Controllability and Falsification of Hybrid Systems

Pieter Collins†

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

Abstract— **In this paper we consider the controllability problem for hybrid systems, namely that of determining the set of states which can be driven into a given target set. We show that given a suitable definition of controllability, we can effectively compute arbitrarily accurate under-approximations to the controllable set using Turing machines. However, due to grazing or sliding along guard sets, we see that it may be able to demonstrate that an initial state can be controlled to the target set, without knowing any trajectory which solves the problem.**

Index Terms— **Hybrid system; controllable set; computable analysis; safety.**

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

I. Introduction

In this paper we consider the problem of computing the controllable set of a general nonlinear hybrid system. We restrict to hybrid systems without noise, but some nondeterminism still unavoidably enters the analysis due to difficulties in computing whether and when a discrete transition should take place.

This controllability problem is dual to the safety problem for noisy closed-loop systems, but harder since we have to simultaneously deal with choice and nondeterminism, whereas for safety we deal only with with nondeterminism. We see that a forwards approach to controllability is complicated by the need to consider sets of possible jump times, leading to multiple possibilities for further evolution which must be considered separately. Instead, a backwards approach yields a simple high-level algorithm which can still be implemented. The problem of computing the controllable set is equivalent to computing the *unsafe* set for a closed-loop system with nondeterministic noise. For we can consider the nondeterministic noise as the input of the "environment", and showing that a state is unsafe is equivalent to showing that the environment can guide the state into the unsafe set.

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

Over-approximations to reachable sets of hybrid systems for safety verification can be computed using various tools, including [1], [2], [3], [4], [5], [6] using a forwards analysis. For the controllability problem, when using a forwards analysis, we need to show that the target can be reached from all points in the initial state set. For these reasons, backwards analysis methods based on dynamical programming are usually preferable. A comparison of forwards and backwards for reachability methods including a discussing of numerical issues is given in [7].

Since the behaviour nonlinear hybrid systems can be extremely complicated, it is unclear whether it is even possible to compute the controllable set in all cases. Further, since we deal with objects in continous spaces, it is not even clean how we should describe such objects, and what the meaning of a valid computation should be. To solve these difficulties, we therefore use a framework of *computable analysis*, which provides a formal theory of computation on objects in continuous spaces, including concrete machine *representations* of fundamental types, a basic collection of *effectively computable operators* on these types, and conditions under which a function or operator is *uncomputable*.

The theory is based on topology and analysis, and the concrete representations of objects in a space correspond to topologies in the space. The fundamental result is that only continuous operators can be computable. If an operator is uncomputable, then we can see if it becomes computable using different representations, which correspond to strengthening the input data or weakening the requirements on the output.

There are many approaches to computable analysis, including *domain theory* [8], *locale theory* [9] and *type-two effectivity* [10]. However, all yield equivalent representations and the same computable operators. In

this paper, we shall therefore only list the classes of objects which we need to work with, and the computable operations, and mostly omit details of how objects are represented and which operations are computable.

It should be noted that computable analysis is a framework for defining *approximate* numerical computations with *nonzero* errors but *known* error bounds. This means that some problems which are solvable in an exact algebraic framework become unsolvable in computable analysis. However, typically only specific instances become solvable in the algebraic setting, and general problem remains unsolvable. In the computable analysis setting, solvability is intimately related to robustness, and we have the advantage of a richer class of computable operators to work with. Computable analysis has the advantage over traditional numerical analysis in that we can write algorithms using high-level operations which have been shown to be computable, rather than try to work explicitly with error bounds and epsilon-delta style proofs.

In this paper, as well as the familiar collection of open subsets, we also make use of the hyperspace of *overt* subsets. An overt set is a closed set represented by listing the open boxes it intersects, or equivalently, by a dense sequence of points. Overt-valued maps are ideal for describing control systems, since we have precisely enough information to determine whether an initial point can be controlled into an open target set.

