Ensuring Safety of Nonlinear Sampled Data Systems through Reachability (Extended Version)*

Ian M. Mitchell† Mo Chen‡ Meeko Oishi§

March 2012

Abstract

In sampled data systems the controller receives periodically sampled state feedback about the evolution of a continuous time plant, and must choose a constant control signal to apply between these updates; however, unlike purely discrete time models the evolution of the plant between updates is important. In contrast, for systems with nonlinear dynamics existing reachability algorithms—based on Hamilton-Jacobi equations or viability theory—assume continuous time state feedback and the ability to instantaneously adjust the input signal. In this paper we describe an algorithm for determining an implicit surface representation of minimal backwards reach tubes for nonlinear sampled data systems, and then construct switched, set-valued feedback controllers which are permissive but ensure safety for such systems. The reachability algorithm is adapted from the Hamilton-Jacobi formulation proposed in Ding and Tomlin (2010). We show that this formulation is conservative for sampled data systems. We implement the algorithm using approximation schemes from level set methods, and demonstrate it on a modified double integrator.

1 Introduction

A wide variety of reachability algorithms for continuous and hybrid systems have been proposed in the literature over the last decade, but they have for the most part been driven by safety verification problems; for example, given initial and terminal sets in the state space,

*Research supported by the National Science and Engineering Council of Canada (NSERC) Discovery Grants #298211 (Mitchell) & #327387 (Oishi), an NSERC Undergraduate Student Research Award (Chen), and CanWheel, the Canadian Institutes of Health Research (CIHR) Emerging Team in Wheeled Mobility for Older Adults #AMG-100925.

†Department of Computer Science, University of British Columbia (email: mitchell@cs.ubc.ca)

‡Department of Electrical Engineering & Computer Science, University of California, Berkeley (email: mochen72@gmail.com)

§Department of Electrical & Computer Engineering, University of New Mexico (email: moishi@ece.unm.edu)

do there exist trajectories leading from the former to the latter? For the purposes of system design and debugging, this boolean decision problem is often augmented by a request for counterexamples if the system is unsafe (for example, see Clarke (2008)). When the system has inputs, however, there is a much less well-studied challenge: Given a particular state, how can those inputs be chosen to maintain safety?

Here we study that problem in the context of sampled data systems. A common design pattern in cyber-physical systems consists of a digital controller receiving periodically sampled state feedback about the continuous time evolution of a continuous (or hybrid) state plant, and then generating a control signal (typically constant) to use until the next sample time. Feedback controllers for such systems are often designed using discrete time approaches, but that treatment ignores the states through which the plant evolves between sample times. Sampled data control takes the continuous time trajectories of the plant into account.

In this paper we extend a sampled data reachability algorithm, first proposed in Ding and Tomlin (2010) and based on a Hamilton-Jacobi (HJ) partial differential equation (PDE) formulation, to synthesize safe but permissive switched feedback control policies for nonlinear continuous state sampled data systems. The contributions of this paper are:

- Adapting the algorithm to find minimal reach tubes and showing that these computed tubes are conservative estimates of the true reach tubes.
- Partitioning the state space into regions where the full control authority can be used safely, where only a subset may be used while maintaining safety, or where it may not be possible to maintain safety.

We seek to design a permissive but safe control policy (also known as a feedback control law). It is safe in the sense that if the system is in a state which is not identified as inevitably unsafe and control signals are chosen from this policy at the sample times then the system will never enter the unsafe set. It is permissive in the sense that it is set-valued when possible, so that other criteria can be taken into account in choosing the final control signal while still maintaining safety; for example, minimum control effort in an energy constrained situation, or proximity to the human operator's input in a collaborative control scenario.

The remainder of the paper is organized as follows. Section 2 formalizes the problem, while section 3 discusses related work. Section 4 adapts the sampled data reachability algorithm to minimal reach tubes and demonstrates conservatism, as well as showing how to determine the set of states which may be unavoidably unsafe. Section 5 determines the sets where limited control and where any control may be used to ensure safety. Section 6 discusses how to utilize information from the computation of the sampled data minimal reach tube to synthesize a switched, permissive (eg: set-valued) and safe control policy. Section 7 discusses implementation details, and section 8 demonstrates the algorithm on a simple example.

This technical report is an extended version (with a full proof of proposition 1 and additional material in section 8) of a paper presented at the 4^{th} IFAC Conference on the Analysis and Design of Hybrid Systems (Eindhoven, the Netherlands, June 6–8, 2012). After the conference, that version may be found at `http://www.ifac-papersonline.net/`

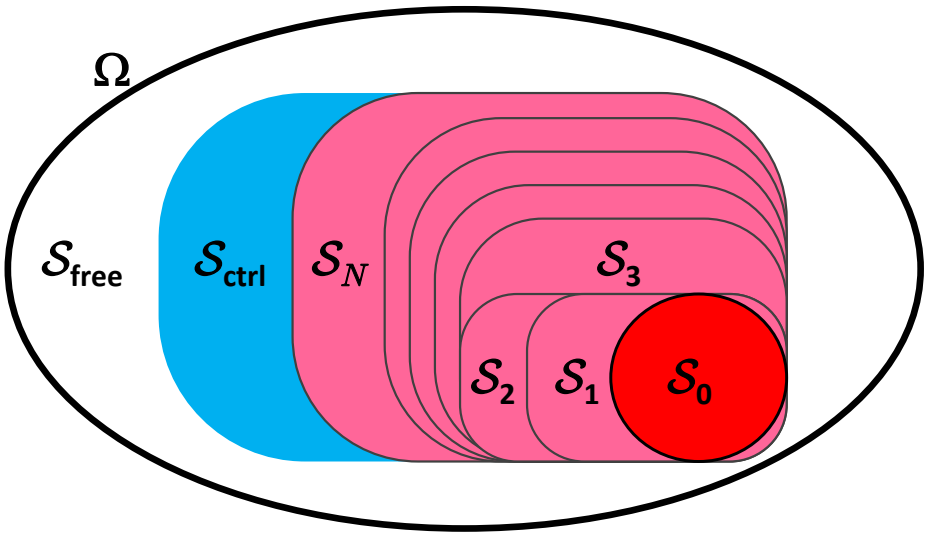

Figure 1: The partition of the state space. The unsafe set $\mathcal{S}_0$ and the state space Ω are specified in the problem definition. The uncontrollable sets $\mathcal{S}_k$ for $k > 0$, the mandatory control set $\mathcal{S}_{\text{ctrl}}$ and the free control set $\mathcal{S}_{\text{free}}$ are determined by the algorithms proposed in this paper.

2 Problem Definition

Consider a deterministic nonlinear system described by the ordinary differential equation (ODE)

$$\dot{x} = f(x, u) \tag{1}$$

with initial condition $x(0) = x_0$, where $x \in \Omega$ is the state, $\Omega \subset \mathbb{R}^{d_x}$ (or some similar vector space of dimension d_x), $u \in \mathcal{U}$ is the control input, $\mathcal{U} \subset \mathbb{R}^{d_u}$ is assumed to be convex and compact, and the dynamics $f : \Omega \times \mathcal{U} \to T\Omega$ are assumed to be bounded, Lipschitz continuous in x, and continuous in u.

We will assume that for feedback control purposes the state is sampled at times $t_k = k\delta$ for some fixed $\delta > 0$ and integer k, and that the control signal is constant between sample times. As a consequence, the actual dynamics are of the form

$$\dot{x}(t) = f(x(t), u_{\text{pw}}(t)) \tag{2}$$

where the piecewise constant input signal $u_{\text{pw}}(t)$ is chosen according to

$$u_{\text{pw}}(t) = u_{\text{fb}}(x(t_k)) \text{ for } t_k \le t < t_{k+1} \tag{3}$$

and $u_{\text{fb}} : \Omega \to \mathcal{U}$ is a feedback control policy. Note that because the feedback control policy is time sampled, the dynamics (2) *cannot* be written in the form $\dot{x} = f(x)$.

The unsafe set $\mathcal{S}_0 \subset \Omega$ that we seek to avoid is assumed to be the closure of an open set; for example, it need not be connected, but every unconnected component must have an interior and an exterior. We divide the state space Ω into a number of subsets as shown in figure 1.

The first is the terminal set $\mathcal{S}_0$. The next sets $\mathcal{S}_k$ may inevitably enter the terminal set by time $k\delta$ no matter what u_{fb} is chosen. The next set $\mathcal{S}_{\text{ctrl}}$ is a superset of $\mathcal{S}_N$—where $N\delta$ will be our safety horizon—and is the set in which we will constrain u_{fb} in order to ensure safety. The final set $\mathcal{S}_{\text{free}}$ is the set of states in which any u_{fb} may be chosen at the next sampling instant.

