A NEXT GENERATION ARCHITECTURE FOR AIR TRAFFIC MANAGEMENT SYSTEMS

by

C. Tomlin, G. Pappas, J. Lygeros, D. Godbole, and S. Sastry

Memorandum No. UCB/ERL M97/7

21 January 1997

A NEXT GENERATION ARCHITECTURE FOR AIR TRAFFIC MANAGEMENT SYSTEMS

by

C. Tomlin, G. Pappas, J. Lygeros, D. Godbole, and S. Sastry

Memorandum No. UCB/ERL M97/7

21 January 1997

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A Next Generation Architecture for Air Traffic Management Systems*

C. Tomlin, G. Pappas, J. Lygeros, D. Godbole, and S. Sastry
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley, CA 94720

Abstract

The study of hierarchical, hybrid control systems in the framework of air traffic management systems (ATMS) is presented. The need for a new ATMS arises from the overcrowding of large urban airports and the need to more efficiently handle larger numbers of aircraft, without building new runways. Recent technological advances, such as the availability of relatively inexpensive and fast real time computers both on board the aircraft and in the control tower, make a more advanced air traffic control system a reality. The usefulness of these technological advances is limited by today's Air Traffic control (ATC), a ground-based system which routes aircraft along predefined jet ways in the sky, allowing the aircraft very little autonomy in choosing their own routes. In this paper, we propose an architecture for an automated ATMS, in which much of the current ATC functionality is moved on board each aircraft so that the aircraft may calculate their own deviations from predefined trajectories without consulting ATC. Within the framework of this architecture, we describe our work in on-board conflict resolution strategies between aircraft, and in deriving the flight mode switching logic in the flight vehicle management systems of each aircraft.

1 Introduction

For decades, commercial air travel has played an indispensable role in our economy and society. The increasing demand for air travel has so far been met by building larger and more modern airports. Little has been done however to improve the efficiency of air traffic management. Most of the effort in this area has been centered on simplifying the job of the air traffic controllers by providing them with advisory systems, better displays, etc. The use of automatic control has mostly been restricted to on-board autopilots with relatively small

*Research supported by NASA under grant NAG 2-1039 and AATT grant NAS 2-14291 (as a subcontract through Honeywell Technology Center), and by ARO under grants DAAH 04-95-1-0588 and DAAH 04-96-1-0341.

degrees of autonomy. The research presented here aims at improving air travel conditions by introducing automation to air traffic management.

The primary objective in our work is to improve the efficiency of air travel. Many of the current air traffic control (ATC) practices are dictated by the absolute desire to maintain safety and the consequent need to keep the task of the human controllers simple. For example, aircraft are currently routed along prespecified paths to avoid having to deal with the complications of "free flight". In addition, because of heavy workload, air traffic controllers are primarily concerned with maintaining safe spacing between aircraft, ignoring considerations such as fuel consumption, travel times, etc. We believe that the introduction of automation can lead to great savings in terms of travel times, unplanned delays, and fuel consumption, and can possibly increase the number of aircraft handled. An additional benefit will be an increase in the safety of the flights (reduced number of aborted landings, near collisions, etc.). The improvement is likely to be more dramatic in the case of degraded conditions of operation, such as aircraft malfunctions, ATC malfunctions (e.g. power failure), shifting winds (that cause changes in approach patterns), bad weather, switching from manual to instrumented landings, etc. It should be noted that conditions like these occur regularly in practice and can cause severe degradation in the system performance. These topics are discussed in greater detail in Section 2.

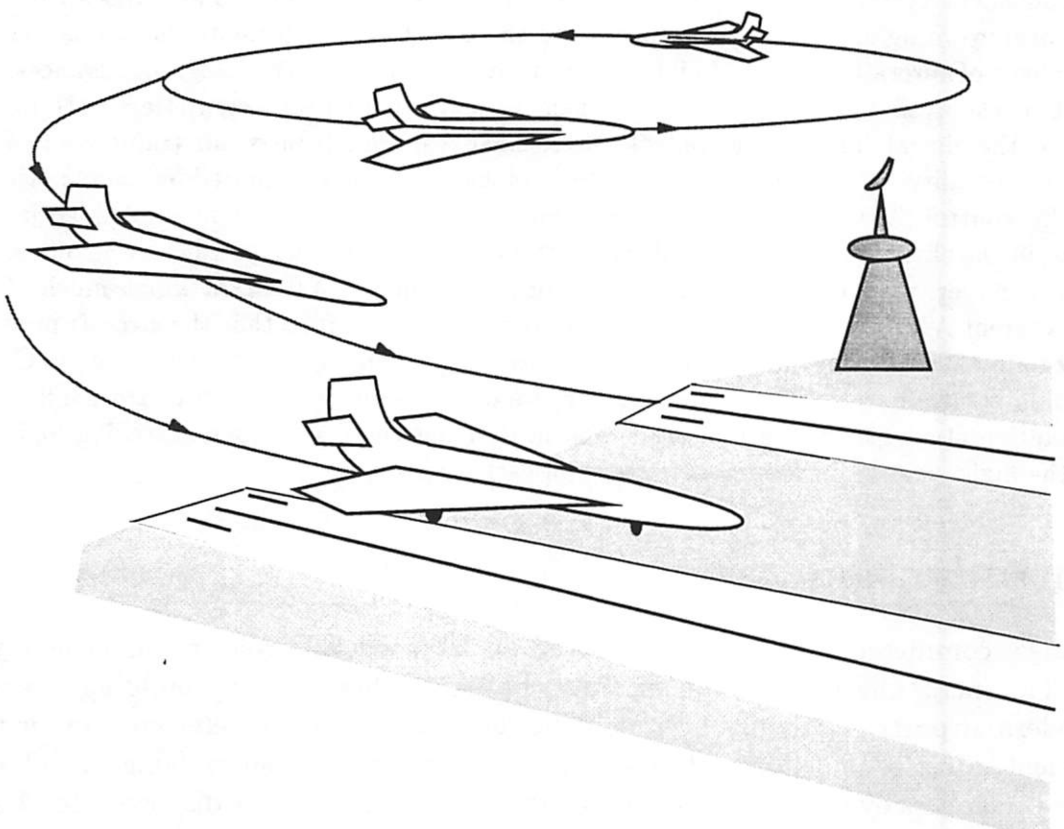

Figure 1: Current Airport Landing Patterns

The air traffic management system (ATMS) we envision will be automated[1] and will involve the harmonious union between on-board air traffic control and flight vehicle management systems. This system uses advances in Communication, Navigation and Surveillance (CNS) both on board aircraft and on the ground, along with advances in avionics on board aircraft. The proposed new architecture for ATMS is inspired by our research on the control of *hierarchical hybrid systems.* Because air traffic management requires coordination and control of a large number of semi-autonomous agents (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. In our architecture, the abstract higher levels will be modeled by discrete event systems and the lower levels by detailed continuous aircraft models and arithmetic control laws.

One of the most important conceptual issues to be addressed in the architecture of these control systems is their degree of decentralization. For example, current air traffic control practice is completely centralized with the regional centers, airport control towers and gate controllers providing all of the instructions, while current roadway driving practice is completely decentralized with individual drivers (usually adopting "greedy strategies") setting their driving control laws. There are clear drawbacks to each: the completely decentralized solution is *inefficient and leads to conflict*, while the completely centralized one is *not tolerant of faults in the central controller, computationally and conceptually complicated and slow to respond to emergencies.* 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 to maintain their safe operation, and try to optimize their own performance measures. They also coordinate with neighboring agents and a centralized controller to resolve conflicts as they arise and maintain efficient operation.*

For reasons of economic and reliable information transfer among the agents and the centralized controller, coordination among the agents is usually in the form of communication protocols which are modeled by discrete event systems. Since the dynamics of individual agents is modeled by differential equations, we are left with a combination of interacting discrete event dynamical systems and differential equations resulting in *hybrid control systems.* An important issue in the area of hybrid systems is the analysis and design of protocols and interfaces between agents as well as continuous control laws for each agent.

In this paper we present an overview of our research effort in the area of ATMS. To motivate the problem, we first give a brief overview of current ATC practice, in Section 2. In Section 3 we present the proposed hierarchical control architecture that we believe can alleviate some of the problems experienced by the current system. A discussion on centralization versus decentralization issues is first given in Section 3.1 followed by an overview of the functionality of each of the levels of the architecture in Section 3.2. In Sections 4 and 5 we present results on two of the research directions pursued within this framework: in Section 4 we present the algorithms proposed for conflict resolution, while in Section 5 we

[1] Parts of our work can also be used to produce advisories for ATC and pilots in a semi-automated ATMS.

discuss some of the hybrid control issues that emerge in our work. We present an example on safety in the operation of individual aircraft and use it to motivate issues in mode switching and hybrid controller design. Due to space limitations only brief discussions are given for certain areas of our research while certain others are only mentioned. We provide references where more details can be found throughout the text.

2 Current ATC Practice

Air Traffic Control (ATC) in the United States is currently organized hierarchically with a single Air Traffic Control System Command Center (ATCSCC) supervising the overall traffic flow management (TFM). This is supported by 20 *Air Traffic Control System Command Centers (ARTCCs)* organized by geographical area. Coastal ARTCCs have jurisdiction over oceanic waters. For example, the Fremont (California) ARTCC has jurisdiction from roughly Eureka in Northern California to Santa Barbara in Central California and from midway to the Hawaiian islands 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, 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 NAS (National Airspace System).

