Proceedings of the 35th
Conference on Decision and Control
Kobe, Japan • December 1996

Conflict Resolution for Multi-Agent Hybrid Systems [1]

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

Abstract

A conflict resolution architecture for multi-agent hybrid systems with emphasis on Air Traffic Management Systems (ATMS) is presented. In such systems, conflicts arise in the form of potential collisions which are resolved locally by inter-agent coordination. This results in a decentralized architecture in which safety issues are resolved locally and central agencies, such as Air Traffic Controllers, focus on global issues such as efficiency and optimal throughput. In order to allow optimization of agents' objectives, inter-agent coordination is minimized by noncooperative conflict resolution methods based on game theory. If noncooperative methods are unsuccessful, then cooperative methods in the form of coordinated maneuvers are used to resolve conflicts. The merging of inter-agent coordination, which is modeled by discrete event systems, and agent dynamics, which are modeled by differential equations, results in *hybrid systems.*

1 Introduction

We are increasingly confronted with the control of distributed multi-agent systems such as Air Traffic Management Systems (ATMS) [1], Intelligent Vehicle Highway Systems (IVHS) [2], control systems of an interconnected power grid, and communication networks. The common feature of these systems is that agents compete for usage of a common resource, such as space time on the jetways, airport runways, highways, etc. One of the most important conceptual issues to be addressed in the architecture of these control systems is their degree of decentralization. The completely decentralized solution is inefficient and leads to conflict, the completely centralized control laws are not tolerant of faults in the central controller, computationally and conceptually complicated and slow to respond to emergencies. *In our design paradigm, agents have control laws to maintain their safe operation, and try to optimize their own performance measures. They also coordinate with neighboring agents and a centralized controller to resolve conflicts as they arise and maintain efficient operation.*

For reasons of economic and reliable information transfer among the agents and the centralized controller, coordination among the agents is usually in the form of communication protocols which are modeled by discrete event systems. Since the dynamics of individual agents is modeled by differential equations, we are left with a combination of interacting discrete event dynamical systems and differential equations resulting in *hybrid control systems.* An important issue in the area of hybrid systems is the analysis and design of protocols and interfaces between agents as well as continuous control laws for each agent. Continuous control laws are usually proven correct by traditional tools of control theory, whereas verification of coordination protocols is performed by computer verification algorithms. There are several approaches to hybrid system design and verification (see, for example, [3, 4]).

A natural framework for formulating problems in which many agents have different objectives is game theory [5]. In this framework ([6]), each agent treats every other agent (for the sake of pairwise interactions) as a disturbance. Assuming a saddle solution to the game exists, the agent chooses an optimal policy assuming the worst possible disturbance. The resulting solution involves switching between different modes of operation and can be represented as a hybrid automaton. Game theoretic methods have been used in a similar way to prove that a set of maneuvers in Intelligent Vehicle Highway Systems is safe [7].

The current paper proposes a conflict resolution methodology for aircraft in the context of a new architecture for Air Traffic Management that has been proposed in [1] to allow for some shift away from the traditional completely centralized Air Traffic Control paradigm.

The organization of this paper is as follows. In Section 2, a conflict resolution architecture for multi-agent path planning systems is described. Section 3 describes a noncooperative zero sum game approach to long range collision avoidance. Section 4 describes the predefined, coordinated maneuvers which guarantee collision avoidance at short range. Section 5 discusses issues for further research.

2 Conflict Resolution Architecture

The design of a conflict resolution architecture has to choose a proper balance between centralized and decentralized authority. Central agencies are concerned with global issues while decentralized agencies are concerned with local problems. For multi-agent path planning problems, a conflict takes the form of a potential collision which is essentially a local property and it seems natural to address this problem at a decentralized level. *We therefore propose a conflict resolution architecture*

[1] Research supported by the Army Research Office under grant DAAH 04-95-1-0588 and by NASA under grant NAG2-243.

