Level Set Methods for Computation in Hybrid Systems*

Ian Mitchell[1] and Claire J. Tomlin[2]

[1] mitchell@sccm.stanford.edu, Scientific Computing and Computational Mathematics Program, Gates 2B, Stanford University, Stanford CA, 94305.
[2] tomlin@leland.stanford.edu, Department of Aeronautics and Astronautics 250 Durand, Stanford University, Stanford CA, 94305.

Abstract. Reachability analysis is frequently used to study the safety of control systems. We present an implementation of an exact reachability operator for nonlinear hybrid systems. After a brief review of a previously presented algorithm for determining reachable sets and synthesizing control laws—upon whose theory the new implementation rests—an equivalent formulation is developed of the key equations governing the continuous state reachability. The new formulation is implemented using level set methods, and its effectiveness is shown by the numerical solution of three examples.

1 Introduction

The reachability operator, a function or algorithm that can determine the evolution of sets of trajectories, is key in the synthesis and verification of controllers for continuous, discrete or hybrid systems. Regardless of whether reachability appears implicitly, such as in the generation of invariant sets, or explicitly, no technique for determining safe control systems can avoid its use. It is natural that methods for its accurate, automatic computation are attracting considerable attention.

Reachability analysis of hybrid systems has been investigated by both the computer science and control communities. Methods have been developed by computer scientists for computing reachable sets for timed automata [1] and linear hybrid automata [2], for which computation is based on the propagation of polygonal sets under constant rate dynamics. Tools have been developed to perform such calculations automatically [3, 4], and to synthesize controllers in such a framework [5, 6]. Control theorists have extended reachability tools from continuous state and time dynamical systems theory to incorporate discrete switches [7–11]. However, the efficient computation of reachable sets for hybrid systems with nonlinear dynamics remains a difficult problem to solve. Numerical techniques which over-approximate the nonlinear dynamics with linear dynamics [12], or which over-approximate the reachable sets [13–16], have recently been developed.

* Research supported by DARPA under the Software Enabled Control Program (AFRL contract F33615-99-C-3014), and by a Frederick E. Terman Faculty Award.

In this paper, we present an implementation of an exact reachability operator for nonlinear hybrid systems. An algorithm which synthesizes control laws for such systems based on the Hamilton-Jacobi equation [9–11] is reviewed, and then a new Hamilton-Jacobi formulation with superior numerical properties is developed and proved to be equivalent. While level set techniques were previously investigated for the solution of such equations in [17], we have added several improvements to the basic level set algorithm. Examples from [11] demonstrate the results of applying the new algorithm to the new equations—examples which have never previously been solved computationally.

2 Deriving Reachable Sets in Hybrid Automata

In [11], an algorithm is presented which characterizes the reachable set of a nonlinear hybrid automaton (with desired safety properties) as that whose boundary is the zero level set of a particular Hamilton-Jacobi equation. The algorithm also computes the continuous and discrete control laws to maximize the safe operating region. In this section, we briefly review this hybrid system model and reachability algorithm, and then present a second characterization using a similar Hamilton-Jacobi algorithm with better numerical properties.

2.1 Hybrid Automata and Hamilton-Jacobi Equations

A **hybrid automaton** is defined as

$$H = ((Q \times X), (U \times D), (\Sigma_u \times \Sigma_d), f, \delta, Inv, \Omega) \tag{1}$$

where Q is a finite set of discrete states, $X = \mathrm{R}^n$, $U \subseteq \mathrm{R}^{n_u}$ is the set of continuous control inputs, $D \subseteq \mathrm{R}^{n_d}$ is the set of continuous disturbances, $\Sigma = \Sigma_u \times \Sigma_d$ is a finite set of actions, where Σ_u denotes the set of discrete control inputs, and Σ_d the set of discrete disturbance inputs, $f : Q \times X \times U \times D \to \mathrm{R}^n$ defines the flow of continuous trajectories, $\delta : Q \times X \times \Sigma_u \times \Sigma_d \to 2^{Q \times X}$ is the discrete transition function, $Inv \subseteq Q \times X$ is the invariant associated to each discrete state, and Ω is an acceptance condition—here $\Omega = (\Box F)$, meaning that the state of the system must remain within a set $F \subseteq Q \times X$. We denote $\mathcal{U}$ as the set of piecewise continuous functions from R to U, and $\mathcal{D}$ the set of piecewise continuous functions from R to D.

Three operators are defined:

$$\begin{aligned}
Pre_u(K) &= \{(q,x) \in Q \times X | \exists \sigma_u \in \Sigma_u \ \forall \sigma_d \in \Sigma_d \ \delta(q,x,\sigma_u,\sigma_d) \subseteq K\} \cap K \\
Pre_d(K) &= \{(q,x) \in Q \times X | \forall \sigma_u \in \Sigma_u \ \exists \sigma_d \in \Sigma_d \ \delta(q,x,\sigma_u,\sigma_d) \cap K^c \neq \emptyset\} \cup K^c \\
Reach(G,E) &= \{(q,x) \in Q \times X \mid \forall u \in \mathcal{U} \ \exists d \in \mathcal{D} \text{ and } t \geq 0 \text{ such that} \\
&\qquad (q(t),x(t)) \in G \text{ and } (q(s),x(s)) \in Inv \setminus E \text{ for } s \in [0,t]\}
\end{aligned}$$

where $K \subseteq Q \times X$; $G, E \subseteq X$; and $(q(s), x(s))$ is the continuous state trajectory of $\dot{x} = f(q(s), x(s), u(s), d(s))$ starting at (q, x). The set $Reach(G, E)$ describes those states from which, for all $u(\cdot) \in \mathcal{U}$, there exists a $d(\cdot) \in \mathcal{D}$, such that the

state trajectory $(q(s), x(s))$ can be driven to a "bad" set G while avoiding an "escape" set E. With these definitions in place, the algorithm for reachability analysis for hybrid systems proceeds as follows [10, 11]:

