Formally Verified Safe Vertical Maneuvers for Non-deterministic, Accelerating Aircraft Dynamics

Yanni Kouskoulas[1](✉), Daniel Genin[1], Aurora Schmidt[1],
and Jean-Baptiste Jeannin[2]

[1] The Johns Hopkins University Applied Physics Laboratory, Laurel, USA
yanni.Kouskoulas@jhuapl.edu
[2] Samsung Research America, Cambridge, USA

Abstract. We present the formally verified predicate and strategy used to independently evaluate the safety of the final version (Run 15) of the FAAs next-generation air-traffic collision avoidance system, ACAS X. This approach is a general one that can analyze simultaneous vertical and horizontal maneuvers issued by aircraft collision avoidance systems. The predicate is specialized to analyze sequences of vertical maneuvers, and in the horizontal dimension is modular, allowing it to be safely composed with separately analyzed horizontal dynamics. Unlike previous efforts, this approach enables analysis of aircraft that are turning, and accelerating non-deterministically. It can also analyze the safety of coordinated advisories, and encounters with more than two aircraft. We provide results on the safety evaluation of ACAS X coordinated collision avoidance on a subset of the system state space. This approach can also be used to establish the safety of vertical collision avoidance maneuvers for other systems with complex dynamics.

1 Introduction

As air travel increases and the airspace grows more crowded, existing air traffic management mechanisms such as altitude separation and manned air-traffic control are expected to experience significant stress. For decades, the Traffic Collision Avoidance System (TCAS) [3], first put into operation in the 1970s, has been the system of last resort, making mid-air collisions rare events. To address limitations that have been identified in TCAS, and to safely handle additional congestion and new participants expected in the future, the US Federal Aviation Administration (FAA), along with international partners, is developing a drop-in replacement, the next-generation Collision Avoidance System called ACAS X [9]. Like TCAS, ACAS X is intended to provide a final measure of safety, giving

This work was supported by the Federal Aviation Administration (FAA) Traffic-Alert & Collision Avoidance System (TCAS) Program Office (PO) AJM-233: Volpe National Transportation Systems Center Contract No. DTRT5715D30011.

M. Ayala-Rincón and C.A. Muñoz (Eds.): ITP 2017, LNCS 10499, pp. 336–353, 2017.
DOI: 10.1007/978-3-319-66107-0_22

advice that helps prevent mid-air collisions when all other preventive measures have failed.

In 2013, our group was designated as the independent verification and validation (V&V) team for ACAS X. We began developing an independent approach to formally verify the safety of the overall ACAS X system, either to establish guaranteed safety under certain operating conditions, or to identify different categories of problems and bring them to the attention of ACAS X developers and the FAA. This proved to be challenging for a number of reasons, including that ACAS X has very complicated behavior, and does not have a precisely stated set of requirements – informally, its goal is to provide an improvement over TCAS, both in terms of safety and alerting behavior. In addition, the system has an enormous state space – over 28×10^{12} state points – and complex logic based on the massive lookup table and complementary run-time components. This analysis, detailed in [6,7] has been so successful that we were able to find hundreds of millions of straight-line flight (i.e., the simplest possible) unsafe conditions in early versions of the system that were not identified by the standard simulation and testing approaches.

Our previous efforts were fundamentally limited to analyzing intruders that flew in a straight line, without any acceleration or maneuvering. The analysis also could not address the safety of an own-ship aircraft (our term for the aircraft in which the observer travels) that turns or makes any sort of horizontal maneuvers; previous analysis limited own-ship non-determinism to vertical motion.

The present work describes a new approach to vertical safety analysis that allows us to analyze the safety of encounters where both the intruder and own-ship are independently accelerating non-deterministically in the vertical and horizontal directions. To do this, we create a vertical safety predicate that relaxes the assumption of constant, relative horizontal velocity that in our previous work restricted us from analyzing horizontal acceleration in maneuvers such as turns. Our predicate has parameters that describe horizontal safety, but is not limited to any particular horizontal dynamics; it can be composed with any horizontal motion that has been correctly analyzed. With the development of appropriate analysis for different horizontal dynamics, this approach could also assess the safety of non-deterministic, accelerating horizontal and vertical dog-fight-like maneuvers.

The main contribution of this paper is in providing a predicate to analyze the safety of vertical advisories during turns and in the presence of non-deterministically accelerating intruders. All the theorems in this paper and safety predicates for vertical motion are formally verified, meaning that their correctness is ensured via a machine-checked mathematical proof.[1]

The rest of the paper is organized as follows: Sects. 2 and 3 provide an overview of how to use the predicate by analyzing safety for an example encounter, describing the parameterization of pilot behavior and horizontal dynamics; Sect. 4 provides a detailed description of the development of vertical

[1] Proofs can be viewed and downloaded at https://bitbucket.org/ykouskoulas/vert_safety_proofs/src/.

safety predicates; Sect. 5 discusses issues related to formalizing our guarantees; Sects. 6 and 7 describes how we extend our safety proofs to a real system, and our results; and Sects. 8 and 9 describe related work, and conclude.

2 Overview

This section presents an overview of the logic of our approach, starting with its basic properties and walking through an illustrative example of how it would be used in practice.

Safety Property. The logic of this approach comes from the definition of *safety* used in this analysis; it allows us to decompose the safety analysis into two steps that we can treat seperately: a horizontal problem, and a vertical problem.

For this work, safety between two aircraft means that one aircraft doesn't come within a certain vertically oriented cylinder with radius r_p and half-height h_p centered on the other aircraft. This definition includes exact collision as well as any dangerously close approach between two aircraft, and is referred to by the aviation community as a Near Mid-Air Collision (NMAC). We call this volume the NMAC puck due to the resemblance with a hockey puck. Aircraft trajectories have uncertainty associated with them, and the puck represents the volume in which the other aircraft location might be found. Entering it represents, in the worst case, an actual collision.

