Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

On the Computability of Reachable and Invariant Sets

Pieter Collins

Abstract— **The computation of reachable and invariant sets of nonlinear dynamic and control systems are important problems of systems theory. In this paper we consider the computability of these sets using Turing machines to perform approximate computations. We use Weihrauch's type-two theory of effectivity for computable analysis and topology, which provides a natural setting for performing computations on sets and maps. The main results are that the reachable set is lower-semicomputable, but upper-semicomputable only if it equals the chain-reachable set, whereas invariant sets are upper-semicomputable.**

Index Terms— **computable analysis, reachable set, invariant set, computable topological space, semicontinuous function, approximation**

AMS subject classifications. 93B40, 93B03

I. INTRODUCTION

The purpose of this paper is to introduce the methods of computable analysis and topology to discuss the computability of system-theoretic properties of nonlinear dynamic and control systems. We focus on the problems of computing reachable sets, viability kernels and invariance kernels, since these often arise in applications as *nonlinear verification* problems for the validation of controllability and safety properties, where a precise analysis of numerical methods is of utmost importance. Further, of all the important problems in nonlinear systems, these seem to be the most amenable to study by the methods of computable analysis and topology, and hence forms a good starting point for the application of these techniques.

There are already many tools which compute approximations to the reachable set, such as d/dt [1], CheckMate [2] and HyTech [3] for linear hybrid systems, and HyperTech [4] and PHAVer [5] for over-approximation of reachable sets. Computation of reachable sets can also be performed by the general-purpose package GAIO [6] for set-based computations. However, since general sets and functions cannot be represented exactly in a finite amount of data, there is always the question of what exactly what these packages compute, and what it is even possible to compute. In particular, we wish to know whether it is possible to compute approximations to reachable and invariant sets to some arbitrary, pre-specified accuracy.

We consider computability in the framework of *type-two effectivity* developed by Weihrauch [7] and co-workers. In this theory, computations are performed by standard Turing machines with *input*, *output*, and *work tapes*. Unlike standard computability theory (type-one effectivity) in which inputs and outputs are *words* (elements of Σ^*), type-two machines can compute on *sequences* (elements of Σ^ω). This allows representations of, and computations on, the standard objects of analysis and topology, such as real numbers, open, closed and compact sets, continuous functions and semicontinuous multivalued functions. Computable topology provides a *standard representation* for elements of a topological space, which allows the extraction of approximations by *denotable* elements with various error bounds. The main result of the theory is that only continuous functions and operators are computable in the standard representation.

We show that given arbitrarily good lower approximations to the initial set and the system, we can compute arbitrarily good lower approximations to the reachable set. Unfortunately, it is not possible, in general, to compute arbitrarily good upper approximations. Instead, for uniformly bounded systems, we show that it is possible to compute outer approximations to the *chain reachable set*, $\mathrm{ChainReach}(F,X_0)$, which contains all points which can be reached by introducing an arbitrarily small amount of noise. We show that the chain reachable set is the best possible upper-computable set containing the reachable set, and that the reachable set is computable to arbitrary precision if, and only if, $\mathrm{cl}(\mathrm{Reach}(F,X_0)) = \mathrm{ChainReach}(F,X_0)$. We present similar results for viability and invariance kernels, which are both upper-semicomputable, but have robust versions which can be effectively approximated from below.

We remark that the negative computability results presented here assume that the *only* information we have about sets and systems are lower and upper approximations. If more detailed information is available (e.g. an algebraic description in terms of polynomials with rational coefficients) then it may be possible to determine these sets exactly, even if they differ. In other words, a lack of computability in the approximative sense used here does not imply a lack of computability in some other computational framework. However, the framework of computable analysis can deal with arbitrary continuous systems, whereas the class of systems which can be considered by algebraic techniques is severely restricted.

There are a number of works in which set-based approximation methods have been used to study dynamic systems. There is a large body of literature on approximation methods in viability theory such as Aubin and Frankowska [8] and Cardaliaguet et. al. [9]. Approximation methods based on ellipsoidal techniques have been considered by Kurzhanski and Varaiya [10], [11]. A number of applications of set-valued methods to control problems are given in Szolnoki [12]. The relation between reachability and chain reachability has been considered by Asarin and Bouajjani [13]. Optimal controllers have been computed by [14] using the tool GAIO.

Centrum voor Wiskunde en Informatica, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands. Email: {Pieter.Collins}@cwi.nl

An alternative approximation framework based on first-order logic over the reals is given by Franzle [15], [16].

The paper is organised as follows. We first give a simple example system for which the reach set fails to be computable, in order to motivate the results of the rest of the paper. In Section II, we give an introduction to the topological aspects of the computable analysis of Weihrauch, which form the core techniques. In Section III, we relate the abstract representations of sets defined in Section II to concrete approximations by denotable elements. In Section IV we develop computable topology for semicontinuous multivalued maps, which provide our basic model for control systems. In Section V we apply these techniques to solve reachability problems for (semi)continuous systems. In Section VI we consider the problem of computing viability and invariance kernels. Finally, we state some conclusions and give directions for future work in Section VII.

Example 1: We now give a simple example which illustrates the difficulties involved in computing reachable sets. Consider the maps $f_\varepsilon : \mathbb{R} \to \mathbb{R}$ given by

$$f_\varepsilon(x) := \varepsilon + x + x^2 - 9x^4,$$

where ε is a small parameter. Since $f'_\varepsilon(x) = 1 + 2x - 36x^3$, the function f_ε is monotone increasing for $|x| \leqslant 5/15$.