0-7803-3590-2/96 $5.00 © 1996 IEEE

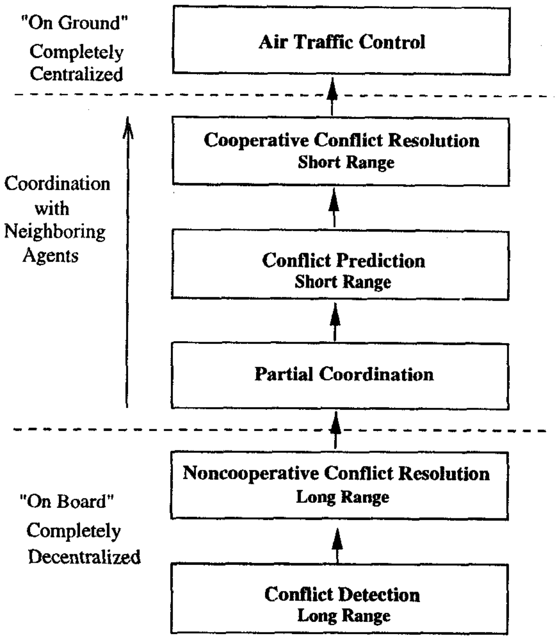

Figure 1: Conflict Resolution Architecture

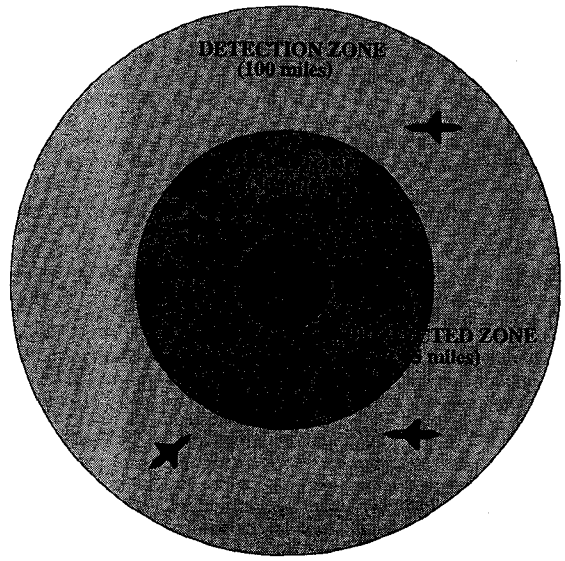

Figure 2: Aircraft Zones

in which conflicts are resolved locally by inter-agent coordination. Although conflicts will be resolved locally among agents, it is rather clear that the success of such an architecture will depend on the relative time that each agent is allowed to optimize its own goals. *It is therefore desirable to minimize inter-agent interaction.* For path planning problems, this will allow aircraft to stay on their nominal optimized trajectories for as long as possible.

Inter-agent coordination is minimized by classifying conflicts according to whether they are *long range* or *short range*, and by attempting to resolve long range conflicts without any coordination among agents. Short range conflicts are more safety critical than long range, and must be solved by coordination among the agents. These issues of centralized vs. decentralized control schemes, local vs. global system properties, and short range vs. long range conflict, are captured in the proposed hierarchical architecture for conflict resolution of Figure 1. The algorithms contained within this architecture, the subject of the rest of this paper, are resident in and are executed by each agent in the system. In Air Traffic Management Systems, the algorithms reside in the *Flight Vehicle Management System* (FVMS) located on board each aircraft.

As the agents referred to in this paper are actually aircraft in an air traffic system, *long range* and *short range* are conveniently depicted according to the sensor and communication ranges shown in Figure 2:

- *Detection Zone (approx. 100 miles[1])*: The detection zone is defined by the radius of the aircraft's sensing capability. Conflicts within this region are classified as *long range* and are resolved noncooperatively by small velocity variations which over the long range horizon will result in sufficiently large spacings between aircraft.

- *Alert Zone (approx. 30 miles)*: Conflicts within this region are classified as *short range*. Within this zone, conflicts are resolved cooperatively using more drastic maneuvers such as altitude and/or direction changes.

- *Protected Zone (approx. 2.5 miles)*: A collision between two aircraft occurs when their respective protected zones have nonempty intersection. Therefore, protected zones essentially provide a minimum safety distance between aircraft.