We define *horizontal conflict* as the condition where the horizontal projections of the two aircraft come within the horizontal bounds of a puck centered on one of them; *vertical conflict* is when their vertical projections come within the vertical bounds of a puck, also centered on one of them. The two aircraft have an NMAC only if they are in horizontal and vertical conflict simultaneously.

To formalize our safety property, we define $J(t) = J_x(t)\hat{x} + J_y(t)\hat{y} + J_z(t)\hat{z}$ to be the trajectory of the ownship, and $K(t) = K_x(t)\hat{x} + K_y(t)\hat{y} + K_z(t)\hat{z}$ to be the trajectory of the intruder, both in a Cartesian coordinate system with x, y and z axes aligned to east, north and up, respectively. We have horizontal conflict whenever

$$C_h(t) \equiv |((J_x(t) - K_x(t))\,\hat{x} + (J_y(t) - K_y(t))\,\hat{y}| \leq r_p \tag{1}$$

is true. We have vertical conflict when

$$C_v(t) \equiv |J_z(t) - K_z(t)| \leq h_p \tag{2}$$

is true. An NMAC occurs at time t only when:

$$C_h(t) \wedge C_v(t) \tag{3}$$

We will first analyze the horizontal dynamics to determine the timing of the encounter, i.e. when the aircraft come together. We call this timing a parameterization of horizontal safety, because it establishes safety within a series of time intervals. Subsequently, the safe-by-design logic can be used to establish safety for a sequence of independent, non-deterministic, vertical maneuvers made by the pilot of each aircraft outside of these intervals. For each safety evaluation, we must choose a sequence and timing of vertical maneuvers for each aircraft, and it is under these assumptions that we can establish safety or the possibility of collision. The following paragraphs go through these steps to apply safety analysis for a specific example.

Parameterizing Horizontal Safety. To parameterize horizontal safety, we must analyze the horizontal motion of the two aircraft and identify time intervals in which the probability of the aircrafts' horizontal projections coming into proximity (i.e., horizontal conflict as defined in Eq. 1) is non-zero. Through this horizontal parameterization, we establish safety outside these intervals, because when the aircraft are far away from each other $C_h(t)$ is false, and Eq. 3 cannot be satisfied – there is no possibility of immediate collision.

We index each time interval of possible horizontal conflict using index i, and define t_{ei} and t_{xi} as times of earliest entry into and latest exit from conditions where horizontal conflict is possible, for interval i. This defines a set of time intervals, $V_i = [t_{ei}, t_{xi}]$, and their union $V = \bigcup_{i\in\{1...n\}} V_i$, during which safety must be established through the absence of vertical conflict.

Consider the example of two aircraft whose horizontal trajectories follow deterministic circular paths, as shown in Fig. 1, where the speed of the own-ship is chosen by the pilot. To simplify our example, we assume that the speed of the

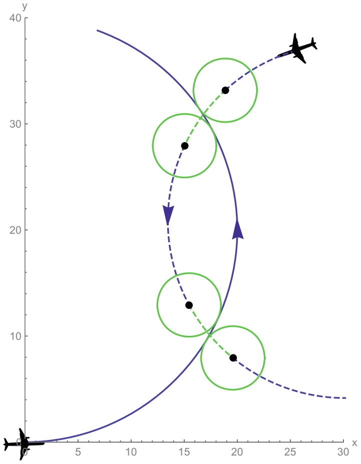

Fig. 1. Example horizontal turning trajectories, projected onto horizontal cartesian coordinate system, viewed from above. The own-ship trajectory is represented by a solid line, and the intruder is represented by a dashed line. Circles and green color indicates the extent of trajectory segments where collision is possible. (Color figure online)

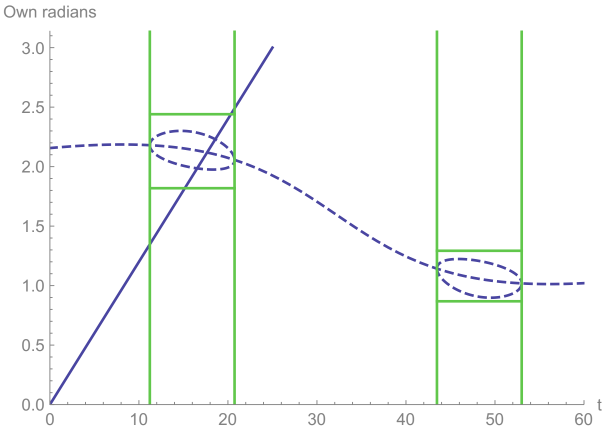

Fig. 2. Analysis of encounter timing for one possible combination of ground speeds. Positions are given as a radian measure on the own-ship's trajectory circle, and ground speeds are assumed constant for this particular scenario. The dashed line is the intruder's center projected on the own-ship's trajectory, and when it intersects the trajectory, the extent of that intersection is plotted above and below the center. Vertical lines correspond to the beginning and end of time intervals when a collision is possible, and to the disks in Fig. 1.

intruder is known, although this is not required for the analysis in general. The solid line represents the own-ship while the dashed line represents the intruder aircraft. One way to visualize when a collision is possible is to imagine a disk representing the top of the NMAC puck traveling along one of the trajectories. When that disk intersects the other trajectory, a collision is possible, depending on the relative speeds of the aircraft. Here we show the disk on the intruder's trajectory at the four points where it touches the own-ship's trajectory, and highlight the parts of its trajectory where a collision is possible. Figure 2 illustrates the timing analysis that is necessary to compute the horizontal conflict interval. Assuming the intruder's ground speed is known and consistent with Fig. 2, the horizontal conflict intervals for this geometry can be read off the plot to determine that $V = [11.2\,\text{s},\ 20.7\,\text{s}] \cup [43.5\,\text{s},\ 53.0\,\text{s}]$.

Our analysis is not limited to these horizontal dynamics; we can also establish safety for more complex horizontal motion and other types of non-determinism, as long as we can compute V.