For $\varepsilon = 0$, there are fixed points at $p = 0$, $q_- = -1/3$ and $q_+ = +1/3$. Since $f'_0(-1/3) = 5/3$ and $f'_0(1/3) = 1/3$, the fixed points q_- and q_+ are hyperbolic, and can be continued to give families of fixed points $q_-(\varepsilon)$ and $q_+(\varepsilon)$ for some neighbourhood of $\varepsilon = 0$. The fixed point p at $x = 0$ can be continued to two branches of fixed points $p_-(\varepsilon)$ and $p_+(\varepsilon)$ for $\varepsilon < 0$, but does not exist for $\varepsilon > 0$.

We now consider the reachable set starting from the initial point $x_0 = -1/4$. For ε sufficiently close to 0, we have $x_0 > q_-(\varepsilon)$ and $x_0 < p_-(\varepsilon)$ if $\varepsilon < 0$. Let $x_i = f^i_\varepsilon(x_0)$ for $i \in \mathbb{Z}^+$. Then the reachable set of f_ε starting from x_0 is just the orbit $\{x_i \mid i \in \mathbb{Z}^+\}$.

If $\varepsilon < 0$, then since $q_-(\varepsilon) < x_0 < p_-(\varepsilon)$, we have $f(q_-(\varepsilon)) < f(x_0) < f(p_-(\varepsilon))$ by monotonicity of f_ε, so $q_-(\varepsilon) < x_1 < p_-(\varepsilon)$. By induction, $x_i \in (q_-(\varepsilon), p_-(\varepsilon))$ for all i. Further, since $f(x) > x$ for $x \in (q_-(\varepsilon), p_-(\varepsilon))$, the orbit (x_i) is an increasing sequence in $[x_0, p_-(\varepsilon)]$. Indeed, we can show that $\lim_{i\to\infty} x_i = p_-(\varepsilon)$. In particular, $\mathrm{Reach}(f_\varepsilon, \{x_0\} \subset [x_0, p_-(\varepsilon)]$.

If $\varepsilon = 0$, we similarly see that $\mathrm{Reach}(f_0, \{x_0\}) \subset [x_0, 0]$.

If $\varepsilon > 0$, the situation is very different. If $\varepsilon > 0$ is sufficiently small, then $f_\varepsilon(x) > x$ for all $x \in (q_-(\varepsilon), q_+(\varepsilon))$, and $f_\varepsilon(x) \geqslant x + \varepsilon$ if $x \in [-1/3, +1/3]$. Since $f_\varepsilon(x) \geqslant x + \varepsilon$ for $x \in (-1/3, +1/3)$, it must be the case that $x_i > 1/3$ for some i. In fact, for ε sufficiently small, we have $\lim_{i\to\infty} x_i = q_+(\varepsilon)$. The reachable set is therefore not contained in a small neighbourhood of $[x_0, 0]$ for $\varepsilon > 0$, even if $\varepsilon \ll 1$, and in fact jumps discontinuously at $\varepsilon = 0$.

Hence, to find a good approximation to the reachable set, it is necessary to determine whether $\varepsilon > 0$. If ε is known precisely (e.g. ε is a given rational), then $\mathrm{Reach}(f_\varepsilon, \{x_0\})$ can be approximated to arbitrary precision. However, if $\varepsilon = 0$ but is only known approximately, then it is impossible to decide whether $\varepsilon > 0$, and hence find a good approximation to $\mathrm{Reach}(f_\varepsilon, \{x_0\})$. The reachable set is therefore uncomputable at $\varepsilon = 0$.

II. COMPUTABLE ANALYSIS AND TOPOLOGY

Computable analysis deals with real numbers, continuous functions on real and Euclidean spaces, and subsets of Euclidean spaces. In this section, we review the elements of the literature which we need. We assume familiarity with the basic concepts of general topology, which can be found in [17]. The computational topology used here for the representation of sets and functions is based mostly on Chapters 5 and 6 of Weihrauch [7], which the reader is strongly advised to consult for more details. We present results for Euclidean spaces, but the theory generalises to arbitrary second-countable locally-compact Hausdorff spaces. For more detailed description of computability on subsets of metric spaces, see Brattka and Presser [18] and Brattka [19].

A. Computability and naming systems

We consider computability in terms of words and sequences on a finite alphabet Σ. Computations are performed by Turing machines with n input tapes and a single output tape, each of which either contains a word or a sequence. A partial function $f :\subset Y_1 \times \cdots \times Y_n \to Y_0$ with $Y_i \in \{\Sigma^*, \Sigma^\omega\}$ for $i = 0, \ldots, n$ is *computable* if there is some Turing machine which outputs $y_0 = f(y_1, \ldots y_k)$ given input $(y_1, \ldots y_k)$, where in the case $Y_0 = \Sigma^*$ the computation halts with y_0 on the output tape, and in the case $Y_0 = \Sigma^\omega$ the computation continuous forever, writing y_0 on the output tape.

To relate computability of words and sequences to more general sets, we use *naming systems*. A *notation* of a set M is a surjective partial function $\nu :\subset \Sigma^* \to M$. A *representation* of a set M is a surjective partial function $\delta :\subset \Sigma^\omega \to M$. Given a naming system for a set M we can give a representation of words M^* and sequences M^ω. We will write $\langle w_0, w_1, w_2, \ldots \rangle$ for a such a tupling, and $w \lhd p$ if $w = w_i$ for some i.