At the lowest level of the architecture, long range *Conflict Detection* is performed. Conflict detection is based on sensory information available to the aircraft, which detects the instantaneous position and heading of each aircraft within the *Detection Zone* around the given aircraft.

The first attempt to resolve the conflict is to perform *Noncooperative Conflict Resolution* with no coordination between the agents. The agents are treated as players in a n-player, zero-sum noncooperative dynamic game. If the game has an unsafe solution then *Cooperative Conflict Resolution*, in which the agents follow predefined maneuvers proven to be safe, is necessary. The class of maneuvers constructed to resolve conflicts must be rich enough to cover most possible conflict scenarios.

At the top of the hierarchy is the centralized ATC, which will intervene and attempt to resolve a conflict if it cannot be resolved by Cooperative Conflict Resolution among agents.

[1]Planar conflict resolution is considered in this paper; for conflict maneuvers which involve altitude changes, altitude profiles would also be specified for these zones.

3 Noncooperative Conflict Resolution

3.1 Game Theoretic Approach

The following methodology is used as a long range collision avoidance scheme. A two agent scenario is considered. Let

$$\dot{x} = f(x, u, d, t) \qquad x(t_0) = x_0 \tag{1}$$

model the dynamics of the relative configuration $x \in \mathbb{R}^n$ between the two agents, where $u \in \mathcal{U}$ is the control input of one agent, called the *evader*, and $d \in \mathcal{D}$ is the control of the other agent, called the *pursuer*. The actions of the evader are controlled whereas the actions of the pursuer are *unknown and uncontrolled* but are known to lie within the disturbance set $\mathcal{D}$. Thus, the actions of the evader are modeled as control inputs whereas the actions of the pursuer as disturbances.

The requirement for collision avoidance is encoded in a cost function $J_s(x_0, u, d)$ and is simply the distance between the two agents. A trajectory of system (1) is called *safe* if

$$J_s(x_0, u, d) \geq C \tag{2}$$

for some constant C determined by the size of the protected zones. In a zero-sum, noncooperative game, the pursuer tries to minimize the distance between the agents whereas the evader tries to maximize it.

A saddle solution to the game exists when there exists input u^* and disturbance d^* such that

$$\begin{aligned} J_s(x_0, u^*, d^*) &= \max_{d \in \mathcal{D}} \min_{u \in \mathcal{U}} J_s(x_0, u, d) \\ &= \min_{u \in \mathcal{U}} \max_{d \in \mathcal{D}} J_s(x_0, u, d) \end{aligned}$$

If a saddle solution exists, the optimal policy for the evader is u^* whereas the worst possible disturbance by the pursuer is d^*. If the trajectory of (1) corresponding to the saddle solution (u^*, d^*) is safe, then collision is avoided by the evader for the worst possible pursuer disturbance. This is the fundamental idea in noncooperative conflict resolution and results in minimal inter-agent interaction.

The safety of a particular control policy also depends on the initial relative configuration x_0. The set of safe initial relative configurations is defined as

$$V_s = \{x_0 \in \mathbb{R}^n | J_s(x_0, u^*, d^*) \geq C\} \tag{3}$$

and given an initial relative configuration $x_0 \in V_s$, the following set

$$\mathcal{U}_s(x_0) = \{u \in \mathcal{U} | J_s(x_0, u, d^*) \geq C\} \tag{4}$$

is defined as the set of control policies which guarantee safety from relative configuration x_0. Since all $u \in \mathcal{U}_s(x_0)$ guarantee safety from x_0, it is advantageous to find the control policy $u \in \mathcal{U}_s(x_0)$ which minimizes deviation from the nominal trajectory. Deviation from the nominal trajectory is encoded in a cost function J_e which is usually a quadratic function of the tracking error. Therefore, minimization of the tracking error which guarantees safety from relative configuration x_0, is performed by solving the following optimal control problem

$$\min_{u \in \mathcal{U}_s(x_0)} J_e \tag{5}$$

subject to the differential equations which describe the dynamics of individual agents in absolute coordinates and can therefore be used to describe the tracking error dynamics. Additional system requirements such as passenger comfort can also be incorporated by extending the above nested chain of games and optimal control problems [6]. The above methodology is now illustrated in the instance of planar conflict resolution.

