Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems

Claire Tomlin, *Student Member, IEEE*, George J. Pappas, *Student Member, IEEE*, and Shankar Sastry, *Fellow, IEEE*

***Abstract*—Air Traffic Management (ATM) of the future allows for the possibility of *free flight*, in which aircraft choose their own optimal routes, altitudes, and velocities. The safe resolution of trajectory conflicts between aircraft is necessary to the success of such a distributed control system. In this paper, we present a method to synthesize provably safe conflict resolution maneuvers. The method models the aircraft and the maneuver as a *hybrid control system* and calculates the maximal set of safe initial conditions for each aircraft so that separation is assured in the presence of uncertainties in the actions of the other aircraft. Examples of maneuvers using both speed and heading changes are worked out in detail.**

***Index Terms*—Air traffic management, conflict resolution, hybrid systems, verification.**

I. Introduction

Air transportation systems are faced with soaring demands for air travel. The annual air traffic rate in the United States is expected to grow by 3–5% annually for at least the next 15 years [1]. The current National Airspace System (NAS) architecture and management will not be able to efficiently handle this increase because of several limiting factors including the following.

- *Inefficient airspace utilization:* Currently, the airspace is very rigidly structured and aircraft are forced to travel along predetermined jetways. This is generally 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% [1] 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. In congested areas, such as the regions close to urban airports referred to as Terminal Radar Approach CONtrols (TRACON's), 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 [2]. Communication is restricted to congested voice communication between the aircraft and ATC. Navigation is performed by flying over fixed VHF Omni-Directional Range (VOR) points.

In view of the above problems, the aviation community is working toward an innovative concept called *Free Flight* [3]. Free Flight allows pilots to choose their own routes, altitude, and speed. User preference would be restricted only in congested airspace 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 like Automatic Dependence Surveillance-Broadcast (ADS-B) [4], [5], Traffic Alert and Collision Avoidance Systems (TCAS) [6], and powerful on-board computation. In addition, tools such as NASA's Center-TRACON Automation System (CTAS) [7] and MITRE's URET [8] will serve as decision support tools for ground controllers in an effort to reduce ATC workload and optimize capacity.

The above technological advances will also enable the current ATC system to accommodate future air traffic growth: sophisticated on-board equipment will allow aircraft to share some of the workload, such as navigation, weather prediction, and aircraft separation, with ground controllers. In order to improve the current standards of safety in an unstructured Free Flight environment, conflict detection and resolution algorithms are vital. Such algorithms would be used either on the ground by Air Traffic Control or in the air by the Flight Management System (FMS) of each aircraft.

In the proposed Free Flight airspace, each aircraft is surrounded by two virtual *cylinders* [4], the *protected zone* and the *alert zone*, shown in Fig. 1. A conflict or loss of separation between two aircraft occurs whenever the protected zones of the aircraft overlap. If the alert zones overlap, either ATC is notified about the potential conflict, or the aircraft exchange sensor and intent information in order to predict and resolve the conflict. The radius and the height of the en-route protected zone over U.S. airspace is currently 2.5 nautical mi and 2000 ft (1000 ft below 29 000 ft, 4000 ft over oceanic airspace), respectively. The size of the alert zone, currently under debate, depends on various factors including airspeed, altitude, accuracy of sensing equipment, traffic situation, aircraft performance, and average human and system response times.

Current research endeavors in conflict prediction and resolution include [9]–[13]. Conflict prediction could be spatial, temporal, or probabilistic. Spatial and temporal approaches, such as [11] and [13], calculate the four-dimensional coordi-

Manuscript received August 2, 1996. Recommended by Associate Editor, P. J. Antsaklis. This work was supported by NASA under Grant NAG 2-1039, by the Army Research Office under Grants DAAH 04-95-1-0588 and DAAH 04-96-1-0341, and by NSERC and ZONTA postgraduate fellowships.

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA.

Publisher Item Identifier S 0018-9286(98)02660-9.

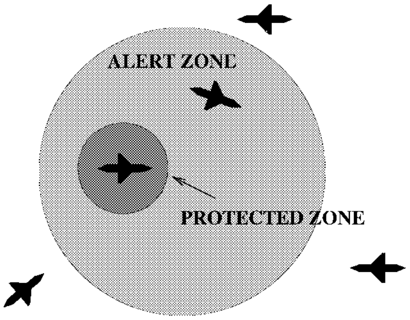


Fig. 1. Protected and alert zone for one aircraft.

nates of a possible conflict. Probabilistic approaches, such as [9] and [10], assume stochastic uncertainty in the measured information and determine the probability of collision. The work of [11] and [12] formulates conflict resolution as an optimal control problem, whereas [13] treats the problem as a convex optimization problem. The user interface of CTAS allows controllers to manually alter aircraft trajectories to resolve conflicts in en route airspace [14]. TCAS [6] provides resolution advisories (flight level changes) to pilots involved in two-aircraft conflicts, however these advisories are not formally verified. (Conflict prediction and resolution are the most important modules that are in need of augmentation and verification in the current implementations of CTAS and TCAS.)

In [15] a possible future architecture for Air Traffic Management (ATM) is presented. In our paradigm, aircraft are allowed to self-optimize in the spirit of Free Flight, communicate state and intent data to each other using an ADS-B datalink for conflict prediction, and coordinate with each other to resolve potential conflicts. State and intent data could be uncertain. Coordination among the aircraft is in the form of *maneuvers* which are finite sequences of *flight modes* such as heading, altitude, and speed changes for each aircraft. These types of maneuvers are routinely used in current Air Traffic Control practice since they are easily understandable by pilots as well as easily implementable by on-board autopilots which regulate the aircraft to heading and speed setpoints. The main thrust of our conflict resolution algorithms is to *verify* that a maneuver successfully resolves the conflict by computing the set of initial conditions for which the maneuver is safe, where safety means that separation is maintained. In the presence of bounded uncertainty in the state or intent data, we take a worst case approach and verify that the worst case system trajectory is safe.

The flight mode switching occurring in each maneuver is modeled by a finite-state automaton with the relative aircraft configuration dynamics residing within each flight mode. A conflict resolution maneuver is therefore modeled by a finite state automaton interacting with a set of continuous control systems, resulting in a *hybrid control system*. The interaction and information exchange of all of the aircraft involved in the maneuver results in a *multiagent hybrid control system*.

There are several approaches to hybrid system modeling, verification, and controller design (see, for example, [16]–[19]). The computer science approach is to extend models of finite-state automata to timed automata [20], linear hybrid automata [21], and hybrid input/output automata [22]. Linear hybrid automata model or abstract the continuous dynamics by differential inclusions of the form $A\dot{x} \leq b$ and verify properties of the resulting abstracted system [23]–[25]. Specifications are verified for these models using either model checking, which exhaustively check all system trajectories, or deductive theorem-proving techniques [26], which prove the specification by induction on all system trajectories. In this framework, controller design has also been developed [27], [28]. Automated computational tools have been developed for both model checking [29], [30] and theorem proving [31]. Control theoretic approaches to modeling, analysis, and controller design for hybrid systems have extended the theory of dynamical systems to include discrete modes of operation. Modeling approaches include those of [32]–[37]. Analysis and design techniques extend existing control techniques, such as stability theory [33], optimal control [33], [36], [37], and control of discrete-event systems [38], [39], to hybrid systems.

Our conflict resolution algorithms are in the spirit of model checking, but we use control theoretic (deductive) techniques to calculate the reachable region for hybrid systems with general nonlinear dynamics. Our method calculates the largest controlled invariant subset of the complement of each aircraft's protected zone, taking into account the uncertainty of the actions of the other aircraft. In order to compute this safe set of states, we first develop a method to compute the controlled invariant subsets for continuous systems in the presence of disturbances. A natural framework for this type of problem is zero-sum noncooperative dynamic game theory [40], [41]. In this framework, uncertain information about neighboring aircraft is treated 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. This is motivated by the work of [42], in which game theoretic methods are used to prove safety of a set of maneuvers in Intelligent Vehicle Highway Systems.

Along with the safe set of initial states, we calculate the corresponding safe set of control inputs as a function of the state. 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. At the boundary of its safe region, the aircraft must apply the particular control which keeps it out of its unsafe region. Thus, we are naturally led to a switching control-based protocol which is *least restrictive*. A more detailed description of this multiobjective methodology may be found in [43]. 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.

The organization of this paper is as follows: In Section II our modeling formalism and design methodology for hybrid systems is described. Section III presents the game theoretic approach to computing the safe set of initial conditions and control inputs for continuous systems. Section IV describes safety verification of coordinated maneuvers using the results of Section III. Section V presents a brief summary and some issues for further research.