Sequence and Timing of Vertical Maneuvers. To match common flight patterns and the ACAS X advisory system, the vertical dynamics of each aircraft is modeled by a sequence of non-deterministic maneuvers, specified by allowed acceleration and velocity ranges. By combining maneuvers it is possible to represent a variety of behaviors, including straight line flight, choice of one of a series of actions (where the decision is unknown at the time of safety analysis), unrestricted vertical flight, compliance with an ACAS X vertical advisory, delayed

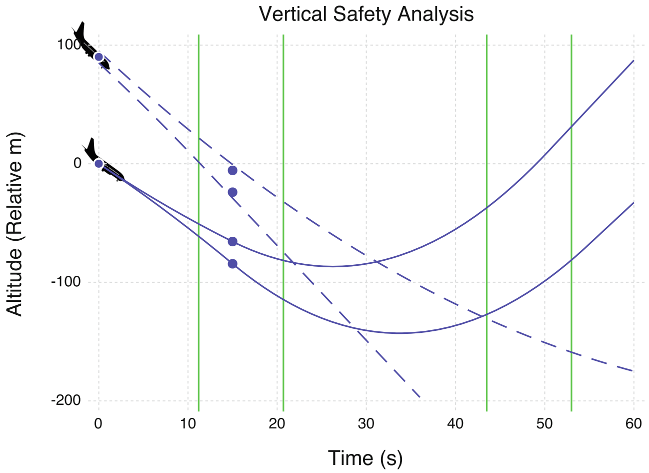

Fig. 3. Bounding envelopes for vertical motion of ownship and intruder (dashed). Horizontal conflict intervals are indicated by vertical lines. Safety is guaranteed despite any maneuvers the pilots may make that cause variations in vertical acceleration and velocity, or variations in horizontal ground speed of the ownship, within assumed dynamics.

compliance with an ACAS X advisory, a reversal of vertical motion direction to ensure safety, or straight line flight followed by a level-off maneuver. Thus, the proposed dynamics captures many, if not most, operationally relevant aircraft encounter scenarios.

For our example from Fig. 1, assume the intruder starts above the own-ship, the aircraft are descending, with the intruder diving towards the ground. The own-ship engages in a vertical chase for the first 15 s of the encounter, diving at a less extreme rate, and then follows advice to sharply accelerate upwards, eventually crossing altitudes with the intruder.

Vertical Safety Predicate. Once we have analyzed horizontal dynamics, and chosen a pilot model (i.e. a sequence of vertical pilot timing and actions) we can apply the vertical safety predicate to establish whether we can definitively avoid collision under our assumptions. Figure 3 illustrates the extent of vertical motion in our example scenario from Fig. 1 by plotting the most extreme vertical trajectories of the own-ship and intruder. These boundaries describe a reachable envelope of altitudes for each moment in time. Our predicate Ψ, described in Sect. 4, guarantees safety for this geometry under our assumptions, and the figure illustrates the intuition behind the predicate's logic, confirming that the aircraft are safely separated vertically during both horizontal conflict intervals.

This envelope introduces non-determinism in our model, the representation of uncertain vertical motion in the future. Even though the limiting trajectories of our envelope are piecewise polynomial and our dynamics are simple, our dynamics are not limited to piecewise polynomial trajectories. This model allows us to represent a continuous family of irregular trajectories within our acceleration limits, all of which travel within the envelope but which include many different types of motion.

While the ownship's upper and lower limiting trajectories issue from a single point at time zero, the intruder's upper and lower limiting trajectories bound a range of altitudes, indicating the uncertainty in the intruder's vertical position, e.g., due to surveillance error.

Time intervals for this plot are subdivided so that each time interval contains a single maneuver for each aircraft. Although, for a generic sequence of maneuvers, time intervals corresponding to individual maneuvers for the ownship and intruder will not agree, we can always subdivide maneuvers as necessary to ensure that exactly one maneuver covers the full duration of the interval for both ownship and intruder. This is possible because a single maneuver of duration d and a sequence of identical (with regard to velocity and acceleration bounds) maneuvers with durations $d_0, d_1, d_2, \ldots, d_n$, such that $\sum_{i=1}^{n} d_i = d$ encompass exactly the same set of aircraft trajectories.

3 Modeling and Assumptions

Modeling Non-deterministic Vertical Maneuvers. Each vertical maneuver is defined by a duration of time d the maneuver is in effect, and a range of vertical velocities, $[v_{\min}, v_{\max}]$. During the maneuver, the pilot accelerates the aircraft with the intention of bringing vertical velocity into the specified range. Acceleration is non-deterministic, and each maneuver has a set of four limiting vertical accelerations $a_{\min} \leq a_a < 0 < a_b \leq a_{\max}$. The subscripts a and b indicate the maximum and minimum acceleration allowed when the aircraft is above and below the target range of vertical velocities, respectively. During a maneuver, the pilot can choose to follow any acceleration $a(t)$, that is continuous, integrable, and satisfies

$$\begin{aligned} \forall t, (v(t) > v_{\max} \rightarrow a_{\min} \leq a(t) \leq a_a)\wedge \\ (v(t) = v_{\max} \rightarrow a_{\min} \leq a(t) \leq 0)\wedge \\ (v_{\min} < v(t) < v_{\max} \rightarrow a_{\min} \leq a(t) \leq a_{\max})\wedge \\ (v(t) = v_{\min} \rightarrow 0 \leq a(t) \leq a_{\max})\wedge \\ (v(t) < v_{\min} \rightarrow a_b \leq a(t) \leq a_{\max}) \end{aligned} \tag{4}$$

where $v(t) = \int_0^t a(t)\mathrm{d}t + v(0)$ is the velocity of the aircraft.

