Advanced Air Traffic Automation: A Case Study in Distributed Decentralized Control *

C. Tomlin, G. Pappas, J. Košecká, J. Lygeros and S. Sastry

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley, Berkeley, CA 94720
clairet,gpappas,janka,lygeros,sastry@eecs.berkeley.edu

Abstract. In this survey paper, we present some of the issues in designing algorithms for the control of distributed, multi-agent systems. The control of such systems is becoming an increasing issue in many areas owing to technological advances which make it possible to take "legacy" systems to new levels of functioning and efficiency. Of specific interest to us in this paper is advanced air traffic management (ATM) to increase the efficiency and safety of air travel while accommodating the growing demand for air traffic. ATM systems will replace the completely centralized, ground-based air traffic control procedures. Within ATM, the concept of *free flight* allows each aircraft to plan four dimensional trajectories in real time, thus replacing the rigid and inefficient discrete airspace structure. These changes are feasible due to technological innovations such as advanced flight management systems with GPS. In this paper, we propose a *decentralized* ATM architecture, in which some of the current air traffic control functionality is moved on board aircraft. Within this framework, we present the issues in hybrid systems verification and design for safe conflict resolution strategies between aircraft. Both cooperative and noncooperative conflict resolution strategies are presented along with verification methods based on Hamilton-Jacobi theory, automata theory, and the theory of games.

1 New Challenges: Intelligent Multi-Agent Systems

To a large extent, control theory has investigated the very important paradigm of Central Control. In this paradigm, sensory information is collected from sensors observing a material process that may be distributed over space. This information is transmitted over a communication network to one center, where the commands that guide the process are calculated and transmitted back to the process actuators that implement those commands. In engineering practice, of course, as soon as the process becomes even moderately large, the Central Control paradigm breaks down. What we find instead is distributed control: a set of control stations, each of which receives some data and calculates some of the actions. Important examples of distributed control are air traffic management, the

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

control system of an interconnected power grid, the telephone network, a chemical process control system. Although a Central Control paradigm no longer applies here, control engineers have with great success used its theories and its design and analysis tools to build and operate these distributed control systems. There are two reasons why the paradigm succeeded in practice, even when it failed in principle. First, in each case the complexity and scale of the material process grew incrementally and relatively slowly. Each new increment to the process was controlled using the paradigm, and adjustments were slowly made after extensive (but by no means exhaustive) testing to ensure that the new controller worked in relative harmony with the existing controllers. Second, the processes were operated with a considerable degree of “slack.” That is, the process was operated well within its performance limits to permit errors in the extrapolation of test results to untested situations and to tolerate a small degree of disharmony among the controllers. However, in each system mentioned above, there were occasions when the material process was stressed to its limits and the disharmony became intolerable, leading to a spectacular loss of efficiency. For example, most air travelers have experienced delays as congestion in one part of the country is transmitted by the control system to other parts. The distributed control system of the interconnected power grid has sometimes failed to respond correctly and caused a small fault in one part of a grid to escalate into a system-wide blackout.

We are now attempting to build control systems for processes that are vastly more complex or that are to be operated much closer to their performance limits in order to achieve much greater efficiency of resource use. The attempt to use the central control paradigm cannot meet this challenge: the material process is already given and it is not practicable to approach its complexity in an incremental fashion as before. Moreover, the communication and computation costs in the central control paradigm would be prohibitive, especially if we insist that the control algorithms be fault-tolerant. What we need to meet the challenge of control design for a complex, high performance material process, is, we believe, a new paradigm for distributed control. It must distribute the control functions in a way that avoids the high communication and computation costs of central control, at the same time that it limits complexity. The distributed control must, nevertheless, permit centralized authority over those aspects of the material process that are necessary to achieve the high performance goals. We believe that such a challenge can be met by organizing the distributed control functions in a hierarchical architecture that makes those functions relatively autonomous (which permits using all the tools of central control), while introducing enough coordination and supervision to ensure the harmony of the distributed controllers necessary for high performance.

1.1 Analysis and Design of Multi-agent Hybrid Control Systems

One of the main incentives to move into the area of multi-agent large scale systems is economic. Preliminary studies indicate that automation can improve coordination in air traffic management systems, highway systems, chemical process control, power generation and distribution, etc. This in turn leads to performance improvement in terms of fuel consumption, safety, efficiency, and environmental impact. To deal with complex systems, engineers use a combination of continuous and discrete controllers. Continuous controllers are used primarily because interaction with the physical plant, through sensors and actuators, is essentially analog, and continuous models and design techniques have been developed, used, and validated extensively. An equally compelling case exists for discrete controllers: since discrete abstractions make it easier to manage system complexity, discrete models are easier to manipulate, and discrete abstractions more naturally accommodate linguistic and qualitative information in the controller design. We will use the term "hybrid systems" to describe systems that incorporate both continuous and discrete dynamics.

Multi-Agent Scarce Resource Systems

An important class of systems that are well suited for hybrid control are multi-agent, scarce resource systems. Their common characteristic is that many agents are trying to make use of a common, congestible resource. For example, in highway systems, the vehicles are agents competing for scarce highway space-time resources, while in air traffic management systems the aircraft compete for air space and runway space. To achieve the common optimum we should ideally have a centralized control scheme that computes the global optimum and commands the agents accordingly. A solution like this may be undesirable, however, for several reasons:

1. it is likely to be very computationally intensive, as a large centralized computer is needed to make all the decisions;
2. it may be less reliable, as the consequences may be catastrophic if the centralized controller is disabled;
3. the information that needs to be exchanged may be too expensive; and
4. the number of agents may be large and/or dynamically changing.

If the performance degradation of a completely decentralized solution is unacceptable and a completely centralized solution is prohibitively complex or expensive, a compromise will have to be found. Such a compromise will feature semi-autonomous agent operation. In this case, each agent is trying to optimize its own usage of the resource and coordinates with "neighboring" agents in case there is a conflict of objectives. It should be noted that *semi-autonomous agent control is naturally suited for hybrid designs.* At the continuous level, each agent chooses its own optimal strategy, while discrete coordination is used to resolve conflicts. Thus, the class of hybrid systems that we will be most interested in is multi-agent systems, where the hybrid dynamics arise from the interaction between continuous single agent "optimal" strategies and discrete conflict resolution or coordination protocols.

We have been involved in such a research program at Berkeley bringing to bear tools from control, robotics, and artificial intelligence into this framework. In addition to synthesizing diverse approaches and experiences into a unified paradigm, we will confirm or validate this new paradigm by using it for controlling our test processes. Thus, our program follows the classical pattern of scientific progress: the first phase of "induction" or the integration of approaches and experiences that go beyond the current practice into a new paradigm which subsumes the current one; and the second phase of "deduction" or the application of the new paradigm to concrete situations to test its validity. We have been guided in our choice of problems by a number of detailed case studies of large, complex systems with multiple agents arising in intelligent vehicle highway systems, air traffic management systems, and intelligent telemedicine.

In this paper, we will give the details of the broad program discussed above in the context of **air traffic management systems**. This is an area of great commercial and technological importance which has unfortunately not yet received the level of attention that it deserves from the research community and exemplifies the broad issues discussed thus far. Section 2 gives a brief background of ATM. In Section 3 we discuss the architectural issues regarding ATM. Sections 4 presents our view of ground and on-board air automation systems in the proposed distributed ATM system. In Section 5, we present hybrid system issues which arise in non-cooperative and cooperative conflict resolution. In particular, we discuss our approach to the design and verification of hybrid systems using Hamilton Jacobi theory, automata theory, and the theory of games. Some concluding remarks are in Section 6.

2 Introduction to Air Traffic Management

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 [1]. The current National Airspace System (NAS) architecture and air traffic management will not be able to efficiently handle this increase because of several limiting factors including inefficient airspace utilization, increased Air Traffic Control (ATC) workload, and out of date technology. 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* [2]. Free Flight allows pilots to choose their own routes, altitude and speed and gives each aircraft the freedom to optimize their routes based on criteria such as fuel consumption, avoidance of bad weather and other factors, referred to as User Preferred Routing or UPR. 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).

The economic benefits of Free Flight are immediate. Direct great circle routes, optimal altitudes, optimal avoidance of developing weather hazards and utilization of favorable winds will result in fuel burn and flight time operating cost savings. NASA studies [3] estimate that in a free flight scenario, user preferred trajectories could have resulted in annual potential savings of \$1.28 billion in 1995 and could result in \$1.47 billion savings in 2005[2]. Free Flight is poten-

[2] Using forecasted air traffic demand for 2005

tially feasible because of enabling technologies such as Global Positioning Systems (GPS), Datalink communications [4], Automatic Dependence Surveillance-Broadcast (ADS-B) [4], Traffic Alert and Collision Avoidance Systems (TCAS) [5] and powerful on-board computation. In addition, tools such as the Center-TRACON Automation System (CTAS) [6] 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 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. 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. The resulting air traffic management system requires coordination and control of a large number of semi-autonomous aircraft. The number of control decisions that have to be made and the complexity of the resulting decision process dictates a hierarchical, decentralized solution. Complexity management is achieved in a hierarchy by moving from detailed, decentralized models at the lower levels to abstract, centralized models at the higher levels. 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, the so called *hybrid systems*. Hybrid systems also arise in the operation of a single aircraft because of *flight mode switching*. The use of discrete modes to describe phases of the aircraft operation is a common practice for pilots and autopilots and is dictated partly by the aircraft dynamics themselves. The modes may reflect, for example, changes in the outputs that the controller is asked to regulate: depending on the situation, the controller may try to achieve a certain airspeed, climb rate, angle of attack, etc. or combinations of those. We do not discuss these further in this paper but refer the reader to [7].

