AA278A Lecture Notes 10. Controller Synthesis for Hybrid Systems

Claire J. Tomlin

May 23, 2005

In the last two weeks, we have discussed controller synthesis for:

- Discrete Systems: design is characterized by a fixed point of a difference equation
- Continuous Systems: design is characterized by a fixed point of a partial differential equation

In each case, the solution is characterized as a fixed point of an equation, which we refer to as the Hamilton-Jacobi equation. In these lecture notes, we bring the discrete and continuous parts together, and discuss the problem of designing controllers for hybrid systems. Our treatment follows that of [1, 2, 3, 4].

1 Problem Formulation

Recall that the plant is modeled by a hybrid automaton $H = (Q, X, \text{Init}, In, f, \text{Dom}, R, Out)$ as described in Lecture Notes 7.

To avoid technical problems we assume that:

1. f is Lipschitz continuous in x and continuous in $w \in In$;
2. for all $q \in Q$ and for all $\sigma \in \Sigma, w \in W$, $\text{Dom}(q)$ is an open set.
3. for all $(q, x) \in Q \times X$, and for all $(\sigma_1, u) \in \Sigma_1 \times U$ there exists $(\sigma_2, d) \in \Sigma_2 \times D$ such that:
$$[(x, (\sigma_1, \sigma_2), (u, d)) \in \text{Dom}(q)] \vee [R(q, x, (\sigma_1, \sigma_2), (u, d)) \neq \emptyset]$$

Part 1 is standard, and is needed for existence of continuous evolution. Part 2 implies that we need not worry about what happens on the boundary of the invariant set. Finally, part 3 (together with part 2) implies that the controller cannot block the system execution. Notice that this assumption is not symmetric for (σ_1, u) and (σ_2, d). The reason is that, since we

are dealing with safety specifications, it will never be to the benefit of (σ_2, d) to stop the execution.

As in the discrete and continuous cases, we will try to establish the largest controlled invariant subset of a given set $F \subseteq Q \times X$, and design a controller that renders this set invariant. Again, to avoid technical problems, we will assume that F is a closed set.

We will again take an adversarial approach and treat the design as a game between (σ_1, u) and (σ_2, d). Whenever possible we will give the advantage to (σ_2, d), in particular:

1. In the order of play ((σ_1, u) will be the "leader" of the game).
2. When resolving non-determinism.

2 Definitions of Operators

Notice that the disturbance has two choices. It can:

1. Try to make the system "jump" outside F.
2. Try to "steer" the system outside F along continuous evolution.

The control also has two choices:

1. Try to "jump" to another state in F when the disturbance tries to steer out of F.
2. Try to "steer" the system and keep it in F along continuous evolution.

To characterize alternative 1 for the control, introduce the *controllable predecessor operator*, $\mathrm{Pre}_1 : 2^{Q \times X} \to 2^{Q \times X}$, which, given a set $K \subseteq Q \times X$ returns:

$$\begin{aligned} \mathrm{Pre}_1(K) \;=\; & \{(q,x) \in K : \exists (\sigma_1, u) \in \Sigma_1 \times U,\ \forall (\sigma_2, d) \in \Sigma_2 \times D,\ (x, \sigma_1, \sigma_2, u, d) \notin \mathrm{Dom}(q) \\ & \wedge R(q, x, \sigma_1, \sigma_2, u, d) \subseteq K\} \end{aligned}$$

To characterize alternative 1 for the disturbance, introduce the *uncontrollable predecessor operator*, $\mathrm{Pre}_2 : 2^{Q \times X} \to 2^{Q \times X}$, which, given a set $K \subseteq Q \times X$ returns:

$$\begin{aligned} \mathrm{Pre}_2(K) \;=\; & \{(q,x) \in K : \forall (\sigma_1, u) \in \Sigma_1 \times U\ \exists (\sigma_2, d) \in \Sigma_2 \times D \\ & R(q, x, \sigma_1, \sigma_2, u, d) \cap K^c \neq \emptyset\} \cup K^c \end{aligned}$$