In the Coq formalization, we prove the following properties about pilot behavior:

Theorem 1 (Pilot-model vertical compliance). *The constraints on $a(t)$ given in Eq. 4 ensure that when the aircraft is below (above) the target range of vertical velocities, it will accelerate towards it with acceleration a_b (a_a) until it is within its bounds.*

Theorem 2 (Pilot-model maintains vertical compliance). *The constraints on $a(t)$ given in Eq. 4 ensure that once the aircraft has entered the allowed range of vertical velocities, the aircraft will stay within that range.*

There are sequences of maneuvers and certain geometries where it is impossible for a pilot to follow Eq. 4 while maintaining continuous acceleration. For example, compliance with one maneuver may require the pilot to increase vertical velocity by maintaining a positive acceleration, which may abruptly change to a requirement to decrease vertical velocity by maintaining negative acceleration, at the beginning of the next maneuver. In this case there will be trajectories with $a(t)$ satisfying the requirements of the first maneuver that will have no continuous extension to the second maneuver.

In order to ensure that any sequence of maneuvers individually satisfying Eq. 4 can be followed while maintaining acceleration $a(t)$ that is continuous (i.e. has a derivative) everywhere, we introduce the concept of an *auxiliary maneuver* for every pair of consecutive maneuvers. The auxiliary maneuver provides a finite time window to allow acceleration to transition continuously from one maneuver to the next, thus avoiding potential discontinuous changes in acceleration at the boundary between maneuvers. This simple device dramatically simplifies analysis by removing the need for additional restrictions that would otherwise be necessary to enforce the global continuity of $a(t)$.

Given a pair of maneuvers with target vertical velocity intervals $[v_{\min}, v_{\max}]$ and $[w_{\min}, w_{\max}]$, and acceleration bounds $a_{\min} \leq a_a < 0 < a_b \leq a_{\max}$ and $b_{\min} \leq b_a < 0 < b_b \leq b_{\max}$, respectively, the matching auxiliary maneuver is given by a target velocity interval $[\min(v_{\min}, w_{\min}), \max(v_{\max}, w_{\max})]$ and the acceleration bounds are $\min(a_{\min}, b_{\min}) \leq \max(a_a, b_a) < 0 < \min(a_b, b_b) \leq \max(a_{\max}, b_{\max})$. The minimal duration of an auxiliary maneuver is bounded below only by the limits on the derivative of the aircraft's acceleration, sometimes also referred to as jerk.

To simplify the formal safety proofs, we have chosen to assume that $a(t)$ is continuous – a natural assumption from the point of view of physics – and treat auxiliary maneuvers as undistinguished from other maneuvers. The alternative would be to have done the safety proofs for $a(t)$ that would be allowed to become discontinuous at the beginning of each maneuver. However, we did not pursue this approach since it is simultaneously less realistic and more difficult to implement in Coq.

4 Vertical Safety Predicates

In this section, we develop formally-verified, quantifier-free predicates establishing pairwise safety between two aircraft. We do this for arbitrary sequences of vertical maneuvers, where both pilots are accelerating non-deterministically. The predicates are also constructed in a modular fashion so they can be composed with a separate analysis of horizontal motion to ensure overall safety of an encounter.

Vertical Safety Predicates. To guarantee vertical separation between two aircraft, we establish a bounding envelope that contains all altitudes reachable by each aircraft for each sequential maneuver as a function of time. We then construct a predicate that computes a bounding envelope for each aircraft separately according to the initial position of each, and ensures that the envelopes don't overlap during V, the vertical conflict intervals. We establish the safety of this predicate via formal proofs in Coq.

The bounding envelope for a single aircraft executing a single maneuver (Eq. 4) depends on the initial range of vertical positions and velocities of the aircraft at the start of the maneuver as well as the maneuver velocity and acceleration bounds. In the time-altitude domain, edges of the bounding envelope are given by the upper and lower limiting trajectories. These trajectories originate from the extremes of the initial velocity and position ranges, and follow the extreme values of acceleration and velocity allowed by the maneuver. Specifically, limiting trajectories have the following form

$$J_z(t) = \begin{cases} \left(\frac{a}{2}t^2 + v_0 t + z_0\right)\hat{z} & \text{if } 0 \leq t < t_r \\ \left(v_t t - t_r \frac{(v_t - v_0)}{2} + z_0\right)\hat{z} & \text{if } t_r \leq t \end{cases} \tag{5}$$

where v_0 and z_0 are the initial vertical velocity and position of the aircraft, v_t is the matching extreme of the velocity range of the maneuver, and $t_r = \frac{v_t - v_0}{a}$ is the time when the limiting trajectory reaches the maneuver velocity range. So we have

$$(v_t, a) = \begin{cases} (v_{\max}, a_a) & \text{if } v_0 > v_{\max} \\ (v_{\max}, a_{\max}) & \text{if } v_0 \leq v_{\max} \end{cases} \tag{6}$$

for the upper limiting trajectory and

$$(v_t, a) = \begin{cases} (v_{\min}, a_{\min}) & \text{if } v_0 > v_{\min} \\ (v_{\min}, a_b) & \text{if } v_0 \leq v_{\min} \end{cases} \tag{7}$$

for the lower limiting trajectory. In the Coq formalization, we prove

Theorem 3. *An aircraft following an arbitrary trajectory satisfying the constraints of Eq. 4 remains within the altitude envelope bounded above and below by limiting trajectories determined by Eqs. 5, 6 and 7.*

Once upper and lower limiting trajectories are constructed we have an envelope of altitudes over time reachable by a non-deterministically maneuvering aircraft, with boundaries that are described piecewise by polynomials of at most degree two.

So far, we have been describing the dynamics for one aircraft, but we can use this model for each aircraft in the encounter, plotting reachable envelopes vs. time, and allowing us to visualize the uncertainty in position and relationship between aircrafts at each moment. Figure 3 provides a visual example of upper and lower limiting trajectories for ownship (solid lines) and intruder (dashed lines) aircraft.

