of all $I \in \beta$ such that $\bar{I} \cap A = \emptyset$.
- A $\kappa_{>}$ -name of compact $C \subset X$ encodes a list of all tuples $(I_1, \dots, I_k) \in \beta^*$ such that $C \subset \bigcup_{i=1}^k I_i$. An equivalent representation encodes a $\psi_{>}$ -name of C together with an $I \in \beta$ such that $C \subset I$.
- A κ -name of compact C encodes both a $\psi_{<}$ -name and a $\kappa_{>}$ -name.

It is easy to see that each of the properties encoded is robust with respect to a small change in the set being described. For example, if \bar{I} is a subset of some open set U , then \bar{I} is also a subset of V for any sufficiently small perturbation V of U . The information given by $\theta_<$ is sufficient to compute a sequence of sets (described as finite unions of boxes) converging to U from inside, and the information given by $\kappa_>$ is sufficient to compute a convergent sequence of outer-approximations to C . The information given by κ is sufficient to compute C to arbitrary accuracy in the Hausdorff metric.

Using these representations, the first natural question to ask is which geometric operations (union, intersection) are computable. It turns out that the finite union of closed sets is upper-semicomputable i.e. $(A, B) \mapsto A \cup B$ is $(\psi_>, \psi_>; \psi_>)$ -computable, and the countable union of open sets is lower-semicomputable. The countable closed union of closed sets is also lower-semicomputable i.e. $(A_1, A_2, \dots) \mapsto \text{cl}(\bigcup_{n=1}^{\infty} A_n)$ is $(\psi_<, \psi_<, \dots; \psi_<)$ -computable. The intersection of two closed sets $(A, B) \mapsto A \cap B$ is upper-semicomputable i.e. $(\psi_>, \psi_>; \psi_>)$ -computable but not lower-semicomputable i.e. not $(\psi, \psi; \psi_<)$ -computable. In other words it is not possible to enumerate all basic open sets $I \in \beta$ such that $(A \cap B) \cap I \neq \emptyset$ from similar enumerations for A and B . However, the closure of the intersection of an open and a closed set is lower-semicomputable i.e. $(U, A) \mapsto \text{cl}(U \cap A)$ is $(\theta_<, \psi_<; \psi_<)$ -computable.

We now wish to describe continuous functions. The standard way of doing this is via the *compact-open* representation. For if $f : X \rightarrow Y$ is continuous and $J \subset Y$ is open, then $f^{-1}(J)$ is open. Hence if \bar{I} is compact, then the property $\bar{I} \subset f^{-1}(J)$ is robust. Alternatively, if \bar{I} is compact, then $f(\bar{I})$ is compact, so the property $f(\bar{I}) \subset J$ is robust, and is equivalent to $\bar{I} \subset f^{-1}(J)$. Hence:

- A γ -name of continuous $f : X \rightarrow Y$ encodes a list of all $(I, J) \in \beta_X \times \beta_Y$ such that $\bar{I} \subset f^{-1}(J)$.

With this representation, the operator $(f, x) \mapsto f(x)$ is $(\gamma, \rho; \rho)$ -computable, the operator $(f, A) \mapsto \text{cl}(f(A))$ is $(\gamma, \psi_<; \psi_<)$ -computable (lower-semicomputable), the operator $(f, C) \mapsto f(C)$ is $(\gamma, \kappa_>; \kappa_>)$ -computable (upper-semicomputable) and $(f, U) \mapsto f^{-1}(U)$ is $(\gamma, \theta_<; \theta_<)$ -computable (lower-semicomputable). Further, the solution operator for Lipschitz differential equations is computable, or in other words, the operator $(f, x, t) \mapsto \Phi^f(t, x)$ is $(\gamma, \rho_X, \rho_{\mathbb{R}}; \rho_X)$ -computable, where $\Phi(t, x)$ denotes the flow of f satisfying $\dot{x}(t) = f(x(t))$ if $x(t) = \Phi(t, x_0)$.

We have the following representations of multivalued maps.

- A $\mu_<$ name of a lower-semicontinuous map $F : X \rightrightarrows Y$ with closed values encodes a list of all pairs $(I, J) \in \beta_X \times \beta_Y$ such that $\bar{I} \subset F^{-1}(J)$.
- A $\mu_>$ name of an upper-semicontinuous map $F : X \rightrightarrows Y$ with compact values encodes a list of all tuples $(I, J_1, \dots, J_k) \in \beta_X \times \beta_Y^*$ such that $F(\bar{I}) \subset \bigcup_{i=1}^k J_i$.

It is easy to show that the closure of the image of a closed set under a closed-valued lower-semicontinuous function is lower-semicomputable i.e. $(F, A) \mapsto \text{cl}(F(A))$ is $(\mu_<, \psi_<; \psi_<)$ -computable, and the image of a compact set under a compact-valued upper-semicontinuous function is upper-semicomputable i.e. $(F, C) \mapsto \text{cl}(F(A))$ is $(\mu_>, \kappa_>; \kappa_>)$ -computable. Further, the information provided by the image of a set, or even a point, under a multivalued function F , is precisely enough to compute a name of F . In other words, if we have a compact-valued upper-semicontinuous multivalued function F ,

and we can compute $x \mapsto F(x)$ in the sense that given a ρ -name of x we have an algorithm to generate a $\kappa_{>}$ -name of $F(x)$, then we can generate a $\mu_{>}$ -name of F .

We have the following representations of multivalued flows.

- A $\phi_{<}$ -name of a lower-semicontinuous multivalued flow Φ encodes a list of all $(I, T, J) \in \beta_X \times \beta_{\mathbb{R}} \times \beta_X$ such that for all $x \in \bar{I}$, there is a solution ξ such that $\xi(0) = x$ and $\xi(\bar{T}) \subset J$.
- A $\phi_{>}$ -name of an upper-semicontinuous multivalued flow Φ encodes a list of all $(I, T, \mathcal{J}) \in \beta_X \times \beta_{\mathbb{R}} \times \beta_X^*$ such that for all $x \in \bar{I}$, and all solutions ξ such that $\xi(0) = x$, ξ is defined on \bar{T} and $\xi(\bar{T}) \subset \mathcal{J}$; equivalently, for all $x \in \bar{I}$ and $t \in \bar{T}$, $\Phi(x, t) \subset \mathcal{J}$.

Note that the information given by a $\phi_{<}$ -name of a lower-semicontinuous multivalued flow Φ is strictly stronger than the information provided by a $\mu_{<}$ -name of Φ considered as a multivalued function $\Phi : X \times \mathbb{R}^+ \rightrightarrows X$. The information given by a $\phi_{>}$ -name of an upper-semicontinuous multivalued flow Φ is equivalent to the information provided by a $\mu_{>}$ -name of Φ considered as a multivalued function $\Phi : X \times \mathbb{R}^+ \rightrightarrows X$.

In this paper, we work directly with flows, and do not consider explicitly consider differentiable formalisms of the continuous dynamics. This is actually no restriction, since we can effectively compute the solution of differential inclusions under standard conditions. The solution of a general locally Lipschitz continuous differential inclusion was shown to be computable (using different terminology) in [27]. We can refine this result and consider lower-semicomputability and upper-semicomputability separately.

Theorem 3.4 (Computability of differential inclusions). *Let $\Phi : \mathbb{R}^+ \times X \rightrightarrows X$ denote the flow of the differential inclusion $\dot{x} \in F(x)$.*

1. *If F is upper-semicontinuous with compact convex values and linear growth at infinity, then the solution operator $F \mapsto \Phi$ is upper-semicomputable; more precisely, $F \mapsto \Phi$ is $(\mu_{>}; \phi_{>})$ -computable..*
2. *If F is locally Lipschitz lower-semicontinuous with closed values, then the solution operator $F \mapsto \Phi$ is lower-semicomputable; more precisely, $F \mapsto \Phi$ is $(\mu_{<}; \phi_{<})$ -computable.*

In this paper, we will usually consider the computability of the solution operator Ψ of a hybrid system H . We will sometimes use the terminology “ Ψ is computable” or “ Ψ is computable from H ” instead of the more precise “the operator $H \mapsto \Psi$ is computable”.

4 Discontinuity in the solution of hybrid automata

Let H be a hybrid system, X_0 be a set of initial states, and T a set of times. We wish to compute the set of points reachable under the evolution of H starting at X_0 for times in T . In other words, we wish to compute the operator $(H, X_0, T) \mapsto \Psi^H(X_0, T)$. Note that this problem includes the problem of computing simulations, in which case we take $X_0 = \{x_0\}$ and $T = \{t\}$ to be singleton sets.

4.1 Hybrid Automata

A simple class of hybrid system is that of *hybrid automata*.

Definition 4.1. A hybrid automaton is a tuple (X, G, f, r) where X is a space, $G \subset X$ is a *guard set*, f is a vector field on X and $r : X \rightarrow X$ is a *reset map*.

The evolution of a hybrid automaton proceeds roughly as follows. The solution evolves according to the differential equation $\dot{x} = f(x)$ as long as $x(t) \notin G$. As soon as $x(t)$ enters G , a *discrete event* occurs and the state is reset to $r(x)$. If $r(x) \in G$, then a further discrete event occurs without any prior continuous evolution.

Typically, the state space X of a hybrid automaton is of the form $X = \bigcup_{i=1}^k \{q_i\} \times X_i$, where q_i is the *mode* of the system and X_i the *continuous state space* corresponding to mode q_i . A state is denoted (q, x) where q is the *discrete state* of the system, and x is the *continuous state*.

