Conflict Resolution for Air Traffic Management: a Study in Multi-Agent Hybrid Systems *

Claire Tomlin, George J. Pappas and Shankar Sastry
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley, CA 94720
{clairet,gpappas,sastry}@eecs.berkeley.edu

Abstract

Air Traffic Management (ATM) of the future allows for the possibility of *free flight*, in which aircraft choose their own optimal routes, altitude and speed. In a free flight environment, the trajectories of different aircraft may be conflicting, in which case aircraft may or may not cooperate in resolving the conflict. In this paper, both cooperative and noncooperative conflict resolution methods are presented. Noncooperative methods are based on game theory in which each aircraft models the actions of other aircraft as disturbances and chooses its actions to be safe against the worst possible disturbance. The solutions to the games partition the state space into safe and unsafe sets and the control strategies are abstracted into discrete protocols on hybrid automata. In cooperative methods, aircraft perform coordinated maneuvers in order to avoid conflict. The merging of inter-aircraft coordination protocols with the individual continuous control laws of each aircraft gives rise to hybrid systems which have been made safe by design. Two examples of conflict resolution using speed and heading changes are worked out in detail.

1 Introduction

Air transportation systems are faced with soaring demands for air travel. According to the Federal Aviation Administration (FAA), the annual air traffic rate in the U.S. is expected to grow by 3 to 5 percent annually for at least the next 15 years. The current National Airspace System (NAS) architecture and management will not be able to efficiently handle this increase because of several limiting factors including:

*Research supported by the Army Research Office under grants DAAH 04-95-1-0588 and DAAH 04-96-1-0341 and by NASA under grant NAG 2-1039.

- **Inefficient airspace utilization:** Currently, the airspace is very rigidly structured and aircraft are forced to travel along predetermined jetways. This is clearly not optimal and disallows aircraft to fly directly to the destination and take advantage of favorable winds. This problem is particularly evident in transoceanic routes which are experiencing the greatest demand growth (for example, nearly 15% annually across the Pacific Ocean).
- **Increased Air Traffic Control (ATC) workload:** Separation among aircraft as well as vectoring aircraft in order to avoid weather hazards is performed centrally by ATC. The resulting centralized architecture places an enormous burden on controllers. In congested areas, such the regions close to urban airports referred to as TRACONS, controllers frequently simplify their heavy workload by keeping aircraft in holding patterns outside the TRACON.
- **Obsolete technology:** The computer technology used in most ATC centers is nearly 30 years old. Communication is restricted to congested voice communication between the aircraft and ATC. Navigation is performed by flying over fixed VHF Visual Omni-Directional Range (VOR) points. Developing weather storms are sometimes reported from aircraft to ATC.

In view of the above problems and in an effort to meet the challenges of the next century, the aviation community is working towards an innovative concept called *Free Flight* [1]. Free Flight allows pilots to choose their own routes, altitude and speed and essentially gives each aircraft the freedom to self-optimize. Aircraft flexibility will be restricted only in congested airspace in order to ensure separation among aircraft, or to prevent unauthorized entry of special use airspace (such as military airspace). Free Flight is potentially feasible because of enabling technologies such as Global Positioning Systems (GPS), Datalink communications, Automatic Dependence Surveillance-Broadcast (ADS-B) [2, 3], Traffic Alert and Collision Avoidance Systems (TCAS) [4] and powerful on-board computation. In addition, tools such as the Center-TRACON Automation System (CTAS) [5] will serve as decision support tools for ground controllers in an effort to reduce ATC workload and optimize capacity close to highly congested urban airports.

The above technological advances will also enable air traffic controllers to accommodate future air traffic growth by restructuring NAS towards a more decentralized architecture. The current system is extremely centralized with ATC assuming most of the workload. Sophisticated on-board equipment allow aircraft to share some of the workload, such as navigation, weather prediction and aircraft separation, with ground controllers. The resulting decentralized architecture clearly reduces the workload of ground controllers. Furthermore, such a distributed architecture is more fault tolerant to failures of centralized agencies. In order to improve the current standards of safety in an unstructured, Free Flight environment, automatic conflict detection and resolution algorithms are vital. Sophisticated algorithms which predict and automatically resolve conflicts would be used either on the ground or on-board, either as advisories or as part of the Flight Vehicle Management

System (FVMS) of each aircraft. Current research endeavors along this direction include [6, 7, 8, 9, 10]. In [6] probabilities of conflict are computed off-line using Monte-Carlo simulations while [7] presents a conflict prediction methodology assuming wind modeling and tracking errors. The work of [8, 9] formulates conflict resolution as an optimal control problem whereas [10] treats the problem as one optimization problem. In our work, we strive for an approach and a modeling scheme for which we can verify the safety of the conflict resolution method. We present in [11] a decentralized architecture for Air Traffic Management (ATM). In our design paradigm, aircraft are allowed to self-optimize in the spirit of Free Flight and coordinate with neighboring aircraft in order to resolve potential conflicts.

In the current paper, conflicts are classified as *cooperative* or *noncooperative*. In noncooperative conflicts, aircraft do not collaborate in resolving the conflict. A natural framework for this type of problem in which many aircraft have conflicting objectives is zero-sum noncooperative dynamic game theory [12, 13], which was first applied to the study of multi-agent hybrid systems in [14]. In this framework, each aircraft treats every other aircraft involved in the potential collision as a disturbance. For a two-aircraft example, assuming a saddle solution to the game exists, each aircraft chooses an optimal policy assuming the worst possible disturbance or actions of the other aircraft. Game theoretic methods have been used in a similar way to prove that a set of maneuvers in Intelligent Vehicle Highway Systems is safe [15]. In cooperative conflicts, aircraft exchange sensor and intent information in order to predict and resolve the conflict. Coordination among the aircraft is in the form of maneuvers made up of sequences of flight modes with differential equations describing the continuous motion of the aircraft within each discrete mode. Both classes of conflict resolution problems are superb examples of hybrid systems. Both the implicit switching control law derived from the saddle solution to the game in the noncooperative class of conflicts, as well as the sequences of flight modes interacting with the continuous dynamics of the aircraft in the cooperative class of conflicts, may be modeled by finite automata with differential equations associated with each state, in order to verify the all-important safety properties of these conflict resolution schemes.

There are several approaches to hybrid system design and verification (see, for example, [16, 17, 18, 19]). One approach is to extend verification techniques which exist for finite state machines [20] to include timed and dynamical systems [21, 22] . These approaches abstract the differential equations by clocks [23] or differential inclusions ([24, 25, 26]) and verify the resulting abstracted system. More unified approaches in which the hybrid system design is not decoupled into the design of the individual continuous and discrete components are [27, 28, 29]. The specific problem of multi-agent systems is addressed in [30, 31]. We are inspired by the modeling scheme of [32], although we allow for more general continuous dynamics within each discrete state.

The organization of this paper is as follows: In Section 2 our conflict resolution strategy is presented, and in Section 3 our modeling formalism for hybrid systems is briefly described, with emphasis on the relative kinematic aircraft model used within this formalism. Section 4 presents the game theoretic approach to noncooperative conflict resolution while Section 5 describes the design and safety verification of coordinated maneuvers using cooperative methods.

2 Conflict Resolution Strategy

We present two classes of conflict resolution algorithms in this paper. The first, called noncooperative conflict resolution, assumes that each aircraft knows the current state of every other aircraft involved in a potential conflict, but that the intent of every other aircraft is unknown. This situation occurs when one of the aircraft, such as general aviation aircraft, may not be equipped with sophisticated avionics or when there is a communication malfunction. In this case, on-board sensing is still able to provide accurate state estimates about the intruding aircraft, possibly with a larger uncertainty. Also, bounds on the control variables of every other aircraft may be assumed. Using this information, each aircraft can then determine its own safe region of operation. The second class, called cooperative conflict resolution algorithms, assumes that the aircraft involved in a conflict are able to communicate intent with each other, so that they may follow an agreed upon maneuver which is proven a-priori to be safe. The communication between aircraft does not currently exist[1] but it will in the near future with the proposed Automatic Dependence Surveillance - Broadcast (ADS-B), in which each aircraft broadcasts to all other aircraft in its vicinity its current state as well as intent in the form of its next two proposed way points.

The algorithms of this paper do not require any additional structure to the airspace than what currently exists. Each aircraft is surrounded by two virtual *hockey pucks*, the *protected zone* and the *alert zone*, shown in Figure 1. A conflict or loss of separation between two aircraft occurs whenever the protected zones of the aircraft overlap. The radius and the height of the en-route protected zone is currently 5 nautical miles and 2,000 ft respectively.[2]. The size of the alert zone depends on various factors including airspeed, altitude, accuracy of sensing equipment, traffic situation, aircraft performance and average human and system response times. The alert zone should be large enough to allow a comfortable system response but also small enough in order to avoid unnecessary conflicts. For the algorithms in this paper, we assume a two-dimensional alert

[1] Although in emergency situations aircraft can currently communicate through an emergency radio frequency: this is what happened in the power failure at the Oakland Center Air Traffic Control in August, 1995, since communication with the ATC was lost.

[2] It has been proposed in [10] that for true 3−dimensional free flight, protected zones need to be balls of radius $3-5$ miles, but this is under debate still.