The main contribution of this paper is that we give a notion of robustly controllable set which can be effectively computed, and an algorithm for performing the computation. We also show that in certain cases, the controllable set cannot be computed, but that our algorithm gives an under-approximation which is in some sense optimal. A weaker notion of robust controllability was given in [11].

The paper is organised as follows. In Section II we give a minimal definition of a hybrid system and describe its solutions. In Section III, we sketch the results from computable analysis which we need. In Section IV, we show that the controllable set varies discontinuously in system parameters as a result of discontinuities in the system evolution. In Section V, we show that the controllable set can be computed given the correct definitions. We give some conclusions and suggestions for further research in Section VI.

II. Hybrid Systems

A hybrid system is a system in which continuous evolution is interspersed with discrete transitions at which the state jumps discontinuously. There are many hybrid system models in the literature; in this paper we choose a framework which is general enough to exhibit the difficulties which may occur and how these can be adressed, without introducing additional complications.

Definition 2.1: A hybrid system is a tuple $H = (X, E, \Phi, \{D_e, A_e, R_e \mid e \in E\})$ where X is the state space, E is a set of events, $\Phi : X \rightrightarrows C(\mathbb{R}^+, X)$ is a multivalued flow giving the system *dynamic*, and for each event $e \in E$, A_e is the set in which e is *active*, D_e is the set in which e can be *delayed* and and $R_e : X \rightrightarrows X$ is the *reset* map.
Intuitively, the system state x may evolve according to the continuous dynamics Φ as long as for every event $e \in E$, we have $x \in D_e$. The system may make jump according to the event e if $x \in A_e$. We allow for the case that at a given state x, both continuous and discrete dynamics are possible.

The arrows $\rightrightarrows$ indicate that the functions Φ and R_e are multivalued, which takes into account the possibility of being able to choose different continuous dynamics or resets from a given point. We assume that Φ and R_e are everywhere-defined.

Notice that in the definition there is no explicit mention of discrete states. However, we can think of the state space X as being the disjoint union of spaces X_q for $q \in Q$, where Q is a finite set of discrete states. Also, rather than a global *invariant*, we instead give, for each event e, a set D_e such that event e will occur if the system is about to leave the set D_e.

To formally represent a trajectory of a hybrid system, we need to take into account the possibility that more than one discrete transition occurs at a given time. To capture the intermediate states, we use the following definition of *hybrid time domain* [12], [13], which is based on work of [14]:

Definition 2.2 (Hybrid trajectory): Let $(t_n)_{n<\infty}$ be an increasing sequence in $\mathbb{R}^+ \cup \{\infty\}$ with $t_0 = 0$. Then the t_n define a *hybrid time domain* $\mathcal{T} \subset \mathbb{R}^+ \times \mathbb{Z}^+$ by

$$\mathcal{T} = \{(t, n) \in \mathbb{R}^+ \times \mathbb{Z}^+ \mid t_n \leqslant t \leqslant t_{n+1}\}$$

A *hybrid trajectory* is a continuous function $\xi : \mathcal{T} \to X$ for some hybrid time domain $\mathcal{T}$. The trajectory ξ is *Zeno* if $\lim_{n\to\infty} t_n < \infty$, and *infinite* otherwise.
We define the solutions of a hybrid system as hybrid trajectories.

Definition 2.3 (Solution of a hybrid system): A hybrid trajectory is a *solution* or *execution* of the hybrid system $H = (X, E, \Phi, \{D_e, A_e, R_e\})$ if there is a sequence of events e_n such that

1) $\xi(t, n) \in \bigcap_{e\in E} D_e$ whenever $t_n \leqslant t < t_{n+1}$,
2) $\xi(\cdot, n) \in \Phi(\xi(t_n, n))$,
3) $\xi(t_n, n-1) \in A_{e_n}$, and
4) $\xi(t_n, n) \in R_{e_n}(\xi(t_n, n-1))$.

Condition 1 means that continuous evolution is only possible at x if $x \in D_e$ for all $e \in E$, though the state may leave $\bigcap_{e\in E} D_e$ at the time of an event. Condition 3 means that event e can only occur if $x \in A_e$.