The main goal of both the ARTCCs and the TRACONs is to maintain safe separation between aircraft while guiding the aircraft to their destinations. Due to their heavy workloads, minimizing flight delays and fuel spent en route are not prime considerations of controllers when they determine trajectories for the aircraft to follow, even though the airline flight dispatch offices and the cockpits do negotiate with the ATC to achieve these objectives. Inefficiencies cause unplanned delays in average flight times, and thus there are deviations from pre-negotiated airline schedules forcing air traffic controllers and flight dispatch offices to manually schedule and reschedule aircraft landings according to when the aircraft enters the TRACON region. In addition, there is minimal communication between the ARTCCs and TRACON ATCs which makes forecasting delays almost impossible. Studies conducted by ATC researchers at NASA Ames have illustrated that, when presented with tables of flight data (position, air velocity, ground velocity, wind speed, etc.) of two aircraft in the TRACON region, a human controller does not have the ability to quickly predict the future motion of the two aircraft. Controllers therefore guide the aircraft along predetermined jet ways both in the TRACON and in the en route airspace. In the TRACON, this results in some aircraft left in holding patterns circling the airport while others are performing their final approach for landing.

Figure 2 depicts the horizontal projection of a typical route inside the TRACON. Because aircraft must land into the wind (with as low a cross-wind as possible) to maintain lift at low ground speed, the runway configuration in large airports is such that, frequently, only one set of two parallel runways is used at any given time. The aircraft are sequenced manually

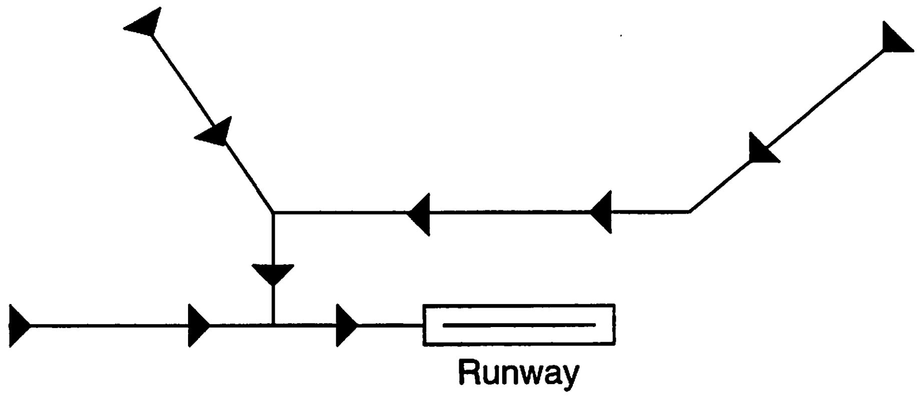

Figure 2: Typical route pattern for arriving aircraft

as they enter the TRACON, and they maintain this sequence along the illustrated route. Where the routes converge, ATC decides which aircraft is allowed to go first and what the ensuing sequence will be. If an aircraft enters the TRACON in an emergency state and must land as quickly as possible, ATC manually reroutes and reschedules the other TRACON aircraft so that priority can be given to the troubled aircraft.

In the regions outside airport TRACONs, the ARTCCs perform the routing and scheduling tasks for each aircraft. These tasks are considerably less intensive and the workload is much lighter than for TRACON controllers. The ARTCC also uses predefined air routes or jet ways (flight maps describing these routes are published each year) and one of their main tasks is to predict and avoid conflicts. If ATC predicts that the separation between two aircraft will become less than the regulatory separation, it either slows down one of the aircraft or puts it into a delay loop. Other current ATC practices are listed below.

- ATC uses only discrete levels of altitude when routing aircraft between TRACONs (for example, Westbound aircraft fly at even thousand feet altitude while Eastbound fly at odd thousand feet, similarly odd five hundreds are used by Northbound aircraft and even five hundreds for Southbound aircraft);

- If the optimal route of an aircraft takes it to an altitude of less than 11,000 feet above an en route TRACON, ATC directs the aircraft *around* the intermediate airport so that the TRACON-ATC's workload is not increased;

- Shifting winds and inclement weather at airports cause problems in scheduling, since the airport must be reconfigured to use different runways, and as a result, aircraft are delayed, often at their originating airports;

- Due to the fixed routes between TRACONs, delays at destination airports are communicated back to origin airports, and aircraft at origins up to 4 hours away from the destinations may be delayed.

ATMS efficiency is a complex quantity to define, but includes the following features:

Airport and Airspace Capacity. Airport capacity is defined as the maximum number of aircraft takeoffs and landings that can be supported by the airfield under given climatic conditions when there is a continuous demand for service. Airport capacity is a function of the runway-taxiway configurations, aircraft mix, weather conditions, and landing aids. Airspace capacity is the maximum number of operations that can be processed per unit time in a certain volume of the airspace given a continuous demand. In this definition a distinction is made between different modes of operation, such as *level flight at fixed heading*, *climbing*, *descending*, and *changes in heading.* Airspace capacity is a function of aircraft count, activity mix, and protocols for collision resolution and detection, as well as Federal Aviation Authority (FAA) regulations. It is our contention in this paper that it is this latter capacity i.e., **airspace capacity** that can be increased by better protocols which do not compromise safety.

Delays caused by ATC. Ground holds that are imposed by the FAA on departing aircraft in anticipation of congestion due to forecast bad weather at the destination are examples of delays caused by ATC. This practice may be inefficient since the inclement weather may fail to materialize (resulting in starvation of arrivals at the destination airport) or because it may be acceptable to have a few aircraft in holding patterns while a TRACON is reconfigured to account for changes in weather conditions.

Operating Costs. Operating costs are incurred because of procedures which could be more flexible. For example, frequently the so-called "user preferred routes" (shorter routes, low fuel consumption routes using tailwinds) are disallowed because of the requirement to use prescribed jet ways or the need to go from point to point along jagged paths over ground based "fixes". Airlines claim that very large savings can be effected (for the U.S. estimates mentioned range from 1 to 3 billion annually) by using advances in avionics and automated ATC capacity both on board the aircraft and on the ground to detect and resolve conflicts. This procedure is referred to as *free flight.*

In order to improve efficiency, researchers at NASA Ames are developing a system which automates some parts of ATC. The system is called the Center-TRACON Automation System (CTAS), and is described in detail in [1], [2], and [3]. CTAS is a program which generates *advisories*, or suggested trajectories, runway assignments, landing sequences, and schedules, which the controller may use in managing air traffic. Its key components are a dynamic planning algorithm and a trajectory synthesis algorithm, which use mathematical models of the aircraft, representations of traffic patterns and approach routes and models of the atmosphere to generate these advisories. CTAS also contains a graphical user interface to provide the controller with displays of estimated and scheduled times of arrival and descent advisories, and a conflict checking and resolution program. The functionality of CTAS is purely advisory: the controller still communicates verbally to the pilot of each aircraft, and may decide to use or ignore the information that CTAS provides. Field tests of CTAS are now underway at the Denver and Dallas/Fort Worth airports [4].

A summary of the efficiency issues of the current ATMS and a description of ATMS technologies that will become available in the near future is presented in [5].

3 A Distributed Decentralized ATMS

3.1 Motivation

The tradeoff between centralized and decentralized decision making raises a fundamental issue that has to be addressed by any proposed ATMS. The above discussion indicates that the current ATC system is primarily centralized; all safety critical decisions are taken centrally (at the ATC centers) and distributed to the local agents (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, as we have seen, to inefficient operation. Recent technological advances, such as Global Positioning Systems (GPS), better communication and navigation equipment and more powerful on board computers make it possible to distribute part of the decision making responsibility to the local agents. It is hoped that this will lead to improved system performance.

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 (refer to the discussion in Section 2) it seems that, in principle, the higher the level of centralization the closer one can get to the globally optimal solution[2]. However, the complexity of the problem also increases in the process; in a sense to implement a centralized design one has to solve a small number of more complex problems as opposed to large number of simpler ones. As a consequence the implementation of a centralized solution requires a greater effort on the part of the designer in order to produce control algorithms and greater computational power in order to execute these algorithms. 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[3]. 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 ATMS 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 transient or local phenomena. Improvements in performance may

[2]Any decentralized solution can also be implemented centrally.

[3]Indeed, in August 1995, the central computer in the FAA control center at Fremont, California, experienced a 65 minute power failure, leaving close to 70 aircraft with no communication to ATC. Catastrophic collisions were narrowly avoided by communication between the pilots, a natural process of decentralized decision making.

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 above discussion indicates that determining an appropriate mix of centralized and decentralized decision making is a delicate process. It seems, however, that given the current demand and technological limitations the system could benefit by distributing more decision making responsibility to the aircraft. In the next section we propose a control architecture that implements what we believe is a reasonable balance between centralization and decentralization.

3.2 Proposed ATMS Architecture

We propose an architecture for a fully automated air traffic management system. In this system each aircraft is equipped with a hierarchical planning and control algorithm, and an algorithm to resolve potential collision conflicts with other aircraft. Each aircraft follows a nominal path from source airport to destination airport. This nominal path is calculated off-line in consultation with ATC and is designed to be time-optimal and conflict-free. However, once the aircraft are airborne and outside the TRACON, bad weather, high winds, or schedule delays which cause conflicts with other aircraft may force the aircraft to deviate from this nominal route. In the current system, these deviations are calculated by the central ATC and each aircraft must obtain a *clearance* from ATC before altering its course. In our proposed ATMS, the aircraft may plan its own deviation trajectories without consulting ATC. This semi-autonomy is enabled by on-board conflict resolution algorithms, which allow the aircraft to coordinate among each other. Inside the airport TRACONs, the aircraft trajectories would continue to be strictly regulated by ATC.

