Robust computations with dynamical systems

Olivier Bournez, Daniel Graça, Emmanuel Hainry

Robust computations with dynamical systems

Olivier Bournez[1], Daniel S. Graça[2,3], and Emmanuel Hainry[4,5]

[1] Ecole Polytechnique, LIX, 91128 Palaiseau Cedex, France
Olivier.Bournez@lix.polytechnique.fr
[2] DM/FCT, Universidade do Algarve, C. Gambelas, 8005-139 Faro, Portugal
dgraca@ualg.pt
[3] SQIG /Instituto de Telecomunicações, Lisbon, Portugal
[4] LORIA, BP 239 - 54506 Vandœuvre-lès-Nancy Cedex, France
Emmanuel.Hainry@loria.fr
[5] Nancy Université, Université Henri Poincaré, Nancy, France

Abstract. In this paper we discuss the computational power of Lipschitz dynamical systems which are robust to infinitesimal perturbations. Whereas the study in [1] was done only for not-so-natural systems from a classical mathematical point of view (discontinuous differential equation systems, discontinuous piecewise affine maps, or perturbed Turing machines), we prove that the results presented there can be generalized to Lipschitz and computable dynamical systems.
In other words, we prove that the perturbed reachability problem (i.e. the reachability problem for systems which are subjected to infinitesimal perturbations) is co-recursively enumerable for this kind of systems. Using this result we show that if robustness to infinitesimal perturbations is also required, the reachability problem becomes decidable. This result can be interpreted in the following manner: undecidability of verification doesn't hold for Lipschitz, computable and robust systems.
We also show that the perturbed reachability problem is co-r.e. complete even for C^∞-systems.

Key words : Verification, Model-checking, Computable Analysis, Analog Computations.

1 Introduction

The investigations on the relationships between dynamics and computations attracted the attention of several research communities. One of them is highly motivated by the question of computer aided verification, and in particular by the question of computer aided verification of hybrid systems (see e.g. [2]).

One main motivation of this community is to get some "as automatic as possible" computer systems, that would take as input the description of a continuous or discrete (or hybrid) system, and the description of some property, call it "safety", and that would tell whether the system satisfies it or not.

The point is, by undecidability of the halting problem of Turing machines, there is no hope to get a fully decidable procedure if the provided formalism for

describing hybrid systems allows the description of Turing machines, and if the formalism for describing the property allows to talk about their halting.

Some classes of models, such as Timed Automata [3], have been shown to provide subclasses of systems for which verification of the reachability property is decidable.

However, unfortunately, very simple classes of linear hybrid automata [4] or piecewise constant derivative systems [5] have been shown of being able to simulate arbitrary Turing machines. As a consequence, verification procedures are semi-decision procedures and not decision procedures. A more general result can be found in [6], where semidecidability is shown for non-linear systems in general, and decidability is proved for systems satisfying some particular condition.

Since the proofs of undecidability or, more generally, of simulation of Turing machines, often involve to encode the configuration of a Turing machine (or of a two counter automata) into some real numbers, and since this require infinite precision, in the hybrid system verification community a folklore conjecture appeared saying that this undecidability is due to non-stability, non-robustness, sensitivity to initial values of the systems, and that it never occurs in "real systems" [1].

For example, Martin Fränzle writes in [7] "Hence, on simple information-theoretic grounds, the undecidability results thus obtained can be said to be artifacts of an overly idealized formalization. However, while this implies that the particular proof pattern sketched above lacks physical interpretation, it does not yield any insight as to whether the state reachability problem for hybrid systems featuring noise is decidable or not. We conjecture that there is a variety of realistic noise models for which the problem is indeed decidable".

There were several attempts to formalize and prove (or to disprove) this conjecture: it has been proved that small perturbations of the trajectory still yields undecidability [8]. Infinitesimal perturbations of the dynamics for a certain model of hybrid systems has shown to rise to decidability [7]. This has been extended to several models by [1]. In [9] it is shown that Turing machines exposed to small stochastic noise can decide the Halting problem, since its computational power when the error converges to 0 is $\approx \Pi_2^0$.

