HYBRID CONTROL IN AIR TRAFFIC MANAGEMENT SYSTEMS

by

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

Memorandum No. UCB/ERL M95/82

18 October 1995

HYBRID CONTROL IN AIR TRAFFIC MANAGEMENT SYSTEMS

by

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

Memorandum No. UCB/ERL M95/82

18 October 1995

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Hybrid Control in Air Traffic Management Systems*

S. Sastry, G. Meyer†, C. Tomlin, J. Lygeros, D. Godbole and G. Pappas
Intelligent Machines and Robotics Laboratory
Department of Electrical Engineering and Computer Science
University of California   Berkeley, CA 94720-1770

Abstract

In a new collaborative project involving the University of California, Berkeley, NASA Ames Research Center, and Honeywell Systems Research Center, we have begun the study of hierarchical, hybrid control systems in the framework of air traffic management systems (ATMS). The need for a new ATMS arises from the overcrowding of large urban airports and the need to more efficiently land and take off larger numbers of aircraft, without building new runways. Technological advances that make a more advanced air traffic control system a reality include the availability of relatively inexpensive and fast real time computers both on board the aircraft and in the control tower. The usefulness of these technological advances is currently limited by today's air traffic control system, which involves the use of "sky freeways" in both the Terminal Radar Approach Control (TRACON) region around an urban airport and in the airspace outside of the airport TRACON. These freeways are set approach patterns to airports which do not allow for the possibility of so-called "free flight" by an aircraft to its destination. Limiting the aircraft trajectories in this manner results in the addition of both planned and unplanned delays to air travel. We propose an architecture for an automated ATMS with decentralized control, and with free flight in effect outside of the TRACON region.

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

*Research supported in part by NASA under grant NAG 2-243.
†Address: MS 210-3, NASA Ames Research Center, Moffett Field CA 94035.

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 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.

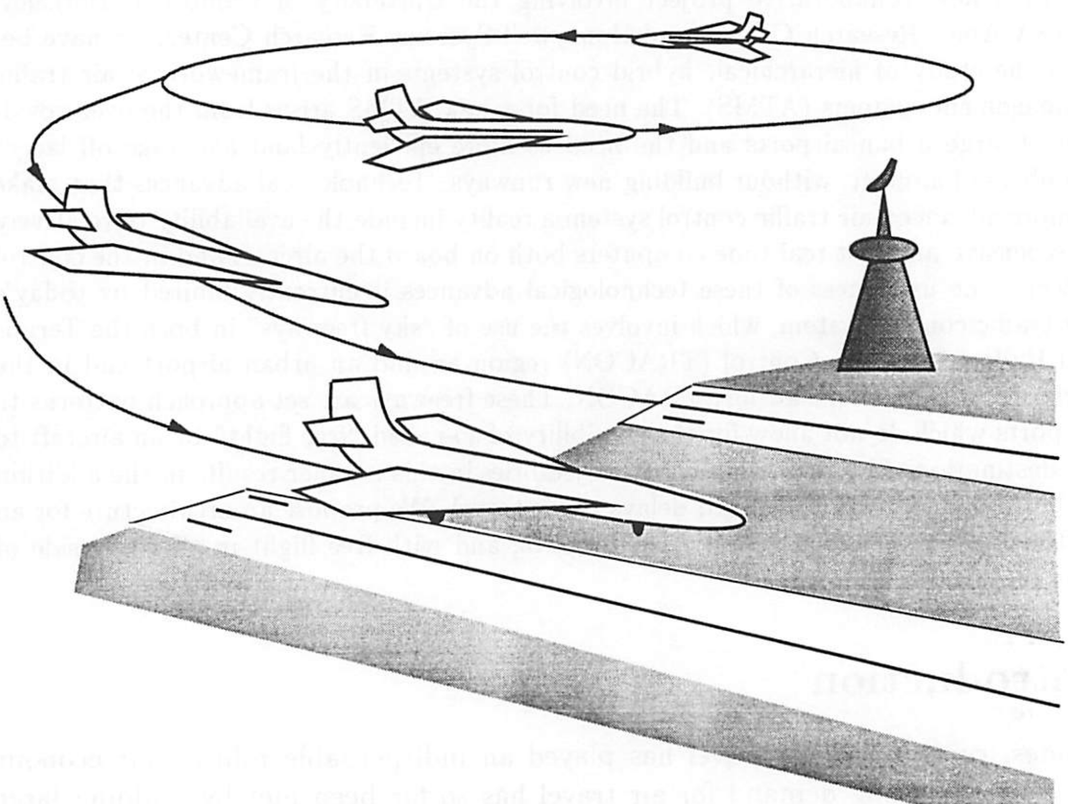

Figure 1: Current Airport Landing Patterns

The air traffic management system (ATMS) we envision will be fully automated[1]. The proposed new architecture for ATMS is inspired by our research on the control of *hierarchical*

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

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 in 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. This marriage of discrete event and continuous dynamics results in an overall system which is hybrid.

The key to our approach is the use of inter-agent coordination to increase aircraft autonomy and consequently reduce the ATC workload. In order to motivate our work we first give a brief overview of current ATC practice, in Section 2. In Section 3 we describe our proposed air traffic management system, including discussions about how interagent coordination would work to resolve conflicts, and a comparison of a decentralized control scheme featuring interagent coordination with a centralized scheme in which ATC is the central controller. The similarities and differences between the proposed architecture and a similar design in the context of automated Highway Systems are discussed in Section 4. Control issues raised by the proposed design will be raised in Section 5. We conclude by listing the problems that need to be addressed before our design can be implemented.

2 Current ATC Practice

Air traffic control (ATC) in the United States is currently organized by geographical region. The country is divided into 20 *centers*, each with its own ATC group. In addition, around each large urban airport is a *TRACON* region with its corresponding ATC group. The main goal of both the ATC in the centers and in the TRACONs is to maintain safe spacing between aircraft while guiding the aircraft to their destinations. Due to their heavy workloads, minimizing flight times and fuel spent are not prime considerations of controllers when they determine trajectories for the aircrafts to follow. As a result, the current ATC system is inefficient. These inefficiencies cause unplanned delays in average flight times, and thus the airline schedules are not accurate and the controllers are forced to manually schedule and reschedule aircraft landings according to when the aircraft enters the TRACON region. In addition, there is minimal communication between the center ATC and the ATC of a TRACON inside this center, 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 well-traveled "freeways" in the TRACON airspace: they force some aircraft to circle the airport while they concentrate on landing others.