We will determine these sets through a series of reachability calculations. Starting from $\mathcal{S}_0$, we determine $\mathcal{S}_k$ for $k = 1, 2, \ldots, N$ through a sampled time backward minimal reachability calculation. We then use $\mathcal{S}_N$ as the terminal set in a traditional, continuous time backward maximal reachability calculation; $\mathcal{S}_{\text{ctrl}}$ is the resulting backwards reach tube. Finally, $\mathcal{S}_{\text{free}}$ is the remainder of the state space.

We represent sets $\mathcal{S} \subset \Omega$ using an implicit surface function $\psi_{\mathcal{S}} : \Omega \to \mathbb{R}$ such that

$$\mathcal{S} = \{x \in \Omega \mid \psi_{\mathcal{S}}(x) \leq 0\}.$$

The implicit surface function representation is very flexible; for example, it can represent nonconvex and disconnected sets. Its main restriction is that sets must have a nonempty interior and exterior. Analytic implicit surface functions for common geometric shapes (such as spheres, hyperplanes, prisms, etc.) are easily constructed. The constructive solid geometry operations of union, intersection and complement of sets are achieved through pointwise minimum, maximum and negation operations on their implicit surface functions.

For notational simplicity, we have restricted the presentation below to the case of a single input which is acting as a control. It is straightforward to modify the schemes in the same manner as Ding and Tomlin (2010) to treat measurable disturbance inputs or hybrid automata dynamics with continuously available mode switches. More extensive modifications would be required to treat the cases where the disturbance input and/or the mode switches were based on sampled data as well.

3 Related Work

We broadly categorize reachability algorithms into Lagrangian (those which follow trajectories of the system) and Eulerian (those which operate on a fixed grid); see Mitchell (2007) for a more extensive discussion of types of reachability algorithms. Most algorithms for systems with nonlinear dynamics and inputs are currently Eulerian; for example, there are schemes based on viability theory (Cardaliaguet et al. (1999); Aubin et al. (2011)), static HJ PDEs (Branicky and Zhang (2000); Sethian and Vladimirsky (2002)), or time-dependent HJ PDEs (Lygeros et al. (1999); Lygeros (2004); Mitchell et al. (2005)). In the static HJ PDE formulation, the value function is constant outside the reachable set; consequently, this formulation is not useful for synthesizing safe controls. Using the viability and time-dependent HJ PDE formulations, it is possible to synthesize control laws that are optimally permissive: constraints are only placed upon the choice of control along the boundary of the safe (viable) set. From a practical perspective, such policies are impossible to implement because they require information about the state at all times and the ability to change the

input signal at any time. In contrast, here we assume that state feedback and control signal modification only occur at the periodic sample times, and the control signal is held constant between sample times.

In Ding and Tomlin (2010) a time-dependent HJ PDE formulation of sampled data reachability is presented for hybrid automata. In that case, the HJ PDE is used to find an implicit surface representation of the sampled data backward reach tube, where the piecewise continuous control input signal attempted to drive the trajectory to a terminal set without entering an avoid set, despite the actions of a measurable disturbance input signal. In that problem the terminal set was considered "good" from the viewpoint of the control input, while in the problem we explore here the terminal set is considered "bad." In section 4 we describe the minor adaptation of the algorithm required to handle the different interpretation of terminal sets, and then examine the relationship between the resulting HJ PDE solutions and the desired reachability operators.

An alternative approach for finding safe trajectories is through sample based planning schemes (for example, see LaValle (2006)), such as the rapidly-exploring random tree (RRT) and its descendants. Adaptations of RRTs to verification/falsification are proposed in Branicky et al. (2006); Plaku et al. (2009), but to synthesize permissive yet safe control policies requires a slightly different but still quite feasible modification of traditional RRTs (to collect sets of safe paths, rather than just the optimal or first path found). Like many sample based schemes RRTs appear to scale better in practice to high dimensional systems than do schemes based on grids, and unlike most Lagrangian approaches they do a good job of covering the state space given sufficient samples. On the other hand, the output of RRTs is not as easily or accurately interpolated into continuous spaces as are grid-based results, and there is no simple method of introducing worst-case disturbance inputs to make the results robust to uncertainty.

4 Sampled Data Reachability

In this section we define a minimal reach tube operator for the sampled data dynamics (2)–(3), show how it can be represented as an implicit surface function constructed from the solutions of HJ PDEs via some simple pointwise operations, demonstrate its (possibly strict) conservatism, and specify a formula for $\mathcal{S}_k$.

4.1 Minimal Reach Tubes

Define the sampled data backward minimal reach tube over time interval $[t_{\min}, t_{\max}]$ as

$$\mathcal{R}^{-}_{\text{sd}}([t_{\min}, t_{\max}], \mathcal{T}) \triangleq \{x_0 \in \Omega \mid \forall u_{\text{pw}}(\cdot),\, \exists t \in [t_{\min}, t_{\max}],\, x(t) \in \mathcal{T}\}, \tag{4}$$

where $x(\cdot)$ solves (2)–(3) with initial condition $x(0) = x_0$. The backward minimal reach tube contains states that give rise to trajectories which cannot avoid entering $\mathcal{T}$ no matter what input signal is chosen. Note that this definition is not precisely the same as those in Mitchell (2007) because in this case we allow only piecewise constant control signals.

4.2 Hamilton-Jacobi Formulation

We adapt the approach from Ding and Tomlin (2010) to approximate the backward minimal reach tube. A finite subset of control input values $\underline{\mathcal{U}} = \{u^{(1)}, u^{(2)}, \ldots, u^{(\ell)}\}$ is considered, where $u^{(j)} \in \mathcal{U}$. Define the backward reach tube for a constant input value $u^{(j)}$ over a single sample period as

$$\mathcal{R}^{(j)} \triangleq \mathcal{R}([0,\delta],\mathcal{T}) \triangleq \{x_0 \in \Omega \mid \exists t \in [0,\delta],\, x(t) \in \mathcal{T}\}$$

where $x(\cdot)$ solves (1) with fixed input $u^{(j)}$ and initial condition $x(0) = x_0$. If $\mathcal{T}$ is represented by the known implicit surface function $\psi_{\mathcal{T}}$, then we can determine an implicit surface function for $\mathcal{R}^{(j)}$ (for example, see Mitchell et al. (2005))

$$\psi_{\mathcal{R}^{(j)}}(x) = \phi(x,0) \tag{5}$$

where $\phi : \Omega \times [0,\delta] \to \mathbb{R}$ is the viscosity solution of the terminal value, time-dependent HJ PDE

$$D_t\phi + \min\left[0, H(x, D_x\phi)\right] = 0 \tag{6}$$

with Hamiltonian

$$H(x,p) = p^T f(x, u^{(j)}) \tag{7}$$

and terminal condition

$$\phi(x,\delta) = \psi_{\mathcal{T}}(x). \tag{8}$$

Unlike Ding and Tomlin (2010), we use the control input to minimize the size of the reach tube, so an implicit surface function for

$$\mathcal{R}_1^-(\mathcal{T}) \triangleq \mathcal{R}_{\text{sd}}^-([0,\delta],\mathcal{T})$$

is given by

$$\psi_{\mathcal{R}_1^-(\mathcal{T})}(x) = \max_{1\le j\le \ell} \psi_{\mathcal{R}^{(j)}}(x). \tag{9}$$

This optimization ensures that if point x is outside $\mathcal{R}^{(j)}$ for any j—in other words, input $u^{(j)}$ generates a trajectory starting at x which does not reach $\mathcal{T}$ during the time interval $[0,\delta]$—then x is outside $\mathcal{R}_1^-(\mathcal{T})$. Because it is the maximum of continuous functions, $\psi_{\mathcal{R}_1^-(\mathcal{T})}$ is itself continuous.

Given some finite horizon $T = N\delta$ for integer $N > 1$, we work recursively through

$$\begin{aligned} \mathcal{R}_{k+1}^-(\mathcal{T}) &\triangleq \mathcal{R}_1^-(\mathcal{R}_k^-(\mathcal{T})) \\ \psi_{\mathcal{R}_{k+1}^-(\mathcal{T})}(x) &= \psi_{\mathcal{R}_1^-(\mathcal{R}_k^-(\mathcal{T}))}(x) \end{aligned} \tag{10}$$

and the equations above to find an implicit surface function for $\mathcal{R}_N^-(\mathcal{T})$.