Let us look at the result presented in [1]: they consider several classes of widely used models of dynamical systems: Turing machines, piecewise affine maps, linear hybrid automata, and piecewise constant derivative systems. For each of them a notion of "perturbed" dynamics is introduced and the computational power of the corresponding perturbed systems is studied. Perturbations are defined for each model using a notion of metrics on the state space. For a given model, with reachability relation R, the idea is to perturb the dynamic by a small ε, and then take (as the perturbed dynamics of the system) the limit (intersection) R_ω of the perturbed reachability relations as this ε tends to 0. In that setting, a system is said "robust" if its reachability relation does not change under small perturbations of the dynamics, i.e. R_ω is equal to R [1]. This has a close resemblance with the notion of "structural stability" for dynamical systems: a system $\mathcal{A}$ is structurally stable if, roughly, ε-perturbed systems converge

to $\mathcal{A}$ as $\varepsilon \to 0$, a concept widely studied in the dynamical system theory see e.g. [10], [11].

In [1], the authors show that for Turing machines, piecewise affine maps, linear hybrid automata, and piecewise constant derivative systems, the relation R_ω belongs to the class Π_0^1 (it is co-recursively enumerable), and moreover, any Π_0^1 relation can be reduced to a relation R_ω of a perturbed system: any complement of a recursively enumerable set, can be semi-decided by an infinitesimally perturbed system.

This means that, for any robust system, its reachability problem is decidable. Indeed, as any system, its reachability problem is semi-decidable (recursively enumerable), and since it is robust, the complement of its reachability problem must be recursively enumerable, from which it follows that the reachability problem must be recursive for robust systems.

In other words, this gives a (partial) answer to the above mentioned conjecture: verification is decidable for robust systems, if the notion of robustness is the one considered here. If one prefers, undecidability of verification arises only when non-robust systems are considered.

In this paper we extend the result of [1] for the case of Lipschitz and computable (in the sense of recursive analysis [12]) systems defined on a compact set, considered as a model of computation. We present both continuous-time and discrete-time versions of our results.

Our aim is to reinforce in some sense the previous result: it follows that verification is decidable for robust systems considered in classical mathematics and computer science, that is to say for robust, Lipschitz, and computable dynamics. In other words, undecidability of verification is really a by-product of non-robustness, even if the system does not rely on trivial (piecewise) dynamics as in [1].

In a more provocative way, undecidability of verification for safety properties over a compact domain is indeed an artifact of modelization for very general and natural classes of systems.

2 Formal setting: continuous-time dynamical systems

We begin with some definitions.

Definition 1. *A function $f : \mathbb{R}^m \to \mathbb{R}^k$ is said Lipschitz over a set X if there is some $K > 0$ such that for all $\mathbf{x}, \mathbf{y} \in X$ one has*

$$\|f(\mathbf{x}) - f(\mathbf{y})\| \leq K \|\mathbf{x} - \mathbf{y}\| . \tag{1}$$

In particular it is well known that C^1 functions are Lipschitz over a compact set $X \subseteq \mathbb{R}^m$ and that an initial-value problem

$$\begin{cases} x' = f(t, x) \\ x(t_0) = x_0 \end{cases}$$

where f is Lipschitz, have an unique solution (see e.g. [13]).

Definition 2 (Dynamical systems). *Let $X \subset \mathbb{R}^d$, and consider some function $f : X \to X$. Then we can define a (homogeneous inputless) discrete or continuous-time dynamical system associated to (X, f) as follows:*

- *In the discrete-time case, a* trajectory *is a sequence of points $\{\mathbf{x}_0, \mathbf{x}_1, ...\} \in X^{\mathbb{N}}$, satisfying $f(\mathbf{x}_{i+1}) = \mathbf{x}_i$ for all $i \in \mathbb{N}$.*
- *In the continuous-time case, a* trajectory *is a solution of the differential equation $\dot{\mathbf{x}} = f(\mathbf{x})$, $\mathbf{x}(0) = \mathbf{x}_0 \in X$, i.e. a derivable function $\phi : \mathbb{R}_0^+ \to X$, satisfying $\phi(0) = \mathbf{x}_0$, and $\phi'(t) = f(\phi(t))$ for all t.*

Note that dynamical systems are deterministic: there is only one trajectory starting in a given initial point.

In this paper we consider dynamical systems as recognizers of languages: Σ denotes the alphabet $\Sigma = \{0, 1\}$ and Σ^* denotes words over this alphabet. Therefore we need to encode words over Σ as points in X. This is done using the encoding $\nu : \Sigma^* \to [0, 1]$ defined by: if $w = w_1 \dots w_n \in \{0, 1\}^*$, where $w_1, \dots, w_n \in \{0, 1\}$, then $\nu(w) = \sum_{i=1}^{n} \frac{(2w_i+1)}{4^i}$.