3 A Distributed Decentralized ATM

One of the most important conceptual issues to be addressed in the architecture of large scale control systems is their degree of decentralization. Completely decentralized systems are inefficient and lead to conflict, while completely centralized ones are not tolerant of faults in the central controller, are computationally and conceptually complicated, and are slow to respond to emergencies.

The tradeoff between centralized and decentralized decision making raises a fundamental issue that has to be addressed by any proposed ATM. The current

ATC system is primarily centralized; all safety critical decisions are taken centrally (at the ATC units) and distributed to the aircraft for execution. Because of the complexity of the problem and the limited computational power (provided primarily by the human operators in the current system) this practice may lead to inefficient operation.

A number of issues should be considered when deciding on the appropriate level of centralization. An obvious one is the *optimality* of the resulting design. Even though optimality criteria may be difficult to define for the air traffic problem it seems that, in principle, the higher the level of centralization the closer one can get to the globally optimal solution. However, the complexity of the problem also increases in the process; to implement a centralized design one has to solve a small number of complex problems as opposed to large number of simple ones. As a consequence the implementation of a centralized solution requires a greater effort on the part of the designer to produce control algorithms and greater computational power to execute them. One would ideally like to reach a compromise that leads to acceptable efficiency while keeping the problem tractable.

Another issue that needs to be considered is *reliability* and *scalability*. The greater the responsibility assigned to a central controller the more dramatic are likely to be the consequences if this controller fails. In this respect there seems to be a clear advantage in implementing a decentralized design: if a single aircraft's computer system fails, most of the ATM system is still intact and the affected aircraft may be guided by voice to the nearest airport. Similarly, a distributed system is better suited to handling increasing number of aircraft, since each new aircraft can easily be added to the system, its own computer contributing to the overall computational power. A centralized system on the other hand would require regular upgrades of the ATC computers. This may be an important feature given the current rate of increase of the demand for air travel.

Finally, the issue of *flexibility* should also be taken into account. A decentralized system will be more flexible from the point of view of the agents, in this case the pilots and airlines. This may be advantageous for example in avoiding turbulence or taking advantage of favorable winds, as the aircraft will not have to wait for clearance from ATC to change course in response to such transients or local phenomena. Improvements in performance may also be obtained by allowing aircraft to individually fine tune their trajectories making use of the detailed dynamical models contained in the autopilot. Finally, greater flexibility may be preferable to the airlines as it allows them to utilize their resources in the best way they see fit.

The focus of our research has been to strike a compromise in the form of partially decentralized control laws for guaranteeing *reliable, safe control of the individual agents* while providing *some measure of unblocked, fair, and optimum utilization of the scarce resource. In our design paradigm, agents have control laws which 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.* In the next section we present a control architecture that implements what we believe is a reasonable balance between complete centralization and complete decentralization.

4 Advanced Air Transportation Architectures

This section describes the balance between the ATM on the ground and in the air. Currently, ATC in the United States is organized hierarchically with a single *Air Traffic Control System Command Center (ATCSCC)* supervising the overall traffic flow management. This is supported by 20 *Air Traffic Control System Command Centers (ARTCCs)*, or simply Centers, organized by geographical area. Coastal Centers have jurisdiction over oceanic waters. For example, the Fremont (California) ARTCC has jurisdiction from roughly Eureka to Santa Barbara and from Japan in the West to the Sierra Nevada mountains in the East. In addition, Around large urban airports there are *Terminal Radar Approach Control facilities (TRACONs)* numbering over 150. For instance, the Bay Area TRACON includes the San Francisco, Oakland and San Jose airports along with smaller airfields at Moffett Field, San Carlos, Fremont, etc. The TRACONs are supported by control towers at more than 400 airports. There are roughly 17,000 landing facilities in the United States serving nearly 220,000 aircraft. Of these the commercial aircraft number about 6,000 and the number of commercially used airstrips is roughly the 400 that have control towers. The overall system is referred to as *National Airspace System (NAS)* [8]. The main goal of both the ARTCCs and the TRACONs is to maintain safe separation between aircraft while guiding them to their destinations.

4.1 Automation on the Ground

In an effort to increase the runway throughput, airport capacity as well as reduce delays, fuel consumption and controller workload in the vicinity of highly congested urban airports, NASA has designed the Center-TRACON Automation System (CTAS) [6]. CTAS is a collection of planning and control functions which generate advisories to assist, but not replace, the controllers in handling traffic in the Center and TRACON areas. CTAS consists of three main components: the *Traffic Management Advisor (TMA)*, the *Descent Advisor (DA)* and the *Final Approach Spacing Tool (FAST)*. TMA and DA coexist and operate in Center airspace whereas FAST operates as a standalone in TRACON airspace. CTAS receives input from radar sensors which transmit the aircraft state; from Center and TRACON controllers who allocate runways and routes to particular aircraft as well as alter the capacity or acceptance rate of the TRACON, airport or runway; and finally from weather reports which include wind, temperature and pressure profiles. The main of outputs of CTAS are arrival schedules which meet all the capacity, separation and flow rate constraints as well as advisories to Center or TRACON controllers. CTAS is currently being field tested at Denver and Dallas-Fort Worth. A similar ground system called User Request Evaluation

Tool (URET) has been developed by MITRE Corp. [9] and is being field tested at Indianapolis.

In our proposed ATM system, we will assume that a ground system (either CTAS or URET) will have jurisdiction over highly congested TRACON airspace, that airspace structure exists inside the TRACON and that controllers have active control over aircraft in the TRACON, sending the aircraft heading, speed and altitude advisories. The advisories provide a suggested arrival schedule at the destination airport, which is designed to meet the announced arrival times while resolving conflicts. The schedule reflects compromises between airline schedules as well as possible negotiation between ATC and the aircraft.

4.2 Automation in the Air

In the less congested Center airspace, aircraft are allowed to choose their own routes in the spirit of Free Flight. In addition, aircraft may resolve potential conflicts by inter-aircraft coordination. The role of the ATC in Center airspace is limited to performing flow management, providing the aircraft with global information about en-route traffic and weather conditions, as well as providing advisories in case aircraft are unable to resolve conflicts on their own. Currently, nominal trajectories through the airspace are defined in terms of *waypoints*, which are fixed points in the airspace defined by VOR (VHF Omni-Directional Range) points on the ground. The waypoints are a necessary navigation tool for aircraft which are not equipped with GPS. Waypoints have resulted in a discrete airspace structure and an underutilization of airspace. On the other hand, they have resulted in a predictable environment which allows controllers to resolve conflicts in congested airspace. GPS and Free Flight will remove this structure which will lead to greater efficiency and airspace capacity. Aircraft may choose their own routes instead of following a sequence of waypoints. However, inside the crowded TRACONs, airspace structure will be necessary in order to simplify the controller's task of landing aircraft while resolving conflicts.

In our proposed ATM structure, each aircraft is equipped with various planning and control algorithms. The aircraft will perform real time trajectory planning and tracking, conflict detection and resolution, as well as automatic mode switching. These smart aircraft will be extremely complex and each will be a large scale system in its own right. In order to reduce the resulting complexity and assist pilots in better performing their task, each aircraft is modeled using the hierarchical structure shown in Figure 1. The levels of architecture below ATC reside on the aircraft and comprise what is known as the aircraft's *Flight Management System*, or FMS. The FMS consists of four layers, the strategic, tactical, and trajectory planners, and the regulation layer. Higher levels of the FMS architecture are associated with higher objectives and coarser models. Each layer of this architecture is described below:

Strategic Planner: The main objectives of the strategic planner are to design a coarse trajectory for the aircraft and to resolve conflicts between aircraft. The