Therefore $\mathrm{Pre}_1(K)$ contains all states in K for which controllable actions (σ_1, u) can force the state to remain in K for at least one step in the discrete evolution. $\mathrm{Pre}_2(K)$, on the other hand, contains all states in K^c, the complement of K, as well as all states from which uncontrollable actions (σ_2, d) may be able to force the state outside of K. In the definition of Pre_1, the controllable actions are required to be able to *force* a transition (hence the Inv in the formula). In contrast, for Pre_2, we simply require that a transition be possible, giving the

advantage to the uncontrollable actions. The controllable and uncontrollable predecessors will form the discrete part of the algorithm for computing controlled invariant sets.

Some simple facts about these two operators:

Proposition 1 *For all* $K \subseteq Q \times X$, $\mathrm{Pre}_1(K) \subseteq K$, $\mathrm{Pre}_2(K) \supseteq K^c$ *and* $\mathrm{Pre}_1(K) \cap \mathrm{Pre}_2(K) = \emptyset$

Remarks:

- The two operators are asymmetric.
- The order of the quantifiers is consistent with the control being the leader in the game.
- Since all non-determinism is resolved in favor of the disturbance, the control has to work harder:
  - it has to ensure a transition back into K exists (second condition in the definition of $\mathrm{Pre}_1(K)$),
  - it has to be able to "force" the transition (no mention of *Dom* in the definition of $\mathrm{Pre}_2(K)$),
  - it has to ensure all possible transitions stay in K (first condition in the definition of $\mathrm{Pre}_1(K)$).

Finally, to characterize alternative 2 for both control and disturbance, we define the Reach : $2^{Q \times X} \times 2^{Q \times X} \to 2^{Q \times X}$ operator:

Definition 2 (Reach) *Consider two subsets* $G \subseteq Q \times X$ *and* $E \subseteq Q \times X$ *such that* $G \cap E = \emptyset$. *The Reach operator is defined as*

$$\begin{aligned} \mathrm{Reach}(G, E) = \; & \{(q, x) \in Q \times X \mid \forall u \in \mathcal{U} \; \exists d \in \mathcal{D} \text{ and } t \geq 0 \text{ such that} \\ & (q(t), x(t)) \in G \text{ and } (q(s), x(s)) \in \Pi(\mathrm{Dom}) \setminus E \text{ for } s \in [0, t]\} \end{aligned} \tag{1}$$

where $(q(s), x(s))$ *is the continuous state trajectory of* $\dot{x} = f(q(s), x(s), u(s), d(s))$ *starting at* (q, x) *and* $\Pi(\mathrm{Dom})$ *represents the state space components of* Dom. *The set* $\mathrm{Reach}(G, E)$ *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* G *while avoiding an "escape" set* E.

3 Basic Algorithm

Using the above definitions, the following algorithm can now be formulated for computing the largest controlled invariant subset of a given set F.

Algorithm 1 (Controlled Invariant Set)

Initialization:
$W^0 = F$, $W^1 = \emptyset$, $i = 0$
while $W^i \neq W^{i+1}$ **do**
begin
$W^{i-1} = W^i \setminus \mathrm{Reach}(\mathrm{Pre}_2(W^i), \mathrm{Pre}_1(W^i))$
$i = i - 1$
end

In the first step of this algorithm, we remove from F all states from which there is a disturbance $d(\cdot) \in \mathcal{D}$ forcing the system either outside F or to states from which an environment action $\sigma_2 \in \Sigma_2$ may cause transitions outside F, without first touching the set of states from which there is a control action $\sigma_1 \in \Sigma_1$ keeping the system inside F. 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 as W^*.

In order to implement this algorithm, we need to calculate Pre_1, Pre_2, and Reach. The computation of Pre_1 and Pre_2 requires inversion of the transition relation R subject to the quantifiers $\exists$ and $\forall$; existence of this inverse can be guaranteed subject to well understood conditions on the map R. The computation of Reach, for each discrete state q, may be formulated as a constrained Hamilton-Jacobi equation or variational inequality.