Given a function $f : M_1 \times \cdots \times M_n \to M_0$, and naming systems $\gamma_i :\subset Y_i \to M_i$, we say f is $(\gamma_1, \ldots, \gamma_n; \gamma_0)$-*computable* if there exists $g :\subset Y_1 \times \cdots \times Y_n \to Y_0$ such that $\gamma_0(g(y_1, \ldots, y_n)) = f(\gamma_1(y_1), \ldots, \gamma_n(y_n))$.

B. Computable topological spaces

The essence of computable topology is to describe a point in a topological space in terms of the open sets it lies in. If (M, τ) is a Kolmogorov (T_0) space, and σ is a sub-base of τ, then every $x \in M$ is specified uniquely by $\{U \in \sigma \mid x \in U\}$. If σ is countable, this gives us a way of representing points in topological spaces in a way which respects the topology.

A *computable topological space* is a quadruple (M, τ, σ, ν) such that M is a non-empty set, τ is a topology on M, $\sigma \subset \tau$ is sub-base of τ, and $\nu : \Sigma^* \to \sigma$ is an notation for σ.

The *standard representation* $\delta_\mathbf{S}$ of a computable topological space $\mathbf{S} = (M, \tau, \sigma, \nu)$ is the representation $\delta_\mathbf{S} :\subset \Sigma^\omega \to M$ given by

$$\delta_\mathbf{S}(p) = x :\iff \{\nu(w) \mid w \lhd p\} = \{J \in \sigma \mid x \in J\}$$

Informally, we can think of the standard representation δ of (M,τ,σ,ν) as encoding a sequence $(J_i)_{i\in\mathbb{N}}$ containing all sets $J_i \in \sigma$ for which $x \in J_i$.

The exact computability properties of a computable topological space depend on the sub-base chosen. However, regardless of the choice of sub-base, only continuous functions are computable in the standard representation. We use the following form of this result, which is Corollary 3.2.12 of [7].

Theorem 2: For $i = 0,\dots,n$ let $\mathbf{S}_i = (M_i,\tau_i,\sigma_i,\nu_i)$ be a computable topological space, and δ_i the standard representation of $\mathbf{S}_i$. Then every $(\delta_1,\dots,\delta_n;\delta_0)$-computable function $f : M_1 \times \cdots \times M_k \to M_0$ is $(\tau_1,\dots,\tau_n;\tau_0)$-continuous.

A computable topological space (X,τ,β,ν) is a *computable Hausdorff space* if (X,τ) is a locally-compact separable Hausdorff space, and β is a base for τ such that for each $J \in \beta$, the set $\mathrm{cl}(J)$ is compact. The elements of β are the *basic (open) sets*, and the sets $\mathrm{cl}(J)$ for $J \in \beta$ are *basic compact sets*. We choose basic sets with nice geometric properties, such as cuboids, simplices, parallelepiped or ellipsoids. Following Brattka and Presser [18], we henceforth consider computable Hausdorff spaces for which intersection, disjointness and covering of basic sets can be effectively decided.

C. Representations of sets

We are interested in the representation of closed, open, and compact subsets of a Euclidean space X, which we denote $\mathscr{A}(X)$, $\mathscr{O}(X)$ and $\mathscr{K}(X)$, respectively. In each case there are natural topologies $\tau_<$ and $\tau_>$ of lower and upper convergence. Convergence in the Hausdorff or metric sense is equivalent to both lower and upper convergence.

The topologies $\tau_<^{\mathscr{A}}$, $\tau_>^{\mathscr{A}}$ and $\tau^{\mathscr{A}}$ on closed sets have standard representations $\psi_<$, $\psi_>$ and ψ defined as follows:

$$\begin{aligned}\psi_<(p) = A &:\iff \{\nu(w) \mid w \lhd p\} = \{J \in \beta \mid A \cap J \neq \emptyset\}\\ \psi_>(p) = A &:\iff \{\nu(w) \mid w \lhd p\} = \{J \in \beta \mid A \cap \overline{J} = \emptyset\}\\ \psi\langle p,q\rangle = A &:\iff \psi_<(p) = A \text{ and } \psi_>(q) = A.\end{aligned}$$

The lower representation $\psi_<$ encodes a list of all basic open sets intersecting A, and the upper representation $\psi_>$ encodes a list of all basic compact sets disjoint from A. The representation ψ is a combination of both $\psi_<$ and $\psi_>$.

Since an open set is the complement of a closed set, we can derive representations $\theta_<$, $\theta_>$ and θ for the topologies $\tau_<^{\mathscr{O}}$, $\tau_>^{\mathscr{O}}$ and $\tau^{\mathscr{O}}$ on $\mathscr{O}(X)$ from the representations of closed sets. In particular, the lower representation is:

$$\theta_<(p) = U :\iff \{\nu(w) \mid w \lhd p\} = \{J \in \beta \mid \overline{J} \subset U\}.$$

The topology $\tau_<^{\mathscr{K}}$ of lower convergence of compact sets is the the restriction of $\tau_<^{\mathscr{A}}$ to $\mathscr{K}$, and we use the same representation $\psi_<$. For $\tau_>^{\mathscr{K}}$, we have the representation:

$$\begin{aligned}\kappa_>(p) = C :\iff& \{(\nu(w_1),\dots,\nu(w_n)) \mid \langle w_1,\dots,w_n\rangle \lhd p\}\\ &= \{(J_1,\dots,J_k) \subset \beta \mid C \subset \textstyle\bigcup_{i=1}^k J_i\}\end{aligned}$$

The representation $\kappa_>$ encodes a list of all open covers $(J_1,\dots,J_k)$ of C, and is denoted $\kappa_>^{\mathrm{cv}}$ in [7].