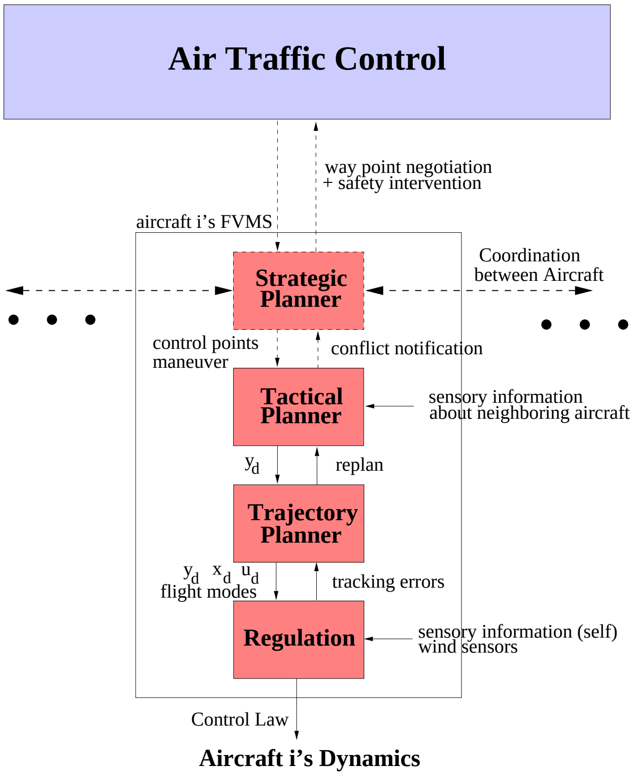

Fig. 1. Proposed ATM Structure

trajectory is designed from origin to destination in some optimal sense, and is frequently redesigned in order to adapt to changes in the environment, such as weather patterns, potential conflicts and airport traffic. Inside TRACONs, the strategic planner simply accepts the advisories of the controllers. In Center airspace, the strategic planners of all aircraft involved in the potential conflict determine a sequence of maneuvers which will result in conflict-free trajectories, either using communication with each other through satellite datalink, or by calculating safe trajectories assuming the worst possible actions of the other aircraft [10]. Each strategic planner sends its most recently designed trajectory to the tactical planner in the form of a sequence of control points and/or a maneuver.

Tactical Planner: The tactical planner refines the strategic plan by interpolating the control points with a smooth output trajectory, denoted by y_d in Figure 1. The tactical planner uses a simple kinematic model of the aircraft for all trajectory calculations. Simple models are used at this stage since very detailed models may unnecessarily complicate the calculations, which are assumed to be approximate and have large safety margins. The output trajectory is then passed to the trajectory planner.

Trajectory Planner: The trajectory planner uses a detailed dynamic model of the aircraft, sensory data about the wind magnitude and direction, and the tactical plan consisting of an output trajectory, to design full state and input trajectories for the aircraft, and a sequence of *flight modes* necessary to execute the dynamic plan. The flight modes represent different modes of operation of the aircraft and correspond to controlling different variables in the aircraft dynamics. A derivation of the flight mode logic necessary for safe operation of a CTOL (Conventional Take Off and Landing) aircraft is presented in [11].

The resulting trajectory, denoted y_d, x_d, and u_d in Figure 1, is given to the regulation layer which directly controls the aircraft. The task of the trajectory planner is complicated by the presence of non-minimum phase dynamics [12], [13] and actuator saturation [14].

Regulation Layer: Once a feasible dynamic trajectory has been determined, the regulation layer is asked to track it. Assuming that the aircraft dynamic model used by the trajectory planner is a good approximation of the true dynamics of the aircraft, tracking should be nearly perfect. In the presence of large external disturbances (such as wind shear or malfunctions), however, tracking can severely deteriorate. The regulation layer has access to sensory information about the actual state of the aircraft dynamics, and can calculate tracking errors. These errors are passed back to the trajectory planner, to facilitate replanning if necessary.

The structure of the proposed Flight Management System leads to various interesting questions regarding hierarchical systems. First, the convergence of the overall scheme to an acceptable and safe trajectory needs to be shown. Due to the complexity of the overall system and very nonlinear nature of the continuous dynamics it is unlikely that purely continuous or purely discrete techniques alone will be adequate in this setting. More elaborate hybrid techniques are needed. In addition, higher levels of the hierarchy use coarser system models or coarser abstractions. This raises the interesting notions of consistent abstractions or implementability, which is the ability of a lower level system to execute the commands of a higher level system. Preliminary work along this direction may be found in [15].

5 Conflict Resolution

The operation of the proposed ATM involves the interaction of continuous and discrete dynamics. Such *hybrid* phenomena arise, for example, from the coordination between aircraft at the strategic level when resolving a potential conflict. The conflict resolution maneuvers are implemented in the form of discrete communication protocols. These maneuvers appear to the (primarily continuous) tactical planner as discrete resets of the desired waypoints. One would like to determine the effect of these discrete changes on the continuous dynamics (and vice versa) and ultimately obtain guarantees on the minimum aircraft separation possible under the proposed control scheme.

Research in the area of conflict detection and resolution for air traffic has been centered on predicting conflict and deriving maneuvers assuming that the intent of each aircraft is known to all other aircraft involved in the conflict, for both deterministic [16],[17],[18], and probabilistic [19],[20] models.

In our research, we differentiate between two types of conflict resolution: *noncooperative* and *cooperative* [10]. In noncooperative conflict resolution, each aircraft involved in the conflict derives a safe avoidance maneuver without coordinating with the other aircraft. Such a situation occurs when there is an emergency and there is not enough time to establish communication with other aircraft, as was encountered by Air Force I with a United Parcel Service aircraft over the coast of Ireland in June 1997. The safest action that this aircraft can take is to choose a strategy which resolves the conflict for any possible action, within bounds, of the other aircraft. We formulate the noncooperative conflict resolution strategy as a zero sum dynamical game of the pursuit-evasion style [21]. The aircraft are treated as players, aware only of the *set* of possible actions of the other agents. These actions are modeled as disturbances, assumed to lie within a known set but with their particular values unknown, and the aircraft solves the game for the worst possible disturbance. The performance index for the game is the relative distance between the aircraft, required to be above a certain threshold (the Federal Aviation Administration requires a 5 mile horizontal separation in en-route airspace). Assuming that a saddle solution to the game exists, the saddle solution is *safe* if the performance index evaluated at the saddle solution is above the required threshold. The sets of *safe states* and *safe control actions* for each aircraft may be calculated: the saddle solution defines the boundaries of these sets. The aircraft may choose any trajectory in its set of safe states, and a control policy from its set of safe control actions; coordination with the other aircraft is unnecessary.

The model used is a relative kinematic model for two aircraft, aircraft 1 and aircraft 2, which describes the motion of aircraft 2 with respect to aircraft 1:

$$\begin{aligned} \dot{x}_r &= -v_1 + v_2 \cos\phi_r + \omega_1 y_r \\ \dot{y}_r &= v_2 \sin\phi_r - \omega_1 x_r \\ \dot{\phi}_r &= \omega_2 - \omega_1 \end{aligned} \tag{1}$$

in which (x_r, y_r, ϕ_r) is the relative position and orientation of aircraft 2 with respect to aircraft 1, and v_i and ω_i are the linear and angular velocities of each aircraft.

In cooperative conflict resolution, safety is ensured by full coordination among the aircraft. The aircraft follow predefined maneuvers, inspired by robot collision avoidance maneuvers, which are proven to be safe. The class of maneuvers constructed to resolve conflicts must be rich enough to cover all possible conflict scenarios. In this case, the predefined resolution protocols dictate a hybrid nature in the overall system.

We will discuss these two scenarios in some detail now.

5.1 Noncooperative Conflict Resolution

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

$$\dot{x} = f(x, u, d) \qquad x(t) = x \tag{2}$$

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 (2), 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 2).

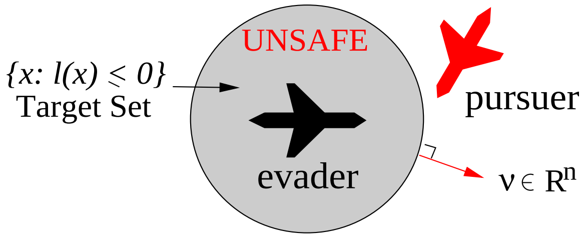

Fig. 2. The evader and pursuer, with Target set and its outward pointing normal ν

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 (2) over the time interval $[t, t_f]$, where t_f is defined as

$$t_f = \inf\{\tau \in \mathbb{R}^+ \mid x(\tau) \in T\} \tag{3}$$

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:

$$J_1(x, t, u, d) = l(x(t_f)) \tag{4}$$

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

$$\nu = \frac{\partial l}{\partial x}(x(t_f)) \tag{5}$$

as in Figure 2 which allows us to define

$$\begin{array}{ll} \text{Safe portion of } \partial T & \{x \in \partial T : \exists u \forall d \quad \nu^T f(x, u, d) \geq 0\} \\ \text{Unsafe portion of } \partial T & \{x \in \partial T : \forall u \exists d \quad \nu^T f(x, u, d) < 0\} \end{array} \tag{6}$$

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 $J_1(x, t, u, d)$:

$$u^* = \arg\max_{u \in \mathcal{U}} J_1(x, t, u, d) \tag{7}$$

$$d^* = \arg\min_{d \in \mathcal{D}} J_1(x, t, u, d) \tag{8}$$

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:

$$J_1^*(x, t) = \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} J_1(x, t, u, d) = \min_{d \in \mathcal{D}} \max_{u \in \mathcal{U}} J_1(x, t, u, d) \tag{9}$$

The concept of a saddle solution is key to our computation of the safe regions of operation of the aircraft, since a solution of (2) 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

$$V_1 = \{x \in \mathbb{R}^n \backslash T \mid \exists u \in \mathcal{U}, J_1(x, t, u, d) \geq 0, \forall d \in \mathcal{D}\} \tag{10}$$

$$= \{x \in \mathbb{R}^n \backslash T \mid \exists u \in \mathcal{U}, J_1(x, t, u, d^*) \geq 0\} \tag{11}$$

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:

$$H^*(x,p) = \max_{u \in U} \min_{d \in D} H(x,p,u,d) = H(x,p,u^*,d^*) \tag{12}$$

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

$$\begin{aligned} \dot{x} &= \frac{\partial H^*}{\partial p}(x,p) \\ \dot{p} &= -\frac{\partial H^*}{\partial x}(x,p) \end{aligned} \tag{13}$$

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:

$$\frac{\partial J_1^*(x,t)}{\partial t} = -H^*(x, \frac{\partial J_1^*(x,t)}{\partial x}) \tag{14}$$

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 (14). However, it is difficult to guarantee that the PDE (14) 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 (14), 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 (14) to determine the safe and unsafe sets over the state space $\mathbb{R}^n$ (see Figure 3).