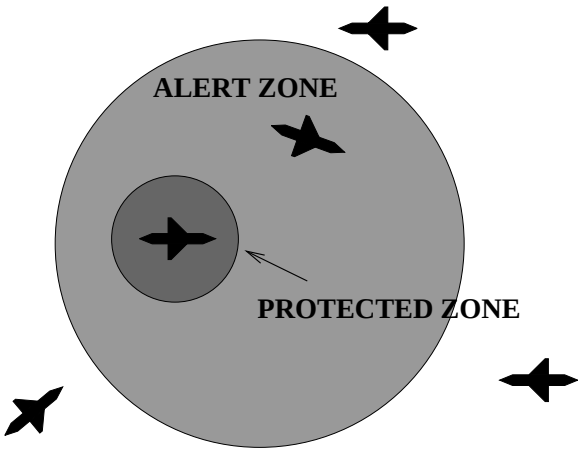

Figure 1: Aircraft Zones

zone although three dimensional extensions of the algorithms are conceptually similar.

We also assume that a *conflict prediction* algorithm exists and is used as a precursor to our conflict resolution schemes. Conflict prediction could be spatial, temporal or probabilistic. Spatial and temporal approaches, such as [8, 33], calculate the four dimensional coordinates of a possible conflict. Probabilistic approaches, such as [6, 7], assume stochastic uncertainty in the measured information and determine the probability of collision. In either case, if there is no (or low-probability) conflict, then the conflict resolution algorithms are not necessary.

The main thrust of both our noncooperative and cooperative conflict resolution algorithms is to calculate the safe region of operation of each aircraft. In the noncooperative algorithm, this corresponds to calculating the largest controlled invariant subset of the complement of the protected zones of the other aircraft involved in the conflict, for the worst possible actions of the other aircraft. Along with this safe set of states, we calculate the corresponding safe set of control inputs as a function of the state. The control scheme is called *least restrictive*, because within its safe region of operation, the aircraft may design its trajectory to optimize over other criteria, such as fuel efficiency or minimal deviation from route. A more detailed description of this methodology may be found in [34]. At the boundary of its safe region, the aircraft must apply the particular control which keeps it out of its unsafe region (the region in which it may come into conflict with the other aircraft). Thus, we are naturally led to a switching control based protocol. The resultant hybrid system is safe by design, as we illustrate with two versions of an interesting example of two aircraft conflict resolution in the horizontal plane.

In the cooperative algorithm, the safe region of operation corresponds to the set of safe initial conditions *for a given maneuver* with possible uncertainties in the states and control inputs. In this problem, the hybridness is *imposed* on the problem by the maneuver design, since the maneuver

involves a sequence of different modes of operation, each of which is modeled by a discrete state with associated continuous dynamics. Furthermore due to possible uncertainty, within each discrete state, a switching control law similar to the one derived in the noncooperative methodology applies.

3 Hybrid System Models for Aircraft

In this section, we present a minimal modeling formalism for hybrid systems that is sufficient for the conflict resolution problem. The hybrid model described below is inspired by that of [32] for linear hybrid automata, with the difference that we allow for a more general continuous dynamic model within each discrete state and a more general discrete transition relation without synchronization labels. More thorough and formal treatment may be found in [28, 35].

Hybrid System Model

A hybrid system H is defined to be the tuple $H = (Q \times M, I, Inv, E, F)$, in which:

1. $Q \times M$ is the state space, with $Q = \{q_1, q_2, \ldots, q_m\}$ a finite set of discrete states, and M an n-manifold; a state of the system is a pair $(q_i, x) \in Q \times M$;
2. $I_H \subset Q \times M$ is the set of initial conditions;
3. $Inv : Q \to 2^M$ is the set of invariants associated with each discrete state, meaning that the state (q, x) may flow within q only if $x \in Inv(q)$;
4. $E \subset Q \times M \times Q \times M$ is the set of jump conditions defined such that $(q, x, q', x') \in E$ means that when the current state is (q, x), the system may jump instantaneously to state (q', x');
5. $F : Q \to \chi(M)$ associates with each discrete state $q \in Q$ a vector field $f(q) : M \longrightarrow TM$.

Hybrid systems evolve in so-called "dense time" by either continuous flows or discrete transitions. Trajectories of hybrid system H starting at a state (q, x) evolve according to vector field $f(q)$ as long as the continuous state remains within $Inv(q)$. If the invariance condition is not satisfied then a discrete transition is forced and the continuous state may be reinitialized. If $(q, x, q', x') \in E$ then the discrete state jumps from q to q' and the continuous state x is reinitialized to x'.

Given a region $R \subset Q \times M$ of H, we define the region $Pre_H(R) \subset I_H$ as the set of initial conditions for which there exists a trajectory linking some initial state in I_H to some state in R.

Kinematic Models of Aircraft in Conflict

We now describe the continuous dynamics within each discrete state q, and the associated flows $f(q)$. Because conflicts between aircraft depend on the relative position and velocity of the agents, the continuous models we use are *relative* models, describing the motion of each aircraft in the system with respect to the other aircraft. For example, to study pairwise conflict between the trajectories of two aircraft, aircraft 1 and aircraft 2, a relative model with its origin centered on aircraft 1 is used. The configuration of an individual aircraft is described by an element of the Lie group G of rigid motions in $\mathbb{R}^2$ or $\mathbb{R}^3$, called $SE(2)$ or $SE(3)$ respectively. In planar situations, in which aircraft are flying at the same altitude, $SE(2)$ will be used.

Following the example described above, let $g_1 \in G$ denote the configuration of aircraft 1, and let $g_2 \in G$ denote the configuration of aircraft 2. The trajectories of both aircraft are kinematically modeled as left invariant vector fields on G. Therefore

$$\dot{g}_1 = g_1 X_1 \tag{1}$$

$$\dot{g}_2 = g_2 X_2 \tag{2}$$

where $X_1, X_2 \in \mathcal{G}$, the Lie algebra associated with the Lie group G.

A coordinate change is performed to place the identity element of the Lie group G on aircraft 1. Thus, let $g_r \in G$ denote the relative configuration of aircraft 2 with respect to aircraft 1. Then

$$g_2 = g_1 g_r \Rightarrow g_r = g_1^{-1} g_2 \tag{3}$$

Differentiation yields the dynamics of the relative configuration,

$$\dot{g}_r \;=\; g_r X_2 - X_1 g_r \tag{4}$$

Note that the vector field which describes the evolution of g_r is neither left nor right invariant. However,

$$\begin{aligned} \dot{g}_r &= g_r X_2 - X_1 g_r \\ &= g_r [X_2 - Ad_{g_r^{-1}} X_1] \end{aligned} \tag{5}$$

where $Ad_{g_r^{-1}} X_1 = g_r^{-1} X_1 g_r \in \mathcal{G}$, the Lie Algebra of the group G.

Consider the Lie group $SE(2)$ and its associated Lie algebra $se(2)$. A coordinate chart for $SE(2)$ is given by x, y, ϕ representing the planar position and orientation of a rigid body. In this coordinate chart, the relative configuration g_r is given in homogeneous coordinates by

$$g_r = \begin{bmatrix} \cos\phi_r & -\sin\phi_r & x_r \\ \sin\phi_r & \cos\phi_r & y_r \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

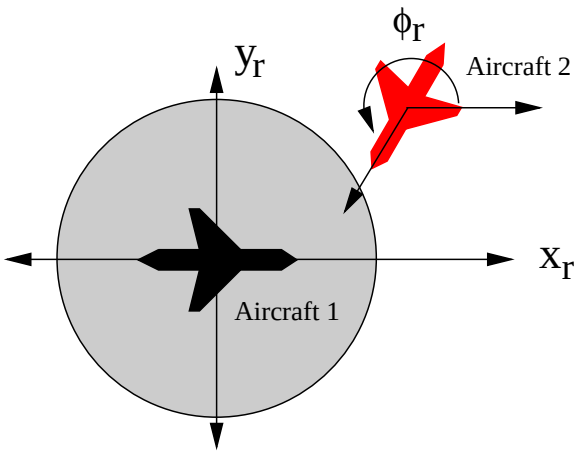

Figure 2: The relative configuration model

where x_r, y_r represent the relative position of aircraft 2 with respect to aircraft 1 and ϕ_r is the relative orientation. In local coordinates, the coordinate transformation (3) is expressed as

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = R(-\phi_1) \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} = \begin{bmatrix} \cos(-\phi_1) & -\sin(-\phi_1) \\ \sin(-\phi_1) & \cos(-\phi_1) \end{bmatrix} \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} \tag{7}$$

$$\phi_r = \phi_2 - \phi_1 \tag{8}$$

with x_i,y_i,ϕ_i parameterizing the absolute position and orientation of aircraft i. The Lie algebra elements $X_1, X_2 \in se(2)$ are represented as matrices in $\mathbb{R}^{3\times 3}$ of the form

$$X_1 = \begin{bmatrix} 0 & -\omega_1 & v_1 \\ \omega_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad X_2 = \begin{bmatrix} 0 & -\omega_2 & v_2 \\ \omega_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{9}$$

where v_i, ω_i represent the linear and angular velocities. Inserting equations (6) and (9) in equation (4) results in the following model

$$\begin{aligned} \dot{x}_r &= -v_1 + v_2 \cos\phi_r + \omega_1 y_r \\ \dot{y}_r &= v_2 \sin\phi_r - \omega_1 x_r \\ \dot{\phi}_r &= \omega_2 - \omega_1 \end{aligned} \tag{10}$$

illustrated in Figure 2. Similar results for $SE(3)$ may be found in [33].