Recall that along continuous evolution the value of the discrete state remains constant. Therefore, since the computation of the Reach operator involves only continuous evolution it can be carried out for each discrete state separately. Fix the value of $q \in Q$ and let $l_G : X \to \mathbb{R}$ and $l_E : X \to \mathbb{R}$ be differentiable functions such that $G \triangleq \{x \in X : l_G(x) \leq 0\}$ and $E \triangleq \{x \in X : l_E(x) \leq 0\}$.

Then the set of states which reaches G without entering E is

$$G(t) = \{x \in X : J_G(x,t) \leq 0\} \quad (2)$$
$$E = \{x \in X : J_E(x) \leq 0\} \quad (3)$$
$$(4)$$

where

$$\frac{\partial J_G(x,t)}{\partial t} + \min(0, H(x, \frac{\partial J_G(x,t)}{\partial x})) = 0 \quad (5)$$
$$\text{subject to } J_G(x,t) \geq J_E(x) \quad (6)$$

This is a constrained Hamilton-Jacobi equation – the constraint $J_G(x,t) \geq J_E(x)$ ensures that the evolution of $J_G^*(x,t)$ is frozen in set E. Figure 1 illustrates a sample evolution.

Remarks

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

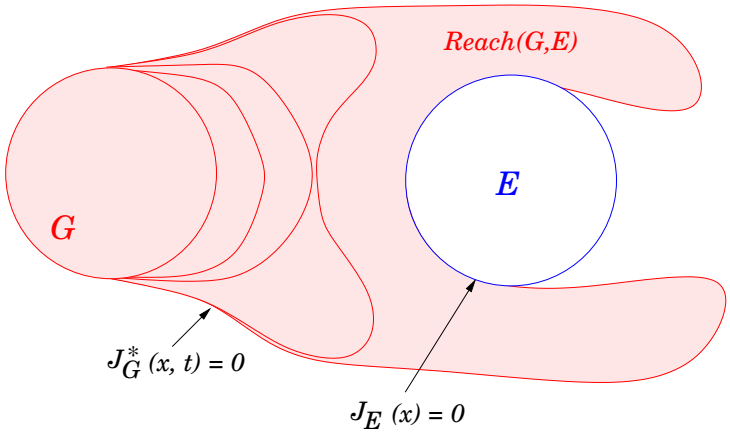

Figure 1: The computation of Reach(G, E) in a single discrete state q.

is known to be restricted [5]. In general, the basic algorithm of the previous section is semi-decidable when the operators $\mathrm{Pre}_1, \mathrm{Pre}_2, \mathrm{Reach}$ are computable. For example, when the continuous state dynamics are constant and the guards and resets are polyhedra, then the operators $\mathrm{Pre}_1, \mathrm{Pre}_2, \mathrm{Reach}$ map polyhedral sets back into polyhedral sets. These hybrid systems are referred to as *linear hybrid automata*.

Consider the three-mode aircraft conflict resolution example from Lecture 11.

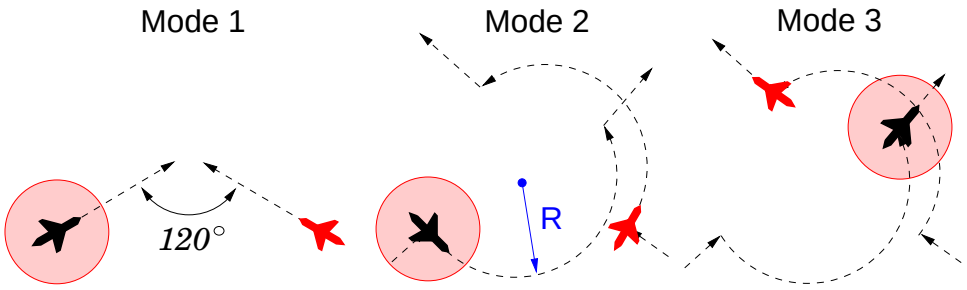

Figure 2: Two aircraft in three modes of operation: in modes 1 and 3 the aircraft follow a straight course and in mode 2 the aircraft follow a half circle. The initial relative heading (120°) is preserved throughout.