This classical encoding is rather arbitrary. Similar encodings would still yield the results proved in this paper.

We want to avoid not-so-interesting ways to get uncomputability:

- First, we restrict ourselves to dynamics over a compact domain. It is known that smooth systems can robustly simulate Turing machines [14], [15] (configurations are coded as integers), if the perturbations are $\leq \epsilon$, for some fixed $\epsilon > 0$. If we do some mapping from an unbounded domain to a bounded domain, these fixed ϵ-perturbations correspond to infinitesimal perturbations in the compact space. Further allowing infinitesimal errors in an unbounded space seems to originate a degree of modelization which is exceedingly artificial to be considered.
  There is no loss in generality in assuming the compact domain to be $[-1, 1]^d$.
- Second, we want to avoid undecidability due to the impossibility of distinguishing two reals in the recursive analysis setting: this is why we assume that when some computation is accepted this can be stated by considering a value that is clearly below some threshold (1/4), and that this quantity is clearly above a bigger threshold (1/2) when the computation is not terminated.

This leads to the following definition:

Definition 3 (Considering a dynamical system as a language recognizer). *Let $\mathcal{H}$ be a discrete/continuous-time dynamical system over space $X = [-1, 1]^d$. Let V_{accept} be the set of $\mathbf{x} \in X$ with $\|\mathbf{x}\| \leq 1/4$ and $V_{compute}$ be the set of $\mathbf{x} \in X$ with $\|\mathbf{x}\| \geq 1/2$. We say that $\mathcal{H}$ computes a language $L \subset \Sigma^*$ (or that L is the language of $\mathcal{H}$), over alphabet $\Sigma = \{0, 1\}$, if the following holds: for all $w \in \Sigma^*$, $w \in L$ iff the trajectory of $\mathcal{H}$ starting from $(\nu(w), 0, \cdots, 0, 1)$ reaches V_{accept}. For robustness reasons, we assume that for any $w \notin L$, the corresponding trajectory always stays in $V_{compute}$.*

3 Formal setting: recursive analysis

Recursive analysis or computable analysis, was introduced by Turing [16], Grzegorczyk [17], Lacombe [18]: see [12] for an up-to-date monograph presentation of recursive analysis using a computability point of view, or [19] for a presentation using a complexity theory point of view.

Following Ker-I Ko [19], let $\nu_{\mathbb{Q}} : \mathbb{N} \to \mathbb{Q}$ be the following representation[6] of dyadic rational numbers by integers: $\nu_{\mathbb{Q}}(\langle p, q, r\rangle) \mapsto \frac{p-q}{2^r}$, where $\langle ., ., .\rangle : \mathbb{N}^3 \to \mathbb{N}$ is an elementarily polynomial time computable bijection.

A sequence of integers $(x_i)_{i\in\mathbb{N}} \in \mathbb{N}^{\mathbb{N}}$ converges quickly toward x (denoted by $(x_i)_{i\in\mathbb{N}} \rightsquigarrow x$) if the following holds for all i: $|\nu_{\mathbb{Q}}(x_i) - x| < 2^{-i}$.

A point $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ is said computable if for all j, there is a computable sequence $(x_i)_{i\in\mathbb{N}} \in \mathbb{N}^{\mathbb{N}}$ (i.e. a computable function $a : \mathbb{N} \to \mathbb{N}$ such that $x_i = a(i)$ for all $i \in \mathbb{N}$) satisfying $(x_i)_{i\in\mathbb{N}} \rightsquigarrow x_j$.

A function $f : X \subset \mathbb{R}^d \to \mathbb{R}$, where X is compact, is said computable if there exists some d-oracle Turing machine M such that, for all $\mathbf{x} = (x_1, \ldots, x_d) \in X$, for all sequences $(x_i^j)_{i\in\mathbb{N}} \rightsquigarrow x_j$, M taking as oracles these d sequences computes a sequence $(x'_i)_{i\in\mathbb{N}}$ with $(x'_i)_{i\in\mathbb{N}} \rightsquigarrow f(\mathbf{x})$. A function $f : X \subset \mathbb{R}^d \to \mathbb{R}^d$, where X is compact, is said computable if all its projections are.

4 Formal setting: robustness

We now introduce the settings of [1], based on an idea of [20].