3.2 Planar Conflict Resolution

Because conflicts between agents depend on the relative position and velocity of the agents, it is useful in the following analysis to derive *relative* kinematic 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 0 and aircraft 1, a relative model with its origin centered on aircraft 0 is used.

If x_r,y_r and θ_r denote the relative planar position and orientation, v_0,ω_0 the translational and rotational velocities of agent 0 and v_1,ω_1 the respective velocities for agent 1, then the relative configuration model is expressed by the following equations,

$$\begin{aligned} \dot{x}_r &= -v_0 + v_1 \cos\theta_r + \omega_0 y_r \\ \dot{y}_r &= v_1 \sin\theta_r - \omega_0 x_r \\ \dot{\theta}_r &= \omega_1 - \omega_0 \end{aligned} \tag{6}$$

A more detailed derivation of (6) can be found in [8]. Suppose we consider the special but interesting case of model (6) in which $\omega_i = 0$, $i = 0, 1$. The agents are restricted to straight line motion, which, for example, corresponds to two aircraft flying along straight lines at the same altitude. Conflicts can then be resolved by altering velocity variations. In this problem, v_0 is the control and v_1 is considered the disturbance. Collision is essentially avoided by altering the velocity profile of the trajectories. The input and disturbance lie in closed subsets of the positive real line,

$$v_0 \in \mathcal{U} = [\underline{v}_0, \overline{v}_0] \subset \mathbb{R} \tag{7}$$

$$v_1 \in \mathcal{D} = [\underline{v}_1, \overline{v}_1] \subset \mathbb{R} \tag{8}$$

In the case in which the agents are aircraft, we have $\underline{v}_0 > 0$ and $\underline{v}_1 > 0$. From now on, let $q = [x, y, \theta]^T$ denote the state with q_0 denoting the initial state. The requirement for collision avoidance is encoded in the cost function J_s,

$$J_s(q_0, v_0, v_1) = \inf_{t \geq 0}(|x_r(t)| + |y_r(t)|) \tag{9}$$

which is a measure of the distance between the pursuer and the evader. To avoid collisions, we require

$$J_s(q_0, v_0, v_1) \geq C, \tag{10}$$

where C describes a safety distance margin around the agents and is determined by the radius of the protected zones. The optimal policy for both the evader and the pursuer will correspond to a saddle point of the optimizing cost. It is clear from the dynamics that the saddle solution will depend on the position and orientation of the pursuer with respect to the evader.

Proposition 1 [Saddle Solution] *The global saddle solution (v_0^*, v_1^*) to the game described by system (6) with*