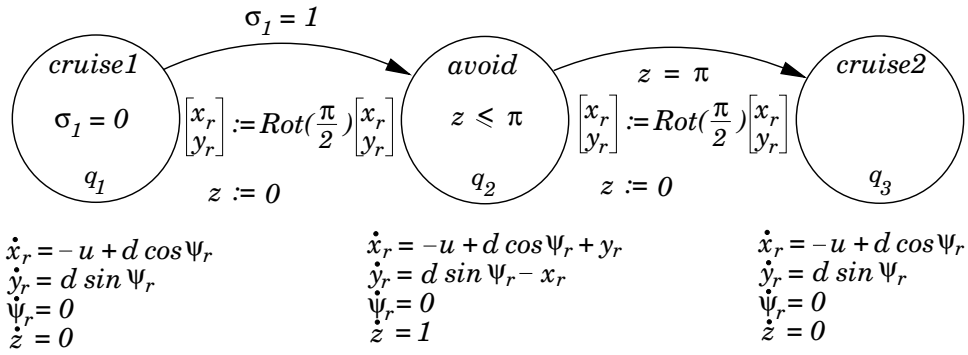

Figure 3: In q_1 both aircraft follow a straight course, in q_2 a half circle, and in q_3 both aircraft return to a straight course.

We assume that for this example the speeds (v_1, v_2) of both aircraft are constant even in the straight modes, so that the input and disturbance sets are singletons $(U = v_1, D = v_2)$ and $u^* = v_1, d^* = v_2$. The general case, in which U and D are ranges of possible speeds, is considered in the examples in [6, 7]. Recall that our goal is to calculate the relative distance at which the system may safely switch from mode 1 to mode 2, and the minimum turning radius R in mode 2, to ensure that separation between aircraft is maintained. The evolution of the protected zone in each mode, assuming no switches, is computed using the continuous-time Hamilton-Jacobi method. The unsafe set G is defined as: $G = \{q_1, q_2, q_3\} \times \{x \in X \mid l(x) \leq 0\}$ where $l(x) = x_r^2 + y_r^2 - 5^2$. Let $G_i = (q_i, \{x \in X \mid l(x) \leq 0\})$ represent the unsafe set in mode i. Thus the set $\{x \in X \mid J^*_{G_i}(x) \leq 0\}$ where $J^*_{G_i}$ is the optimal cost, is the backwards evolution of the protected zone in mode i, assuming no switches between modes.

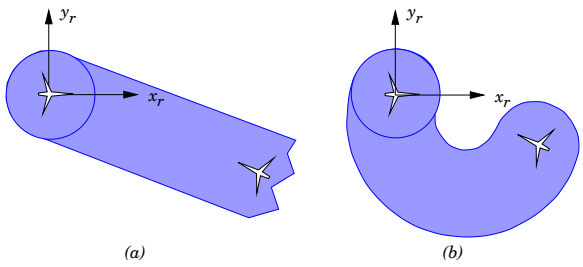

Figure 4: $J^*_{G_i}(x) \leq 0$ for (a) Modes 1 and 3 ($i = 1, 3$), $\omega_1 = \omega_2 = 0$ (the jagged edge means the set extends infinitely), (b) Mode 2 ($i = 2$), $\omega_1 = \omega_2 = 1$. In both cases, $\psi_r = 2\pi/3$, and $v_1 = v_2 = 5$.