As we shall see, hybrid automata are sufficiently rich to allow us to find discontinuous dynamics.

4.2 Temporal discontinuities

We first give a trivial example to show that the evolution may vary discontinuously in time.

Example 4.2. Let H be the hybrid automaton with two modes, q_1 and q_2 , with $X_1 = \mathbb{R}$ and $X_2 = \mathbb{R}^0 = \{0\}$. The dynamics in X_1 is constant, $\dot{x} = c$. There is a single event e with reset map $r(q_1, x) = (q_2)$ which is activated when $x > a$.

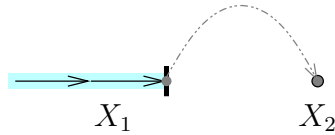


Figure 3: A hybrid automaton with two discrete modes and piecewise-constant dynamics exhibiting a temporal discontinuity.

Let the initial condition be $x(0) = x_0$. Then if $x_0 > a$, the event e is immediately activated and the final state is (q_2) for all $t > 0$. If $x_0 < a$ and $c > 0$, then the event e is activated when $t = t_1 = (a - x_0)/c$. Hence for $t < t_1$, then state is $(q_1, x_0 + ct)$, and for $t > t_1$, the state is (q_2) . Hence the evolution is discontinuous in t .

Of course, time discontinuities are the essence of hybrid automaton dynamics. In Section 5 we see that temporal discontinuities can be handled as long as they do not occur at the final evolution time, and are not also associated with spacial discontinuities.

4.3 Spacial discontinuities

We now give several examples to show that the evolved sets vary *discontinuously* with system parameters and initial condition, even when no transition occurs at the final evolution time.

Example 4.3 (Discontinuity induced by tangency with guard set). Let H be a hybrid automaton with two modes q_1 and q_2 , with $X_1 = \mathbb{R}^2$ and $X_2 = \mathbb{R}^0$. The dynamics in X_1 is affine, $(\dot{x}, \dot{y}) = (2y, -1)$. There is a single reset map with $r(q_1, x, y) = (q_2, 0)$ which is

activated when the constraint c_1 given by $x \geq a$. The solution to the continuous dynamics in mode q_1 is $(x(t), y(t)) = (x_0 + 2y_0t - t^2, y_0 - t)$. The maximum value of x is $x_0 + y_0^2$ and is attained when $t = y_0$.

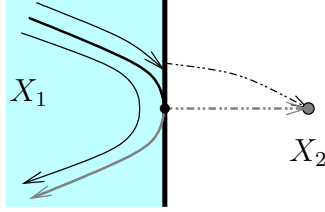


Figure 4: A hybrid automaton with two discrete modes and piecewise-affine dynamics exhibiting a grazing discontinuity.

Suppose the initial condition is (q_1, x_0, y_0) with $x_0 = -1$ and $y_0 = +1$. Then $x(t)$ reaches a maximum value of 0 at $t = 1$. Consider the set $\Psi^H((x_0, y_0), t = 2)$. Then if $a > 0$, the constraint c_1 is not satisfied, and the reached state is $(q_1, -1, -1)$. However, if $a < 0$, the constraint c_1 is satisfied for some $t < 1$, and the state at time $t = 2$ is (q_2) . Hence the evolution is discontinuous in the parameter a .

Now suppose that a is fixed at 0, and the initial condition is $(x_0, 1)$ with $x_0 < 0$. Then for $x_0 < -1$, the maximum value of x is $1 + x_0$ which is less than a , so the constraint is never active and the reached state is $(q_1, x_0, -1)$. However, if $x_0 > -1$, the constraint c_1 is satisfied for some $t < 1$ and the reached state is (q_2) . Hence the evolution is discontinuous in the parameter a .

Example 4.4 (Discontinuity induced by corner collisions). Let H be a hybrid automaton with three modes q_1, q_2 and q_3 , with $X_1 = \mathbb{R}^2$ and $X_2 = X_3 = \mathbb{R}^0$. The dynamics in X_1 has constant derivative, $(\dot{x}, \dot{y}) = (1, 1)$. There are two events, e_2 and e_3 , with reset maps, r_2 and r_3 with $r_2(q_1, x, y) = (q_2)$ and $r_3(x, y) = (q_3)$, and activations c_2 which is activated when $x \geq a$, and c_3 which is activated when $y \geq b$.

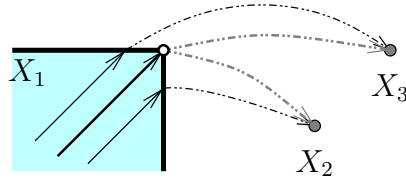


Figure 5: A hybrid automaton with three discrete modes, affine guard sets and piecewise-constant dynamics exhibiting a corner discontinuity.

Suppose the initial condition is (q_1, x_0, y_0) with $x_0 = y_0 = 0$. Then if $0 < a < b < 1$, the event e_2 is activated before e_3 , and the state at time $t = 1$ is (q_2) . If $0 < b < a < 1$, then event e_3 is activated before e_2 , and the state at time $t = 1$ is (q_3) . Hence the evolution is discontinuous in the parameters a and b . In a similar way, we can show that the evolution is discontinuous in the initial state.

Example 4.5 (Discontinuity induced by immediately activated events). Let H be a hybrid automaton with three modes q_1, q_2 and q_3 , with $X_1 = \mathbb{R}^2$, $X_2 = \mathbb{R}$ and $X_3 = \mathbb{R}^0$. The dynamics in X_1 has constant derivative, $(\dot{x}, \dot{y}) = (1, 0)$, and the dynamics in X_2 is $\dot{x} = 1$.

The event e_1 may occur in mode q_1 , with activation $x \geq a$ and reset $r_1(q_1, x, y) = (q_2, x + y)$. The event e_2 may occur in mode q_2 , with activation $x \leq 0$ and reset $r_2(q_2, z) = (q_3)$.

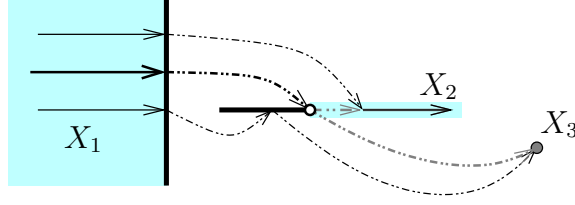


Figure 6: A hybrid automaton with three discrete modes, affine guard sets and piecewise-constant dynamics exhibiting a discontinuity caused by an immediately activated event.

Suppose the initial condition is (q_1, x_0, y_0) with $x_0 = -1$ and $y_0 = 0$. Then the event e_1 is activated at $(q_1, a, 0)$ and the state is reset to (q_2, a) . If $a < 0$, event e_2 is immediately activated, and a transition occurs to state (q_3) . If $a > 0$, then the continuous state z in mode q_2 satisfies $z \geq a > 0$, and so event e_2 is never activated, and the state at time t for $t > 1$ is $(q_2, t - 1)$. Hence the evolution is discontinuous in the parameters.

If the initial state is $(q_1, -1, y_0)$, then the event e_1 is activated at (q_1, a, y_0) and the state is reset to (q_2, z) with $z = a + y_0$. If $y_0 > -a$, the state remains in mode q_2 whereas if $y_0 < -a$, then $z < 0$ and event e_2 is immediately activated and the state is reset to (q_3) . Hence the evolution is discontinuous in the initial state.

4.4 Coherent semantics of evolution

We have seen that the evolution operator $\Psi : X \times \mathbb{R}^+ \rightarrow X$ of a non-Zeno hybrid automaton may be discontinuous in both space and time, even for affine systems. By the fundamental theorem of computable analysis, this means that the evolution is uncomputable, at least near the discontinuity points. This does not in itself rule out the possibility of regularising the evolution in some way so that the evolution becomes computable. In Section 5 we shall show that by using appropriately-defined nondeterministic semantics, we can make the evolution semicomputable. In this subsection we prove that it is impossible to regularise the evolution near continuity points to make the solution fully computable i.e. both lower- and upper-semicomputable.

Definition 4.6 (Coherent semantics of evolution). Let $H = (X, R, \Phi)$ be a hybrid automaton, and let $U \subset X \times \mathbb{R}^+$ be the domain of continuity of the solution operator $\Psi : X \times \mathbb{R}^+ \rightarrow X$. We say that a set-valued solution operator $\widehat{\Psi} : X \times \mathbb{R}^+ \rightrightarrows X$ has *coherent semantics* if $\widehat{\Psi}(x, t) = \{\Psi(x, t)\}$ for all $(x, t) \in U$.

In other words, away from discontinuities, the solution operator $\widehat{\Psi}$ must be single-valued, with the value given by Ψ . This condition eliminates trivial approximations, such as taking $\widehat{\Psi}(x, t) = X$ for all $x \in X, t \in \mathbb{R}^+$. For maximum flexibility, we give no restrictions on the discontinuity set.

Theorem 4.7 (Uncomputability of the evolution of hybrid automata). *Let \mathcal{H} be a class of hybrid automata. Then for any coherent semantics of evolution, the finite-time evolution of a hybrid system is uncomputable. This result holds even if we restrict to (x, t) -values for which no event occurs at time t .*

In particular, the operator $(X_0, t) \mapsto \Psi^H(X_0, t)$ is not $(\kappa, \rho; \kappa)$ -computable. Further, even if no event is possible at time t , the operator $x \mapsto \Psi^H(x, t)$ is not $(\kappa; \kappa)$ -computable.