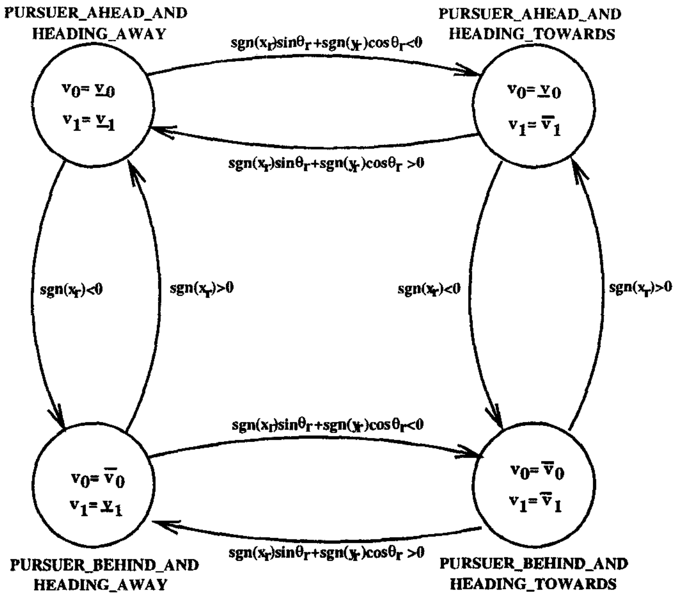

Figure 3: Abstraction of Saddle Solution as a Hybrid Automaton

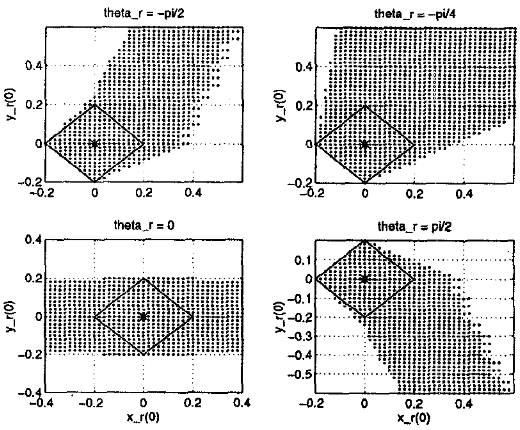

Figure 4: Unsafe sets $(x_r(0), y_r(0))$ for $[\underline{v}_0, \overline{v}_0] = [2, 4]$, $[\underline{v}_1, \overline{v}_1] = [1, 5]$ and $\theta_r = -\pi/2, -\pi/4, 0,$ and $\pi/2$. $C = 0.2$.

$\omega_0 = \omega_1 = 0$ for the cost $J_s(q_0, v_0, v_1)$ given by equation (9) is

$$v_0^* = \begin{cases} \underline{v}_0 & \text{if } sgn(x_r) > 0 \\ \overline{v}_0 & \text{if } sgn(x_r) < 0 \end{cases} \quad (11)$$

$$v_1^* = \begin{cases} \underline{v}_1 & \text{if } sgn(x_r)\cos\theta_r + sgn(y_r)\sin\theta_r > 0 \\ \overline{v}_1 & \text{if } sgn(x_r)\cos\theta_r + sgn(y_r)\sin\theta_r < 0 \end{cases} \quad (12)$$

Proof: In [8]. □

As can be seen from equation (11), the optimal control depends on the position of the pursuer relative to the evader. If the pursuer is ahead of the evader then the evader should move as slowly as possible whereas if the pursuer is behind the evader then the evader should move as quickly as possible. The worst possible disturbance is described by equation (12). Equation (12) may be interpreted intuitively as follows: if the pursuer is heading towards the evader, then the pursuer moves as quickly as possible; if heading away from the evader, the pursuer should move as slowly as possible.

Notice that the bang-bang nature of the saddle solution allows us to abstract the system behavior by the hybrid automaton shown in Figure 3.

It is clear from the feedback laws (11,12) that the resulting closed loop system is described by a discontinuous differential equation. The surfaces of discontinuity, as can be seen from equations (11,12), are simply the x_r- and y_r-axes. The following proposition establishes conditions under which chattering solutions in the sense of Fillipov are possible.

Proposition 2 [Chattering Saddle Solutions] *Consider model (6) with $\omega_0 = \omega_1 = 0$ and with control laws (11,12). Then the x_r-axis ($\{(x_r, y_r) \in \mathbb{R}^2 | y_r = 0\}$) is a chattering surface only if* [2]

$$\sin\theta_r \neq 0 \ \text{ and } \ \underline{v}_1 < 0 < \overline{v}_1 \quad (13)$$

and the y_r-axis ($\{(x_r, y_r) \in \mathbb{R}^2 | x_r = 0\}$) is a chattering surface only if either

$$\cos\theta_r < 0 \ \text{ and } \ \underline{v}_1 \cos\theta_r > \overline{v}_0 \ \text{ and } \ \overline{v}_1 \cos\theta_r < \underline{v}_0 \quad (14)$$

or

$$\cos\theta_r > 0 \ \text{ and } \ \overline{v}_1 \cos\theta_r > \overline{v}_0 \ \text{ and } \ \underline{v}_1 \cos\theta_r < \underline{v}_0 \quad (15)$$