Informally, we say a set-valued operator is *lower-semicomputable* if it is possible to compute a $\theta_<$ or $\psi_<$ name for the result (as appropriate), given a suitable representation of the input. Likewise, an operator is *upper-semicomputable* if it is possible to compute a $\psi_>$ or $\kappa_>$ name for the result. An operator is *computable* if it is both lower-semicomputable and upper-semicomputable.

Theorem 4.1.13 of [7] shows that union $(A,B) \mapsto A \cup B$ on $\mathscr{A}$ is computable, being $(\psi_<,\psi_<;\psi_<)$-computable and $(\psi_>,\psi_>;\psi_>)$-computable. Intersection is only upper-semicomputable, being $(\psi_>,\psi_>;\psi_>)$-computable but not $(\tau^{\mathscr{A}},\tau^{\mathscr{A}};\tau_<^{\mathscr{A}})$-continuous.

D. Representations of continuous functions

The natural topology for the space of continuous functions $f : X \to Y$ is the *compact-open topology*, $\tau^{\mathscr{C}}$. This topology is generated by sets $\{f \in C(X \to Y) \mid f(C) \subset U\}$ for C compact and U open. The *compact-open representation* is the standard representation of this topological space, and is given by

$$\begin{aligned}\delta^{\mathrm{co}}(p) = f :\iff& \{(\nu_X(w_1),\nu_Y(w_2)) \mid (w_1,w_2) \lhd p\}\\ &= \{(I,J) \in \beta_X \times \beta_Y \mid f(\overline{I}) \subset J\}.\end{aligned}$$

The representation δ^{co} encodes a list of pairs $(\overline{I},J)$ where $\overline{I}$ is a basic compact subset of X, J is a basic open subset of Y, and $f(\overline{I}) \subset J$ (equivalently, $\overline{I} \subset f^{-1}(J)$).

The compact-open representation has the following properties (see [7], Theorem 6.2.1).

Theorem 3:

1) The evaluation map $(f,x) \mapsto f(x)$ is $(\delta^{\mathrm{co}},\rho;\rho)$-computable where ρ denotes the standard representation of X and Y.
2) The composition map $(g,f) \mapsto g \circ f$ is $(\delta^{\mathrm{co}},\delta^{\mathrm{co}};\delta^{\mathrm{co}})$-computable.
3) The closed set-image map $(f,A) \mapsto \mathrm{cl}(f(A))$ for $A \in \mathscr{A}(X)$ is $(\delta^{\mathrm{co}},\psi_<;\psi_<)$-computable.
4) The set-image map $(f,C) \mapsto f(C)$ for $C \in \mathscr{K}(X)$ is $(\delta^{\mathrm{co}},\kappa_<;\kappa_<)$-computable, $(\delta^{\mathrm{co}},\kappa_>;\kappa_>)$-computable and $(\delta^{\mathrm{co}},\kappa;\kappa)$-computable.

III. APPROXIMATION METHODS

Although the standard representations given in Section II are convenient for a general analysis of computability properties, they require an infinite amount of data and infinite computation time. We often want to describe an element of a topological space set by giving an approximation using a finite amount of data which can be computed in finite time.

Consider a function $\xi :\subset \Sigma^* \to X$ whose range is a dense subset of X. We say an element $x \in X$ is *denotable* if $x = \xi(w)$ for some $w \in \mathrm{dom}(\xi)$. An *approximation representation* of (X,τ,ξ) is a function $\delta :\subset \Sigma^\omega \to X$ such that

$$\begin{aligned}\delta\langle w_1,w_2,\dots\rangle = x :\iff& \langle w_1,w_2,\dots\rangle \in \mathrm{dom}(\delta)\\ &\text{and } \lim_{i\to\infty} \xi(w_i) = x.\end{aligned}$$

In other words, an approximation representation encodes a convergent sequence of denotable elements $x_i := \xi(w_i)$.

Since no finite portion of a general convergent sequence gives any information about its limit, we restrict the domain of δ so that meaningful approximations can be extracted. Restricting to increasing/decreasing sequences (x_i) allows us to deduce lower/upper bounds for x. Similarly, restricting to *effective Cauchy sequences* with $d(x_i,x_j) < 2^{-\min\{i,j\}}$ allows us to deduce that $d(x_i,x) \leqslant 2^{-i}$ for any i.

For closed/compact sets, we can find approximation representations equivalent to the standard representations.

Definition 4 (Denotable set): A set B is denotable if there are basic compact sets $\overline{I}_1,\ldots,\overline{I}_k$ such that $B=\bigcup_{i=1}^{k}\overline{I}_i$.

There are approximation representations equivalent to the standard representations.

Theorem 5 (Approximation representations):

1) The representation of open sets by increasing sequences A_i with $A_i \subset A_j$ for $i<j$ is equivalent to $\theta_<$.
2) The representation of closed sets by sequences A_i with $A_i \subset N_{2^{-i}}(A_j)$ for $i<j$ is equivalent to $\psi_<$.
3) The representation of compact sets by decreasing sequences $A_j \subset A_i$ for $i<j$ is equivalent to $\kappa_>$.
4) The representation of compact sets by decreasing sequences with $A_j \subset A_i \subset N_{2^{-i}}(A_j)$ is equivalent to κ.