In order to ensure that the system is non-blocking, we make the following key assumption:

Assumption 2.4 (Non-blocking): For every event e, we have $D_e^\circ \cup A_e = X$.

This assumption is natural in the sense that an event cannot block continuous evolution if it is not active. We need to take the interior of D_e since the formal semantics prohibits event e at a point $x \notin A_e$ even if $x \in \partial A_e$. We note that if $D_e^\circ \cup A_e^\circ = X$, then the event time is nondeterministic, whereas if $D_e^\circ \cap A_e^\circ = \emptyset$, then the event e occurs as soon as the state touches ∂A_e.

III. Computable analysis

We now outline how to describe objects such as points, sets and functions in the framework of computable analysis. In this article we use a higher-level form loosely based on *type theory* or *lambda calculus* rather than the low-level foundational approaches. Much of the material in this section can be found in [10], [15]. We say that a *representation* of a type is a way of describing objects of that type by infinite binary streams, and a *name* of an object is a stream describing it. In a topological space, we consider representations which are *admissible* with respect to the topology, which means that the names can be interpreted as giving increasingly accurate approximations to the object. An operation $f : X \to Y$ between types is *computable* if it is possible to write a program on a digital computer (as modelled by a Turing machine) transforming any name of an object x in X to a name of $f(x)$ in Y.

We consider a state space X, which can be taken as any locally-compact second countable Hausdorff spacee. In Euclidean space, we can describe a point x as a monotone intersection of open rational boxes.

We will be interested in the hyperspaces of *open* and *overt* subsets of X, denoted $\mathcal{O}$ and $\mathcal{V}$ respectively. We can describe an open set U as a countable union of compact rational boxes, and an overt set A by listing all open rational boxes intersecting A. If A is overt and U is open, we can prove that A intersects U by finding a rational box I such that $A \cap I \neq \emptyset$ and $\overline{I} \subset U$. If A is overt, then given $\epsilon > 0$ we can compute *ϵ-lower approximations* to A, by which we mean concrete sets B (described as unions of boxes) such that $B \subset N_\epsilon(A)$. The space of closed subsets of X with the lower Fell topology is equivalent to our overt sets; for a more detailed description of overtness, see [16], [17].

Theorem 3.1: We have the following computable operations on sets:

- Finite intersection $\mathcal{O} \times \mathcal{O} \to \mathcal{O}$.
- Countable union $\mathcal{O}^{\mathbb{N}} \to \mathcal{O}$.

and the following computable predicates:

- intersection $A \cap U \neq \emptyset$ as a function $\mathcal{V} \times \mathcal{O} \to S$, where S is the Sierpinski space $\{\top, \uparrow\}$.

We now consder the space of continuous functions $X \to Y$. If X is a locally-compact Hausdorff space, we use the *compact-open* topology on $C(X, Y)$, which is generated by the sets

$$\beta(K, U) = \{f \in C(X, Y) \mid f(K) \subset U\} \qquad (1)$$

for compact K and open U. If $f : X \to Y$ is continous and $x \in X$, then we can effectively evaluate $f(x)$ from names of f and x in the corresponding admissible representations.

We also wish to consder multivalued functions $X \rightrightarrows Y$, in particular, functions $X \to \mathcal{V}(Y)$.

Theorem 3.2: Let $F : X \to \mathcal{V}(Y)$ an overt function. Then if $A \subset X$ is overt and $V \subset Y$ is open, we can compute $F(A) \in \mathcal{V}(Y)$ and $F^{-1}(V) \in \mathcal{O}(X)$ from names of F, A and V.

In other words, if we can compute the image of a point as an overt set, then we can compute the image of any overt set, or the preimage of any open set. The condition on the preimages says that the effectively lower-semicontinuous closed-valued functions are precisely the overt functions.