Proof: In [8]. □

Having calculated the optimal policies for both the pursuer and the evader, one can find initial conditions for which the game is won by the evader and therefore collision is avoided regardless of what the pursuer does. Define the safe set of initial conditions as

$$V_s = \{(x_r(0), y_r(0)) \in \mathbb{R}^2 | J_s(x_r(0), y_r(0), v_0^*, v_1^*) \geq C\} \quad (16)$$

A computer program was written to calculate the *unsafe* sets of initial conditions ($(x_r(0), y_r(0))$ which result in a collision) for prespecified $\theta_r, \underline{v}_0, \overline{v}_0, \underline{v}_1, \overline{v}_1$. Figure 4 illustrates these unsafe sets for a variety of relative angles θ_r. If the initial condition lies in the unsafe set, then the saddle solution results in a collision. In this case, the agents cannot resolve the possible conflict in a noncooperative manner and thus some form of coordination among the agents is necessary.

4 Cooperative Conflict Resolution

This section addresses the problem of cooperative conflict resolution among aircraft. This kind of collision avoidance scheme must take place if noncooperative collision avoidance is not sufficient to resolve the conflict between the two aircraft. Cooperative collision

[2] Note that condition (13) does not apply to aircraft.

avoidance involves a direction change for at least one of the aircraft involved in the conflict. Chosen for its simplicity and inspired by [9], the path deviation for each pairwise conflict is chosen to be two consecutive heading changes resulting in a triangular deviation from the desired path as shown in Figure 5. For a conflict involving three or more aircraft, a scheme which guides each aircraft along a circular path around the conflict point is proposed.

4.1 Conflict Prediction

In this section, it is assumed that aircraft are cruising in straight lines at a constant altitude, with constant velocity (a valid assumption in the relatively sparse *en route* airspace away from the airport airspace). The velocities and headings of both aircraft involved in the conflict are assumed to be known by each aircraft: once an aircraft enters the alert zone of another aircraft, it transmits its heading and velocity information. This defines its course of action over a certain time horizon which enables the prediction of possible collisions.

Consider the Euclidean distance between two aircraft given by

$$J_s(t) = \sqrt{x_r^2(t) + y_r^2(t)} \qquad (17)$$

and recall that a collision is defined to be the nonempty intersection of the protected zones of two aircraft. The algorithm minimizes $J_s(t)$ over all positive time.

Proposition 3 *Given $x_r(t)$ and $y_r(t)$ by the solutions of (6) with $\omega_0 = \omega_1 = 0$, the global minimum value of $J_s(t)$ over all positive time occurs at*

$$t^* = \begin{cases} -\frac{a_1}{2a_2} & \text{if } a_2 \neq 0 \text{ and } \frac{-a_1}{2a_2} \geq 0 \\ 0 & \text{if } a_2 \neq 0 \text{ and } \frac{-a_1}{2a_2} < 0 \\ \mathbb{R}^+ & \text{if } a_2 = 0 \end{cases} \qquad (18)$$

where a_2,a_1 and a_0 are defined as

$$\begin{aligned} a_2 &= v_1^2 \sin^2\theta_r + (v_1\cos\theta_r - v_0)^2 \\ a_1 &= 2x_r(0)(v_1\cos\theta_r - v_0) + 2y_r(0)v_1\sin\theta_r (19) \\ a_0 &= x_r^2(0) + y_r^2(0) = J_s(0) \end{aligned}$$

Proof: In [8]. □

4.2 Protocol for Two Aircraft

A general short range conflict scenario is depicted in Figure 5. Aircraft 1 with speed v_1 and initial heading θ_r has desired *relative* trajectory $(x_r^d(t), y_r^d(t))$, which is the straight line path joining point A and point C a distance d away from the origin (seen as the dotted line in Figure 5). To simplify the analysis, the protected zone of aircraft 1 is translated to aircraft 0, to make the protected zone around aircraft 0 twice its original radius. To avoid the protected zone, the proposed deviation for aircraft 1 is the triangular path ABC tangent to the protected zone at two places and parameterized by the deviation angle θ (represented by the dashed line in Figure 5).