4.3 Conservatism of the Hamilton-Jacobi Reach Tube Formulation

Proposition 1. *The true sampled data reach tube is a subset of the estimated reach tube computed in section 4.2:*

$$\mathcal{R}^-_{sd}([0, N\delta], \mathcal{T}) \subseteq \{x \in \Omega \mid \psi_{\mathcal{R}^-_N(\mathcal{T})}(x) \leq 0\}. \tag{11}$$

It may be a strict subset.

Proof. We can prove (11) by showing

$$x \in \mathcal{R}^-_{\text{sd}}([0, N\delta], \mathcal{T}) \implies \psi_{\mathcal{R}^-_N(\mathcal{T})}(x) \leq 0.$$

The proof itself is identical to the first part of lemma 8 in Mitchell et al. (2005) and so we do not repeat it here (all lemmas and the corollary referenced in this proof are from the same citation). To show that the true reach tube may be a strict subset of the computed tube, we consider the other part of lemma 8, which turns out to be false for the sampled data case:

$$\psi_{\mathcal{R}^-_N(\mathcal{T})}(x) \leq 0 \implies x \in \mathcal{R}^-_{\text{sd}}([0, N\delta], \mathcal{T}). \tag{12}$$

The proof of this part of lemma 8 fails because the dynamics (2) depend implicitly on time through the sampled control (3), the trajectories of the augmented system can be forced into states which are not visited by the original system, and hence lemma 4 and corollary 5 are false.

Of course, the failure of the proof in Mitchell et al. (2005) does not prove (12). As a concrete example for which $\mathcal{R}^-_k(\mathcal{T})$ contains states that are not in $\mathcal{R}^-([0, k\delta], \mathcal{T})$ for $k > 1$, consider the system

$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} u \\ -1 \end{bmatrix} = f(x, u)$$

with input set $\underline{U} = \{-1, +1\}$, sample time $\delta = 1$ and a terminal set $\mathcal{T}$ consisting of the two black triangles shown in figure 2. Trajectories can escape through the gap of width 0.5 between the two triangles near the origin. Figure 2 shows that $\mathcal{R}^-_1(\mathcal{T})$ is correctly computed as a diamond; however, for $k > 1$ the sets $\mathcal{R}^-_k(\mathcal{T})$ are too large and touch one another. As $k \to \infty$, $\mathcal{R}^-_k(\mathcal{T})$ will contain the entire upper half plane except for the narrow corridors running diagonally just above $\mathcal{T}$; however, it is easy to determine that $\mathcal{R}^-_{\text{sd}}([0, k\delta], \mathcal{T})$ actually looks more like a lattice of diamond shapes (each the size of $\mathcal{R}^-_1(\mathcal{T})$), as shown in figure 3. With this counterexample, we prove (12). □

The conservatism arises from the time dependence of (2). Care must be taken when adapting HJ PDEs to time-varying optimal control problems—for example, see Vladimirsky (2006)—and it is not yet clear how to adapt the formulation of reach tubes from Mitchell et al. (2005) to avoid conservatism when treating time-varying dynamics. A tempting approach is to work with the union over time of reach sets; however, for minimal or multi-input reachability the union of reach sets was shown to be unable to verify safety in Mitchell (2007), and the same problem occurs in the sampled data context.

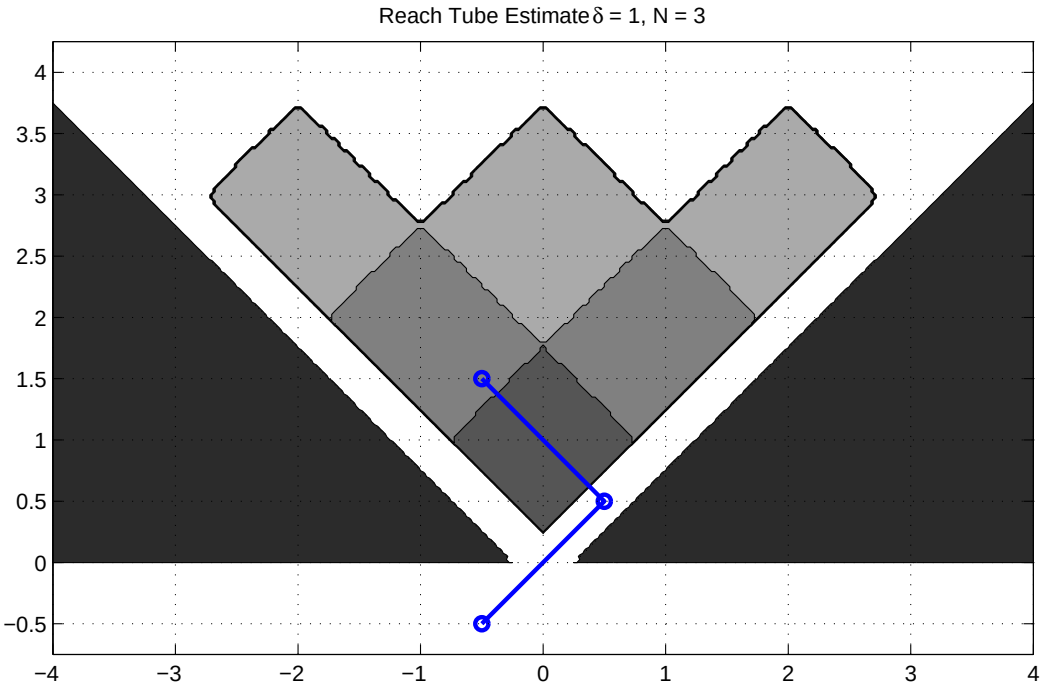

Figure 2: A demonstration that $\mathcal{R}_N^-(\mathcal{T})$ may include states which can avoid hitting the terminal set. The terminal set $\mathcal{T}$ is the two black triangles. The remaining shaded regions are $\mathcal{R}_k^-(\mathcal{T})$ for $k = 1, 2, 3$ (darkest to lightest). The blue line shows a trajectory starting from within $\mathcal{R}_2^-(\mathcal{T})$ which nonetheless avoids the terminal set. The input is sampled at the points marked by small circles.

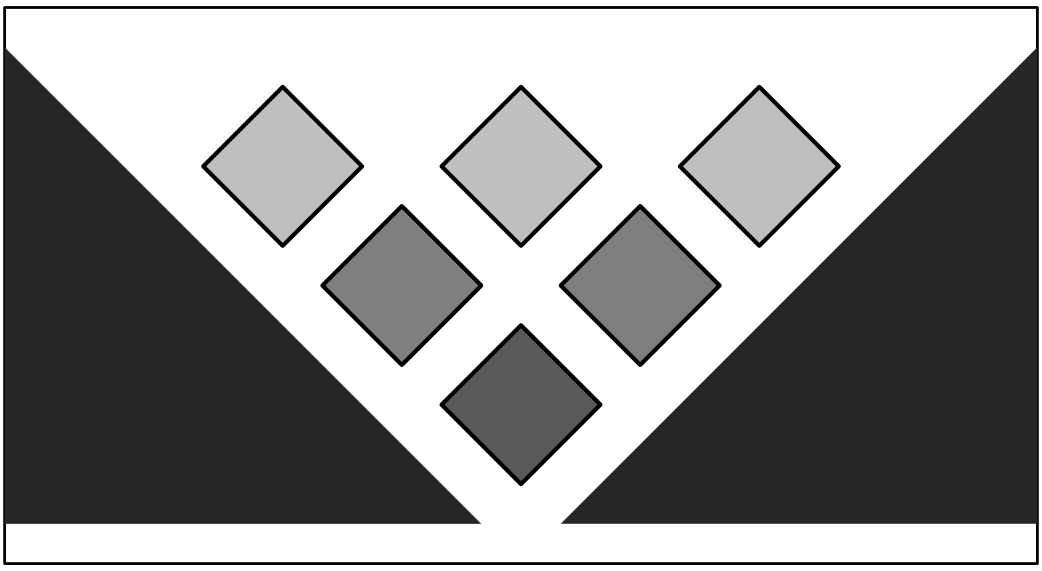

Figure 3: A sketch of the actual $\mathcal{R}^-([0, k\delta], \mathcal{T})$ for $k = 1, 2, 3$ for the example in figure 2.

4.4 Determining $\mathcal{S}_k$

Using the operators defined above, we (conservatively) determine the set of states that may inevitably enter $\mathcal{S}_0$ over time horizon $k\delta$ as

$$\mathcal{S}_k = \mathcal{R}_k^-(\mathcal{S}_0). \tag{13}$$

Using (5)–(10) we can determine an implicit surface representation $\psi_{\mathcal{S}_k}$.

