FORMAL ANALYSIS OF PARALLEL LANDING SCENARIOS

Víctor Carreño, Assessment Technology Branch

César Muñoz, Institute for Computer Applications in Science and Engenieering

NASA Langley Research Center, Hampton, Virginia

Abstract

The Airborne Information for Lateral Spacing (AILS) is a project being conducted at the NASA Langley Research Center. Its general objective is to reduce air traffic delays and increase airport efficiency by enabling approaches to closely spaced parallel runways under Instrument Meteorological Conditions. In this paper, we apply formal techniques to study a critical component of the AILS concept which provides situational awareness to the crew of the aircraft involved on a closely parallel landing. In particular, we focus on the AILS alerting algorithm. This algorithm analyses aircraft states and makes time projections of possible collision scenarios. Based on these projections and risk criteria, the algorithm triggers a sequence of caution and warning alerts. To show that the algorithm satisfies its requirements, we define a mathematical model of collision trajectories. The alerting algorithm is analyzed in the context of the trajectory model to determine if the algorithm complies with its requirements for all possible states and collision trajectories.

Introduction

The main objective of the Airborne Information for Lateral Spacing (AILS) project [2,5,10] is to reduce traffic delays and increase airport efficiency by enabling approaches to closely spaced parallel runways in Instrument Meteorological Conditions (IMC). Independent approaches to parallel runways are currently limited to 4300 feet in IMC. Specially equipped airports with fast scan radar, high resolution monitoring systems, and approach-specific air traffic controllers can perform parallel approaches to 3400 feet [7,12].

The AILS project aims at shifting the responsibility of maintaining separation during parallel approaches from the air traffic controller to the aircraft crew. Using the AILS concept, approaches to parallel runways 2500 feet apart in IMC are expected. AILS eliminates the delay inherent in the communication between air traffic controller and crew by displaying parallel traffic information in the cockpit. The degree of safety is enhanced by an alerting system that warns the crew when one of the aircraft involved in a parallel landing is deviating from the intended flight path. The alerting algorithm is a critical part of the AILS concept. Flaws in its logic could lead to non-alerted collision incidents. The algorithm has been extensively tested in simulators and in real flights.

The objective of this work is to conduct a formal analysis of the alerting algorithm in order to discover any possible errors that have not been detected during testing and simulation. We develop a formal model of parallel landing scenarios. Based on this model, we study the behavior of the AILS alerting algorithm with respect to collision incidents. In particular, we have found maximum and minimum times when an alarm will first sound prior to a collision. Indeed, we have proven that for any trajectory leading to a collision, an alarm is issued at least 4 seconds before the collision. Conversely, we have found that there exist trajectories leading to a collision where the alarm in the evader aircraft will not sound before 11 seconds. We believe that for all cases the largest time when an alarm will first sound prior to a collision is closer to 11 than to 4.

The paper is organized as follows. First, in section 2, we shortly review the alerting features which are integrated in the AILS concept. The alerting algorithm is then described in details in section 3. Our model of collision trajectories is studied in section 4. Section 5 contains the formal analysis that we have developed. We summarize our work in section 6.

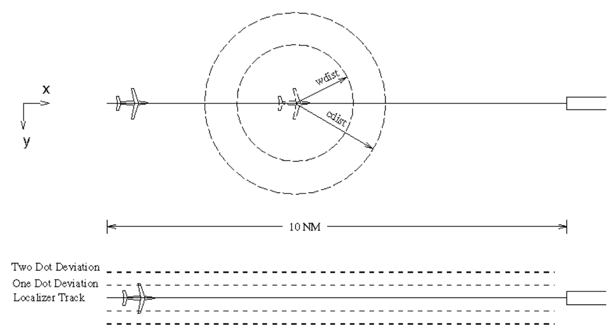

Figure 1. Parallel Runway Approach

System Description

In a typical independent parallel approach, aircraft intersect their localizer track (longitudinal runway center) approximately 10 nautical miles from the runway threshold (Figure 1). During localizer intersection, aircraft have a 1000 feet vertical separation. After the aircraft are established in their localizer track, vertical separation is eliminated and aircraft start a normal glide path for landing.