Definition 4 (ε-perturbation). *Consider a discrete/continuous-time dynamical system $\mathcal{H} = (X, f)$. Given $\varepsilon > 0$, its ε-perturbation $\mathcal{H}_\varepsilon$ is the discrete/continuous-time system $\mathcal{H}_\varepsilon$ defined over the same space X, where:*

1. *$(\mathbf{x}_0, \mathbf{x}_1, ...)$ is a trajectory of $\mathcal{H}_\varepsilon$ (the trajectory may not be unique), in the case where $\mathcal{H}$ is discrete-time, if $\|\mathbf{x}_{i+1} - f(\mathbf{x}_i)\| \leq \varepsilon$ for all $i \in \mathbb{N}$;*
2. *$\phi : \mathbb{R}_0^+ \to X$ is a trajectory of $\mathcal{H}_\varepsilon$ (the trajectory may not be unique), in the case where $\mathcal{H}$ is continuous-time, if $\phi(0) \in X$ and $\|\phi'(t) - f(\phi(t))\| \leq \varepsilon$ for all $t \in \mathbb{R}_0^+$.*

Note that the ε-perturbation $\mathcal{H}_\varepsilon$ of a dynamical system $\mathcal{H}$ is not, in general, a dynamical system since it is not deterministic (several trajectories may start with a given initial point).

Similarly to what is done in Definition 3, we can define a language computed by $\mathcal{H}_\varepsilon$, which we denote as L_ε: $w \in L_\varepsilon$ iff there is some trajectory of $\mathcal{H}_\varepsilon$ which starts from $(\nu(w), 0, \cdots, 0, 1)$ and reaches V_{accept}.

The following result is immediate.

Lemma 1. *For $0 < \varepsilon < \varepsilon'$, $L_\varepsilon \subset L_{\varepsilon'}$.*

[6] Many other natural representations of rational numbers can be chosen: they still yield the same class of computable functions – see [12, 19].

Following the idea given in [1], we consider $L_\omega = \cap_{\varepsilon>0} L_\varepsilon$.

Definition 5 (Robustness). *A dynamical system $\mathcal{H}$ is said robust iff its language L is equal to L_ω.*

5 Main results

The following results are extensions from those obtained in [1].

Theorem 1 (L_ω is co-r.e. for Lipschitz and computable Systems I). *Assume that language L is computed by a discrete-time system $\mathcal{H}$ defined over $[-1,1]^d$ which transition function is Lipschitz and computable. Then L_ω is co-recursively enumerable.*

Proof. Let $n \in \mathbb{N}\backslash\{0\}$ and decompose X in d-dimensional hypercubes $V_1, ..., V_s$ of size $\frac{1}{n}$. We build a finite automaton A_n, which states are $V_1, ..., V_s$, that roughly recognizes $L_{\frac{1}{n}}$. To complete the description of this automaton we need to define two things: (i) the set of accepting states and (ii) the transition rule δ_n. The set of accepting states consists of those hypercubes which overlap V_{accept}, i.e. hypercubes that have vertices within distance $\leq 1/4$ of the origin. These hypercubes can easily be identified.

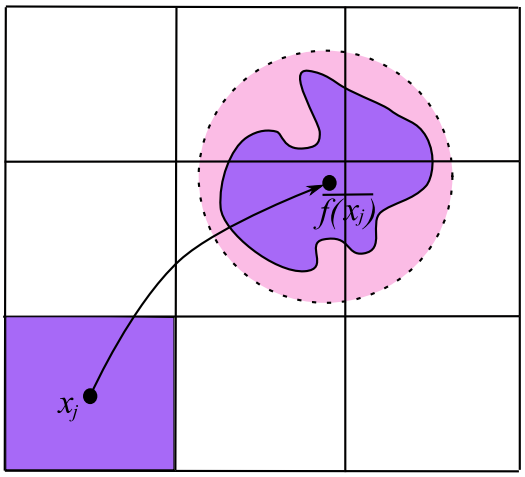

Fig. 1. A figure depicting various elements used in the demonstration of Theorem 1.