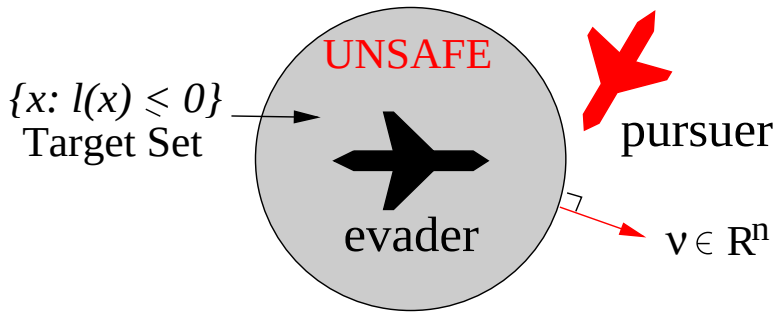

Figure 3: The evader and pursuer, with Target set and its outward pointing normal ν

4 Noncooperative Conflict Resolution

4.1 Design philosophy: game theoretic approach

First, we describe our noncooperative conflict resolution design philosophy on a general relative configuration model in $\mathbb{R}^n$. Consider the system

$$\dot{x} = f(x, u, d) \qquad x(t) = x \tag{11}$$

where $x \in \mathbb{R}^n$ describes the relative configuration of one of the aircraft with respect to the other, $u \in \mathcal{U}$ is the control input of one agent, and $d \in \mathcal{D}$ is the control of the other agent. We assume that the system starts at state x at initial time t. Both $\mathcal{U}$ and $\mathcal{D}$ are known sets, but whereas the control input u may be chosen by the designer, the disturbance d is unknown.

The goal is to maintain safe operation of the system (11), meaning that the system trajectories do not enter a prespecified unsafe region of the state space, called the Target set and denoted T with boundary ∂T. We assume that there exists a differentiable function $l(x)$ so that $T = \{x \in \mathbb{R}^n \mid l(x) \leq 0\}$ and $\partial T = \{x \in \mathbb{R}^n \mid l(x) = 0\}$. In this paper, T represents the protected zone around the aircraft at the origin of the relative axis frame (Figure 3).

Suppose that the two aircraft are conflict-prone, and they cannot cooperate to resolve conflict due to any one of the reasons mentioned in the previous section. Then the safest possible strategy of each aircraft is to fly a trajectory which guarantees that the minimum allowable separation with the other aircraft is maintained, *regardless* of the actions of the other aircraft. Since the intent of each aircraft is unknown to the other, then this strategy must be safe for the *worst possible actions* of the other aircraft. We formulate this problem as a two-person, zero-sum dynamical game of the pursuer-evader variety. Call the aircraft at the origin of the relative frame the *evader* with control input u, and the other aircraft the *pursuer* with control input d; the goal of the evader is to drive the system outside T whereas the worst possible action of the disturbance is to try to drive the system into T. We solve the dynamical game for system (11) over the time interval $[t, t_f]$, where

t_f is defined as

$$t_f = \inf\{\tau \in \mathbb{R}^+ \mid x(\tau) \in T\} \tag{12}$$

with initial state x at time t. If $t_f = \infty$, then for all possible control actions and disturbances the trajectory never enters T.

The game is a variational problem without a running cost, or Lagrangian: we are interested only in whether or not the state enters T. The cost $J_1(x,t,u,d)$ is therefore defined as a function (only) of the terminal state:

$$J_1(x,t,u,d) = l(x(t_f)) \tag{13}$$

Given $J_1(x,t,u,d)$, we first characterize the *unsafe* portion of ∂T, defined as those states $x \in \partial T$ for which there exists some disturbance $d \in \mathcal{D}$ such that for all inputs $u \in \mathcal{U}$ the vector field points into T; the *safe* portion of ∂T consists of the states $x \in \partial T$ for which there is some input $u \in \mathcal{U}$ such that for all disturbances $d \in \mathcal{D}$, the vector field points outward from T. More formally, we denote the outward pointing normal to T as

$$\nu = \frac{\partial l}{\partial x}(x(t_f)) \tag{14}$$

as in Figure 3 which allows us to define

$$\begin{array}{lll} \text{Safe portion of } \partial T & \{x \in \partial T : \exists u \forall d & \nu^T f(x,u,d) \geq 0\} \\ \text{Unsafe portion of } \partial T & \{x \in \partial T : \forall u \exists d & \nu^T f(x,u,d) < 0\} \end{array} \tag{15}$$

Given the above anatomy of ∂T, the game is won by the pursuer if the terminal state $x(t_f)$ belongs in the unsafe portion of the boundary, and is won by the evader otherwise. It is clear that the optimal control $u^* \in \mathcal{U}$ is the one which maximizes $J_1(x,t,u,d)$, and the worst disturbance $d^* \in \mathcal{D}$ is the one which minimizes $J_1(x,t,u,d)$:

$$u^* = \arg\max_{u \in \mathcal{U}} J_1(x,t,u,d) \tag{16}$$

$$d^* = \arg\min_{d \in \mathcal{D}} J_1(x,t,u,d) \tag{17}$$

The game is said to have a saddle solution if the cost $J_1^*(x,t)$ does not depend on the order in which the maximization and minimization is performed:

$$J_1^*(x,t) = \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} J_1(x,t,u,d) = \min_{d \in \mathcal{D}} \max_{u \in \mathcal{U}} J_1(x,t,u,d) \tag{18}$$

The concept of a saddle solution is key to our computation of the safe regions of operation of the aircraft, since a solution of (11) with $u = u^*$ and $d = d^*$ represents an optimal trajectory for *each* player under the assumption that the other player plays its optimal strategy.

Safety is maintained by operating within the *safe set* of states V_1, which is the largest subset of $\mathbb{R}^n \backslash T$ which can be rendered invariant using inputs $u \in \mathcal{U}$ regardless of the disturbance $d \in \mathcal{D}$. We

formally define V_1 as

$$\begin{aligned} V_1 &= \{x \in \mathbb{R}^n \backslash T \mid \exists u \in \mathcal{U}, J_1(x,t,u,d) \geq 0, \forall d \in \mathcal{D}\} \quad (19) \\ &= \{x \in \mathbb{R}^n \backslash T \mid \exists u \in \mathcal{U}, J_1(x,t,u,d^*) \geq 0\} \quad (20) \end{aligned}$$

Let ∂V_1 denote the boundary of V_1. At any instant t, the set $\{x \in \mathbb{R}^n \backslash T \mid J_1^*(x,t) \geq 0\}$ defines the set of safe states starting from time t. We would like to calculate the "steady state" safe set, or the safe set of states for all $t \in (-\infty, t_f]$. For this purpose we construct the Hamilton-Jacobi (Isaacs) equation for this system and attempt to calculate its steady state solution. Define the Hamiltonian $H(x,p,u,d) = p^T f(x,u,d)$ where $p \in T^*\mathbb{R}^n$ is the costate. The optimal Hamiltonian is given by:

$$H^*(x,p) = \max_{u \in U} \min_{d \in D} H(x,p,u,d) = H(x,p,u^*,d^*) \quad (21)$$

and satisfies Hamilton's equations (provided $H^*(x,p)$ is smooth in x and p):

$$\begin{aligned} \dot{x} &= \frac{\partial H^*}{\partial p}(x,p) \\ \dot{p} &= -\frac{\partial H^*}{\partial x}(x,p) \end{aligned} \quad (22)$$

with the boundary conditions $p(t_f) = \partial l(x(t_f))/\partial x$ and $x(t_f) \in \partial T$. If $J_1^*(x,t)$ is a smooth function of x and t, then $J_1^*(x,t)$ satisfies the Hamilton-Jacobi equation:

$$\frac{\partial J_1^*(x,t)}{\partial t} = -H^*(x, \frac{\partial J_1^*(x,t)}{\partial x}) \quad (23)$$

with boundary condition $J_1^*(x,t_f) = l(x(t_f))$. Our goal is to compute the safe set $V_1 = \{x \in \mathbb{R}^n \backslash T \mid J_1^*(x,-\infty) \geq 0\}$ where $J_1^*(x,-\infty)$ is the steady state solution of (23). However, it is difficult to guarantee that the PDE (23) has solutions for all $t \leq 0$, due to the occurrence of "shocks", ie. discontinuities in J as a function of x. If there are no shocks in the solution of (23), we may compute $J_1^*(x,-\infty)$ by setting the left hand side of the Hamilton-Jacobi equation to zero, thus $H^*(x, \frac{\partial J_1^*(x,-\infty)}{\partial x}) = 0$ which implies that $\frac{\partial J_1^*(x,-\infty)}{\partial x}$ is normal to the vector field $f(x,u^*,d^*)$.

To compute the safe set, we can propagate the boundaries of the safe set (those points for which $l(x) = 0$ and $H^*(x,p) = 0$) backwards in time, using the Hamilton-Jacobi (Isaacs) equation (23) to determine the safe and unsafe sets over the state space $\mathbb{R}^n$ (see Figure 4). Using the definitions of Section 3 we can think of V_1, the safe set of initial conditions, as $Pre_H(T)$, where in this case the hybrid automaton H has one discrete state and the continuous evolution is given by optimal flow (22). The solution of the Hamilton-Jacobi (Isaacs) equation provides $\partial Pre_H(T)$ which is important in trying to extend the Pre computation from continuous to hybrid systems.