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 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

$$\mathcal{U}_1(x) = \{u \in \mathcal{U} \mid J_1(x,t,u,d) \geq 0, \forall d \in \mathcal{D}\} \tag{15}$$

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:

$$\min_{u \in \mathcal{U}_1} J_2 \tag{16}$$

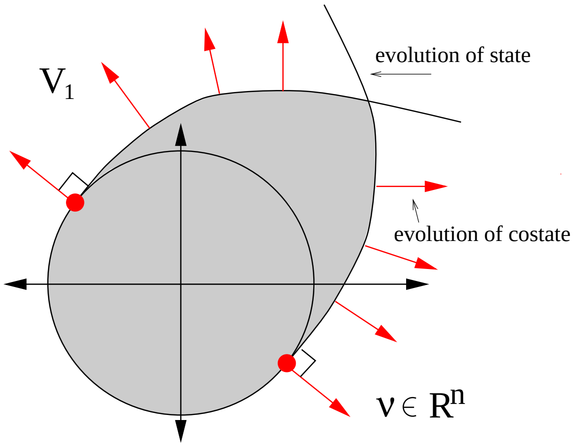

Fig. 3. The unsafe set of states (shaded) and its complement (the safe set V_1)

subject to the original differential equations (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 [11].

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 the model (1) becomes:

$$\begin{aligned} \dot{x}_r &= -v_1 + v_2 \cos\phi_r + u y_r \\ \dot{y}_r &= v_2 \sin\phi_r - u x_r \\ \dot{\phi}_r &= d - u \end{aligned} \tag{17}$$

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:

$$T = \{(x_r, y_r) \in \mathbb{R}^2, \phi_r \in [-\pi, \pi) \mid x_r^2 + y_r^2 \leq 5^2\} \tag{18}$$

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

$$l(x) = x_r^2 + y_r^2 - 5^2 \tag{19}$$

The optimal Hamiltonian is

$$H^*(x,p) = \max_{u\in U} \min_{d\in D} [-p_1v_1 + p_1v_2\cos\phi_r + p_2v_2\sin\phi_r + (p_1y_r - p_2x_r - p_3)u + p_3d] \tag{20}$$

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

$$\begin{aligned} s_1(t) &= p_1(t)y_r(t) - p_2(t)x_r(t) - p_3(t) \\ s_2(t) &= p_3(t) \end{aligned} \tag{21}$$

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

$$\begin{aligned} u^* &= sgn(s_1) \\ d^* &= -sgn(s_2) \end{aligned} \tag{22}$$

The equations for $\dot{p}$ are obtained through (13) and are

$$\begin{aligned} \dot{p}_1 &= u^*p_2 \\ \dot{p}_2 &= -u^*p_1 \\ \dot{p}_3 &= p_1v_2\sin\phi_r - p_2v_2\cos\phi_r \end{aligned} \tag{23}$$

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

$$-v_1x_r + v_2(x_r\cos\phi_r + y_r\sin\phi_r) < 0 \tag{24}$$

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

$$-v_1x_r + v_2(x_r\cos\phi_r + y_r\sin\phi_r) = 0 \tag{25}$$

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 (22) 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 [22] (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

$$\dot{s}_1 = y_rv_1 \tag{26}$$

$$\dot{s}_2 = x_rv_2\sin\phi_r - y_rv_2\cos\phi_r \tag{27}$$

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

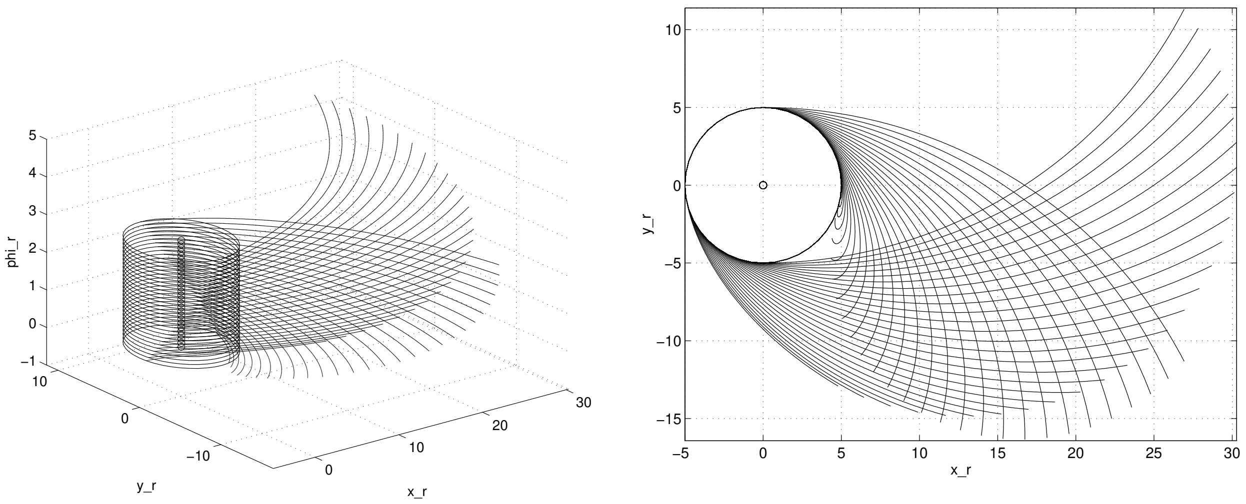

Fig. 4. The Target set $T = \{(x_r, y_r), \phi_r \in (0, \pi) \mid x_r^2 + y_r^2 \leq 5^2\}$ (cylinder) and the boundary of the safe set V_1 for $t \leq t_f$ until the first switch in either $s_1(t)$ or $s_2(t)$. The unsafe set is enclosed by the boundary. The second picture is a top view of the first.

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 4 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 5. 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 4 intersect. At the intersection, it is not clear that u^* is the unique safe input.

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 (1) reduces to:

$$\begin{aligned} \dot{x}_r &= -u + d\cos\phi_r \\ \dot{y}_r &= d\sin\phi_r \\ \dot{\phi}_r &= 0 \end{aligned} \tag{28}$$

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^*)

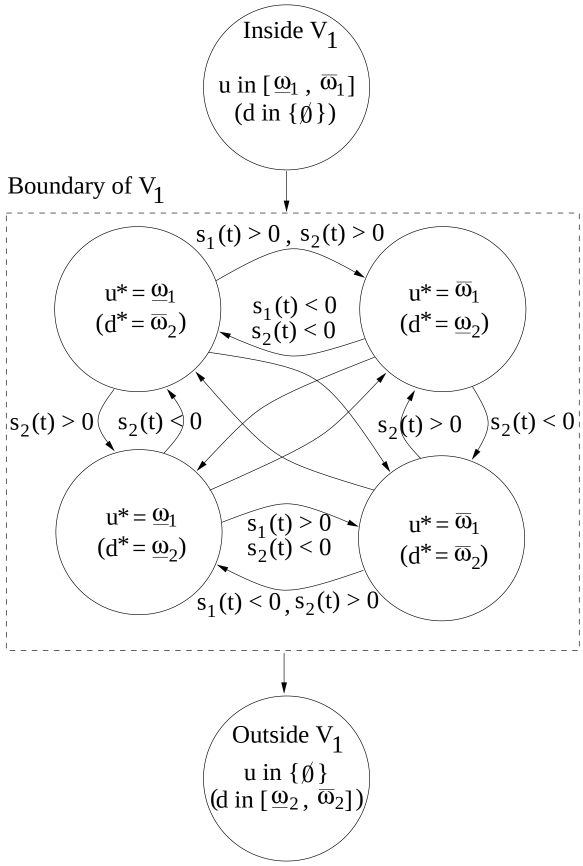

Fig. 5. Switching law governing the two aircraft system with angular velocity control inputs. The law is least restrictive in that the control u is not restricted when the state is inside V_1. The diagonal transitions in the automaton for the boundary of V_1 are not labeled for legibility. This automaton 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.