A block diagram of the ATMS proposed architecture is presented in Figure 3. The levels of architecture below ATC reside on the individual aircraft and comprise what is known as the aircraft's *Flight Vehicle Management System*, or FVMS. The FVMS consists of four layers, the strategic, tactical, and trajectory planners, and the regulation layer. Each layer of this architecture is described in the following sections. We begin with a discussion of the airspace structure.

Airspace Structure

Nominal trajectories through the airspace are defined in terms of *waypoints*, which are fixed points in the airspace defined by VOR (Visual Omni Range) points on the ground. Aircraft flying in the range of the waypoint's radio transmission (shown as an inverse cone in Figure 4) obtain *fixes* as to their position and orientation relative to the waypoint. The waypoints are a necessary navigation tool for aircraft which are not equipped with the more sophisticated GPS. Figure 4 also illustrates the approach routes into the San Francisco airport in terms of these waypoints.

We assume for our architecture that the waypoint structure of the airspace is intact, so that trajectories are defined at the coarsest level in terms of sequences of these waypoints.

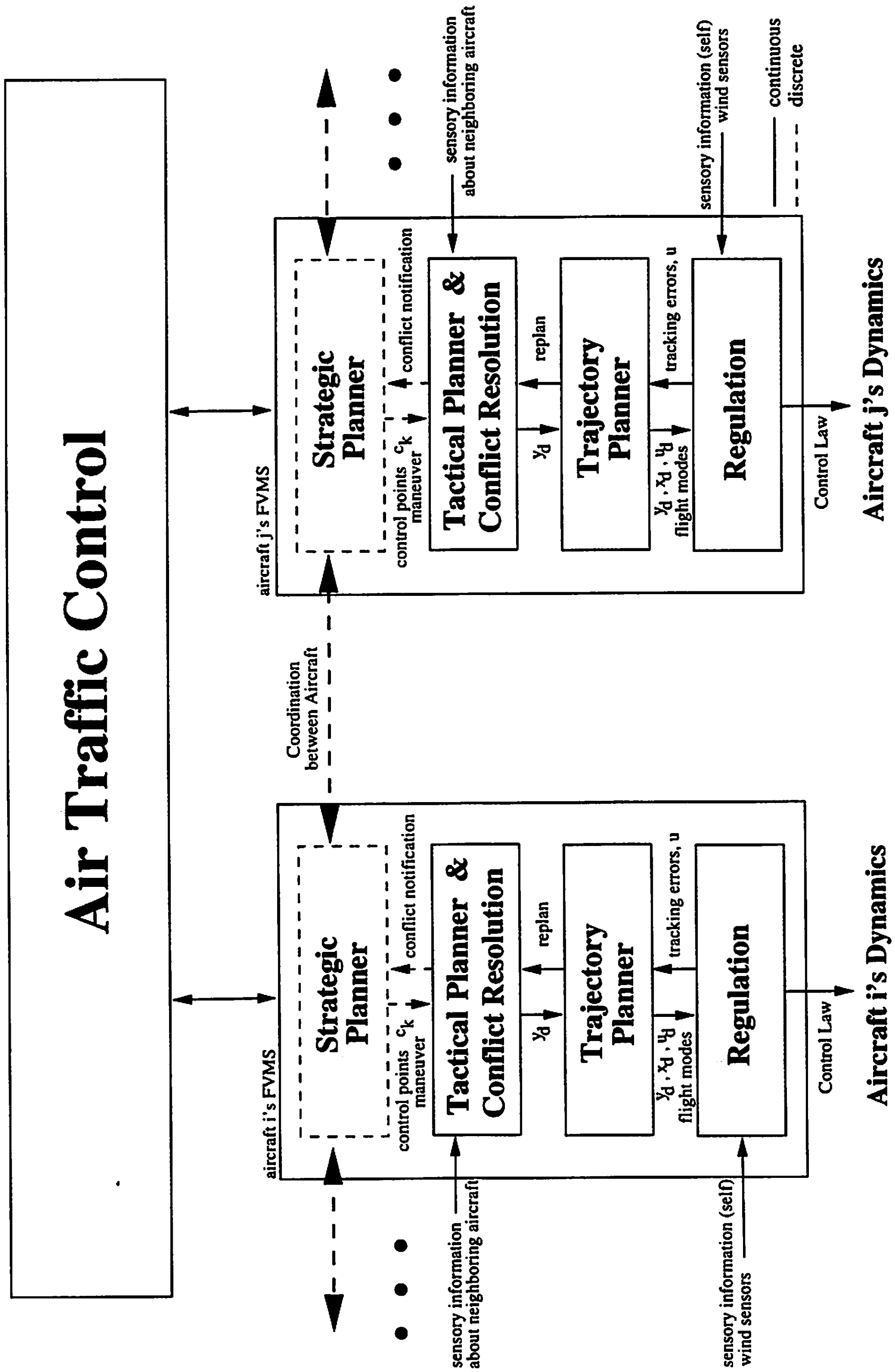

Figure 3: Proposed ATMS Architecture

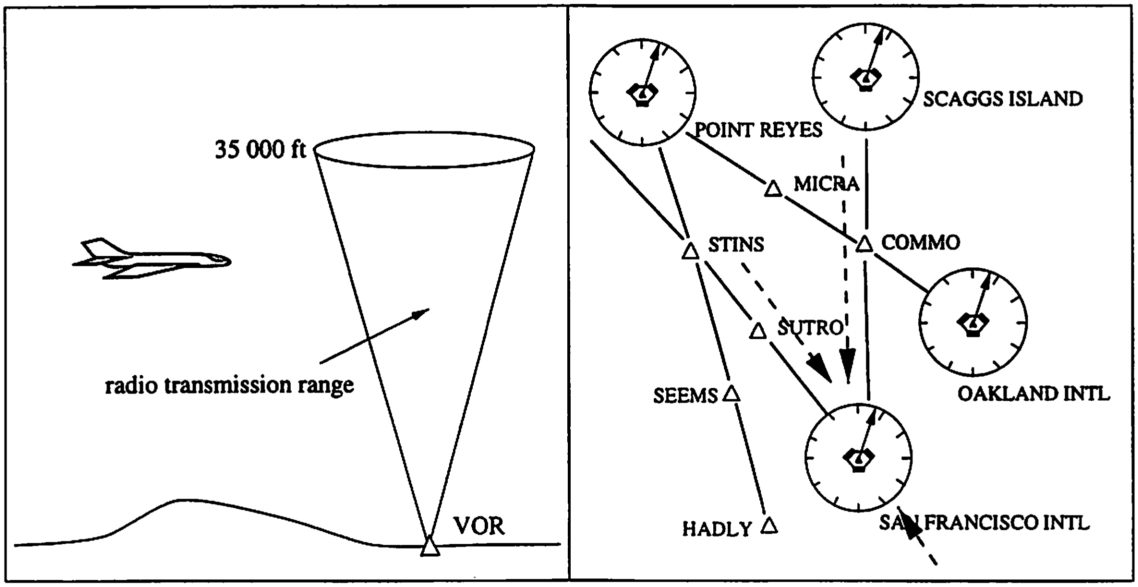

Figure 4: Airspace Structure

These are the trajectories that are communicated between each aircraft and ATC: the FVMS of each aircraft refines the waypoints into full state and input trajectories.

Air Traffic Control

ATC has more control over aircraft in the TRACON than over aircraft outside the TRACON airspace. In both regions, ATC passes a sequence of waypoints to the strategic planner on board the aircraft, defining a nominal trajectory. These waypoints are a discretization of a kinematic trajectory, accessed from a database of stored kinematic trajectories, which have been calculated offline for different combinations of aircraft kinematics, wind magnitude and direction, and runway configurations. These pre-computed trajectories have been optimized to provide a minimum-time path for the given aircraft kinematics. The waypoints from ATC are time-stamped to provide a suggested arrival schedule at the destination airport, which is designed to meet the announced arrival times and reflects conflict resolution and compromises between airline schedules. Once these waypoints have been negotiated they are passed to the strategic planner, and all of the planning and control tasks are taken over by the FVMS on board the individual aircraft.

Outside the TRACON region, the FVMS is allowed to alter its nominal trajectory by changing the waypoints and coordinating with the FVMSs of other aircraft. For these deviations, the tactical planner takes over the role of calculating an initial kinematic trajectory for the aircraft. The role of the ATC is limited to keeping track of these changes and providing the aircraft with global information about enroute traffic and weather conditions.

Strategic Planner

The main objectives of the strategic planner are to design a coarse trajectory for the aircraft in the form of a sequence of control points, c_k, which interpolate the waypoints from ATC, and to resolve conflicts between aircraft.

If the tactical planner on board the aircraft predicts that a conflict will occur between its aircraft and another aircraft, it notifies the strategic planner. 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. Each strategic planner then commands its own tactical planner to follow these maneuvers.

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 3. The tactical planner is also responsible for predicting conflicts.

The tactical planner uses a simple kinematic model of the aircraft for all trajectory calculations. For conflict prediction, it uses information about the positions and velocities of neighboring aircraft (available through radar) and kinematic models to predict their movement. If more information, such as neighboring aircraft type and capabilities, is available through communication, the models can be refined. 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 assumptions made in extrapolating aircraft trajectories plays a crucial role in conflict prediction. If we assume no a-priori knowledge of the other aircrafts' intentions we can assume that they will maintain the same velocity over the horizon of prediction. A more conservative approach is to assume that the other aircraft will do their worst to cause conflict. Predicting the trajectories under this assumption involves solving an optimal control problem in which the cost function encodes the spacing between the aircraft in question and its neighbors (that the neighbors seek to minimize). Clearly this approach will predict more conflicts than the constant velocity extrapolation. If the conflict cannot be resolved using this optimal control theoretic approach, the aircraft communicate with each other at the strategic level to resolve the conflict. In this case, the maneuvers and resulting commands are accessed from a database of precomputed solutions to possible conflicts. A detailed discussion of conflict resolution is presented in the next section, and in [6].