We now consider continuous-time evolution. Recall that a *flow* is a function $\phi : X \times \mathbb{R} \to X$, such that (i) $\phi(x, 0) = x$ for all $x \in X$ and (ii) $\phi(x, s + t) = \phi(\phi(x, s), t)$ for all $x \in X$ and $s, t \in \mathbb{R}$. A flow satisfies the differential equation $\dot{x} = f(x)$ if $\dot{\phi}(x, t) = f(\phi(x, t))$ for all x, t. Equivalently, we can also think of a flow as a function $\hat{\phi} : X \to C(\mathbb{R}, X)$ such that $\xi = \hat{\phi}(x)$ if $\xi(0) = x$ and $\xi(t) = \phi(x, t)$.

Analogously, a multivalued flow is a function $\Phi : X \rightrightarrows C(\mathbb{R}, X)$ which satisfies the multiflow conditions (i) $\xi(0) = x$ for all $\xi \in \Phi(x)$, (ii) if $\xi \in \Phi(x)$, then the function η defined by $\eta(t) = \xi(t+s)$ is in $\Phi(\xi(s))$, and (iii) if $\xi \in \Phi(x)$ and $\eta \in \Phi(\xi(s))$, then the function ζ given by $\zeta(t) = \xi(t)$ for $t \leqslant s$ and $\zeta(t) = \eta(t - s)$ for $t \geqslant s$ is in $\Phi(x)$.

A multivalued flow is *overt* if it is continuous as a function $X \to \mathcal{V}(C(\mathbb{R}, X))$. From Theorem 3.2, we immediately deduce:

Corollary 3.3: If Φ is an overt flow, K a compact interval and U is open, then

$$\Phi^{-1}\big(\{\eta \mid \eta(K) \subset U\}\big) = \{x \in X \mid \exists \xi \in \Phi(x) \text{ s.t. } \xi(K) \subset U\}$$

is an open subset of X, and can be computed from names of Φ, K and U.

We can generate multivalued flows by differential inclusions $\dot{x} \in F(x)$. 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 the solution of a locally Lipschitz continuous differential inclusion was shown to be effectively computable (using different terminology) in [18]. We can refine this result and consider only lower-semicomputability.

Theorem 3.4 (Differential inclusions): Denote by $\Phi : X \rightrightarrows C(\mathbb{R}, X)$ the flow of the differential inclusion $\dot{x} \in F(x)$. If F is overt locally Lipschitz with convex values, then the solution operator flow Φ is overt, and can be effectively computed from a name of F.

IV. Nondeterministic Behaviour at Discontinuities

In this section we consider discontinuities caused by the discrete events in the evolution of a hybrid system.

Definition 4.1 (Controllability): A hybrid system H is *controllable* from x_0 to $T \subset X$ if there exists a solution ξ of H, $t \in \mathbb{R}^+$ and $n \in \mathbb{Z}^+$ such that $\xi(0,0) = x_0$ and $\xi(t,n) \in T$.

A system H is *robustly controllable* from x_0 to T if for any sufficiently small perturbations x_0' of x_0, T' of T and H' of H, the system H' is controllable from x_0' to T'.

Example 4.2: Consider a hybrid system on $\mathbb{R}$ with flow $\dot{x} = -1$, and a single event with $D = (0, \infty]$, $A = [-\infty, 0]$) and reset $x' = x + 3$. Then the system is controllable from $x_0 = 1$ to $T = (2, 4)$ since the hybrid trajectory $\xi(t, 0) = 1 - t$ for $t \in [0, 1]$ and $\xi(1, 1) = 3$ is a solution. However the perturbed system H' with $A' = [-\infty, -\epsilon]$ for $\epsilon > 0$ is blocking at state $x = 0$, since no any trajectory must leave D' before entering A'. Hence H' is not controllable.

The system H' in the above example does not satisfy the Assumption 2.4. This shows that the non-blocking assumption $D \cup A = X$ is not a *topological* condition, but a *logical* condition on the flow, and that a numerical approach to computing the system evolution without explicitly considering this non-blocking assumption will necessarily fail.