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

If the algorithm terminates after a finite number of steps, then the fixed point W^* is the largest set of states for which the control $(u(\cdot), \sigma_u[\cdot])$ can guarantee that the state of the hybrid system remains inside F despite the action of the disturbance $(d(\cdot), \sigma_d[\cdot])$. In order to implement this algorithm, Pre_u, Pre_d, and $Reach$ need to be computed. The calculation of Pre_u and Pre_d requires inversion of the transition relation δ subject to the quantifiers $\exists$ and $\forall$. The computation of $Reach$ requires an algorithm for determining the set of initial conditions from which trajectories can reach one set, avoiding a second set along the way. Our focus in this paper is on numeric computation of the latter operator.

Let $l_G : X \to \mathrm{R}$ and $l_E : X \to \mathrm{R}$ be differentiable functions such that $G \triangleq \{x \in X | l_G(x) \leq 0\}$ and $E \triangleq \{x \in X | l_E(x) \leq 0\}$. Consider the following system of interconnected Hamilton-Jacobi equations [11, 17]:

$$-\frac{\partial J_G^*(x,t)}{\partial t} = \begin{cases} H_G^*(x, \frac{\partial J_G^*(x,t)}{\partial x}), & \text{for } \{x \in X \mid J_G^*(x,t) > 0\}, \\ \min\{0, H_G^*(x, \frac{\partial J_G^*(x,t)}{\partial x})\}, & \text{for } \{x \in X \mid J_G^*(x,t) \leq 0\} \end{cases} \tag{2}$$

$$-\frac{\partial J_E^*(x,t)}{\partial t} = \begin{cases} H_E^*(x, \frac{\partial J_E^*(x,t)}{\partial x}), & \text{for } \{x \in X \mid J_E^*(x,t) > 0\}, \\ \min\{0, H_E^*(x, \frac{\partial J_E^*(x,t)}{\partial x})\}, & \text{for } \{x \in X \mid J_E^*(x,t) \leq 0\} \end{cases} \tag{3}$$

where $J_G^*(x, u(\cdot), d(\cdot), 0) = l_G(x)$ and $J_E^*(x, u(\cdot), d(\cdot), 0) = l_E(x)$, and

$$H_G^*(x, \frac{\partial J_G^*}{\partial x}) = \begin{cases} 0, & \text{for } \{x \in X \mid J_E^*(x,t) \leq 0\} \\ \max_{u \in U} \min_{d \in D} \frac{\partial J_G^*}{\partial x} f(x,u,d), & \text{otherwise} \end{cases} \tag{4}$$

$$H_E^*(x, \frac{\partial J_E^*}{\partial x}) = \begin{cases} 0, & \text{for } \{x \in X \mid J_G^*(x,t) \leq 0\} \\ \min_{u \in U} \max_{d \in D} \frac{\partial J_E^*}{\partial x} f(x,u,d), & \text{otherwise} \end{cases} \tag{5}$$

Theorem 1 (Characterization of Reach-Avoid [11]) *Assume that $J_G^*(x,t)$ ($J_E^*(x,t)$ respectively) satisfies the Hamilton-Jacobi equation (2) ((3) respectively), and that it converges uniformly in x as $t \to -\infty$ to a function $J_G^*(x)$ ($J_E^*(x)$ respectively). Then,*

$$Reach(G, E) = \{x \in X \mid J_G^*(x) < 0\} \tag{6}$$

Proof. Please see [11]. □

By our convention, we assume that the unsafe sets, defined as G° and its backwards reachable set under (2)-(5), are open; and safe sets, defined as E and its backwards reachable set, are closed.

2.2 An Equivalent Hamilton-Jacobi Formulation

Although the *Reach* operator can be computed by solving the equations (2)–(5), in practice the discontinuous right hand sides of the equations introduce serious numerical instabilities into the computation. Consider instead the standard form of the Hamilton-Jacobi equation:

$$-\frac{\partial J_G(x,t)}{\partial t} = H_G(x, \frac{\partial J_G(x,t)}{\partial x}) = \max_{u\in U}\min_{d\in D} \frac{\partial J_G}{\partial x} f(x,u,d), \tag{7}$$

$$-\frac{\partial J_E(x,t)}{\partial t} = H_E(x, \frac{\partial J_E(x,t)}{\partial x}) = \min_{u\in U}\max_{d\in D} \frac{\partial J_E}{\partial x} f(x,u,d), \tag{8}$$

with the same initial conditions as those used for J_G^* and J_E^*: $J_G(x,0) = l_G(x)$ and $J_E(x,0) = l_E(x)$. Now let:

$$J_G^{\min}(x,t) = \min_{\tau\in[t,0]} J_G(x,\tau), \tag{9}$$

$$J_E^{\min}(x,t) = \min_{\tau\in[t,0]} J_E(x,\tau), \tag{10}$$

$$J_G(x,t) \geq -J_E^{\min}(x,t), \tag{11}$$

$$J_E(x,t) \geq -J_G^{\min}(x,t). \tag{12}$$

Constraints (9) and (10) replace the "min" on the right hand side of equations (2) and (3), thus ensuring that sublevel sets of $J_G^{\min}(x,t)$ and $J_E^{\min}(x,t)$ do not shrink as time flows backwards; constraints (11) and (12) replace the "freezing" of the Hamiltonian on the right hand sides of equations (4) and (5) and ensure that the interiors of the two sets do not overlap, since for a given $x \in X$, if $J_E^{\min}(x,t) < 0$, then (11) will force $J_G(x,t) \geq 0$; conversely, if $J_G^{\min}(x,t) < 0$ then $J_E(x,t) \geq 0$.

Lemma 1 (Equivalence of Solutions) *The solution $J_G^*(x,t)$ to (2)–(5), and the solution $J_G^{\min}(x,t)$ to (7)–(12), are equivalent in that, for any $x \in X$, they satisfy one of*

$$J_G^*(x,t) \leq 0 \text{ if and only if } J_G^{\min}(x,t) \leq 0 \tag{13}$$

$$J_G^*(x,t) < 0 \text{ if and only if } J_G^{\min}(x,t) < 0 \tag{14}$$

for all $t \leq 0$.

Proof. We choose a particular $x \in X$ and assume that the computation starts at final time $t = 0$ and works backwards into negative time. Also, assume that the interiors of the initial sets do not intersect: $G^\circ \cap E^\circ = \emptyset$.

Case 1 (x is in G at $t = 0$). Thus $l_G(x) \leq 0$, which implies from (2) $\forall t < 0$ that $J_G^*(x,t) \leq 0$ and from (9) that $J_G^{\min}(x,t) \leq 0$. Thus, for such x, (13) holds.

Case 2 (x is in E at $t = 0$). Thus $l_E(x) \leq 0$, meaning that $J_E^*(x,0) \leq 0$ and $J_E(x,0) \leq 0$, and in addition, due to our assumption that the interiors of the initial sets are disjoint, $J_G^*(x,0) \geq 0$ and $J_G(x,0) \geq 0$. By (3), $\forall t < 0$

$J_E^*(x,t) \leq 0$, and so by (4) $J_G^*(x,t) = J_G^*(x,0) \geq 0$. In our new formulation, $J_E(x,0) \leq 0$ implies $J_E^{\min}(x,t) \leq 0 \; \forall t \leq 0$; by (11) $J_G(x,t) \geq 0$, which in turn implies $J_G^{\min}(x,t) \geq 0, \forall t < 0$. Thus, $\forall t \leq 0$, $J_G^*(x,t) \geq 0$ and $J_G^{\min}(x,t) \geq 0$. By the contrapositive, for such x, (14) is true.

Case 3 (x is outside both G and E at $t = 0$). Thus, $l_G(x) > 0$ and $l_E(x) > 0$. Now, for all $t \leq 0$, x will remain outside both the reach and avoid sets as long as the following constraints are satisfied:

$$\begin{array}{l} J_G^*(x,t) > 0 \\ J_G^{\min}(x,t) > 0 \end{array} \text{ and } \begin{array}{l} J_E^*(x,t) > 0 \\ J_E^{\min}(x,t) > 0 \end{array} \tag{15}$$

For an x under these conditions, (13) is trivially true. Furthermore, while this situation holds, the constrained PDEs (2)–(5) are equivalent to the PDEs and constraints (7)–(12), and so $J_G^*(x,t) = J_G(x,t)$ and $J_E^*(x,t) = J_E(x,t)$. Now consider what will happen if the boundary of one or both of the reach or avoid sets reaches x. Choose $\tau < 0$ to be the first time t that either boundary reaches x.

If $J_G^*(x,\tau) = J_G(x,\tau) = 0$, then (2) guarantees $J_G^*(x,t) \leq 0$ for $t \leq \tau$ and (9) guarantees $J_G^{\min}(x,t) \leq 0$ for $t \leq \tau$. Consequently, for such x, (13) holds $\forall t \leq 0$.

By choice of τ, we know that if $J_E^*(x,\tau) = J_E(x,\tau) = 0$, then $J_G^*(x,\tau) \geq 0$ and $J_G^{\min}(x,\tau) \geq 0$. By (3), $\forall t \leq \tau$, $J_E^*(x,t) \leq 0$, which implies by (4) that $J_G^*(x,t) \geq 0$. Since $J_E(x,\tau) = 0$ implies $\forall t \leq \tau$ that $J_E^{\min}(x,\tau) \leq 0$, (11) requires $\forall t \leq \tau$ that $J_G(x,t) \geq 0$, and so $J_G^{\min}(x,t) \geq 0$. Therefore, for such x, (13) holds for $\tau < t \leq 0$ and (14) holds for $t \leq \tau$. □

We wish to use Lemma 1 and Theorem 1 to claim

$$\mathit{Reach}(G,E) = \{x \in X \mid J_G^{\min}(x,t) < 0\}.$$

However, the two cases (13) and (14) allowed by Lemma 1 must be reconciled before such a claim is true. We do so by making the assumption that the sets defined by (13) are the closures of the sets defined by (14)[1].