The set V_1 defines the *least restrictive control scheme* for safety. If the pursuer is inside V_1, any control input may be safely applied by the evader, whereas on the boundary, the only input which may be safely applied to ensure safety is u^*. If the pursuer is inside the unsafe set, it will eventually

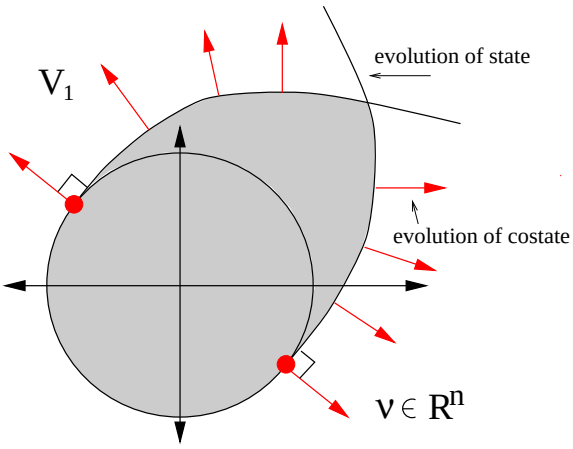

Figure 4: The unsafe set of states (shaded) and its complement (the safe set V_1)

end up in the target set regardless of the actions of the evader. The safe set of control inputs associated with each state $x \in V_1$ is

$$\mathcal{U}_1(x) \quad = \quad \{u \in \mathcal{U} \mid J_1(x,t,u,d) \geq 0, \forall d \in \mathcal{D}\} \tag{24}$$

Since all $u \in \mathcal{U}_1$ guarantee safety from state x, it is advantageous to find the optimal control policy $u \in \mathcal{U}_1$, for example the one that minimizes deviation from the nominal trajectory, which is encoded by a second cost function J_2, usually a quadratic function of the tracking error. To do this, we solve the optimal control problem which is *nested inside* the differential game calculation:

$$\min_{u \in \mathcal{U}_1} J_2 \tag{25}$$

subject to the original differential equations (1), (2) which describe the aircraft motion in absolute coordinates. Additional system requirements, such as *passenger comfort*, can now be incorporated by extending the above nested chain of games and optimal control problems following the multiobjective design methodology of [34].

4.2 Constant Altitude Conflict Resolution

In this section, we apply this general framework to conflicts which occur among aircraft at the same altitude. These conflicts are resolved either by speed variations or by path deviations. We therefore use the planar $SE(2)$ relative model (10) in local coordinates (x_r, y_r, θ_r).

Resolution by Angular Velocity

Let us first consider the case in which the linear velocities of both aircraft are fixed, $v_1, v_2 \in \mathbb{R}$, and the aircraft avoid conflict solely by using their angular velocities, thus $u = \omega_1$ and $d = \omega_2$, and

model (10) becomes:

$$\begin{aligned} \dot{x}_r &= -v_1 + v_2 \cos\phi_r + u y_r \\ \dot{y}_r &= v_2 \sin\phi_r - u x_r \\ \dot{\phi}_r &= d - u \end{aligned} \tag{26}$$

with state variables $x_r, y_r \in \mathbb{R}$, $\phi_r \in [-\pi, \pi)$, and control and disturbance inputs $u \in \mathcal{U} = [\underline{\omega}_1, \overline{\omega}_1] \subset \mathbb{R}$, $d \in \mathcal{D} = [\underline{\omega}_2, \overline{\omega}_2] \subset \mathbb{R}$. Without loss of generality (we scale the coefficients of u and d if this is not met), assume that $\underline{\omega}_i = -1$ and $\overline{\omega}_i = 1$, for $i = 1, 2$.

The target set T is the protected zone of the evader:

$$T = \{(x_r, y_r) \in \mathbb{R}^2, \phi_r \in [-\pi, \pi) \mid x_r^2 + y_r^2 \leq 5^2\} \tag{27}$$

which is a 5-mile-radius cylinder in the (x_r, y_r, ϕ_r) space. Thus the function $l(x)$ may be defined as

$$l(x) = x_r^2 + y_r^2 - 5^2 \tag{28}$$

The optimal Hamiltonian is

$$H^*(x,p) = \max_{u \in U} \min_{d \in D} [-p_1 v_1 + p_1 v_2 \cos\phi_r + p_2 v_2 \sin\phi_r + (p_1 y_r - p_2 x_r - p_3)\, u + p_3 d] \tag{29}$$

Defining the *switching functions* $s_1(t)$ and $s_2(t)$, as

$$\begin{aligned} s_1(t) &= p_1(t) y_r(t) - p_2(t) x_r(t) - p_3(t) \\ s_2(t) &= p_3(t) \end{aligned} \tag{30}$$

the saddle solution u^*, d^* exists when $s_1 \neq 0$ and $s_2 \neq 0$ and are calculated as

$$\begin{aligned} u^* &= sgn(s_1) \\ d^* &= -sgn(s_2) \end{aligned} \tag{31}$$

The equations for $\dot{p}$ are obtained through (22) and are

$$\begin{aligned} \dot{p}_1 &= u^* p_2 \\ \dot{p}_2 &= -u^* p_1 \\ \dot{p}_3 &= p_1 v_2 \sin\phi_r - p_2 v_2 \cos\phi_r \end{aligned} \tag{32}$$

with $p(t_f) = (x_r, y_r, 0)^T = \nu$, the outward pointing normal to ∂T at any point (x_r, y_r, ϕ_r) on ∂T.

The safe and unsafe portions of ∂T are calculated using equations (15) with $\nu = (x_r, y_r, 0)^T$. Thus, those (x_r, y_r, ϕ_r) on ∂T for which

$$-v_1 x_r + v_2 (x_r \cos\phi_r + y_r \sin\phi_r) < 0 \tag{33}$$

constitute the unsafe portion, and those (x_r, y_r, ϕ_r) on ∂T for which

$$-v_1 x_r + v_2 (x_r \cos\phi_r + y_r \sin\phi_r) = 0 \tag{34}$$

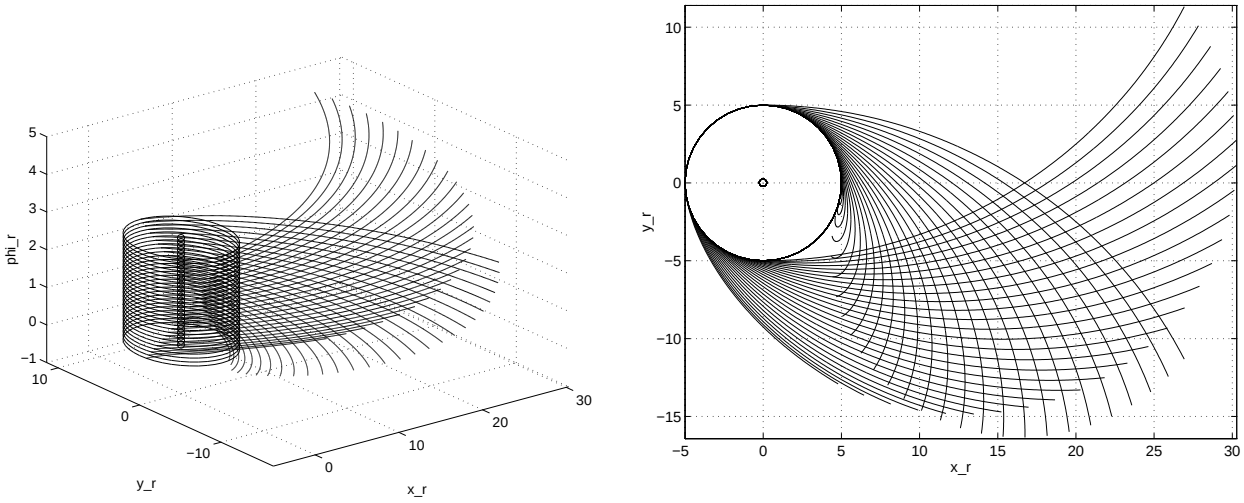

Figure 5: The Target set $T = \{(x_r, y_r), \phi_r \in (0, \pi) \mid x_r^2 + y_r^2 \leq 5^2\}$ (cylinder) and the boundary of the safe set V_1 for $t \leq t_f$ until the first switch in either $s_1(t)$ or $s_2(t)$. The unsafe set is enclosed by the boundary. The second picture is a top view of the first.

are the final state conditions for the boundary of the safe set V_1. To solve for $p(t)$ and $x(t)$ along this boundary for $t < t_f$, we must first determine $u^*(t_f)$ and $d^*(t_f)$. Equations (31) are not defined at $t = t_f$, since $s_1 = s_2 = 0$ on ∂T, giving rise to "abnormal extremals" (meaning that the optimal Hamiltonian loses dependence on u and d at these points. Analogously to [12] (Chapter 8), we use an indirect method to calculate $u^*(t_f)$ and $d^*(t_f)$: at any point (x_r, y_r, ϕ_r) on ∂T, the derivatives of the switching functions s_1 and s_2 are

$$\dot{s}_1 = y_r v_1 \tag{35}$$

$$\dot{s}_2 = x_r v_2 \sin\phi_r - y_r v_2 \cos\phi_r \tag{36}$$

For example, for points $(x_r, y_r, \phi_r) \in \partial T$, such that $\phi_r \in (0, \pi)$, it is straightforward to show that $\dot{s}_1 > 0$ and $\dot{s}_2 > 0$, meaning that for values of t slightly less than t_f, $s_1 < 0$ and $s_2 < 0$. Thus for this range of points along ∂T, $u^*(t_f) = -1$ and $d^*(t_f) = 1$. These values for u^* and d^* remain valid for $t < t_f$ as long as $s_1(t) < 0$ and $s_2(t) < 0$. When $s_1(t) = 0$ and $s_2(t) = 0$, the saddle solution switches and the computation of the boundary continues with the new values of u^* and d^*, thus introducing "kinks" into the safe set boundary. These points correspond to the shocks in the Hamilton-Jacobi (Isaacs) equation discussed above. Figure 5 displays the resulting boundary of the safe set V_1, for $t < t_f$ until the first time that either $s_1(t)$ or $s_2(t)$ switches.