By terminating the computation of an approximation representation, an approximate result can be obtained in finite time.

IV. MULTIVALUED MAPS

In system theory, it is useful to consider multivalued maps $F : X \rightrightarrows Y$, since a control system $f : X \times U \to X$ can be specified by the multivalued map as $F(x) = f(x,U)$. We define $F(A) := \{y \in Y \mid \exists x \in A,\ y \in F(x)\}$. If $F : X \rightrightarrows Y$ and $G : Y \rightrightarrows Z$, then $G \circ F(x) : X \rightrightarrows Z$ is defined by $G(F(x)) = \{z \in Z \mid \exists y \in Y,\ y \in F(x) \text{ and } z \in G(y)\}$.

There are two natural set-valued preimages of $F : X \rightrightarrows Y$, the *weak preimage* $F^{-1}(B) := \{x \in X \mid F(x) \cap B \neq \emptyset\}$, and the *strong preimage*, $F^{\Leftarrow}(B) := \{x \in X \mid F(x) \subset B\}$. We say F is *lower-semicontinuous* if $F^{-1}(U)$ is open whenever U is open, or equivalently, if $F^{\Leftarrow}(A)$ is closed whenever A is closed. F is *upper-semicontinuous* if $F^{-1}(A)$ is closed whenever A is closed, or equivalently, if $F^{\Leftarrow}(U)$ is open whenever U is open. F is *weakly upper-semicontinuous* if $F^{-1}(C)$ is closed whenever C is compact. F is *(weakly) continuous* if it is both lower-semicontinuous and (weakly) upper-semicontinuous.

For more information on multivalued functions, see [20].

A. Representations of multivalued semicontinuous functions

We are most interested in representations of lower-semicontinuous open- and closed-valued functions, weakly upper-semicontinuous closed-valued functions, and upper-semicontinuous compact-valued functions.

The standard representation $\mu_<^{\psi}$ of $LSC^{\mathscr{A}}$ is given by

$$\mu_<^{\psi}(p) :\iff \{(\nu_X(v),\nu_Y(w)) \mid \langle v,w\rangle \lhd p\} = \{(I,J) \in \beta_X \times \beta_Y \mid \overline{I} \subset F^{-1}(J)\}.$$

A $\mu_<^{\psi}$-name of $F \in LSC^{\mathscr{A}}$ encodes a list of all pairs $(\overline{I},J)$ such that $\overline{I} \subset F^{-1}(J)$. Similarly, a $\mu_<^{\theta}$-name of $G \in LSC^{\mathscr{O}}$ encodes a list of all pairs $(\overline{I},\overline{J})$ such that $\forall x \in \overline{I},\ \overline{J} \subset G(x)$, a $\mu_>^{\psi}$-name of $F \in USC^{\mathscr{A}}$ encodes a list of all pairs $(\overline{I},\overline{J})$ such that $\overline{I} \cap F^{-1}(\overline{J}) = \emptyset$, and a $\mu_>^{\kappa}$-name of $F \in USC^{\mathscr{K}}$ encodes a list of all pairs $(\overline{I},J_1,\ldots,J_k)$ such that $F(\overline{I}) \subset \bigcup_{i=1}^{k} J_i$.

B. Computing images and preimages of sets

The following theorem shows that if the set-image operator is continuous, then it is computable.

Theorem 6:

1) $(G,U) \mapsto G(U)$ is $(\mu_<^{\theta},\theta_<;\theta_<)$-computable.
2) $(F,A) \mapsto \mathrm{cl}(F(A))$ is $(\mu_<^{\psi},\psi_<;\psi_<)$-computable.
3) $(F,C) \mapsto F(C)$ is $(\mu_>^{\psi},\kappa_>;\psi_>)$-computable and $(\mu^{\psi},\kappa;\psi)$-computable.
4) $(F,C) \mapsto F(C)$ is $(\mu_>^{\kappa},\kappa_>;\kappa_>)$-computable and $(\mu^{\kappa},\kappa;\kappa)$-computable.

However, the image of a closed set under a compact-valued upper-semicontinuous map need not be closed, and the closure is not continuous or computable.

It is immediate from Thm. 6 that the preimage $(G,U) \mapsto G^{-1}(U)$ is $(\mu_<^{\theta},\theta_<;\theta_<)$-computable and that $(F,C) \mapsto F^{-1}(C)$ is $(\mu_>^{\psi},\kappa_>;\psi_>)$-computable. Additionally,

Theorem 7: The preimage operator $(F,U) \mapsto F^{-1}(U)$ is $(\mu_<^{\psi},\theta_<;\theta_<)$-computable.

V. REACHABILITY PROBLEMS

We now apply the material developed so far to the study of the reachability problem for semicontinuous systems.

A. Computability of the reachable set

Definition 8 (Reachability): Let $F : X \rightrightarrows X$ be a multivalued map, and $X_0 \subset X$. Then the *reachable set* of F from X_0 is

$$\mathrm{Reach}(F,X_0) := \{y \in X \mid \exists x_0,\ldots,x_n \text{ s.t. } x_0 \in X_0,\ x_{i+1} \in F(x_i) \text{ and } x_n = y\}.$$

It is easy to see that $\mathrm{Reach}(F,X_0) = \bigcup_{n=0}^{\infty} F^n(X_0)$. As the reachable set need not be closed, we define the *closed reachable set* $\overline{\mathrm{Reach}}(F,X_0) := \mathrm{cl}(\mathrm{Reach}(F,X_0))$.