Consider the example in which aircraft 1 has the same initial heading as aircraft 0 ($\theta_r = 0$), and its original

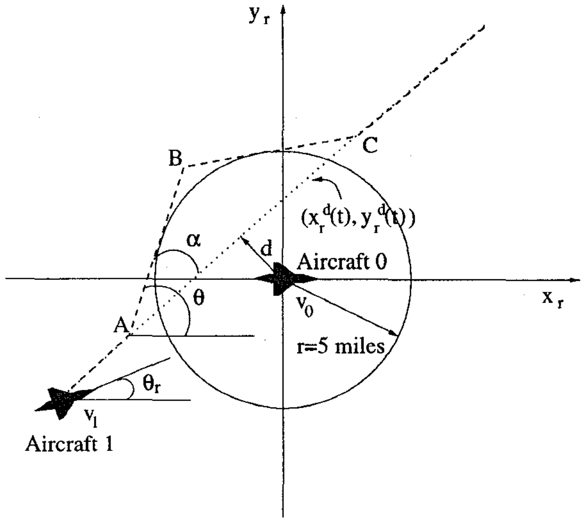

Figure 5: Showing the triangular path deviation (dashed line), at optimal angle θ^*, to be used in pairwise conflict avoidance

desired path is along the x_r axis ($d = 0$). A potential conflict exists if v_1 is greater than v_0. The minimum time triangular maneuver to avoid conflict, called an *Overtake maneuver* in this case, may be calculated geometrically.

Proposition 4 (Overtake Maneuver) *For an* Overtake, *in which aircraft 1 is approaching aircraft 0 from behind with a greater speed, the minimum time triangular maneuver for aircraft 1 approaches a departure angle of $\theta \to 45°$ from the horizontal, as the speed ratio $v_1/v_0 \to \infty$.*

Proof: In [8]. □

This makes sense intuitively: a deviation angle of 45° means that aircraft 1 remains on the horizontal path with $\theta_r = 0$ for as long as possible, since the speed differential between v_1 and v_0 is greatest along this path.

Consider now a *HeadOn* conflict in which aircraft 1 is heading towards aircraft 0 ($\theta_r = 180°$) along the x_r axis ($d = 0$). A potential conflict exists regardless of the speeds of aircraft 0 and aircraft 1. Although the conflict may be resolved using the general maneuver discussed above, the issue of *fairness* arises. If $v_1 \approx v_0$, it is not clear how to choose which aircraft deviates from its original trajectory. A natural solution is to define a maneuver in which both aircraft deviate from their original trajectories. Inspired by the Overtake maneuver, θ_0^* and θ_1^* are set to 45° and −45°, respectively, when $d = 0$ and $\theta_r = 180°$. The Overtake maneuver is *safe by design*, since the construction of the deviation path explicitly avoids the protected zone of one of the aircraft. In order to ensure that the HeadOn conflict is safe by design, both aircraft must deviate a horizontal distance of 5 miles (the minimum aircraft separation) away from their original paths. This is clearly not optimal, since it is worse than the previous case in which only one of the aircraft has to deviate a horizontal distance of $5\sqrt{2}$ miles. It is difficult to prove that less conservative designs, in which the horizontal deviation for each aircraft is less than 5 miles, are *a priori* safe for various speed ratios, unless a formal verification tool such as Cospan [10] or HyTech [11] is used. This type of tool automates a mathematical proof that, for given sets of initial conditions and constraints on the system variables, certain states will or will not be reached. For the maneuver discussed here, such a tool could be used to verify that the positions of the aircraft do not come within 5 miles of each other. The particular tool which has been used to verify similar maneuvers is the Hybrid system verification tool HyTech [12].

As with the Overtake maneuver, the HeadOn maneuver in its general form may be used for relative headings θ_r other than 180°. Once the Overtake or HeadOn maneuver is complete, a *Catch Up* maneuver is performed by the aircraft, to catch up to their original trajectories in time.