Given this assumption, the formulation (7)–(12) provides a characterization of the reach-avoid operator which is numerically more stable than (2)–(5). While the new formulation does smooth out the solution of the Hamilton-Jacobi equations, it is worth noting that discontinuities in u, d, or f will still lead to non-smooth solutions of (7)–(12), and that even if these system parameters are all smooth, it is possible for discontinuous "shocks" to develop as the solution evolves.

3 Computing Reachable Sets

The continuous Hamilton-Jacobi partial differential equation appears frequently in applied mathematics, and so numerical methods for its solution have been

[1] This assumption will hold true as long as the functions J_G^* and $J_G^{\min}$ do not develop plateaus. It turns out to be prudent to avoid plateaus for numerical reasons as well, and we describe a method to avoid their formation in the next section.

well studied [18]. In particular, a set of algorithms called level set methods [19, 20] have been developed to study the propagation of moving interfaces and boundaries using these equations.

A numerical algorithm to solve the Hamilton-Jacobi equations (7)–(12) was developed in [17]; however, the emergence of numerical instabilities meant that the reach set could be computed for only a few dozen timesteps, and even over that short period, sharp edges tended to become rounded by diffusion. Armed with the better behaved (7)–(12) and a new level set implementation, we are able to tackle more complex examples below, tracking the reach set over any finite time interval without significant loss to diffusion.

3.1 Level Set Method Design

The basic method for solving (7) and (8) is the same as that described in [17]: a first-order, upwinding, finite difference scheme that produces an approximation of the viscosity solution to the Hamilton-Jacobi equation [20–22]. We outline several details of our implementation.

Initial conditions: A characteristic of level set methods is that the "level set function" (we use J in the following to represent J_G in (7) or J_E in (8)) is defined as the distance to the boundary being tracked, where distance is negative on the inside of the boundary. Such a definition is compatible with the analysis in the previous section, and so we adopt it for our level set functions.

Boundary conditions: The spatial derivatives in the Hamilton-Jacobi equation are approximated at a grid point by taking differences between the function values at neighboring grid points. For points at the edge of the finite grid, this procedure breaks down. Typical level set methods use Neumann boundary conditions ($\frac{\partial J(x,t)}{\partial n} = 0$, where n is an outward pointing normal) to determine the value of grid points on the boundary. This procedure tends to introduce plateaus to the level set function J close to the boundary, so that it no longer properly measures the distance to the boundary.

Enforcing the constraints: To enforce the constraints (11) and (12), a "max" operator is applied: at each timestep t, for all x,

$$J_G(x,t) = \max(J_G(x,t), -J_E^{\min}(x,t))$$

and similarly for $J_E(x,t)$. This procedure, called *masking J_G with $J_E^{\min}$*, is used in level set methods to ensure that the moving boundary represented by J_G does not enter the forbidden region defined by $J_E^{\min}$ (since $J_E^{\min}(x,t) \leq 0 \implies J_G(x,t) \geq 0$).

An additional complication arises from the discrete timesteps taken by the numeric solver: it is possible for the constraints (11) and (12) to become violated since the various J functions are changing over time and the masking procedure is only applied at the end of each timestep. A conservative solution is

to compute (7)–(12) in the order:

$$\begin{aligned}
&\text{compute } J_G(x, t-\Delta t) \text{ from Hamilton-Jacobi equation,}\\
&\text{compute } J_E(x, t-\Delta t) \text{ from Hamilton-Jacobi equation,}\\
J_G(x, t-\Delta t) &= \max(J_G(x, t-\Delta t), -J_E^{\min}(x, t))\\
J_G^{\min}(x, t-\Delta t) &= \min(J_G(x, t-\Delta t), J_G^{\min}(x, t))\\
J_E(x, t-\Delta t) &= \max(J_E(x, t-\Delta t), -J_G^{\min}(x, t-\Delta t))\\
J_E^{\min}(x, t-\Delta t) &= \min(J_E(x, t-\Delta t), J_E^{\min}(x, t))
\end{aligned}$$

Masking J_G with $J_E^{\min}$ from the previous timestep, but masking J_E with $J_G^{\min}$ from the current timestep ensures that if the reach and avoid sets grow together and overlap, the reach (unsafe) set is over-approximated, and the avoid (safe) set is under-approximated.

Reinitialization: Level set methods attempt to maintain the level set function as a distance measure to the boundary as it evolves. Numeric solutions tend to distort the distance function considerably: the level set function becomes distorted by limited precision computations, discretization and the Neumann boundary conditions. Because the zero level set is the only information of importance to us, a procedure which resets the level set function so that it correctly measures the distance to the current zero level set—without changing the shape of that level set—would smooth out numerical errors in the level set function and yet leave its important data unharmed. This process, called reinitialization, is accomplished in the examples below by running a few discrete timesteps of a solver for the partial differential equation

$$\frac{\partial J_G(x,t)}{\partial t} = \operatorname{sign}(J_G(x,t))(1 - |\operatorname{grad}(J_G(x,t))|)$$

(and similarly for J_E). This process restores the property $|\operatorname{grad}(J_G(x,t))| \approx 1$ near the zero level set, so that J_G is smoothed to approximate a distance measure.