To develop quantifier-free predicates that indicate the absence of vertical conflict for a pair of aircraft, we take the difference of their opposite limiting trajectories (lower-upper and upper-lower), and then compute whether the resulting polynomial is positive. Physically, this means the aircraft are safely separated. We first define the predicate

$$
\begin{aligned}
\mathit{\Gamma}((A,B,C),t_b,t_e) \equiv t_b \le t_e \rightarrow \\
(A > 0 \wedge ((0 \le D \wedge (R_1 > t_e \vee R_2 < t_b)) \vee D < 0) \vee \\
A < 0 \wedge (0 < D \wedge R_2 < t_b \wedge R_1 > t_e) \vee \\
A = 0 \wedge (B > 0 \wedge -C/B < t_b \vee B < 0 \wedge -C/B > t_e \vee \\
B = 0 \wedge C > 0))
\end{aligned}
\tag{8}
$$

to compute whether an arbitrary polynomial $At^2 + Bt + C$ represented by the vector of its coefficients (A, B, C) is positive over the interval $[t_b, t_e]$, where the subscripts b and e represent the beginning and ending times of the interval. In this predicate, we define $D \equiv B^2 - 4AC$, $R_1 \equiv \frac{(-B-\sqrt{D})}{2A}$, and $R_2 \equiv \frac{(-B+\sqrt{D})}{2A}$ – the expressions for the discriminant and roots of a quadratic. The predicate is made of a disjunction of three clauses, which analyze the polynomial when second order coefficient A is positive, zero, or negative. If A is non-zero there are two cases corresponding to an upward, $A > 0$, or downward, $A < 0$, extending parabola with at most two roots. If $A = 0$ the polynomial is linear with at most one root. The rest of the logic compares the location of the roots with the end points of the time interval $[t_b, t_e]$ and determines whether the curve is positive in that interval. We formalize and prove the following theorem in Coq:

Theorem 4 (Safely separated second-order polynomial interval). *The predicate $\mathit{\Gamma}((A,B,C),t_b,t_e)$ computes whether a polynomial $At^2 + Bt + C$ is positive over the interval $[t_b, t_e]$.*

Each limiting trajectory within each maneuver is a piecewise function composed of at most two pieces: a quadratic piece, corresponding to the aircraft accelerating toward the maneuver's target velocity range, and a linear piece, corresponding to the aircraft maintaining one of the extremal velocities in the maneuver's target velocity range. Either of these pieces could be missing depending on the state of the aircraft at the beginning of the maneuver and the maneuver's duration. We next define a predicate

$$
\begin{aligned}
\mathit{\Phi}(Q_1, L_1, t_{t1}, Q_2, L_2, t_{t2}, t_b, t_e) \equiv \\
\mathit{\Gamma}(Q_1 - Q_2 - P, \max(t_b, 0), \min(t_e, t_{t1}, t_{t2})) \wedge \\
\mathit{\Gamma}(L_1 - L_2 - P, \max(t_b, t_{t1}, t_{t2}), t_e) \wedge \\
(t_{t1} > t_{t2} \rightarrow \\
\mathit{\Gamma}(Q_1 - L_2 - P, \max(t_b, \min(t_{t1}, t_{t2})), \min(t_e, \max(t_{t1}, t_{t2})))) \wedge \\
(t_{t1} < t_{t2} \rightarrow \\
\mathit{\Gamma}(L_1 - Q_2 - P, \max(t_b, \min(t_{t1}, t_{t2})), \min(t_e, \max(t_{t1}, t_{t2}))))
\end{aligned}
\tag{9}
$$

to compute whether two limiting trajectories described by Q_1, L_1, and Q_2, L_2 are safely separated in interval $[t_b, t_e]$. In this predicate, $P = (0, 0, h_p)$ and h_p is the half-height of the NMAC puck. Each Q_i and L_i is a 3-vector containing the coefficients of the polynomials corresponding to the quadratic and linear pieces of trajectory i, respectively. The times t_{t1} and t_{t2} are the times when each respective trajectory transitions from one piece to the next. The predicate $\mathit{\Phi}$ computes the separation and determines whether it is adequate, (i.e. $> h_p$) for all points in the interval of interest, ensuring that the correct polynomial is used for each trajectory at each point. Given that each limiting trajectory is composed of at most two pieces, there are four possible combinations of polynomials that appear in the analysis: (Q_1, Q_2), (Q_1, L_2), (L_1, Q_2), (L_1, L_2). Each of these possibilities corresponds to one term of the conjunction in the definition of Φ. The predicate Φ has four instances of Γ, since it establishes safety for the different pieces (linear and quadratic) of a trajectory for an entire maneuver.

We formalize, and prove the following theorem in Coq:

Theorem 5 (Safely separated trajectory interval, above). *The predicate*

$$\Phi((\alpha_1, \beta_1, \gamma_1), (\delta_1, \epsilon_1, \zeta_1), t_{t1}, (\alpha_2, \beta_2, \gamma_2), (\delta_2, \epsilon_2, \zeta_2), t_{t2}, t_e, t_x) \tag{10}$$

computes whether a trajectory

$$T_1(t) = \begin{cases} \left(\alpha_1 t^2 + \beta_1 t + \gamma_1\right) & 0 \leq t < t_{t1} \\ \left(\delta_1 t^2 + \epsilon_1 t + \zeta_1\right) & t_{t1} \leq t \end{cases} \tag{11}$$

is safely separated and above trajectory

$$T_2(t) = \begin{cases} \left(\alpha_2 t^2 + \beta_2 t + \gamma_2\right) & 0 \leq t < t_{t2} \\ \left(\delta_2 t^2 + \epsilon_2 t + \zeta_2\right) & t_{t2} \leq t \end{cases} \tag{12}$$

by a distance of h_p over the interval $[t_b, t_e]$.