The automaton illustrating the *least restrictive control scheme* for safety is shown in Figure 6. The computation of the boundary of V_1 is in general difficult. For certain ranges of $\mathcal{U}$ and $\mathcal{D}$, the surfaces shown in Figure 5 intersect. At the intersection, it is not clear that the property of u^* being the

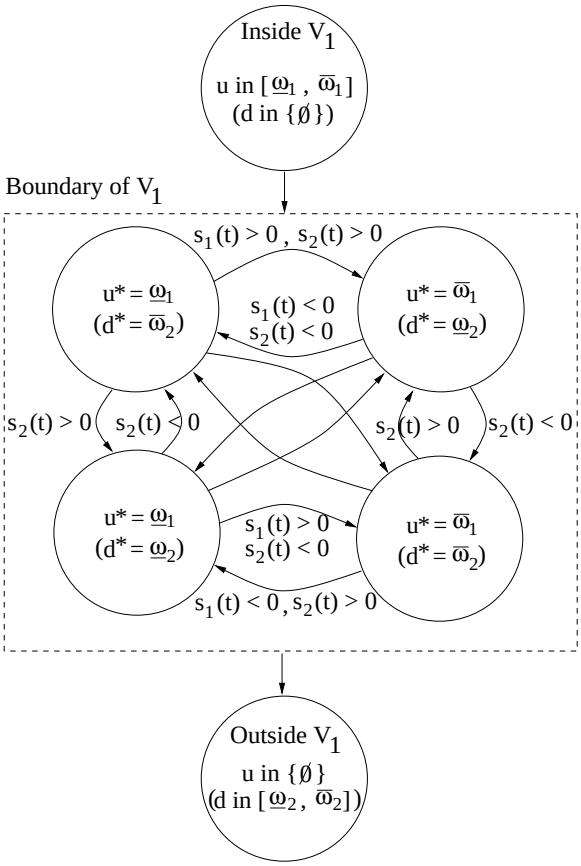

Figure 6: Switching law governing the two aircraft system with angular velocity control inputs. The law is least restrictive in that the control u is not restricted when the state is inside V_1. The diagonal transitions in the automaton for the boundary of V_1 are not labeled for legibility. This automaton may be thought of as the composition of two switching automata, one each for the pursuer and evader. The individual switching automaton for u is easily derived by neglecting transitions for d, and conversely.

unique safe input in maintained.

Resolution by Linear Velocity

We now consider the case in which the angular velocities of the two aircraft are zero, and collision is avoided by altering the velocity profile of the trajectories. Thus, $u = v_1$, $d = v_2$, and model (10) reduces to:

$$\begin{aligned} \dot{x}_r &= -u + d\cos\phi_r \\ \dot{y}_r &= d\sin\phi_r \\ \dot{\phi}_r &= 0 \end{aligned} \tag{37}$$

The input and disturbance lie in closed subsets of the positive real line $u \in \mathcal{U} = [\underline{v}_1, \overline{v}_1] \subset \mathbb{R}^+$, $d \in \mathcal{D} = [\underline{v}_1, \overline{v}_2] \subset \mathbb{R}^+$.

The Target set T and function $l(x)$ are defined as in the previous example. In this example, it is straightforward to calculate the saddle solution (u^*, d^*) directly, by integrating equations (37) for piecewise constant u and d, and substituting the solutions into the cost function (13). To do this we first define the switching functions s_1 and s_2 as

$$\begin{aligned} s_1(t) &= x_r \\ s_2(t) &= x_r\cos\phi_r + y_r\sin\phi_r \end{aligned} \tag{38}$$

Proposition 1 [Saddle Solution for Linear Velocity Controls] *The global saddle solution* (u^*, d^*) *to the game described by system (37) for the cost* $J_1(x, t, u, d)$ *given by equation (13) is*

$$u^* = \begin{cases} \underline{v}_1 & \text{if } sgn(s_1) > 0 \\ \overline{v}_1 & \text{if } sgn(s_1) < 0 \end{cases} \tag{39}$$

$$d^* = \begin{cases} \underline{v}_2 & \text{if } sgn(s_2) > 0 \\ \overline{v}_2 & \text{if } sgn(s_2) < 0 \end{cases} \tag{40}$$

Proof: In Appendix. □

As can be seen from equation (39), the optimal speed of the evader depends on the position of the pursuer relative to the evader. If the pursuer is ahead of the evader in the relative axis frame, then u^* is at its lower limit, if the pursuer is behind the evader in the relative axis frame then u^* is at its upper limit. If the pursuer is heading towards the evader, then d^* is at its upper limit, if the pursuer is heading away from the evader, d^* is at its lower limit. The bang-bang nature of the saddle solution allows us to abstract the system behavior by the hybrid automaton shown in Figure 7, which describes the least restrictive control scheme for safety. The unsafe sets of states are illustrated in Figure 8 for various values of ϕ_r, and speed ranges as illustrated.

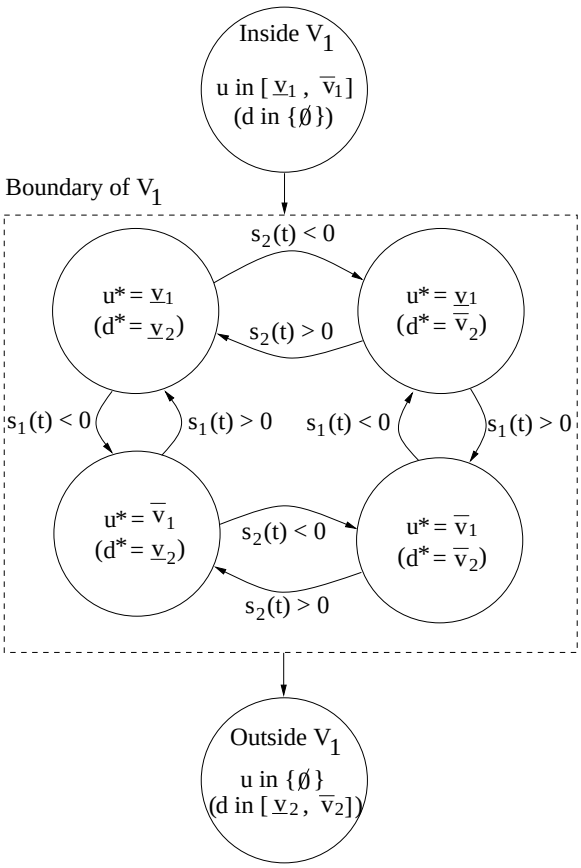

Figure 7: Switching law governing the two aircraft system with linear velocity control inputs. The note in the caption of the automaton of the previous example, about the composition of automata for u and d, applies here also.

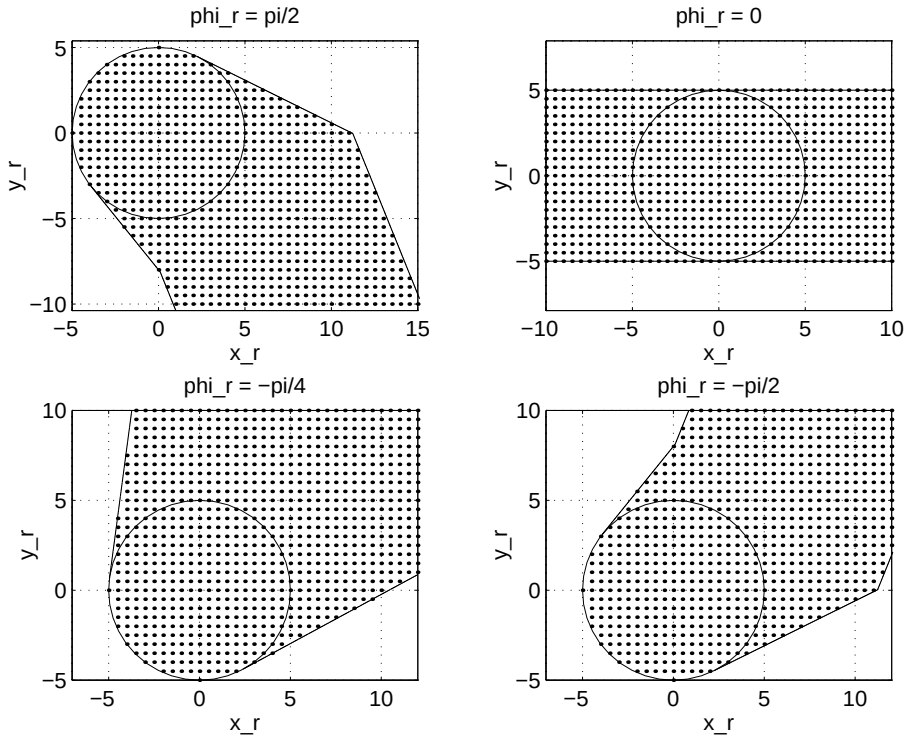

Figure 8: Unsafe sets (x_r, y_r) for $[\underline{v}_1, \overline{v}_1] = [2, 4]$, $[\underline{v}_2, \overline{v}_2] = [1, 5]$ and $\phi_r = \pi/2$, 0, $-\pi/4$, and $-\pi/2$.