3.2 A Single State, Straight Flight Example

Consider an example representing two aircraft flying at a fixed altitude and constant heading. Each aircraft is allowed to choose its own speed from a given range of values; we control one aircraft and the other is considered the disturbance. Using relative coordinates, in which the controlled aircraft is at the origin with a heading angle of zero, the dynamics of the system are described by

$$\dot{x}_r = -u + d\cos\psi_r, \qquad \dot{y}_r = d\sin\psi_r, \qquad \dot{\psi}_r = 0, \tag{16}$$

where x_r and y_r are the relative spatial coordinates, and ψ_r is the relative heading. The controller fails if the disturbance aircraft manages to enter a circle of radius five units centered at the controlled aircraft at the origin, so $l_G(x) = x_r^2 + y_r^2 - 5^2$.

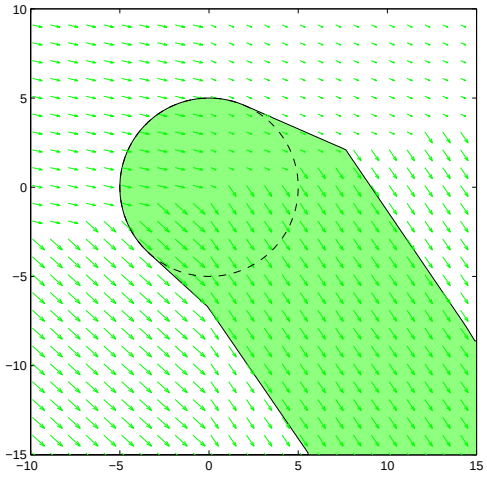

Fig. 1. Shaded Region represents $Reach(G, \emptyset)$ for the Straight Flight Single State Example

If the control (speed of the controlled aircraft) is restricted to $u \in U = [\underline{u}, \overline{u}] \subset \mathrm{R}^+$ and the disturbance (speed of the disturbance aircraft) is restricted to $d \in D = [\underline{d}, \overline{d}] \subset \mathrm{R}^+$, then it was shown in [11, 23] that the optimal control and worst disturbance are

$$u^* = \begin{cases} \underline{u}, & \text{if } x_r > 0, \\ \overline{u}, & \text{if } x_r < 0, \end{cases} \qquad d^* = \begin{cases} \underline{d}, & \text{if } (x_r \cos\psi_r + y_r \sin\psi_r) > 0, \\ \overline{d}, & \text{if } (x_r \cos\psi_r + y_r \sin\psi_r) < 0. \end{cases} \tag{17}$$

Because there is only a single discrete state, the controlled aircraft has no discrete action to force an unsafe continuous state to become safe, and so the avoid set is empty. Given the definition of the unsafe set $G = \{x \in X | l_G(x) \leq 0\}$, the set of unsafe states $Reach(G, \emptyset)$ is shown shaded in Figure 1. The parameters for the example were chosen to be the normalized values:

$$\psi_r = \frac{7\pi}{12}, \qquad U = [\underline{u}, \overline{u}] = [2, 4], \qquad D = [\underline{d}, \overline{d}] = [1, 5].$$

The dashed circle shows the initial unsafe set G, and the grey arrows show the flow field (16) induced by the optimal control choices (17). Notice that the level set algorithm resolves the sharp corners of $Reach(G, \emptyset)$ at the points where u^* or d^* switch.

This example and those below were coded in Matlab 5.3 on an unloaded Sun UltraSparc 10 (a 300 MHz UltraSparc processor with 512 KB cache and 128 MB main memory). Figure 1 was produced from a run with grid spacing $\Delta x = 0.1$ (requiring about 63000 grid points). The 360 timesteps took just under four minutes to complete.

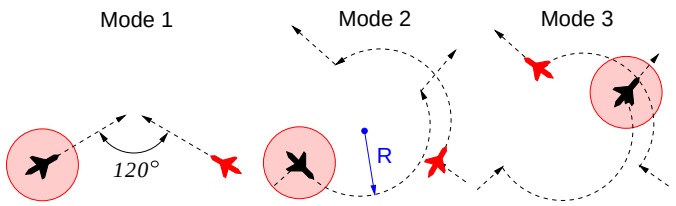

Fig. 2. Aircraft Behavior in the Three Modes

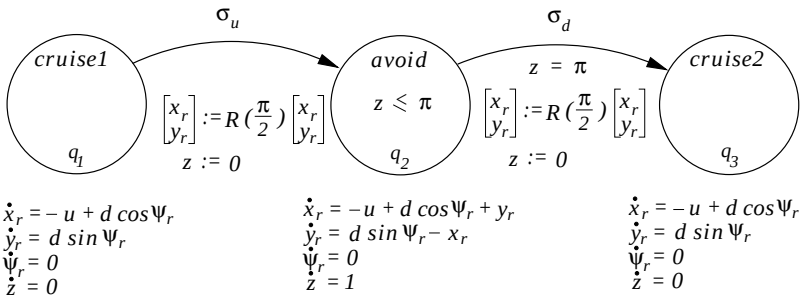

Fig. 3. System Dynamics for the Three Mode Example

3.3 A Three State Example

This example again features the collision avoidance maneuvers of two aircraft at fixed altitude; however, the control is now allowed to initiate a discrete change of state for the system. As shown in Figure 2, the aircraft begin in straight flight at a fixed relative heading (mode 1). At some time, the control may switch both aircraft into mode 2; at which point each makes an instantaneous heading change of 90°, and begins a circular flight path. After completing a semicircular arc in π time units, both aircraft switch to mode 3, make another instantaneous 90° turn, and resume their original headings from mode 1.