Example 4.3: Consider a hybrid system with state space $\mathbb{R}^2$, flow $\dot{x} = 1,\ \dot{y} = \alpha$, activation $A = \{(x, y) \mid y \geqslant x^2\}$, delay set $D = \mathbb{R}^2 \backslash A$ and reset $(x', y') = (x + 2, y + \beta)$. Suppose the initial state is $p_0 = (-1, 0)$ and the target set T is $\{(x, y) \mid (x-2)^2 + (y-\gamma)^2 \leqslant 1/2\}$. Then if $\alpha = 0$ and $\gamma = \beta = 1$, the continuous evolution touches the guard set $G = \partial A$ at $p_1 = (0, 0)$ when $t = 1$ and jumps to $p_2 = (2, 1)$ in T. Now suppose that there is a small negative drift of y in the flow, so $\alpha \in (-\epsilon, 0)$. Then the continuous evolution misses the guard set and so misses the target. The system is not robustly controllable.

Now suppose $\gamma = \beta = 0$. Then for $\alpha \in [0, \epsilon)$ the continuous evolution hits the guard set at $p_1 \approx (0, 0)$ and jumps to $p_2 \approx (2, 0)$ in T. For $\alpha \in (-\epsilon, 0)$, the continuous evolution misses the guard set, but continues to reach the target set at $p_3 = (2, 3\alpha)$. Hence the system is robustly controllable from x_0 to T, even though we cannot say which path it follows.

The above example illustrates that discontinuity points of the evolution, such as points of tangential grazing with guard sets, can cause non-robustness of the controllable set, and hence that it may not be possible to compute the controllable set using approximative numerical methods. The difficulty is that there are two possibilities for the evolution, either an event occurs or an event does not occur, and due to numerical errors we cannot determine which. We need to consider both possibilities, and can only deduce that the system is controllable if we can continue from both eventualities to the target set.

Let us now consider how we might perform forwards reachability analysis from an initial state. Due to numerical error, we may encounter points for which we cannot determine whether an event occurs in the model or not, and if so, which event does occur. Notice that this numerical nondeterminism, on which we have no control, is different from the system nondeterminism, which we assume can be controlled by user input. We therefore consider all possible cases, and only say that a point is controllable if the target can be reached in all possible continuations. It may even be the case that the evolution slides along the common boundary of D_e and A_e for some event e, before entering A_e°, in which case we have a compact set of possibilities, each corresponding to a different event time. Since we have to consider all possible different qualitative behaviours, possibly infinitely many, and show that for each the system could be controlled into the target set, the procedure for controllability is quite complicated. It turns out that it is easier to analyse the system using a backwards analysis.

V. Computability of the controllable set

In this section, we compute the set of points which can be robustly controlled into a target set by a recursive backwards construction. The construction is

based on the *one-step controllable set*, which contains all points which can be controlled into T either by purely continuous evolution, or by continuous evolution followed by a single jump.

Definition 5.1: Let $T \subset X$ be an open target set. We say that x is *one-step controllable* into T if there exists a continuous trajectory η with $\eta(0) = x$ such that

1) $\eta(t) \in \bigcap_{e\in E} D_e$ for all $t \in [0, \tau)$, and
2) $\eta(\tau) \in T \cup \big(\bigcup_{e\in E}(A_e \cap R_e^{-1}(T))\big)$.

It is clear that the controllable set C can be written as $C = \bigcup_{n=0}^{\infty} C_n$, where C_0 is the *continuously controllable set* and C_{n+1} is the one-step controllable set for C_n.

When trying to *prove* numerically that a point x is one-step controllable into T, we need to consider a *robust* verision of the one-step controllable set, which involves taking the *interiors* of the sets D_e and A_e, and also checking the invariant at time $t = \tau$. The conditions for one-step controllability become

1) $\eta(t) \in \bigcap_{e\in E} D_e^\circ$ for all $t \in [0, \tau]$, and.
2) $\eta(\tau) \in T \cup \bigcup_{e\in E}(A_e^\circ \cap R_e^{-1}(T)))$.