Figure 2 depicts the horizontal projection of a typical route inside the TRACON. Because aircraft must land into the wind 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 as they enter the TRACON,

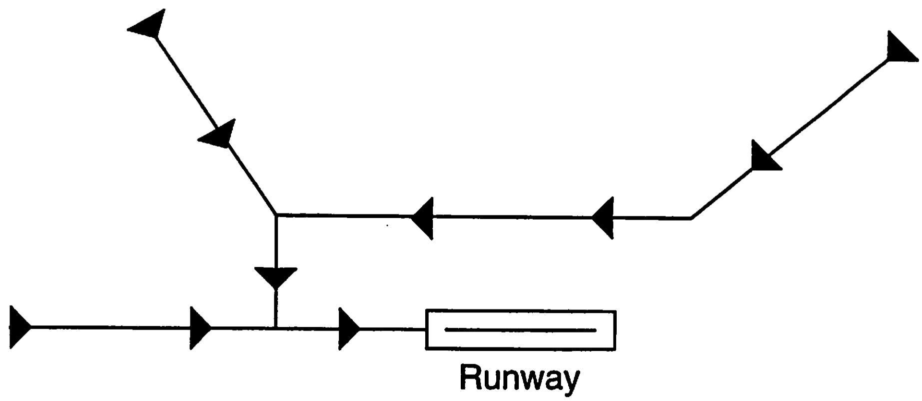

Figure 2: Typical route pattern for arriving aircraft

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 Center-ATC performs 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 Center-ATC also uses predefined air routes (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 foot altitude while eastbound fly at odd thousand feet);

- Even if the optimal route of an aircraft takes it above an airport enroute to another, ATC directs the aircraft *around* the intermediate airport so that the TRACON-ATC's workload is not increased;

- Shifting winds at airports cause havoc in scheduling, since the airport must be reconfigured to use different runways, and aircraft are delayed as a result;

- 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.

In order to improve the efficiency of the current ATC system, NASA Ames is 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 a graphical 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].

3 Proposed Air Traffic Management System

We propose a partially decentralized air traffic management system which transfers some of the current air traffic control functionality to the individual aircrafts. The proposed system is expected to be more robust and reliable, reduce ATC workload, and be more suitable for free flight. We discuss the advantages of such an architecture in the next section, and in the following sections we describe the proposed architecture.

3.1 A Decentralized Decision Making System

We believe that an air traffic management system which distributes decision making about trajectory planning and conflict resolution to the aircraft has clear advantages over today's system, which is centrally controlled by ATC.

- A distributed system is more fault tolerant. If a single aircraft's computer system fails, most of the ATMS system is still intact, and the aircraft may be guided by voice to the nearest airport. If in a centralized system the central computer fails, the results could be catastrophic.

- Congestion is increasing monthly in large urban airports. A distributed system is more suited to handling increasing numbers of aircraft than is a centralized system, since a new aircraft and its own computer system may be added easily to the system. A centralized system would require upgrades to the ATC computer regularly.

- Free flight, one of the features of a distributed control system, minimizes fuel consumption since each aircraft may optimally plan its own trajectory, which in turn results in reduced delays outside TRACONs. Free flight is also an advantage in avoiding turbulence, since the aircraft would not have to wait for clearance from ATC to avoid rough weather patches.

- A distributed system reduces the ATC workload, allowing ATC to spend more time in resolving safety critical situations.

- In a centralized system, because of the large number of aircraft it has to deal with, ATC can only use a rough approximation of each aircraft's dynamics to calculate feasible trajectories. In our ATMS model, each aircraft's autopilot contains a detailed model of its own dynamics, and thus the aircraft itself is well equipped to plan its own trajectory in a conflict resolution maneuver.

3.2 Conflict and Coordination

With each aircraft free to plan and track its own trajectory, we need to examine the inter-aircraft conflict and coordination.

3.2.1 The origins of conflict

In a multiagent setting there is always the possibility of conflict between the objectives of the agents. In the ATMS problem, the most dangerous of these is that of conflicting trajectories, a situation that is very likely, especially if one allows aircraft to independently plan and track their trajectories. An automated ATMS should be able to predict such conflicts well in advance and resolve them by altering the plans of one or more of the agents.

Even though quantitatively all path conflicts are the same (two or more aircraft occupy the same space at the same time or, more precisely, get closer than a certain threshold to one another), we can distinguish certain qualitative scenarios. Here we introduce three such examples which we call *Merge*, *Overtake* and *Collision*[2]. The Merge conflict is the situation depicted in Figure 3. Aircraft 1 (solid line) and 2 (dashed line) are coming from different

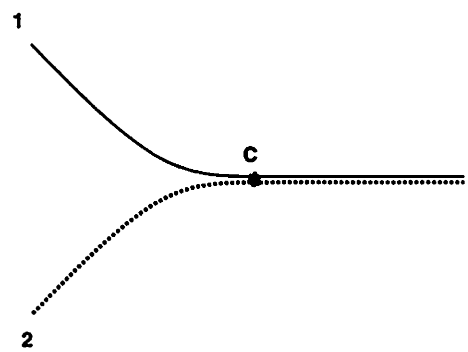

Figure 3: Merge conflict

directions or altitudes (or both) but want to follow the same path from point C onwards. We will refer to point C as the *conflict point.* In Overtake (Figure 4) the aircraft are already moving along the same path, with 2 trailing 1 and moving faster. At the conflict point, C, 2 catches up with 1. Finally in the Collision conflict (Figure 5) the aircraft trajectories cross transversally or meet head on.

Conflict resolution for these examples can be carried out in a number of ways. For Merge, aircraft 1 can decelerate while aircraft 2 accelerates (or vice versa) to create acceptable time separation at the conflict point C. Overtake can be resolved by a transient change in altitude or/and direction of aircraft 2 (dashed line in Figure 4). Finally the Collision conflict can be resolved by any of the techniques used for Merge and Overtake. Which of the possible

[2]The names reflect similar situations encountered in highway traffic

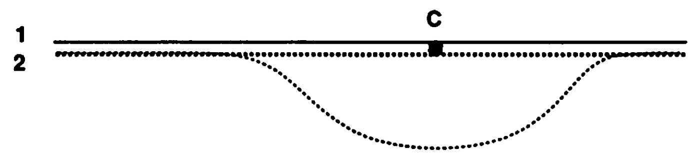

Figure 4: Overtake conflict

actions is taken should be dictated by considering factors such as the passenger comfort, fuel efficiency, and resulting delay.

3.2.2 Resolution by coordination

Traditionally, conflict resolution has been carried out in a centralized manner by the ATC. Modern approaches to ATC make use of simple kinematic models to extrapolate aircraft trajectories and predict conflicts a short while into the future. The system is used to inform the human controllers of possible conflicts and provide suggestions for resolution. Only in cases of emergency (for example when the ATC is disabled by a power failure) is communication between pilots used to resolve conflicts.

We propose to distribute the responsibility of trajectory planning and conflict resolution to the individual agents. Distributed decision making, however, implies the need for inter-agent communication and coordination as the agents that were centrally controlled become semi-autonomous and are forced to cooperate to achieve a common goal.