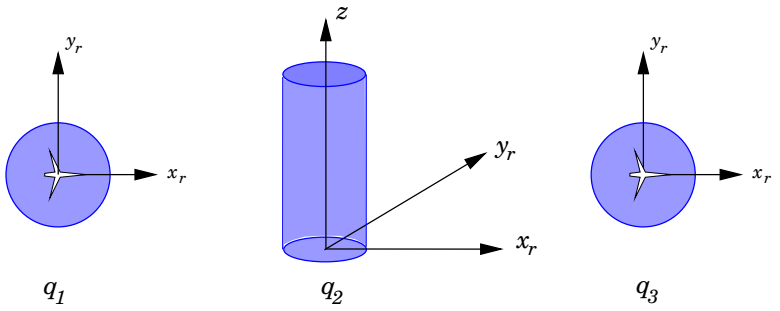

Figure 5: $(W^0)^c$.

These sets are shown in Figure 4. In both cases, the relative heading between aircraft is assumed fixed at $\psi_r = 2\pi/3$ (because of our assumption that aircraft switch modes instantaneously).

We implement Algorithm 1 for this example, at each step computing the sets Pre_1, Pre_2, and $\text{Reach}(\text{Pre}_2, \text{Pre}_1)$. In the first step, $W^0 = F \triangleq G^c$, the complement of G:

$$\begin{aligned} W^0 &= ((q_1, \{x \in X \mid l(x) \leq 0\}^c \cap \{x \in X \mid z = 0\}) \cup (q_2, \{x \in X \mid l(x) \leq 0\}^c) \\ &\quad \cup (q_3, \{x \in X \mid l(x) \leq 0\}^c \cap \{x \in X \mid z = 0\})) \end{aligned} \tag{7}$$

as shown in Figure 5 (the complement is shown in the figure).

$$\text{Pre}_1(W^0) = (q_1, \{x \in X \mid l(x) \leq 0\}^c \cap \{x \in X \mid z = 0\}) \tag{8}$$

$$\text{Pre}_2((W^0)^c) = G \tag{9}$$

Note that $\text{Pre}_1(W^i) \subseteq \{(q_1, X)\}$ for all i, since σ_1 labels transitions from q_1. The set W^{-1} (Figure 6) is

$$W^{-1} = W^0 \backslash \text{Reach}(\text{Pre}_2((W^0)^c), \text{Pre}_1(W^0)) \tag{10}$$

The set W^{-2} involves computing $\text{Reach}(\text{Pre}_2((W^{-1})^c), \text{Pre}_1(W^{-1}))$, this computation is illustrated in Figure 7(a) and the set is shown in Figure 7(b) as the shaded region. Continuing, a fixed point is reached after 3 iterations: Figure 8 illustrates this fixed point $W^* = W^{-3}$ in q_1. Since we assumed in this example that the continuous control input $u = v_1$ is fixed, we need only design the discrete part of the controller σ_1 and the radius of the maneuver R. The

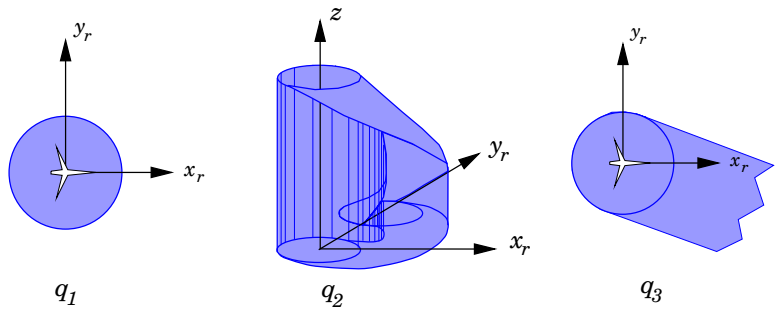

Figure 6: $(W^{-1})^c$. The jagged edge in q_3 means that the set extends infinitely.