To compute reachable sets, we need countable unions:

Lemma 9: Countable closed union $(A_1,A_2,\ldots) \mapsto \mathrm{cl}(\bigcup_{n\in\mathbb{N}} A_n)$ on $\mathscr{A}$ is $(\psi_<,\psi_<,\ldots;\psi_<)$-computable.

The closed reachability operator is lower-computable.

Theorem 10 (Computability of closed reachability):

1) The closed reachability operator $(F,A) \mapsto \overline{\mathrm{Reach}}(F,A)$ is $(\mu_<^{\psi},\psi_<;\psi_<)$-computable.
2) The closed reachability operator for bounded discrete-time systems is not $(\tau^{\mathscr{MK}},\tau^{\mathscr{K}};\tau_>^{\mathscr{K}})$-continuous.

We can also compute reachable sets for open-valued lower-semicontinuous systems.

Theorem 11: The reachability operator $(G,U) \mapsto \mathrm{Reach}(G,U)$ is $(\mu_<^{\theta},\theta_<;\theta_<)$-computable.

Indeed, it may be numerically more efficient to compute the reachable set of an open-valued system than the closed reachable set of a closed-valued system.

We can use Theorem 10(1) to verify system controllability. Suppose we wish to check whether it is possible to reach an open set U starting from some initial point x. We compute a $\psi_<$-name of $\overline{\mathrm{Reach}}(F,\{x\})$, and verify controllability if the $\psi_<$-name contains some set J with $\overline{J} \subset U$. If U is not reachable, then the procedure does not terminate.

Theorem 10(2) shows that we cannot, in general, compute a converging over-approximation to $\overline{\mathrm{Reach}}(F,C)$.

B. The chain-reachable set

We now briefly recall the concept of ε-chains as considered by Conley [21].

Definition 12: A sequence of points $x_0, x_1, \ldots, x_n$ is an ε-*chain* of $F : X \rightrightarrows X$ if there exist $y_1, \ldots, y_n \in X$ with $y_{i+1} \in F(x_i)$ and $d(y_{i+1}, x_{i+1}) < \varepsilon$ for $i = 0, \ldots, n-1$.

The ε-reachable set of F from X_0 is

$$\text{Reach}(F, X_0, \varepsilon) := \{x \in X \mid \exists\ \varepsilon\text{-chain } x_0, x_1, \ldots, x_n \text{ for } F \text{ s.t. } x_0 \in X_0 \text{ and } x_n = x\}$$

The *chain reachable set* of F from X_0 is

$$\text{ChainReach}(F, X_0) := \textstyle\bigcap_\varepsilon \text{Reach}(F, X_0, \varepsilon).$$

It is straightforward to show [21] that $\text{ChainReach}(F, X_0)$ is closed for any system F and any initial set X_0.

The following characterisation of the chain reachable set is useful for computability analysis.

Lemma 13: Let $F \in USC^{\mathscr{K}}$ and C a compact set. Suppose $\text{ChainReach}(F, C)$ is compact. Then

$$\text{ChainReach}(F, C) = \textstyle\bigcap\{U \supset C \mid F(\text{cl}(U)) \subset U\}.$$

The chain-reachable set is an upper-computable outer-approximation to the reachable set.

Theorem 14:

1) $(F, C) \mapsto \text{ChainReach}(F, C)$ is neither $(\tau^{\mathscr{MK}}, \tau^{\mathscr{K}}, \tau_>^{\mathscr{A}})$-continuous nor $(\tau^{\mathscr{MK}}, \tau^{\mathscr{K}}; \tau_<^{\mathscr{A}})$-continuous.
2) If $\text{ChainReach}(F, C)$ is compact, then $(F, C) \mapsto \text{ChainReach}(F, C)$ is $(\mu_>^{\kappa}, \kappa_>; \kappa_>)$-computable.

We can use Theorem 14(2) to verify system safety. Suppose we wish to check that $\text{Reach}(F, X_0) \subset S$ for some safe set S. Since $\text{Reach}(F, X_0)$ is not upper-computable, it is not possible to verify $\text{Reach}(F, X_0) \subset S$ directly. However, we have $\text{Reach}(F, X_0) \subset \text{ChainReach}(F, X_0)$, and the inclusion $\text{ChainReach}(F, X_0) \subset S$ can be verified by computing an open cover $\{J_1, \ldots, J_k\}$ of $\text{ChainReach}(F, X_0)$.

We say that the reachable set is *robust* if $\overline{\text{Reach}}(F, C) = \text{ChainReach}(F, C)$. It is immediate from the definitions that $\text{ChainReach}(F, C) = \limsup_{(F', C') \to (F, C)} \text{Reach}(F', C')$. This means that $\text{ChainReach}(F, C)$ is the *best possible* outer approximation to $\text{Reach}(F, C)$.

Theorem 15: Any outer approximation to $\text{Reach}(F, C)$ computed from a μ^κ name of F and a κ-name of C must contain $\text{ChainReach}(F, C)$. $\overline{\text{Reach}}(F, C)$ is $(\mu^\kappa, \kappa; \kappa)$-computable if and only if $\overline{\text{Reach}}(F, C) = \text{ChainReach}(F, C)$ i.e. the reachable set is robust.

VI. VIABILITY AND INVARIANCE KERNELS

Viable and invariant sets are also important system properties. Recall that a set A is *viable* for a system F if for every point x of A, there is an orbit through x remaining in A, and *invariant* if every orbit starting in A remains in A. A viable set may also be described as *control-invariant*, and an invariant set as *perturbation invariant*. See [22] for a detailed exposition of viability theory.