Unfortunately, this presents us with a problem. For if e is an *urgent* event, by which we mean $D_e = X \setminus \overline{A}_e$, then $D_e^\circ \cap A_e^\circ = \emptyset$, so any continuous trajectory entering A_e° must first leave D_e°, which is forbidden by Definition 2.3. If, on the other hand, D_e and A_e are open for all e, then the robustly controllable set and the one-step controllable set are equal.

In order to solve the problem, we use the conditions of Assumption 2.4 to analyse the defining formula for the controllable set before passing to the robust interpretation. Suppose x is one-step controllable into T due to an event which occurs at time τ. Then

$$\forall t \in [0, \tau),\ \eta(t) \in \bigcap_{e\in E} D_e$$

However, x is also one-step controllable into T if an event occurs at time $t < \tau$ for which we can jump into T. Hence we can weaken the controllability condition to

$$\forall t \in [0, \tau),\ \eta(t) \in \big(\textstyle\bigcap_{e\in E} D_e\big) \cup \big(\bigcup_{e\in E}(A_e \cap R_e^{-1}(T))\big).$$

By taking the sets $A_e \cup R_e^{-1}(T)$ into the first set in the above formula, we have

$$\begin{aligned}\forall t \in [0, \tau),\ \eta(t) \in \big(\textstyle\bigcap_{e\in E}(D_e \cup (A_e \cap R_e^{-1}(T)))\big)\\ \cup \big(\textstyle\bigcup_{e\in E}(A_e \cap R_e^{-1}(T))\big).\end{aligned}$$

Now since $D_e \cup A_e = X$, we have

$$\begin{aligned}D_e \cup (A_e \cap R_e^{-1}(T)) &= (D_e \cup A_e) \cap (D_e \cup R_e^{-1}(T))\\ &= X \cap (D_e \cup R_e^{-1}(T)) = D_e \cup R_e^{-1}(T).\end{aligned}$$

Hence the flow condition 1) is equivalent to

$$\begin{aligned}\forall t \in [0, \tau),\ \eta(t) \in \big(\textstyle\bigcap_{e\in E}(D_e \cup R_e^{-1}(T))\big)\\ \cup \big(\textstyle\bigcup_{e\in E}(A_e \cap R_e^{-1}(T))\big).\end{aligned}$$

Taking a robust version of this predicate gives

$$\begin{aligned}\forall t \in [0, \tau],\ \eta(t) \in \big(\textstyle\bigcap_{e\in E}(D_e^\circ \cup R_e^{-1}(T))\big)\\ \cup \big(\textstyle\bigcup_{e\in E}(A_e^\circ \cap R_e^{-1}(T))\big).\end{aligned}$$

At the final time, the robust variant of 2) is

$$\eta(\tau) \in T \cup \big(\textstyle\bigcup_{e\in E}(A_e^\circ \cap R_e^{-1}(T))\big).$$

Note that the improvement in the condition for being controllable occurs *only* in the robust version, and not in the formal version.

Definition 5.2: Let $T \subset X$ be an open target set. We say that x is *robustly one-step controllable* into T if there exists a continuous trajectory η with $\eta(0) = x$ such that

1) $\eta(t) \in \bigcap_{e\in E}(D_e^\circ \cup R_e^{-1}(T))$ for all $t \in [0, \tau]$, and
2) $\eta(\tau) \in T \cup \big(\bigcup_{e\in E} A^\circ \cap R^{-1}(T))\big)$.

Note that we take $R_e^{-1}(T)$ in the whole space X, and not just in the set of points for which an action is possible. This is in order to keep the sets open, and does not cause any difficulties since if no transition is possible at a time t, then the conditions ensure that either a transition is possible at some time $t' > t$, or that the target may be reached without transitions.

Theorem 5.3 (Robustly controllable set): Let $H = (X, E, \Phi, \{D_e, A_e, R_e \mid e \in E\})$ be a hybrid system such that Φ is an overt multiflow and the R_e are overt multimaps. Suppose further that $D_e^\circ \cup A_e = X$ for all e, so the system is non-blocking. Let T be an open set. Then the robustly controllable set C is an open set, and can be effectively computed from names of T, Φ, R_e, D_e° and A_e°.