directly, by integrating equations (28) for piecewise constant u and d, and substituting the solutions into the cost function (4). To do this we first define the switching functions s_1 and s_2 as

$$\begin{aligned} s_1(t) &= x_r \\ s_2(t) &= x_r \cos\phi_r + y_r \sin\phi_r \end{aligned} \tag{29}$$

Proposition 1. [Saddle Solution for Linear Velocity Controls] *The global saddle solution (u^*, d^*) to the game described by system (28) for the cost $J_1(x, t, u, d)$*

given by equation (4) is

$$u^* = \begin{cases} \underline{v}_1 \text{ if } sgn(s_1) > 0 \\ \overline{v}_1 \text{ if } sgn(s_1) < 0 \end{cases} \tag{30}$$

$$d^* = \begin{cases} \underline{v}_2 \text{ if } sgn(s_2) > 0 \\ \overline{v}_2 \text{ if } sgn(s_2) < 0 \end{cases} \tag{31}$$

As can be seen from equation (30), 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 6, which describes the least restrictive control scheme for safety. The unsafe sets of states are illustrated in Figure 7 for various values of ϕ_r, and speed ranges as illustrated.

5.2 Cooperative Conflict Resolution

In *cooperative conflict resolution* the aircraft coordinate amongst themselves if trajectory conflicts occur, and perform predefined, *a-priori safe* maneuvers in order to resolve the conflict. The class of maneuvers constructed to resolve conflicts must be rich enough to cover most possible conflict scenarios. Examples of *head-on* and *overtake* maneuvers can be seen in Figure 8 and 9.

In order to construct a complete set of collision avoidance maneuvers which cover general conflict scenarios involving more than two agents, there is a need to classify all kinds of possible collisions, and the maneuvers to resolve these. To do this, we use the strategy outlined in Figure 10. First we employ the automated method of potential field based motion planning to generate "prototype maneuvers", which inspire the actual collision avoidance maneuver. We believe that distributed on-line motion planning techniques and their application to ATM can be inspirational for deriving a set of possible maneuvers for collision avoidance between aircraft. In spite of the fact that the feasibility of the individual trajectories can be asserted by simulations, the proof of the safety of the maneuvers for dynamic models of aircraft remains a challenging problem. It is for this reason that we wish to construct the simplest possible maneuvers from the prototypes, those made up of straight lines. This discretized version of the maneuvers can be modeled as a hybrid system and can be proven to be safe using hybrid verification techniques. The verification step is crucial for building an off-line database of safe conflict resolution maneuvers.

Robot Collision Avoidance: There is a large number of theoretical studies in the classical motion planning literature regarding the multiple robot planning problem. Algorithms embedded in time-extended configuration space [23] prove

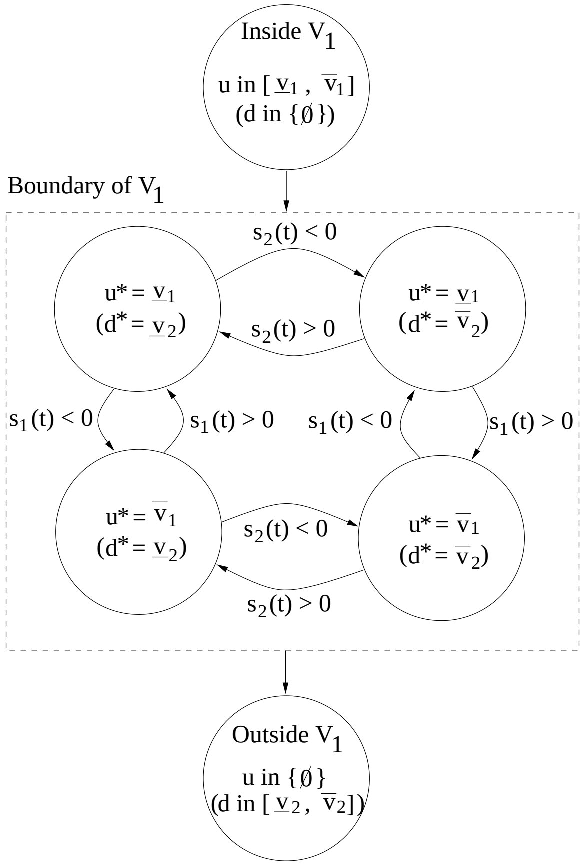

Fig. 6. Switching law governing the two aircraft system with linear velocity control inputs. The note in the caption of the automaton of the previous example, about the composition of automata for u and d, applies here also.

to be computationally hard, and with additional velocity bounds the multi-agent motion planning problem has been shown to be NP-complete [24]. In applications the scenarios considered most often involve navigation in the presence of other moving agents and obstacles [25]. The proposed solutions are geared towards distributed settings, in which only the local information about the state of the environment and the other agents in the vicinity is available to each agent. These techniques are based on potential and vortex field methods [26] and the complexity is proportional to the number of agents. An attempt to guarantee that the agents achieve their goals without colliding with each other has been proposed by Masoud [27]. In spite of the fact that collision avoidance is an integral part of agents' navigation capabilities, the requirements for safety and optimality have not been addressed to any great extent. This is partly due to the fact that the agent velocities have traditionally been relatively small and

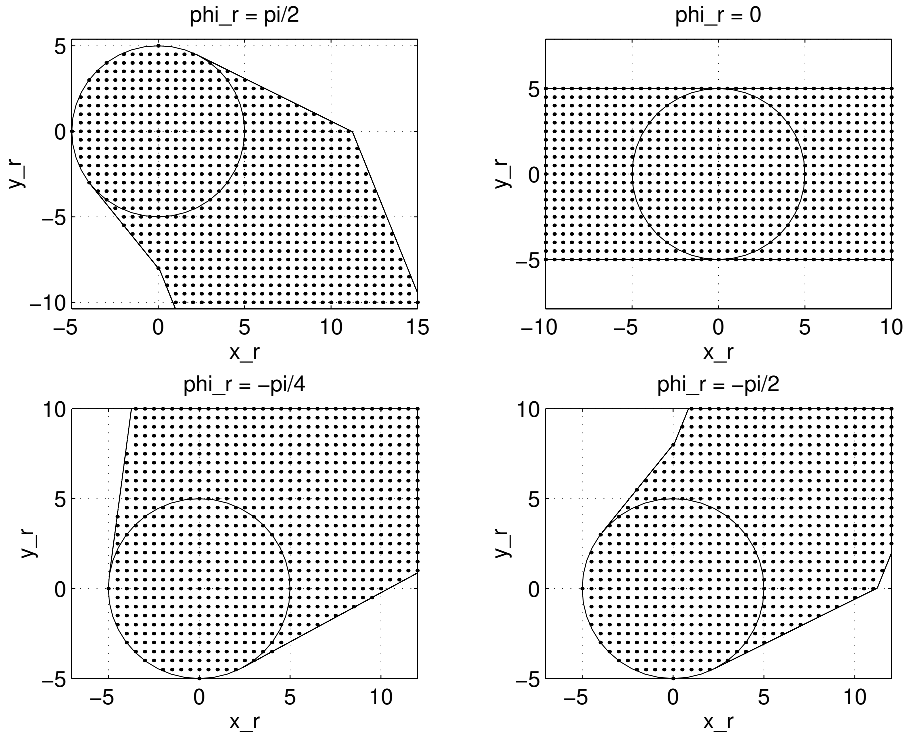

Fig. 7. Unsafe sets (x_r, y_r) for $[\underline{v}_1, \overline{v}_1] = [2, 4]$, $[\underline{v}_2, \overline{v}_2] = [1, 5]$ and $\phi_r = \pi/2,\ 0,\ -\pi/4$, and $-\pi/2$.

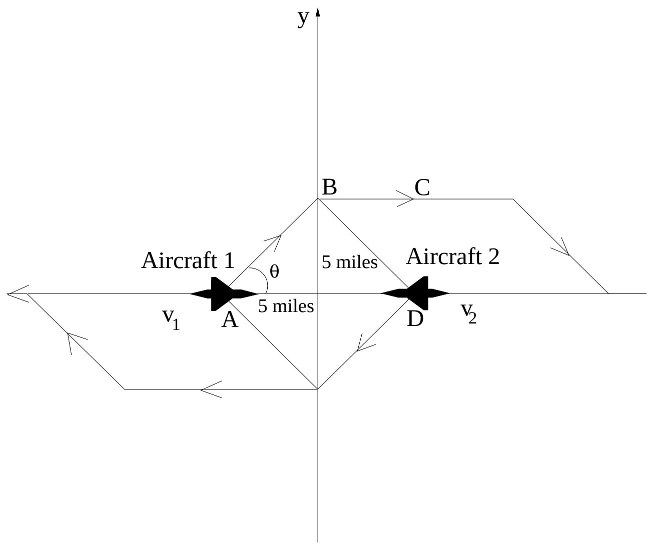

Fig. 8. Generalized *head-on* conflict.