5 Cooperative Conflict Resolution

5.1 Design Philosophy

This section addresses the problem of cooperative conflict resolution among aircraft. In cooperative methods, aircraft exchange state and intent information and perform coordinated maneuvers in order to resolve conflicts. The form of the maneuver is chosen to be a finite sequence of heading changes resulting in a trapezoidal deviation from the desired path. These maneuvers are routinely used in current Air Traffic Control practice since they are easily understandable by pilots as well as easily implementable by current on-board auto-pilots which regulate the aircraft to heading and speed setpoints. This will be technologically feasible in the near future using datalink and Automatic Dependence Surveillance - Broadcast (ADS-B). We will therefore assume that each aircraft has access to the state and intent aircraft within its alert zone. However, due to wind, sensor and modeling uncertainty, the broadcasted information regarding other aircraft may not be exact but may lie within a bounded interval.

Our approach to cooperative conflict resolution is to develop a set of maneuvers which is proven or verified a-priori to be safe and covers all possible conflict scenarios within the alert zone. Initially, a maneuver is constructed and is modeled as a hybrid automaton H using the formalism of Section

3. The form of the maneuver is inspired by the potential field techniques described in [36]. The state space of H is $Q \times \mathbb{R}^2 \times S^1$ where Q is the discrete state space which models the different segments, or flight modes, in the maneuver and $(x_r, y_r, \phi_r) \in \mathbb{R}^2 \times S^1$ are the continuous variables which evolve within each discrete location according to the relative configuration dynamics (10).

The specification for the hybrid system H is the safety requirement that the protected zones of the two aircraft never intersect. More formally, the safe region $T \subset Q \times \mathbb{R}^2 \times S^1$ of the hybrid automaton H is defined as

$$T = Q \times \{(x_r, y_r, \phi_r) \in \mathbb{R}^2 \times S^1 \mid x_r^2 + y_r^2 > r_p^2\} \tag{41}$$

where r_p miles is the minimum allowable loss of separation.

Given the hybrid automaton H describing a coordinated maneuver and the safe region T, we compute $Pre_H(T) \subset I_H$, the set of initial conditions of H for which this maneuver is safe. This procedure is repeated until we obtain a finite number of maneuvers H_i, $1 \leq i \leq n$ such that

$$\{(x_r, y_r, \phi_r) \in \mathbb{R}^2 \times S^1 \mid r_a^2 \geq x_r^2 + y_r^2 \geq r_p^2\} \subseteq \bigcup_{1 \leq i \leq n} Pre_{H_i}(T) \tag{42}$$

or otherwise when all the initial conditions within the alert zone (with radius r_a) but outside the protected zone (with radius r_p) are partitioned or covered by the safe set of a finite number of maneuvers. This will generate a database of coordinated maneuvers covering all possible conflict scenarios. The database can be computed off-line and can be retrieved efficiently on-line by the conflicting aircraft.

A maneuver which deserves special consideration is to simply stay on course. Given the uncertain measurements this problem can be thought of as a special case of the game theoretic methods described in Section 4. In this case one does not think of aircraft as noncooperative but as cooperating with uncertain information[3]. We use the methods of Section 4 to determine the worst possible disturbance which is then used to determine the safety of the maneuver.

The computation of the safe set of initial conditions given a hybrid automaton and a safety specification is a very important and challenging problem. Verification tools for hybrid systems such as HyTech [21] and Kronos [22] computationally attack this problem for linear hybrid automata, a decidable class of hybrid systems. In [36], under some simplifying assumptions on the coordinated maneuvers, these computational tools are used to symbolically calculate the safe set of initial conditions. The methodology of Section 4 is an elegant way to determine the boundary of the safe set of states of the continuous system within each discrete state, by solving the Hamilton-Jacobi (Isaacs) equation.

[3]Clearly the uncertainty is much smaller in this case.

Conceptually, the methodology of this section is an extension of the technique of the previous section to hybrid system models. Thus, we use the Hamilton-Jacobi formalism inside each discrete or "marked" state along with possible switches and resets allowed at transitions among the marked states to compute safe sets. Thus, our approach is in the spirit of the calculations of [25]. We note that in this paper and more generally in the model checking literature on hybrid systems verification, the accent is on finding computational bounds on the complexity of the calculations involved in performing the Pre computation to calculate the safe sets. It has been shown in particular that safe sets can be computed for certain classes of timed and rectangular automata. Our approach allows for more general nonlinear dynamics in the marked states. We have, however, not as yet completed a full study of computational issues of numerical solutions of the Hamilton-Jacobi (Isaacs) equations and of an efficient representation of the safe sets. These topics will be covered in future work.

5.2 Protocol for Two Aircraft

The above design philosophy is now illustrated on a conflict resolution protocol for two aircraft. We assume that only two aircraft are in conflict, the aircraft are initially in cruise mode flying at constant headings and velocities. Furthermore, each aircraft is aware of the position, orientation as well as velocity of the other aircraft. The position and orientation sensing is exact; the velocity of the other aircraft is known to within bounds due to uncertainty.

Consider the conflict scenario and resolution maneuver shown in Figure 9. The protocol may be linguistically expressed as follows:

1. Cruise until aircraft are 10 miles apart
2. Make a heading change of $\Delta\phi$ until a lateral displacement of d miles is achieved.
3. Cruise until aircraft are more than 10 miles away
4. Make a heading change of $-\Delta\phi$ until a lateral displacement of -d miles is achieved.
5. Return to original heading and cruise

The above protocol can be modeled as the hybrid automaton shown in Figure 10. The discrete states, $Q = \{CRUISE, LEFT, STRAIGHT, RIGHT\}$, simply model the discrete heading changes which occur simultaneously on both aircraft. Within each discrete state, the continuous variables $(x_r, y_r) \in \mathbb{R}^2$ of the aircraft flow using the relative model (10) with fixed relative orientation ϕ_r. Due to the uncertain velocity of the other aircraft, the worst case scenario is assumed for v_2 and therefore the dynamics evolve according to the saddle solution (40). This introduces additional

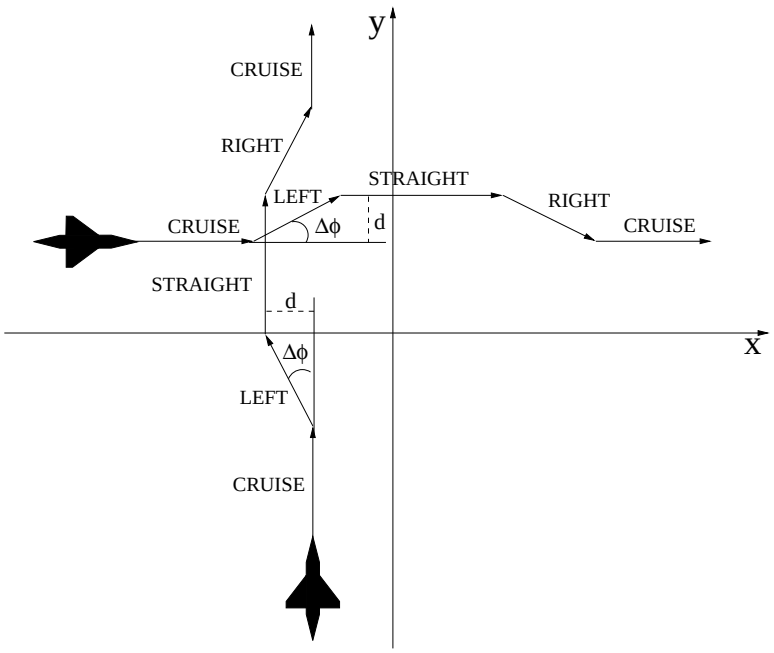

Figure 9: Conflict resolution maneuver

switching surfaces within each discrete state. The initial condition of the automaton is $I_H = CRUISE \times \{x_r^2 + y_r^2 \geq r_p^2\}$, that is the aircraft are assumed to be initially cruising and the protected zones do not intersect.

The automaton of Figure 10 starts in the $CRUISE$ mode and flows in that state until the inter-aircraft distance is less than 10 miles, at which point both aircraft make a heading change of $\Delta\phi$. Discrete heading changes have the effect of reseting the initial condition by a rotation matrix since the coordinate frame depends on the orientation of the aircraft (7,8). In mode $LEFT$, both aircraft wish to make a nominal lateral displacement of d. This is achieved by using a timer variable t which counts up to the time needed to achieve this. At that point both aircraft return to their original heading and cruise until the distance between them is greater than 10 miles. Once this is achieved, the reverse maneuver is performed in order return to the original cruise path and heading.

