Controller Synthesis for Hybrid Systems: the Hamilton-Jacobi Approach

Claire J. Tomlin
Stanford University, Stanford CA 94305
tomlin@leland.stanford.edu

John Lygeros and **S. Shankar Sastry**
University of California, Berkeley CA 94720
lygeros,sastry@eecs.berkeley.edu

Abstract

We present a procedure for synthesizing controllers for safety specifications for hybrid systems. The procedure depends on the construction of the set of states of a continuous dynamical system that can be driven to a subset of the state space, avoiding another subset of the state space (the *Reach-Avoid* set). We present a characterization of the Reach-Avoid set in terms of the solution of a pair of coupled Hamilton-Jacobi partial differential equations. We also discuss a computational algorithm for solving such partial differential equations, and present an example derived from aircraft conflict resolution.

Introduction

The synthesis of controllers that meet safety specifications for discrete, continuous and hybrid systems has attracted considerable attention (see (Thomas 1995; Başar & Olsder 1995; Maler, Pnueli, & Sifakis 1995; Wong-Toi 1997) for an overview). Our work has been based on casting the problem as a two player, zero sum game, between a *controller*, that tries to ensure that the safety specification is satisfied and a *disturbance* (modeling the nondeterminism of the system), that tries to violate the safety specification (Lygeros, Tomlin, & Sastry 1999). In (Tomlin, Lygeros, & Sastry 1998) we proposed a procedure for systematically carrying out the controller synthesis for general hybrid systems. The procedure relies on the solution of partial differential equations (PDEs) (Lygeros, Tomlin, & Sastry 1998), known as the *Hamilton-Jacobi* equations. Here, we bring the synthesis procedure one step closer to implementation, by proposing a numerical scheme for solving these partial differential equations.

In this paper, we first briefly review the modeling formalism and the controller synthesis problem introduced in (Lygeros, Tomlin, & Sastry 1999). We then review the algorithm proposed in (Tomlin, Lygeros, & Sastry 1998) for solving the controller synthesis problem. The algorithm requires the computation of the set of states of a continuous dynamical system that can be driven to a given subset of the state space, avoiding another subset of the state space (the *Reach-Avoid* set). We present a procedure for characterizing the Reach-Avoid set, in terms of the solution to a pair of coupled Hamilton-Jacobi PDEs (Tomlin 1998). The advantage of this characterization (over the single PDE characterization of (Lygeros, Tomlin, & Sastry 1998), for example) is that it can deal with situations in which the closures of the Reach and Avoid sets overlap, without resorting to approximation. For the class of systems we consider, the situation is complicated by the fact that the initial data may be non-smooth, shocks (discontinuities in the solution to the PDE as time evolves) may develop along the solution, the right hand side of the PDE may be non-smooth due to the bang-bang nature of the optimal controls and disturbances, and the right hand side of the PDE may be discontinuous due to saturation effects introduced to guarantee the monotonicity of the Reach-Avoid set. While an analytical solution to the Hamilton Jacobi PDEs is likely to be impossible to obtain for most realistic examples, numerical solutions are possible. We present a procedure for numerically computing the Reach-Avoid set, based on the *level set method* of (Osher & Sethian 1988). The advantage of this method is that it can systematically deal with all the technical problems highlighted above, based on the viscosity solution concept for the PDEs. Finally, we demonstrate the application of this approach to an example from aircraft collision avoidance. The material in this paper is discussed in more detail in (Tomlin 1998; Tomlin, Lygeros, & Sastry 1999a; 1999b).

Model

For a finite collection V of variables, let $\mathbf{V}$ denote the set of valuations of these variables, i.e. the set of all possible assignments of the variables in V. For example, if x is a state variable taking values in $\mathbb{R}^n$ we write $X = \{x\}$ with $\mathbf{X} = \mathbb{R}^n$. By abuse of notation, we use lower case letters to denote both a variable and its valuation; the interpretation should be clear from the context. We call a variable discrete if its set of valuations is countable and continuous if it is a subset of Euclidean space. We assume the discrete topology for countable sets and the Euclidean metric topology for

subsets of Euclidean space. For a topological space X and a set $K \subseteq X$ we denote by K^c the complement, by $\overline{K}$ the closure, by K^o the interior, and by $\partial K = \overline{K} \backslash K^o$ the boundary of K in the topology of X. Given a set of valuations $W \subseteq \mathbf{V}$ and a subset of the variables $V' \subset V$ we denote by $W|_{V'} \subset \mathbf{V}'$ the restriction of W to V'.