When the tactical planner predicts that a conflict will occur, it sends a discrete signal to the strategic planner. After conflict resolution, a new tactical plan needs to be established and new conflicts predicted. Verification is needed to guarantee that this process eventually leads to an acceptable, conflict-free trajectory. Because of the relative simplicity of the kinematic models we hope to be able to carry out this verification using finite state and timed automata techniques.

Trajectory Planner

The trajectory planner uses a detailed dynamic model of the aircraft, sensory input about the wind's magnitude and direction, and the tactical plan consisting of an output trajectory, to design a full state and input trajectory for the aircraft, and the sequence of *flight modes* necessary to execute the dynamic plan. These flight modes represent different modes of operation of the aircraft and they correspond to controlling different variables in the aircraft dynamics. An analysis of deriving the flight mode logic necessary for safe operation of a CTOL (Conventional Take Off and Landing) aircraft is presented in Section 5.

The resulting trajectory, denoted y_d, x_d, and u_d in Figure 3, 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 [7] and actuator saturation [8].

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. Clearly verification is needed to show that the scheme eventually converges to an acceptable trajectory. Due to the increased complexity of the models it is unlikely that timed automata techniques will be adequate in this setting. More elaborate (possibly hybrid) techniques may be necessary.

4 Conflict Resolution

In this section, we describe an algorithm for resolving possible collision conflicts between aircraft. This algorithm is presented in greater depth in [6]. 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 [9], [10] and probabilistic [11] models. Any conflict resolution scheme should work not only when the aircraft have the ability to communicate with each other, but also when this communication breaks down, when the distances between the aircraft are too large, for example, or because one or more of the aircraft involved in the conflict is a general aviation aircraft not equipped with the sensing and communication technology of the larger commercial aircraft. We therefore differentiate between two types of conflict resolution: *noncooperative* and *cooperative* (Figure 5). The algorithms described in this section fit into the ATMS architecture as shown in the detail in Figure 6.

4.1 Noncooperative Conflict Resolution

If an aircraft detects that a conflict may occur between itself and another aircraft, and it is not able to communicate with this aircraft to determine its intentions or to resolve the conflict, then the safest action that this aircraft can take is to choose a strategy which resolves the conflict for the *worst possible action of the other aircraft.* We therefore formulate the

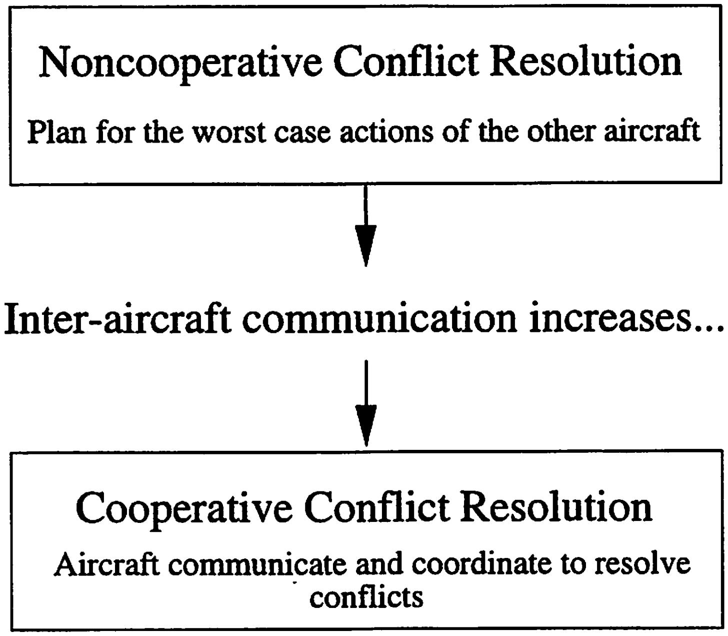

Figure 5: Noncooperative and cooperative Conflict Resolution

noncooperative conflict resolution strategy as a zero sum dynamical game of the pursuit-evasion style [12], [13]. The aircraft are treated as players in this game. Each player is aware only of the 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 uncontrolled. Each aircraft solves the game for the worst possible disturbance. The performance index over which the aircraft compete is the relative distance between the aircraft, required to be above a certain threshold (the Federal Aviation Administration requires a 5 mile horizontal separation). 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 therefore unnecessary, since these actions are a priori safe. If the saddle solution to the game is unsafe, it may be because the disturbance sets are too large. Partial or full coordination between the agents is then necessary in order to reduce the disturbance sets.

For kinematic aircraft models in two dimensions, it is straightforward to work out the noncooperative conflict resolution strategy. Consider two aircraft with kinematic models in the Lie group $SE(2)$

$$\begin{aligned} \dot{g}_1 &= g_1 X_1 \\ \dot{g}_2 &= g_2 X_2 \end{aligned} \tag{1}$$

where $g_1, g_2 \in SE(2)$ and $X_1, X_2 \in se(2)$, the Lie algebra associated with $SE(2)$. The relative configuration of aircraft 2 with respect to aircraft 1 is denoted $g_r = g_1^{-1} g_2$. The

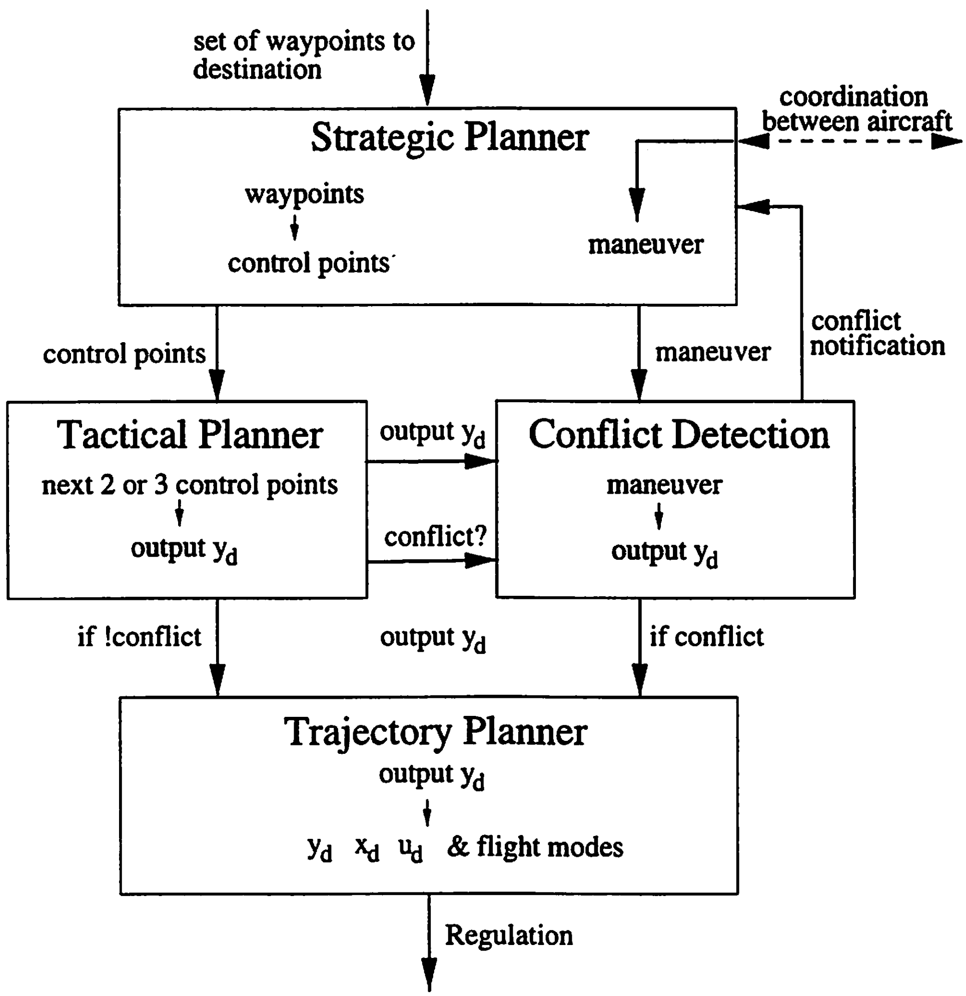

Figure 6: ATMS Architecture, showing Conflict Resolution

resulting model is

$$\begin{aligned} \dot{x}_r &= -v_1 + v_2 \cos\theta_r + \omega_1 y_r \\ \dot{y}_r &= v_2 \sin\theta_r - \omega_1 x_r \\ \dot{\theta}_r &= \omega_2 - \omega_1 \end{aligned} \tag{2}$$

where $X = (x_r, y_r, \theta_r)$ represents the relative position and orientation, and ω_i, v_i represent the angular and linear velocities of each aircraft. We consider this system in the framework of a pursuit-evasion game, in which aircraft 1, at the origin of the relative axis frame, is the *evader*, and aircraft 2 is the *pursuer*. The control inputs are the actions of the evader, and the disturbances are the actions of the pursuer:

$$\begin{aligned} u &= [v_1,\ \omega_1]^T \in \mathbb{R}^2 \\ d &= [v_2,\ \omega_2]^T \in \mathbb{R}^2 \end{aligned}$$

The cost function in the game is the relative distance between the two aircraft:

$$J_s(X_0, u, d) = \inf_{t \geq 0} \sqrt{x_r(t)^2 + y_r(t)^2} \tag{3}$$

with a threshold of 5 miles.