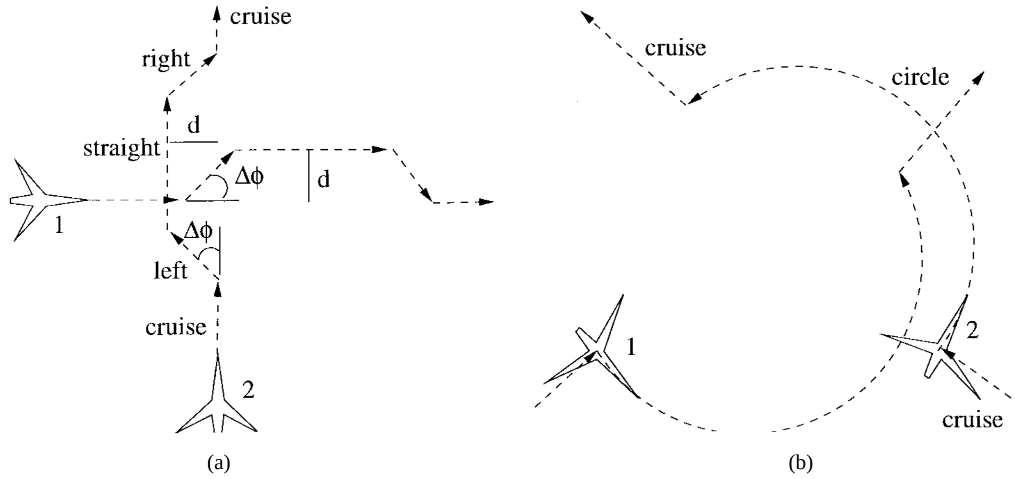


Fig. 2. Two different conflict resolution maneuvers, with associated modes.

II. Hybrid Model and Design Methodology

In this section, we present a hybrid system model for conflict resolution maneuvers and a method to verify the safety of, and synthesize control schemes for, these maneuvers. The discrete states of the hybrid system model the different flight modes that each aircraft steps through while executing the maneuver. For example, consider the two-aircraft examples of Fig. 2. In the first, the aircraft avoid each other by transitioning through a sequence of heading changes: "left," "straight," "right," and then back to the original "cruise" mode; in the second the conflict is avoided by both aircraft transitioning to a "circle" mode from a "cruise" mode.

Each mode has associated with it the relative aircraft configuration dynamics. The verification of the safety of each maneuver, with possible variations in the control inputs of each aircraft and changes in the switching times between modes, is complicated and in general not possible to compute manually. The hybrid model presented in this section provides an organized, formal way to model and prove the safety of the maneuver.

The hybrid model described below is inspired by that of [23] for linear hybrid automata, with the difference that we allow for a nonlinear continuous dynamic model within each discrete state and a general discrete transition relation.

Hybrid System Model

A hybrid system H is defined to be the tuple $H = (Q \times M, U \times D, \Sigma, I, \mathrm{Inv}, E, f)$, in which

- $Q \times M$ is the state space, with $Q = \{q_1, q_2, \cdots, q_m\}$ a finite set of discrete states and M an n-manifold; a state of the system is a pair $(q_i, x) \in Q \times M$;
- $U \times D \subset \mathbb{R}^u \times \mathbb{R}^d$ is the product of the input set and disturbance set; the space of acceptable control and disturbance trajectories are denoted by $\mathcal{U} = \{u(\cdot) \in PC^0 | u(\tau) \in U \ \forall \tau \in \mathbb{R}\}, \mathcal{D} = \{d(\cdot) \in PC^0 | d(\tau) \in D \ \forall \tau \in \mathbb{R}\}$;
- Σ is a finite set of transition labels;
- $I \subset Q \times M$ is the set of initial conditions;
- Inv: $Q \rightarrow 2^M$ is the invariant associated with each discrete state, meaning that the state (q, x) may flow within q only if $x \in \mathrm{Inv}(q)$;
- $E \subset Q \times M \times \Sigma \times Q \times M$ is the set of discrete jumps with $(q, x, \sigma, q', x') \in E$ meaning that if the current state is (q, x), the system may instantaneously take a discrete transition σ to state (q', x');
- f: $Q \times M \times U \times D \rightarrow TM$ is a map which associates with each discrete state $q \in Q$ a control system $f(q, x, u, d)$. For notational convenience we use $f_q(x, u, d)$ to denote $f(q, x, u, d)$.

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

Relative Aircraft Configuration Models

We now describe the continuous dynamics within each discrete state 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.

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

$$g_r = \begin{bmatrix} \cos\phi_r & -\sin\phi_r & x_r \\ \sin\phi_r & \cos\phi_r & y_r \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

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

$$\begin{aligned} \begin{bmatrix} x_r \\ y_r \end{bmatrix} &= R(-\phi_1) \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(-\phi_1) & -\sin(-\phi_1) \\ \sin(-\phi_1) & \cos(-\phi_1) \end{bmatrix} \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} && (6) \\ \phi_r &= \phi_2 - \phi_1 && (7) \end{aligned}$$

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

$$X_1 = \begin{bmatrix} 0 & -\omega_1 & v_1 \\ \omega_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad X_2 = \begin{bmatrix} 0 & -\omega_2 & v_2 \\ \omega_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8}$$

where v_i, ω_i represent the linear and angular velocities. Inserting (5) and (8) in (4) results in the $SE(2)$ relative configuration dynamics in local coordinates

$$\begin{aligned} \dot{x}_r &= -v_1 + v_2 \cos\phi_r + \omega_1 y_r \\ \dot{y}_r &= v_2 \sin\phi_r - \omega_1 x_r \\ \dot{\phi}_r &= \omega_2 - \omega_1. \end{aligned} \tag{9}$$

Similar results for $SE(3)$ may be found in [44]. Thus for each discrete state q, the dynamics $\dot{x} = f_q(x, u, d)$ is described by (9), with $x = (x_r, y_r, \phi_r)^T$. The linear (or angular) velocity of aircraft 1 is the control input u, and the uncertain linear (or angular) velocity of aircraft 2 is considered to be the disturbance d.

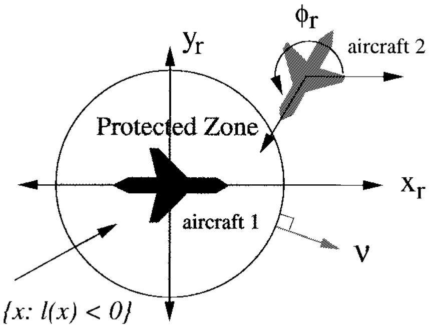


Fig. 3. The relative configuration, showing the protected zone and outward pointing normal.

Design and Verification Methodology

In the remainder of this paper we derive a method to generate the unsafe region of the state space $Q \times M$, which is the subset of initial states I for which, regardless of the control input, there exists a trajectory of H from this subset to an illegal region of the state space.

Given a subset $K \subset Q \times M$, we define the predecessor of K under continuous flows as

$$\begin{aligned} \mathrm{Pre}_t(K) = \{(q,x) \in Q \times M \mid\, & \exists (q', x') \in K \\ & \text{such that } q' = q \text{ and} \\ & \exists d(\cdot) \in \mathcal{D}, \forall\, u(\cdot) \in \mathcal{U}, \text{and} \\ & x(\tau) \in \mathrm{Inv}(q), \forall\, \tau \in [t, 0], \text{satisfying} \\ & x(t) = x, x(0) = x', \text{and} \\ & \dot{x}(\tau) = f_q(x(\tau), u(\tau), d(\tau))\}. \end{aligned} \tag{10}$$

Similarly, the predecessor of K under discrete transitions $\sigma \in \Sigma$ is defined to be

$$\begin{aligned} \mathrm{Pre}_\sigma(K) = \{(q,x) \in Q \times M \mid\, & \exists (q', x') \in K \text{ and} \\ & (q, x, \sigma, q', x') \in E\}. \end{aligned} \tag{11}$$

The predecessor under continuous flows or discrete transitions is defined as

$$Pre(K) = \left(\bigcup_{\sigma \in \Sigma} \mathrm{Pre}_\sigma(K) \right) \cup \mathrm{Pre}_{-\infty}(K). \tag{12}$$

For two-aircraft conflicts, we define the illegal region to be the *relative* protected zone, or the 5-mi radius cylinder around aircraft 1, denoted T with boundary ∂T, illustrated in Fig. 3. In Section III, a methodology is developed to compute $\mathrm{Pre}_t(T)$, using level sets of an appropriate Hamilton–Jacobi–Isaacs partial differential equation (PDE). This computation is subsequently used in Section IV to verify the conflict resolution maneuver.

III. The Hamilton–Jacobi–Isaacs Approach for Continuous Systems

Consider the dynamics of the aircraft in one discrete state $q \in Q$ (for notational simplicity we drop the subscript q in this section)

$$\dot{x} = f(x,u,d) \qquad x(t) = x \tag{13}$$

where $x \in \mathbb{R}^n$ describes the relative configuration of aircraft 2 with respect to aircraft 1, $u \in U \subset \mathbb{R}^u$ is the control input which models the actions of aircraft 1, and $d \in D \subset \mathbb{R}^d$ is the disturbance input which models the actions of aircraft 2. We assume that the system starts at state x at initial time t. Both U and D are known sets, but whereas the control input u may be chosen by the designer, the disturbance d is unknown, and models the uncertainty of the actions of aircraft 2.