Hybrid Automata

Definition 1 *A* **hybrid automaton**, H, *is a collection* (X, V, I, f, E, ϕ), *with:*

- **State and input variables:** X *and* V *are disjoint collections of state and input variables. We assume that* $X = X_D \cup X_C$ *and* $V = V_D \cup V_C$, *where* X_C *and* V_C *contain continuous, and* X_D *and* V_D *discrete variables. We refer to the valuations* $x \in \mathbf{X}$ *and* $v \in \mathbf{V}$ *as the state and the input of the hybrid automaton.*
- **Initial states:** $I \subset \mathbf{X}$ *is a set of initial valuations of the state variables.*
- **Continuous evolution:** $f : \mathbf{X} \times \mathbf{V} \to T\mathbf{X_C}$ *is a vector field (*$T\mathbf{X_C}$ *is called the* tangent space *of* $\mathbf{X_C}$*).*
- **Discrete transitions:** $E \subset \mathbf{X} \times \mathbf{V} \times \mathbf{X}$ *is a set of discrete transitions.*
- **Admissible inputs:** $\phi : \mathbf{X} \to 2^{\mathbf{V}}$ *gives the set of admissible inputs at a given state* $x \in \mathbf{X}$.

To fix notation we let $\mathbf{X_C} \subseteq \mathbb{R}^n$ and $\mathbf{V_C} \subseteq \mathbb{R}^m$. To ensure that the continuous evolution is well-posed we assume that f is Lipschitz continuous in x and continuous in v.

Definition 2 *A* **hybrid time trajectory**, τ, *is a finite or infinite sequence of intervals* $\tau = \{I_i\}$ *of the real line, starting with* I_0 *and satisfying:*

- I_i *is closed unless* τ *is a finite sequence and* I_i *is the last interval, in which case it is left closed but can be right open.*
- *Let* $I_i = [\tau_i, \tau_i']$. *Then for all* i, $\tau_i \leq \tau_i'$ *and for* $i > 0$, $\tau_i = \tau_{i-1}'$.

We denote by $\mathcal{T}$ the set of all hybrid time trajectories. For $t \in \mathbb{R}$ and $\tau \in \mathcal{T}$ we use $t \in \tau$ as a shorthand for "there exists a j such that $t \in [\tau_j, \tau_j'] \in \tau$".

Definition 3 *An* **execution** *of a hybrid automaton* H *is a collection* (τ, x, v) *with* $\tau \in \mathcal{T}$, $x : \tau \to \mathbf{X}$, *and* $v : \tau \to \mathbf{V}$ *which satisfies:*

- **Initial Condition:** $x(\tau_0) \in I$.
- **Discrete Evolution:** $(x(\tau_{i-1}'), v(\tau_{i-1}'), x(\tau_i)) \in E$, *for all* i.
- **Continuous Evolution:** *for all* i *with* $\tau_i < \tau_i'$, x *is continuous and* v *is piecewise continuous in* $[\tau_i, \tau_i']$ *and for all* $t \in [\tau_i, \tau_i')$, $(x(t), v(t), x(t)) \in E$. *Moreover, for all* $t \in [\tau_i, \tau_i']$ *where* v *is continuous* $\frac{d}{dt}(x(t)|_{X_C}) = f(x(t), v(t))$.
- **Input Constraints:** *for all* $t \in \tau$, $v(t) \in \phi(x(t))$.

We use χ to denote an execution of H and $\mathcal{H}$ to denote the set of all executions of H. We use $x^0 = x(\tau_0)$ to denote the initial state of an execution.

A *property*, P, of a hybrid automaton H is a map:

$$P : \mathcal{H} \to \{\text{True}, \text{False}\} \tag{1}$$

We say an execution $\chi \in \mathcal{H}$ satisfies property P if $P(\chi) = \text{True}$; we say a hybrid automaton satisfies a property P if $P(\chi) = \text{True}$ for all $\chi \in \mathcal{H}$. Given a set $F \subset \mathbf{X}$ we define a *safety property*, denoted by $\Box F$, by:

$$\Box F(\chi) = \begin{cases} \text{True} & \text{if } \forall t \in \tau, x(t) \in F \\ \text{False} & \text{otherwise} \end{cases}$$

Controller Synthesis

Assume that we are given a hybrid automaton H, which we refer to as the *plant*, and we are asked to control it using its input variables so that its executions satisfy certain properties. For the purposes of control the input variables of the plant are partitioned into two classes: *controls* and *disturbances*. We write $V = U \cup D$ where U and D are respectively control and disturbance variables. The interpretation is that the controls can be influenced using a *controller*, in an attempt to guide the system, whereas the disturbances are determined by the *environment* and may potentially disrupt the controller's plans.