Consider an aircraft executing a sequence of m maneuvers, defined by minimum and maximum velocity bounds $([v_{\min 1}, v_{\max 1}], [v_{\min 2}, v_{\max 2}], \ldots, [v_{\min m}, v_{\max m}])$, for durations $(d_1, d_2, \ldots, d_m)$, each maneuver having an envelope of possible trajectories bounded by Eq. 5. We define $\{t_{mi}\}$ as the set of times that identify the start of each maneuver. We also assume the aircraft have horizontal dynamics for which there are n time intervals $([t_{e1}, t_{x1}], [t_{e2}, t_{x2}], \ldots, [t_{en}, t_{xn}])$ when the probability of horizontal conflict is non-zero. For convenience, we compute a set of times $(\tau_{mn}, \upsilon_{mn})$ that are the entry and exit times for conflict interval n, intersecting maneuver m, relative to the starting time of the maneuver:

$$(\tau_{mn}, \upsilon_{mn}) = \begin{cases} (\max(0, t_{en}), \min(d_1, t_{xn})) & \text{for } m = 1 \\ \left(\max(0, t_{en}) - \sum_{i=1}^{m-1} d_i, \min(d_m, t_{xn} - \sum_{i=1}^{m-1} d_i)\right) & \\ & \text{for } m > 1 \end{cases} \tag{13}$$

For each aircraft there is an upper and lower bounding trajectory; each of these bounding trajectories has a quadratic and a linear piece for each maneuver. We define Q and L to be 3-dimensional vectors representing the quadratic and linear parts of the bounding trajectory for a single maneuver and a single aircraft, and the time t_r to indicate when each limiting trajectory transitions between the quadratic and linear pieces. Each of these quantities uses a superscript with a tag to represent which aircraft (own or intruder), an up or down arrow indicating whether the bound is a trajectory that bounds the aircraft from above or below, respectively. Each variable also has a subscript index i that identifies the maneuver it describes.

So collectively, $Q_i^{\text{Own}\uparrow}$, $L_i^{\text{Own}\uparrow}$, and $t_{ri}^{\text{Own}\uparrow}$ represent the upper limiting trajectory for the ownship for maneuver i, and $Q_i^{\text{Own}\downarrow}$, $L_i^{\text{Own}\downarrow}$, and $t_{ri}^{\text{Own}\downarrow}$ to describe the lower limiting trajectory for the ownship in the same way. These vectors contain the second, first, and zeroth order coefficients from Eq. 5. So

$$\begin{array}{ll} Q_i^{\text{Own}\uparrow} \equiv \left(\frac{a}{2}, v_0, z_0\right) & Q_i^{\text{Own}\downarrow} \equiv \left(\frac{a}{2}, v_0, z_0\right) \\ L_i^{\text{Own}\uparrow} \equiv \left(0, v_{\text{max}i}, z_0 - \frac{(v_{\text{max}i}-v_0)^2}{2a}\right) & L_i^{\text{Own}\downarrow} \equiv \left(0, v_{\text{min}i}, z_0 - \frac{(v_{\text{min}i}-v_0)^2}{2a}\right) \\ t_{ri}^{\text{Own}\uparrow} \equiv \frac{v_{\text{max}i}-v_0}{a} & t_{ri}^{\text{Own}\downarrow} \equiv \frac{v_{\text{min}i}-v_0}{a} \end{array} \quad (14)$$

represents upper and lower bounding trajectories for the own-ship. The initial conditions v_0 and z_0 are set so that velocity and position are continuous at the boundary between the different maneuvers, and a is set according to Eqs. 6 and 7.

Similarly, we define $Q_i^{\text{Int}\uparrow}$, $L_i^{\text{Int}\uparrow}$, $t_i^{\text{Int}\uparrow}$, $Q_i^{\text{Int}\downarrow}$, $L_i^{\text{Int}\downarrow}$, and $t_{ri}^{\text{Int}\downarrow}$ to describe the upper and lower limiting trajectories of the intruder aircraft, replacing parameters with the ones appropriate for that aircraft.

Finally, we define the predicate

$$\begin{aligned} \Psi = \bigwedge_{j\in\{1,\dots,n\}} \Bigg(\Bigg(& \bigwedge_{i\in\{1,\dots,m\}} \Phi(Q_i^{\text{Own}\downarrow}, L_i^{\text{Own}\downarrow}, t_{ri}^{\text{Own}\downarrow}, Q_i^{\text{Int}\uparrow}, L_i^{\text{Int}\uparrow}, t_{ri}^{\text{Int}\uparrow}, \tau_{ij}, \upsilon_{ij}) \Bigg) \vee \\ \Bigg(& \bigwedge_{i\in\{1,\dots,m\}} \Phi(Q_i^{\text{Int}\downarrow}, L_i^{\text{Int}\downarrow}, t_{ri}^{\text{Int}\downarrow}, Q_i^{\text{Own}\uparrow}, L_i^{\text{Own}\uparrow}, t_{ri}^{\text{Own}\uparrow}, \tau_{ij}, \upsilon_{ij}) \Bigg) \Bigg) \end{aligned} \quad (15)$$

that helps establish safety between aircraft during a series of horizontal conflict intervals, as they follow a series of maneuvers. Its construction mirrors the following logic. An encounter is safe if each of its horizontal conflict intervals is safe; the outer conjunction over j ensures safety for each interval. Each conflict interval is safe if either the own-ship is always safely above the intruder, or vice versa; the left side and right side of the disjunction account for these two possibilities. One aircraft is safely above the other if they are safely separated during each of the maneuvers in the conflict interval; the inner conjunction over i accounts for each maneuver. We formalize and prove the following theorem in Coq:

Theorem 6 (Safely separated vertical trajectories). *The predicate Ψ computes whether a particular encounter is safe (i.e. collision-free) according to Eq. 15, for n time intervals $([t_{e1}, t_{x1}], [t_{e2}, t_{x2}], \dots, [t_{en}, t_{xn}])$ during a sequence of m maneuvers $([v_{min1}, v_{max1}], [v_{min2}, v_{max2}], \dots, [v_{minm}, v_{maxm}])$, with respective durations given by $(d_1, d_2, \dots, d_m)$.*