For many problems of interest, we find that this calculation converges such that $\mathcal{S}_k = \mathcal{S}_{k-1}$ for sufficiently large k, and hence we can determine the infinite horizon $\mathcal{S}_\infty$.

5 Continuous Time Maximal Reachability

In order to allow maximum flexibility in the choice of control in $\mathcal{S}_{\text{free}}$ at time t, we must ensure that it contains no states which can give rise to trajectories that enter $\mathcal{S}_N$ before the next state observation and change of input at time $t + \delta$. In this section we recall the standard maximal reach tube operator and the HJ PDE whose solution provides an implicit surface representation of this set. We use this operator to determine $\mathcal{S}_{\text{ctrl}}$ and hence $\mathcal{S}_{\text{free}}$.

5.1 Maximal Reach Tubes

The backward maximal reach tube for a given terminal set $\mathcal{T}$ over time interval $[t_{\min}, t_{\max}]$ is defined as

$$\mathcal{R}^+([t_{\min}, t_{\max}], \mathcal{T}) \triangleq \{x_0 \in \Omega \mid \exists u(\cdot),\, \exists t \in [t_{\min}, t_{\max}],\, x(t) \in \mathcal{T}\}, \tag{14}$$

where $x(\cdot)$ solves (1) with initial condition $x(0) = x_0$. Note that this definition is exactly the same as in Mitchell (2007), and we allow for measurable input signals $u(\cdot)$ rather than those that are just piecewise constant.

5.2 Hamilton-Jacobi Formulation

We use the algorithm from Mitchell et al. (2005) to determine an implicit surface function representation of $\mathcal{R}^+([0, \delta], \mathcal{T})$:

$$\psi_{\mathcal{R}^+([0,\delta],\mathcal{T})}(x) = \phi(x, 0),$$

where $\phi : \Omega \times [0, \delta] \to \mathbb{R}$ is the viscosity solution of the terminal value, time-dependent HJ PDE (6) with Hamiltonian

$$H(x, p) = \min_u p^T f(x, u) \tag{15}$$

and terminal condition (8).

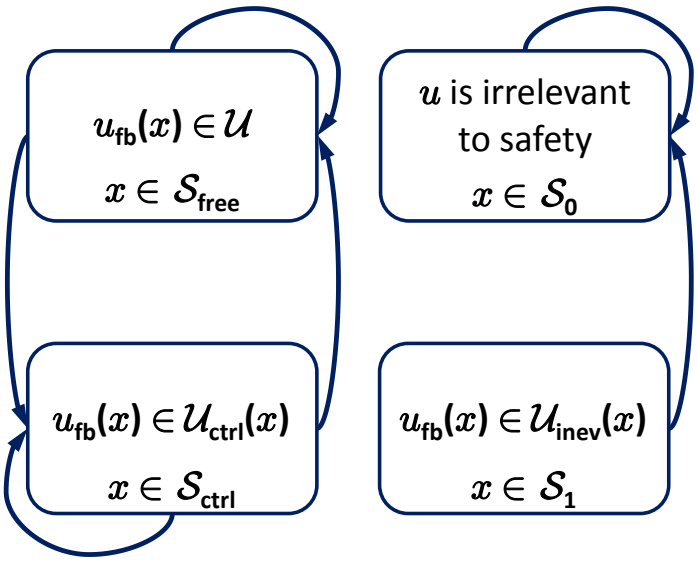

Figure 4: The general form of the switched sampled data control policy. Arrows show transitions which are possible under the policy.

5.3 Determining $\mathcal{S}_{\text{ctrl}}$ and $\mathcal{S}_{\text{free}}$

We choose

$$\mathcal{S}_{\text{ctrl}} = \mathcal{R}^+([0, \delta], \mathcal{S}_N) \setminus \mathcal{S}_1.$$

We omit $\mathcal{S}_1$ (and hence $\mathcal{S}_0$) because it is impossible to ensure safety for even a single sample period when starting in those states. By using a maximal reach set, we capture the set of states which might enter $\mathcal{S}_N$ over the next δ time units before the input is able to change. While the HJ PDE formulation of the continuous time backwards maximal reach tube is exact (unlike the sampled data case above), determining $\mathcal{S}_{\text{ctrl}}$ in this fashion is still conservative because it may include states which can only reach $\mathcal{S}_N$ by applying a control which is not piecewise constant.

Finally,

$$\mathcal{S}_{\text{free}} = \Omega \setminus \mathcal{R}^+([0, \delta], \mathcal{S}_N).$$

6 Safe but Permissive Sampled Data Feedback Control Policies

In this section we synthesize several sampled data control policies which are guaranteed to ensure that trajectories starting in $\mathcal{S}_{\text{free}} \cup \mathcal{S}_{\text{ctrl}}$ do not enter $\mathcal{S}_0$ during the time interval $[0, N\delta]$. All are switched feedback control policies of the form shown in figure 4. It was shown in Lygeros et al. (1999) that there exists a control policy which renders the system safe if and only if there exists a feedback control policy which renders the system safe, so we restrict ourselves to feedback control policies without loss of generality. We do not

synthesize a policy for $x \in \mathcal{S}_0$, since the system has already failed the safety criterion in such states and so its future evolution is irrelevant to that criterion.

6.1 Policy in $\mathcal{S}_{\text{free}}$

As shown in figure 4, it is impossible by the construction of $\mathcal{S}_{\text{ctrl}}$ for the state to go from $\mathcal{S}_{\text{free}}$ to $\mathcal{S}_0$ over a single sampling interval no matter what input is chosen. Consequently, the permissive control policy in $\mathcal{S}_{\text{free}}$ is always $u \in \mathcal{U}$.

6.2 Policies in $\mathcal{S}_{\text{ctrl}}$

For states in $\mathcal{S}_{\text{ctrl}}$ some choices of input may lead trajectories directly into $\mathcal{S}_0$ and others may unnecessarily reduce the horizon over which safety can be maintained; consequently, the range of inputs must be constrained. We propose two possible choices of $\mathcal{U}_{\text{ctrl}}(x)$. The first maximizes permissiveness by providing the largest set of controls which maintains safety over the maximum possible time horizon. The second aggressively attempts to drive the trajectory back into $\mathcal{S}_{\text{free}}$ (which might provide more long-term flexibility in choice of input). Both choices ensure safety, and are subsets of the finite collection of input samples $\underline{\mathcal{U}}$ used in constructing the $\mathcal{S}_k$.

Let $N\delta$ be the safety horizon used to compute $\mathcal{S}_{\text{ctrl}}$. Given $x_0 \in \mathcal{S}_{\text{ctrl}}$ define the safety horizon of x_0 as

$$n(x_0) \triangleq \begin{cases} \infty, & \text{if } \mathcal{S}_N = \mathcal{S}_{N-1} = \mathcal{S}_\infty \text{ and } \psi_{\mathcal{S}_N}(x_0) > 0; \\ N, & \text{if } \mathcal{S}_N \neq \mathcal{S}_{N-1} \text{ and } \psi_{\mathcal{S}_N}(x_0) > 0; \\ k, & \text{if } \psi_{\mathcal{S}_{k+1}}(x_0) \leq 0 \text{ and } \psi_{\mathcal{S}_k}(x_0) > 0. \end{cases} \tag{16}$$

Observe that $n(x) \geq 1$ because $\mathcal{S}_{\text{ctrl}}$ does not contain $\mathcal{S}_1$. Define the value at the next sample time under input $u^{(j)} \in \underline{\mathcal{U}}$ as

$$\psi_\delta^{(j)}(x_0) \triangleq \psi_{\mathcal{S}_{n(x_0)-1}}(x^{(j)}(\delta)), \tag{17}$$

where $x^{(j)}(\cdot)$ solves (2) with fixed input $u = u^{(j)}$ and initial condition $x(0) = x_0$.

The permissive policy is given by

$$\mathcal{U}_{\text{ctrl}}^{\rightarrow}(x) \triangleq \{u^{(j)} \in \underline{\mathcal{U}} \mid \psi_\delta^{(j)}(x) \geq \psi_{\mathcal{S}_{n(x)}}(x)\}. \tag{18}$$

The aggressive policy is given by

$$\mathcal{U}_{\text{ctrl}}^{\nearrow}(x) \triangleq \underset{u^{(j)} \in \underline{\mathcal{U}}}{\operatorname{argmax}}\, \psi_\delta^{(j)}(x). \tag{19}$$

Note that neither policy is guaranteed to be unique.