An instance of the *controller synthesis problem* consists of a pair, (H, P), of a plant hybrid automaton and a property of that automaton. In this paper we restrict our attention to controller synthesis problems where $P = \Box F$. A *static state feedback controller* for H is a map:

$$g : \mathbf{X} \to 2^{\mathbf{U}} \tag{2}$$

Given a plant automaton H and a controller g for H one can define the set of closed loop executions as:

$$\mathcal{H}_g = \{(\tau, x, (u, d)) \in \mathcal{H} | \forall t \in \tau \;\; u(t) \in g(x(t))\} \tag{3}$$

It is easy to see that this is precisely the set of executions of another hybrid automaton, H_g. We say that controller g solves the synthesis problem $(H, \Box F)$ if H_g satisfies $\Box F$.

A subset $W \subseteq \mathbf{X}$ is *controlled invariant* if the controller synthesis problem $(H, \Box W)$ can be solved when $I = W$. It can be shown (Lygeros, Tomlin, & Sastry 1999) that the controller synthesis problem $(H, \Box F)$ can be solved if and only if there exists a unique maximal controlled invariant subset of F. In the next sections we highlight a procedure (introduced in (Tomlin, Lygeros, & Sastry 1998)) for computing this subset.

Controller Synthesis for Hybrid Systems

Construction of Controlled Invariant Sets

For the synthesis problem $(H, \Box F)$ we seek to construct the largest set of states for which the control u can

guarantee that the property $\Box F$ is satisfied, despite the action of the disturbance d. We first introduce some notation. For any $v = (u, d)$ define the set:

$$Inv(v) = \{x \in \mathbf{X} | v \in \phi(x) \text{ and } (x, v, x) \in E\} \quad (4)$$

For a state $x \in \mathbf{X}$ and input $v = (u, d)$ consider the sets:

$$Next(x, v) = \begin{cases} \{x' \in \mathbf{X} | (x, v, x') \in E\} & \text{if } v \in \phi(x) \\ \emptyset & \text{if } v \notin \phi(x) \end{cases} \quad (5)$$

$Inv(v)$ is the set of states from which continuous evolution is possible under input v, while $Next(x, v)$ is the set of states that can be reached from state x under input v through a discrete transition. Abusing notation slightly, for any set $K \subseteq \mathbf{X}$ and input $v = (u, d)$ we define the *successor of K under v* as the set:

$$Next(K, v) = \bigcup_{x \in K} Next(x, v) \quad (6)$$

For any set $K \subseteq \mathbf{X}$ we define the *controllable predecessor of K*, $Pre_u(K)$, and the *uncontrollable predecessor of K*, $Pre_d(K)$, by:

$$\begin{aligned} Pre_u(K) = & \{x \in \mathbf{X} | \exists u \in \mathbf{U}\ \forall d \in \mathbf{D}\ x \notin Inv(v) \\ & \text{and } Next(K, (u, d)) \subseteq K\} \cap K \\ Pre_d(K) = & \{x \in \mathbf{X} | \forall u \in \mathbf{U}\ \exists d \in \mathbf{D}\ Next(K, (u, d)) \\ & \cap K^c \neq \emptyset\} \cup K^c \end{aligned} \quad (7)$$

$Pre_u(K)$ contains all states in K for which u can force a transition back into K. $Pre_d(K)$, on the other hand, contains all states outside K, as well as all states from which a transition outside of K is possible whatever u does. The controllable and uncontrollable predecessors will be used in the discrete part of the algorithm for determining controlled invariant subsets. For the continuous part we introduce the *Reach-Avoid* operator:

Definition 4 (Reach-Avoid) *For two disjoint sets $B \subseteq \mathbf{X}$ and $G \subseteq \mathbf{X}$, define the Reach-Avoid operator as:*

$$\begin{aligned} Reach(B, G) = \{x^0 \in \mathbf{X} \mid & \forall u \in \mathcal{U}_f\ \exists d \in \mathcal{D} \text{ and } t \geq 0 \\ & \text{such that } x(t) \in B \text{ and } x(s) \notin G \text{ for all } s \in [0, t]\} \end{aligned} \quad (8)$$

Here $\mathcal{U}_f$ denotes the set of all $\mathbf{U}$-valued feedback strategies, $\mathcal{D}$ denotes the set of piecewise continuous functions from $\mathcal{T}$ to $\mathbf{D}$ and $x(\cdot)$ the (unique) continuous state trajectory starting at $x(0) = x^0$ under input (u, d).

The set $Reach(B, G)$ contains the states from which, for all $u(\cdot)$, there exists a $d(\cdot)$, such that the state trajectory can be driven to B while avoiding an "escape" set G.