The AILS system starts operating when the aircraft are on their localizers. At this time the aircraft are approximately at the same altitude. An algorithm implementing the alerting features of the AILS concept runs independently on each aircraft twice every 0.5 seconds. The first time the algorithm assumes that the own-ship is the intruder aircraft and the adjacent aircraft is the evader. In the next iteration the algorithm assumes that the own-ship is the evader and the adjacent aircraft is the intruder. When the intruder aircraft deviates from its airspace, one of six kind of different alerts, depending on the severity of the deviation, is displayed in the evader or intruder aircraft primary and navigation displays. Alerts in the intruder aircraft should be followed by a corrective maneuver. The evader aircraft is not expected to perform an evasive maneuver until a warning alert is issued, at which time landing is aborted and an emergency escape maneuver is performed. The intruder aircraft always receives an alarm before the respective alarm is issued to the evader.

Several assumptions were made by the AILS project researchers in the development of the alerting algorithm. Physical characteristics and operational constraints justify these assumptions. They are as follows:

- Time is discrete and divided in increments of 0.5 seconds.
- The bank angle and ground speed determine the turn rate.
- The speeds of the aircraft are constant.
- The vertical separation between the aircraft is assumed to be zero during a landing approach.
- Only the intruder aircraft will deviate from its path in a parallel approach. The evader aircraft is assumed to stay in its localizer with a heading angle of zero degrees.

  It should be noted that the experimental AILS system, as currently designed, forms part of the Traffic Alert and Collision Avoidance System (TCAS) [11]. In this work, we assume that the AILS alerting algorithm is running in isolation from other aircraft components. We concentrate on the alerting kernel of the AILS alerting system.

The AILS Alerting Algorithm

The original AILS algorithm was written in FORTRAN at Langley Research Center. It has been revised several times. Honeywell provided the latest version flown in the Boeing 757 experimental aircraft. For the work presented in this paper, we created a high level abstract model of the alerting algorithm in the specification language of the general verification system PVS [9].

The alerting algorithm determines when an alarm will be triggered based on projections of the actual state of the aircraft. It compares possible future aircraft locations with predetermined time and distance thresholds. The state of an aircraft at time *t* is given by its coordinates $x(t), y(t)$, its heading $\psi(t)$, its bank angle $\phi(t)$, and its ground speed *v*[1]. The algorithm is executed in two modes every 0.5 seconds. First, it assumes its own aircraft is a threat to the adjacent aircraft and the adjacent aircraft is

[1] Notice that the ground speed is constant, i.e., it does not vary on time.

following the localizer; and second, it assumes the adjacent aircraft is a threat to its own and the own is following the localizer. In either mode, one aircraft is the intruder and one is the evader.

The algorithm considers two cases depending on whether the intruder is changing direction or not. When the intruder aircraft is not changing direction, i.e., its bank angle is zero, the algorithm determines if the two aircraft are diverging or converging and the point of closest separation. This is done by first obtaining the derivative of the distance between the aircraft and then solving for the time when the derivative equals zero. Figure 2 illustrates that calculation, where *(*$x_{in}(t)$*,*$y_{in}(t)$*,※*$_{in}(t)$*, ❖*$_{in}(t)$*,*v_{in}*)* and *(*$x_{ev}(t)$*,*$y_{ev}(t)$*,※*$_{ev}(t)$*,❖*$_{ev}(t)$*,*v_{ev}*)* are states of the intruder and evader aircraft at time *t*, respectively. Time ▼, relative to the current time *t,* gives the time of closest separation of the aircraft. If ▼ is negative the tracks are diverging, if ▼ is zero the tracks are parallel, otherwise, ▼ is greater than zero and the tracks are converging (Figures 3 and 4). The correctness of the derivation was checked on the computer algebra tool MuPAD [3].

$$\Delta_x(t) = x_{in}(t) - x_{ev}(t)$$
$$\Delta_y(t) = y_{in}(t) - y_{ev}(t)$$
$$\frac{d}{dt}\Delta_x(t) = v_{in}\cdot\cos(\theta_{in}(t)) - v_{ev}$$
$$\frac{d}{dt}\Delta_y(t) = v_{in}\cdot\sin(\theta_{in}(t))$$
$$R(t) = \sqrt{\Delta_x(t)^2 - \Delta_y(t)^2}$$
$$\frac{d}{dt}R(t) = \frac{\Delta_x(t)\cdot\frac{d}{dt}\Delta_x(t) + \Delta_y(t)\cdot\frac{d}{d}}{\sqrt{R(t)}}$$
$$\tau(t) = -\frac{\Delta_x(t)\cdot\frac{d}{dt}\Delta_x(t) + \Delta_y(t)\cdot\frac{d}{dt}}{\frac{d}{dt}\Delta_x(t)^2 + \frac{d}{dt}\Delta_y(t)^2}$$

Figure 2. Derivation of Closest Separation

When tracks are diverging or parallel, the algorithm checks the aircraft separation at the present time against the threshold distance for an alert. When tracks are converging, the algorithm compares the time and distance of closest separation against time and distance thresholds, respectively. In either case, an alarm is triggered when the calculated time and distance are within the time and distance alert thresholds.