Why is coordination (as opposed to fully autonomous operation) necessary in an ATMS which features a distributed decision making process? The simple examples introduced in the previous section indicate that one of the main reasons is safety. Coordination is needed to guarantee that the agents do not take contradictory actions. To resolve a head on Collision, for example, both aircraft may decide to lose altitude, resulting in a new conflict. Similarly for the Merge conflict both planes may decide to decelerate. Finally, for Overtake conflict, coordination may be needed to guarantee that the overtaken plane does not accelerate while the maneuver is in progress and that it is not going slowly for a special reason (e.g. because it is involved in a Merge with a third aircraft). Coordination is also needed for efficiency and comfort. As discussed above, it may be possible to resolve a conflict in a number of different ways and considerations of comfort and efficiency become important in choosing between them. For example it may be better for an aircraft that is banking to the right to turn right to resolve a collision conflict, rather than turning left or losing altitude.

In the cases discussed here the necessary coordination can be established by simple com-

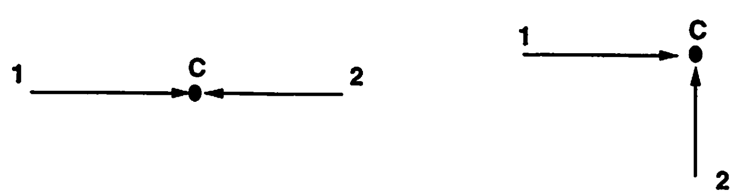

Figure 5: Collision conflict

munication protocols. More complicated protocols may be needed for other maneuvers (e.g. conflicts involving more than two aircraft). The protocols must be designed so that local coordination between a few aircraft has minimal adverse effects on other aircraft in the surrounding region. Once protocols have been designed they need to be formalized and verified. More importantly, the hybrid interaction between the protocols and the continuous dynamics has to be investigated.

3.3 Proposed Architecture for ATMS

We propose an architecture for a fully automated air traffic management system. The functionality of the architecture is slightly different for the region inside the TRACON (take off and landing) than it is for the region outside the TRACON (en-route airspace). In the region outside the TRACON, the density of aircraft is relatively small, and we propose that each aircraft be allowed to completely plan its own trajectory (to be called the "free flight" region). In this region, conflicts will be resolved by coordination between aircraft, the role of the ground based *Air Traffic Control* (ATC) will simply be to provide information about the route, such as upcoming weather conditions. Inside the TRACON, the ATC will determine the approach trajectory for each aircraft. These trajectories will depend on local weather conditions and traffic. A block diagram of our proposed architecture for ATMS inside the TRACON region is presented in Figure 6 and that for ATMS outside the TRACON is presented in Figure 7. 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.

3.3.1 Air Traffic Control

The functionality of ATC differs in the two architectures: ATC has more control over aircraft in the TRACON than over aircraft in the en-route airspace. When an aircraft enters the TRACON region of an airport, ATC passes a sequence of *waypoints* to the strategic planner on board the aircraft. 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;
- 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 schedule of landings, which is designed to meet the announced arrival times and reflects conflict resolution and compromises between airline schedules. Once ATC has calculated these waypoints and passed them to the strategic planner, all of the planning and control tasks are taken over by the FVMS on board the individual aircraft. If the FVMS changes the waypoints for some reason (such as a safety critical situation), then the new set of waypoints is passed to ATC.

Outside the TRACON region, free flight is under effect and the role of ATC is minimal. ATC passes only TRACON exit and entry information to the tactical planner of the aircraft, and then the tactical planner takes over the role of calculating an initial kinematic trajectory for the aircraft.

3.3.2 Strategic Planner

The main role of the strategic planner is to resolve conflicts between aircraft. Inside the TRACON, the strategic planner has the additional role of designing a coarse trajectory for the aircraft in the form of a sequence of control points, c_k, which interpolate the waypoints from ATC.

If the tactical planner on board the aircraft predicts that a conflict will occur between its aircraft and other aircraft, it notifies the strategic planner. As discussed in Section 3.2, conflict resolution is achieved by communication between aircraft at the strategic level. The strategic planners of all aircraft involved in the potential conflict determine a sequence of maneuvers which will result in conflict-free trajectories, and then each strategic planner commands its own tactical planner to follow these maneuvers. These maneuvers and resulting commands are accessed from a database of precomputed solutions to possible conflicts. Inside the TRACON, the commands are passed down to the tactical planner in the form of a modified sequence of control points. ATC is notified if this modified sequence deviates from the original set of waypoints. Outside the TRACON, where the upper levels of the ATMS architecture have looser control over trajectory planning, the commands themselves are passed.

3.3.3 Tactical Planner

Inside the TRACON, the tactical planner refines the strategic plan by interpolating the control points with a smooth output trajectory, denoted y_d in Figures 6 and 7. Outside the TRACON, it calculates this trajectory from scratch, using only the initial and final conditions of free flight (the TRACON exit and entry information). In addition to the output trajectory, the tactical planner determines the sequence of *flight modes* necessary to execute the kinematic plan. In both regions, the tactical planner is responsible for predicting conflicts.

The tactical planner uses a simple kinematic model of the aircraft for all trajectory calculation. 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 where the cost function is 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 we assume that the aircraft in question is allowed to move away from its neighbors, conflict prediction becomes a pursuit-evasion dynamical game ([5], [6]), with the neighbors pursuing (doing their best to cause conflict) and our aircraft evading (doing its best to avoid it).

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.

The kinematic model and the planning techniques used by the tactical level can play a crucial role in the closed loop system performance as well as the verification process. One approach, investigated in [7], deals with the so-called "landing tower problem": given a sequence of waypoints, specifying the position and orientation of the aircraft, and times of arrival by the control tower, plan a trajectory which interpolates these points. This problem was approached using a kinematic model for the motion of the aircraft. More specifically, let $g \in SE(3)$ model the position and orientation of the aircraft, the inputs u_1, u_2, u_3 stand for the rates of rotation about the principal (body) axes of the aircraft and v the velocity of the (CTOL) aircraft. Then the kinematic equations of motion are given by:

$$\dot{g} = g \begin{bmatrix} 0 & -u_3 & u_2 & v \\ u_3 & 0 & -u_1 & 0 \\ -u_2 & u_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \qquad (1)$$

Here g is represented in homogeneous coordinates and the velocity v is normalized to 1, representing the fact that the velocity of the aircraft is roughly constant. We solve the problem of steering the aircraft between the initial and final configurations, g_o and g_f in "optimal fashion", that is we determine trajectories that minimize:

$$\int_0^T \sum_{i=1}^{3} u_i^2 \, dt.$$

This problem was solved in [7]. It was shown that the optimal inputs are elliptic functions (so-called Weierstrass P-functions). Computational procedures for steering the system have also been devised. One very interesting aspect of solution is that it includes *helices, circles* and *straight line* segments. This is especially pleasing, since such solutions are frequently used by the current ATC. The above solution is currently being refined to reduce the use of the yawing input and to treat the velocity of the aircraft as an additional input. Computational approaches to the efficient generation of trajectories are also being pursued.