Figure 11 displays computations of $Pre_H(T)$ for the maneuver shown in Figure 10. The parameters were chosen to be $v_1 = 3$, $v_2 \in [2, 4]$, $\phi_r = 90^o$ and the size of the protected zone was 2.5 miles while the alert zone had a radius of 20 miles[4]. The upper left figure displays the unsafe set assuming no maneuver is performed. This is a special case of resolution by linear velocity described in Section 4 with v_1 known and v_2 belonging in an interval. The upper right figure displays Maneuver 1, the coordinated maneuver described above with maneuver parameters $\Delta\phi = 45^o$ and $d = 2.5$. The lower two figures display Maneuvers 2 and 3 which are geometrically similar to maneuver 1 but

[4]The sizes of the zones were scaled in order to produce visualizable figures

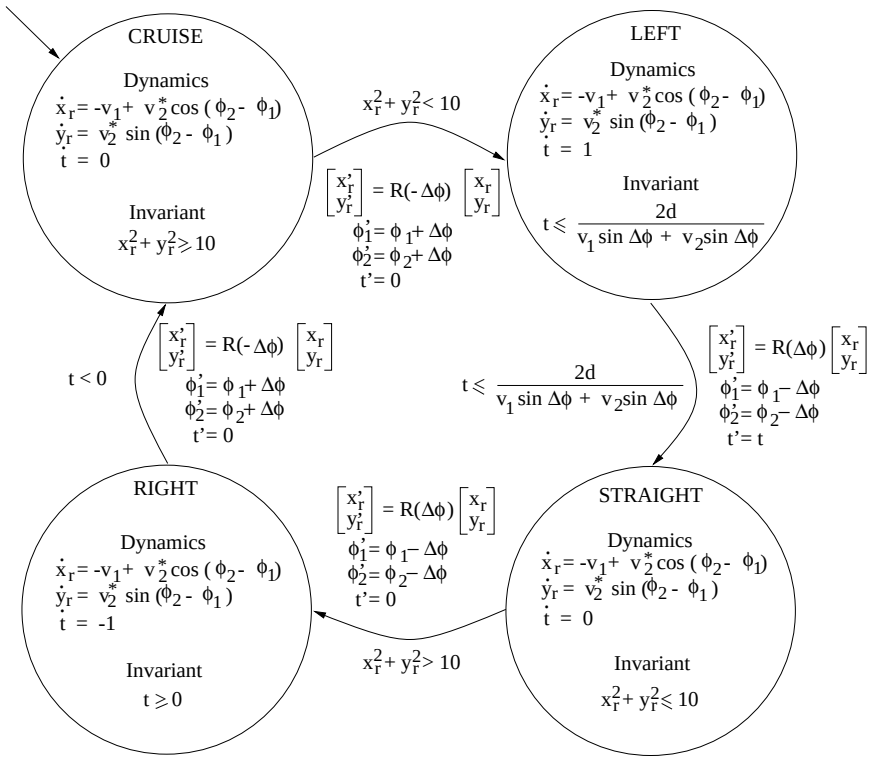

Figure 10: Modeling of Conflict resolution maneuver as a Hybrid Automaton

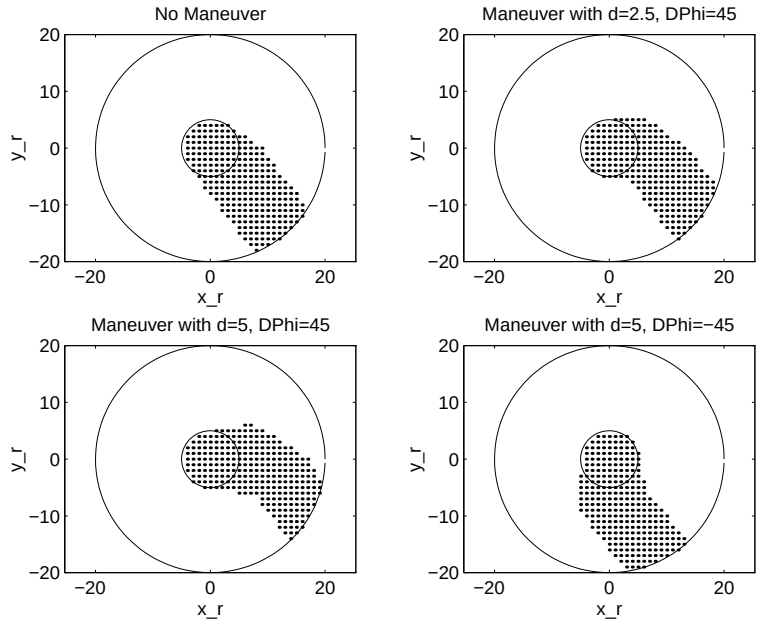

Figure 11: Unsafe Sets for Various Maneuvers. Upper left: no maneuver; upper right: Maneuver 1; lower left: Maneuver 2; lower right: Maneuver 3.

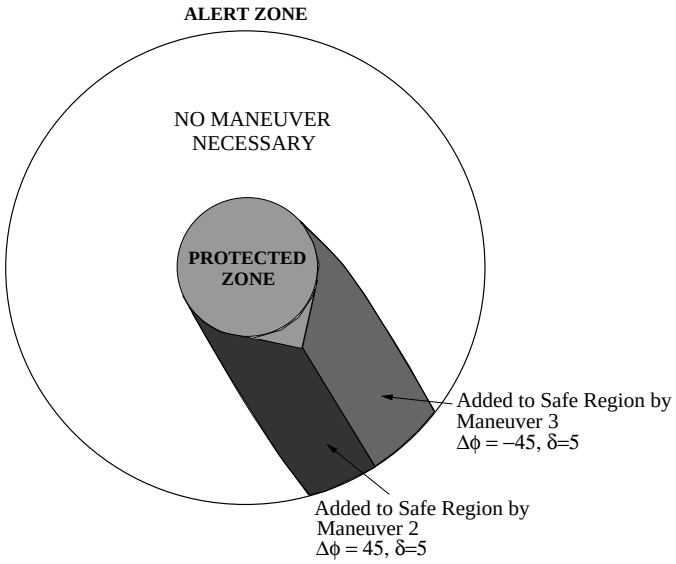

Figure 12: State Space Partitioning using Maneuver Safe Sets

have parameters $d = 5$, $\Delta\phi = 45^o$ and $d = 5$, $\Delta\phi = -45^o$ respectively. The union of the safe sets for the four maneuvers shown in Figure 11 covers almost all of the initial conditions x_r, y_r in the alert zone for $\phi_r = 90^o$. This is shown in Figure 12 which displays which maneuver the coordinating aircraft should perform in order to safely resolve the conflict.

The computations described in this section are easily performed for all values of ϕ_r as well as other velocity ranges. In addition, more complicated protocols such as asymmetric or unsynchronized maneuvers are dealt with using the same methodology.

6 Conclusions

In this paper, a conflict resolution strategy for free flight based Air Traffic Management is presented. Both cooperative and noncooperative methods are described, and two examples are presented to illustrate the methodology. Noncooperative conflict resolution methods are developed using game theory; cooperative methods are developed by modeling and verifying the safety of maneuvers using hybrid automata.

The primary focus of this paper has been on 2−aircraft conflict resolution. Our current research addresses conflict resolution for multiple (more than 2) aircraft. One approach to the design of protocols for conflict resolution is described in [36]. Here, we use artificial potential field methods from robotic path planning to produce conflict-free maneuvers, for given scenarios. The resulting protocols are verified and safe sets calculated using the techniques of Sections 4 and 5 above. As an example of the sort of maneuver produced by this procedure, consider the case of three aircraft

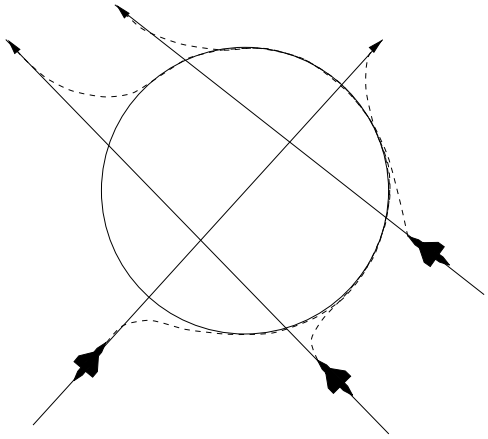

Figure 13: Conflict Resolution for three aircraft: the Roundabout maneuver

coming into conflict as shown in Figure 13. Our approach has generated the *Roundabout* maneuver illustrated in this figure. The aircraft are restricted to fly along the circular path segments with a given speed, as not to overtake the other aircraft already involved in the maneuver. An aircraft may not enter the Roundabout until the other aircraft are outside of its protected zone. Verification of the safety of the Roundabout maneuver, using the cooperative methodology described in this paper, is currently underway.

Finally, we are extending the verification methodology of this paper to dynamic rather than kinematic models of the aircraft, with different flight modes. In a hierarchical flight vehicle management system, such as the one proposed in [11], we have explained the role of dynamic models. Some preliminary results in this context have been presented in [26], [34].

7 Appendix

Proof of Proposition 1: Starting at time t (free) and integrating to the final time t_f, the solution to equations (37) has $\phi_r(t) = \phi_r(0)$ and

$$\begin{array}{ll} x_r(t_f) & = x_r(t) - \int_t^{t_f} u(\tau)d\tau + \cos\phi_r \int_t^{t_f} d(\tau)d\tau \\ y_r(t_f) & = y_r(t) + \sin\phi_r \int_t^{t_f} d(\tau)d\tau \end{array} \tag{43}$$

Substituting equations (43) into the cost index (13), (28), and ignoring the constant 5^2 results in

$$\begin{aligned} J_1(x,t,u,d) &= x_r^2(t_f) + y_r^2(t_f) \\ &= x_r^2(t) + y_r^2(t) - x_r(t)\int_t^{t_f} u(\tau)d\tau - x_r(t_f)\int_t^{t_f} u(\tau)d\tau \\ &\quad + \int_t^{t_f} d(\tau)d\tau[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] + \int_t^{t_f} d(\tau)d\tau[x_r(t_f)\cos\phi_r + y_r(t_f)\sin\phi_r] \end{aligned}$$

Define the *switching functions* $s_1(t), s_2(t)$ as in equations (38). Consider the case in which, $\forall t \leq t_f$,

$$sgn(s_1(t)) > 0, \quad sgn(s_2(t)) > 0$$

We will show that in this case the saddle solution is $u^* = \underline{v}_1$ and $d^* = \underline{v}_2$. Note that we assume that in the interval $[t, t_f]$, both $s_1(t)$ and $s_2(t)$ *do not change sign.* If t is such that the switching functions do change sign on this interval, then the interval must be broken into two intervals, and the saddle solution calculated separately for each interval.