Proof: Let C_0 be the set of points which are continuously controllable into T, and C_{n+1} be the set of points which are one-step controllable into C_n. Note that trivially, every point of C_n is single-step controllable into C_n.

To show that C_0 is effectively computable we write

$$\begin{aligned}C_0 &= \big\{x \in X \mid \exists \eta \in \Phi(x), \tau \in \mathbb{Q}^+ \text{ s.t. } \eta(\tau) \in T\\ &\qquad \text{and } \eta([0,\tau]) \subset \textstyle\bigcap_{e\in E} D_e^\circ\big\}\\ &= \textstyle\bigcup_{\tau\in\mathbb{Q}^+} \Phi^{-1}\big(\beta(\{\tau\}, T) \cap \beta([0,\tau], \bigcap_{e\in E} D_e^\circ)\big)\end{aligned}$$

where $\beta(K, U)$ is given in (1). Now for $\overline{I}$ a compact interval with rational endpoints, in $\mathbb{R}$, and J a basic open subset of X, the set $\beta(\overline{I}, J) = \{\eta : \mathbb{R} \to X \mid \eta(\overline{I}) \subset J\}$ is a basic open subset of $C(\mathbb{R}; X)$. Since Φ is overt-valued and computable, the set $\Phi^{-1}(U)$ is

computable for any computable open $U \subset C(\mathbb{R};X)$. Hence C_0 is computable, since it can be obtained by applying computable operations to computable objects.

To show C_{n+1} is effectively computable we write

$$\begin{aligned} C_{n+1} = \{x \in X \mid \exists \eta \in \Phi(x),\ \tau \in \mathbb{Q}^+ \\ \text{s.t. } \eta(\tau) \in \textstyle\bigcup_{e\in E}(A_e^\circ \cap R_e^{-1}(C_n)) \\ \text{and } \eta([0,\tau]) \subset \textstyle\bigcap_{e\in E}(D_e^\circ \cup R_e^{-1}(C_n)) \\ = \textstyle\bigcup_{\tau\in\mathbb{Q}^+} \Phi^{-1}\big(\beta(\{\tau\}, \bigcup_{e\in E}(A_e^\circ \cap R_e^{-1}(C_n))) \\ \cap\, \beta([0,\tau], \textstyle\bigcap_{e\in E}(D_e^\circ \cup R_e^{-1}(C_n)))\big), \end{aligned}$$

expressing C_{n+1} in terms of computable operations on computable objects.

The result follows since $C = \bigcup_{n=0}^{\infty} C_n$ and countable union of open sets is computable. ■

Since the above proof gives an explicit formula for the robustly controllable set in terms of computable operations, we have an algorithm for computing this set which can in principle be effectively implemented (though highly non-trivial to implement efficiently). If the sets D_e and A_e are open we immediately obtain:

Theorem 5.4 (Interior controllable set): Let $H = (X, E, \Phi, \{D_e, A_e, R_e \mid e \in E\})$ be a hybrid system such that Φ is an overt multiflow, the R_e are overt multimaps, and the D_e and A_e are open sets. Suppose further that $D_e \cup A_e = X$ for all e, so the system is non-blocking. Let T be an open set. Then the controllable set equals the robustly controllable set, so is an open set, and can be effectively computed from names of T, Φ, R_e, D_e and A_e.

VI. Concluding Remarks

We have considered the effective computability of controllable sets for a hybrid system using techniques from computable analysis. We have seen that convergent under-approximations to the robustly controllable set can be computed from the system data, and that for some systems, the robustly controllable set equals the controllable set. The algorithm is based on a backwards computation of the controllable set, and is easy to write down in terms of fundamental computable operations. The algorithm can also be seen as computing the set of unsafe initial states of a closed-loop hybrid system with nondeterministic noise.