Consider the following algorithm.

$$\begin{aligned} &\text{Let} \quad W^0 = F, W^{-1} = \emptyset, i = 0. \\ &\text{While} \quad W^i \neq W^{i-1} \text{ do} \\ &\qquad W^{i-1} = W^i \setminus Reach(Pre_d(W^i), Pre_u(W^i))) \\ &\qquad i = i - 1 \\ &\text{end} \end{aligned} \quad (9)$$

In the first step of this algorithm, we remove from F all states for which there is a disturbance $d(\cdot)$ which through continuous evolution can bring the system either outside F, or to states from which a transition outside F is possible, without first touching the set of states from which a transition keeping the system inside F can be forced. Since at each step $W^{i-1} \subseteq W^i$, the set W^i decreases monotonically as i decreases. If the algorithm terminates, we denote the fixed point by W^*. In this case, W^* can be shown to be the largest controlled invariant subset contained in F.

For two disjoint sets $B \subseteq \mathbf{X_C}$ and $G \subseteq \mathbf{X_C}$, let $l_B : \mathbf{X_C} \to \mathbb{R}$ and $l_G : \mathbf{X} \to \mathbb{R}$ be differentiable functions such that $B \triangleq \{x \in \mathbf{X_C} | l_B(x) \leq 0\}$ and $G \triangleq \{x \in \mathbf{X_C} | l_G(x) \leq 0\}$[1]. Consider the following system of coupled Hamilton-Jacobi equations:

$$-\frac{\partial J_B(x,t)}{\partial t} = \begin{cases} H_B^*(x, \frac{\partial J_B(x,t)}{\partial x}) \\ \quad \text{for } \{x \in X \mid J_B(x,t) > 0\} \\ \min\{0, H_B^*(x, \frac{\partial J_B(x,t)}{\partial x})\} \\ \quad \text{for } \{x \in X \mid J_B(x,t) \leq 0\} \end{cases} \quad (10)$$

and

$$-\frac{\partial J_G(x,t)}{\partial t} = \begin{cases} H_G^*(x, \frac{\partial J_G(x,t)}{\partial x}) \\ \quad \text{for } \{x \in X \mid J_G(x,t) > 0\} \\ \min\{0, H_G^*(x, \frac{\partial J_G(x,t)}{\partial x})\} \\ \quad \text{for } \{x \in X \mid J_G(x,t) \leq 0\} \end{cases} \quad (11)$$

where $J_B(x, 0) = l_B(x)$ and $J_G(x, 0) = l_G(x)$, and

$$H_B^*(x, \frac{\partial J_B}{\partial x}) = \begin{cases} 0 \text{ for } \{x \in X \mid J_G(x,t) \leq 0\} \\ \max_{u \in \mathbf{U}} \min_{d \in \mathbf{D}} \frac{\partial J_B}{\partial x} f(x, u, d) \\ \quad \text{otherwise} \end{cases} \quad (12)$$

$$H_G^*(x, \frac{\partial J_G}{\partial x}) = \begin{cases} 0 \text{ for } \{x \in X \mid J_B(x,t) \leq 0\} \\ \min_{u \in \mathbf{U}} \max_{d \in \mathbf{D}} \frac{\partial J_G}{\partial x} f(x, u, d) \\ \quad \text{otherwise} \end{cases} \quad (13)$$

Equation (10) describes the evolution of the set B under the Hamiltonian H_B^*. This is the solution to the "$\max_u \min_d$" game for reachability in purely continuous systems (see for example (Tomlin, Lygeros, & Sastry 1998)), with the modification that $H_B^* = 0$ in $\{x \in \mathbf{X_C} \mid J_G(x,t) \leq 0\}$. This ensures that the evolution of $J_B(x,t)$ is frozen once this set is reached. Similarly, equation (11) describes the evolution of the set G under the Hamiltonian H_G^*. Here a "$\min_u \max_d$" is used, since it is assumed that the control tries to push the system into G, to escape from B. $H_G^* = 0$ in $\{x \in \mathbf{X_C} \mid J_B(x,t) \leq 0\}$ to ensure that the evolution of $J_G(x,t)$ is frozen once this set is reached. Note that in both games, the disturbance is given the advantage by assuming that the control plays first. In the following sequence of Lemmas (see (Tomlin 1998) for a complete set of proofs) we show that the resulting set $\{x \in \mathbf{X_C} \mid J_B(x,t) < 0\}$ contains neither G nor states

[1] More generally, B and G may be expressed as the maximum of a set of differentiable functions.