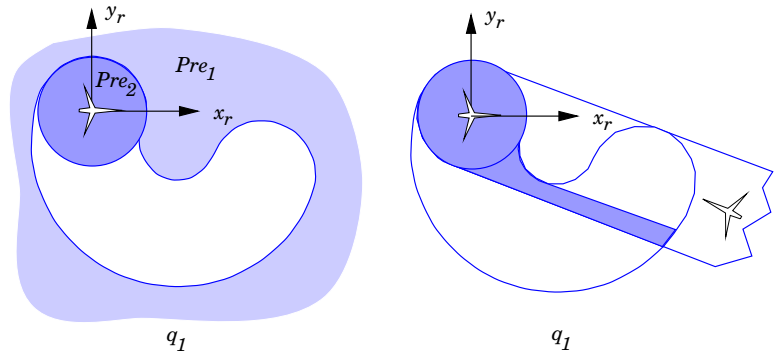

Figure 7: (a) $\mathrm{Pre}_1(W^{-1})$ and $\mathrm{Pre}_2(W^{-1})$ in q_1; (b) $\mathrm{Reach}(\mathrm{Pre}_2(W^{-1}), \mathrm{Pre}_1(W^{-1}))$ in q_1.

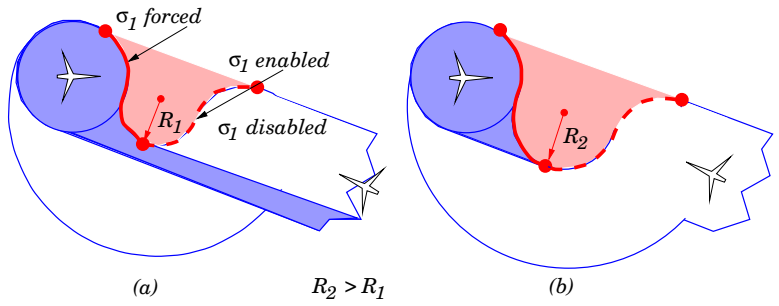

Figure 8: Showing the enabling and forcing boundaries for σ_1 in state q_1; and the result of increasing the radius of the turn in the avoid maneuver to increase W^*.

design is as illustrated in Figure 8(a): the enabling and forcing of σ_1 occurs at the boundary of W^* as shown, as explained below. The transition from q_1 to q_2, governed by σ_1, must be disabled until the relative position of the two aircraft reach the dashed line as shown, otherwise the aircraft will lose separation with each other either during the maneuver or after the maneuver is complete. At the dashed line, σ_1 is enabled, meaning the transition from q_1 to q_2 may occur at any time. σ_1 remains enabled until the dynamics reach the solid line (boundary of W^*), at which point it must be both enabled and forced: otherwise the aircraft lose separation immediately. Note that there are states (x_r, y_r) which are not rendered safe by the maneuver. Indeed, if the initial state is in the darker shaded region shown in Figure 8(a), then the aircraft are doomed to collide. Figure 8(b) displays the result of increasing the radius of the turn in q_2. Notice that the set W^* (the complement of the shaded region) increases as the turning radius increases. This implies that the maneuver renders a larger subset of the state space safe. Figure 8(b) shows the critical value of the turning radius, for which the maneuver is guaranteed to be safe, provided the conflict is detected early enough.

We have recently designed a tool for computing reachable sets for hybrid systems based on this level set technique [8, 3], have implemented it in C, and we have used it to compute reachable sets for several examples, including the first example in this paper. Using a grid spacing of $\Delta x = 0.1$ (or about 90000 grid points) each iteration of this example required about 1400 timesteps on a Sun UltraSparc 10 (a 300 MHz UltraSparc processor with 512 KB cache and 128 MB main memory).

Using this for the computation of the continuous evolution in the hybrid system algorithm, the three-mode conflict resolution example may be computed automatically as shown in Figure 9 below.