3.3.4 Trajectory Planner

The trajectory planner uses a detailed dynamic model of the aircraft, sensory input which measures the wind's magnitude and direction, and the tactical plan consisting of an output trajectory and sequence of modes, to design a full state and input trajectory for the aircraft. This trajectory, denoted y_d, x_d, and u_d in Figures 6 and 7, 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 and actuator saturation.

Non-Minimum Phase Effects

To illustrate the problems associated with non-minimum phase dynamics, consider a planar, dynamic CTOL model, which describes the longitudinal axis dynamics of the aircraft in normal aerodynamic flight [8]. The horizontal and vertical axes are respectively the x, z axes and θ is the angle made by the aircraft axis with the x axis (*pitch angle*). There are two inputs available to the system: the thrust along the aircraft axis, u_1, and the pitch moment u_2. The *flight path angle*, γ, is defined as:

$$\gamma = \tan^{-1}(\frac{\dot{z}}{\dot{x}}),$$

and the *angle of attack*, α, is defined as:

$$\alpha = \theta - \gamma.$$

The aerodynamic forces of lift (L) and drag (D) are given by:

$$\begin{aligned} L &= a_L(\dot{x}^2 + \dot{z}^2)(1 + c\alpha) \\ D &= a_D(\dot{x}^2 + \dot{z}^2)(1 + b(1 + c\alpha)^2). \end{aligned} \tag{2}$$

where a_L, a_D are the lift and drag coefficients, and b and c are dimensionless positive constants. In this notation, the equations of motion are:

$$\begin{bmatrix} \ddot{x} \\ \ddot{z} \end{bmatrix} = R(\theta)\left[R^T(\alpha)\begin{bmatrix} -D \\ L \end{bmatrix} + \begin{bmatrix} u_1 \\ -\epsilon u_2 \end{bmatrix}\right] + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tag{3}$$

$$\ddot{\theta} = u_2 \tag{4}$$

$R(\alpha), R(\theta)$ are rotation matrices of the form:

$$R(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}. \tag{5}$$

In this model, all masses and moments of inertia have been normalized and the gravity has been set to one unit.

The non-minimum phase characteristic of this model is a result of the fact that the process of generating an upward pitch moment produces a small parasitic downward force. Input-output linearization allows the x and z position of the aircraft to be tracked exactly, but provides no way of regulating the dynamics of the pitch angle θ (zero or inverse dynamics). These dynamics can become unstable, in view of the non-minimum phase nature of the plant. We are currently studying different methods for both exact and approximate tracking of a given desired trajectory. In terms of exact tracking, we have applied the algorithm proposed in [9] and [10] to this model. These methods need an "acausal" look-ahead for generating the control law and typically lead to large feed-forward inputs. This may be undesirable, in view of the subsequent discussion on actuator saturation. In contrast the approximate linearization techniques proposed in [12] and [13] do not require lookahead and result in reasonable tracking inputs. A detailed discussion of all these issues can be found in [14].

Actuator Saturation

Because of dynamic constraints it may turn out that the kinematic trajectory is infeasible. The reason may be that it requires inputs exceeding the actuator capabilities, or that it implies unrealistic state values (e.g., angles of attack beyond stall). Even feasible kinematic trajectories may be undesirable if they violate limits on orientation and velocity dictated by passenger comfort. In the case of unacceptable kinematic trajectories, the trajectory planner notifies the tactical planner to replan. A possible time reparameterization of the trajectory is suggested though the signal $s(t)$, such that $y_d(t) \longrightarrow y_d(s(t))$. If, inside the TRACON, the tactical planner is unable to find an acceptable kinematic plan which interpolates the original waypoints, the strategic planner is notified to renegotiate waypoints with the ATC and neighboring aircraft.

To motivate the problem of actuator saturation consider a single input, single output, affine nonlinear system:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{6}$$

with state $x \in \Re^n$, input $u \in \Re$, output $y \in \Re$, vector fields f, g on $\Re^n$ and output function h. Assuming that system (6) has well defined relative degree γ, standard feedback linearization techniques lead to the following control law:

$$u = \frac{1}{L_g L_f^{\gamma-1} h(x)}(-L_f^{\gamma} h(x) + y_r^{\gamma}(t) - c^T e) \tag{7}$$

In the absence of input saturation this law results in asymptotic output tracking of a desired trajectory $y_r(t)$. $e \in \Re^\gamma$ stands for the tracking error vector of the output and its first γ derivatives, and the feedback gain $c \in \Re^\gamma$ has been appropriately selected to make the error polynomial Hurwitz.

The tracking control law (7) suggests that the control cost associated with asymptotically tracking a desired trajectory can be broken up into a cost of feedback linearization $(\frac{-L_f^{\gamma} h(x)}{L_g L_f^{\gamma-1} h(x)})$, a cost of exactly tracking the trajectory $(\frac{y_r^{\gamma}(t)}{L_g L_f^{\gamma-1} h(x)})$ and a cost of stabilizing the tracking error to zero $(\frac{c^T e}{L_g L_f^{\gamma-1} h(x)})$. In the presence of control constraints, these costs are competing with each other. For example, there is a tradeoff between the space of feasible trajectories and the region of attraction of the tracking error. The feasible space of trajectories is maximized when the tracking error is zero and, conversely, for sufficiently large values of the error the space of trajectories becomes empty.

This problem may be encountered in the case of trajectory planning and tracking in ATMS. In an aircraft there are inherent actuator constraints as well as thrust limitations and the additional complication that the system is multi-input, multi-output. While the trajectory planner is working on a dynamic trajectory, it may discover that the input cost from linearization is large and does not leave enough room for the exact trajectory tracking cost. Similarly, while the regulation level is tracking the dynamic trajectory, it may discover that the tracking error cost becomes large enough to cause actuator saturation. In either case the tactical planner needs to be notified to replan the kinematic (output) trajectory. As the trajectory planner makes use of the detailed dynamical model, it can help with the replanning by suggesting possible alternatives. An easy modification that may work in some cases involves reparametrizing (slowing down) the existing trajectory with respect to time. This typically results in smaller feed-forward inputs, but may be insufficient in extreme cases.

A more detailed treatment of feedback linearization in the presence of constraints may be found in [15]. The problem of stabilization of linear and nonlinear systems in the presence of actuator saturations has also been extensively studied in [16, 17]; the stabilization approaches are inapplicable to trajectory tracking, however. The analysis of [15] also addresses the issue of tracking.

3.3.5 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 Comparison to the PATH architecture

4.1 Similarities to the PATH AHS architecture

The ATMS control architecture proposed here is comparable to the PATH architecture of [18] for Automated Highway Systems (AHS), which is also in the form of a multi-level hierarchy (Figure 8).