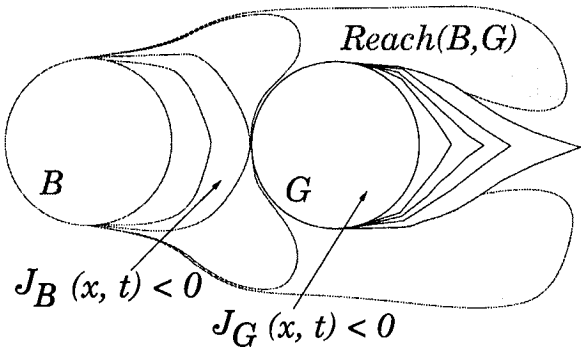

Figure 1: The computation of $Reach(B, G)$ in a single discrete state q.

for which there is a control which drives the system into G; and the set $\{x \in \mathbf{X_C} \mid J_G(x,t) < 0\}$ contains neither B nor states for which there is a disturbance which drives the system into B. We then prove that $\{x \in \mathbf{X_C} \mid J_B(x,t) < 0\}$ is the set $Reach(B,G)$. Figure 1 illustrates an example.

Assume that differentiable functions J_B and J_G satisfying the above partial differential equations exist. For all $t \leq 0$, let

$$B(t) \triangleq \{x \in \mathbf{X_C} \mid J_B(x,t) \leq 0\} \tag{14}$$

$$G(t) \triangleq \{x \in \mathbf{X_C} \mid J_G(x,t) \leq 0\} \tag{15}$$

Note that $B = B(0)$ and $G = G(0)$.

Lemma 1 *For all $t_2 \leq t_1 \leq 0$, $B(t_1) \subseteq B(t_2)$ and $G(t_1) \subseteq G(t_2)$.*

Lemma 2 *If $B^\circ(0) \cap G^\circ(0) = \emptyset$ then for all $t \leq 0$, $B^\circ(t) \cap G^\circ(t) = \emptyset$.*

Lemma 3 *For all $t \leq 0$, $B(t) \cap G(t) = \partial B(t) \cap \partial G(t)$. Moreover, for all $t' \leq t$, $B(t) \cap G(t) \subseteq \partial B(t') \cap \partial G(t')$.*

Theorem 1 (Characterization of Reach-Avoid) *Assume that $J_B(x,t)$ ($J_G(x,t)$ respectively) satisfies the Hamilton-Jacobi equation (10) ((11) respectively), and that it converges uniformly in x as $t \to -\infty$ to a function $J_B^*(x)$ ($J_G^*(x)$ respectively). Then,*

$$Reach(B,G) = \{x \in \mathbf{X_C} \mid J_B^*(x) < 0\} \tag{16}$$

Using the function J_B^* obtained once the algorithm has converged, a controller which renders W^* invariant can be constructed as:

$$\begin{array}{l} g(x) = \{ \\ \{u \in \phi(x)|_U \mid \forall d \in \phi(x)|_D \; Next(x,(u,d)) \subseteq W^*\} \\ \qquad \text{if } x \in (W^*)^o \\ \{u \in \phi(x)|_U \mid \forall d \in \phi(x)|_D (\frac{\partial J_B^*(x)}{\partial x} f(x,(u,d)) \geq 0 \wedge \\ x \in Inv(u,d)) \vee (Next(x,(u,d)) \subseteq W^* \wedge \\ x \notin Inv(u,d))\} \qquad \text{if } x \in \partial W^* \\ \phi(x)|_U \qquad \text{if } x \in (W^*)^c\} \end{array} \tag{17}$$

Here $\wedge$ stands for the logical AND and $\vee$ for the logical OR.

In general, one cannot expect to solve for W^* using a finite computation. The class of hybrid systems for which algorithms like the one presented here are guaranteed to terminate is known to be restricted (Henzinger *et al.* 1995). Techniques have been proposed to resolve this problem, making use of approximation schemes to obtain estimates of the solution (some are discussed in the next section). In practice, we are helped by the fact that we are usually interested in finite time computations, rather than computing for $t \to -\infty$ or until a fixed point is reached. Another problem is the requirement that the controller resulting from our algorithm be *non-Zeno* (does not enforce the safety requirement by preventing time from diverging). The algorithm proposed here has no way of preventing such behavior. A practical method of resolving the Zeno problem is adding a requirement that the amount of time the system remains in each discrete state is bounded below by a positive number (representing, for example, the clock period of a digital computer).