Proposition 2. *For all $x \in \mathcal{S}_{ctrl}$, $\mathcal{U}_{ctrl}^{\rightarrow}(x) \neq \emptyset$.*

Proof. The HJ PDE (6)–(8) used to compute $\psi_{\mathcal{R}^{(j)}}$ and the optimization in (9) implies that $\psi_{\mathcal{R}_1^-(\mathcal{T})}$ is the value function for the finite horizon terminal value optimal control problem

$$\psi_{\mathcal{R}_1^-(\mathcal{T})}(x_0) = \max_{u^{(j)} \in \underline{\mathcal{U}}} \min_{s \in [0,\delta]} \psi_{\mathcal{T}}(x^{(j)}(s)) \tag{20}$$

where $x^{(j)}(\cdot)$ solves (2) with initial condition $x^{(j)}(0) = x_0$ and constant control $u^{(j)}$.

Consider $x_0 \in \mathcal{S}_{\text{ctrl}}$. Let $\tilde{n} = n(x_0)$ and $\tilde{\psi} = \psi_{\mathcal{S}_{\tilde{n}}}(x_0)$. By (13) and (10), $\mathcal{S}_{\tilde{n}} = \mathcal{R}_1^-(\mathcal{S}_{\tilde{n}-1})$, which implies that

$$\psi_{\mathcal{R}_1^-(\mathcal{S}_{\tilde{n}-1})}(x_0) = \psi_{\mathcal{S}_{\tilde{n}}}(x_0) = \tilde{\psi}.$$

Therefore, by (20) there exists $\tilde{\jmath} \in \{1, 2, \ldots, \ell\}$ and $u^{(\tilde{\jmath})} \in \underline{\mathcal{U}}$ such that

$$\min_{s \in [0,\delta]} \psi_{\mathcal{S}_{\tilde{n}-1}}(x^{(\tilde{\jmath})}(s)) = \tilde{\psi};$$

consequently, by (17)

$$\psi_{\delta}^{(\tilde{\jmath})}(x_0) = \psi_{\mathcal{S}_{\tilde{n}-1}}(x^{(\tilde{\jmath})}(\delta)) \geq \tilde{\psi}.$$

Therefore, $u^{(\tilde{\jmath})} \in \mathcal{U}_{\text{ctrl}}^{\rightarrow}(x_0)$. □

Corollary 1. *For all $x \in \mathcal{S}_{ctrl}$, $\mathcal{U}_{ctrl}^{\nearrow}(x) \neq \emptyset$. For all $u^{(j)} \in \mathcal{U}_{ctrl}^{\nearrow}(x)$, $\psi_{\delta}^{(j)}(x) \geq \psi_{\mathcal{S}_{n(x)}}(x)$.*

6.3 Policies in $\mathcal{S}_1$

For $x_0 \in \mathcal{S}_1 = \mathcal{R}_1^-(\mathcal{S}_0)$, $x(\delta) \in \mathcal{S}_0$ for all $u^{(j)} \in \underline{\mathcal{U}}$. Furthermore, the computation of $\mathcal{R}_1^-(\mathcal{S}_0)$ does not suffer from the conservatism discussed in proposition 1, because the dynamics (2)–(3) are not time-dependent over a single sample interval. Consequently, one possible response is to sit back and wait for the inevitable failure. The corresponding most permissive control policy is $\mathcal{U}_{\text{inev}}^{\circ}(x) \triangleq \mathcal{U}$. However, if there is some conservatism in the model—for example, if disturbance inputs have been introduced into (2) and treated in a worst-case manner when computing $\mathcal{R}_1^-(\mathcal{S}_0)$—it may be possible to avoid $\mathcal{S}_0$. The corresponding policy

$$\mathcal{U}_{\text{inev}}^{\nearrow}(x) \triangleq \operatorname*{argmax}_{u^{(j)} \in \underline{\mathcal{U}}} \psi_{\mathcal{S}_0}(x^{(j)}(\delta)) \tag{21}$$

is defined in a manner similar to $\mathcal{U}_{\text{ctrl}}^{\nearrow}(x)$, although for $x \in \mathcal{S}_1$ there is no guarantee that application of this policy will avoid $\mathcal{S}_0$ for even a single sample interval.

6.4 Safety of the Policies

Proposition 3. *Let trajectory $x(\cdot)$ solve (2)–(3) with initial condition $x(0) = x_0$ and sampled feedback control policy*

$$u_{fb}(x) \in \begin{cases} \mathcal{U}_{ctrl}(x), & \text{for } x \in \mathcal{S}_{ctrl}; \\ \mathcal{U}, & \text{for } x \in \mathcal{S}_{free}. \end{cases} \tag{22}$$

If $x_0 \in \mathcal{S}_{free}$, then $x(t) \notin \mathcal{S}_0$ for all $t \in [0, (N+1)\delta]$, where $N\delta$ is the horizon used in the computation of $\mathcal{S}_{ctrl}$. If $x_0 \in \mathcal{S}_{ctrl}$, then $x(t) \notin \mathcal{S}_0$ for all $t \in [0, n(x_0)\delta]$.

Proof. Consider first $x_0 \in \mathcal{S}_{\text{ctrl}}$. Let $\tilde{n} = n(x_0)$ and $\tilde{\psi} = \psi_{\mathcal{S}_{\tilde{n}}}(x_0)$. By (16), $\tilde{\psi} > 0$, which implies that $\psi_{\mathcal{S}_{\tilde{n}-1}}(x(\delta)) > 0$ by (17) and either (18) and proposition 2 (if $\mathcal{U}_{\text{ctrl}} = \mathcal{U}^{\rightarrow}_{\text{ctrl}}$) or corollary 1 (if $\mathcal{U}_{\text{ctrl}} = \mathcal{U}^{\nearrow}_{\text{ctrl}}$). Therefore $x(\delta) \notin \mathcal{S}_{\tilde{n}-1}$. Use induction to show that $x(k\delta) \notin \mathcal{S}_{\tilde{n}-k}$ and hence that $x(t) \notin \mathcal{S}_0$ for $t \in [0, \tilde{n}\delta]$.

Now consider $x_0 \in \mathcal{S}_{\text{free}}$, which implies

$$\psi_{\mathcal{R}^+([0,\delta],\mathcal{S}_N)}(x_0) > 0. \tag{23}$$

The HJ PDE (6), (15) and (8) used to compute $\psi_{\mathcal{R}^+([0,\delta],\mathcal{S}_N)}$ implies that it is the value function for the finite horizon terminal value optimal control problem

$$\psi_{\mathcal{R}^+([0,\delta],\mathcal{S}_N)}(x_0) = \min_{u(\cdot)\in\mathfrak{U}} \min_{s\in[0,\delta]} \psi_{\mathcal{S}_N}(\tilde{x}(s)) \tag{24}$$

where $\tilde{x}(\cdot)$ solves (1) with initial condition $\tilde{x}(0) = x_0$ and $\mathfrak{U}$ is the set of all measurable input signals $u(\cdot)$ such that $u(s) \in \mathcal{U}$ for all $s \in [0, \delta]$. Note that $\mathfrak{U}$ is a much broader choice of input signals than piecewise constant, and it draws values from $\mathcal{U}$ not $\underline{\mathcal{U}}$, so the set of all possible trajectories $\tilde{x}(\cdot)$ contains all sampled data trajectories $x(\cdot)$. From (23) and (24) we conclude that $\psi_{\mathcal{S}_N}(x(s)) > 0$ for all $s \in [0, \delta]$ and hence that either $x(\delta) \in \mathcal{S}_{\text{free}}$ or $x(\delta) \in \mathcal{S}_{\text{ctrl}}$ with $n(x(\delta)) = N$. Using either induction in the former case or the proof above for $x_0 \in \mathcal{S}_{\text{ctrl}}$ in the latter case, it is easily shown that $x(t) \notin \mathcal{S}_0$ for all $t \in [0, (N+1)\delta]$ □

Corollary 2. *If the sampled data reachability calculation converged such that* $\mathcal{S}_{N-1} = \mathcal{S}_N = \mathcal{S}_\infty$ *and* $x_0 \in \mathcal{S}_{free} \cup (\mathcal{S}_{ctrl} \setminus \mathcal{S}_\infty)$*, then using the control policy (22) will ensure that* $x(t) \in \mathcal{S}_{free} \cup (\mathcal{S}_{ctrl} \setminus \mathcal{S}_\infty)$ *for all* $t > 0$.