The goal is to maintain safe operation of (13), meaning that the system trajectories do not enter T, the "Target set." We assume that there exists a differentiable function $l(x)$ so that $T = \{x \in \mathbb{R}^n | l(x) < 0\}$ and $\partial T = \{x \in \mathbb{R}^n | l(x) = 0\}$.

A. The Value Function and the Hamilton–Jacobi–Isaacs Equation

This section describes the computation of the unsafe subset of the state space, denoted $\mathrm{Pre}_t(T) \subset M$, which is the subset from which there exists a disturbance action $d(\cdot)$ such that the resulting trajectory of (13) enters T in at most t s. Due to the uncertainty in the actions of aircraft 2, the safest possible strategy of aircraft 1 is to fly a trajectory which guarantees that the minimum allowable separation with aircraft 2 is maintained, *regardless* of the actions of aircraft 2. We formulate this problem as a two-person, zero-sum dynamical game and calculate the "losing" states for aircraft 1.

Consider (13) over the time interval $[t, 0]$, where $t < 0$. The value function of the game is defined by

$$J(x,u(\cdot),d(\cdot),t)\colon \mathbb{R}^n \times \mathcal{U} \times \mathcal{D} \times \mathbb{R}_- \to \mathbb{R} \tag{14}$$

such that $J(x,u(\cdot),d(\cdot),t) = l(x(0))$. This value function may be interpreted as the cost of a trajectory $x(\cdot)$ which starts at x at time $t \leq 0$, evolves according to (13) with input $(u(\cdot), d(\cdot))$, and ends at the final state $x(0)$. Note that the value function depends only on the final state: there is no running cost, or *Lagrangian*. This encodes the fact that we are only interested in whether or not the system trajectory ends in T and are not concerned with intermediate states. The game is lost by aircraft 1 if the terminal state $x(0)$ is inside T.

Given $J(x,u(\cdot),d(\cdot),t)$, we first characterize the *unsafe* portion of ∂T, defined as those states $x \in \partial T$ for which there exists some disturbance $d \in D$ such that for all inputs $u \in U$ 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 $u \in U$ such that for all disturbances $d \in D$, the vector field points outward from T. Define the outward pointing normal to T as $\nu^T = Dl(x)$, then

$$\begin{aligned}
&\text{Safe portion of } \partial T\\
&\qquad \{x \in \partial T\colon \exists u \forall d \quad \nu^T f(x,u,d) \geq 0\}\\
&\text{Unsafe portion of } \partial T\\
&\qquad \{x \in \partial T\colon \forall u \exists d \quad \nu^T f(x,u,d) < 0\}.
\end{aligned} \tag{15}$$

The optimal control u^* and the worst disturbance d^* are given by

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

$$d^* = \arg \min_{d \in \mathcal{D}} \max_{u \in \mathcal{U}} J(x,u(\cdot),d(\cdot),t). \tag{17}$$

The game is said to have a saddle solution (u^*, d^*) if the resulting optimal cost does not depend on the order in which the maximization and minimization is performed

$$\begin{aligned}
J^*(x,t) &= \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} J(x,u(\cdot),d(\cdot),t)\\
&= \min_{d \in \mathcal{D}} \max_{u \in \mathcal{U}} J(x,u(\cdot),d(\cdot),t).
\end{aligned} \tag{18}$$

The concept of a saddle solution is key to our computation of the safe regions of operation of the aircraft, since a solution of (13) with $u = u^*$, and $d = d^*$ represents an optimal trajectory for *each* player under the assumption that the other player plays its optimal strategy.

Aircraft 1 maintains safety at time t by operating outside of $\mathrm{Pre}_t(T)$

$$\begin{aligned}
\mathrm{Pre}_t(T) &= \{x \in \mathrm{Inv}(q) | \exists d(\cdot) \in \mathcal{D}, J(x,u(\cdot),d(\cdot),\tau) < 0,\\
&\qquad \forall\, u(\cdot) \in \mathcal{U}, \forall\, \tau \in [t,0]\} && (19)\\
&= \{x \in M | \exists d(\cdot) \in \mathcal{D}, J(x,u^*(\cdot),d(\cdot),\tau) < 0,\\
&\qquad \forall\, \tau \in [t,0]\} \cap \mathrm{Inv}(q). && (20)
\end{aligned}$$

Let $\partial\mathrm{Pre}_t(T)$ denote the boundary of $\mathrm{Pre}_t(T)$. To calculate the unsafe set of states for all $t \in (-\infty, 0]$, we construct the Hamilton–Jacobi–Isaacs PDE 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

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

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

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

with the boundary conditions $p(0) = Dl(x(0))$ and $x(t) = x$. If $J^*(x,t)$ is a smooth function of x and t, then $J^*(x,t)$ satisfies the Hamilton–Jacobi–Isaacs equation

$$-\frac{\partial J^*(x,t)}{\partial t} = H^*\left(x, \frac{\partial J^*(x,t)}{\partial x}\right) \tag{23}$$

with boundary condition $J^*(x,0) = l(x)$. It is difficult to guarantee that the PDE (23) has smooth solutions for all $t \leq 0$, due to the occurrence of "shocks," i.e., discontinuities in J as a function of x. If there are no shocks in the solution of (23), we characterize the set

$$\mathrm{Pre}_{-\infty}(T) = \{x \in \mathrm{Inv}(q) | J^*(x, \tau) < 0, \forall \tau \in (-\infty, 0]\} \tag{24}$$

by solving the modified Hamilton–Jacobi–Isaacs equation

$$-\frac{\partial J^*(x,t)}{\partial t} = \min\left\{0, H^*\left(x, \frac{\partial J^*(x,t)}{\partial x}\right)\right\} \tag{25}$$

with boundary condition $J^*(x, 0) = l(x)$. The "min" is added to the right-hand side of (25) to ensure that states which are once unsafe cannot become safe. In practical applications, since one is concerned only with aircraft in the alert zone, the calculation of (24) may be approximated by computing $\mathrm{Pre}_t(T)$, for sufficiently large t, such as $t = 20$ min.

The set $\mathrm{Pre}_{-\infty}(T)$ defines the *least restrictive control scheme* for safety. If aircraft 2 is outside $\mathrm{Pre}_{-\infty}(T)$, any control input may be safely applied by aircraft 1, whereas on the boundary, the only input which may be safely applied to ensure safety is u^*. The safe set of control inputs associated with each state at time t is

$$\mathcal{U}_s(x, t) = \{u(\cdot) \in \mathcal{U} | J(x, u(\cdot), d^*(\cdot), t) \geq 0\}. \tag{26}$$

Additional system requirements, such as optimal fuel trajectories and passenger comfort, can now be incorporated by optimizing secondary and tertiary criteria within the constraints of set (26), following the multiobjective design methodology of [43].

We now apply this general framework to the planar $SE(2)$ relative model (9) in local coordinates (x_r, y_r, ϕ_r), with the control actions either the angular or linear velocities.

B. Angular Velocities as Control Actions

Consider the case in which the linear velocities of both aircraft are fixed, $v_1, v_2 \in \mathbb{R}^+$, and the control inputs of the aircraft are the angular velocities, $u = \omega_1$ and $d = \omega_2$

$$\begin{aligned} \dot{x}_r &= -v_1 + v_2 \cos \phi_r + u y_r \\ \dot{y}_r &= v_2 \sin \phi_r - u x_r \\ \dot{\phi}_r &= d - u \end{aligned} \tag{27}$$

with state variables $x_r, y_r \in \mathbb{R}, \phi_r \in [-\pi, \pi)$, and control and disturbance inputs $u \in U = [\underline{\omega}_1, \overline{\omega}_1] \subset \mathbb{R}, d \in D = [\underline{\omega}_2, \overline{\omega}_2] \subset \mathbb{R}$. 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 in the relative frame

$$T = \{(x_r, y_r) \in \mathbb{R}^2, \phi_r \in [-\pi, \pi) | x_r^2 + y_r^2 < 5^2\} \tag{28}$$

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

$$l(x) = x_r^2 + y_r^2 - 5^2. \tag{29}$$

The optimal Hamiltonian is

$$\begin{aligned} H^*(x, p) = \max_{u \in U} \min_{d \in D} [&-p_1 v_1 + p_1 v_2 \cos \phi_r + p_2 v_2 \sin \phi_r \\ &+ (p_1 y_r - p_2 x_r - p_3) u + p_3 d]. \end{aligned} \tag{30}$$