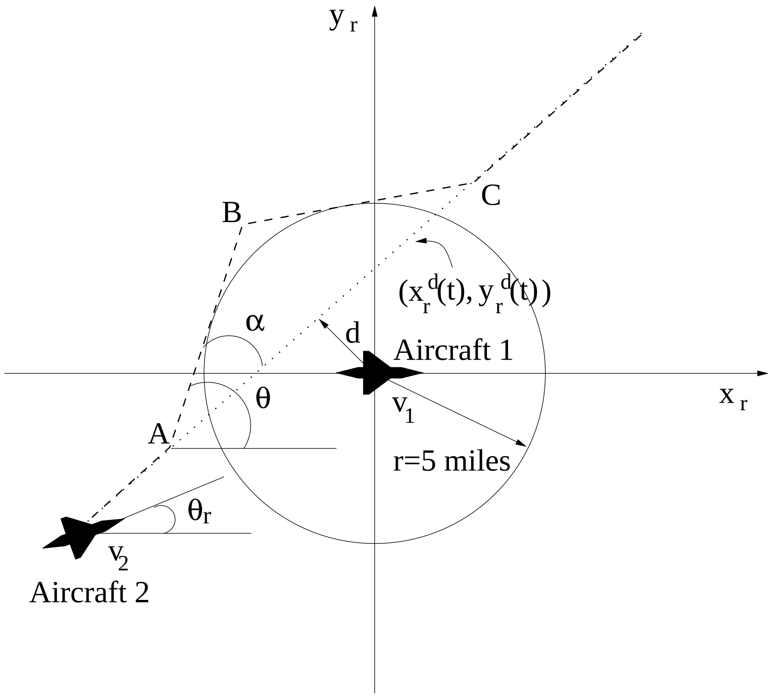

Fig. 9. Generalized *overtake* conflict.

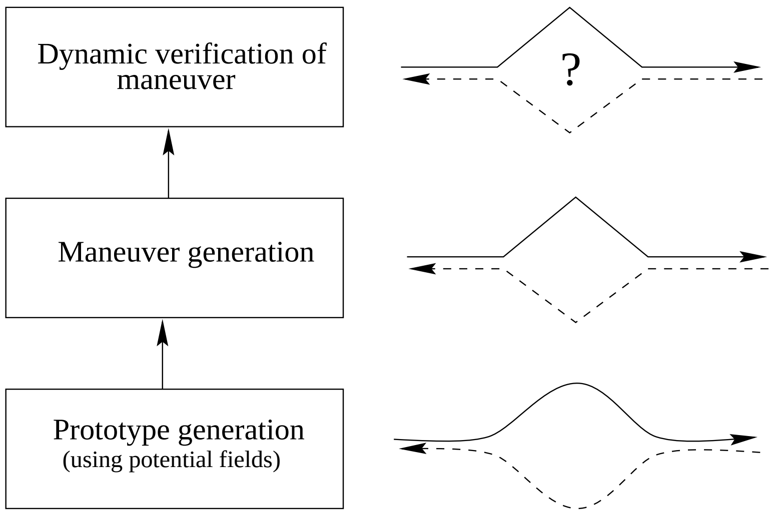

Fig. 10. Generation of Potential Field inspired maneuvers.

safety issues not so prominent: low velocity collisions occasionally occur, but various recovery strategies allow the agents to further pursue their tasks. In path planning for more than two agents, prioritized schemes have been used to fix the order in which the conflicts are resolved.

In our air traffic collision avoidance problem, we use the approach based on potential and vortex fields to generate *prototype* maneuvers from which we derive the *qualitative* properties of the actual maneuver that each aircraft follows, and then classify these prototypes into discrete sets of avoidance maneuvers which we prove to be safe.

Maneuver Generation: We adopt the potential and vortex field approach for distributed motion planning proposed in [27]. In ATM the absence of stationary obstacles and the approximation of individual agents by circles with a specified radius constitute reasonable assumptions prior to formulating the collision avoidance strategy[3]. We consider the *planar* collision avoidance problem with multiple moving agents.

The planner is obtained by the superposition of several vector fields representing qualitatively different steering actions of each agent. Suppose we have m agents, with the ith agent represented by a circle with radius r_i and its configuration denoted by $\boldsymbol{x}_i = (x_i, y_i)$. The desired destination of the ith agent $\boldsymbol{x}_{di} = (x_{di}, y_{di})$ is represented by an attractive potential function:

$$U_a(\boldsymbol{x}_i, \boldsymbol{x}_{di}) = \frac{1}{2}(\boldsymbol{x}_{di} - \boldsymbol{x}_i)^2$$

In order to achieve the desired destination a force proportional to the negative gradient of the U_a needs to be exerted:

$$F_a(\boldsymbol{x}_i, \boldsymbol{x}_{di}) = -\nabla U_a(\boldsymbol{x}_i, \boldsymbol{x}_{di}) = -(\boldsymbol{x}_i - \boldsymbol{x}_{di})$$

To prevent collisions between agents i and j, the following spherically symmetric repulsive field $U_r(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is associated with each agent:

$$U_r(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} -\frac{(\boldsymbol{r}_{ij} - (\boldsymbol{r}_j + \delta_{rj}))^2}{2\delta_{rj}} & \text{if } \boldsymbol{r}_j \leq \boldsymbol{r}_{ij} \leq \boldsymbol{r}_j + \delta_{rj} \\ 0 & \text{otherwise} \end{cases}$$

where $\boldsymbol{r}_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ is the distance between the ith and jth agent, $\boldsymbol{r}_j$ is the radius of jth agent and δ_{rj} is the influence zone of its repulsive field. The repulsive force associated with this field is:

$$F_r(\boldsymbol{x}_i, \boldsymbol{x}_j) = \nabla U_r(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

A vortex field, used to ensure that all agents turn in the same direction when encountering a conflict, is constructed around each agent tangential to the

[3] The aircraft is considered to be a "hockey puck" of a specified safety radius representing the desired clearance from the other aircraft.

repulsive field $U_r(\boldsymbol{x}_i, \boldsymbol{x}_j)$:

$$F_v(\boldsymbol{x}_i, \boldsymbol{x}_j) = \pm \begin{bmatrix} \frac{\partial U_r(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial y} \\ -\frac{\partial U_r(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial x} \end{bmatrix}$$

Note that by the choice of the sign in the above vortex field expression one can determine the direction of the circulating field. Setting the direction to a particular sign for all agents corresponds essentially to a "rule of the road" which specifies the direction of the avoidance for conflict maneuvers.

The dynamic planner for a single agent in the presence of multiple agents is obtained by superposition of participating potential and vector fields and becomes:

$$\dot{\boldsymbol{x}}_i = \frac{F_a(\boldsymbol{x}_i, \boldsymbol{x}_{di})}{\|F_a(\boldsymbol{x}_i, \boldsymbol{x}_{di})\|} + \sum_j (k_{ri}\, F_r(\boldsymbol{x}_i, \boldsymbol{x}_j) + k_{vi}\, F_v(\boldsymbol{x}_i, \boldsymbol{x}_j))$$

where $j = 1, ..., m, \ \ i \neq j$. The contributions from repulsive and vortex fields range between $[0, 1]$, increasing as the agent approaches the boundary of another agent. Normalization of the attractive field component makes its contribution comparable to the magnitudes of the repulsive and vortex fields. The strength of the field then becomes independent of the distance to the goal, capturing merely the heading to the goal. The individual contributions are then weighted by the by k_{ri}, k_{vi} and the resulting vector is again normalized and scaled by k_{di}, a constant proportional to the desired velocity of the ith agent. The velocity of ith agent is then:

$$\boldsymbol{v}_i = k_{di} \frac{\dot{\boldsymbol{x}}_i}{\|\dot{\boldsymbol{x}}_i\|}$$

In the following paragraph we demonstrate the capability of the planner to generate trajectories for general classes of collision avoidance maneuvers.