The dynamics for the system are shown in Figure 3. In this example, the controller has only a single action: the switch from mode 1 to mode 2. The speed of both aircraft is constant, and the only disturbance action is the uncontrolled switch from mode 2 to mode 3, which occurs a fixed time after mode 2 is entered; the variable z in mode 2 is simply a clock to enforce this switch. The parameters used in the run below are

$$\psi_r = \frac{2\pi}{3} = 120°, \qquad u^* = 3, \qquad d^* = 4.$$

More details on this example can be found in [9, 11].

Running the reachability analysis algorithm to compute W^* requires computing the Pre_d and Pre_u operators for each mode. Let R_i^k be the set of unsafe states computed for mode i in iteration k; in other words, the projection of

$Reach(Pre_d(W^{k+1}), Pre_u(W^{k+1}))$ onto the continuous state space of mode i for iteration $k < 0$ (let $R_i^0 = G$ to handle the $k = 0$ case). Then the set of safe states at iteration $k < 0$ can be written as $W^k = (\cup_{j=k}^{0} \cup_{i=1}^{3} R_i^j)^c$. Define the collision set as before: $G = \{x \in X | l_G(x) \leq 0\}$, where $l_G(x) = x_r^2 + y_r^2 - 5^2$. We can then deduce the precursor operators.

- For mode 3, there are no discrete actions. This mode may be inhabited for any length of time. The projections of the precursor operators onto the continuous state space of mode 3 are:

$$Pre_u(W^k) = \emptyset, \qquad Pre_d(W^k) = R_3^k.$$

- For mode 2, an uncontrolled discrete action switches the system to mode 3, and there are no controlled discrete actions. This mode is inhabited for exactly π time units. The projections of the precursor operators onto the continuous state space of mode 2 are:

$$Pre_u(W^k) = \emptyset, \qquad Pre_d(W^k) = (R_3^k \text{ rotated } \tfrac{\pi}{2}) \cup R_2^k.$$

- For mode 1, a controlled discrete action switches the system to mode 2, and there are no uncontrolled discrete actions. This mode may be inhabited for any length of time. The projections of the precursor operators onto the continuous state space of mode 1 are:

$$Pre_u(W^k) = (R_2^k \text{ rotated } \tfrac{\pi}{2})^c, \qquad Pre_d(W^k) = R_1^k.$$

Figure 4 shows the results of the reach-avoid computation at each iteration for each mode; unsafe states (complement of W^k) are shaded. The set R_i^k appears in column i and row k. A fixed point W^* of safe states is computed after three iterations, and the corresponding bad states of the fixed point $(W^*)^c$ are shaded in the final row of plots.

The unsafe region for mode 1 is the most interesting—as long as the disturbance aircraft is not in this region, the control may initiate the switch to mode 2 and have confidence that the remainder of the maneuver will be carried out safely. The width of the unbounded portion of the unsafe set is controlled by the radius of the turn in mode 2, and can be removed entirely by making the radius large enough.

The four iterations of this simulation, with a grid spacing of $\Delta x = 0.1$ (or about 90000 grid points) each required about 1400 timesteps; for stability reasons, mode 2 was slightly more than half of the work. Wall clock time was about 75 minutes.

3.4 A Three Dimensional Example

To show that this technique extends easily to higher dimensions, we look at a final aircraft collision avoidance scenario. The model is very similar to that examined