The PATH control hierarchy consists of four layers: the *regulation layer*, *coordination layer*, *link layer* and *network layer*. The aim of this architecture is to allow cars to form platoons (software trains) which move on the highway under computer control at relatively low intervehicular spacing (1-3 meters) and at relatively high speeds. The expectation is that the inter platoon spacing can be maintained large enough to be able to keep disturbances in the highway traffic from propagating upstream. The link and network layers determine trajectories for individual vehicles involving trajectories in different lanes in order to maximize the throughput of the highway and thereby minimize transit time on the average. The coordination layer receives the planning commands of the link layer and converts them into an alphabet of atomic maneuvers for the individual vehicles. For platooning these maneuvers are *join* to form a platoon, *split* to break up a platoon, *change lane* to move from one lane to another, *entry* to get on the highway and *exit* to get off. These maneuvers result in specific control actions taken by the regulation layer. In terms of abstraction, the regulation layer is continuous, the link layer is discrete[3] while the coordination layer is hybrid.

The resemblance between the AHS and ATMS architectures should come as no surprise, since in both cases the objective is the efficient utilization of a scarce resource (either highway or runway) by a large number of semi-autonomous agents (vehicles or airplanes). For AHS, the lower level is responsible for the regulation of individual vehicles and makes use of very detailed, differential equation models for the vehicle and engine dynamics. In terms of the proposed ATMS architecture this would correspond to the trajectory planner and the regulator. The intermediate level (coordination) is responsible for macroscopically planning the vehicle trajectory, which it does by selecting a sequence of maneuvers. To carry out these maneuvers it first communicates with neighboring vehicles to guarantee their cooperation and then commands switching between the various regulation layer control laws. The actions of the coordination layer are based on a more abstract, discrete model of the vehicle behavior. In terms of the ATMS architecture this would correspond to the strategic and tactical planning levels. Finally, the link and network layers are responsible for the centralized control of vehicles in large sections of the highway. As with the ATC level of the ATMS architecture, their objective is to maximize the utilization of the system, which in the case of AHS implies maximizing the throughput of the highway. It should be noted that, similar to the ATMS, the coordination and regulation layers control vehicles in a decentralized manner and effectively reside on individual vehicles. Decisions at the link and network layers, on the other hand, are taken by roadside controllers, that model the traffic on the given section using queuing or flow models.

4.2 Differences from the PATH AHS architecture

While there are many resemblances between the PATH AHS architecture and the ATMS architecture that we propose, there are also some important differences:

1. *State Space of an Individual Agent:* Because the aircraft motion takes place in three dimensions, its state space is the manifold $SE(3)$, as opposed to the predominantly one and a half dimensional state space of road vehicles. A direct consequence of this is that the description of the state space has to be in terms of charts. The extra dimensions of the state space give us greater freedom in planning trajectories, but also make the

[3]Though the actions of the link layer may be determined by continuum approximations to the traffic.

planning process harder. This is the reason for the greater number of planning layers and the hybrid nature of the strategic and tactical planners in the ATMS architecture.

2. *Interagent Coordination:* For each vehicle moving on the highway it is relatively simple to identify neighboring vehicles (front, back, left and right) and thus define a "neighborhood" that roughly moves along with the vehicle in question. This allows for communication and coordination between neighboring agents, without intervention of the link layer. Thus safe operation of the system can be maintained by appropriate design of the decentralized layers, leaving the centralized layers to deal with improving efficiency. The three dimensional motion of the aircraft, on the other hand, makes it harder to define the "neighbors" of each agent. As a result conflict prediction and resolution may be significantly more complicated and two planning layers (strategic and tactical) are needed to replace the coordination layer.

3. *Different Dynamics:* The dynamics of the aircraft are strongly nonlinear and depend in a subtle fashion on the aerodynamic forces of lift and drag. These forces may be ignored or very coarsely approximated in the case of vehicle dynamics. Furthermore, since the aircraft needs to move in order to generate lift, it is not possible to ask an aircraft to stop in mid-flight. Finally, the aircraft control designer has to deal with a greater number of inputs which are strongly coupled to one another. For highway operation on the other hand, when the steering angles are small, the coupling between the lateral and longitudinal inputs of the vehicle is very weak. The complicated nonlinear dynamics make the continuous control a lot more challenging and dictate the presence of two distinct continuous levels, the trajectory planner and the regulation. For AHS both the planning and regulation can be dealt with by one layer.

4. *NonHomogeneous Agents:* The agent dynamics also vary greatly from aircraft to aircraft. The ATMS system must have the breadth to incorporate both small, private aircraft and large, commercial jets.

5. *Number of Agents:* Because of the large numbers of vehicles that will use the automated highway (of the order of 5000 per lane per hour) it is important in the AHS to have an aggregate way of describing strategies for maximizing throughput. This dictates the presence of two roadside planning levels, the network and link, one dealing with the entire highway system and the other with pieces of highway a few miles in length. The target take offs and landings for ATMS are of the order of 45-60 per hour per runway, which is also large, but an order of magnitude smaller than the AHS. Hence a single layer ATC will probably be sufficient.

5 Design Issues

5.1 Controllers with Switches

While certain subclasses of switched dynamical systems such as those arising from relaxed control or sliding mode control have been well investigated, it is fair to say that a systematic investigation of switching between different nonlinear control laws is still in its infancy. The literature in gain scheduling does little from a design point of view in establishing rules of transition to preserve continuity of the controller inputs. We have begun a systematic study of performance specifications and design guidelines for control systems with switches. We will be aided in this regard by some related work in the project on highway automation. The hybrid nature of the AHS problem is due to two types of switching. The first is caused by switching between different maneuvers commanded by the coordination layer (planning) and the second is introduced because of the physical layer constraints. For example, in the autonomous vehicle controller of [19], phase space based switching between different feedback control laws was introduced to guarantee that the system response to disturbances lies within state and input constraints due to actuator saturations and passenger comfort limitations respectively.

Similarly, the air traffic management control system involves switching at different levels. At the higher level, the tactical planning level introduces switching between different operating modes for the aircraft. For every mode segment, the trajectory planner designs a smooth trajectory. While the underlying dynamics of the aircraft are given by smooth functions, the control laws based on full state approximate linearization are different in different modes of operation. This has to do with changes in the independent and dependent outputs of the aircraft depending on the flight regime. Thus the regulation level controller is switched at each instant of mode switching. In addition, the coordinate charts that are used to parametrize trajectories in SE(3) also encounter singularities. At this point, one has to switch to a different chart and correspondingly a different control law.

Another kind of switching is necessitated by the disturbances. Because the aircraft engine and control surfaces can produce a limited amount of thrust and moments, the space of trajectories that can be tracked satisfactorily by an aircraft is limited. The effect of small disturbances (as well as passenger comfort violations) may be nullified by reparametrization of the trajectory as mentioned in the previous section. In the case of severe disturbances, such as wind shear, the overall sequencing of modes has to be changed thereby introducing switching in the controller. Some of these disturbances may be due to changes in plans of other aircrafts in the vicinity.