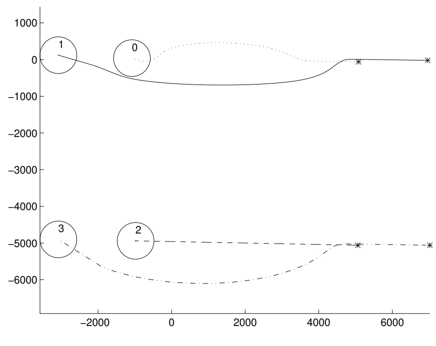

Fig. 11. *Overtake* maneuvers. Top: $1.5k_{d0} = k_{d1}$, $k_{r0} = k_{r1} = k_{v0} = k_{v1} = 1.0$. Bottom: $1.5k_{d2} = k_{d3}$, $k_{r2} = k_{v2} = 0.0$; $k_{r3} = k_{v3} = 1.0$

Overtake conflicts: The *overtake* conflict can be resolved by the planner in several qualitatively different ways obtained by adjusting the parameters in the individual contributions of the participating vector fields. In the outlined experiments two agents having different velocities participate in conflict resolution. In Figure 11 agent 1 is 1.5 times faster than agent 0. In the top maneuver, agent 1 overtakes agent 0 and agent 0 moves away from agent 1, resulting in smaller deviations from the original trajectory for both agents. The strength of contribution from the repulsive and vortex fields is the same for both agents. The willingness of the slower agent to cooperate in the overtake maneuver can be modeled by the strength of the agent's repulsive and vortex fields: in the bottom maneuver of Figure 11 the contributions of agent 2's vortex and repulsive fields are set to zero and agent 2 does not deviate from its original trajectory. Figure 12 demonstrates generalized overtake maneuvers

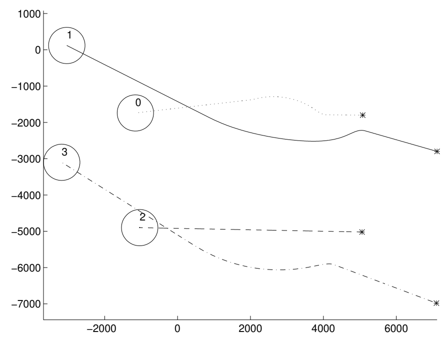

Fig. 12. Generalized *overtake* maneuver. In the conflict at the top both agents participate in the maneuver while at the bottom the conflict is resolved solely by agent 3.

Head-on conflicts: Similarly, several qualitatively different *head-on* maneuvers can be generated by changing the contributions of individual fields. In Figure 13 there are two head-on maneuvers where both agents actively cooperate in resolving the conflict, i.e. the strength of the repulsive and vortex fields is the same for each agent. Figure 14 demonstrates generalized head-on maneuvers where the agents are not initially aligned.

Multiple aircraft conflicts: When multiple aircraft are involved in conflict the vector field based planner is very instructional: the direction of the vortex field contribution serves as a coordination element between the aircraft. Figure 15 depicts a symmetric *roundabout* maneuver similar to the one proposed in [10]. The agents involved in the resolution of the conflict are homogeneous, having the same velocities, willing to participate equally in the maneuver (the strength of

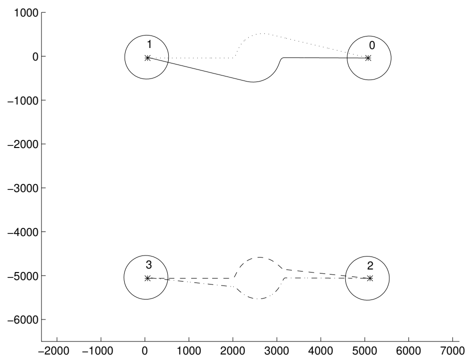

Fig. 13. *Head-on* maneuvers. Top: symmetric head-on with all the parameters the same. Bottom: $k_{d2} = k_{d3} = 10.0$, $k_{r2} = k_{r3} = 0.3$, $k_{v2} = k_{v3} = 5.0$ with the influence of the vortex field emphasized.

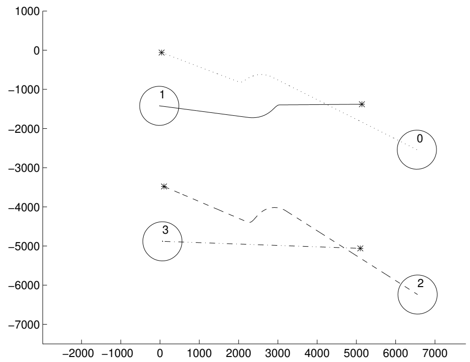

Fig. 14. Generalized *head-on* maneuver. Top: the velocities of the agents are the same and both agents participate in the maneuver. Bottom: agent 3 does not participate in the maneuver since $k_{r3} = k_{v3} = 0$.

the repulsive and vortex fields is the same for all agents). Figure 16 demonstrates a scenario where agent 0 does not participate in the coordination (k_{v0} and k_{r0} are 0) and is willing only to adjust its velocity slightly. This particular conflict can be still resolved and the resulting trajectories are flyable.

Observations: The presented planner has the capability of changing the spatial behavior of individual agents and always resolved the conflict if the agents were homogeneous and there were no restrictions on the temporal profiles of the agents' paths. Given particular constraints on agents' velocities certain conflicts may result in "loss of separation" or trajectories which are not flyable, due to the violation of the limits on turn angles (Figure 17). In such cases the shape of the path can be affected by changing parameters of contributing vector fields. The adjustment of influence zones δ_{ri} and δ_{vi} as well as the relative strength

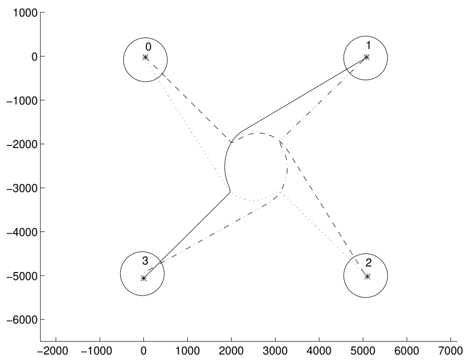

Fig. 15. Symmetric *roundabout*, gain factors for individual agents are the same.

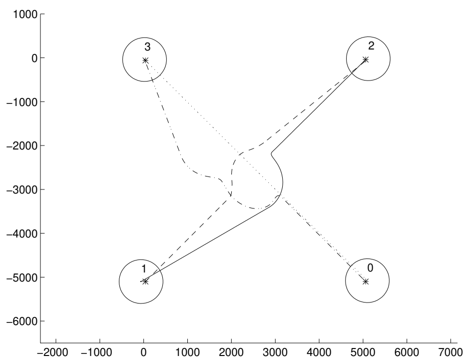

Fig. 16. Partial *roundabout*, $k_{vi} = k_{ri} = 1.0$ and $k_{d0} = 0.5k_{di}$ for i = 1,2,3 with the maximal velocity of agent 0 reduced by a factor of 2 and $k_{r0} = k_{v0} = 0$.

of the repulsive and vortex vector fields, k_{ri} and k_{vi}, can affect the turn angle and maximal deviation from the original trajectory needed to resolve the conflict. The change of the temporal profiles of the path by adjusting the velocities of individual agents (k_{di}) has the most profound affect on the capability of resolving general conflict scenarios. In Figure 18 the unflyable trajectory from Figure 17 can be changed by adjusting the velocity of agent 2 resulting in a flyable trajectory.

Maneuver Approximation and Verification: The discretization of the prototype maneuver is motivated by techniques currently performed by air traffic controllers which resolve conflicts by "vectoring" the aircraft in the airspace. This is partly due to the current status of the communication technology between the air traffic control center and the aircraft as well as the state of current avionics (autopilot) on board the aircraft which operate in a set-point mode. We consider two types of approximations: turning point and offset. The individual

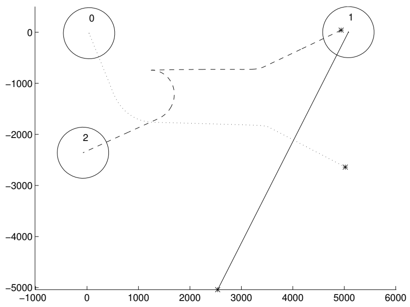

Fig. 17. General conflict scenario. Trajectory of agent 2 is not flyable.

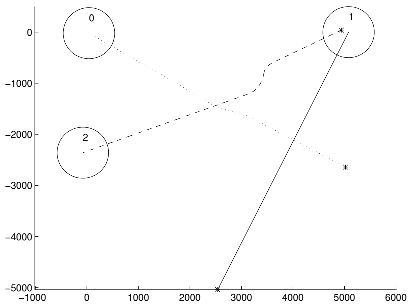

Fig. 18. Velocity profile agent 2 is adjusted resulting in a flyable trajectory.

approximation can be obtained from the trajectories generated by the dynamic planner by recursive least squares linear fit (see Figure 19).

Verification of the Maneuvers

The approximation phase is followed by the verification of the obtained maneuvers. The purpose of the verification step is to prove the safety of the maneuver by taking into account the velocity bounds and sets of initial conditions of individual aircraft. The collision avoidance problem lends itself to a hybrid system description: the continuous modes of the hybrid model correspond to individual parts of the maneuver (e.g. straight, turn right θ_1 degrees, turn left θ_2 degrees) and the transitions between modes correspond to switching between individual modes of the maneuver. Within each mode the speed of each aircraft can be specified in terms of lower and upper bounds. This suitable simplification of the problem allows us to model the collision avoidance maneuver in terms of hybrid automata. Each aircraft is modeled by a hybrid automaton, and an additional controller automaton implements the discrete avoidance maneuver strategy. The

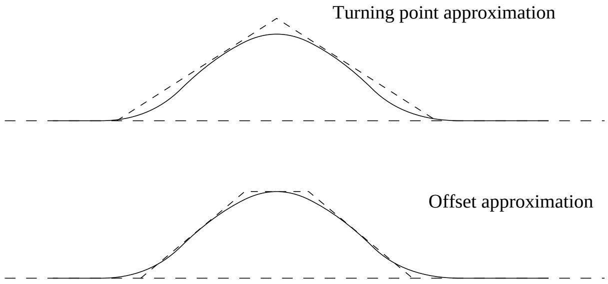

Fig. 19. Turning point and offset approximation.