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

$$\begin{aligned} s_1(t) &= p_1(t) y_r(t) - p_2(t) x_r(t) - p_3(t) \\ s_2(t) &= p_3(t) \end{aligned} \tag{31}$$

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

$$\begin{aligned} u^* &= \mathrm{sgn}(s_1) \\ d^* &= -\mathrm{sgn}(s_2). \end{aligned} \tag{32}$$

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

$$\begin{aligned} \dot{p}_1 &= u^* p_2 \\ \dot{p}_2 &= -u^* p_1 \\ \dot{p}_3 &= p_1 v_2 \sin \phi_r - p_2 v_2 \cos \phi_r \end{aligned} \tag{33}$$

with $p(0) = (x_r, y_r, 0)^T = \nu$, 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 (15) with $\nu = (x_r, y_r, 0)^T$. Thus, those (x_r, y_r, ϕ_r) on ∂T for which

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

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

$$-v_1 x_r + v_2(x_r \cos \phi_r + y_r \sin \phi_r) = 0 \tag{35}$$

are the final state conditions for the boundary of the unsafe set $\mathrm{Pre}_t(T)$. To solve for $p(t)$ and $x(t)$ along this boundary for $t < 0$, we must first determine $u^*(0)$ and $d^*(0)$. Equations (32) are not defined at $t = 0$, 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 [41, pp. 442–443], we use an indirect method to calculate $u^*(0)$ and $d^*(0)$: at any point (x_r, y_r, ϕ_r) on ∂T, the derivatives of the switching functions s_1 and s_2 are

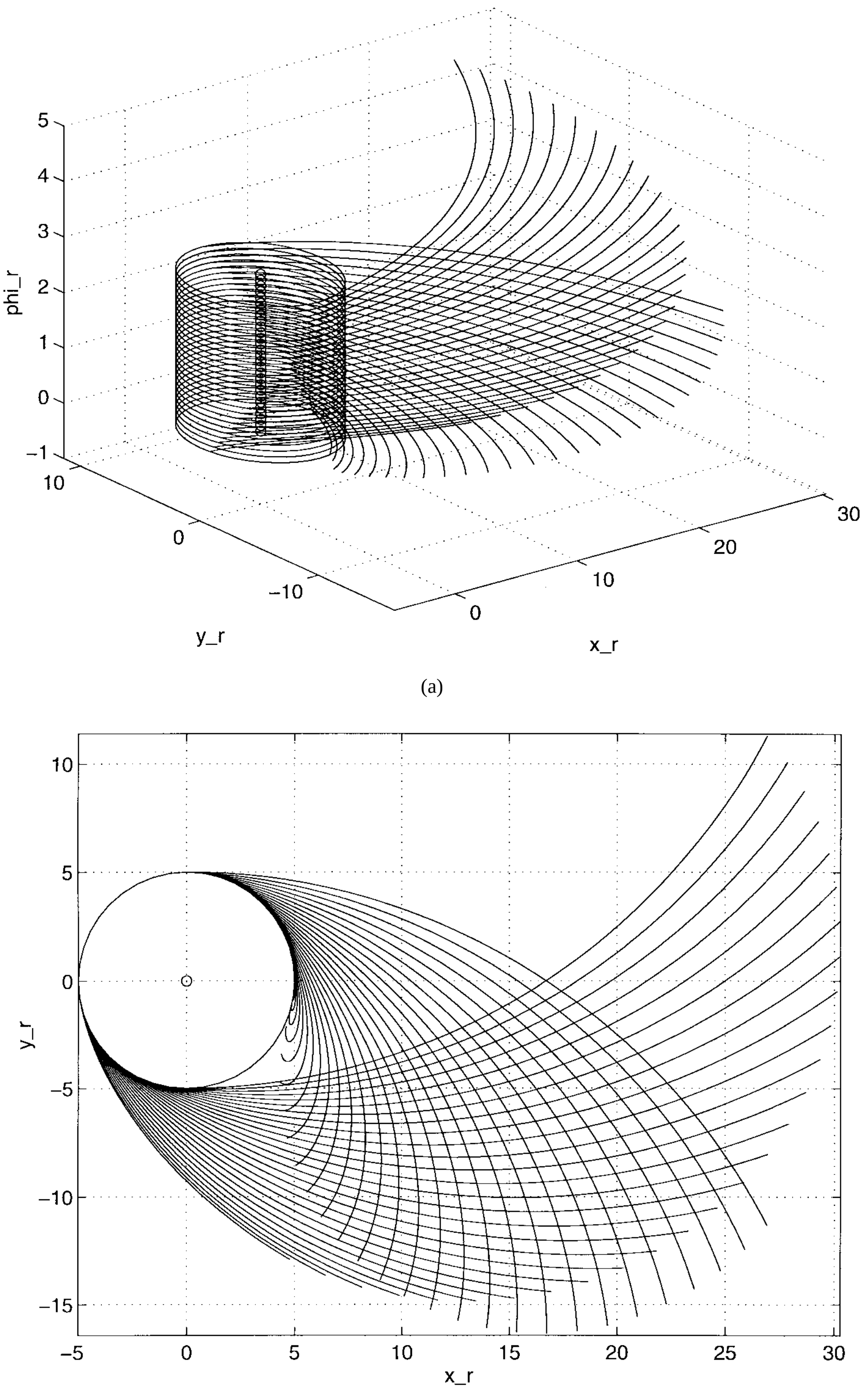
$$\dot{s}_1 = y_r v_1 \tag{36}$$

$$\dot{s}_2 = x_r v_2 \sin \phi_r - y_r v_2 \cos \phi_r. \tag{37}$$

Consider, for example $(x_r, y_r, \phi_r) = (0, 5, \pi) \in \partial T$ for the case $v_1 = v_2$. This point satisfies (35) and thus belongs to $\mathrm{Pre}_t(T)$ for all t. At this point, $\dot{s}_1 > 0$ and $\dot{s}_2 > 0$, meaning that for values of t slightly less than zero, $s_1 < 0$ and $s_2 < 0$. Thus for this point, $u^*(0) = -1$ and $d^*(0) = 1$.

These values for u^* and d^* remain valid for $t < 0$ 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 boundary. These points correspond to loss of smoothness in the Hamilton–Jacobi–Isaacs equation discussed above. Fig. 4 displays the resulting boundary of the unsafe set $\mathrm{Pre}_t(T)$, for $t < 0$ 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 Fig. 5. The computation of the boundary of $\mathrm{Pre}_{-\infty}(T)$ is in general difficult. For certain ranges of $\mathcal{U}$ and $\mathcal{D}$, the surfaces shown in Fig. 4 intersect,



(a)

(b)

Fig. 4. The Target set $T = \{(x_r, y_r), \phi_r \in (0, \pi) | x_r^2 + y_r^2 \leq 5^2\}$ (cylinder) and the boundary of the set $\mathrm{Pre}_t(T)$ (enclosed by the boundary) for $t < 0$ until the first switch in either $s_1(t)$ or $s_2(t)$. The second picture is a top view of the first.

and at the intersection, it is not clear that u^* is the unique safe input.

C. Linear Velocities as Control Actions

Now consider the case in which the angular velocities of the two aircraft are zero and the control inputs are the linear velocities of the aircraft: $u = v_1, d = v_2$, and model (9) reduces to

$$\begin{aligned}\dot{x}_r &= -u + d\cos\phi_r\\ \dot{y}_r &= d\sin\phi_r\\ \dot{\phi}_r &= 0.\end{aligned} \tag{38}$$

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

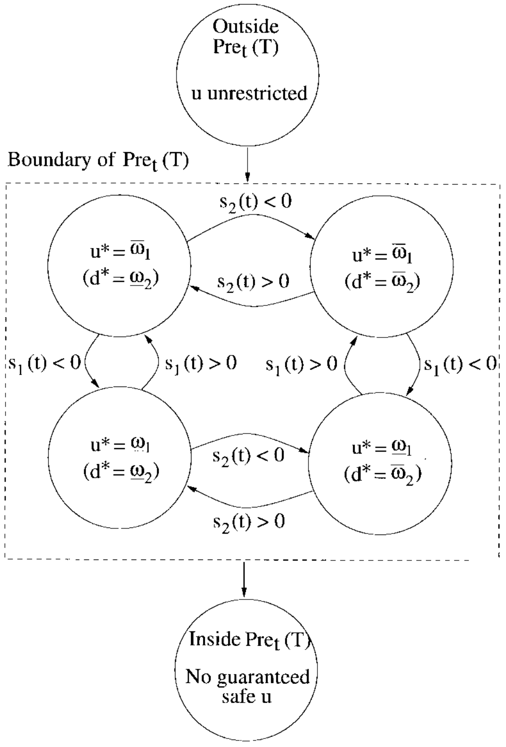

Outside Pre_t (T)
u unrestricted
Boundary of Pre_t (T)
$s_2(t) < 0$
$u^* = \overline{\omega}_1$ $(d^* = \underline{\omega}_2)$
$s_2(t) > 0$
$u^* = \overline{\omega}_1$ $(d^* = \overline{\omega}_2)$
$s_1(t) < 0$
$s_1(t) > 0$
$s_1(t) > 0$
$s_1(t) < 0$
$u^* = \underline{\omega}_1$ $(d^* = \underline{\omega}_2)$
$s_2(t) < 0$
$u^* = \underline{\omega}_1$ $(d^* = \overline{\omega}_2)$
$s_2(t) > 0$
Inside Pre_t (T)
No guaranteed safe u

Fig. 5. 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 outside $\mathrm{Pre}_t(T)$. The diagonal transitions in the automaton for the boundary of $\mathrm{Pre}_t(T)$ are not shown for legibility. In practice, t should be chosen large enough to take into account aircraft in the alert zone.