Most of the controllers will be relatively easy to analyze individually, using results from nonlinear control theory. The analysis of the overall switched system can be very tricky however, as indicated by the examples considered in [19, 20]. The problem becomes even more complicated when the planning layers are added to the picture. The additional complication is due to the fact that at the planning level we not only have to deal with switched continuous system but we also have to consider the change in the model abstraction from continuous to discrete.

5.2 Specification and Verification

In systems which have comparable functional and hierarchical complexity of flight control, systematic design tools for verifying that the control schemes meet the specification need to be developed. The discrete part of the model can be designed and analyzed using standard tools (typically computational) such as STATEMATE or COSPAN. Even after this kind of analysis however we can not be sure that the resulting hybrid system will perform adequately. An investigation carried out on the AHS system [21] indicates that, even if the individual subsystems are verified independently, the overall system can fail in a critical way.

The problem can be reduced to showing that the discrete models used by the higher levels to describe the lower levels (including the plant) are indeed an abstraction of the behavior of those levels. More formally, in discrete event system terminology, we would like to show that the language generated by the lower levels (when looked at from the discrete point of view of the higher layers) is contained in the language of the related abstraction used for the design of the higher level controllers. The level of sophistication of current hybrid systems methodologies is severely limited to the analysis of systems with clocks or systems whose dynamic performance can be abstracted by clocks. New frameworks and a fresh new approach to these problems is warranted. Indeed for the ATMS problem, an accurate timed abstraction of the continuous dynamics will be difficult to obtain because of the complicated nonlinear dynamics of the plant and the on-line reparametrization of the reference trajectory being carried out between the trajectory planning and regulation layers to avoid saturation limits. We also need to show that the language generated by the discrete layer (which is a sequence of control modes) is rich enough to monitor and control the continuous dynamics.

Ultimately the actions of the ATC also need to be verified. A similar abstraction problem and language containment problem will have to be dealt with in this case as well. Additional complications arise from the multiagent character of the system at this level. We hope that by comparing and contrasting the methodology applied to the ATMS problem with the one used for the AHS problem we will be able to draw conclusions about basic, generic properties of hybrid and hierarchical systems.

5.3 Simulation & Visualization Issues

The complexity of large scale projects, such as the proposed Air Traffic Management System, renders simulation a valuable tool both in the design of various control laws and coordination protocols as well as in the evaluation of overall system performance. Furthermore, a good simulation package may also be used as a debugging tool in the design process. This requires the development of a simulation package for hybrid systems that will be able to simulate both the low level differential equation models as well as the high level finite state machine models. The complexity of the system also emphasizes the need for efficient computational schemes, such as parallel computation algorithms.

Due to the large size of the project, each simulation run results in a tremendous number of data that need to be analyzed and interpreted. Visualization techniques, such as animation, can be used to present the simulation results in a manner which is much easier to analyze by the designer. In this direction, we have started the development of *SmartPlanes*, a simulation and visualization facility for ATMS. At the current stage SmartPlanes, which is shown in Figure 9, is a visualization tool which allows the user to view the trajectory of a single aircraft from various perspectives. For example, the user has a choice to view the aircraft from the control tower, from a fixed location, or to have the pilot's perspective from the cockpit. In future versions, multiple aircraft will be shown as well as a local radar. Moreover, the user will have the ability to configure his/her own airport so as to meet the needs of different cities, e.g., Denver Airport, JFK International Airport, etc.

6 Concluding Remarks

The details of the architecture presented here need to be worked out. Possible connections between the ATC work carried out in NASA and the requirements of the automated design need to be investigated. The resolution protocols of the strategic planner need to be designed from scratch and the interaction with the ATC needs to be formalized. More work needs to be done for the tactical planner, especially in terms of conflict prediction. Also the trajectory planner and regulation level design needs to be completed. More importantly the hybrid interaction between all those layers needs to be investigated. For this task the development of a flexible simulation platform will be crucial. Finally, some analysis from the point of view of transportation studies needs to be carried out to estimate the improvement in terms of fuel consumption, travel times, safety, passenger comfort, etc. that can be expected if the proposed system is implemented in practice.

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 *American Control Conference*, 1995.

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

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

[6] 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.

[7] G. Walsh, R. Montgomery, and S. Sastry, "Optimal control on lie groups," in *IEEE Control and Decision Conference*, 1994.

[8] S. Sastry, "Nonlinear systems: Analysis, stability and control." to be published by Addison-Wesley, 1996.

[9] P. Martin, S. Devasia, and B. Paden, "A different look at output tracking: control of a VTOL aircraft," in *Proceedings of the 33rd IEEE Conference on Decision and Control*, pp. 2376–2381, 1994.

[10] S. Devasia and B. Paden, "Exact output tracking for nonlinear time-varying systems," in *Proceedings of the 33rd IEEE Conference on Decision and Control*, pp. 2346–2355, 1994.

[11] G. Meyer, L. R. Hunt, and R. Su, "Nonlinear system guidance in the presence of transmission zero dynamics," tech. rep., NASA Technical Memorandum, TM 4661, NASA Ames Reseacrh Center, Moffett Field CA 94035, March 1995.

[12] J. Hauser, S. Sastry, and G. Meyer, "Nonlinear control design for slightly nonminimum phase systems: Application to V/STOL aircraft," *Automatica*, vol. 28, pp. 665–679, 1992.

[13] L. Benvenuti, M. D. Di Benedetto, and J. W. Grizzle, "Approximate output tracking for nonlinear non-minimum phase systems with an application to flight control," *Journal of Nonlinear and Robust Control*, vol. 4, pp. 397–414, 1993.

[14] C. Tomlin, J. Lygeros, L. Benvenuti, and S. Sastry, "Output tracking for a non-minimum phase dynamic CTOL aircraft model," in *IEEE Control and Decision Conference*, 1995.

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

[16] E. Sontag and H. Sussmann, "Nonlinear output feedback design for linear systems with saturating controls," in *Proceedings of the 29th IEEE Conference on Decision and Control*, pp. 3414–3416, 1990.

[17] A. R. Teel, *Feedback Stabilization: Nonlinear Solutions to Inherently Nonlinear Problems.* PhD thesis, Department of Electrical Engineering, University of California, Berkeley, California, 1992.

[18] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Transactions on Automatic Control*, vol. AC-38, no. 2, pp. 195–207, 1993.

[19] D. N. Godbole and J. Lygeros, "Longitudinal control of the lead car of a platoon," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 4, pp. 1125–1135, 1994.

[20] M. S. Branicky, "Stability of switched and hybrid systems," in *IEEE Control and Decision Conference*, pp. 3498–3503, 1994.