Let $d = d^*$ and vary u, ie. let $u = \underline{v}_1 + \delta v_1$, where $\delta v_1 \geq 0$. Then

$$\begin{aligned} J_1(x,t,u,d^*) &= x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(t_f - t) - x_r(t_f)\underline{v}_1(t_f - t) - x_r(t)\int_t^{t_f} \delta v_1(\tau)d\tau - x_r(t_f)\int_t^{t_f} \delta v_1(\tau)d\tau \\ &\quad + \underline{v}_2(t_f - t)[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] + \underline{v}_2(t_f - t)[x_r(t_f)\cos\phi_r + y_r(t_f)\sin\phi_r] \\ &\leq x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(t_f - t) - x_r(t_f)\underline{v}_1(t_f - t) \\ &\quad + \underline{v}_2(t_f - t)[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] + \underline{v}_2(t_f - t)[x_r(t_f)\cos\phi_r + y_r(t_f)\sin\phi_r] \\ &= J_1(x,t,u^*,d^*) \end{aligned} \tag{44}$$

Similarly, let $u = u^*$ and vary d, ie. let $d = \underline{v}_2 + \delta v_2$, where $\delta v_2 \geq 0$. Then

$$\begin{aligned} J_1(x,t,u^*,d) &= x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(t_f - t) - x_r(t_f)\underline{v}_1(t_f - t) \\ &\quad + \underline{v}_2(t_f - t)[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] + \underline{v}_2(t_f - t)[x_r(t_f)\cos\phi_r + y_r(t_f)\sin\phi_r] \\ &\quad + \int_t^{t_f} \delta v_2(\tau)d\tau[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] + \int_t^{t_f} \delta v_2(\tau)d\tau[x_r(t_f)\cos\phi_r + y_r(t_f)\sin\phi_r] \\ &\geq x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(t_f - t) - x_r(t_f)\underline{v}_1(t_f - t) \\ &\quad + \underline{v}_2(t_f - t)[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] + \underline{v}_2(t_f - t)[x_r(t_f)\cos\phi_r + y_r(t_f)\sin\phi_r] \\ &= J_1(x,t,u^*,d^*) \end{aligned} \tag{45}$$

Summarizing, we have shown above that in this case,

$$J_1(x,t,u,d^*) \leq J_1(x,t,u^*,d^*) \leq J_1(x,t,u^*,d) \tag{46}$$

Therefore, $u^* = \underline{v}_1$, $d^* = \underline{v}_2$ is a saddle solution in this case. The three other cases can be shown in a similar manner. □

References

[1] Radio Technical Commission for Aeronautics. Final report of RTCA task force 3: Free flight implementation. Technical report, Washington DC, October 1995.

[2] Honeywell Inc. Technology and Procedures Report. Technical Report NASA Contract NAS2-114279, 1996.

[3] Honeywell Inc. Concepts and Air Transportation Systems Report. Technical Report NASA Contract NAS2-114279, 1996.

[4] W. H. Harman. TCAS : A system for preventing midair collisions. *The Lincoln Laboratory Journal*, 2(3):437–457, 1989.

[5] H. Erzberger. CTAS : Computer intelligence for air traffic control in the terminal area. Technical Report NASA TM-103959, NASA Ames Research Center, Moffett Field, CA, July 1992.

[6] L. Yang and J. Kuchar. Prototype conflict alerting logic for free flight. In *Proceedings of the 35th AIAA Aerospace Sciences Meeting & Exhibit*, Reno, NV, January 1997.

[7] Russell A. Paielli and Heinz Erzberger. Conflict probability and estimation for free flight. In *Proceedings of the 35th Meeting of the American Institute of Aeronautics and Astronautics, AIAA-97-0001*, Reno, 1997.

[8] J. Krozel, T. Mueller, and G. Hunter. Free flight conflict detection and resolution analysis. In *Proceedings of the American Institute of Aeronautics and Astronautics Guidance Navigation and Control Conference, AIAA-96-3763*, 1996.

[9] Yiyuan Zhao and Robert Schultz. Deterministic resolution of two aircraft conflict in free flight. In *Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA-97-3547*, New Orleans, LA, August 1997.

[10] J.-H. Oh and E. Feron. Fast detection and resolution of multiple conflicts for 3-Dimensional free flight. Submitted to the 36th IEEE Conference on Decision and Control, 1997.

[11] C. Tomlin, G. Pappas, J. Lygeros, D. Godbole, and S. Sastry. Hybrid control models of next generation Air Traffic Management. In P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems IV*, Lecture Notes in Computer Science, New York, 1997. Springer Verlag.

[12] T. Başar and G. J. Olsder. *Dynamic Non-cooperative Game Theory.* Academic Press, second edition, 1995.

[13] Joseph Lewin. *Differential Games.* Springer-Verlag, 1994.

[14] John Lygeros, Datta Godbole, and Shankar Sastry. A game theoretic approach to hybrid system design. Technical report, UCB-ERL Memo M95/77, Electronics Research Laboratory, University of California, Berkeley, CA 94720, 1995.

[15] John Lygeros, Datta N. Godbole, and Shankar Sastry. A verified hybrid controller for automated vehicles. To appear in the IEEE Transactions on Automatic Control, Special Issue on Hybrid Systems, 1997.

[16] Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors. *Hybrid Systems.* Lecture Notes in Computer Science. Springer-Verlag, 1993.

[17] Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors. *Hybrid Systems II.* Lecture Notes in Computer Science. Springer-Verlag, 1995.

[18] R. Alur, T.A. Henzinger, and E.D. Sontag, editors. *Hybrid Systems III.* Lecture Notes in Computer Science 1066. Springer-Verlag, 1996.

[19] P Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors. *Hybrid Systems IV.* Lecture Notes in Computer Science. Springer-Verlag, 1997.

[20] R. P. Kurshan. *Computer Aided Verification of Coordinating Processes: The Automata Theoretic Approach.* Princeton University Press, Princeton, N. J., 1994.

[21] Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. *A User Guide to HyTech.* Department of Computer Science, Cornell University, 1996.

[22] C. Daws, A. Olivero, and S. Yovine. Verifying ET-LOTOS programs with KRONOS. In D. Hogrefe and S. Leue, editors, *Proc. 7th. IFIP WG G.1 International Conference of Formal Description Techniques, FORTE'94*, pages 227–242, Bern, Switzerland, October 1994. Formal Description Techniques VII, Chapman & Hall.

[23] Rajeev Alur and David Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[24] T.A. Henzinger. Hybrid automata with finite bisimulations. In Z. Fülöp and F. Gécseg, editors, *ICALP 95: Automata, Languages, and Programming*, Lecture Notes in Computer Science 944, pages 324–335. Springer-Verlag, 1995.

[25] Anuj Puri and Pravin Varaiya. Decidability of hybrid systems with rectangular differential inclusions. In *Computer Aided Verification*, pages 95–104, 1994.

[26] George J. Pappas and Shankar Sastry. Towards continuous abstractions of dynamical and control systems. In P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems IV*, Lecture Notes in Computer Science. Springer Verlag, New York, 1997.

[27] R.W. Brockett. Hybrid models for motion control systems. In H. Trentelman and J.C. Willems, editors, *Perspectives in Control*, pages 29–54. Birkhauser, Boston, 1993.

[28] Michael S. Branicky. *Studies in Hybrid Systems: Modeling, Analysis and Control.* PhD thesis, Massachusetts Institute of Technology, 1995.

[29] Akash Deshpande. *Control of Hybrid Systems.* PhD thesis, Department of Electrical Engineering, University of California, Berkeley, California, 1994.

[30] A. Nerode and W. Kohn. Multiple agent hybrid control architecture. In Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors, *Hybrid Systems*, pages 297–316. Springer Verlag, New York, 1993.

[31] W. Kohn, A. Nerode, and J. Remmel. Hybrid systems as Finsler manifolds: Finite state control as approximation to connections. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems II*, pages 294–321. Springer Verlag, New York, 1995.

[32] T.A. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, pages 278–292. IEEE Computer Society Press, 1996.

[33] C. Tomlin, G. Pappas, and S. Sastry. Conflict resolution for air traffic management: A case study in multi-agent hybrid systems. Technical report, UCB/ERL M96/38, Electronics Research Laboratory, University of California, Berkeley, 1996.

[34] J. Lygeros, C. Tomlin, and S. Sastry. Multiobjective hybrid controller synthesis. In *Springer-Verlag Proceedings of the International Workshop on Hybrid and Real-Time Systems*, pages 109–123, Grenoble, 1997.

[35] John Lygeros. *Hierarchical, Hybrid Control of Large Scale Systems.* PhD thesis, University of California at Berkeley, 1996.

[36] Jana Košecká, Claire Tomlin, George Pappas, and Shankar Sastry. Generation of conflict resolution maneuvers for air traffic management. In *International Conference on Intelligent Robots and Systems (IROS)*, Grenoble, 1997.