The result is immediate from the following general lemma, since we have seen examples for which the evolution has non-removable discontinuities, even away from discrete events.

Lemma 4.8. *Let $f : U \rightarrow Y$ be single-valued and continuous on an open, dense subset U of X , and let Y be compact. Suppose f has no continuous closed-valued extension over X . Then f has no continuous multivalued extension F over X .*

Proof. For let x be an essential discontinuity point of f , and $A = \bigcap_{x \in V} \text{cl}(f(V \cap U))$ over open sets V . Suppose $F(x) \subsetneq A$, let $y \in A \setminus F(x)$, and take a closed neighbourhood B of y such that $A \cap B = \emptyset$. Then $F^{-1}(B)$ does not contain x , but contains points arbitrarily close to x , so F would not be upper-semicontinuous. Suppose $F(x) \supset A$ and A has two distinct elements y and z . Let W be an open neighbourhood of y such that $\text{cl}(W)$ is disjoint from z . Then $F^{-1}(W)$ contains x but does not contain points in $F^{-1}(X \setminus \text{cl}(W))$ which come arbitrarily close to x , so F is not lower-semicontinuous. \square

4.5 Sliding along switching boundaries

A particularly nasty form of discontinuity occurs when a solution slides along the boundary of a guard set before crossing.

Example 4.9 (Uncomputability caused by sliding). Consider a hybrid automaton in two-dimensions with a transition which is active for $y \geq 0$. Consider the flow $\dot{x} = 1$, and

$$\dot{y} = \begin{cases} a + 3x^2 - y & \text{if } x \leq 0; \\ a - y & \text{if } 0 \leq x \leq b; \\ a + 3(x - b)^2 - y & \text{if } x \geq b \end{cases} .$$

For $a = 0$, and let (x_0, y_0) be a point with $x_0 < 0$ such that the continuous orbit starting at (x_0, y_0) exactly reaches the point $(0, 0)$. Then for $b > 0$, the continuous evolution starting at (x_0, y_0) slides along the surface $y = 0$ for $0 \leq x \leq b$, and then crosses into $y > 0$. The hybrid orbit therefore undergoes a discrete transition at some point $(x, 0)$ with $0 \leq x \leq b$, but the exact value of x at which this transition occurs is undetermined. For $a > 0$, we see that $\dot{y} > 0$ when $y = 0$, and the orbit starting at (x_0, y_0) undergoes a discrete transition with $x < 0$, whereas for $a < 0$, the orbit starting at (x_0, y_0) undergoes a discrete transition with $x > b$. Hence the spacial evolution is discontinuous at the parameter value $a = 0$. Since for a lower-approximation to the solution we may only consider solutions which persist under perturbations, the hybrid evolution starting at (x_0, y_0) cannot be continued past the point $(0, 0)$.

Now consider the case $a = 0$ and $b = 0$, which is the limit of the cases $a = 0$, $b > 0$. Since the hybrid orbit starting at (x_0, y_0) is blocked at $(0, 0)$ for $b > 0$, in the limit $b = 0$, the orbit cannot be continued past $(0, 0)$ when computing lower-approximations. However, the dynamics in this case is given by the differential equation $(\dot{x}, \dot{y}) = (1, 3x^2 - y)$, so all solutions which reach $y = 0$ cross topologically transversely. This implies that topological transversality of crossing a guard set is not in itself sufficient to ensure that a discrete transition is enabled at the crossing point.

Now consider the flow $(\dot{x}, \dot{y}) = (1, 0)$ the guard set $x = y$ and reset map $(x, y, q_0) \mapsto (y, q_1)$. The flow is transverse to the guard set, and if the initial state is (x, c, q_0) with $x < y$, then after the first reset the new state is (c, q_1) . However, it is possible to make a C^0 perturbation of the guard set, so that the flow is parallel to the guard set for $y = a$. By the previous discussion, this means that the evolution of the perturbed hybrid system

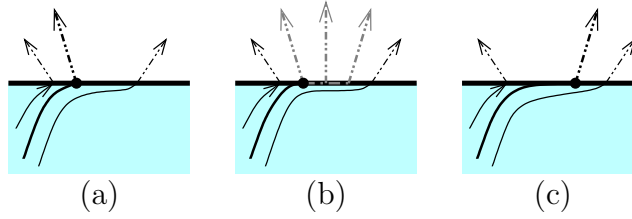


Figure 7: Sliding along a guard set. The discontinuity in (b) can be perturbed to give a continuous system in (a) and (c), but the evolution of the original discontinuity point depends on the perturbation.

cannot undergo a discrete transition for initial conditions with $y_0 = a$. Since we wish to compute lower approximations to the flow which persist under perturbation, this means that the evolution of the original hybrid system cannot undergo a discrete transition if $y_0 = a$. Since the above argument holds for arbitrary a , the evolution of the original hybrid system cannot undergo a discrete transition at any point of the guard set.

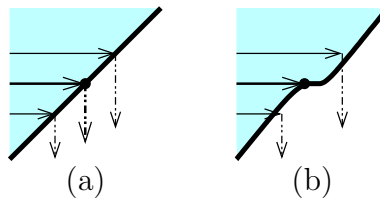


Figure 8: A C^0 -perturbation of the guard set at a transverse crossing (a) can result in sliding at any point (b). The evolution cannot be continued, since we cannot rule out the presence of sliding.

We have therefore demonstrated that, at least without additional information on the behaviour, the evolution of a hybrid system cannot be allowed undergo a discrete transition at a crossing of a guard set if we are to compute lower-approximations to the evolution which are robust with respect to perturbations. However, the above situation is pathological, in the sense that “most” systems do not exhibit this kind of sliding behaviour. Further, transverse crossings are generic for hybrid systems with differentiable flows (such as from a Lipschitz differential equation) and differentiable guard sets, and in the C^1 topology, transverse crossings cannot be perturbed away. This suggests that this pathological behaviour can be treated numerically by computing derivatives, and this is indeed the case. However, trajectories which slide along the guard set can occur even in C^r flows with C^r guard sets in the neighbourhood of a C^r singularity, and such singularities occur generically in r -dimensional hybrid systems. Hence even taking higher-order derivatives might not be enough in some cases.

In this paper, we resolve the difficulty by giving a topological definition of a “detectable” crossing (which is weaker than topological transversality), and show that if we restrict to systems with detectable crossings, then it is possible to compute the evolution. It is possible to prove that crossings are detectable numerically by computing derivatives of the flow and guard set.

In the above example, a discontinuity in the evolution resulting in a loss of lower-semicomputability can occur at a degree- d crossing if perturbations of order $d - 1$ are

allowed. Hence, a purely topological approach to lower-approximations in systems with crossings of guard sets is bound to fail.

5 Semicontinuity of evolution of hybrid systems

We now introduce a class of nondeterministic hybrid systems and consider conditions under which the evolution is semicomputable.

5.1 Nondeterministic hybrid systems

In this paper, we use the following definition of hybrid system, which slightly extends that of [4], and is essentially equivalent to that used in [19].

Definition 5.1 (Hybrid system). A *hybrid system* is a tuple $H = (X, D, A, \Phi, R)$ where

- the state space X is a manifold,
- $D \subset X$ is the *domain set*,
- $A \subset X$ is the *activation set*.
- $\Phi : X \rightrightarrows C(\mathbb{R}^+ \dashrightarrow X)$ is a multivalued *flow*, and
- $R : X \rightrightarrows X$ defines a *reset map* $x' \in R(x)$.

In typical examples, the flow will be defined by a differential equation $\dot{x} = f(x)$ or differential inclusion $\dot{x} \in F(x)$.

Note that instead of working with differential inclusions, we work directly with flows, since this separates the core hybrid systems theory (e.g. detecting crossings with guard sets) from the technicalities of integrating with differential inclusions. If A and D form a topological partition of X (i.e. $D \cup A = X$ and $D^\circ \cap A^\circ = \emptyset$, where $^\circ$ denotes the interior of a set), and Φ is given by the differential inclusion $\dot{x} \in F(x)$ then $H = (X, A, F, R)$ is an *impulse differential inclusion* as defined in [4].

Definition 5.2 (Trajectories of hybrid systems). A trajectory or solution of a hybrid system $H = (X, D, A, \Phi, R)$ is a hybrid trajectory $\xi : \mathcal{T} \rightarrow X$ such that

- $\xi(t, n) \in D$ whenever $t_n \leq t \leq t_{n+1}$,
- $\xi(t_n, n-1) \in A$,
- $\xi(t, n) = \eta|_{[0, t_{n+1}-t_n]}(t-t_n)$ for some $\eta \in \Phi$, and
- $\xi(t_n, n) \in R(\xi(t_n, n-1))$.

Note that even if Φ and R are single-valued, then the evolution can still be nondeterministic. For if $x \in D \cap A$, then both continuous evolution and a discrete jump may be possible starting from x .