Now we have to present the transition rule of A_n. The following construction is depicted in Fig. 1. Let V_j be some hypercube. Then pick its central point $\mathbf{x}_j$ (this is an easily computable rational) and compute a rational approximation $\overline{f(\mathbf{x}_j)}$ of $f(\mathbf{x}_j)$ with precision $\frac{1}{n}$. Because f is Lipschitz, there will be some

Lipschitz constant $K > 0$ satisfying condition (1) for all $\mathbf{x}, \mathbf{y} \in X$. Then, if $\mathbf{x} \in V_j$ is another point of the same hypercube, we have

$$\|f(\mathbf{x}) - \mathbf{y}\| \leq \frac{1}{n} \quad \Rightarrow \tag{2}$$
$$\left\|\overline{f(\mathbf{x}_j)} - \mathbf{y}\right\| \leq \left\|\overline{f(\mathbf{x}_j)} - f(\mathbf{x}_j)\right\| + \|f(\mathbf{x}_j) - f(\mathbf{x})\| + \|f(\mathbf{x}) - \mathbf{y}\| \leq \frac{K+2}{n}.$$

By other words, if $\mathbf{y}$ is an ε-perturbed image of a point of V_j, then this point will be within distance $\frac{K+2}{n}$ of $\overline{f(\mathbf{x}_j)}$. We use this fact to proceed as follows. After computing $\overline{f(\mathbf{x}_j)}$, determine all the hypercubes which are within distance $\leq (K+2)/n$ of this point (in Fig. 1 this corresponds to all hypercubes covered by the ball of center $\overline{f(\mathbf{x}_j)}$ and radius $(K+2)/n$). This can be done algorithmically, in finite time, since it is only necessary to check which are the hypercubes (which are finitely many) that have vertices within distance $\leq (K+2)/n$ of $\overline{f(\mathbf{x}_j)}$. Let $W_1, ..., W_i$ be these hypercubes. Then we define the transition rule over the hypercubes as follows: $\delta(V_j) = \{W_1, ..., W_i\}$. This defines the automaton A_n.

Now we say that a point $\mathbf{x} \in X$ is accepted by A_n if it lies in an accepted hypercube. Let $L^*_{\frac{1}{n}}$ be the language accepted by A_n. From (2), the dynamics of A_n includes those of the $\frac{1}{n}$-perturbed system $\mathcal{H}_{\frac{1}{n}}$. Hence $L_{\frac{1}{n}} \subseteq L^*_{\frac{1}{n}}$. On the other side, it is not difficult to see that the dynamics of A_n are included in those of $\mathcal{H}_{\frac{K+3}{n}}$.[7] Therefore

$$L_{\frac{1}{n}} \subseteq L^*_{\frac{1}{n}} \subseteq L_{\frac{K+3}{n}} \quad \Rightarrow \quad \cap_{n=1}^{\infty} L^*_{\frac{1}{n}} = \cap_{\varepsilon>0} L_\varepsilon = L_\omega.$$

Let us now show that $L_\omega = \cap_{n=1}^{\infty} L^*_{\frac{1}{n}}$ is co-r.e., as required. Let $w \in \Sigma^*$. Then build a Turing machine which performs the following steps:

```
i=0
Repeat
    i++
    Simulate A_i with input ν_X(w)
Until ν_X(w) is rejected
Reject w
```

This shows that the complement of L_ω is r.e., as required.

We now introduce a (classical) tool to discretize a continuous-time system (see e.g. [21]).

Definition 6 (Stroboscopic map). *Consider the continuous-time system $\mathcal{H} = (X, f)$. Then its corresponding* stroboscopic map $g : X \to X$ *is given by $g(\mathbf{x}_0) = \phi(1)$, where $\phi : \mathbb{R}_0^+ \to \mathbb{R}$ satisfies $\phi'(t) = f(\phi(t))$, $\phi(0) = \mathbf{x}_0$.*

In other words, $g(\mathbf{x}_0)$ gives the point where the trajectory of $\mathcal{H}$, starting on $\mathbf{x}_0$, would reach after one time unit.

[7] Here we suppose that $\|\mathbf{x}\| = \|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$. However, a similar result holds for other norms since all norms are equivalent in a finite-dimensional space.