In further research, we plan to give an implementation of the algorithm within the hybrid systems analysis tool Ariadne. This will extend the existing functionality for reachability analysis and safety verification.

References

[1] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "Hytech: A model checker for hybrid systems," in *CAV '97: Proceedings of the 9th International Conference on Computer Aided Verification*. London, UK: Springer-Verlag, 1997, pp. 460–463.

[2] T. A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi, "Beyond hytech: Hybrid systems analysis using interval numerical methods," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, N. Lynch and B. Krogh, Eds., no. 1790. Berlin Heidelberg New York: Springer-Verlag, 2000, pp. 130–144.

[3] E. Asarin, T. Dang, and O. Maler, "d/dt: A verification tool for hybrid systems," in *Proceedings of the 40th IEEE Conference on Decision and Control*. New York: IEEE Press, 2001.

[4] B. K. B. Izaias Silva, Keith Richeson and A. Chutinan, "Modeling and verification of hybrid dynamical system using CheckMate," in *Proceedings of the 4th International Conference on Automation of Mixed Processes*, 2000, pp. 189–194.

[5] G. Frehse, "Phaver: Algorithmic verification of hybrid systems past hytech." in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, M. Morari and L. Thiele, Eds., vol. 3414. Springer, 2005, pp. 258–273.

[6] A. M. Bayen, E. Crück, and C. Tomlin, "Guaranteed overapproximations of unsafe sets for continuous and hybrid systems: Solving the hamilton-jacobi equation using viability techniques," in *HSCC*, ser. Lecture Notes in Computer Science, C. Tomlin and M. R. Greenstreet, Eds., vol. 2289. Springer, 2002, pp. 90–104.

[7] I. M. Mitchell, "Comparing forward and backward reachability as tools for safety analysis," in *Hybrid systems: computation and control*, ser. Lecture Notes in Comput. Sci. Berlin: Springer, 2007, vol. 4416, pp. 428–443.

[8] D. S. Scott, "Domains for denotational semantics," in *Automata, languages and programming (Aarhus, 1982)*, ser. Lecture Notes in Comput. Sci. Berlin: Springer, 1982, vol. 140, pp. 577–613.

[9] S. Vickers, *Topology via logic*, ser. Cambridge Tracts in Theoretical Computer Science. Cambridge: Cambridge University Press, 1989, vol. 5.

[10] K. Weihrauch, *Computable analysis*, ser. Texts in Theoretical Computer Science. An EATCS Series. Berlin: Springer-Verlag, 2000, an introduction.

[11] P. Collins, "Semantics and computability of the evolution of hybrid systems," Centrum voor Wiskunde en Informatica, Tech. Rep., 2008, CWI Report MAS-R0801.

[12] ——, "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.

[13] R. Goebel, J. Hespanha, A. R. Teel, C. Cai, and R. Sanfelice, "Hybrid systems: Generalized solutions and robust stability," in *Proceedings of the Symposium on Nonlinear Control Systems*. Elsevier, 2004.

[14] J. Lygeros, K. H. Johansson, S. Sastry, and M. Egerstedt, "On the existence of executions of hybrid automata," in *Proceedings of the 38th IEEE Conference on Decision and Control*. New York: IEEE Press, 1999, pp. 2249–2254.

[15] P. Collins, "Continuity and computability of reachable sets," *Theoret. Comput. Sci.*, vol. 341, no. 1-3, pp. 162–195, 2005.

[16] M. Escardó, "Synthetic topology of data types and classical spaces," *Electronic Notes in Theoretical Computer Science*, vol. 87, pp. 21–156, 2004. [Online]. Available: www.elsevier.com/locate/entcs

[17] P. Taylor, "A lambda calculus for real analysis," 2008, http://www.monad.me.uk/.

[18] A. Puri, P. Varaiya, and V. Borkar, "Epsilon-approximation of differential inclusions," in *Hybrid Systems III*, ser. LNCS, R. Alur, T. A. Henzinger, and E. D. Sontag, Eds., vol. 1066. Berlin: Springer, 1996, pp. 362–376.