7 Approximation and Implementation

In this section we describe a particular approach to approximating the solution of the equations above for the common case where we do not have analytic solutions to those equations.

7.1 Calculating the Sets

We use the Toolbox of Level Set Methods (TOOLBOXLS) as described in Mitchell and Templeton (2005) to manipulate implicit surface functions. Implicit surface functions are represented by values sampled at nodes on a regular orthogonal grid. When values are needed away from grid points, interpolation is used (eg: `interpn` in MATLAB). Maximum and minimum operations are done pointwise at each node in the grid.

The HJ PDE (6)–(8) used to determine $\mathcal{S}_k$ is purely convective because there are no inputs; consequently, it can be solved using an upwind finite difference scheme. High order of accuracy finite difference approximations of the spatial and temporal derivatives are used to evolve the equation (for example, see Osher and Fedkiw (2002)). However, because we

use the value of $\psi_{\mathcal{S}_k}$, and not just its zero level set, during construction of the control policies (via (17)), it is important that reinitialization and/or velocity extension not be applied when approximating the solution of these PDEs.

The HJ PDE (6), (15) and (8) used to determine $\mathcal{S}_{\text{ctrl}}$ involves an input, so a Lax-Friedrichs centered difference scheme is used. The same spatial and temporal finite difference approximations are used. Only the zero level set of $\psi_{\mathcal{S}_{\text{ctrl}}}$ is referenced, so it is possible to use reinitialization and/or velocity extension during this process; however, it is unlikely to be needed because the equations are solved only over δ time units.

7.2 Constructing the Feedback Controller

For a state $x_0 \in \mathcal{S}_{\text{free}} \cup \mathcal{S}_0$ the control policies are straightforward to implement. For $x_0 \in \mathcal{S}_{\text{ctrl}}$, we approximate (17) for each $j = 1, 2, \ldots, \ell$ by using an ODE solver (eg: `ode45` in MATLAB) to find the point $x^{(j)}(\delta)$ and then interpolate over the approximation of $\psi_{\mathcal{S}_{n(x_0)-1}}$ to determine $\psi_\delta^{(j)}(x_0)$. The set-valued policy is constructed from (18) or (19), where $\psi_{\mathcal{S}_{n(x_0)}}(x_0)$ might also need to be interpolated in (18).

7.3 Guaranteeing an Overapproximation

While the analytic formulation presented in sections 4–6 guarantees safety, the numerical implementation described above does not maintain those guarantees. The decision to use an unsound implementation was primarily driven by convenience, and also the empirical accuracy that the level set schemes have demonstrated in the past.

It is possible to reformulate the reachability calculations described above in viability theory and then use sound numerical implementations such as those described in Cardaliaguet et al. (1999). The sampled data minimal reachability calculation in section 4 is solved by a series of fixed input reachability calculations with switches at the sample times, but could also be solved by a series of fixed input viability kernels with switches at the sample times. The maximal reachability calculation in section 5 can be solved with a capture basin. It is less obvious how to synthesize safe controls from the indicator-like viability kernel representation, but there are several approaches to reformulate HJ PDEs as viability kernels if necessary. The primary shortcoming of these viability schemes is their relative inaccuracy when compared to the schemes implemented in TOOLBOXLS. It is possible that a combination of the two approaches might be able to achieve both sound and accurate approximations.

8 Example

Computations were done on an Intel Core2 Duo at 1.87 GHz with 4 GB RAM running 64-bit Windows 7 Professional (Service Pack 1), 64-bit MATLAB version 7.11 (R2010b),

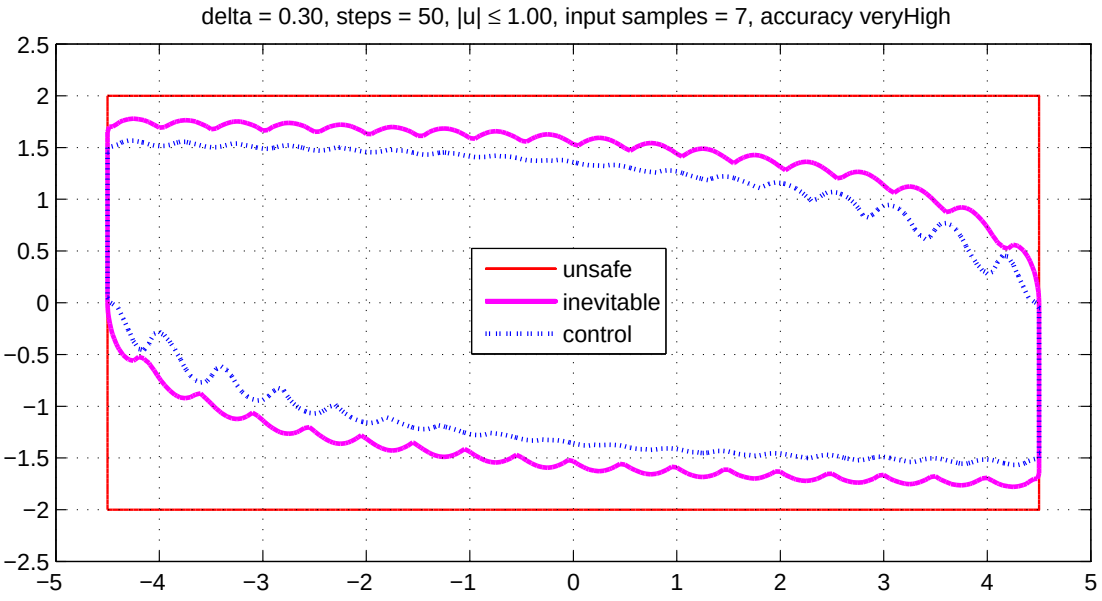

Figure 5: The partition of Ω for the spatially varying double integrator with $\delta = 0.3$ and horizon $N = 50$ (eg: $T = 15$, long enough for convergence). Because this is an envelope protection problem, the relationship of the sets is opposite that shown in figure 1: $\mathcal{S}_0$ is everything *outside* the outermost thin red rectangle, $\mathcal{S}_\infty$ is everything outside the thick magenta contour, $\mathcal{S}_{\text{ctrl}}$ is everything outside the dotted blue contour, and $\mathcal{S}_{\text{free}}$ is everything *inside* that innermost contour (where the legend is).

and ToolboxLS version 1.1. Matlab code can be found at the first author's web site `http://www.cs.ubc.ca/~mitchell`

We demonstrate the algorithms using an envelope protection problem for a variation on the double integrator because it is much easier to visualize results in two dimensions. In the standard double integrator, once deceleration begins the optimal control stays constant until the system stops no matter what the state; consequently, the results are very similar in a sampled data environment to what they would be in a continuous time environment. Instead, we modify the double integrator so that the optimal choice of input depends on state (a "spatially varying double integrator"). The dynamics are given by

$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \cos(2\pi x_1)u \end{bmatrix} = f(x,u)$$

with $|u| \leq 1$. Note that the effect of the input varies considerably over the domain, and the sign of the optimal input will switch every 0.5 units in the x_1 direction. The safe envelope is the interior of the rectangle $[-4.5, +4.5] \times [-2.0, +2.0]$, so $\mathcal{S}_0$ is everything *outside* this rectangle. For the sampled data problem, we choose $\delta = 0.3$ and $N = 50$ (which is empirically sufficient time for convergence). We choose sampled input set

$$\underline{\mathcal{U}} = \left\{-1, -\tfrac{2}{3}, -\tfrac{1}{3}, 0, +\tfrac{1}{3}, +\tfrac{2}{3}, +1\right\}.$$

Figure 5 shows the results for the above parameters. They were calculated on a grid of size 201×101 using a fifth order accurate spatial and a third order accurate temporal