4 Other Computational Methods involving Approximations

Other methods have been presented for approximating the reach set calculation. One idea has been to use rectangular hybrid automata to approximate conservatively the reach set of general hybrid automata. This procedure consists of sub-dividing the state space into regions where one can find upper and lower bounds for each component of the right hand side of the continuous dynamics and using the reach set analysis for the resulting rectangular hybrid system. The package HyTech does precisely this computation provided that the guards and invariants are polyhedra [9]. A synthesis procedure based on this appears in the paper of Wong-Toi [10]. The main advantage of this approximation procedure is that it deals with a class of systems for which the synthesis algorithm is semi-decidable. The main drawback is that there is an exponential growth in the number of discrete states in approximating the continuous dynamics. The successor to HyTech is a package called HyperTech [11] which reduces the conservativeness of HyTech by using interval arithmetic with some systematic checks to reduce the divergence of interval arithmetic estimates to approximate reach sets. A controller design procedure using HyperTech has yet to be completed.

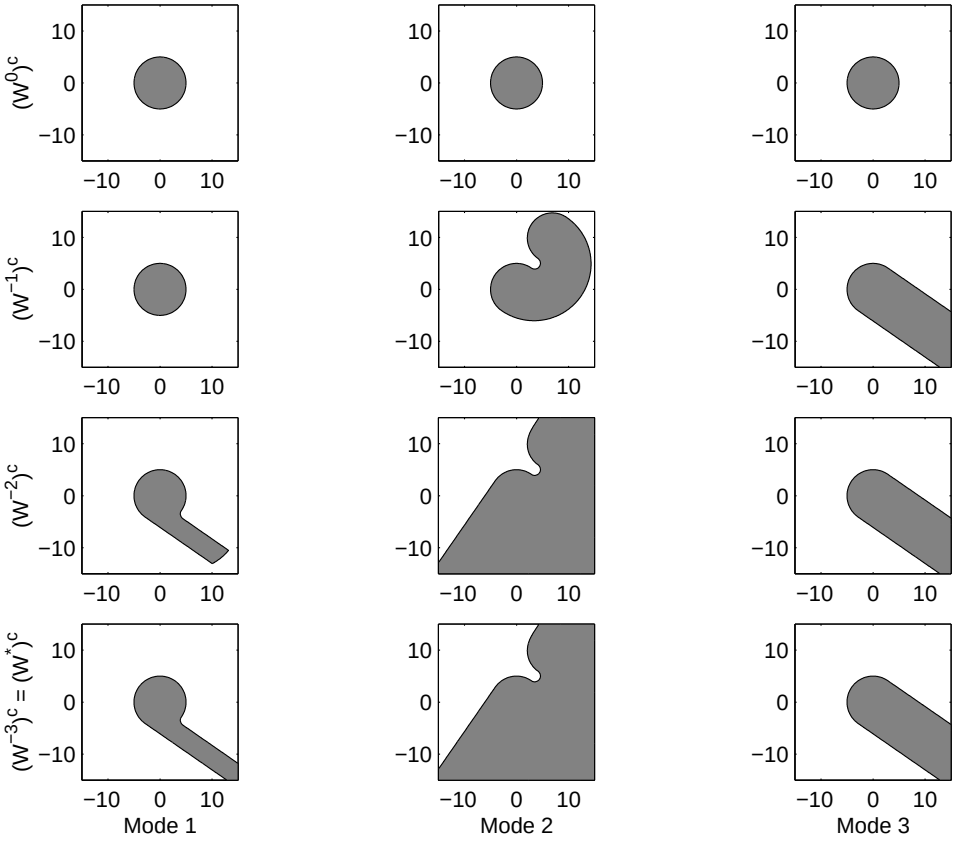

Figure 9: Unsafe Sets for Three Mode Example

Approximating Dynamics with Differential Inclusions. Suppose the continuous dynamics in the nonlinear hybrid system were approximated with the differential inclusion

$$\dot{x} \in g(q, x) \tag{11}$$

where $g(q, x) = \{f(q, x, u, d) \mid \forall u \in U, d \in D\}$. A computationally efficient method for approximating the reach set of $g(q, x)$ is to conservatively approximate $g(q, x)$ by a set of constant inclusions, each of the form