In this section, we will make regular use of the set-valued indicator function $I_S : X \rightrightarrows X$ defined by $I_S(x) = \{x\}$ if $x \in S$, and $I_S(x) = \emptyset$ if $x \notin S$. It is straightforward to

show that if A is closed, then I_A is upper-semicontinuous, and if U is open, then I_U is lower-semicontinuous. We will also use the *restricted reset map* $R|_A$ defined by

$$R|_A(x) := R(I_A(x)) = \begin{cases} R(x) & \text{if } x \in A \\ \emptyset & \text{if } x \notin A. \end{cases} \quad (2)$$

and the *restricted dynamic* $\Phi|_D : X \times \mathbb{R}^+ \rightrightarrows \mathbb{R}^+$ given by

$$\Phi|_D(x, t) := \{y \mid \exists \eta \in \Phi \text{ s.t. } \eta(0) = x, \eta([0, t]) \subset D \text{ and } \eta(t) = y\}. \quad (3)$$

5.2 Upper-semicomputability of the evolution

We now consider upper-semicomputability of the evolution of set-based hybrid systems.

Definition 5.3. A set-based hybrid system $H = (X, D, A, \Phi, R)$ is *upper-semicontinuous* if

- D and A are closed,
- Φ and R are upper-semicontinuous with compact values.

Note that by saying Φ is upper-semicontinuous with compact values, it follows that if ξ_n is a sequence of solutions $\xi_n \in \Phi(x_n)$ with $\lim_{n \rightarrow \infty} x_n = x_\infty$, then there is a subsequence of the ξ_n converging to a curve ξ_∞ .

Upper-semicontinuity of the solution of upper-semicontinuous hybrid systems was proved in [19]. In this work, we prove that the solution operator is also upper-semicomputable, as stated in [16].

Theorem 5.4 (Upper-semicomputability for nondeterministic hybrid systems). *The evolution of an upper-semicontinuous hybrid system is upper-semicomputable.*

More precisely, let $H = (X, D, A, \Phi, R)$ be a hybrid system such that D and A are closed, and Φ and R are upper-semicontinuous with compact values. Let $X_0 \subset X$ be a compact initial state set, $T \subset \mathbb{R}^+$ a compact set of times, and N a bound on the number of events. Then the operator $(D, A, \Phi, R) \mapsto \Psi^H$ is $(\psi_>, \psi_>, \phi_>, \mu_>; \mu_>)$ -computable.

Lemma 5.5. *Let R be a compact-valued upper-semicontinuous map, and A a closed set. Then the operator $(R, A) \mapsto R_A$ is $(\mu_>, \psi_>; \mu_>)$ -computable.*

Proof. For any compact set, $R_A(C) = R(C \cap A)$, so the image is computable. \square

Lemma 5.6. *Let Φ be an upper-semicontinuous compact-valued flow, and D a closed set. Then the operator $(\Phi, D) \mapsto \Phi|_D$ is $(\phi_>, \psi_>; \phi_>)$ -computable.*

Proof. Consider the restricted flows $\Phi_{D,n}$ given by $\eta \in \Phi_{D,n} \iff \xi \in \Phi$ and $\eta(t) \in D$ for $t \in \{m/2^n \mid m = 0, 1, \dots, 2^{2^n}\}$. Clearly, any trajectory in $\Phi_{D,n+1}$ lies in $\Phi_{D,n}$, so the allowable orbits form a monotone decreasing set. Further, since $\Phi(X_0)$ is a compact subset of $C(\mathbb{R}^+ \dashrightarrow X)$ for any compact set of initial states X_0 , and $C(\mathbb{R}^+ \dashrightarrow X)$ is a Polish space, $\Phi(X_0)$ is sequentially-compact. Hence if η_n is a sequence of solutions of Φ such that $\eta_n(0) \in X_0$ and $\eta_n \in \Phi_{A,n}$, then there is a subsequence η_{n_i} which converges to a continuous function η_∞ . Since D is closed and $\eta_i(m/2^n) \in D$ for all i sufficiently large, we have $\eta_\infty(m/2^n) \in D$. Since η_∞ is continuous and $\{m/2^n \mid m, n \in \mathbb{N}\}$ is dense in \mathbb{R}^+ , we have $\eta(t) \in D$ for all t . Hence $\Phi|_D(x) = \bigcap_{n \in \mathbb{N}} \Phi_{D,n}(x)$.

It remains to show that each $\Phi_{D,n}$ is computable. It is sufficient to consider $\{\eta \mid \xi(t) \in D\}$ is a computable closed set in D for all $t \in \mathbb{R}^+$, since then we can write $\Phi_{D,n}(x_0) = \Phi(x_0) \cap \bigcap_{m=0}^{2^{2n}} \{\xi \mid \eta(m/2^n) \in D\}$. This follows since

$$\{\eta \mid \eta(t) \notin D\} = \bigcup_{\{(T,J) \in \beta_{\mathbb{R}} \times \beta_X \mid \bar{J} \cap D \neq \emptyset \text{ and } t \in T\}} \{\eta \mid \eta(\bar{T}) \subset \bar{J}\}$$

and

$$\{\eta \in \Phi \mid \eta(\bar{T}) \subset \bar{J}\} = \text{cl}(\{\eta \in \Phi \mid \eta(\bar{T}) \subset J\})$$

Hence $\{\eta \mid \eta(t) \in D\}$ is the complement of $\bigcup_{(J,T) \mid \bar{J} \cap D \neq \emptyset \text{ and } t \in T} \text{cl}(\{\eta \mid \eta(\bar{T}) \subset J\})$, which can be computed from D . \square

We can now give the proof of Theorem 5.4.

Proof. By Lemmas 5.5 and 5.6, we can obtain a $\mu_{>}$ -name of the restricted reset $R|_A$ and a $\phi_{>}$ -name of the restricted flow $\Phi|_D$. It remains to compute a $\mu_{>}$ -name of the evolution Ψ .

Consider the case $D = A = X$, so $\Phi|_D = \Phi$ and $R_A = R$. Let $\Psi_h(x) := \Phi|_D(x, t)$. Then we can obtain a $\mu_{>}$ name of $(x, t) \mapsto \Phi(x, t)$. Define

$$\begin{aligned} \Psi(x, t; t_1, \dots, t_n) &= \{y \in X \mid \exists \text{ solution } \xi \text{ of } H \text{ with event times } t_1 \leq \dots \leq t_n \leq t \\ &\quad \text{s.t. } \xi(0, 0) = x \text{ and } \xi(t, n) = y\}. \end{aligned}$$

Then since we can write

$$\Psi(x, t, (t_1, \dots, t_n)) = \Phi_{t-t_n} \circ R \circ \Phi_{t_n-t_{n-1}} \circ \dots \circ R \circ \Phi_{t_1}(x)$$

we see that the map $(x, t, t_1, \dots, t_n) \mapsto \Psi(x, t, (t_1, \dots, t_n))$ is a composition of functions for which we have $\mu_{>}$ -names, and hence we can compute $\mu_{>}$ -name of Ψ . We can write $\Psi(x, t, n) = \Psi(x, t, T_{t,n})$ where $T_{t,n} = \{(t_1, \dots, t_n) \in \mathbb{R}^n \mid \forall i, 0 \leq t_i \leq t_{i+1} \leq t\}$. Since $t \mapsto T_{t,n}$ is $(\rho, \kappa_{>})$ -computable, we can compute a $\mu_{>}$ -name of the function $(x, t, n) \mapsto \Psi(x, t, n)$. Then $\Psi(x, t, [0, N]) = \bigcup_{n=0}^{\infty} \Psi(x, t, n) = \bigcup_{n=0}^N \Psi(x, t, n)$ is a finite union of $\mu_{>}$ -computable functions. Hence we can compute a $\mu_{>}$ -name of Ψ . \square

We say a system is *uniformly non-Zeno* if there exist (T, N) such that in any time interval of length at most T , there occur at most N discrete events. As shown in [19], any non-Zeno upper-semicontinuous hybrid system with a compact global attractor must be uniformly non-Zeno. For non-Zeno systems, we can drop the bounds on the number of events.

Corollary 5.7 (Upper-semicomputability for non-Zeno hybrid systems). *Let $H = (X, D, A, \Phi, R)$ be a uniformly non-Zeno upper-semicontinuous hybrid system. Let $X_0 \subset X$ be a compact initial state set, and $T \subset \mathbb{R}^+$ a compact set of times. Then the operator $(D, A, \Phi, R, X_0, T) \mapsto \Psi^H(X_0, T)$ is $(\psi_{>}, \psi_{>}, \phi_{>}, \mu_{>}; \mu_{>})$ -computable.*

5.3 Lower-semicomputability of evolution

Definition 5.8. A hybrid system $H = (X, D, A, \Phi, R)$ is *lower-semicontinuous* if

- D and A are open, and
- Φ and R are lower-semicontinuous with closed values.

In this situation, we have the following computability result.

Theorem 5.9. *The evolution of a lower-semicontinuous domain-activation hybrid system is lower-semicomputable.*

More precisely, let $H = (X, D, A, \Phi, R)$ be a hybrid system, where D and A are open, Φ is a lower-semicontinuous multivalued flow, and $R : X \rightrightarrows X$ is lower-semicontinuous. Let $X_0 \subset X$ be closed and $T \subset \mathbb{R}^+$ be closed. Then the operator $(D, A, \Phi, R) \mapsto \text{cl}\Psi^H$ is $(\theta_<, \theta_<, \theta_<, \mu_<; \mu_<)$ -computable. Equivalently, the operator $(D, A, \Phi, R, X_0, T) \mapsto \text{cl}\Psi^H(X_0, T)$ is $(\theta_<, \theta_<, \theta_<, \mu_<, \psi_<, \psi_<; \psi_<)$ -computable

The basic idea of the proof is as follows. We let h be a time step, and consider all trajectories of H such that discrete events are constrained to occur at times kh with $0 < kh < t$. We show that the evolution defined by this semantics is lower-semicomputable, and that in the limit as $h \rightarrow \infty$ we obtain all trajectories. It is important that we do not allow a discrete transition to occur at the initial or final time of the evolution.