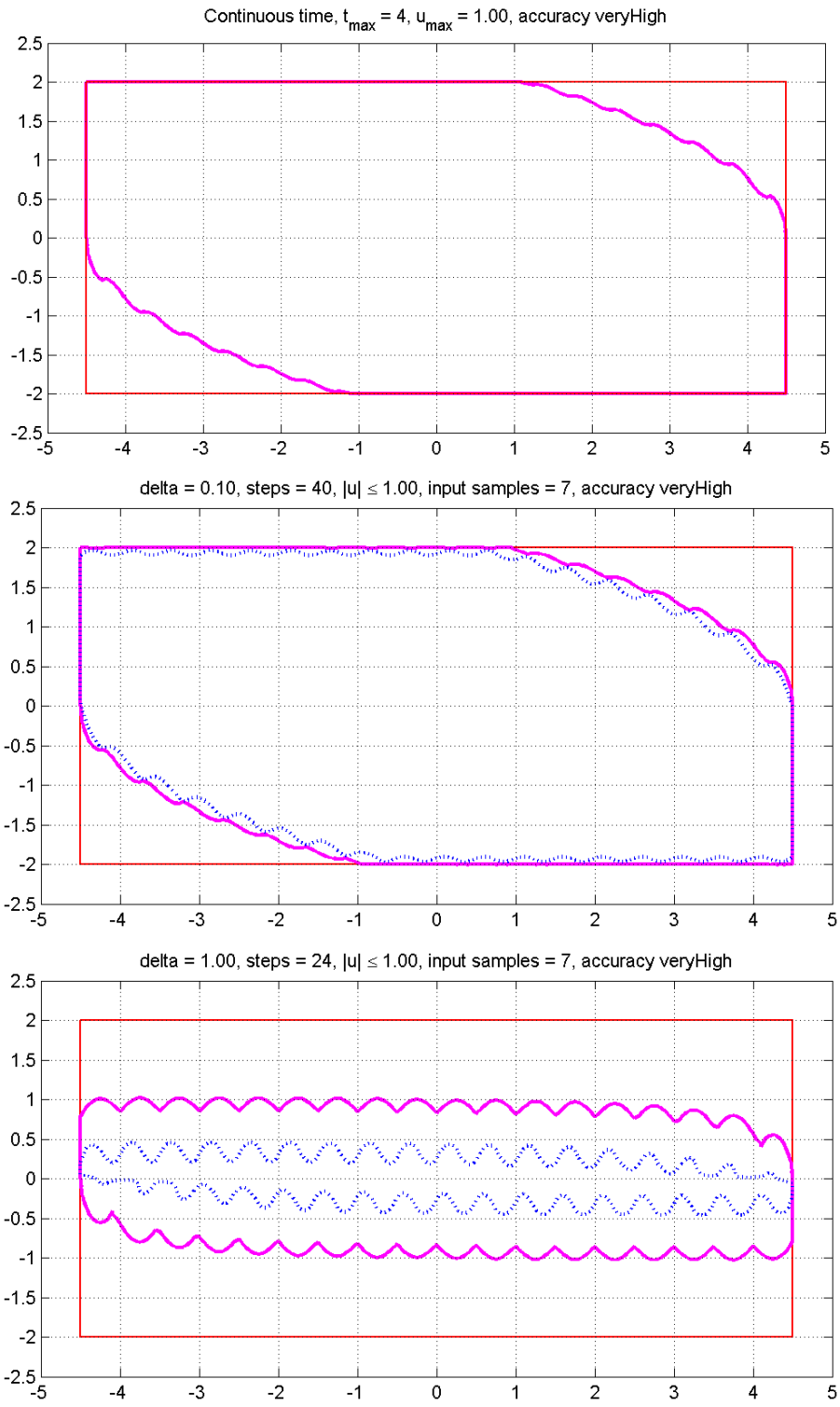

Figure 6: The effect of δ on the spatial partition. Top: Traditional reachability with continuous state feedback and measurable control signals ($T = 4$). Middle: Sampled data with $\delta = 0.1$, $N = 40$ ($T = 4$). Bottom: Sampled data with $\delta = 1.0$, $N = 24$ ($T = 24$).

derivative approximation. Figure 6 shows results for the continuous time version, and also for versions with $\delta = 0.1$ and $\delta = 1.0$. Notice that the continuous time version has a much larger $\mathcal{S}_{\text{free}}$ because it can always choose an input that generates deceleration. Furthermore, $\mathcal{S}_{\text{ctrl}} = \mathcal{S}_\infty$ in this case, because $\delta = 0$. In contrast, as δ becomes large the envelope becomes increasingly uncontrollable.

Figures 7 and 8 show some sample trajectories generated using the policy (22) with $\mathcal{U}^{\rightarrow}_{\text{ctrl}}$ and $\mathcal{U}^{\nearrow}_{\text{ctrl}}$ respectively. For illustrative purposes the control was chosen to drive the trajectory back toward $\mathcal{S}_{\text{ctrl}}$ for $x \in \mathcal{S}_{\text{free}}$, and was chosen for $\mathcal{U}^{\rightarrow}_{\text{ctrl}}$ to keep the trajectory as deeply within $\mathcal{S}_{\text{ctrl}}$ as possible, but other choices are available. In the bottom of each plot, notice that the value of $\psi_{\mathcal{S}_\infty}$ may decrease along a trajectory between samples, but if the trajectory is in $\mathcal{S}_{\text{ctrl}}$ (as indicated by the blue dots) at the sample time, then the value of $\psi_{\mathcal{S}_\infty}$ does not decrease at the subsequent sample time.

9 Conclusions and Future Work

We have adapted the sampled data reachability algorithm from Ding and Tomlin (2010) to a safety maintenance / viability problem, and demonstrated how the algorithm computes a conservative estimate of the sampled data backward reach tube. We have then demonstrated how to synthesize a permissive but safe control policy from this calculation, which may have applications in multi-objective or collaborative control problems. In the future we plan to apply this scheme to more complex nonlinear and hybrid systems with disturbance inputs, taking into account model, state, and discretization uncertainty.

References

Jean-Pierre Aubin, Alexandre M. Bayen, and Patrick Saint-Pierre. *Viability Theory: New Directions*. Systems & Control: Foundations & Applications. Springer, 2011. doi: 10.1007/978-3-642-16684-6.

Michael S. Branicky and Gang Zhang. Solving hybrid control problems: Level sets and behavioral programming. In *Proceedings of the American Control Conference*, pages 1175–1180, Chicago, IL, 2000.

M.S. Branicky, M.M. Curtiss, J. Levine, and S. Morgan. Sampling-based planning, control and verification of hybrid systems. *IEE Proceedings Control Theory and Applications*, 153(5):575 – 590, 2006.

P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre. Set-valued numerical analysis for optimal control and differential games. In M. Bardi, T. E. S. Raghavan, and T. Parthasarathy, editors, *Stochastic and Differential Games: Theory and Numerical Methods*, volume 4 of *Annals of International Society of Dynamic Games*, pages 177–247. Birkhäuser, 1999.