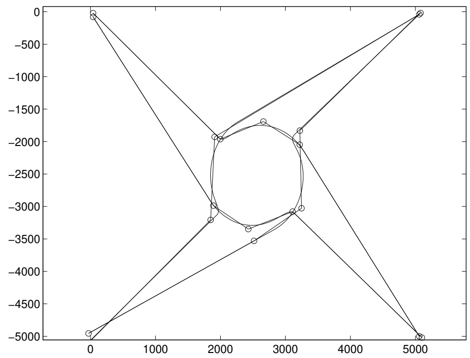

Fig. 20. Discretized *roundabout* maneuver.

verification results can assert that the maneuver is safe for given velocity bounds and given set of initial conditions. To relate the verification of the cooperative schemes to the use of the Hamilton-Jacobi equation of the previous section, we only mention that this approach can be used to compute the $\mathcal{P}re$ of an "unsafe" set in the iterations required to verify the regions of safety of a given maneuver. Further details may be found in [28].

The previously presented simulation results suggest that the generalized *overtake* and generalized *head-on* maneuvers may be used to solve all possible two-aircraft conflicts. This allows us to classify two-aircraft maneuvers by the angle at which the aircraft approach each other, and to design simple deviation maneuvers as sequences of straight line segments which approximate the trajectories derived from the potential and vortex field algorithm. For more than two aircraft the obtained discretized version of the *roundabout* maneuver is proposed. For this maneuver, the radius of a circular path around the conflict point is proportional to the influence zones of the aircrafts' repulsive and vortex fields. We propose this methodology as a suitable step of automation of conflict resolution in ATM given currently available technology. The complete classification of a library of reasonably complete conflict scenarios and maneuvers remains a challenging problem.

6 Conclusions

The technological advances that make free flight feasible include on-board GPS, satellite datalink, and powerful on-board computation such as the Traffic Collision and Avoidance System (TCAS), currently certified by the FAA to provide warnings of ground, traffic, and weather proximity. Navigation systems use GPS which provides each aircraft with its four dimensional coordinates with extreme precision. For conflict detection, current radar systems are adequate. Conflict prediction and resolution, however, require information regarding the position, velocity and intent of other aircraft in the vicinity. This will be accomplished by the proposed ADS-B broadcast information system. These advances will be economically feasible only for commercial aviation aircraft: how to merge the proposed architecture with general aviation aircraft (considered disturbances in the system in this paper) is a critical issue. Furthermore, the transition from the current to the proposed system must be smooth and gradual. Above all, the algorithms must be verified for correctness and safety before the implementation stage. This is one of the main challenges facing the systems and verification community. The accent in this paper has been on "safety" proofs for hybrid systems. In fact there are other properties of hybrid system such as *non-blockage of time*, *fairness*, etc. which are so-called liveness properties which also need to verified. Techniques for studying these are in their infancy except for very simple classes of hybrid system models.

Another important area of investigation in large scale systems design (such as the ATMS just described) is the *global or emergent* characteristics of the system. We have discussed how conflict resolution can provide autonomy for aircraft to decide how to plan their trajectories in the airspace between TRACONs, and for air traffic controllers to implement conflict resolution inside the TRACONs. The study of the composite automated system frequently reveals some surprising characteristics. For example, it was found from the implementation of CTAS at Dallas Fort Worth and UPR in the flight sector from Dallas to Washington that all aircraft tended to prefer the same route resulting in congestion at specific times in the Dallas TRACON. Another phenomenon associated with UPR is the formation of "convoys" of aircraft in the Asian airspace en-route from South East Asia to Europe. This latter phenomenon has spurred the study of the benefits of explicitly convoying aircraft in groups to their destination. Theoretical tools for the study of aggregate behavior arising from protocols for individual groups of agents are necessary to be able to assess the economic impact of air traffic automation strategies.

References

1. Honeywell Inc., "Markets Report," Tech. Rep. NASA Contract NAS2-114279, 1996.
2. Radio Technical Commission for Aeronautics, "Final report of RTCA task force 3: Free flight implementation," tech. rep., Washington DC, October 1995.
3. G. J. Couluris and S. Dorsky, "Advanced air transportation technologies (AATT)

potential benefits analysis," Tech. Rep. NASA Contract NAS2-13767, Seagull Technology Inc., Cupertino, CA, September 1995.
4. Honeywell Inc., "Technology and Procedures Report," Tech. Rep. NASA Contract NAS2-114279, 1996.
5. W. H. Harman, "TCAS : A system for preventing midair collisions," *The Lincoln Laboratory Journal*, vol. 2, no. 3, pp. 437–457, 1989.
6. H. Erzberger, "CTAS : Computer intelligence for air traffic control in the terminal area," Tech. Rep. NASA TM-103959, NASA Ames Research Center, Moffett Field, CA, July 1992.
7. J. Lygeros, C. Tomlin, and S. Sastry, "Mulit-objective hybrid controller synthesis," in *Proceedings of HART97* (O. Maler, ed.), no. 1201 in LNCS, pp. 109–123, Berlin: Springer Verlag, 1997.
8. S. Kahne and I. Frolow, "Air traffic management: Evolution with technology," *IEEE Control Systems Magazine*, vol. 16, no. 4, pp. 12–21, 1996.
9. D. J. Brudnicki and A. L. McFarland, "User request evaluation tool (URET) conflict probe performance and benefits assessment," in *Proceedings of the U.S.A./Europe ATM Seminar*, (Eurocontrol, Paris), 1997.
10. C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A case study in multi-agent hybrid systems," tech. rep., UCB/ERL M97/33, Electronics Research Laboratory, University of California, Berkeley, 1997. Accepted to appear in the IEEE Transactions on Automatic Control.
11. 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*, (Grenoble), pp. 109–123, 1997. Longer version available as UCB/ERL Memo M97/59, submitted to Automatica.
12. C. Tomlin, J. Lygeros, L. Benvenuti, and S. Sastry, "Output tracking for a non-minimum phase dynamic CTOL aircraft model," in *Conference on Decision and Control*, 1995.
13. C. Tomlin and S. Sastry, "Bounded tracking for nonminimum phase nonlinear systems with fast zero dynamics," in *Conference on Decision and Control*, (Kobe, Japan), pp. 2058–2063, 1996. Longer version available as UCB/ERL Memo M96/46, accepted to appear in the International Journal of Control.
14. G. J. Pappas, J. Lygeros, and D. N. Godbole, "Stabilization and tracking of feedback linearizable systems under input constraints," in *Conference on Decision and Control*, pp. 596–601, 1995.
15. G. J. Pappas and S. Sastry, "Towards continuous abstractions of dynamical and control systems," in *Hybrid Systems IV* (P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, eds.), Lecture Notes in Computer Science, New York: Springer Verlag, 1997.
16. 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.
17. Y. Zhao and R. 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), Aug. 1997.
18. M. Shewchun and E. Feron, "Linear matrix inequalities for analysis of free flight conflict problems," in *Conference on Decision and Control*, (San Diego, CA), 1997.
19. J. K. Kuchar, *A Unified Methodology for the Evaluation of Hazard Alerting Systems*. PhD thesis, Massacussets Institute of Technology, 1995.

20. R. A. Paielli and H. Erzberger, "Conflict probability estimation and resolution for free flight." NASA Ames Research Center, Preprint, 1996.
21. R. Isaacs, *Differential Games.* John Wiley, 1967.
22. T. Başar and G. J. Olsder, *Dynamic Non-cooperative Game Theory.* Academic Press, second ed., 1995.
23. M. Erdman and T. Lozano-Perez, "On multiple moving objects," *Algorithmica*, vol. 2, pp. 477–595, 1987.
24. J. Canny and J. Reif, "New lower bound techniques for robot motion planning problems," in *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science*, pp. 49–60, 1987.
25. M. J. Mataric, "Issues and approaches in the design of collective autonomous agents," *Robotics and Autonomous Systems*, vol. 16, pp. 321–331, December 1995.
26. C. D. Medio and G. Oriolo, "Robot obstacle avoidance using vortex fields," in *Advances in Robot Kinematics* (S. Stifter and J.Lenarcic, eds.), pp. 227 – 235, 1991.
27. A. Masoud, "Using hybrid vector-harmonic potential fields for multi-robot, multi-target navigation in stationary environment," in *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 3564–3571, 1996.
28. J. Košecká, C. Tomlin, G. Pappas, and S. Sastry, "Verification of cooperative conflict resolution maneuvers," in *Submitted to Hybrid Systems V*, 1997.