$$\dot{x} \in [g_{\min}, g_{\max}] \tag{12}$$

and then to compute the reach set of the constant inclusions. This method is presented in [12], [13] where it is proved that the approximation error can be made arbitrarily small by approximating the differential inclusion arbitrarily closely (ϵ-approximation). An advantage of this method is that the class of constant inclusions used to approximate the differential inclusion is known to be decidable, thus one can guarantee that the reachable set as $t \to -\infty$ can be computed in a finite number of steps. The amount of preprocessing required to initially approximate the dynamics may be quite formidable however, especially to achieve a close approximation of the true reach set.

Approximating non-smooth sets with smooth sets. We have shown that the reach set at any time $t \in (-\infty, 0]$ may have a non-smooth boundary due to switches in (u^*, d^*), non-smooth initial data, or the formation of shocks. The level set scheme propagates these discontinuities, yet its implementation may require a very small time step to do this accurately. In [14] we present a method for over-approximating such non-smooth sets with sets for which the boundary is continuously differentiable, by using smoothing functions to derive smooth inner and outer approximations. By applying Algorithm 2 to smooth inner and outer approximations of the sets G and E, we calculate smooth inner and outer approximations to the true reach set.

Ellipsoidal Methods. A similar idea is to use ellipsoids as inner and outer approximations to the reach set [15], [16]. To preserve the propagation of ellipsoids the continuous dynamics in each of the discrete locations needs to be approximated by linear dynamics. Bounds on the conservativeness of this approximation and their validity have not yet been worked out. However, [16] presents efficient algorithms for calculating both the minimum volume ellipsoid containing given points, and the maximum volume ellipsoid in a polyhedron, using a matrix determinant maximization procedure subject to linear matrix inequality constraints.

References

[1] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999.

[2] C. J. Tomlin, J. Lygeros, and S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88(7):949–970, July 2000.

[3] I. Mitchell, A. M. Bayen, and C. J. Tomlin. Validating a Hamilton-Jacobi approximation to hybrid system reachable sets. In M. D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, LNCS 2034, pages 418–432. Springer Verlag, 2001.

[4] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A Time-Dependent Hamilton-Jacobi Formulation of Reachable Sets for Continuous Dynamic Games. *IEEE Transactions on Automatic Control*, June 2005.

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

[6] 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*, 43(4):509–521, April 1998.

[7] Claire J. Tomlin. *Hybrid Control of Air Traffic Management Systems*. PhD thesis, Department of Electrical Engineering, University of California, Berkeley, 1998.

[8] I. Mitchell and C. Tomlin. Level set methods for computation in hybrid systems. In B. Krogh and N. Lynch, editors, *Hybrid Systems: Computation and Control*, LNCS 1790, pages 310–323. Springer Verlag, 2000.

[9] T. A. Henzinger, P. H. Ho, and H. Wong-Toi. A user guide to HYTECH. In E. Brinksma, W. Cleaveland, K. Larsen, T. Margaria, and B. Steffen, editors, *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*, number 1019 in LNCS, pages 41–71. Springer Verlag, 1995.

[10] 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.

[11] T. A. Henzinger, B. Horowitz, and R. Majumdar. Rectangular hybrid games. In *Proceedings of the 10th International Conference on Concurrence Theory (CONCUR)*. 1999.

[12] A. Puri. *Theory of Hybrid Systems and Discrete Event Systems*. PhD thesis, Department of Electrical Engineering, University of California, Berkeley, 1995.

[13] A. Puri, P. Varaiya, and V. Borkar. ϵ-approximation of differential inclusions. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2892–2897, New Orleans, LA, 1995.

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

[15] A. B. Kurzhanski and I. Valyi. *Ellipsoidal calculus for estimation and control*. Birkhauser, Boston, 1997.

[16] L. Vandenberghe, S. Boyd, and S.-P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, 1998.