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 (38) for piecewise constant u and d and substituting the solutions into the cost function (14). We define the switching functions s_1 and s_2 as

$$\begin{aligned} s_1(t) &= x_r \\ s_2(t) &= x_r \cos\phi_r + y_r \sin\phi_r. \end{aligned} \tag{39}$$

Proposition 1 (Saddle Solution for Linear Velocity Controls): The global saddle solution (u^*, d^*) to the game described by system (38) for the cost $J(x, u(\cdot), d(\cdot), t)$ given by (14) is

$$u^* = \begin{cases} \underline{v}_1, & \text{if } \mathrm{sgn}(s_1) > 0 \\ \overline{v}_1, & \text{if } \mathrm{sgn}(s_1) < 0 \end{cases} \tag{40}$$

$$d^* = \begin{cases} \underline{v}_2, & \text{if } \mathrm{sgn}(s_2) > 0 \\ \overline{v}_2, & \text{if } \mathrm{sgn}(s_2) < 0. \end{cases} \tag{41}$$

Proof: See the Appendix. □

As can be seen from (40), u^* depends on the position of aircraft 2 relative to aircraft 1. If aircraft 2 is ahead of aircraft 1 in the relative axis frame, then u^* is at its lower limit; if aircraft 2 is behind aircraft 1 in the relative axis frame, then u^* is at its upper limit. If aircraft 2 is heading toward aircraft 1, then d^* is at its upper limit; if aircraft 2 is heading away from aircraft 1, 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 Fig. 6, which describes the least restrictive control scheme for safety. The unsafe sets of states are illustrated in Fig. 7 for various values of ϕ_r and speed ranges as illustrated.

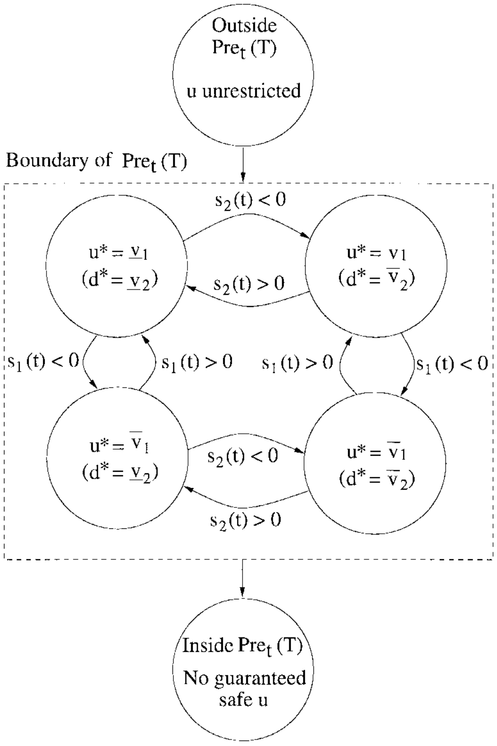

Outside Pre_t (T)
u unrestricted
Boundary of Pre_t (T)
$s_2(t) < 0$
$u^* = \underline{v}_1$ $(d^* = \underline{v}_2)$
$s_2(t) > 0$
$u^* = v_1$ $(d^* = \overline{v}_2)$
$s_1(t) < 0$
$s_1(t) > 0$
$s_1(t) > 0$
$s_1(t) < 0$
$u^* = \overline{v}_1$ $(d^* = \underline{v}_2)$
$s_2(t) < 0$
$u^* = \overline{v}_1$ $(d^* = \overline{v}_2)$
$s_2(t) > 0$
Inside Pre_t (T)
No guaranteed safe u

Fig. 6. Switching law governing the two-aircraft system with linear velocity control inputs.

IV. Verification of Conflict Resolution Maneuvers

In this section we apply the $\mathrm{Pre}_t(T)$ calculation of Section III to calculate the unsafe set of initial conditions for a conflict resolution maneuver. We illustrate the methodology on a maneuver whose form is chosen to be a finite sequence of heading changes resulting in a trapezoidal deviation from the desired path. Consider the conflict scenario and resolution maneuver shown in Fig. 2(a). The protocol may be linguistically expressed as follows.

1) Cruise until aircraft are α_1 mi apart.
2) Make a heading change of $\Delta\phi$ and fly until a lateral displacement of at least d mi is achieved for both aircraft.
3) Make a heading change to original heading and fly until the aircraft are α_2 mi apart.
4) Make a heading change of $-\Delta\phi$ and fly until a lateral displacement of d mi is achieved for both aircraft.
5) Make a heading change to original heading and cruise.

The maneuver is modeled as the hybrid automaton H shown in Fig. 8. [The state space of H is $Q \times \mathbb{R}^2 \times S^1 \times \mathbb{R}$ where $Q = \{CRUISE, LEFT, STRAIGHT, RIGHT\}$ and models the different flight modes in the maneuver, and $(x_r, y_r, \phi_r, t) \in \mathbb{R}^2 \times S^1 \times \mathbb{R}$ are the continuous variables which evolve within each discrete location according to the relative configuration dynamics (9).] The initial condition of the automaton is

$$I = CRUISE \times \{(x_r, y_r) \in \mathbb{R}^2 | 5^2 \le x_r^2 + y_r^2 \le r_a^2\} \tag{42}$$

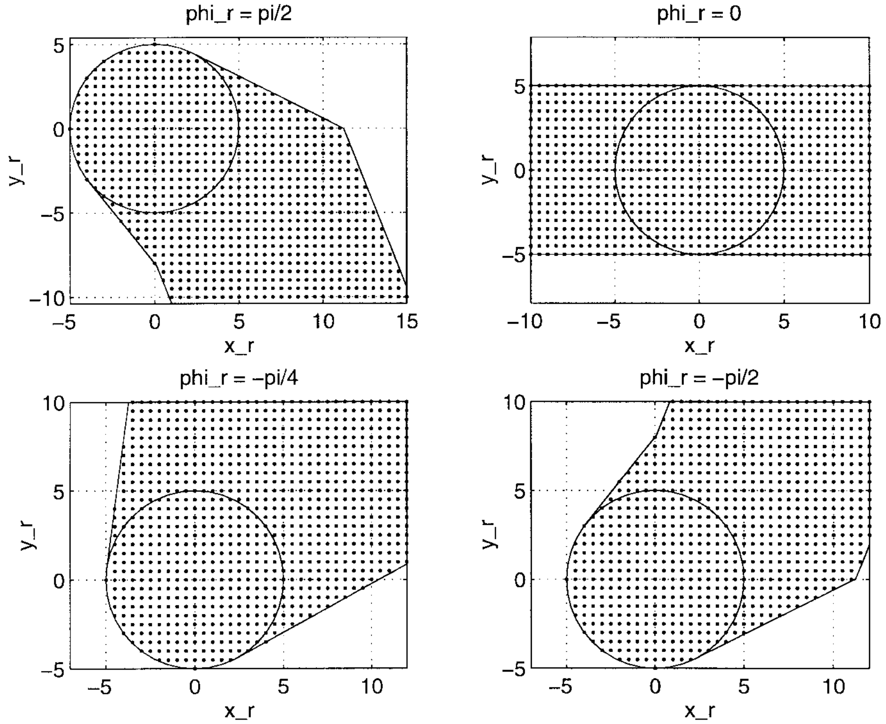


Fig. 7. $\mathrm{Pre}_t(T)$ shown in the (x_r, y_r)-plane for $[\underline{v}_1, \overline{v}_1] = [2, 4]$, $[\underline{v}_2, \overline{v}_2] = [1, 5]$, and $\phi_r = \pi/2, 0, -\pi/4, -\pi/2$.