Computation using Level Set Methods

One of the key challenges in hybrid systems research is the efficient numerical computation of the backwards reachable set. Hamilton-Jacobi equations are difficult to solve numerically, due to the occurrence of *shocks* or discontinuities that occur in the solution as time evolves. We are currently exploring some *approximation techniques* to derive an efficient numerical solution: here we present one of these techniques. The level set methods of Osher and Sethian (Osher & Sethian 1988) are a set of computation schemes for propagating interfaces in which the speed of propagation is governed by a partial differential equation. These numerical techniques compute the *viscosity solution* to the partial differential equation, which is the solution ensuring that shocks are preserved. In order for the numerical scheme to closely approximate the gradient $\frac{\partial J^*(x,t)}{\partial x}$, especially at points of discontinuity, an appropriate approximation to the spatial derivative must be used. Consider an example in two dimensions, with X discretized into a grid with spacing Δx_1 and Δx_2. The *forward difference operator* D^{+x_i} at $x = (x_1, x_2)$ is defined as (for x_1, similarly for x_2):

$$D^{+x_1} J^*(x,t) = \frac{J^*((x_1 + \Delta x_1, x_2), t) - J^*(x,t)}{\Delta x_1} \tag{18}$$

The *backward difference operator* D^{-x_i} is defined as (for x_1, similarly for x_2):

$$D^{-x_1} J^*(x,t) = \frac{J^*(x,t) - J^*((x_1 - \Delta x_1, x_2), t)}{\Delta x_1} \tag{19}$$

Similarly, the *central difference operator* D^{0x_i} is defined as (for x_1, similarly for x_2):

$$D^{0x_1} J^*(x,t) = \frac{D^{+x_1} J^*(x,t) + D^{-x_1} J^*(x,t)}{2} \tag{20}$$

At each grid point in x, the partial derivatives $\frac{\partial}{\partial x_1}$ and $\frac{\partial}{\partial x_2}$ may be approximated to first order using either the forward, backward, or central difference operators. The correct choice of operator depends on the direction of $f(x, u^*, d^*)$ (in our case it depends on $-f(x, u^*, d^*)$ since we compute backwards in time). If $-f(x, u^*, d^*)$ flows from left to right (from smaller to larger values of x_1), then then D^{-x_1} should be used to approximate $\frac{\partial J(x,t)}{\partial x_1}$; and if $-f(x, u^*, d^*)$ flows from bottom to top (from smaller to larger values of x_2), then then D^{-x_2} should be used to approximate $\frac{\partial}{\partial x_2}$ (and vice versa). Such an approximation is called an *upwind* scheme, since it uses information upwind of the direction that information propagates.

The algorithm for the two dimensional example proceeds as follows. Choose a domain of interest in $\mathbf{X_C}$ and discretize the domain with a grid of spacing $\Delta x_1, \Delta x_2$. Let x_{ij} represent the grid point $(i\Delta x_1, j\Delta x_2)$ and let $\tilde{J}(x_{ij}, t)$ represent the numerical approximation of $J(x_{ij}, t)$. Set $t = 0$ and compute the initial condition $\tilde{J}(x_{ij}, 0) = l(x_{ij})$. While for some x_{ij}, $\tilde{J}(x_{ij}, t) \neq \tilde{J}(x_{ij}, t - \Delta t)$ perform the following steps:

1. Compute $u^*(x_{ij}, D^{0x_1}\tilde{J}(x_{ij}, t), D^{0x_2}\tilde{J}(x_{ij}, t))$ and $d^*(x_{ij}, D^{0x_1}\tilde{J}(x_{ij}, t), D^{0x_2}\tilde{J}(x_{ij}, t))$.
2. Calculate $f(x_{ij}, u^*, d^*)$
3. If $(-f(x_{ij}, u^*, d^*))$ flows from greater to lesser values of x_1, let $\frac{\partial}{\partial x_1} = D^{+x_1}$, otherwise let $\frac{\partial}{\partial x_1} = D^{-x_1}$.
4. If $(-f(x_{ij}, u^*, d^*))$ flows from greater to lesser values of x_2, let $\frac{\partial}{\partial x_2} = D^{+x_2}$, otherwise let $\frac{\partial}{\partial x_2} = D^{-x_2}$.
5. Compute $\tilde{J}(x_{ij}, t - \Delta t)$:
   For x_{ij} such that $\tilde{J}(x_{ij}, t) > 0$,

$$\tilde{J}(x_{ij}, t-\Delta t) = \tilde{J}(x_{ij}, t)+\Delta t\frac{\partial \tilde{J}(x_{ij}, t)}{\partial x} f(x_{ij}, u^*, d^*) \tag{21}$$