A. Computation of viability kernels

We first consider the computation of the maximal viable subset of a given set.

Definition 16: The *viability kernel* of A under F is

$$\text{Viab}(F, A) := \{x \mid \exists x_0, x_1, \ldots \text{ s.t. } x = x_0, \text{ and } \forall i,\ x_{i+1} \in F(x_i) \text{ and } x_i \in A\}.$$

It is easy to see that $\text{Viab}(F, C) = \bigcap_{n=0}^{\infty} F^{-n}(C)$.

It was shown by Saint-Pierre [23] that the viability kernel varies upper-semicontinuously in (F, C), and an algorithm to compute it was given. The viability kernel is also outer-computable in the framework of computable analysis.

Theorem 17:

1) $(F, C) \mapsto \text{Viab}(F, C)$ is $(\mu_>^{\psi}, \kappa_>; \kappa_>)$-computable.
2) $(F, C) \mapsto \text{Viab}(F, C)$ is not $(\tau^{\mathscr{MK}}, \tau^{\mathscr{K}}; \tau_<^{\mathscr{A}})$-continuous, so is not $(\mu^\kappa, \kappa; \psi_<)$-computable.

Unfortunately, it is not possible to compute a good lower-approximation to $\text{Viab}(F, C)$ for a compact set C.

Example 18: Let $F(x) = 2x$ and $C = [0, 1]$. We can take approximations C_n to C by finite sets of rational points, and (lower or upper) semicontinuous approximations F_n to F mapping rational points to irrational points. Then $F_n(C_n) \cap C_n = \emptyset$ for all n, so $\text{Viab}(F_n, C_n) = \emptyset$. Hence for any (F, C) we have $\liminf_{(F', C') \to (F, C)} \text{Viab}(F, C) = \emptyset$.

The following example shows that the viability kernel may depend continuously on the system.

Example 19: Consider $F \in C(\mathbb{R} \rightrightarrows \mathbb{R})$ given by $F(x) = \{2x\}$, and $C = [-1, 1]$. Then, clearly, $\text{Viab}(F, C) = \{0\}$. Further, $\text{Viab}(F, C) \neq \emptyset$ for any continuous perturbation of F in $C(\mathbb{R} \rightrightarrows \mathbb{R})$.

Recall that a set A is viable if $A \subset F^{-1}(A)$. We say that A is *robustly viable* if $\text{cl}(A) \subset F^{-1}(\text{int}(A))$.

Definition 20: The *robust viability kernel* of U is

$$\text{RobustViab}(F, U) := \textstyle\bigcup\{C \subset U \mid C \subset F^{-1}(\text{int}(C))\}.$$

If F is lower-semicontinuous, then $F^{-1}(V)$ is open whenever V is open, and it is easy to see that the robust viability kernel is open. Using Theorem 7, we can show it is also computable.

Theorem 21: The operator $(F, U) \mapsto \text{RobustViab}(F, U)$ is $(\mu_<^{\psi}, \theta_<; \theta_<)$-computable.

B. Computation of invariance kernels

We now consider computability of the maximal invariant subset of a given set.

Definition 22: The *invariance kernel* of A under F is

$$\text{Inv}(F, A) := \{x \mid \forall x_0, x_1, \ldots \text{ s.t. } x_0 = x \text{ and } x_{i+1} \in F(x_i),\ x_i \in A\ \forall i\}.$$

Equivalently, $\text{Inv}(F, A) = X \setminus \bigcup_{n=0}^{\infty} F^{-n}(X \setminus A)$.

It is trivial to show that we can compute set differences $U \setminus A$, $A \setminus U$ and $C \setminus U$. We obtain the following result on computability of the invariance kernel:

Theorem 23:

1) $(F, A) \mapsto \text{Inv}(F, A)$ is $(\mu_<^{\psi}, \psi_>; \psi_>)$-computable.
2) $(F, C) \mapsto \text{Inv}(F, C)$ is $(\mu_<^{\psi}, \kappa_>; \kappa_>)$-computable.
3) $(F, C) \mapsto \text{Inv}(F, C)$ is not $(\tau^{\mathscr{MK}}, \tau^{\mathscr{A}}; \tau_<^{\mathscr{A}})$-continuous, so is not $(\mu^\kappa, \kappa; \psi_<)$-computable.

Notice that we can compute an *upper* approximation to $\mathrm{Inv}(F,C)$ using a *lower* approximation to F.

Just as in the case of the viability kernel, $\liminf_{(F',C')\to(F,C)} \mathrm{Inv}(F',C') = \emptyset$ for all (F,C). To obtain lower approximations to the invariance kernel, we consider robust invariance. Recall that a set A is invariant if $F(A) \subset A$. We say that A is *robustly invariant* if $F(\mathrm{cl}(A)) \subset \mathrm{int}(A)$, or equivalently, if $\mathrm{cl}(A) \subset F^{\Leftarrow}(\mathrm{int}(A))$.

Definition 24 (Robust invariance): The *robust invariance kernel* of U is

$$\mathrm{RobustInv}(F,U) := \bigcup\{C \subset U \mid F(C) \subset \mathrm{int}(C)\}.$$

If F is upper-semicontinuous, then $F^{\Leftarrow}(V)$ is open whenever V is open, and it is easy to see that the robust invariance kernel is open. We have the following computability result:

Theorem 25: The operator $(F,U) \mapsto \mathrm{RobustInv}(F,U)$ is $(\mu_{>}^{\kappa},\theta_{<};\theta_{<})$-computable.