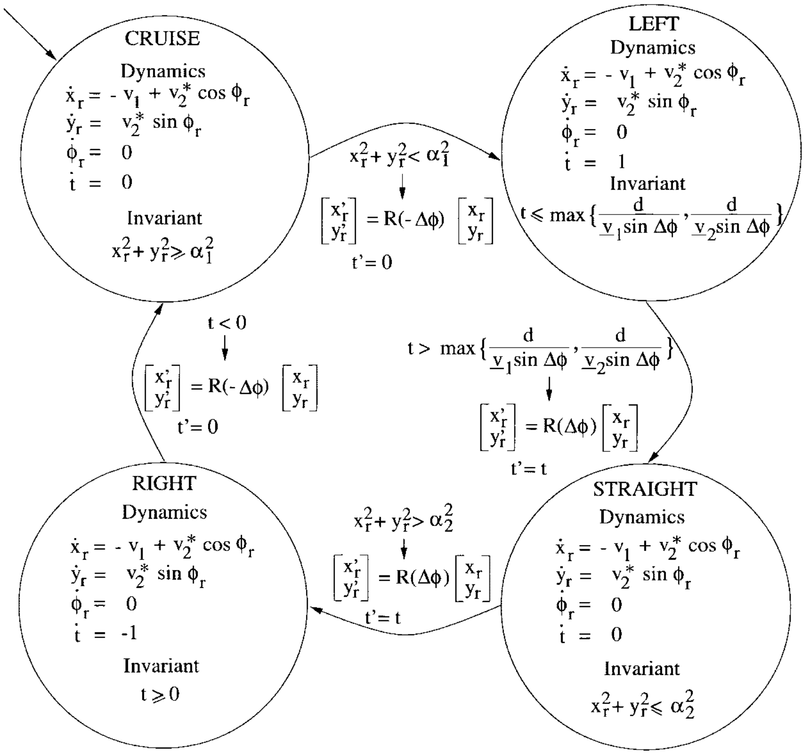


Fig. 8. Modeling conflict resolution maneuver as a hybrid automaton.

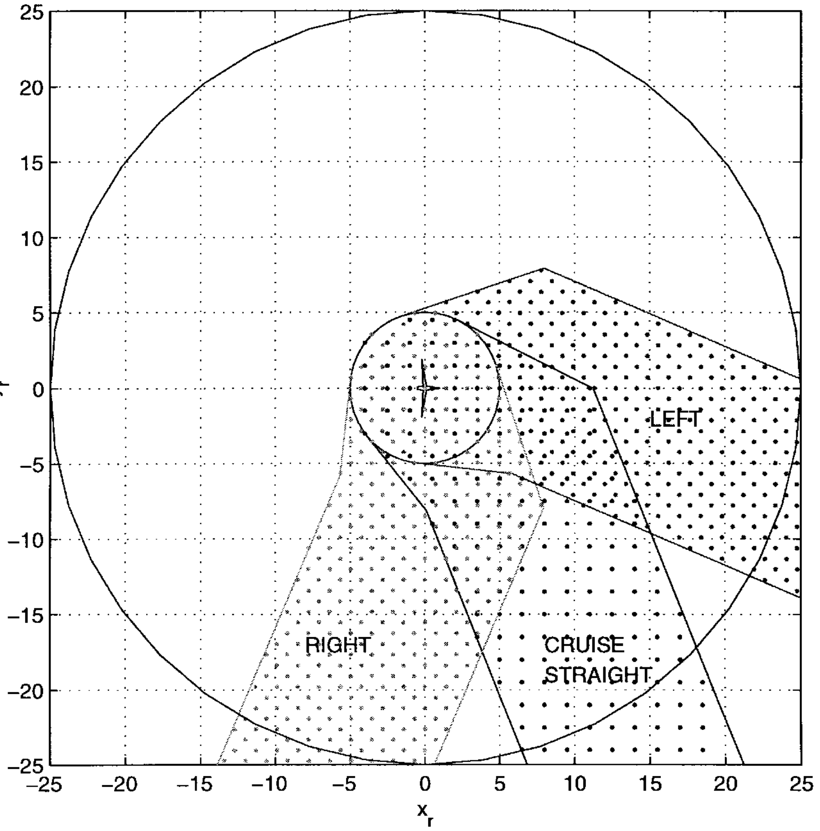


Fig. 9. Computation of $\mathrm{Pre}_{-\infty}(T)$ for each discrete state in $\{CRUISE, LEFT, STRAIGHT, RIGHT\}$.

where r_a is the radius of the alert zone. Thus the aircraft are assumed to be initially cruising and their protected zones do not intersect. The safety specification for H is that the state does not enter T, defined as

$$T = \{CRUISE, LEFT, STRAIGHT, RIGHT\} \times \{x_r^2 + y_r^2 < 5^2\}. \tag{43}$$

Due to uncertainties in the velocity of the other aircraft, the worst case scenario is assumed for v_2, and therefore the dynamics evolve according to the saddle solution (41). This introduces additional switching surfaces within each discrete state.

The automaton of Fig. 8 starts in the $CRUISE$ mode and flows in that state until the interaircraft distance is less than α_1 miles, at which point both aircraft make a heading change of $\Delta\phi$. Discrete heading changes have the effect of resetting the state by a rotation matrix since the coordinate frame depends on the orientation of the aircraft [6], [7]. In mode $LEFT$, both aircraft make a nominal lateral displacement of at least d. This is achieved using a timer variable t as shown. Both aircraft then return to their original heading and cruise until their relative distance is greater than α_2 miles. Once this is achieved, the reverse maneuver is performed in order return to the original cruise path and heading. The heading changes of both aircraft are assumed to occur simultaneously.

For this example, the velocities of the aircraft are chosen to be the same as in the second example of Section III: $v_1 \in [2, 4], v_2 = [1, 5]$, and $\phi_r = \pi/2$. The radius of the relative protected zone is 5 mi while the alert zone has a radius of 25 mi.[1] Instead of fixing values for the parameters d, α_1, α_2, and $\Delta\phi$, we initially leave the first three unrestricted and let $\Delta\phi \in \{-45°, 45°\}$. Their values will be determined in order to minimize the unsafe set of initial states of the maneuver.

Fig. 9 displays the union of each $\mathrm{Pre}_{-\infty}(T)$ calculated within the alert zone for each discrete state $(CRUISE, LEFT, STRAIGHT, RIGHT)$, in the absence of invariants for each discrete state since the parameters d, α_1, and α_2 are unconstrained. The set labeled $CRUISE$ (respectively, $LEFT, STRAIGHT, RIGHT$) displays the set of states which could flow into T under the $CRUISE$ mode (respectively, $LEFT, STRAIGHT, RIGHT$ modes). $\mathrm{Pre}_{-\infty}(T)$ in the $LEFT$ and $RIGHT$ modes are rotations of this set in the $CRUISE$ mode by $-\Delta\phi$, corresponding to aircraft 1 at the origin of the relative frame rotating by $\Delta\phi$. The intersection of the sets $\mathrm{Pre}_{-\infty}(T)$ in this figure represents those states which are unsafe under *all* modes, since outside of this intersection the aircraft may always switch modes to enter a safe region, by choosing appropriate values for the parameters d, α_1, α_2. Fig. 10 displays the minimal unsafe set as a subset of the $\mathrm{Pre}_{-\infty}(T)$ in the $CRUISE$ maneuver (shown as the shaded set). The values of d, α_1, and α_2 are chosen so that the switches between modes occur on the boundary of this minimal set.

[1] The velocities and size of the alert zone are scaled in order to produce visualizable figures.

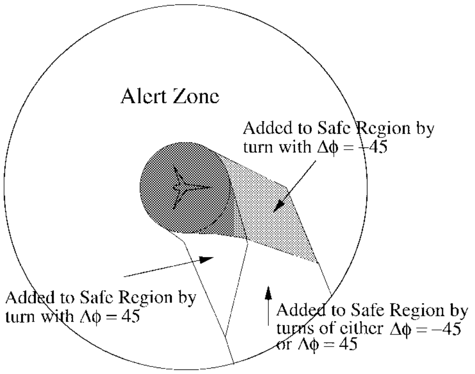

Alert Zone
Added to Safe Region by turn with Δϕ = -45
Added to Safe Region by turn with Δϕ = 45
Added to Safe Region by turns of either Δϕ = -45 or Δϕ = 45

Fig. 10. Partitioning the Alert Zone into safe and unsafe regions.

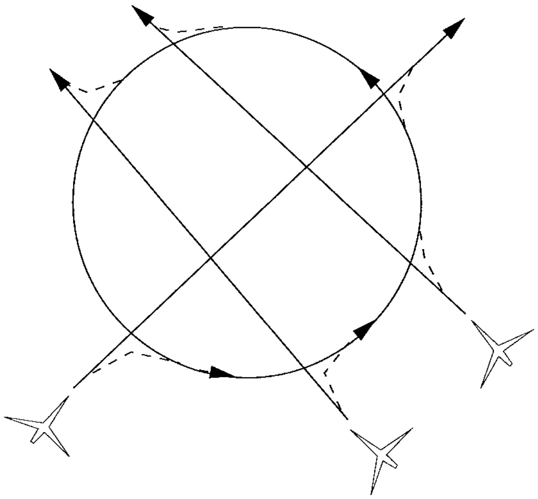


Fig. 11. Conflict Resolution for three aircraft: the Roundabout maneuver.