Lemma 5.10. *Let R be a closed-valued lower-semicontinuous map, and U an open set. Then the operator $(R, U) \mapsto R|_U$ is $(\mu_<, \theta_<; \mu_<)$ -computable.*

Proof. From the definition of I_U , we have $\bar{J} \subset I_U^{-1}(K)$ iff $\bar{J} \subset U \cap K$ so the function $U \mapsto I_U$ is $(\theta_<; \mu_<)$ -computable. The result follows since $R_U = R \circ I_U = \text{cl}(R \circ I_U)$, and composition of functions is a lower-semicomputable operation. \square

Lemma 5.11. *Let Φ be an lower-semicontinuous closed-valued flow, and D an open set. Then the operator $(\Phi, D) \mapsto \Phi|_D$ is $(\phi_<^A, \theta_<; \phi_<^A)$ -computable.*

Proof. We first show that for all $t > 0$, $\{\eta \mid \eta([0, t]) \subset D\}$ is computable. We see that for fixed η and $t \in \mathcal{Q}$ that $\eta([0, t]) \subset D \iff \exists 0 = t_0 < t_1 < \dots < t_k = t, J_1, \dots, J_k$ with $\bar{J}_i \subset D$ such that $\eta([t_{i-1}, t_i]) \subset J_i$. so we can write

$$\{\eta \mid \eta([0, t]) \subset D\} = \bigcup_{\{(t_i, J_i) \in (\mathcal{Q} \times \beta)^* \mid \bar{J}_i \subset D\}} \bigcap \{\eta \mid \eta([t_{i-1}, t_i]) \subset J_i\}$$

which is a computable (from a $\theta_<$ -name of D) countable union of finite intersections of basic open sets.

Hence $\Phi|_D$ is the closure of the intersection of Φ with the union of partial trajectories such that $\eta([0, t_n]) \subset D$, so is computable. \square

We now present the proof of Theorem 5.9

Proof. First consider the case $D = A = X$, so that $\Phi = \Phi|_D$ and $R = R|_A$. Note that from a $\phi_<$ -name of Φ as a map $X \rightrightarrows C(\mathbb{R}^+ \dashrightarrow X)$, we can compute a $\mu_<$ -name of Φ as a map $X \times T \rightrightarrows X$. Define $\Psi(x, t; t_1, \dots, t_n)$ and $T_{t,n}$ as in the proof of Theorem 5.4. Then the map $t \mapsto T_{t,n}$ is lower-semicomputable i.e. $(\rho; \psi_<)$ -computable, and since $(x, t, t_1, \dots, t_n) \mapsto \Psi(x, t; t_1, \dots, t_n)$ is a composition of maps of the form $\Phi_{t_i - t_{i-1}}$ and R , for which we have $\mu_<$ -names, we can compute a $\mu_<$ -name of $t \mapsto \Psi(x, t)$. Hence we can compute a $\mu_<$ -name of the closed composition $(x, t) \mapsto \text{cl}(\Psi(x, t))$.

The general case follows from the fact that $(D, \Phi) \mapsto \Phi|_D$ and $(A, R) \mapsto R|_A$ are lower-semicomputable i.e. respectively $(\theta_<, \phi_<; \phi_<)$ -computable and $(\theta_<, \mu_<; \mu_<)$ -computable. \square

The following result will be useful in Section 5.6. It shows that the evolution of H is the same as the evolution we obtain by considering only trajectories with distinct event times. Indeed, any solution of H is the limit of solutions with distinct event times. Formally, let H be a hybrid system and define $\tilde{\Psi}$ by

$$\tilde{\Psi}(x, t, n) = \{y \mid \exists \text{ solution } \xi \text{ of } H \text{ with event times } 0 < t_1 < t_2 < \dots < t_n < t \\ \text{such that } \xi(0) = x \text{ and } \xi(t) = y\}. \quad (4)$$

Proposition 5.12. *Let Ψ be the evolution of a lower-semicontinuous hybrid system H , and let $\tilde{\Psi}$ be the evolution of H consisting of trajectories with distinct event times. Then $\text{cl}(\tilde{\Psi}(x, t)) = \text{cl}(\Psi)(x, t)$.*

Proof. Define $U_{t,n} = \{(t_1, \dots, t_n) \in \mathbb{R}^n \mid 0 < t_1 < \dots < t_n < t\}$. Then $\tilde{\Psi}(x, t) = \Psi(x, t, U_{t,n})$ where $\Psi(x, t, (t_1, \dots, t_n))$ is as defined previously. Since $\Psi : X \times \mathbb{R}^+ \times (\mathbb{R}^+)^n \rightrightarrows X$ is lower-semicontinuous, we have $\text{cl}(\tilde{\Psi}(x, t, n)) = \text{cl}(\Psi(x, t, U_{t,n})) = \text{cl}(\Psi(x, t, \bar{U}_{t,n})) = \text{cl}(\Psi(x, t, T_{t,n})) = \text{cl}(\Psi(x, t, n))$ as required. \square

5.4 Closure and interior semantics of evolution

General hybrid systems of the form given by Definition 5.1 need not be upper- or lower-semicontinuous. In order to compute upper- or lower approximations to the solution, we need to convert the system into either upper- or lower-semicontinuous form. We can do this by regularising the guard sets to be open or closed, and

Definition 5.13. Let $F : X \rightrightarrows Y$ be a multivalued function. Define

- \bar{F} by $\bar{F} = \bigcup \{\hat{F} \mid \hat{F} \text{ is upper-semicontinuous and } \hat{F} \subset F\}$, and
- \underline{F} by $\underline{F} = \bigcup \{\tilde{F} \mid \tilde{F} \text{ is lower-semicontinuous and } \tilde{F} \subset F\}$.

An alternative definition of \bar{F} is in terms of its graph; $\text{Graph}(\bar{F}) = \bigcap_{\epsilon > 0} N_\epsilon(\text{Graph}F)$. It is easy to show that if F locally takes pre-compact values (i.e. $\text{cl}(F(\bar{I}))$ is compact for any compact \bar{I}) then \bar{F} is compact-valued upper-semicontinuous, and that \underline{F} is closed-valued lower-semicontinuous.

Definition 5.14. Let $H = (X, D, A, F, R)$ be a set-based hybrid system. Then

- $\xi : \mathcal{T} \rightarrow X$ is a trajectory of H using *closure-semantics* if ξ is a trajectory of the upper-semicontinuous system $\bar{H} = (X, \text{cl}(D), \text{cl}(A), \bar{F}, \bar{R})$.
- $\xi : \mathcal{T} \rightarrow X$ is a trajectory of H using *interior-semantics* if ξ is a trajectory of the lower-semicontinuous system $\underline{H} = (X, D^\circ, A^\circ, \underline{F}, \underline{R})$.

From Theorems 5.4 and 5.9, we deduce

Corollary 5.15. *Let $H = (X, D, A, F, R)$ be a set-based hybrid system. Then the evolution of \bar{H} is upper-semicomputable and the evolution of \underline{H} is lower-semicomputable.*

We now show that \bar{H} is “smallest” hybrid system for which the evolution is upper-semicomputable. In other words, *any* attempt to compute an over-approximation to the evolved set using approximative methods must necessarily compute an over-approximation to the evolved set of \bar{H} .

We let \mathcal{H} be the space of hybrid systems, where D and A are in the space of regular sets with both the lower representation $\theta_<$ and upper representation $\psi_>$, Φ is in the space of compact-valued flows with representation $\phi = \phi_< \vee \phi_>$, and R is in the space of compact-valued maps with representation $\mu_< \vee \mu_>$. In other words, D is a regular set, and we have access to a list of boxes filling D° , and a list of boxes filling $X \setminus \text{cl}(D)$.

Theorem 5.16. *Let $H = (X, D, A, \Phi, R)$ be a hybrid system, and suppose that $\Psi : \mathcal{H} \times X \times \mathbb{R} \mapsto \mathcal{K}(X)$ is upper-semicomputable and $\Psi(x, t) \supset \Psi^H(x, t)$ for all x, t . Then $\Psi(x, t) \supset \Psi_{\underline{H}}(x, t)$.*

We have a similar result for lower-semicontinuity.

Theorem 5.17. *Let $H = (X, D, A, \Phi, R)$ be a hybrid system, and suppose that $\Psi : X \times \mathcal{H} \times \mathbb{R} \mapsto \mathcal{A}(X)$ is lower-semicomputable and $\Psi(x, t) \subset \text{cl}(\Psi^H(x, t))$ for all x, t . Then $\Psi(x, t) \subset \text{cl}(\Psi_{\underline{H}}(x, t))$.*

The significance of these results is that, in general, it is impossible to do better than compute over- or lower- approximations to the evolution which converge to smaller sets than those given by the upper or lower semantics, as long as only approximate information is used.

5.5 Deficiencies of interior semantics

Unfortunately, the definition of interior-semantics given causes difficulties in the modelling of systems with urgent transitions. This is because there is no way in the formalism of specifying a coupling between the invariants and activations.