VII. CONCLUSIONS AND FURTHER RESEARCH

We have considered the computability of reachable sets, viability kernels and invariance kernels in the setting of computable analysis and topology based on type-two computation. This theory provides a formal model of computation which can be realised on digital computers, and hence algorithms expressed in this theory can be practically implemented.

We have seen that the reachable set is in generally uncomputable in this approximative setting, but that lower approximations to the reachable set and outer approximations to the chain reachable set can be computed. The chain reachable set is the best possible outer approximation of the reachable set. The difference between the reachable and the chain reachable sets can be viewed as a measure of the "robustness" of the system. We have also shown that viability and invariance kernels can be outer-approximated, and robust viability and invariance kernels can be inner-approximated.

The results presented here have mostly been developed for discrete-time systems. By using set-based integration of Lipschitz differential inclusions [24], we can prove similar results for continuous-time systems. We can extend the results further to deal with hybrid-time systems using the methods of [25].

VIII. ACKNOWLEDGEMENT

The author gratefully acknowledges the financial support of the European Commission through the project Control and Computation (IST-2001-33520) of the Program Information Societies and Technologies.

REFERENCES

[1] E. Asarin, T. Dang, and O. Maler, "d/dt: A verification tool for hybrid systems," in *CDC*. New York: IEEE Press, 2001.

[2] B. K. B. Izaias Silva, Keith Richeson and A. Chutinan, "Modeling and verification of hybrid dynamical system using CheckMate," in *Proceedings of ADMP*, 2000, pp. 189–194.

[3] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "HyTech: A model checker for hybrid systems," *Software Tools for Technology Transfer*, vol. 1, pp. 110–122, 1997.

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

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

[6] M. Dellnitz, G. Froyland, and O. Junge, "The algorithms behind GAIO-set oriented numerical methods for dynamical systems," in *Ergodic theory, analysis, and efficient simulation of dynamical systems*, B. Fiedler, Ed. Berlin: Springer, 2001, pp. 145–174, 805–807.

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

[8] J.-P. Aubin and H. Frankowska, *Set-valued analysis*, ser. Systems & Control: Foundations & Applications. Boston: Birkhäuser, 1990.

[9] P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre, "Set-valued numerical analysis for optimal control and differential games," in *Stochastic and differential games*, ser. Ann. Internat. Soc. Dynam. Games. Boston: Birkhäuser, 1999, no. 4, pp. 177–247.

[10] A. B. Kurzhanski and P. Varaiya, "On ellipsoidal techniques for reachability analysis. I. External approximations," *Optim. Methods Softw.*, vol. 17, no. 2, pp. 177–206, 2002.

[11] ——, "On ellipsoidal techniques for reachability analysis. II. Internal approximations box-valued constraints," *Optim. Methods Softw.*, vol. 17, no. 2, pp. 207–237, 2002.

[12] D. Szolnoki, "Set oriented methods for computing reachable sets and control sets," *Discrete Contin. Dyn. Syst. Ser. B*, vol. 3, no. 3, pp. 361–382, 2003.

[13] E. Asarin and A. Bouajjani, "Perturbed Turing machines and hybrid systems," in *Proceedings of the Sixteenth Annual IEEE Symposium on Logic in Computer Science*. IEEE, 2001.

[14] O. Junge and H. Osinga, "A set oriented approach to global optimal control," *ESAIM Control Optim. Calc. Var.*, vol. 10, no. 2, pp. 259–279, 2004.

[15] M. Fräzle, "Analysis of hybrid systems: An ounce of realism can save an infinity of states," in *Computer Science Logic*, ser. LNCS, J. Flum and M. Rodriguez-Artalejo, Eds., no. 1683. Berlin Heidelberg New York: Springer-Verlag, 1999.

[16] ——, "What will be eventually true of polynomial hybrid automata," in *Theoretical Aspects of Computer Software*, ser. LNCS, N. Kobayashi and B. C. Pierce, Eds., no. 2215. Berlin Heidelberg New York: Springer-Verlag, 2001, pp. 340–359.

[17] J. R. Munkres, *Topology: a first course*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1975.

[18] V. Brattka and G. Presser, "Computability on subsets of metric spaces," *Theoretical Comp. Sci.*, vol. 305, pp. 43–76, 2003.

[19] V. Brattka, "Computability over topological structures," in *Computability and models*, ser. Univ. Ser. Math., S. B. Cooper and S. S. Goncharov, Eds. New York: Kluwer/Plenum, 2003, pp. 93–136.

[20] E. Klein and A. C. Thompson, *Theory of correspondences. Including applications to mathematical economics*, ser. Canadian Mathematical Society SERIES of Monographs and Advanced Texts. New York: Wiley, 1984.

[21] C. Conley, *Isolated Invariant Sets and the Morse Index*, ser. CBMS Regional Conference SERIES in Mathematics. Providence, Rhode Island: American Mathematical Society, 1978, vol. 38.

[22] J.-P. Aubin, *Viability theory*, ser. Systems & Control: Foundations & Applications. Boston: Birkhäuser, 1991.

[23] P. Saint-Pierre, "Approximation of the viability kernel," *Appl. Math. Optim.*, vol. 29, no. 2, pp. 187–209, 1994.

[24] 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.

[25] P. Collins and J. Lygeros, "Computability of finite-time reachable sets for hybrid systems," proceedings of CDC/ECC 2005.