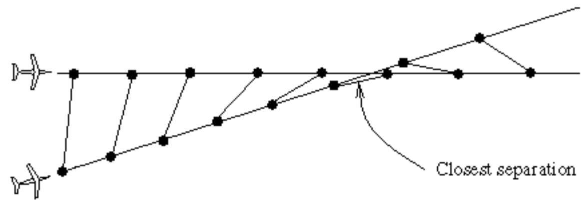

Figure 3. Distance at Closest Separation

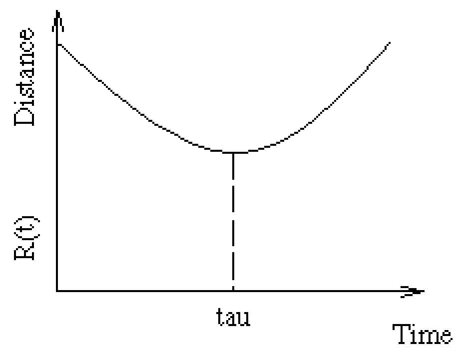

Figure 4. Time at Closest Separation

When the intruder aircraft is changing direction, i.e., its bank angle is not zero, the algorithm calculates the turn radius and the rate of change of direction. Tangential tracks are calculated from the arc path as to produce tangents that are 1.5 to 3 degrees in angular separation (Figure 5). For each of these tangential tracks the algorithm determines whether the two aircraft tracks are diverging or converging and performs time and distance comparisons as explained above.

Note that the AILS algorithm considers a limited set of possible trajectories for the intruder aircraft, i.e., assuming a constant radius turn at the original bank angle, only tangent track escapes to the turn arc are considered. This assumption is reasonable under normal circumstances, i.e., the intruder aircraft is not intentionally trying to collide with the evader aircraft. However, to evaluate the behavior of the algorithm in a wider range of possible landing scenarios, a more general model of trajectories for the intruder aircraft is necessary. In the next section, we develop such a model.

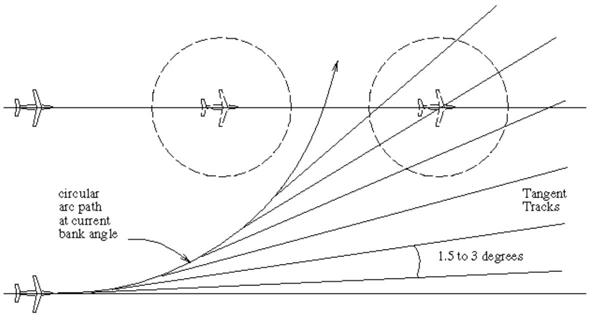

Figure 5. Tangential Tracks

Parallel Landing Scenarios

According to the characteristics and assumptions of the AILS algorithm, we propose a time-discrete model of trajectories with time increments of 0.5 seconds where the bank angle and ground speed of the intruder aircraft determine intrusion paths. Given a ground speed *v* greater than zero, a bank angle ❖, and the gravitational acceleration constant *g*, the heading turn rate is given by the formula

$$\text{trkrate}(v, \varphi) = \frac{\tan(\varphi) \cdot g \cdot 180}{v \cdot \pi}.$$

Although under normal operation the bank angle of a commercial aircraft is limited to -30 to 30 degrees, we allow the bank angle to range from -45 to 45 degrees. For a minimum ground speed of 180 feet per second, it means a maximum heading turn rate of about 6 degrees per second. These values produce very aggressive blundering situations quite consistent with worst cases scenarios tested by the AILS developing group.

Intruder Trajectories

An intruder trajectory is a sequence of aircraft states satisfying

$$x(t+\tfrac{1}{2}) = x(t) + \tfrac{1}{2} \cdot v \cdot \cos(\theta(t))$$

$$y(t+\tfrac{1}{2}) = y(t) + \tfrac{1}{2} \cdot v \cdot \sin(\theta(t))$$

$$\theta(t+\tfrac{1}{2}) = \theta(t) + \tfrac{1}{2} \cdot \text{trkrate}(v, \varphi(t))$$

Evader Trajectories

For the evader aircraft, we assume that it stays in its localizer with a constant speed and constant heading of zero degrees. Heading and bank angles are irrelevant in the definition of an evader trajectory. An evader trajectory is a sequence of aircraft states satisfying

$$x(t+\tfrac{1}{2}) = x(t) + \tfrac{1}{2} \cdot v$$

$$y(t+\tfrac{1}{2}) = y(t)$$

Collision Scenarios

We are interested in trajectories leading to collision incidents. Aircraft are said to be in collision if the distance between them is less than or equal to 200 feet, which is approximately the wing span of a Boeing 747.