Consider the case in which the aircraft do not deviate from their original paths, but only change their linear velocities to resolve the conflict. In this case, ω_1 and ω_2 are set to zero, and equations (2) may be solved analytically. The control and disturbance variables are restricted to lie in intervals of the positive real line:

$$\begin{aligned} u &\in [\underline{v_1},\ \overline{v_1}] \in \mathbb{R}^+ \\ d &\in [\underline{v_2},\ \overline{v_2}] \in \mathbb{R}^+ \end{aligned}$$

The saddle solution for the game, which describes the *best* control strategy for the *worst* disturbance, is summarized in Figure 7. The saddle solution may be described in words as: if the pursuer is in front of evader, the evader should fly as slowly as possible, otherwise, the evader should fly as quickly as possible; if the pursuer is heading towards the evader, the pursuer should fly as quickly as possible, otherwise, the pursuer should fly as slowly as possible. Having calculated the saddle solution, we can calculate the unsafe sets of initial states for the pursuer. These are illustrated in Figure 8 for various relative orientations of the two aircraft. The arrows indicate the relative orientations of the evader (at the center of the protected zone) and the pursuer.

4.2 Cooperative Conflict Resolution

In cooperative conflict resolution, safety is ensured by full coordination among the aircraft. The aircraft follow predefined 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.

Protocol for Two Aircraft

A general conflict scenario is depicted in Figure 9. Aircraft 2 with speed v_2 and initial heading θ_r has desired *relative* trajectory $(x_r^d(t), y_r^d(t))$, which is the straight line path joining point

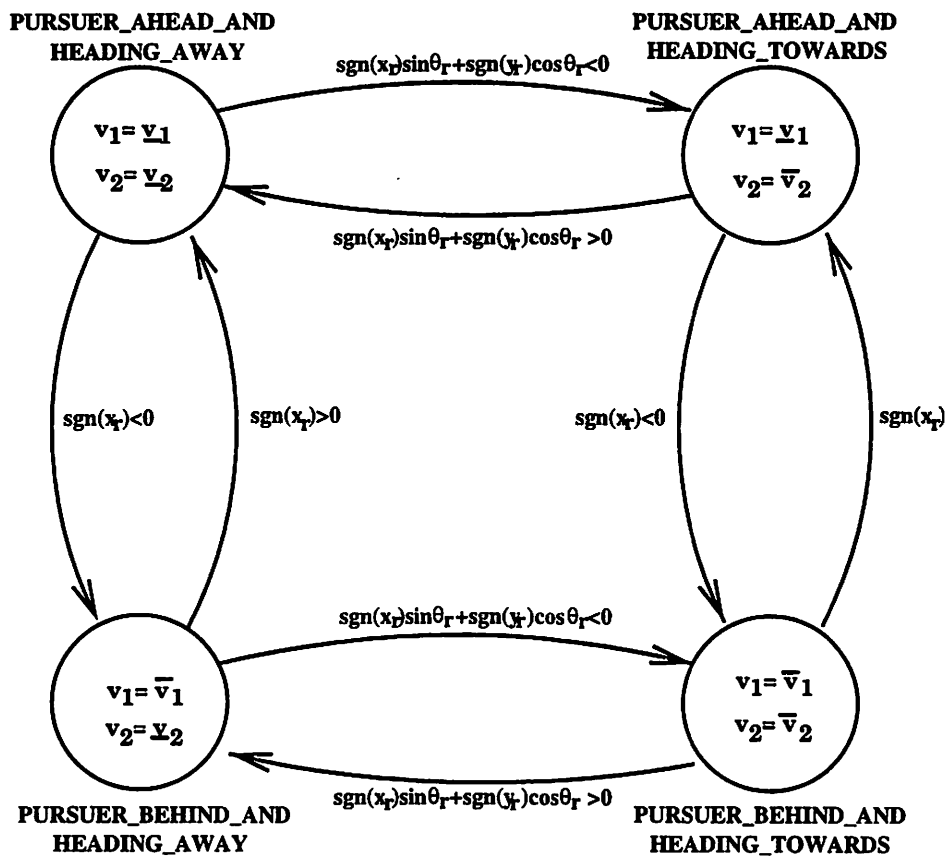

Figure 7: Abstraction of Saddle Solution as a Hybrid Automaton

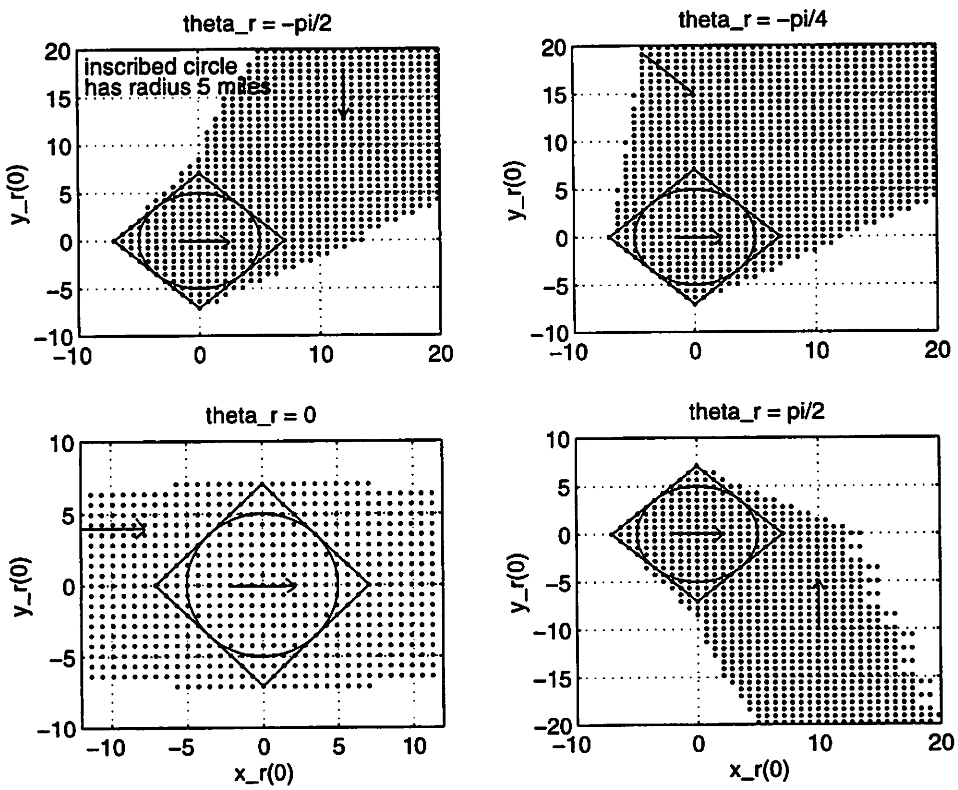

Figure 8: Unsafe sets $(x_r(0), y_r(0))$ for $[\underline{v}_1, \overline{v}_1] = [2,4]$, $[\underline{v}_2, \overline{v}_2] = [1,5]$ and $\theta_r = -\pi/2$, $-\pi/4$, 0, and $\pi/2$.

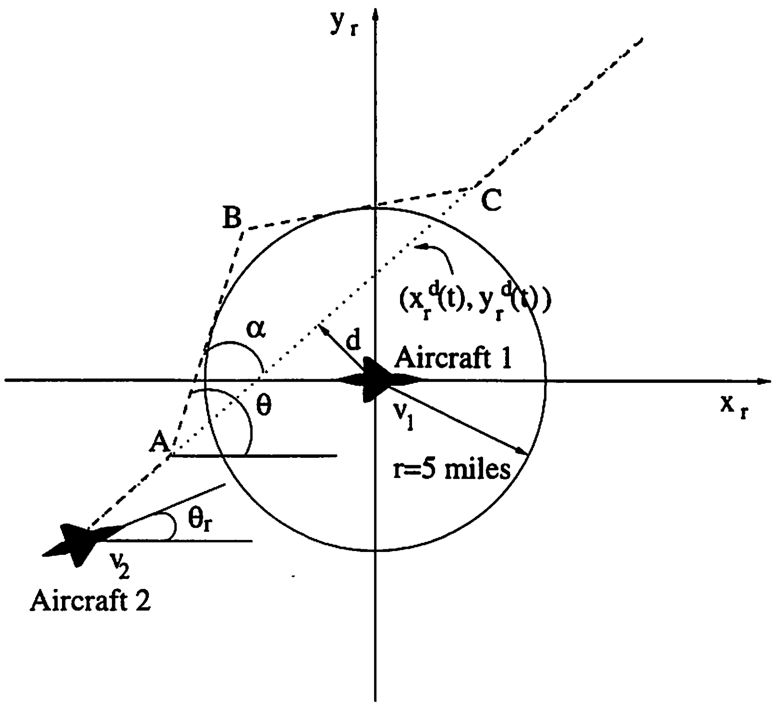

Figure 9: Showing the triangular path deviation (dashed line), at optimal angle θ, to be used in pairwise conflict avoidance

A and point C a distance d away from the origin (seen as the dotted line in Figure 9). To simplify the analysis, the protected zone of aircraft 2 is translated to aircraft 1, to make the protected zone around aircraft 1 twice its original radius. If aircraft 2 were to continue along its original desired path, it would cut through this protected zone, and come into conflict with aircraft 1. To avoid the protected zone, the proposed deviation for aircraft 2 is the triangular path ABC tangent to the protected zone at two places and parameterized by the deviation angle θ (represented by the dashed line in Figure 9).