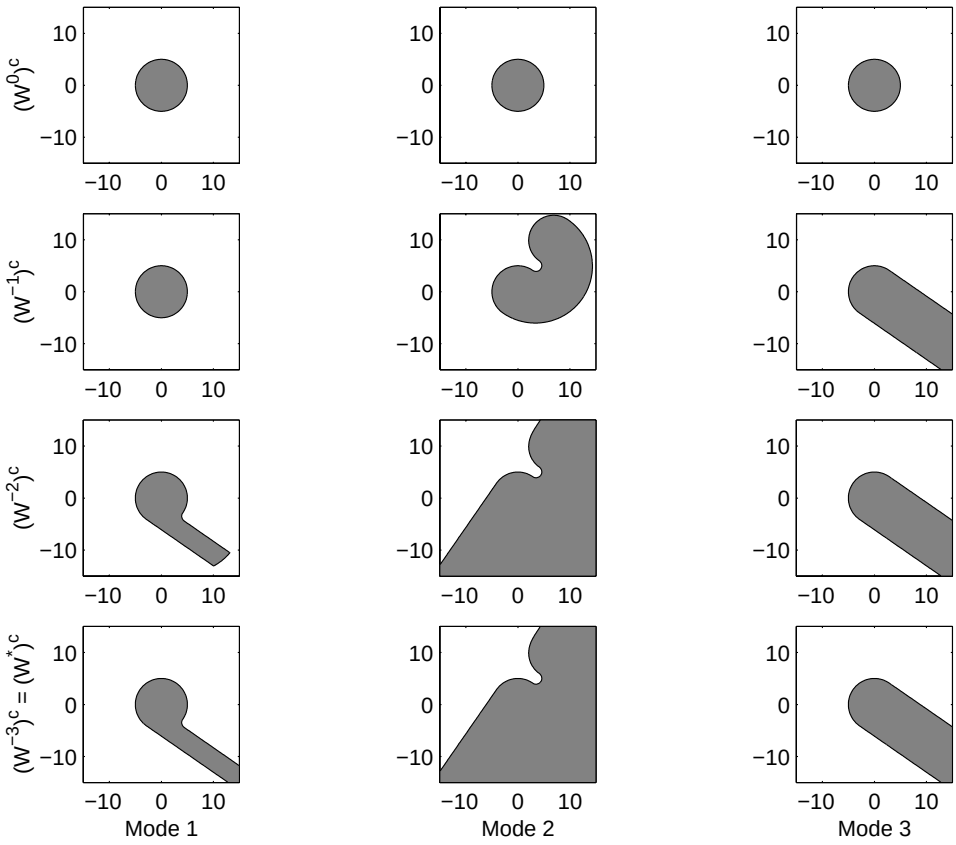

Fig. 4. Unsafe Sets for Three Mode Example

in the first example, except that this time we allow the relative heading of the aircraft to change. Relative angle $\psi_r \in [0, 2\pi)$ is thus our third dimension.

We fix the airspeed of the control aircraft at v_1 and that of the disturbance aircraft at v_2. The control and disturbance inputs are now the angular velocity of the aircraft: $u \in U = [\underline{\omega_1}, \overline{\omega_1}] \subset \mathrm{R}$ and $d \in D = [\underline{\omega_2}, \overline{\omega_2}] \subset \mathrm{R}$. The model is

$$\dot{x}_r = -v_1 + v_2 \cos\psi_r + u y_r, \qquad \dot{y}_r = v_2 \sin\psi_r - u x_r, \qquad \dot{\psi}_r = d - u,$$

For the case where

$$\underline{\omega_1} = \underline{\omega_2} = -1, \qquad \overline{\omega_1} = \overline{\omega_2} = +1,$$

it was shown in [11, pp. 60-62] that the optimal control and disturbance are given by

$$u^* = \mathrm{sign}\left(y_r \frac{\partial J_G}{\partial x_r} - x_r \frac{\partial J_G}{\partial y_r} - \frac{\partial J_G}{\partial \psi_r}\right) \qquad d^* = -\,\mathrm{sign}\left(\frac{\partial J_G}{\partial \psi_r}\right)$$

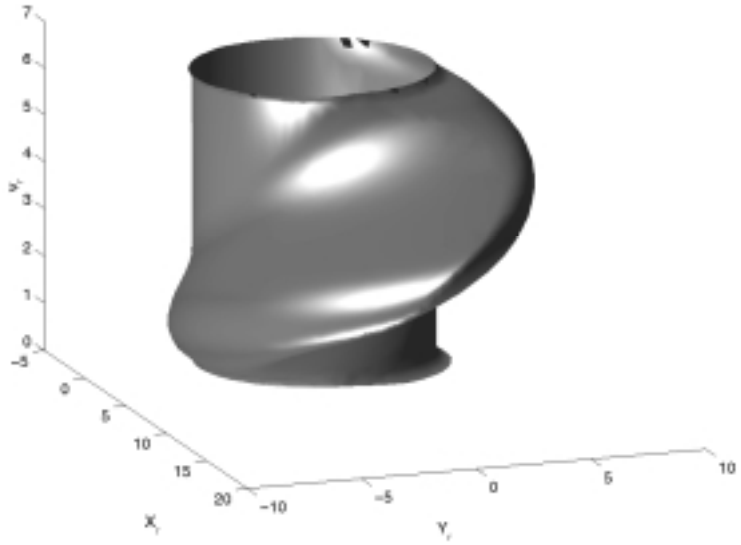

Fig. 5. Unsafe Region for Three Dimensional Example

Because there is only a single discrete state and no discrete actions, the avoid set is empty; the unsafe set is the cylinder $G = \{x \in X | l_G(x) \leq 0\}$ where $l_G(x) = x_r^2 + y_r^2 - 5^2$. A view of $Reach(G, \emptyset)$ for airspeed $v_1 = v_2 = 5$ is shown in Figure 5.

Extending the level set code to three dimensions was painless—a new index for all matrices and a set of boundary conditions (periodic in ψ_r) had to be added. Visualization of the zero sublevel set becomes considerably trickier, but it can be done with Matlab's new isosurface tools. With a grid spacing of $\Delta x = 0.2$ (approximately 400000 grid points), the 400 timesteps required to generate Figure 5 took about 80 minutes to complete.

4 Research Directions

We have presented a numerical algorithm for computing reachable sets of hybrid automata. The algorithm handles nonlinear dynamics with discontinuities, as illustrated by example calculations of both continuous and multi-mode aircraft conflict resolution maneuvers. We are currently investigating further in several directions.

For the examples above, the discrete predecessor maps Pre were determined by hand and hard-coded into the scripts which computed the continuous reach-avoid operator. It is necessary to automatically compute those maps; this will require elimination of existential and universal quantifiers over the set of discrete actions.