5 Formalizing Guarantees

We used Coq to formalize our proofs for this work, and this had both advantages and disadvantages compared with KeYmaera, which we had used previously. (A version of KeYmaera with scripting capabilities was unavailable for use since the system was between versions at the time of this work.) The immediate disadvantages of this change were that we could not concisely express our system using the specialized terms used for hybrid programs, and we did not have access to the reasoning strategies made available in differential-dynamic logic (dL), since presently there is no mechanization of dL in the Coq environment. Consequently, we expressed our model in terms of the more general framework of inductive constructions using higher order logic and Coq's expressive system of dependent types, and had to develop a set of lemmas about non-deterministic vertical motion from scratch, using Coq's Real library. The immediate advantage of this change was access to the well-developed scripting and automation capabilities of the relatively mature Coq environment, and the potential for integrating our present work with proofs that reason about trajectories involving trigonometric functions, as might be required for some types of non-deterministic horizontal turning behavior.

6 Extending Safety Guarantees to ACAS X

Our initial objective was to use this predicate formally verify that whenever possible, the system provided sequences of advice to the pilot that guaranteed safety and absence of collision under our acceleration assumptions.

ACAS X's complicated behavior is contained in a data structure that when uncompressed more than five hundred megabytes in size. The table is an optimal policy that minimizes costs associated with a Markov decision process representing the aircraft encounter. Reasoning about the table is challenging. There is discretization in the MDP, undersampling in the state space, and the logic of the table is related to optimizing a set of weights, whose relationship with actual safety in the real world is not straightforward.

The approach we took to formal verification treats the logic as opaque. Instead of creating a model of ACAS X that faithfully reproduces its details and quirks and trying to load it into a prover, we instead focused on evaluating its behavior throughout the state space. We developed the model described in Sect. 4, an independent logic for a collision-avoidance system that is safe-by-design. We prove it to be safe everywhere, and extend proofs about its safety to proofs about the safety of the real system. This extension is done via exploration of the system's state space, and comparison of the behavior at state points in the table to the allowable range of geometrically safe behaviors identified by our logic. The states in the table definitively determine the system's behavior in the continuous state space – the score function at off-table states are interpolated from the table's values in a local neighborhood. To evaluate each state, our predicate evaluates the future possibilities, taking into account pilot nondeterminism, sensor noise, and delay in the system, using the envelopes we previously described, acceleration limets, and the parameters of the NMAC puck. This approach makes it possible to do formal verification and draw conclusions about ACAS X over the entirety of its state space, but also makes the logic reusable for other collision avoidance systems.

To formally verify the system in its entirety with this approach, we would need to do two things: first, we would run the logic over all of the table's states, and then we would have to develop guarantees about off-table points in the state space. Proofs and reasoning would have to be developed to fill in the rest of the state space after the table's states were evaluated.

We ran an comprehensive evaluation of all the table's states in an earlier version of the system for straight-line trajectories. Our first comprehensive run took nearly a month to set up and run on our local cluster, returned so many examples of unsafe behavior that we had difficulty characterizing them. The initial results were that we quickly proved the system was not safe, and identified where. We almost immediately found areas where it gave unsafe advice, but where advice was possible that would guarantee safety.

Since we had counterexamples that will not be resolved, we could not prove safety comprehensively. At this point, we switched our focus from making comprehensive guarantees about the system's behavior to making local guarantees of safety or dangerous conditions, and characterizing the safety tradeoff made during its design.

7 Application to ACAS X Coordination Logic

This section describes how the vertical safety predicate was used to evaluate safety of ACAS X, for encounters where both aircraft are equipped with ACAS X and are executing coordinated vertical safety maneuvers simultaneously. This analysis was not possible earlier, because the previous framework we used [6,7] was fundamentally limited to analyzing a non-accelerating intruder; even vertical maneuvers for the intruder were not analyzable.

Using our new framework, we analyzed the advice generated by a prototype of ACAS X on a subset of the system's behavior table cut-points. We first collected the advisories that ACAS X issues on the chosen state space samples by querying ACAS X for both the ownship and intruder aircraft advisories.

The pilots of each aircraft are assumed to begin responding to an advisory 5 s after the first advisory is issued, and 3 s after each subsequent advisory. The safety predicate Ψ is evaluated at each selected state point with the harvested advisories assigned to the ownship and intruder accordingly. The horizontal motion model chosen here is the deterministic straight line model.

We called the state points where Ψ fails with the ACAS X advisories but succeeds with another set of ownship and intruder advisories *counterexamples*. A counterexample is a point in the state space where ACAS X issues advisories that are not guaranteed to be safe according to Ψ but there are other advisories that would guarantee safety. In the terminology of [6,7], Ψ is a *safeable* predicate.