We have implemented the model of trajectories, together with our high-level version of the alerting algorithm, in Java. The implementation serves a double purpose. First, it allows us to graphically visualize all the collision trajectories for a given time and initial values of the intruder and evader aircraft. Second and more importantly, by studying those trajectories, we were able to extract conjectures that we have then formally proven in PVS. Conversely, we have rejected some conjectures by finding counter-examples via simulation of collision trajectories. Figure 6 depicted collision scenarios, generated by the Java model, for an intruder aircraft located at (860,0) and an evader aircraft located at (0,2500). Ground speed for both aircraft is 250 feet per second. The initial heading of the intruder aircraft is 3 degrees. In this example, in order to reduce the amount of data, the time step is 4 seconds. The grid is set to 500 feet. Given those inputs, the dark line represents the collision trajectory that issues its first alarm closest to a collision point.

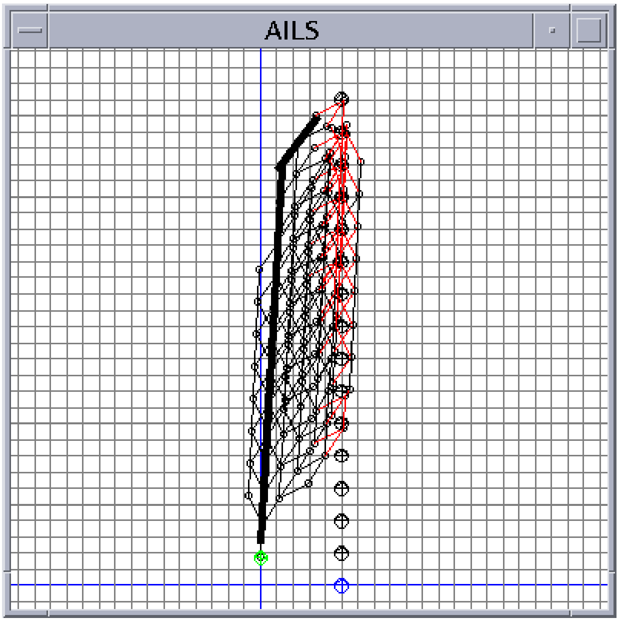

Figure 6. Trajectories Leading to Collision

In the next section, we formally study the behavior of the alerting algorithm with respect to our model of collision trajectories.

Formal Analysis

The objective of this modeling and verification work is (1) to show that the method implemented in the AILS alerting algorithm to predict trajectories and trigger alarms is adequate and does not lead to dangerous situations, and (2) to explore possible trajectory scenarios which lead to unacceptable risk. To this effect we created models of the algorithm and aircraft trajectories in PVS, created simulations in JAVA to graphically visualize the behavior and characteristics of the landing scenario, and derived in the computer algebra tool MuPAD equations for time and distance of closest approach.

Our intention is to show that for all aircraft trajectories leading to a collision and all initial states[2] an alarm is issued before a collision. In our formal development, we have found maximum and minimum times when an alarm will be first issued prior to a collision.

In first place, we have proven that an alarm is triggered when the distance between the aircraft is within the alerting range of 1400 feet. For an intruder ground speed of 240 feet per second this result in an alarm at least 4 seconds before the collision. This property holds independently of the values of any other state variables of the aircraft. An effort to prove that an alarm is issued in any case 19 seconds before a collision failed. Indeed, we have found a collision trajectory that allows two aircraft to fly from more than 2500 feet separation to a distance of less than 1900 feet, without triggering an alarm 11 seconds before the collision. The dark line in Figure 6 represents this trajectory. Therefore, we can state that (1) there is a trajectory for which an alarm will not sound before 11 seconds and (2) for all trajectories an alarm will sound at least 4 seconds before a collision. We believe that for all cases the largest time prior to a collision when the alarm will first sound is closer to 11 than to 4.