As with all finite difference methods, this implementation finds an approximation to the actual solution of the Hamilton-Jacobi equation. In fact, the final example provides proof of the dangers of such approximations: the helical bulge of the unsafe set shown in Figure 5 is computed to protrude farther out if grid spacing is reduced. Methods to quantify the error between exact and approximate reachable sets have not been developed, yet are crucial for proving safety properties. In the reach-avoid calculation, we could use information about error to provide an over-approximation of the unsafe set and an under-approximation of the safe set.

In [9–11], control laws are synthesized assuming that the reach and avoid sets are computed exactly. The implications of set approximation on this process must be evaluated.

As can be seen from the final example, these techniques extend easily to higher dimensions—beyond three dimensions visualization becomes impossible, but the basic level set algorithm remains the same. Of major concern, though, is the exponential growth in the number of grid points as dimension increases. Because the timestep depends on the grid size, using rectilinear gridding with a grid spacing of h in d dimensions requires $\mathcal{O}(h^{d+1})$ work. However, we are currently investigating techniques which will lead to considerable time savings: using compiled code instead of Matlab, computing only on grid points near the zero level set (effectively reducing the dimension of the problem by one), taking advantage of the abundant opportunities for parallelism in the algorithm, and projecting higher dimensional sets onto lower dimensional subspaces.

References

1. R. Alur and D. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, pp. 183–235, 1994.
2. R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho, "Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems," in *Hybrid Systems* (R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, eds.), LNCS, pp. 366–392, New York: Springer Verlag, 1993.
3. T. A. Henzinger, P. H. Ho, and H. Wong-Toi, "A user guide to HYTECH," in *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems* (E. Brinksma, W. Cleaveland, K. Larsen, T. Margaria, and B. Steffen, eds.), no. 1019 in LNCS, pp. 41–71, Springer Verlag, 1995.
4. C. Daws, A. Olivero, S. Tripakis, and S. Yovine, "The tool KRONOS," in *Hybrid Systems III, Verification and Control*, no. 1066 in LNCS, pp. 208–219, Springer Verlag, 1996.
5. O. Maler, A. Pnueli, and J. Sifakis, "On the synthesis of discrete controllers for timed systems," in *STACS 95: Theoretical Aspects of Computer Science* (E. W. Mayr and C. Puech, eds.), no. 900 in LNCS, pp. 229–242, Munich: Springer Verlag, 1995.
6. H. Wong-Toi, "The synthesis of controllers for linear hybrid automata," in *Proceedings of the IEEE Conference on Decision and Control*, (San Diego, CA), 1997.
7. A. Deshpande, *Control of Hybrid Systems*. PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 1994.

8. J. Lygeros, *Hierarchical, Hybrid Control of Large Scale Systems.* PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 1996.
9. C. Tomlin, J. Lygeros, and S. Sastry, "Synthesizing controllers for nonlinear hybrid systems," in *Hybrid Systems: Computation and Control* (T. Henzinger and S. Sastry, eds.), no. 1386 in LNCS, pp. 360–373, New York: Springer Verlag, 1998.
10. J. Lygeros, C. Tomlin, and S. Sastry, "On controller synthesis for nonlinear hybrid systems," in *Proceedings of the IEEE Conference on Decision and Control*, (Tampa, FL), pp. 2101–2106, 1998.
11. C. J. Tomlin, *Hybrid Control of Air Traffic Management Systems.* PhD thesis, Department of Electrical Engineering, University of California, Berkeley, 1998.
12. M. Greenstreet and I. Mitchell, "Integrating projections," in *Hybrid Systems: Computation and Control* (S. Sastry and T. Henzinger, eds.), no. 1386 in LNCS, pp. 159–174, Springer Verlag, 1998.
13. T. Dang and O. Maler, "Reachability analysis via face lifting," in *Hybrid Systems: Computation and Control* (S. Sastry and T. Henzinger, eds.), no. 1386 in LNCS, pp. 96–109, Springer Verlag, 1998.
14. A. Chutinan and B. H. Krogh, "Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations," in *Hybrid Systems: Computation and Control* (F. Vaandrager and J. H. van Schuppen, eds.), no. 1569 in LNCS, pp. 76–90, New York: Springer Verlag, 1999.
15. A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis," in *Hybrid Systems: Computation and Control* (B. Krogh and N. Lynch, eds.), LNCS (these proceedings), Springer Verlag, 2000.
16. O. Botchkarev and S. Tripakis, "Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations," in *Hybrid Systems: Computation and Control* (B. Krogh and N. Lynch, eds.), LNCS (these proceedings), Springer Verlag, 2000.
17. C. Tomlin, J. Lygeros, and S. Sastry, "Computing controllers for nonlinear hybrid systems," in *Hybrid Systems: Computation and Control* (F. Vaandrager and J. H. van Schuppen, eds.), no. 1569 in LNCS, pp. 238–255, New York: Springer Verlag, 1999.
18. M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman equations.* Boston: Birkh auser, 1997.
19. S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.
20. J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science.* New York: Cambridge University Press, 1996.
21. M. G. Crandall and P.-L. Lions, "Viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society*, vol. 277, no. 1, pp. 1–42, 1983.
22. M. G. Crandall, L. C. Evans, and P.-L. Lions, "Some properties of viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society*, vol. 282, no. 2, pp. 487–502, 1984.
23. C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A case study in multi-agent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, pp. 509–521, April 1998.