4.3 Protocol for Three Aircraft

For three aircraft coming into potential conflict, there are many more possibilities for types of conflict. For example, two aircraft could have intersecting trajectories, and then conflict resolution between these two could result in a new conflict with a third aircraft. Pairwise conflict resolution may not work in cases such as these: it is worthwhile to design a maneuver which works for three aircraft, with the possibility to extend it to more than three aircraft. A maneuver which is inspired by the traffic rotaries on the ground is the *Roundabout* maneuver. For this maneuver, a circular path is defined around the conflict points of all three trajectories. 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 its protected zone; in extreme cases this may force an aircraft to enter a holding pattern to delay its entry.

Figure 6 depicts the hybrid automaton describing the conflict resolution architecture described in this paper. If noncooperative conflict resolution (described by the hybrid automaton of Figure 3) results in an unsafe solution, then cooperative collision resolution involving one of the three maneuvers described in this section must be invoked. Once the maneuvers are complete, the aircraft enter the *Catch Up* mode, and the system returns to its original *No conflict* state. Once again, HyTech is used to verify the safety of the protocols.

5 Issues for Further Research

Further research topics include the extension of our planar example to include turning, extensions to multiple aircraft coordination, and collision resolution in three dimensional space. Non competitive games with n-players will be considered as well, with the design doing its best to minimize the level of verification required in subsequent stages, using conventional tools such as HyTech or Cospan. The protocols involving coordination were suggestive of Nash equilibria for noncompetitive games. In future work, this will be worked out in detail, along with information exchange requirements.

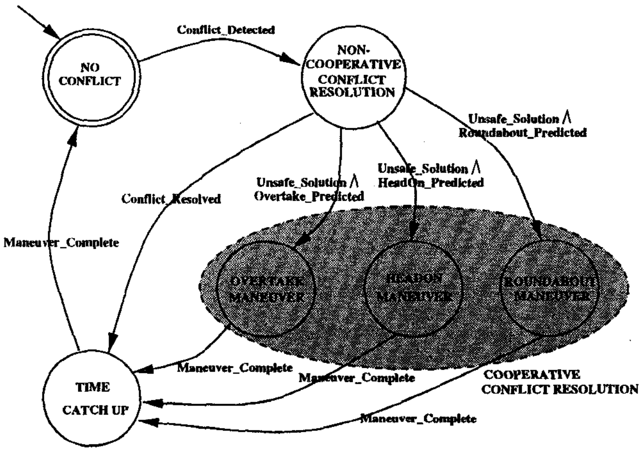

Figure 6: Hybrid Automaton describing Conflict Resolution

References

[1] S. Sastry, G. Meyer, C. Tomlin, J. Lygeros, D. Godbole, and G. Pappas. Hybrid control in air traffic management systems. In *Proceedings of the 1995 IEEE Conference in Decision and Control*, pages 1478–1483, New Orleans, LA, December 1995.

[2] Pravin Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, AC-38(2):195–207, 1993.

[3] Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors. *Hybrid Systems.* Springer-Verlag, 1993.

[4] Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors. *Hybrid Systems II.* Springer-Verlag, 1995.

[5] T. Basar and G. J. Olsder. *Dynamic Noncooperative Game Theory.* Academic Press, second edition, 1995.

[6] John Lygeros, Datta Godbole, and Shankar Sastry. A game theoretic approach to hybrid system design. Technical Report UCB/ERL M95/77, University of California, Berkeley, 1995.

[7] John Lygeros, Datta N. Godbole, and Shankar Sastry. A verified hybrid controller for automated vehicles. Submitted to IEEE Transactions on Automatic Control, Special Issue on Hybrid Systems, 1996.

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

[9] Russell A. Paielli and Heinz Erzberger. Conflict probability estimation and resolution for free flight. NASA Ames Research Center, Preprint, 1996.

[10] Z. Har'El and R.P. Kurshan. *Cospan User's Guide.* AT&T Bell Laboratories, 1987.

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

[12] Claire Tomlin. Verification of an air traffic management protocol using HyTech. University of California at Berkeley, 1996.