The type of maneuver that may be verified with this technique can be much more general than that described here: in [45] we construct various parameter-dependent maneuvers for two, three, and four aircraft by using artificial potential field methods from robotic path planning to produce the maneuvers. For three aircraft coming into conflict this approach produces the *Roundabout* maneuver, shown in Fig. 11.

V. Conclusions

In this paper, we have presented a methodology for generating provably safe conflict resolution maneuvers for two aircraft. The method is based on calculating reachable sets for hybrid systems with nonlinear dynamics within each discrete state. The approach allows for uncertainty in the intent of one of the aircraft and calculates the least restrictive control scheme for the other aircraft, based on the worst case uncertainty. This calculation is then used to determine, for a given maneuver with possible variation in its parameters, the minimal unsafe operating region for each aircraft.

Important research issues that we are currently addressing are the computation of numerical solutions to the Hamilton–Jacobi–Isaacs PDE and the efficient representation and manipulation of the reachable sets. The computation of the solution to the Hamilton–Jacobi–Isaacs PDE when $J^*(x,t)$ is *not* a smooth function of x and t is possible, a survey paper [46] presents efficient computation schemes. In addition, we are extending the verification methodology to include lift and drag aerodynamic forces in the dynamics of the aircraft. Some preliminary results in this context have been presented in [43]. Finally, we are investigating *probabilistic verification,* which calculates the probability of a system trajectory entering an unsafe region.

Appendix

Proof of Proposition 1: Starting at time t (free) and integrating to the final time zero, the solution to (38) has $\phi_r(t) = \phi_r(0)$ and

$$\begin{aligned} x_r(0) &= x_r(t) - \int_t^0 u(\tau)\,d\tau + \cos\phi_r \int_t^0 d(\tau)\,d\tau \\ y_r(0) &= y_r(t) + \sin\phi_r \int_t^0 d(\tau)\,d\tau. \end{aligned} \tag{44}$$

Substituting (44) into the cost index (14), (29), and ignoring the constant 5^2 results in

$$\begin{aligned} &J(x,u(\cdot),d(\cdot),t) \\ &\quad = x_r^2(0) + y_r^2(0) \\ &\quad = x_r^2(t) + y_r^2(t) - x_r(t)\int_t^0 u(\tau)\,d\tau \\ &\qquad - x_r(0)\int_t^0 u(\tau)\,d\tau \\ &\qquad + \int_t^0 d(\tau)\,d\tau[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] \\ &\qquad + \int_t^0 d(\tau)\,d\tau[x_r(0)\cos\phi_r + y_r(0)\sin\phi_r]. \end{aligned}$$

Define the *switching functions* $s_1(t), s_2(t)$ as in (39). Consider the case in which, $\forall\ t \leq 0$

$$\operatorname{sgn}(s_1(t)) > 0, \quad \operatorname{sgn}(s_2(t)) > 0.$$

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 $[t, 0]$, 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, i.e., let $u = \underline{v}_1 + \delta v_1$, where $\delta v_1 \geq 0$. Then

$$\begin{aligned} &J(x,u(\cdot),d^*(\cdot),t) \\ &\quad = x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(0-t) - x_r(0)\underline{v}_1(0-t) \\ &\qquad - x_r(t)\int_t^0 \delta v_1(\tau)\,d\tau - x_r(0)\int_t^0 \delta v_1(\tau)\,d\tau \\ &\qquad + \underline{v}_2(0-t)[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] \\ &\qquad + \underline{v}_2(0-t)[x_r(0)\cos\phi_r + y_r(0)\sin\phi_r] \\ &\qquad \leq x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(0-t) - x_r(0)\underline{v}_1(0-t) \\ &\qquad + \underline{v}_2(0-t)[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] \\ &\qquad + \underline{v}_2(0-t)[x_r(0)\cos\phi_r + y_r(0)\sin\phi_r] \\ &\quad = J(x,u^*(\cdot),d^*(\cdot),t). \end{aligned} \tag{45}$$

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

$$\begin{aligned} J(x, u^*(\cdot), d(\cdot), t) \\ &= x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(0-t) - x_r(0)\underline{v}_1(0-t) \\ &\quad + \underline{v}_2(0-t)[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] \\ &\quad + \underline{v}_2(0-t)[x_r(0)\cos\phi_r + y_r(0)\sin\phi_r] \\ &\quad + \int_t^0 \delta v_2(\tau)\, d\tau[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] \\ &\quad + \int_t^0 \delta v_2(\tau)\, d\tau[x_r(0)\cos\phi_r + y_r(0)\sin\phi_r] \\ &\quad \geq x_r^2(t) + y_r^2(t) - x_r(t)\underline{v}_1(0-t) - x_r(0)\underline{v}_1(0-t) \\ &\quad + \underline{v}_2(0-t)[x_r(t)\cos\phi_r + y_r(t)\sin\phi_r] \\ &\quad + \underline{v}_2(0-t)[x_r(0)\cos\phi_r + y_r(0)\sin\phi_r] \\ &= J(x, u^*(\cdot), d^*(\cdot), t). \end{aligned} \tag{46}$$

Summarizing, we have shown above that in this case

$$\begin{aligned} J(x, u(\cdot), d^*(\cdot), t) &\leq J(x, u^*(\cdot), d^*(\cdot), t) \\ &\leq J(x, u^*(\cdot), d(\cdot), t). \end{aligned} \tag{47}$$

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] Honeywell Inc., "Markets report," Tech. Rep. NASA Contract NAS2-114279, Final Rep. for AATT Contract, 1996.

[2] T. S. Perry, "In search of the future of air traffic control," *IEEE Spectrum*, vol. 34, pp. 18–35, 1997.

[3] *Radio Technical Commission for Aeronautics. Final report of RTCA Task Force 3: Free flight implementation*, RTCA, Washington DC, Tech. Rep., Oct. 1995.

[4] Honeywell Inc., "Technology and procedures report," Tech. Rep. NASA NAS2-114279, Final Rep. for AATT Contract, 1996.

[5] S. Kahne and I. Frolow, "Air traffic management: Evolution with technology," *IEEE Contr. Syst. Mag.*, vol. 16, pp. 12–21, Aug. 1996.

[6] W. H. Harman, "TCAS: A system for preventing midair collisions," *Lincoln Lab. J.*, vol. 2, no. 3, pp. 437–457, 1989.

[7] H. Erzberger, T. J. Davis, and S. Green, "Design of center-TRACON automation system," in *Proc. AGARD Guidance and Control Symp. Machine Intelligence in Air Traffic Management*, Berlin, Germany, 1993, pp. 11.1–11.12.

[8] D. J. Brudnicki and A. L. McFarland, "User request evaluation tool (URET) conflict probe performance and benefits assessment," in *Proc. U.S.A./Europe ATM Seminar*, Eurocontrol, Paris, 1997.

[9] L. Yang and J. Kuchar, "Prototype conflict alerting logic for free flight," in *Proc. 35th AIAA Aerospace Sciences Meeting & Exhibit, AIAA 97-0220*, Reno, NV, Jan. 1997.

[10] R. A. Paielli and H. Erzberger, "Conflict probability and estimation for free flight," in *Proc. 35th AIAA Aerospace Sciences Meeting & Exhibit, AIAA 97-0001*, Reno, NV, Jan. 1997.

[11] J. Krozel, T. Mueller, and G. Hunter, "Free flight conflict detection and resolution analysis," in *Proc. AIAA Guidance, Navigation and Control Conf., AIAA-96-3763*, San Diego, CA, Aug. 1996.

[12] Y. Zhao and R. Schultz, "Deterministic resolution of two aircraft conflict in free flight," in *Proc. AIAA Guidance, Navigation and Control Conference, AIAA-97-3547*, New Orleans, LA, Aug. 1997.

[13] J.-H. Oh and E. Feron, "Fast detection and resolution of multiple conflicts for 3-Dimensional free flight," in *Proc. IEEE Conf. Decision and Control*, San Diego, CA, 1997.

[14] R. Slattery and S. Green, "Conflict free trajectory planning for air traffic control automation," NASA Ames Research Center, Moffett Field, CA, Technical Rep. NASA TM-108790, Jan. 1994.

[15] C. Tomlin, G. Pappas, J. Lygeros, D. Godbole, and S. Sastry, "Hybrid control models of next generation air traffic management," in *Hybrid Systems IV*, Lecture Notes in Computer Science 1273, P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, Eds. New York: Springer-Verlag, 1997, pp. 378–404; longer version available as UCB/ERL Memo M97/7.

[16] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds., *Hybrid Systems*, Lecture Notes in Computer Science 736. New York: Springer-Verlag, 1993.