Example 5.18 (Uncomputability caused aliasing). Consider a system with dynamic $\dot{x} = 1$, invariant $x \leq a$ and activation $x \geq b$ with $a, b > 0$. If $a < b$, then the invariant is violated before the transition is activated, and further evolution is blocked. If $a > b$, then the transition is activated before the invariant is violated, and a transition may occur at any time $b < x(t) < a$. If $a = b$ and we are computing an over-approximation to the evolution, then a transition must occur exactly when $x(t) = a$. However, if we are computing a lower-approximation to the evolution, then since equality is undecidable, we need to consider the possibility that $a < b$. Hence a lower-approximation to the evolution must block, since this is the worst-case scenario.

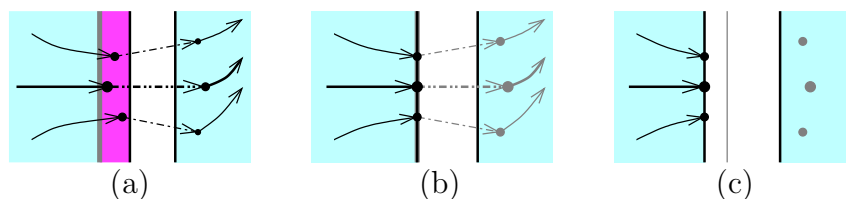


Figure 9: Discrete transitions are blocked using inner semantics even at a transverse crossing. In (a) the domain and activation regions overlap and crossings are possible. In (b) the boundaries of the domain and activation regions touch, and discrete transitions are forced with upper semantics, but disallowed using inner semantics. An arbitrarily small perturbation gives (c) in which no transitions are possible.

At first sight, it may seem that the evolution “should” continue for $a = b$. However, the correct semantics for lower-approximation is to block the evolution. This is because

if the invariant and activation are determined by independent parameters, then it is only a coincidence that the transition is activated at exactly the same point as the invariant is violated, and under a small change in the parameters then the evolution may be blocked. It is only if we give the additional, combinatorial information that the invariant and activation boundaries lie exactly at the same point, that we can deduce that the evolution may continue. From an implementation standpoint, we see that $x \leq a$ and $x \geq b$ are aliases for the same constraint $x \leq c$ with $c = a = b$.

5.6 Lower-semicomputability of hybrid systems using crossing semantics

Let $H = (X, D, A, \Phi, R)$ be a hybrid system and suppose D and A are regular open sets and Φ is closed-valued lower-semicontinuous. We would like to know when trajectories of H cross instantaneously from D to A .

Definition 5.19. A continuous trajectory ξ crosses from D to A at time t and point x if for all $\delta > 0$, $\xi(t - \delta, t) \cap D \neq \emptyset$ and $\xi(t, t + \delta) \cap A \neq \emptyset$. We say that x is a *crossing point* for ξ .

Note that trivially if $x \in D \cap A$, then x is a crossing point for any trajectory though it. If $x \notin \text{cl}(D) \cap \text{cl}(A)$, then x cannot be a crossing point. The real interest is when x lies in ∂D and ∂A . By the observations of Example 5.18, if D and A are disjoint, then by a small perturbation, we can make their boundaries disjoint, and so any lower approximation to the flow will have blocking. We therefore need more information about the sets D and A , and the flow Φ than is given by the names $\theta_{<}$ and $\phi_{<}$.

Let us consider the case in which D and A form a topological partition of X ; that is, $D \cap A = \emptyset$ and $\text{cl}(D) \cup \text{cl}(A) = X$. Suppose ξ is a trajectory such that $\xi(t_1) \in D$ and $\xi(t_2) \in A$, so ξ apparently crosses from D to A . We would like to be able to deduce that ξ crosses from D to A in the sense of Definition 5.19. Unfortunately, from Example 4.9, it may be the case that ξ slides inside $\partial D \cap \partial A$ rather than crossing transversely, and as we have seen, we cannot handle sliding solutions.

Definition 5.20. Let H be a hybrid system and $\delta > 0$. We say H has δ -*detectable crossings* if for all trajectories ξ of Φ such that $\xi(0) \in D$ and $\xi(t) \in A$ for some $t < \delta$, then there exists $c \in \mathbb{R}$ and $\eta \in \Phi$ such that $\xi([0, c]) \subset D$, $\eta(0) = \xi(c)$ and $\eta([0, \epsilon]) \cap A \neq \emptyset$ for all $\epsilon > 0$.

In other words, if there is a trajectory which moves from D to A in time less than δ , then from the point where the state leaves D , there is a possibly different trajectory η which immediately enters A . Note that the condition of detectable crossings precludes the degenerate situation in which a solution slides along a common boundary of D and A for time less than δ , and also the case of Example 5.18 in which the solution leaves D briefly before entering A .

In Section 6.3, we will give conditions under which a system has detectable crossings.

We now define a new notion of solution for hybrid systems.

Definition 5.21. Let $H = (X, D, A, \Phi, R)$ be a hybrid system where D and A are open sets. Then a hybrid trajectory ξ is a solution of H using *crossing semantics* if

- $0 < t_1 < t_2 < \dots < t_i < t_{i+1} < \dots$.
- $\xi[t, t_n) \in D$ for $t_n \leq t < t_{n+1}$.

- There exists $\zeta_n \in \Phi$ such that $\zeta_n(0) = \xi(t_n, n)$ and $\zeta_n([0, \epsilon]) \cap A \neq \emptyset$ for all $\epsilon > 0$.
- $\xi(\cdot, n) \in \Phi$, and
- $\xi(t_n, n) \in R(\xi(t_n, n - 1))$.

Intuitively, between discrete events, solutions must remain in the interior of D ; this prevents grazing contact with guard sets. A discrete event may occur at the boundary of D if it is possible to continue the trajectory directly into A . Note that after a reset, we require either that $x \in D$ or that another discrete transition occurs immediately. Note that we allow $\xi(t_n, n) \notin D$ if another event occurs exactly at time t_n .

Using this notion of solution, we can prove the following result.

Theorem 5.22 (Lower-semicomputability of the evolution of hybrid systems with detectible crossings). *Let H be a lower-semicontinuous hybrid system with δ -detectable crossings. Then the evolution $H \mapsto \Psi_H(x, t)$ is lower-semicomputable using crossing semantics.*

More precisely, let $H = (X, D, A, \Phi, R)$ be a hybrid system where D and A are open sets, and Φ and R are lower-semicontinuous with closed values. Suppose that crossings of Φ from D to A are δ -detectable. Let X_0 be a closed set of initial states, and T is an open set of times. Then the operator $(D, A, \Phi, R) \mapsto \Psi^H$ is $(\theta_<, \theta_<, \phi_<, \mu_<; \mu_<,)$ -computable.

We use the following lemma, which shows that the crossing times and points can be computed.

Lemma 5.23. *Let D and A be open sets and Φ be a lower-semicontinuous closed-valued flow. Suppose that the crossings of trajectories of Φ from D to A are δ -detectable. Define the crossing function $\Gamma : X \rightrightarrows \mathbb{R}^+ \times X$ by*

$$\Gamma(x_0) = \{(t, x) \in \mathbb{R} \times X \mid t > 0 \text{ and } \exists \xi \in \Phi \text{ s.t. } \xi([0, t]) \subset D \text{ and } \xi([0, t + \epsilon]) \cap A \neq \emptyset \text{ for all } \epsilon > 0\}. \quad (5)$$

Then the function $(D, A, \Phi) \mapsto \Gamma$ is $(\theta_<, \theta_<, \phi_<; \mu_<)$ -computable.

Proof. Consider the set of trajectories of Φ which have a first crossing from D to A at time $t \in]t_1, t_2[$ and point $x \in J$, and suppose $t_1 < t_2 < t_1 + \delta$. If η is such a trajectory, then by the definition of crossing, we have $\eta([0, t_1]) \subset D$, $\eta([t_1, t_2]) \subset J$ and $\eta([t_1, t_2]) \cap A \neq \emptyset$. Further, by the δ -detectable crossing condition, this is a sufficient condition for the existence of a crossing at time $t \in]t_1, t_2[$ and point $x \in J$. Hence $\bar{I} \subset \Gamma^{-1}(T \times J)$ if and only if

$$\bar{I} \subset \Phi^{-1}(\{\eta \mid \eta([0, t_1]) \subset D\} \cap \{\eta \mid \eta([t_1, t_2]) \subset J\} \cap \{\eta \mid \eta([t_1, t_2]) \cap A \neq \emptyset\})$$

The set $\{\eta \mid \eta([0, t_1]) \subset D\}$ can be lower-semicomputed from a $\theta_<$ name of D , and the set $\{\eta \mid \eta([t_1, t_2]) \subset J\}$ is a basic open set of the flow. The set $\{\eta \mid \eta([t_1, t_2]) \cap A \neq \emptyset\}$ can be written as $\bigcup_{T \subset]t_1, t_2[} \{\eta \mid \eta(T) \subset A\}$, so is a countable union of lower-semicomputable sets. Hence we can enumerate all tuples $(I, T, J) \in \beta_X \times \beta_R \times \beta_X$ such that for all $x \in \bar{I}$, there exists a trajectory of Φ starting at x such that Φ has a first crossing from D to A at time $t \in T$ and point $y \in J$, which means we have a $\mu_<$ -name of Γ . \square