In order to find a largest time prior to a collision, we have to discover strong geometrical properties on collision trajectories. One of these properties states for example that any intruder aircraft out of the circle of center *(x,y)* and radius *200 + v* ✄ *t*, needs a larger time than *t* to collide with an evader aircraft located at *(x,y)*. We intend to use that property, together with some others derived from physical constraints, to find a bound greater than 4 seconds for any collision scenario. Under the assumption that the intruder bank angle is always zero, we have proven that an alarm is issued 19 seconds before a collision. We are trying to generalize the proof for an arbitrary trajectory and a time of 9 seconds.

Conclusion

Several case studies have been performed on the application of hybrid automata to the modeling of systems that include continuous and discrete domains. In particular, a simplified TCAS system was modeled in [8] using hybrid automata. That work focuses on establishing a hybrid model of the closed loop system formed by several aircraft flying under TCAS assumptions. Although it is claimed that the model is suitable for formal analysis, there is no explicit attempt to automate the proof process. On the other hand, state exploration techniques have been used to analyze the system requirement specification of TCAS II [6]; we refer for instance to [1,4]. These works focus on the reactive aspect of the whole system.

[2] At the initial state an aircraft is on its localizer.

In the work presented in this paper, we constructed a formal model of the kernel of an alerting algorithm and we studied its behavior with respect to a model of collision trajectories. In our analysis, we assumed that the alerting algorithm runs in isolation of the other components of the system. We defer the integration of the alerting algorithm with rest of the system, for example TCAS, for future research.

An abstract model of the algorithm and its properties were developed in the general verification system PVS. Differential equations, resulting from physical phenomena, were mechanically checked in the computer algebra tool MuPAD. Models of the algorithm and collision trajectories were implemented in Java. The implementation allowed us to graphically explore collision scenarios before performing rigorous attempts to prove properties.

Lower and upper bounds for a time when an alarm will be issued before a collision were found. Our immediate goal, in the verification of the AILS algorithm, is to prove certain facts about the characteristics of the aircraft trajectories. We hope that these facts allow us to prove the adequacy of the alerting algorithm for a time large enough to avoid any possible collision incident.

References

[1] Chan, W., R. Anderson, P. Beame, D. Notkin, 1998, Improving efficiency of symbolic model checking for state-based system requirements. Technical Report TR-98-01-03, University of Washington, Department of Computer Science and Engineering.

[2] Doyle, T., F. McGee, 1998, Air traffic and operational data on selected U.S. airports with parallel runways. Technical Report NASA/CR-1998-207675, NASA.

[3] Fuchssteiner, B., 1996, MuPAD User's Manual, John Wiley and Sons, Chichester, New York, first edition.

[4] Heimdahl, M.P.E, N.G. Leveson, 1995, Completeness and Consistency Analysis of State-Based Requirements, Proceedings of the 17th International Conference on Software Engineering, pp. 3-14.

[5] Koczo, S., 1996, Coordinated parallel runway approaches, Technical Report NASA-CR-201611, NASA.

[6] Leveson, N.G., M.P.E. Heimdahl, H. Hildreth, J.D. Reese, 1992, Requirements specification for process-control systems, Technical Report ICS-TR-92-106, University of California, Irvine, Department of Information and Computer Science.

[7] Lind, A.M., 1993, Two simulation studies of precision runway monitoring of independent approaches to closely spaced parallel runways, Technical Report AD-A263433 ATC-190 DOT/FAA/NR-92/9, NASA.

[8] Lygeros, J., N. A. Lynch, 1997, On the formal verification of the TCAS conflict resolution algorithms., In Proceedings 36th IEEE Conference on Decision and Control, pp. 1829-1834.

[9] Owre, S., J. M. Rushby, N. Shankar, 1992, PVS: A prototype verification system, Proceedings of the 11th International Conference on Automated Deduction (CADE), volume 607 of Lecture Notes in Artificial Intelligence, Springer Verlag, pp. 748-752.

[10] Rine, L., T. Abbott, G. Lohr, D. Elliott, M. Waller, R. Perry, 2000, The flight deck perspective of the NASA Langley AILS concept, Technical Report NASA/TM-2000-209841, NASA.

[11] RTCA, 1990, Minimum operational performance standards for traffic alert and collision avoidance system (TCAS) airborne equipment, consolidated edition, Guideline DO-185, Radio Technical Commission for Aeronautics, One McPherson Square, 1425 K Street N.W., Suite 500, Washington DC 20005.

[12] Wong, G., 1993, Development of precision runway monitor system for increasing capacity of parallel runway operations, AGARD, Machine Intelligence in Air Traffic Management, page 12.