Aircraft 2 follows the specified path ABC if the component of its relative velocity normal to this path is zero. Since straight line paths are considered, the relative velocity of aircraft 2 is described by the model (2). The angle θ is calculated to minimize the time it takes for aircraft 1 to travel along the path ABC. Its optimal value is obtained by minimizing with respect to θ the length of ABC divided by the speed of the aircraft along this path. As the ratio v_2/v_1 gets large, the optimal value for θ approaches 45° [6].

This *Overtake* maneuver is a special case of the general class of triangular conflict resolution maneuvers. In each aircraft's FVMS, a routine exists which computes θ for the different parameters r, d, θ_r, and v_2/v_1:

$$\theta = Overtake(r, d, \theta_r, v_2/v_1) \tag{4}$$

It is assumed in this architecture that the aircraft with the greater speed must perform the maneuver; the other aircraft remains on its original course.

Consider now a *HeadOn* conflict, in which aircraft 1 is heading towards aircraft 2 ($\theta_r = 180°$) along the x_r axis ($d = 0$). A potential conflict exists regardless of the speeds of aircraft

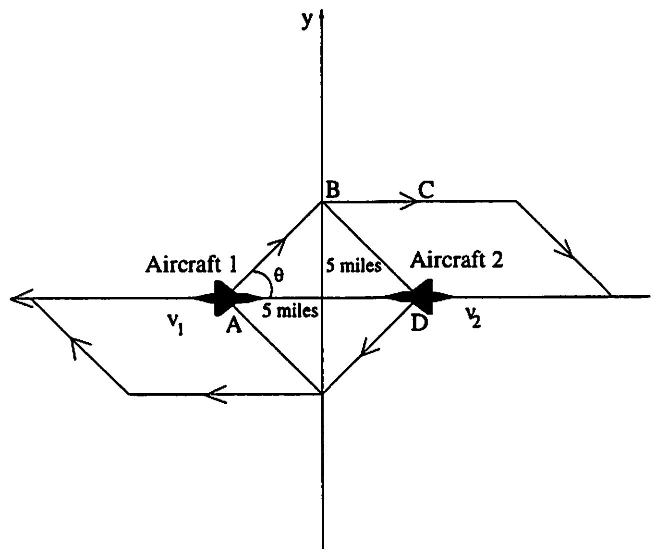

Figure 10: Showing the HeadOn conflict and subsequent conflict resolution maneuver

2 and aircraft 1. Although the conflict may be resolved using the general maneuver discussed above, the issue of *fairness* arises. If $v_1 \approx v_2$, it is not clear how to choose which aircraft deviates from its original trajectory. A natural solution is to define a maneuver in which both aircraft deviate from their original trajectories:

$$(\theta_1, \theta_2) = HeadOn(r, d, \theta_r, v_2/v_1) \tag{5}$$

Inspired by the Overtake maneuver, θ_1 and θ_2 are set to 45° and −45°, respectively, when $d = 0$ and $\theta_r = 180°$. The Overtake maneuver is *safe by design*, since the construction of the deviation path explicitly avoids the protected zone of one of the aircraft. In order to ensure that the HeadOn conflict is safe by design, both aircraft must deviate a horizontal distance of 5 miles (the minimum aircraft separation) away from their original paths. Figure 10 illustrates why, in the absolute frame of the two aircraft. As with the Overtake maneuver, the HeadOn maneuver in its general form may be used for relative headings θ_r other than 180°.

Protocol for Three Aircraft

For three aircraft coming into potential conflict, there are many more possibilities for types of conflict. For example, two aircraft could have intersecting trajectories, and then conflict resolution between these two could result in a new conflict with a third aircraft. Pairwise conflict resolution may not work in cases such as these: it is worthwhile to design a maneuver which works for three aircraft, with the possibility to extend it to more than three aircraft. A maneuver which is inspired by the potential field algorithms of the robotics literature [14] is the *Roundabout* maneuver, illustrated in Figure 11 for the case of three aircraft with two initial points of conflict. For this maneuver, a circular path is defined around the conflict points of all three trajectories as shown. The aircraft are restricted to fly along the circular path segments with a given speed, as not to overtake the other aircraft already involved in

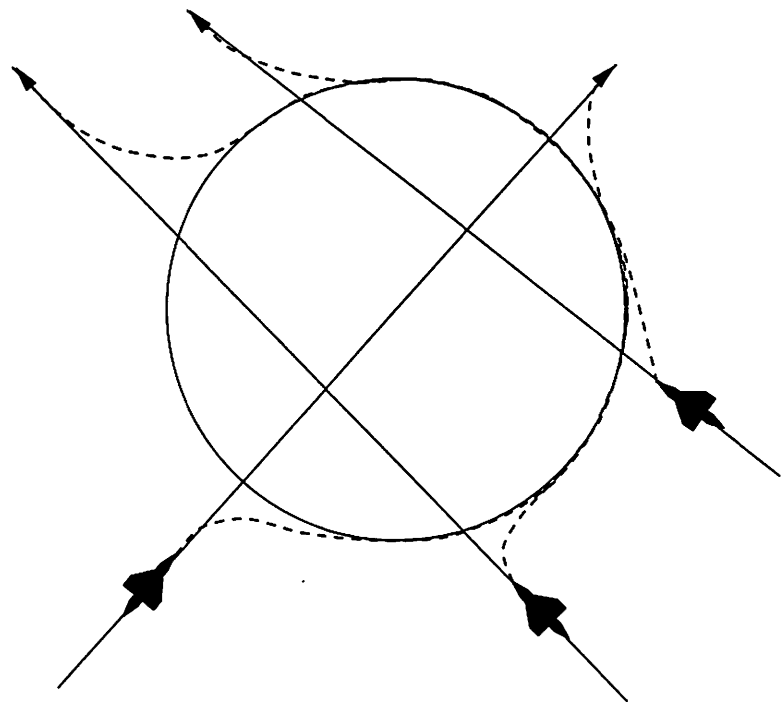

Figure 11: Conflict Resolution for three aircraft: the Roundabout maneuver

the maneuver. An aircraft may not enter the Roundabout until the other aircraft are outside its protected zone; in extreme cases this may force an aircraft to enter a holding pattern to delay its entry.

5 Hybrid Control in FVMS

The operation of the proposed ATMS involves the interaction of continuous and discrete dynamics. Such *hybrid* phenomena arise, for example, from the coordination between aircraft at the strategic level. 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.

Discrete phenomena also arise in the operation of a single aircraft. In the trajectory and regulation levels discrete changes are observed 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. The modes may also be dictated by input constraints: saturated inputs can no longer be used effectively, certain controls (e.g. the flaps) may not be used in certain situations (e.g. high airspeeds), etc.

To illustrate some of these issues we present in this section a simplified example of hybrid dynamics that arise on a single FVMS. This example was originally presented as part of a research program to develop models of hybrid systems [15], [16]. In the example, the goal of the FVMS is to keep the state of the aircraft in a given subset of the state space dictated in principle by stall constraints. The task is complicated by input saturation which also dictates the flight mode switching.

5.1 Problem Description

Our example is based on the Conventional Take Off and Landing (CTOL) dynamic aircraft equations and the design specification of [17]. The equations model the speed and the flight path angle dynamics of a commercial aircraft in still air. The inputs to the equations are the thrust T, accessed through the engine throttle, and the pitch angle θ, accessed through the elevators, and the outputs are the speed V and the flight path angle γ. There are three primary modes of operation. In **Mode 1**, the thrust T is between its specified operating limits ($T_{min} < T < T_{max}$), the inputs are T and θ, and both V and γ are controlled outputs. In **Mode 2**, the thrust saturates ($T = T_{min} \vee T_{max}$) and thus is no longer available as an input; the only input is θ, and the only controlled output is V. Finally, in **Mode 3**, the thrust saturates ($T = T_{min} \vee T_{max}$); the input is again θ, and the controlled output is γ. Within Modes 2 and 3 there are two submodes depending on whether $T = T_{min}$ (idle thrust) or $T = T_{max}$ (maximum thrust).

Safety regulations for the aircraft dictate that V and γ must remain within specified limits: for ease of presentation we simplify this *safety envelope*, S, of [17] to

$$S = \{(V,\gamma) | (V_{min} \leq V \leq V_{max}) \wedge (\gamma_{min} \leq \gamma \leq \gamma_{max})\}$$

where $V_{min}, V_{max}, \gamma_{min}, \gamma_{max}$ are constants. We would like to design a control scheme which will cause the aircraft to reach a target operating point $(V,\gamma)_{target}$ in S from any initial operating point in S. The resulting trajectory $(V(t),\gamma(t))$ must satisfy acceleration constraints imposed for passenger comfort, and must not exit the envelope at any time. Here we describe the minimally restrictive set of controllers which guarantees safe operation of the aircraft, by classifying all of the controls that keep the $(V(t),\gamma(t))$ trajectory within the safety envelope and establishing the mode switching logic required for safety. The secondary requirement for passenger comfort is then optimized within the class of safe controls.