   For x_{ij} such that $\tilde{J}(x_{ij}, t) \leq 0$,

$$\tilde{J}(x_{ij}, t-\Delta t) = \begin{cases} \tilde{J}(x_{ij}, t) + \Delta t\frac{\partial \tilde{J}(x_{ij}, t)}{\partial x} f(x_{ij}, u^*, d^*) \\ \qquad \text{if } \frac{\partial \tilde{J}(x_{ij}, t)}{\partial x} f(x_{ij}, u^*, d^*) < 0 \\ \tilde{J}(x_{ij}, t) \text{ otherwise} \end{cases} \tag{22}$$

Two-Aircraft Conflict Resolution

Consider two aircraft flying in a collision course on the same horizontal plane (Figure 2). To avoid the collision the aircraft go through a coordinated avoidance maneuver: when they come within a certain distance of each other, they both start to turn to the right, following a trajectory which is a sequence of arcs of circles of fixed radii, and straight lines (*trimmed flight* segments). We assume that aircraft 1 initiates the avoidance maneuver and that the aircraft communicate and switch modes simultaneously. We also assume that the angles of the avoid maneuver are fixed,

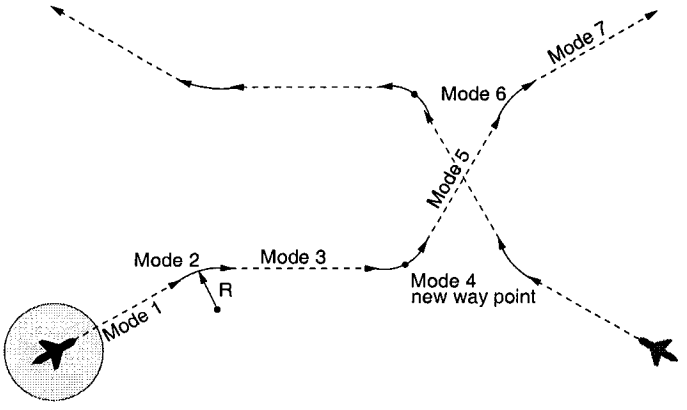

Figure 2: Two aircraft in seven modes of operation: in modes 1, 3, 5, and 7 the aircraft follow a straight course and in modes 2, 4, and 6 the aircraft follow arcs of circles. The initial relative heading is preserved throughout.

so that the straight path of mode 3 is at a $-45°$ angle to the straight path of mode 1, and that of mode 5 is at a $45°$ to that of mode 1. Also, the length of each arc is fixed at a pre-specified value, and the lengths of the segments in modes 3 and 5 are equal to each other, but unspecified. Given some uncertainty in the actions of the aircraft, we would like to generate the relative distance between aircraft at which the aircraft may switch safely from mode 1 to mode 2, and the minimum lengths of the segments in modes 3 and 5, to ensure that a 5 nautical mile separation is maintained.

The two aircraft system can be modeled by a hybrid automaton with seven discrete states ($\mathbf{X_D} = \{straight1, arc1, straight2, arc2, straight3, arc3, straight4\}$) and four continuous states, the relative position, (x_r, y_r), and heading, ψ_r, of the two aircraft, and a clock variable, z, to keep track of how long the aircraft have stayed in each mode. Overall, $\mathbf{X_C} = \mathbb{R}^2 \times [0, 2\pi] \times \mathbb{R}$. A discrete control input $\sigma \in \mathbf{U_D} = \{0, 1\}$ can be used to initiate the maneuver. There is also a continuous control input, the airspeed of aircraft 1, $v_1 \in \mathbf{U_C} = [\underline{v_1}, \overline{v_1}]$ and a continuous disturbance input, the airspeed of aircraft 2, $v_2 \in \mathbf{D} = [\underline{v_2}, \overline{v_2}]$. The speed of aircraft 2 is treated as a disturbance because we assume that aircraft 1 can estimate it only approximately. The unsafe set G is given by:

$$G = \mathbf{X_D} \times \{(x_r, y_r, \psi_r, z) \in \mathbf{X_C} | x_r^2 + y_r^2 \leq 5^2\} \tag{23}$$

To simplify the calculation we assume that the speed of both aircraft remains constant during the circular parts of the maneuver, but can take on any allowable value in the straight parts. In other words, $\phi(x) = \mathbf{U_D} \times \{(\hat{v}_1, \hat{v}_2)\}$ if $x|_{X_D} \in \{arc1, arc2, arc3\}$ and $\phi(x) = \mathbf{U_D} \times \mathbf{U_C} \times \mathbf{D}$ otherwise. Our goal is to compute the relative distance at which the maneuver must start, the length of the straight legs *straight2*