[21] D. N. Godbole, J. Lygeros, and S. S. Sastry, "Hierarchical hybrid control: an IVHS case study," in *IEEE Control and Decision Conference*, pp. 1592–1597, 1994.

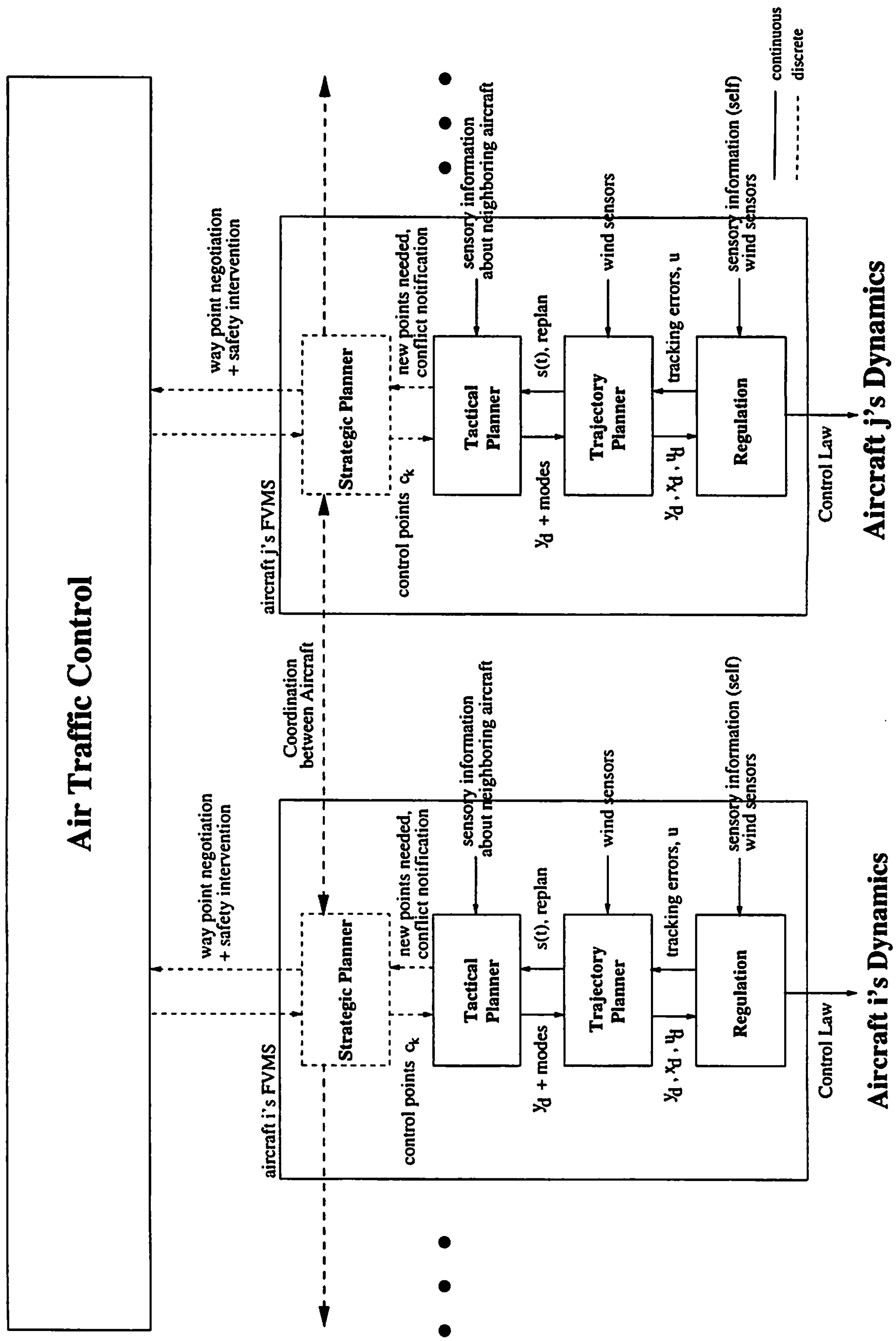

Figure 6: Proposed ATMS Architecture: Inside TRACON