The flight path angle dynamics of the aircraft can be summarized using two state variables, $x = [V \ \ \gamma]^T \in \mathbb{R} \times S^1$, where V (m/s) is the airspeed and γ (rad) is the flight path angle. The dynamics of the system are given by:

$$\dot{V} = \frac{T-D}{m} - g\sin\gamma \quad (6)$$

$$\dot{\gamma} = \frac{L}{mV} - g\cos\gamma \quad (7)$$

where T (N) is the thrust, m (kg) is the mass of the aircraft, g (m/s^2) is gravitational acceleration and L and D are the aerodynamic lift and drag forces. The aerodynamic forces can be modeled by:

$$L = a_L V^2(1 + c(\theta - \gamma)) \quad (8)$$
$$D = a_D V^2(1 + b(1 + c(\theta - \gamma))^2) \quad (9)$$

where a_L and a_D are the lift and drag coefficients, b and c are small positive constants, and θ is the aircraft pitch angle. Substituting the lift and drag equations into the dynamic equations, and assuming that b is small enough to neglect the quadratic term in the drag, the system dynamics are:

$$\dot{V} = -\frac{a_D V^2}{m} - g\sin\gamma + (\frac{1}{m})T \quad (10)$$
$$\dot{\gamma} = \frac{a_L V(1 - c\gamma)}{m} - \frac{g\cos\gamma}{V} + (\frac{a_L V c}{m})\theta \quad (11)$$

For these equations to be meaningful we need to assume that $V > 0$ and $-\pi/2 \leq \gamma \leq \pi/2$. Clearly this will be the case for realistic aircraft. Moreover, physical considerations also impose constraints on the inputs: $u = [T \;\; \theta]^T \in U = [T_{min}, T_{max}] \times [\theta_{min}, \theta_{max}]$.

To guarantee safety we need to ensure that $x(t) \in S$ for all t. Let ∂S denote the boundary of S. The requirement that the state stays within S can be encoded by a cost function:

$$J_1(x^0, u) = -\min_{t\geq 0}(x(t) - \partial S) \quad (12)$$

by defining:

$$x(t) - \partial S = \begin{cases} \min_{y\in\partial S} \|x(t) - y\| & \text{if } x \in S \\ -\min_{y\in\partial S} \|x(t) - y\| & \text{if } x \notin S \end{cases}$$

Here $\|\cdot\|$ denotes the Euclidean metric on $\mathbb{R}^2$. For the given set S the expression for J_1 becomes:

$$J_1(x^0, u) = -\min\left\{\min_{t\geq 0}(V(t) - V_{min}), \min_{t\geq 0}(V_{max} - V(t)), \min_{t\geq 0}(\gamma(t) - \gamma_{min}), \min_{t\geq 0}(\gamma_{max} - \gamma(t))\right\}$$

To ensure that the state stays within S we impose the threshold $J_1(x^0, u) \leq 0$.

Cost functions involving the linear and angular accelerations can be used to encode the requirement for passenger comfort:

$$J_2(x^0, u) = \max_{t\geq 0}(\dot{V}(t)) \quad \text{and} \quad J_2'(x^0, u) = \max_{t\geq 0}(V(t)\dot{\gamma}(t)) \quad (13)$$

The requirement that the linear and angular acceleration remain within the limits determined for comfortable travel are encoded by the thresholds $J_2(x^0, u) \leq 0.1g$ and $J_2'(x^0, u) \leq 0.1g$.

In all of the calculations we use the aircraft parameters and state and input limits for a $DC-8$ at cruising speed, at an altitude of 35000 ft.

5.2 The Least Restrictive Class of Safe Controls

To find the controls that keep the state within the safety envelope we solve the following optimal control problem:

$$J_1^*(x^0) = \min_{u\in\mathcal{U}} J_1(x^0,u) \quad \text{and} \quad u^*(x^0) = \arg\min_{u\in\mathcal{U}} J_1(x^0,u) \tag{14}$$

Proposition 1 (Optimally Safe Controls) *The optimally safe control input is*

$$u^*(x) = \begin{cases} (T_{max},\theta_{min}) & \forall x=(V,\gamma)\in S\cap\{(V,\gamma): \frac{\gamma-\gamma_{min}}{\gamma_{max}-\gamma_{min}} > \frac{V-V_{min}}{V_{max}-V_{min}}\} \\ (T_{min},\theta_{max}) & \forall x=(V,\gamma)\in S\cap\{(V,\gamma): \frac{\gamma-\gamma_{min}}{\gamma_{max}-\gamma_{min}} < \frac{V-V_{min}}{V_{max}-V_{min}}\} \end{cases} \tag{15}$$

The optimal control calculation allows us to determine the set of safe states and the class of controls that renders this set safe. Note that, if $J_1^*(x^0) > 0$ there is no control that will keep the trajectory starting at $x^0 \in S$ within S. If, however, $J_1^*(x^0) \le 0$ there exists at least one (and maybe multiple) such safe controls. Our goal therefore is to determine:

$$V_1 = \{x^0 \in S | J_1^*(x^0) \le 0\} \quad \text{and} \quad \mathcal{U}_1(x^0) = \{u \in \mathcal{U} | J_1(x^0,u) \le 0\}$$

We start by analyzing the system equations (10, 11) along ∂S. Consider an arbitrary point $x^0 \in \partial S$. We can distinguish three cases. If $f(x^0,u)$ points "inside" S for all $u \in U$ then all controls are safe for the given point x^0, i.e. $\mathcal{U}_1(x^0) = U$. If $f(x^0,u)$ points "outside" S for some u, let $\hat{U} \subset U$ be the controls for which this happens. These inputs are unsafe for the point x^0, i.e. $\mathcal{U}_1(x^0) = U \setminus \hat{U}$. Finally, if $f(x^0,u)$ points outside S for all $u \in U$ then all controls are unsafe for the given point x^0, i.e. $\mathcal{U}_1(x^0) = \emptyset$.

A special case of the second situation is one where $f(x^0,u)$ is tangent to ∂S for some u and points outside for all others. In this case, the set of controls that make $f(x^0,u)$ tangential to ∂S will be exactly u^*. This allows us to extend the safe set construction to the interior of S. The system equations are integrated backwards for the unique safe input from that point to determine the boundary of the safe set of states on the interior of the envelope.

Consider the left hand edge of ∂S: the complete set of controls moves from being safe to unsafe as γ varies from γ_{min} to γ_{max}. We can determine which values of (T,θ) in U are unsafe along ∂S by determining where the vector field along this boundary is tangent to ∂S. We calculate this by setting $\dot{V} = 0$, $T = \hat{T}$ in equation (10) and solving for $\hat{T}$ as a function of γ:

$$\hat{T}(\gamma) = a_D V_{min}^2 + mg\sin\gamma$$

For each γ, $\hat{T}(\gamma)$ is the value of the input thrust for which the vector field is tangent to ∂S. $\hat{T}(\gamma)$ does not depend on θ, so the safe set of inputs along ∂S may be parameterized solely by T, and is those T for which $T(\gamma) \ge \hat{T}(\gamma)$. When γ is such that $\hat{T}(\gamma) = T_{min}$, the cone of vector fields points completely "inside" S; when γ is such that $\hat{T}(\gamma) = T_{max}$, the cone of vector fields points completely "outside" S, and T_{max} is the unique thrust input which keeps the system trajectory inside S. We define γ_1 and γ_2 to be such that $\hat{T}(\gamma_1) = T_{max}$ and $\hat{T}(\gamma_2) = T_{min}$ and calculate the boundary of the safe set of states on the interior of the envelope by by integrating the system equations backward in time from (V_{min},γ_1) using the constant control (T_{max},θ_{min}). For ease of notation, we denote this part of the safe set

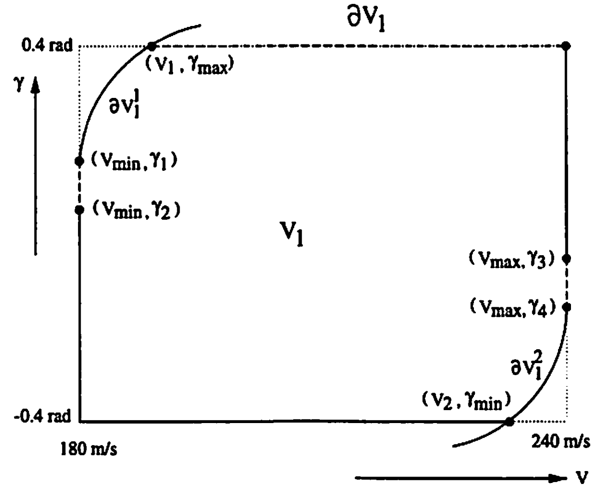

Figure 12: The safe set of states, V_1, and its boundary ∂V_1

boundary on the interior of S as ∂V_1^1, and the point of intersection of ∂V_1^1 with the upper edge of ∂S as (V_1, γ_{max}).

A similar calculation along the upper edge of ∂S using equation (11) yields that the values of θ for which the vector field becomes tangent to ∂S are

$$\hat{\theta}(V) = \frac{m}{a_L V c}\left(\frac{g\cos\gamma_{max}}{V} - \frac{a_L V(1 - c\gamma_{max})}{m}\right)$$

Again, $\hat{\theta}(V)$ does not depend on T, so the set of safe inputs along ∂S may be parameterized solely by θ, and is those θ for which $\theta(V) \le \hat{\theta}(V)$. When V is such that $\hat{\theta}(V) = \theta_{min}$, θ_{min} is the unique pitch angle input which keeps the system trajectory inside S.

The calculations may be repeated for the right hand side and lower boundaries of S. Along the right hand side, the safe set of controls is those T for which $T(\gamma) \le \hat{T}'(\gamma)$, where

$$\hat{T}'(\gamma) = a_D V_{max}^2 + mg\sin\gamma$$

We define γ_3 and γ_4 to be such that $\hat{T}'(\gamma_3) = T_{max}$ and $\hat{T}'(\gamma_4) = T_{min}$ and calculate the boundary of the safe set of states on the interior of the envelope (denoted ∂V_1^2) by integrating the system equations backward in time from (V_{max}, γ_4) using the constant control (T_{min}, θ_{max}). ∂V_1^2 intersects the lower edge of ∂S at (V_2, γ_{min}). All controls are safe for the lower boundary.