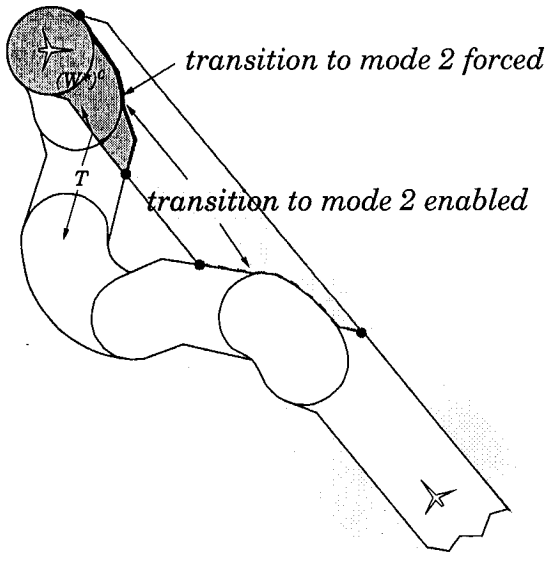

Figure 3: $(W^*)^c = (W^{-7})^c$ in *straight*1.

and *straight*3, as well as the airspeed v_1^* along those legs, to ensure safety. The details of the example are presented in (Tomlin 1998): Figure 3 displays the fixed point $W^* = W^{-7}$ for the initial mode *straight*1. The controller that switches between the modes is also illustrated in Figure 3. The time spent in the straight legs of the maneuver T, may be chosen to maximize W^*.

Concluding Remarks

A common concern for all approximations for the computation of the set of reachable states is that, for safety properties, one typically would like a conservative overapproximation. This is not easy to satisfy with the level set method, however. One needs to keep accurate bounds on the numerical errors and grow the final estimate of the reach set appropriately. An additional issue one needs to consider in the context of controller synthesis is the controlled invariance of this conservative approximation. This issue has not to our knowledge been addressed by any of the methods proposed in the literature (since they are primarily concerned with verification). One would like to ensure that the numerical estimate of the reach set (for the case of the level set method, this could be some interpolation between the collection of discrete points produced by the algorithm) is controlled invariant. If this is indeed the case, one would also like to obtain a controller that renders the approximation invariant.

Acknowledgements

C. Tomlin's research supported by a Terman Faculty Fellowship; J. Lygeros and S. Sastry's research supported by ONR under grant N00014-97-1-0946, and by DARPA under contract F33615-98-C-3614.

References

Başar, T., and Olsder, G. J. 1995. *Dynamic Noncooperative Game Theory.* Academic Press, second edition.

Henzinger, T. A.; Kopke, P. W.; Puri, A.; and Varaiya, P. 1995. What's decidable about hybrid automata. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing.*

Lygeros, J.; Tomlin, C.; and Sastry, S. 1998. On controller synthesis for nonlinear hybrid systems. In *Proceedings of the IEEE Conference on Decision and Control*, 2101–2106.

Lygeros, J.; Tomlin, C.; and Sastry, S. 1999. Controllers for reachability specifications for hybrid systems. *Automatica* 35(3). To appear.

Maler, O.; Pnueli, A.; and Sifakis, J. 1995. On the synthesis of discrete controllers for timed systems. In Mayr, E. W., and Puech, C., eds., *STACS 95: Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science 900. Munich: Springer Verlag. 229–242.

Osher, S., and Sethian, J. A. 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics* 79:12–49.

Thomas, W. 1995. On the synthesis of strategies in infinite games. In Mayr, E. W., and Puech, C., eds., *Proceedings of STACS 95, Volume 900 of LNCS.* Munich: Springer Verlag. 1–13.

Tomlin, C.; Lygeros, J.; and Sastry, S. 1998. Synthesizing controllers for nonlinear hybrid systems. In Henzinger, T., and Sastry, S., eds., *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science 1386. New York: Springer Verlag. 360–373.

Tomlin, C.; Lygeros, J.; and Sastry, S. 1999a. Computation of controllers for nonlinear hybrid systems. In *Proceedings of the International Federation of Automatic Control.* To appear.

Tomlin, C.; Lygeros, J.; and Sastry, S. 1999b. Computing controllers for nonlinear hybrid systems. In Vaandrager, F., and van Schuppen, J., eds., *Hybrid Systems: Computation and Control*, LNCS. Springer Verlag. To appear.

Tomlin, C. J. 1998. *Hybrid Control of Air Traffic Management Systems.* Ph.D. Dissertation, Department of Electrical Engineering, University of California, Berkeley.

Wong-Toi, H. 1997. The synthesis of controllers for linear hybrid automata. In *Proceedings of the IEEE Conference on Decision and Control.*