Lemma 2. *Let $\mathcal{H} = (X, f)$ be a continuous-time system, and let $\widehat{\mathcal{H}} = (X, g)$ be its discrete-time counterpart, where g is the stroboscopic map of $\mathcal{H}$. Then, if $L, \hat{L}$ are the languages computed by $\mathcal{H}$ and $\widehat{\mathcal{H}}$, respectively, under the same input encoding, we have:*

1. *$L = \hat{L}$;*
2. *$L_\omega = \hat{L}_\omega$ and $L_\varepsilon = \hat{L}_\varepsilon$, for all $\varepsilon > 0$.*

Proof. Note that for all $\mathbf{x}_0 \in X$, if $\phi : \mathbb{R}_0^+ \to \mathbb{R}$, $\phi(0) = \mathbf{x}_0$ is a trajectory of $\mathcal{H}$ and $(\mathbf{x}_0, \mathbf{x}_1, ...)$ is a trajectory of $\widehat{\mathcal{H}}$, we must have $\phi(k) = \mathbf{x}_k$ for all $k \in \mathbb{N}$. Therefore, because only the accepting trajectories reach V_{accept} and then stay there, we must have $L = \hat{L}$. Similar arguments can be used to show the identity $L_\varepsilon = \hat{L}_\varepsilon$ that, by its turn, implies $L_\omega = \hat{L}_\omega$.

Theorem 2 (L_ω is co-r.e. for Lipschitz and computable Systems II). *Assume that language L is computed by a continuous-time system $\mathcal{H}$ defined over $[-1, 1]^d$ with a Lipschitz and computable transition function. Then L_ω is co-recursively enumerable.*

Proof. Using the same assumptions of the previous lemma, to show that the language L computed by the continuous-time system $\mathcal{H} = (X, f)$, where $X = [-1, 1]^d$, is co-r.e., it is enough to show that $\hat{L}$ is a co-r.e. language. If we show that g is Lipschitz and computable, then this can be done as indicated in the proof of Theorem 1. It is well-known (see e.g. [13]) that if f satisfies equation (1) over X, and if ϕ_0, ϕ_1 are solutions of the differential equation $x' = f(x)$ over X, then

$$\|\phi_0(t) - \phi_1(t)\| \leq \|\phi_0(0) - \phi_1(0)\| \, e^{Kt}.$$

By other words, the stroboscopic map g is Lipschitz on X, with Lipschitz constant e^K. Moreover, g is computable by using an algorithmic version of Euler's method on the time interval $[0, 1]$ for the differential equation $\dot{\mathbf{x}} = f(\mathbf{x})$.(cf. [19]).

The following lemma is a corollary from the results of [6], [22]: one can semi-compute any trajectory of $\mathcal{H}$. Therefore one can semi-decide if a trajectory finishes in V_{accept}, and thus semi-decide L.

Lemma 3. *Let L be the language computed by a system $\mathcal{H} = (X, f)$, where $X = [-1, 1]^d$ and f is Lipschitz and computable. Then L is r.e.*

Since a system is robust iff $L = L_\omega$, the following corollary is immediate.

Corollary 1 (Robust =>Recursive for Lipschitz and Computable Systems). *In the conditions of the lemma above, if $\mathcal{H}$ is robust, then L is recursive.*

We now prove that Turing machines can be simulated robustly with C^∞-systems. Recall that C^∞-systems are also Lipschitz. This is a generalization of a similar result from [1], which holds for piecewise systems (which are not of class C^1).

Theorem 3 (Perturbed reachability is complete in Π_1^0). *Let A be a recursively enumerable language. Then there is a computable and C^∞ dynamical system $\mathcal{H}$ such that $L_\omega = \bar{A}$.*

Proof. For any Turing machine M, Theorems 4.4 and 4.5 from [23] give an explicit way to build a continuous-time dynamical system (X, f) where X is compact and f_M is $\mathcal{C}^\infty$ such that $y' = f_M(y)$ simulates M. It is easy to observe that the function f_M built there is computable. Changing coordinates, and reasoning through homothety, we can assume without loss of generality X to be $[-T, T]^3$ and that the accepting configuration of M correspond to the origin $c = (0, 0, 0) \in [-T, T]^3$, and that the initial starting point corresponding to some input w is of the form $c = (\nu(w), 0, 1)$ as in Definition 3. This yields a dynamical system such that a word w is accepted by Turing machine M iff the trajectory starting from $(\nu(w), 0, 1)$ reaches the origin c. Let C be the cubic neighborhood of c defined by $[-0.1, 0.1]^3$, and C' be the cubic neighborhood of c defined by $[-0.2, 0.2]^3$. The resulting system is also such that a non-accepted word corresponds to a trajectory that stays forever outside of neighborhood C'.