We are now in a position to describe explicitly the safe set of states V_1 and the safe controls $\mathcal{U}_1(x^0)$. Define the boundary of V_1 as

$$\begin{aligned}\partial V_1 = \{(V,\gamma) \ | \ & (V = V_{min}) \wedge (\gamma_{min} \le \gamma \le \gamma_1) \vee \partial V_1^1 \vee \\ & (\gamma = \gamma_{max}) \wedge (V_1 \le V \le V_{max}) \vee (V = V_{max}) \wedge (\gamma_4 \le \gamma \le \gamma_{max}) \vee \\ & \partial V_1^2 \vee (\gamma = \gamma_{min}) \wedge (V_{min} \le V \le V_2)\}\end{aligned} \quad (16)$$

V_1 is defined as the set enclosed by ∂V_1 (Figure 12). $\mathcal{U}_1(x^0)$ is defined by the feedback map:

$$G : S \to 2^U$$

$$
G(V,\gamma) = \begin{cases}
\emptyset, & (V,\gamma) \in S \setminus V_1 \\
[T_{min}, T_{max}] \times [\theta_{min}, \hat{\theta}(V)], & (V,\gamma) \in (\gamma = \gamma_{max}) \wedge (V_1 \leq V \leq V_{max}) \\
[\hat{T}(\gamma), T_{max}] \times [\theta_{min}, \theta_{max}], & (V,\gamma) \in (V = V_{min}) \wedge (\gamma_2 \leq \gamma \leq \gamma_1) \\
[T_{min}, \hat{T}'(\gamma)] \times [\theta_{min}, \theta_{max}], & (V,\gamma) \in (V = V_{max}) \wedge (\gamma_4 \leq \gamma \leq \gamma_3) \\
\{T_{max}\} \times \{\theta_{min}\}, & (V,\gamma) \in \partial V_1^1 \\
\{T_{min}\} \times \{\theta_{max}\}, & (V,\gamma) \in \partial V_1^2 \\
[T_{min}, T_{max}] \times [\theta_{min}, \theta_{max}] & \text{otherwise}
\end{cases} \tag{17}
$$

This map defines the *least restrictive* control scheme which satisfies the safety requirement and it determines the mode switching logic. On ∂V_1^1 and ∂V_1^2, the system must be in **Mode 2** or **Mode 3**. Anywhere else in V_1, any of the three modes is valid as long as the input constraints of equation (17) are satisfied. In the regions $S \setminus V_1$ (the upper left and lower right corners of S), no control inputs are safe.

5.3 Additional Constraints for Passenger Comfort

Within the class of safe controls, a control scheme which addresses the passenger comfort (efficiency) requirement can be constructed. To do this, we solve the optimal control problems:

$$
J_2^*(x^0) = \min_{u \in \mathcal{U}_1} J_2(x^0, u), \qquad u^*(x^0) = \arg\min_{u \in \mathcal{U}_1} J_2(x^0, u) \tag{18}
$$

$$
J_2'^*(x^0) = \min_{u \in \mathcal{U}_1} J_2'(x^0, u), \qquad u'^*(x^0) = \arg\min_{u \in \mathcal{U}_1} J_2'(x^0, u) \tag{19}
$$

for $x^0 \in V_1$.

From this calculation, we determine the set of "comfortable" states and controls:

$$
V_2 = \{x^0 \in V_1 | J_2^*(x^0) \leq 0.1g \wedge J_2'^*(x^0) \leq 0.1g\} \tag{20}
$$

$$
\mathcal{U}_2(x^0) = \{u \in \mathcal{U}_1 : J_2(x^0, u) \leq 0.1g \wedge J_2'(x^0, u) \leq 0.1g\} \tag{21}
$$

These sets may be easily calculated by substituting the bounds on the accelerations into equations (10, 11) to get

$$
T \leq 0.1mg + a_D V^2 + mg \sin\gamma \tag{22}
$$

$$
\theta \leq \frac{0.1mg}{a_L V^2 c} - \frac{1 - c\gamma}{c} + \frac{mg\cos\gamma}{a_L V^2 c} \tag{23}
$$

These constraints provide upper bounds on the thrust and the pitch angle which may be applied at any point (V, γ) in V_2, and are illustrated in Figure 13.

6 Conclusions

The first aircraft that flew were essentially experiencing free flight. As air traffic increased, inadequate technology at the time forced standard operational procedures and structured airspace in order to avoid conflicts. This has resulted in a continual sacrifice of airspace

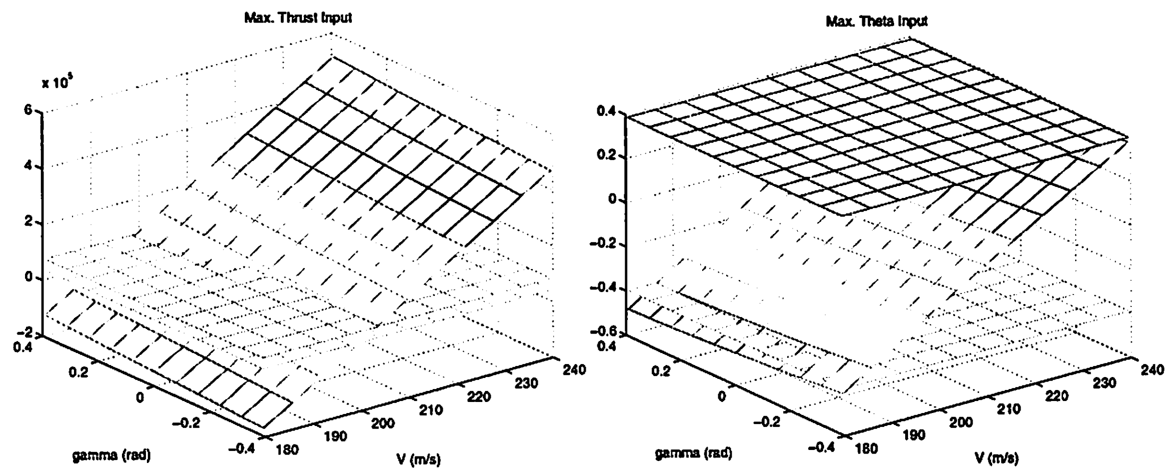

Figure 13: Showing comfort constraint on thrust and pitch angle intersected with existing bounds

utilization and flexibility. Today, technology allows us to remove some of these restrictions and turn back in the direction of free flight.

The technological advances that make the return to free flight feasible include on-board GPS, satellite datalinks, 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 satellite datalinks which will provide this information to sophisticated algorithms, such as the ones presented in this paper. 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.

This is an exciting time in aviation history. In some sense, a new airspace is being completely redesigned by our choices of technological tools and sophisticated algorithms. Different conflict resolution algorithms may result in different macroscopic behaviors of the airspace. Whatever the design choices, however, aviation is moving towards a new era of increased safety and efficiency.

References

[1] H. Erzberger, T. J. Davis, and S. Green, "Design of center-tracon automation system," in *Proceedings of the AGARD Guidance and Control Syposium on Machine Intelligence*

in Air Traffic Management, pp. 11.1–11.12, 1993.

[2] S. M. Green, W. den Braven, and D. H. Williams, "Development and evaluation of a profile negotiation process for integrating aircraft and air traffic control automation," tech. rep., NASA Technical Memorandum, TM 4360, NASA Ames Research Center, Moffett Field CA 94035, April 1993.

[3] R. A. Slattery, "Terminal area trajectory synthesis for air traffic control automation," in *Proceedings of American Control Conference*, 1995.

[4] S. M. Green, R. A. Vivona, and B. Sanford, "Descent advisor preliminary field test." AIAA Guidance, Navigation, and Control Conference, Baltimore, 1995.

[5] S. Kahne and I. Frolow, "Air traffic management: Evolution with technology," *IEEE Control Systems Magazine*, vol. 16, no. 4, pp. 12–21, 1996.

[6] 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 M96/38, Electronics Research Laboratory, University of California, Berkeley, 1996.

[7] C. Tomlin, J. Lygeros, L. Benvenuti, and S. Sastry, "Output tracking for a non-minimum phase dynamic CTOL aircraft model," in *Proceedings of IEEE Conference on Decision and Control*, (New Orleans, LA), pp. 1867–1872, 1995.

[8] G. J. Pappas, J. Lygeros, and D. N. Godbole, "Stabilization and tracking of feedback linearizable systems under input constraints," in *Proceedings of IEEE Conference on Decision and Control*, 1995.

[9] Y.-B. Chen and A. Inselberg, "Conflict resolution for air traffic control," tech. rep., USC-CS-93-543, Computer Science Department, University of Southern California, 1993.

[10] 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.

[11] R. A. Paielli and H. Erzberger, "Conflict probability and estimation for free flight," in *Proceedings of the 35th Meeting of the American Institute of Aeronautics and Astronautics, AIAA-97-0001*, 1997.

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

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

[14] J. Košecká, C. Tomlin, G. Pappas, and S. Sastry, "Generation of conflict resolution maneuvers for air traffic management." submitted to the IEEE Conference on Intelligent Robots and Systems, 1997.

[15] J. Lygeros, D. Godbole, and S. Sastry, "A game theoretic approach to hybrid system design," tech. rep., UCB-ERL Memo M95/77, Electronics Research Laboratory, University of California, Berkeley, CA 94720, 1995.

[16] J. Lygeros, C. Tomlin, and S. Sastry, "Multiobjective hybrid controller synthesis." To appear in the Proceedings of the International Workshop on Hybrid and Real-Time Systems (Grenoble), 1996.

[17] C. Hynes and L. Sherry, "Synthesis from design requirements of a hybrid system for transport aircraft longitudinal control." preprint, NASA Ames Research Center, Honeywell Air Transport Division, 1996.