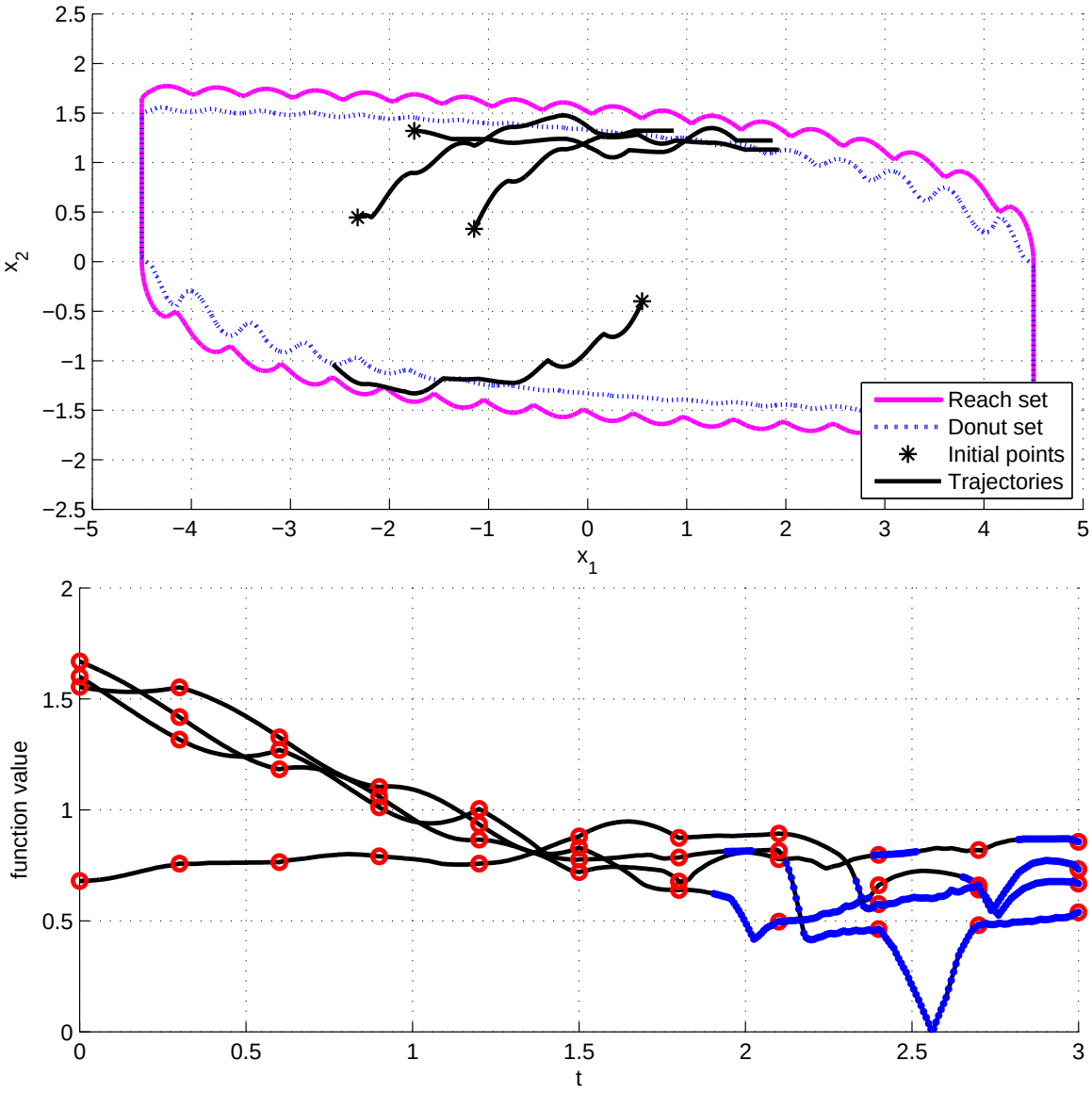

Figure 7: Sample trajectories using the permissive safe policy $\mathcal{U}^{\rightarrow}_{\text{ctrl}}$ for $\delta = 0.3$. Top: Trajectories $x(\cdot)$ in phase space overlaid on the state space partition. Bottom row: $\psi_{\mathcal{S}_\infty}(x(t))$ versus t. Sample times are shown as red circles, and periods during which $x(t) \in \mathcal{S}_{\text{ctrl}}$ are shown with blue dots.

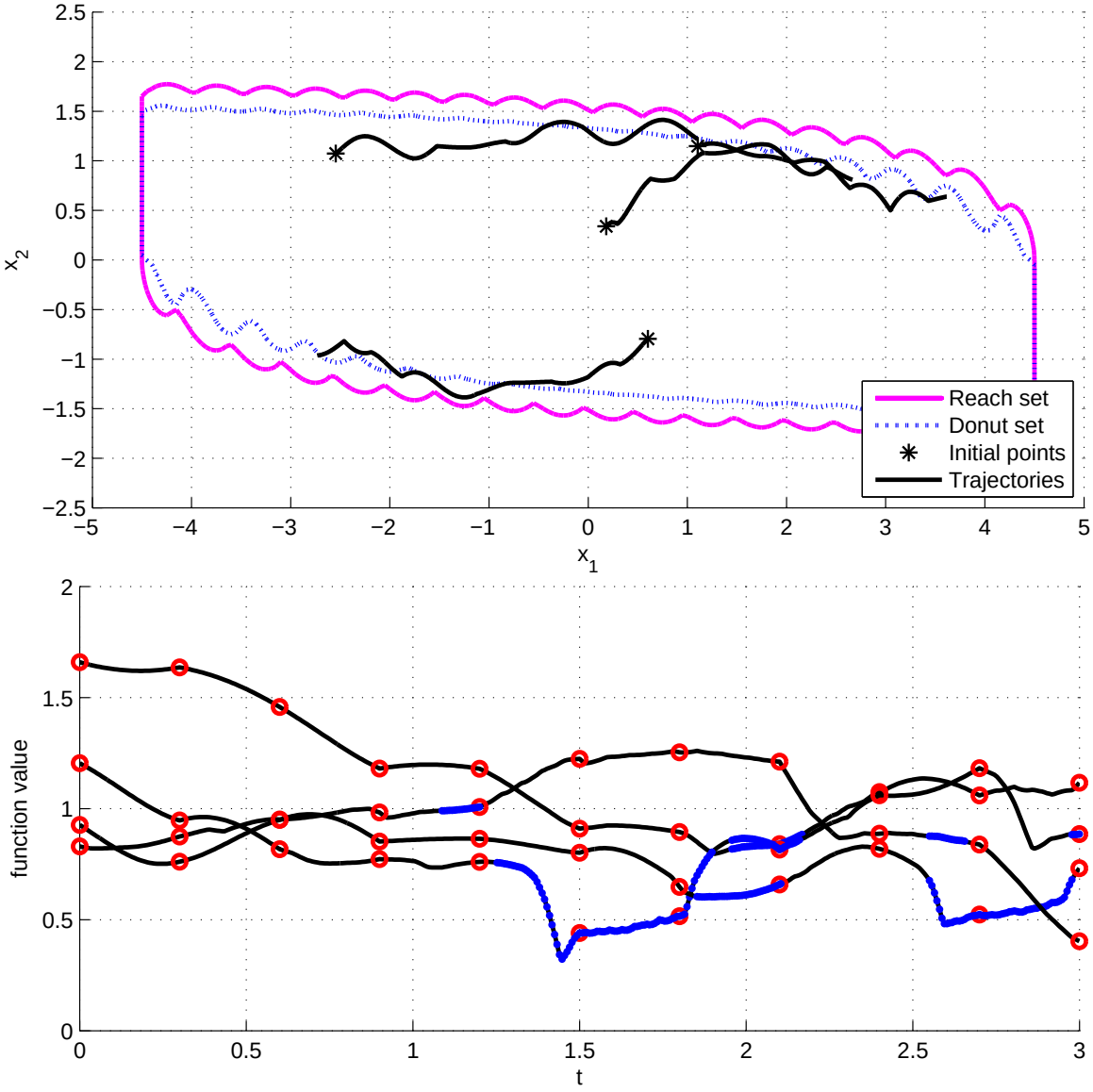

Figure 8: Sample trajectories using the aggressive safe policy $\mathcal{U}_{\text{ctrl}}^{\nearrow}$ for $\delta = 0.3$. Top: Trajectories $x(\cdot)$ in phase space overlaid on the state space partition. Bottom row: $\psi_{\mathcal{S}_\infty}(x(t))$ versus t. Sample times are shown as red circles, and periods during which $x(t) \in \mathcal{S}_{\text{ctrl}}$ are shown with blue dots.

Edmund M. Clarke. The birth of model checking. In Orna Grumberg and Helmut Veith, editors, *25 Years of Model Checking*, number 5000 in Lecture Notes in Computer Science, pages 1–26. Springer Verlag, 2008. doi: 10.1007/978-3-540-69850-0_1.

Jerry Ding and Claire J. Tomlin. Robust reach-avoid controller synthesis for switched nonlinear systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6481–6486, Atlanta, GA, 2010. doi: 10.1109/CDC.2010.5717115.

Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, New York, 2006.

John Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40(6): 917–927, 2004. doi: 10.1016/j.automatica.2004.01.012.

John Lygeros, Claire Tomlin, and Shankar Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999. doi: 10.1016/S0005-1098(98)00193-9.

Ian M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In Alberto Bemporad, Antonio Bicchi, and Giorgio Buttazzo, editors, *Hybrid Systems: Computation and Control*, number 4416 in Lecture Notes in Computer Science, pages 428–443. Springer Verlag, 2007. doi: 10.1007/978-3-540-71493-4_34.

Ian M. Mitchell and Jeremy A. Templeton. A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems. In Manfred Morari and Lothar Thiele, editors, *Hybrid Systems: Computation and Control*, number 3414 in Lecture Notes in Computer Science, pages 480–494. Springer Verlag, 2005. doi: 10.1007/978-3-540-31954-2_31.

Ian M. Mitchell, Alexandre M. Bayen, and Claire J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005. doi: 10.1109/TAC.2005.851439.

Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2002. doi: 10.1007/b98879.

Erion Plaku, Lydia Kavraki, and Moshe Vardi. Hybrid systems: from verification to falsification by combining motion planning and discrete search. *Formal Methods in System Design*, 34:157–182, 2009. doi: 10.1007/s10703-008-0058-5.

James A. Sethian and Alexander Vladimirsky. Ordered upwind methods for hybrid control. In C. J. Tomlin and M. R. Greenstreet, editors, *Hybrid Systems: Computation and Control*, number 2289 in Lecture Notes in Computer Science, pages 393–406. Springer Verlag, 2002.

Alexander Vladimirsky. Static PDEs for time-dependent control problems. *Interfaces and Free Boundaries*, 8(3):281–300, 2006. doi: 10.4171/IFB/144.