We construct from this 3-dimensional smooth dynamical system f_M a smooth 4-dimensional dynamical system g_M whose perturbed version semi-recognizes the complement of the language recognized by M.

Actually, robustness will only be shown for one component, the extra component added to the original 3-dimensional system, which is the most critical for the simulation, . The other 3 components can be made robust using the construction presented in [14], [15] which, although originally done for unbounded spaces, can be brought to the compact space $[-T, T]^3$ via a transformation using a function like, e.g. arctan, and can still be used here. However, giving all the details would span the contents of this paper outside allowable size, so we only show robustness for the crucial component.

We will use the notation $\mathbf{x}, \mathbf{y}$ for 3-dimensional vectors, and h for the fourth dimension. This 4-dimensional system is mainly the original smooth dynamical system embedded in the hyperplane $h = 3$ of $\mathbb{R}^4$, however with two changes. First, the neighborhood C of the original dynamical system becomes rejecting in the new system. Second, the zone $h \leq 1$ becomes accepting in the new system.

The idea is that for any word w accepted by M, the original system f_M will eventually come to C, and hence the perturbed dynamical system g_M will eventually go up and reject. For any word w not accepted by M, the system g_M will slowly drift "down" until it reaches the accepting zone $h \leq 1$.

In order to get a smooth dynamics, we start by the following technicalities. This is an easy and classical mathematical exercise to prove that for any computable reals $a < b$, function $h_{a,b}(x) = \exp(-1/(x-a)^2 - 1/(b-x)^2)$ for $a < x < b$ and 0 for $x \geq b$ or $x \leq a$ is $\mathcal{C}^\infty$ and computable. If we denote by γ the constant $\gamma = \int_a^b h_{a,b}(x)dx$, and by $\chi_{a,b}(x)$ the function $1/\gamma \int_a^x h_{a,b}(x)$, one gets a computable and $\mathcal{C}^\infty$ function that values 0 for $x \leq a$, and 1 for $x \geq b$, and a value in interval $[0, 1]$ in between. Given $a < b < c$, the function $\chi_{a,b,c}(x) = \chi_{a,b}(x) - \chi_{b,c}(x)$ values 0 for $x \leq a$, 1 in b, and 0 for $x \geq c$, with a

value in interval $[0,1]$ in between. In the same spirit, let $\chi_C : \mathbb{R}^3 \to \mathbb{R}$ be some function that values 1 over neighborhood C, and 0 outside neighborhood C'.

Formally the new system g_M is then defined on $[-T,T]^3 \times [-1,5] \subset \mathbb{R}^4$ as follows: $g_M(\mathbf{x},h) = (\mathbf{y},h')$ where

$$h' = \chi_{4,5}(h) + \chi_C(\mathbf{x})\chi_{2,3,4}(h) - \chi_{0,1,2}(h) + \chi_{-1,-0.5,0}(h)$$

and

$$\mathbf{y} = f_M(\mathbf{x})(1-\chi_C(x))\chi_{2,3,4}(h) + C(\mathbf{x})\chi_{0,0.5,1}(h),$$

where $\dot{x} = C(\mathbf{x})$ is a smooth and computable vector field with all the trajectories going to the origin.

In other words,

- if $h \geq 4$, anything that arrives in the layer $h \geq 4$ goes "up" and is rejected.
- if $2 < h < 4$:
  - if $\mathbf{x} \in C$, then it goes "up" and it is ultimately rejected.
  - if $\mathbf{x} \notin C$, then it basically simulates the original system f_M.
- if $1 < h < 2$, then it goes "down" until it reaches $h \leq 1$,
- if $-1 \leq h < 1$, then it goes to the origin.

Consider the trajectory starting from $(\nu(w), 0, 1, 3)$: suppose that word w is accepted by M: the corresponding trajectory of f_M will eventually go to the origin in some finite time t, and then the trajectory of g_M will reach neighborhood C, and hence will go up for ever. Taking ϵ small enough (depending on t) any ϵ-perturbed trajectory of g_M will be close at time t to this trajectory, and hence also in C, and hence going for ever up afterwards.

Suppose that word w is not accepted by M: the corresponding trajectory of f_M will run for ever outside of C'. For any $\epsilon > 0$, we can easily construct a trajectory of g_M that drifts down slowly in the fourth coordinate until reaching $h \leq 1$, and then going to the origin.

Through a change of coordinates, the obtained system fulfills all constraints of Definition 3 and of the statement of the Theorem.