[17] P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, Eds., *Hybrid Systems II*, Lecture Notes in Computer Science 999. New York: Springer-Verlag, 1995.

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

[19] P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, Eds., *Hybrid Systems IV*, Lecture Notes in Computer Science 1273. New York: Springer-Verlag, 1997.

[20] R. Alur and D. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, pp. 183–235, 1994.

[21] R. Alur, C. Courcoubetis, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," in *Proc. 11th Int. Conf. Analysis and Optimization of Systems: Discrete-Event Systems*, Lecture Notes in Control and Information Sciences 199, G. Cohen and J.-P. Quadrat, Eds. New York: Springer-Verlag, 1994, pp. 331–351.

[22] N. Lynch, R. Segala, F. Vaandrager, and H. B. Weinberg, "Hybrid I/O automata," in *Hybrid Systems III*, Lecture Notes in Computer Science 1066. New York: Springer-Verlag, 1996, pp. 496–510.

[23] T. A. Henzinger, "The theory of hybrid automata," in *Proc. 11th Ann. Symp. Logic in Computer Science*. New York: IEEE Computer Soc. Press, 1996, pp. 278–292.

[24] A. Puri and P. Varaiya, "Decidability of hybrid systems with rectangular differential inclusions," in *Proc. 6th Int. Computer Aided Verification Conf., Stanford, CA*, 1994, pp. 95–104.

[25] G. Pappas and S. Sastry, "Toward continuous abstractions of dynamical and control systems," in *Hybrid Systems IV*, Lecture Notes in Computer Science 1273, P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, Eds. New York: Springer-Verlag, 1997, pp. 329–341.

[26] Z. Manna and A. Pnueli, *Temporal Verification of Reactive Systems: Safety*. New York: Springer-Verlag, 1995.

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

[28] H. Wong-Toi, "The synthesis of controllers for linear hybrid automata," in *Proc. IEEE Conf. Decision and Control*, San Diego, CA, 1997.

[29] T. A. Henzinger, P. H. Ho, and H. Wong-Toi, "A user guide to HYTECH," in *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science 1019, E. Brinksma, W. Cleaveland, K. Larsen, T. Margaria, and B. Steffen, Eds. New York: Springer Verlag, 1995, pp. 41–71.

[30] C. Daws, A. Olivero, S. Tripakis, and S. Yovine, "The tool KRONOS," in *Hybrid Systems III, Verification and Control*, Lecture Notes in Computer Science 1066. New York: Springer-Verlag, 1996, pp. 208–219.

[31] N. Bjorner, A. Browne, E. Chang, M. Colon, A. Kapur, Z. Manna, H. Sipma, and T. Uribe, "STeP: The Stanford Temporal Prover (educational release), user's manual," Dept. Computer Science, Stanford Univ., Tech. Rep., STAN-CS-TR-95-1562, 1995.

[32] R. W. Brockett, "Hybrid models for motion control systems," in *Perspectives in Control*, H. Trentelman and J. C. Willems, Eds. Boston, MA: Birkhauser, 1993, pp. 29–54.

[33] M. S. Branicky, "Control of hybrid systems," Ph.D. dissertation, Dept. Electrical Engineering and Computer Sciences, MIT, 1994.

[34] L. Tavernini, "Differential automata and their discrete simulators," *Nonlinear Analysis, Theory, Methods and Applications*, vol. 11, no. 6, pp. 665–683, 1987.

[35] A. Deshpande, "Control of hybrid systems," Ph.D. dissertation, Dept. Electrical Engineering and Computer Sciences, Univ. California, Berkeley, 1994.

[36] J. Lygeros, "Hierarchical, hybrid control of large scale systems," Ph.D. dissertation, Dept. Electrical Engineering and Computer Sciences, Univ. California, Berkeley, 1996.

[37] A. Nerode and W. Kohn, "Models for hybrid systems: Automata, topologies, controllability, observability," in *Hybrid Systems*, Lecture Notes in Computer Science 736, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds. New York: Springer Verlag, 1993, pp. 317–356.

[38] M. Lemmon, J. A. Stiver, and P. J. Antsaklis, "Event identification and intelligent hybrid control," in *Hybrid Systems*, Lecture Notes in Computer Science 736, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds. New York: Springer Verlag, 1993, pp. 268–296.

[39] M. Heymann, F. Lin, and G. Meyer, "Control synthesis for a class of hybrid systems subject to configuration-based safety constraints," in *Hybrid and Real Time Systems*, Lecture Notes in Computer Science 1201, O. Maler, Ed. New York: Springer Verlag, 1997, pp. 376–391.

[40] R. Isaacs, *Differential Games*. New York: Wiley, 1967.

[41] T. Başar and G. J. Olsder, *Dynamic Non-Cooperative Game Theory*, 2nd ed. New York: Academic, 1995.

[42] J. Lygeros, D. N. Godbole, and S. Sastry, "A verified hybrid controller for automated vehicles," Inst. Transportation Studies, Univ. California, Berkeley, Tech. Rep. {UCB-ITS-PRR-97-9}, 1997.

[43] J. Lygeros, C. Tomlin, and S. Sastry, "Multiobjective hybrid controller synthesis," in *Hybrid and Real-Time Systems*, Lecture Notes in Computer Science 1201, O. Maler, Ed. Grenoble, France: Springer-Verlag, 1997, pp. 109–123.

[44] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A case study in multi-agent hybrid systems," Electronics Research Laboratory, Univ. California, Berkeley, Tech. Rep., UCB/ERL M96/38, 1996.

[45] J. Košecká, C. Tomlin, G. Pappas, and S. Sastry, "Generation of conflict resolution maneuvers for air traffic management," in *Proc, Int. Conf. Intelligent Robots and Systems (IROS)*, Grenoble, 1997, pp. 1598–1603.

[46] J. A. Sethian, "Theory, algorithms, and applications of level set methods for propagating interfaces," Center for Pure and Applied Mathematics (PAM-651), Univ. California, Berkeley, Tech. Rep., 1995.

Claire Tomlin (S'94) was born in 1969. She received the B.A.Sc. degree in electrical engineering from the University of Waterloo, Canada, in 1992, the M.Sc. degree in electrical engineering from Imperial College, University of London, U.K., in 1993, and is currently a candidate for the Ph.D. degree in electrical engineering from the University of California, Berkeley.

She was a graduate fellow at Harvard University in 1994, and she has been a Visiting Researcher at NASA Ames Reasearch Center since 1994. Her research interests include hybrid systems, nonlinear control systems, air traffic management, and flight vehicle dynamics and control

Ms. Tomlin is a recipient of the Zonta Amelia Earhart Award for Aeronautics Research (1996–1998) and of the Natural Sciences and Engineering Research Council of Canada 1967 Scholarship (1992).

George J. Pappas (S'90) received the B.S. degree in computer and systems engineering in 1991 and the M.S. degree in computer and systems engineering in 1992, both from Rensselaer Polytechnic Institute, Troy, NY. He is currently a Ph.D. candidate in electrical engineering at the University of California, Berkeley.

His research interests include nonlinear control systems, geometric control theory, and hierarchical and hybrid systems, with applications to air traffic management systems.

Shankar Sastry (S'79–M'80–SM'90–F'95) received the Ph.D. degree in 1981 from the University of California, Berkeley.

He was a member of faculty at the Massachusetts Institute of Technology (MIT), Cambridge, from 1980 to 1982 and the Gordon McKay Professor at Harvard University, Cambridge, in 1994. He is currently a Professor of Electrical Engineering and Computer Sciences and the Director of the Electronics Research Laboratory at the University of California, Berkeley. He has held visiting appointments at the Australian National University, Canberra, the University of Rome, Scuola Normale and the University of Pisa, the CNRS Laboratory LAAS in Toulouse, and as a Vinton Hayes Visiting Fellow at the Center for Intelligent Control Systems at MIT. His research interests include nonlinear and adaptive control, robotic telesurgery, control of hybrid systems, and biological motor control. He is a coauthor of *Adaptive Control: Stability, Convergence and Robustness* (Englewood Cliffs, NJ: Prentice Hall, 1989) and *A Mathematical Introduction to Robotic Manipulation* (New York: CRC, 1994). He has coedited *Hybrid Control II* and *Hybrid Control IV*, Lecture Notes in Computer Science (New York: Springer, 1995, 1997, respectively).

Dr. Sastry was an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE CONTROL MAGAZINE, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and the *Journal of Mathematical Systems, Estimation and Control* and is an Associate Editor of the *IMA Journal of Control and Information, International Journal of Adaptive Control and Signal Processing,* and *Journal of Biomimetic Systems and Materials.* He received the President of India Gold Medal in 1977, the IBM Faculty Development Award for 1983–1985, the NSF Presidential Young Investigator Award in 1985, and the Eckman Award of the American Automatic Control Council in 1990.