We can now give the proof of Theorem 5.22

Proof. Define $\tilde{\Psi} : X \times \mathbb{R}^+ \times \mathbb{N}^+ \rightrightarrows X$ to be the evolution of H with crossing semantics. Note that this means that all events must occur at distinct times. Define $\hat{R} : X \times \mathbb{R} \rightrightarrows X \times \mathbb{R}$ by $\hat{R}(x, t) = R(x) \times \{t\}$, and $\Theta((x, s), t) = \Phi(x, t - s)$ for $x \in X$ and $0 < s < t$. Then $\tilde{\Psi}(x, t, 1) = \Theta(\hat{R}(\Gamma(x)), t)$. It is clear that we can compute a $\mu_{<}$ -name of \hat{R} and $\hat{\Theta}$ from $\mu_{<}$ -names of R and Φ , respectively, and by Lemma 5.23 we can compute a $\mu_{<}$ -name of Γ . Hence we can compute a $\mu_{<}$ name of the composition $(x, t) \mapsto \text{cl}(\tilde{\Psi}(x, t, 1))$. The result by induction since $\tilde{\Psi}(x, t, n+1) = \bigcup_{s \in]0, t[} \tilde{\Psi}(\tilde{\Psi}(x, s, b), t - s, 1)$ and $\text{cl}(\tilde{\Psi}(x, t)) = \bigcup_{n=0}^{\infty} \tilde{\Psi}(x, t, n)$ \square

Note that in the case of crossing semantics, it is not true that every trajectory with multiple events at the same time is a limit of trajectories with distinct event times.

6 Modelling, Simulation and Implementation

6.1 Hybrid Automata with guard sets

The description of hybrid automata introduced in Section 4 is sufficient to define the dynamic evolution, but is inexpressive as a modelling framework. Many hybrid system models are defined using explicit *discrete state*, and *discrete events* and allow for *urgent transitions* when a trajectory first touches a *guard set*. Guard sets therefore form both the boundary of the invariant domain of continuous evolution, and of the activation region of the discrete event. We should therefore use crossing semantics for determining the activation of events given by guard sets when computing lower-approximations to the evolution. We can use interior semantics when determining the activation of non-urgent or *permissive* transitions.

In our definition of hybrid systems we will describe sets in terms of *constraint functions*. A *constraint* is a continuous function $c : X \rightarrow \mathbb{R}$, and we say a constraint is *regular* if $\{x \in X \mid c(x) = 0\}$ is a codimension-1 topological manifold. If c is differentiable and $\nabla c(x) \neq 0$ whenever $c(x) = 0$, then c is regular, and changes sign on a differentiable manifold.

A constraint defines sets $\{x \in X \mid c(x) \leq 0\}$ and $\{x \in X \mid c(x) \leq 0\}$. The operator $c \mapsto \{x \mid c(x) < 0\}$ is $(\gamma; \theta_{<})$ -computable and the operator $c \mapsto \{x \mid c(x) \leq 0\}$ is $(\gamma; \psi_{<})$ -computable.

We now give a standard definition of hybrid automata. A transition for a constraint hybrid automaton is either *urgent* (sometimes called *just-in-time*), which means that continuous evolution is not allowed when the event is active, or *non-blocking*, which means that continuous evolution is allowed in addition to a discrete transition, and hence the dynamics is nondeterministic.

Definition 6.1 (Hybrid automaton). A *constraint hybrid system* is a tuple $H = (Q, E = E_U \cup E_P, \gamma, \{X_q \mid q \in Q\}, \{F_q \mid q \in Q\}, \{D_q \mid q \in Q\}, \{(R_{q,e} \mid (q, e) \in \text{dom } \gamma\}, \{(A_{q,e} \mid (q, e) \in \text{dom } \gamma, e \in E_P\}, \{(g_{q,e} \mid (q, e) \in \text{dom } \gamma, e \in E_U\}))$ where

- Q is a finite set of *discrete states* and E is a finite set of *discrete events*, which is partitioned into subsets E_U of *urgent events* and E_P of *permissive events*.
- $\gamma : \subset Q \times E \rightarrow Q$ is a partial *discrete transition function* with domain $\text{dom } \gamma$.
- For each q , the manifold X_q is the *continuous state space* for discrete state q .

- For each q , $F_q : X_q \rightrightarrows TX_q$ is a differential inclusion giving the *continuous dynamics*.
- For each q , $D_q \subset X_q$ is the *invariant domain* for the continuous dynamics.
- For each $(q, e) \in \text{dom } \gamma$, $R_{q,e} : X_q \rightrightarrows X_{\rho(q,e)}$ is the *reset map*.
- For each $(q, e) \in \text{dom } \gamma$ with $e \in E_P$, $A_{q,e} \subset X_q$ is an *activation region*.
- For each $(q, e) \in \text{dom } \gamma$ with $e \in E_U$, $g_{q,e} : X_q \rightarrow \mathbb{R}$ is a *guard constraint*.

Notice that we have two types of restrictions on the continuous dynamics, namely those given by the invariants and those given by the guards. We also have two types of restrictions on the discrete dynamics, namely those given by the activations and those given by the guards. When computing lower-approximations to the evolution, it is appropriate to use interior semantics for invariants and activations, and crossing semantics for guards.

It is straightforward to translate the system to an upper-semicontinuous hybrid system in the form of Definition 5.1. We take $X = \bigcup_{q \in Q} \{q\} \times X_q$ as the state space. The invariant domain is constructed from both the explicit invariants D_q and the guards $g_{q,e}$ by $D = \bigcup_{q \in Q} D_q \cap \bigcap_{e \in E_U} \{x \mid g_{q,e}(x) \leq 0\}$. We construct the flow Φ by integrating the differential inclusions F_q , and $\Phi|_D$ by restricting to the invariant. For the discrete events, we immediately restrict to the activation regions by taking take $R|_A(q, x) = \bigcup_{e \in E_P} \{\gamma(q, e)\} \times R_{q,e}|_{A_{q,e}}(x)$ where $A_{q,e} := \{x \mid g(x) \geq 0\}$ for urgent events e .

Computing lower approximations to the evolution is more challenging, since we have to treat urgent and permissive transitions differently. For each permissive event, we take $R_e|_A(q, x) = R_{q,e}|_{A_{q,e}}(x)$ as the restricted reset map. For each urgent event e and each discrete state q , we can compute the crossing time set $\Gamma_{q,e} : X_q \mapsto X_q \times \mathbb{R}^+$ for the restricted flow $\Phi_q|_{D_q}$. We can then compute each urgent transition separately as in the proof of Theorem 5.22. By combining the discrete transitions which can occur for each individual event, we can show that the evolution $\Psi : X \times \mathbb{R}^+ \rightrightarrows X$ is $\mu_{<}$ -computable from the data describing the system.

6.2 Reliable simulation of hybrid systems

We wish to be able to reliably simulate the trajectory of a deterministic hybrid automaton starting at some initial point x . Away from discontinuity points in the spacial dependence of the evolution, the meaning of a simulation is clear; there is a unique trajectory, which we can compute using upper semantics. However, at the discontinuity points, there are at least two possible choices for how to continue the evolution; at a grazing or external corner collision point, we must choose between carrying on with the continuous dynamics, or applying a discrete reset. At a point where multiple events are activated, we must choose between which of the two or more events occurs. Even near the discontinuity points, we may not be able to reliably distinguish which of the possible continuations occurs due to numerical error.

One way of resolving these different possibilities is to make either a random choice, or rank the possible events in some order and apply the preferred event. However, this runs the risk of missing qualitatively different evolutions. Another option is to continue with all possible different evolutions. This is only feasible if the discontinuity set is entered at a discrete set of time instances.

If crossings with the guard set $G = \partial D \cap \partial A$ are δ -detectable, then we can compute the set of grazing points as $G_0 = \{x \in G \mid \Phi(x, [0, \delta/2]) \subset D\}$, which is $\psi_{<}$ -computable from Φ and D . Taking e_0 to be the special *grazing event* with guard G_0 , we can compute the discontinuity set as the union of all intersections of pairs of guard sets. Hence the discontinuity set is $\bigcup_{(e_i, e_j) \in E \cup \{e_0\}} G_i \cap G_j$, and is $\psi_{<}$ -computable from the system data.

We can think of simulation as computing the evolution Ψ of the system from a single initial point. In order to distinguish between the multiple possibilities at branching points, we need only store a list of event labels and times. Since we can compactify \mathbb{R}^+ by adding the point at infinity, the set $E \times (\mathbb{R}^+ \cup \{\infty\})$ is itself second-countable and locally compact in the product topology. By extending the state variable with the time t , and the reset relation by updating the list of events whenever a reset occurs, we obtain a new hybrid system in which the evolution operator stores the sequence of discrete events and the total time used to reach a particular state, from which the entire trajectory can be completely reconstructed.

Hence by Theorem 5.4, we can compute the set of all possible evolutions distinct evolutions from a given initial point for a uniformly non-Zeno hybrid system.

6.3 Implementation issues

Throughout the paper, every effort has been made to present the minimal assumptions necessary in order to perform a computation. In particular, no assumptions on the differentiability of various objects were made. Further, the counterexamples to computability were all based on simple affine systems, so adding differentiability assumptions makes no difference to the ability to compute arbitrarily accurate approximations to the evolution. However, efficient numerical methods require differentiability assumptions on the inputs in order to obtain high-order convergence. Therefore, when implementing the operations involved, particularly the algorithms for computing system evolution and crossing of guard sets, it is important to use differentiability to obtain efficient algorithms. As an example, the crossing time to a transverse guard set can be computed to an order which is the maximum differentiability of the guard constraint and the flow. This can allow more efficient stepping over guard constraints than methods relying purely on checking for crossing using set inclusions.