In other words, "perturbed reachability is complete in Π_1^0" as this is termed in [1].

6 Conclusion

In this paper we showed that, on compact sets, the perturbed reachability problem is co-r.e. for Lipschitz and computable systems. We also proved that for Lipschitz and computable systems which are robust to infinitesimal perturbations, the reachability problem is decidable. The results are both for the discrete and continuous-time case. It would be interesting to know what happens at a more refined level, i.e. if from a complexity point of view.

Acknowledgments. D. Graça was partially supported by *Fundação para a Ciência e a Tecnologia* and EU FEDER POCTI/POCI via SQIG - Instituto de Telecomunicações.

References

1. Asarin, E., Bouajjani, A.: Perturbed Turing machines and hybrid systems. In: Logic in Computer Science, 2001. Proc. 16th Annual IEEE Symposium. (2001) 269–278
2. Alur, R., Pappas, G.J., eds.: Hybrid Systems: Computation and Control: 7th International Workshop (HSCC 2004). LNCS 2993. Springer (2004)
3. Alur, R., Dill, D.L.: Automata for modeling real-time systems. In: Automata, Languages and Programming, 17th International Colloquium. LNCS 443, Springer (1990) 322–335
4. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What's decidable about hybrid automata? J. Comput. System Sci. **57**(1) (1998) 94–124
5. Asarin, E., Maler, O., Pnueli, A.: Reachability analysis of dynamical systems having piecewise-constant derivatives. Theoret. Comput. Sci. **138** (1995) 35–65
6. Collins, P.: Continuity and computability of reachable sets. Theor. Comput. Sci. **341** (2005) 162–195
7. Fränzle, M.: Analysis of hybrid systems: An ounce of realism can save an infinity of states. In Flum, J., Rodríguez-Artalejo, M., eds.: Computer Science Logic (CSL'99). LNCS 1683, Springer (1999) 126–140
8. A.Henzinger, T., Raskin, J.F.: Robust undecidability of timed and hybrid systems. In Lynch, N.A., Krogh, B.H., eds.: Proc. Hybrid Systems: Computation and Control, Third International Workshop, HSCC 2000. LNCS 1790, Springer (2000) 145–159
9. Asarin, E., Collins, P.: Noisy Turing machines. In Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M., eds.: ICALP. LNCS 3580, Springer (2005)
10. Guckenheimer, J., Holmes, P.: Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Fields. Springer (1983)
11. Hirsch, M.W., Smale, S., Devaney, R.: Differential Equations, Dynamical Systems, and an Introduction to Chaos. Academic Press (2004)
12. Weihrauch, K.: Computable Analysis: an Introduction. Springer (2000)
13. Birkhoff, G., Rota, G.C.: Ordinary Differential Equations. 4th edn. John Wiley & Sons (1989)
14. Graça, D.S., Campagnolo, M.L., Buescu, J.: Robust simulations of Turing machines with analytic maps and flows. In Cooper, S.B., Löwe, B., Torenvliet, L., eds.: CiE 2005: New Computational Paradigms. LNCS 3526, Springer (2005) 169–179
15. Graça, D.S., Campagnolo, M.L., Buescu, J.: Computability with polynomial differential equations. Adv. Appl. Math. **40**(3) (2008) 330–349
16. Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proc. London Math. Soc. (Ser. 2–42) (1936) 230–265
17. Grzegorczyk, A.: Computable functionals. Fund. Math. **42** (1955) 168–202
18. Lacombe, D.: Extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables réelles III. C. R. Acad. Sci. Paris **241** (1955) 151–153
19. Ko, K.I.: Computational Complexity of Real Functions. Birkhäuser (1991)
20. Puri, A.: Dynamical properties of timed automata. In Ravn, A.P., Rischel, H., eds.: Proc. Formal Techniques in Real-Time and Fault-Tolerant Systems, 5th International Symposium, FTRTFT'98. LNCS 1486, Springer (1998) 210–227
21. Ott, E.: Chaos in Dynamical Systems. 2nd edn. Cambridge University Press (2002)
22. Collins, P., Graça, D.S.: Effective computability of solutions of differential inclusions — the ten thousand monkeys approach. Journal of Universal Computer Science **15**(6) (2009) 1162–1185

23. Bournez, O., Cosnard, M.: On the computational power of dynamical systems and hybrid systems. Theoret. Comput. Sci. **168**(2) (1996) 417–459