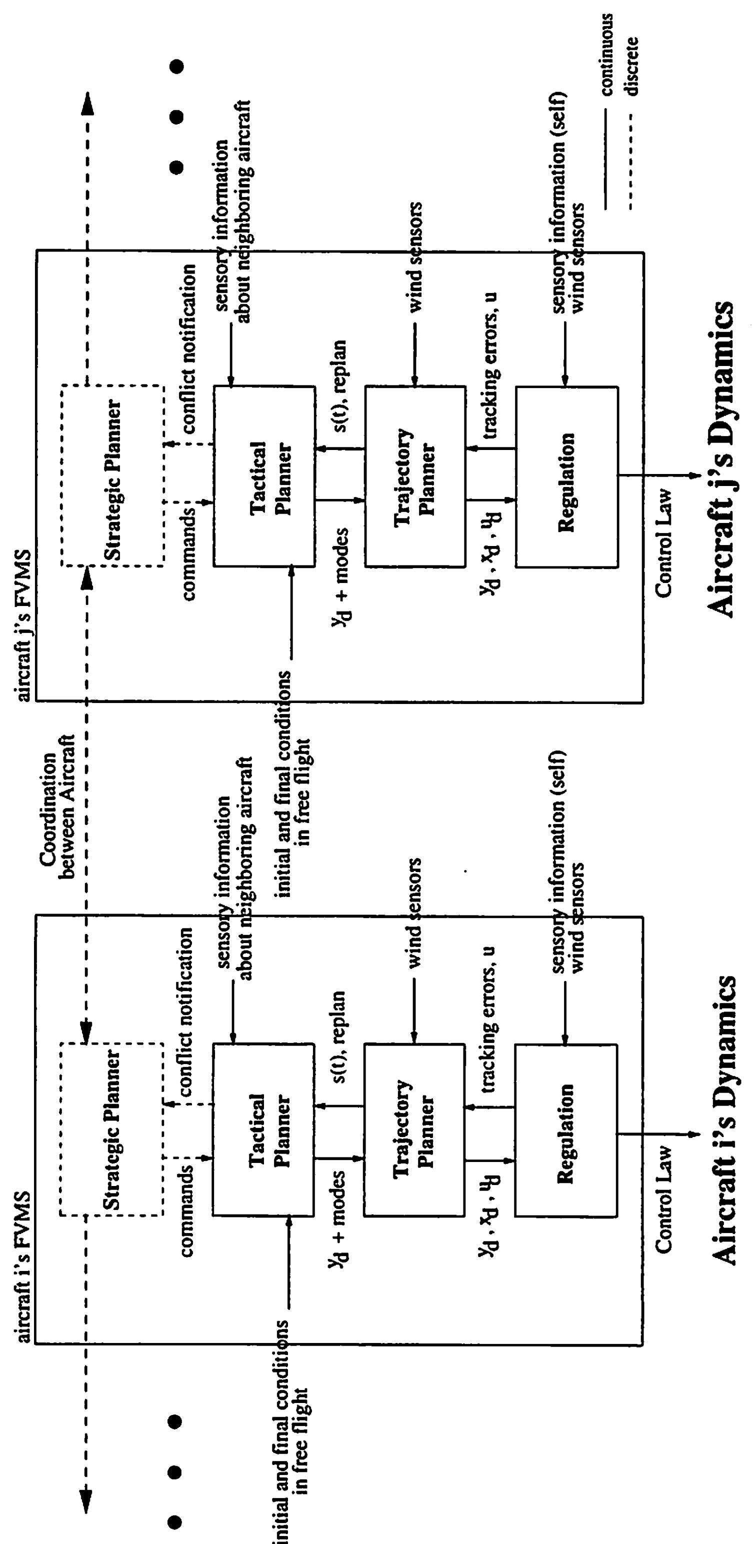

Figure 7: Proposed ATMS Architecture: Outside TRACON

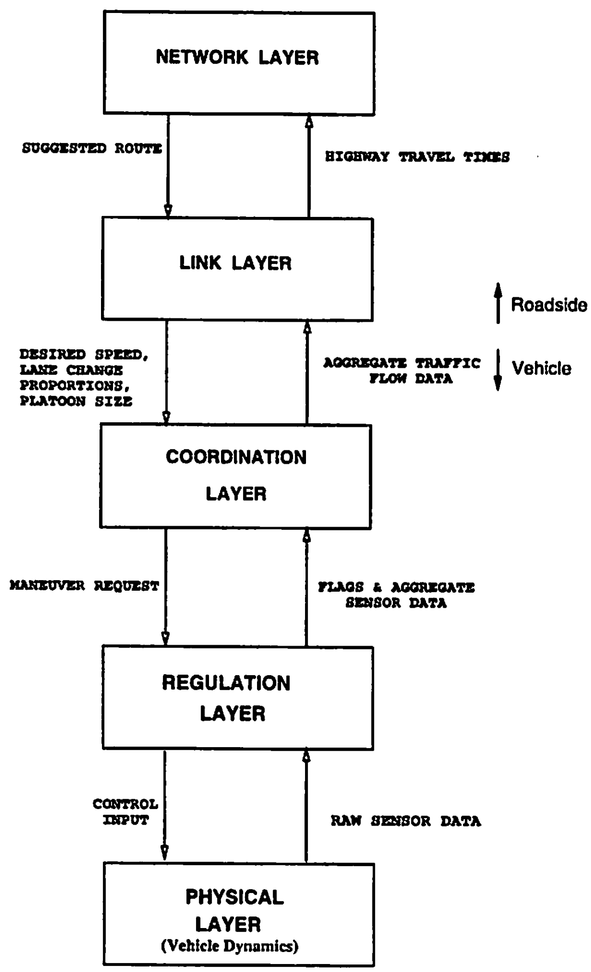

Figure 8: Architecture for PATH AHS control system

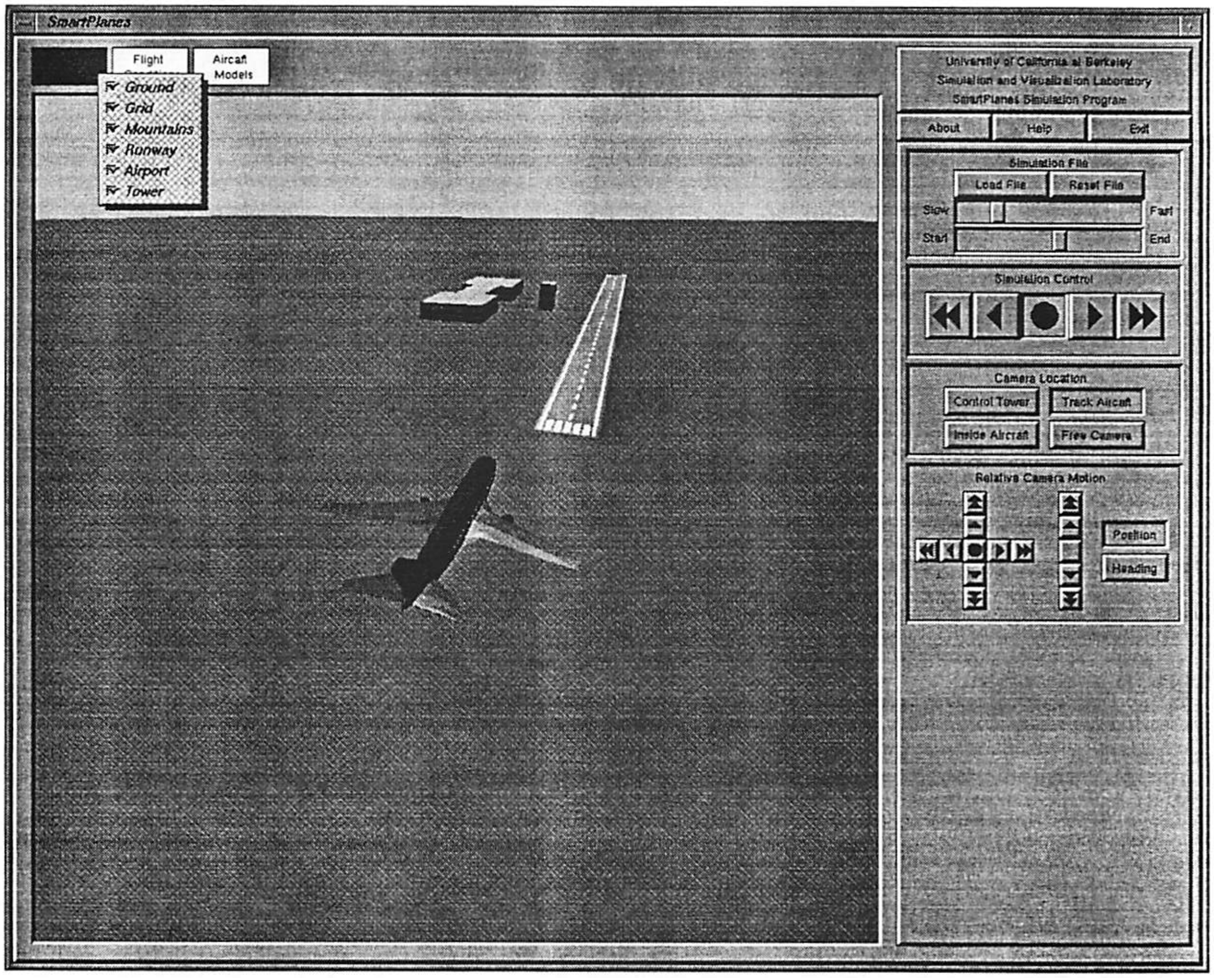

Figure 9: SmartPlanes Visualization Program