The theory presented in this paper has been implemented in the tool ARIADNE for reachability analysis of hybrid systems. Examples of computations performed using ARIADNE can be found in [6].

7 Concluding Remarks

In this paper, we have considered the computability of the evolution of a hybrid system, in which input and output data are specified by arbitrarily-accurate approximations to the exact values.

The main points are summarised below:

1. It is impossible in general to compute the evolution (simulation, reachable sets) of a hybrid system to arbitrary accuracy, and this holds even for simple classes of hybrid system, such as piecewise-constant derivative systems.
2. The obstruction to computability is due to discontinuities in the temporal evolution and in the spacial dependence on the initial conditions. Away from discontinuity points, the evolution is computable. Essentially the only hybrid systems for which

the evolution can be computed to arbitrary accuracy for any initial condition are those for which every trajectory starting in a given mode undergoes the same sequence of discrete events.

3. It is possible to regularise any hybrid system such that it is possible to compute convergent approximations to the evolution from above (“closure semantics”) or below (“interior semantics”), but the regularisations admit different solution sets. The regularisation of a deterministic system is necessarily either nondeterministic or admits blocking.
4. The regularisation using interior semantics cannot handle crossings of guard sets properly. Instead, we need to use a different regularisation “crossing semantics”. Under a regularity condition on the crossings of the guard sets, it is possible to compute convergent lower-approximations to the evolution; otherwise spurious solutions may be introduced.
5. The semicomputability results are valid for general classes of system, including nonsmooth, discontinuous and nondeterministic systems. Restricting to a special subclass of hybrid system does not change what is possible to compute, but may allow for more efficient algorithms.
6. The framework of “computable analysis” is a powerful machinery for discussing computational aspects of hybrid systems theory. It provides a clear notion of what we should be aiming to compute about a given mathematical object which can be interpreted in terms of convergent sequences of approximations. It gives natural topological conditions under which the computation can be proved to be impossible. It also provides a methodology of proving computability results using natural mathematical language without having to resort to the details of ϵ - δ style proofs.

There are many interesting areas for further research, especially in the analysis of lower-semicomputability. In particular, it would be useful to have generic verifiable conditions under which all crossings are detectable. For general systems, it would also be interesting to give an exact classification of the computability of the evolution with respect to the arithmetic hierarchy. It would also be interesting to extend this analysis to other problems, such as verification and control synthesis, and to other classes of systems. There is some evidence to suggest that stochastic systems with a diffusion term [13, 7], may have better computability properties than deterministic systems. It would also be interesting to compare decidability of system properties in the framework of computable analysis, in which *only* approximations to the input are considered, with computability in some algebraic framework in which exact computations are possible. In the light of previous work on piecewise-constant derivative systems [2] and (non) o-minimal affine systems [24, 10], it seems likely that while there are be specific problem instances which can be decided using algebraic methods, using exact descriptions does not fundamentally change the class of solvable problems. Finally, it is vital to develop more efficient numerical algorithms for the computation of upper- and lower-approximations to the system evolution.

References

- [1] Eugene Asarin, Theo Dang, and Oded Maler. d/dt: A verification tool for hybrid systems. In *Proceedings of the 40th IEEE Conference on Decision and Control*, New York, 2001. IEEE Press.

- [2] Eugene Asarin, Oded Maler, and Amir Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoret. Comput. Sci.*, 138(1):35–65, 1995.
- [3] Jean-Pierre Aubin and Arrigo Cellina. *Differential inclusions*, volume 264 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1984. Set-valued maps and viability theory.
- [4] Jean-Pierre Aubin, John Lygeros, Marc Quincampoix, and Shankar Sastry. Impulse differential inclusions: A viability approach to hybrid systems. *IEEE Trans. Automatic Control*, 47(1):2–20, 2002.
- [5] Bruce Krogh B. Izaias Silva, Keith Richeson and Alongkritt Chutinan. Modeling and verification of hybrid dynamical system using CheckMate. In *Proceedings of the 4th International Conference on Automation of Mixed Processes*, pages 189–194, 2000.
- [6] Andrea Balluchi, Alberto Casagrande, Pieter Collins, Alberto Ferrari, Tiziano Villa, and Alberto L. Sangiovanni-Vincentelli. Ariadne: a framework for reachability analysis of hybrid automata. In *Proceedings of the 17th International Symposium on the Mathematical Theory of Networks and Systems, Kyoto, Japan, July 24-28, 2006*, pages 1269–1267, 2006.
- [7] Julien Bect, Hana Baili, and Gilles Fleury. Generalized Fokker-Planck equation for piecewise-diffusion processes with boundary hitting resets. In *Proceedings of the 17th International Symposium on the Mathematical Theory of Networks and Systems, Kyoto, Japan, July 24-28, 2006*, 2006.
- [8] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and real computation*. Springer-Verlag, New York, 1998. With a foreword by Richard M. Karp.
- [9] Oleg Botchkarev and Stavros Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In N. Lynch and B. Krogh, editors, *Hybrid Systems: Computation and Control*, number 1790 in *Lecture Notes in Computer Science*, pages 73–88, Berlin Heidelberg New York, 2000. Springer-Verlag.
- [10] Thomas Brihaye and Christian Michaux. On the expressiveness and decidability of o-minimal hybrid systems. *J. Complex.*, 21(4):447–478, 2005.
- [11] Mireille Broucke and Ari Arapostathis. Continuous interpolation of solutions of Lipschitz inclusions. *J. Math. Anal. Appl.*, 258(2):565–572, 2001.
- [12] Mireille Broucke and Ari Arapostathis. Continuous selections of trajectories of hybrid systems. *Systems Control Lett.*, 47(2):149–157, 2002.
- [13] Manuela L. Bujorianu and John Lygeros. Theoretical foundations of stochastic hybrid systems. In *Proceedings of the Sixteenth International Symposium on Mathematical Theory of Networks and Systems*, 2004.
- [14] Pieter Collins. A trajectory-space approach to hybrid systems. In *Proceedings of the International Symposium on the Mathematical Theory of Networks and Systems, Katholiek Univ. Leuven, Belgium., August 2004*, 2004.
- [15] Pieter Collins. Continuity and computability of reachable sets. *Theoret. Comput. Sci.*, 341(1-3):162–195, 2005.
- [16] Pieter Collins and John Lygeros. Computability of finite-time reachable sets for hybrid systems. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4688–4693, 2005.
- [17] Michael Dellnitz, Gary Froyland, and Oliver Junge. The algorithms behind GAIO-set oriented numerical methods for dynamical systems. In Bernold Fiedler, editor, *Ergodic theory, analysis, and efficient simulation of dynamical systems*, pages 145–174, 805–807. Springer, Berlin, 2001.
- [18] Goran Frehse. Phaver: Algorithmic verification of hybrid systems past hytech. In Manfred Morari and Lothar Thiele, editors, *Hybrid Systems: Computation and Control*, volume 3414 of *Lecture Notes in Computer Science*, pages 258–273. Springer, 2005.
- [19] R. Goebel and A. R. Teel. Solutions to hybrid inclusions via set and graphical convergence with stability theory applications. *Automatica J. IFAC*, 42(4):573–587, 2006.
- [20] Rafal Goebel, Joao Hespanha, Andrew R. Teel, Chaohong Cai, and Ricardo Sanfelice. Hybrid systems: Generalized solutions and robust stability. In *Proceedings of the Symposium on Nonlinear Control Systems*. Elsevier, 2004.
- [21] Lars Grüne and Oliver Junge. Optimal stabilization of hybrid systems using a set oriented approach. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, pages 2089–2093, 2006.

- [22] Thomas A. Henzinger, Benjamin Horowitz, Rupak Majumdar, and Howard Wong-Toi. Beyond hytech: Hybrid systems analysis using interval numerical methods. In N. Lynch and B. Krogh, editors, *Hybrid Systems: Computation and Control*, number 1790 in Lecture Notes in Computer Science, pages 130–144, Berlin Heidelberg New York, 2000. Springer-Verlag.
- [23] Ker-I Ko. *Complexity theory of real functions*. Progress in Theoretical Computer Science. Birkhäuser Boston Inc., Boston, MA, 1991.
- [24] Gerardo Lafferiere, George J. Pappas, and Shankar Sastry. O-minimal hybrid systems. *Math. Control Signals Systems*, 13:1–21, 2000.
- [25] John Lygeros, Karl Henrik Johansson, Shankar Sastry, and Magnus Egerstedt. On the existence of executions of hybrid automata. In *Proceedings of the 38th IEEE Conference on Decision and Control*, pages 2249–2254, New York, 1999. IEEE Press.
- [26] A. Puri and P. Varaija. Decidable hybrid systems. *Math. Comput. Modelling*, 23(11/12):191–202, 1996.
- [27] Anuj Puri, Pravin Varaiya, and Vivek Borkar. Epsilon-approximation of differential inclusions. In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems III*, volume 1066 of *LNCS*, pages 362–376, Berlin, 1996. Springer.
- [28] Klaus Weihrauch. *Computable analysis*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2000. An introduction.