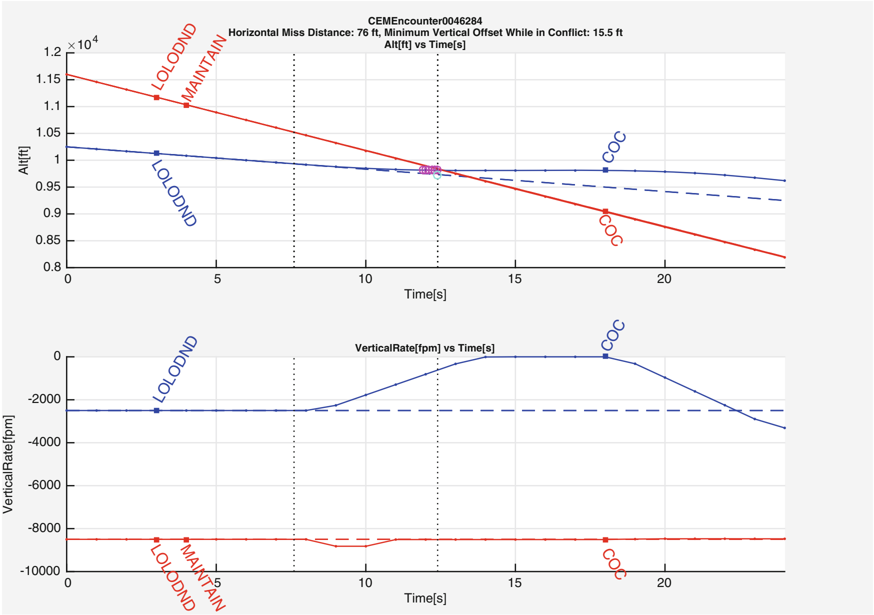

Fig. 4. An example NMAC found where the set of ACAS X advisories does not prevent a close approach in altitude during the period where the aircraft are within 500 ft horizontally, denoted by vertical dashed lines. Ownship and intruder trajectories are shown in blue and red, respectively. (Color figure online)

Recall, that a state point advisory combination is safeable if it is safe or can be made safe in the future by issuing additional advisories after a limited delay.

Of the 589,560 state points examined, 29,295 were identified as safeable counterexamples. To identify the most dangerous state space configurations the safeable counterexample set was further run through full ACAS X simulations with nominal trajectory accelerations set to zero. The result was a set of 3,301 state points where the system issued advice that created NMACs.

Examining the above set of dangerous aircraft configurations in terms of their state space coordinates, we observed a striking pattern—all of them had a low or moderate horizontal closing speed of between 10 and 200 ft/s. In practice, this means that the aircraft will remain in horizontal proximity for an extended period of time. For example, at the horizontal closing rate of 100 ft/s it can take the aircraft up to 10 s to clear the horizontal projection of the NMAC region.

Figure 4 shows conditions found by our analysis where ACAS X advice that does not guarantee safety. The two aircraft follow nearly parallel horizontal paths that cross at a very small angle (not shown). The intruder aircraft (red track) descends rapidly at −8500 ft/min, while the ownship (blue track) descends at a more moderate rate of −2500 ft/min. The dotted vertical lines indicate the time interval during which the aircraft are within 500 ft of each other and, hence, must maintain vertical separation of at least 100 ft to avoid NMAC.

The resolution advisories issued by ACAS X—DO NOT DESCEND (DND) and MAINTAIN VERTICAL SPEED (MAINTAIN), for the ownship and intruder aircraft respectively – result in an NMAC at time 12 s. The dotted blue line indicates the straight line continuation of the ownship trajectory that would have occurred with no advisory. To guarantee safety, ACAS X could continue to advise DO NOT DESCEND to the intruder, while advising the ownship to MAINTAIN vertical velocity.

These results pointed to an important flaw in the system assumptions about the possible range of durations of horizontal proximity. The problems stemming from slow horizontal closing configurations are actively being addressed in the final ACAS X system.

8 Related Work

Many efforts have explored developing correct and comprehensive guarantees about collision avoidance decisions over a system's state space. This paper improves on these because it develops guaranteed geometric safety under more realistic dynamics. The ACAS X system logic [8] is based on a policy that results from optimizing a Markov Decision Process (MDP) using value iteration to minimize a set of costs; [2,10] analyze the state space of a similar MDP using probabilistic model checking and an adaptive Monte Carlo tree search respectively, to identify undesirable behavior. Collision avoidance algorithms are developed for both horizontal and vertical motion in 3D in [13,14] for polynomial trajectories with a finite time horizon, and formally verified with PVS. TCAS, the predecessor for ACAS X. Its resolution advisories have been formalized in PVS. In [12], the logic for TCAS is formalized in PVS and used to identify straight-line encounter geometries that generate advisories in a noiseless environment.

There are a number of simulation approaches [1,5] that allow for more precise description of dynamics than the present work. However are limited to evaluating safety for a finite number of trajectories.

Prior efforts that match our dynamics as well as providing a formal proof of safety can be found in [4,11,15,16]. All these use a hybrid system model to develop safe horizontal maneuvers, unlike the present work which develops vertical maneuvers, and is applied to a practical system.

The most closely related work is [6,7]. We retain the overall approach to verification, very similar non-deterministic dynamics, and the idea of computing reachable envelopes to make guarantees about a range of future possibilities. The present work differs because it can analyze the safety of encounters with each aircraft making independent sequences of non-deterministic maneuvers, including acceleration, turns, and pilot delay. The proofs here are formalized in Coq.

9 Conclusion

This framework and the detailed vertical predicates offer a flexible approach to a formally verified analysis of the safety of a collision avoidance system. It relaxes restrictive assumptions about acceleration and horizontal motion and allows us to ensure the safety of a wider variety of pilot behavior and ACAS X system conditions than before. This analysis can ensure the safety of intruders that accelerate vertically, aircraft that make horizontal turns, coordinated ACAS X advisories, and multi-threat encounters. Its flexibility extends, further, to ensuring safe vertical motion in the presence of mixed horizontal and vertical advisories.

Acknowledgments. We gratefully acknowledge Neal Suchy and Josh Silbermann for their leadership and support. We thank André Platzer, Ryan Gardner and Christopher Rouff for their comments and technical discussion.

References

1. Chludzinski, B.J.: Evaluation of TCAS II version 7.1 using the FAA fast-time encounter generator model. Technical report ATC-346, MIT Lincoln Laboratory (2009)
2. Essen, C., Giannakopoulou, D.: Analyzing the next generation airborne collision avoidance system. In: Ábrahám, E., Havelund, K. (eds.) TACAS 2014. LNCS, vol. 8413, pp. 620–635. Springer, Heidelberg (2014). doi:10.1007/978-3-642-54862-8_54
3. Federal Aviation Administration: Introduction to TCAS II, Version 7.1 (2011)
4. Ghorbal, K., Jeannin, J.B., Zawadzki, E., Platzer, A., Gordon, G.J., Capell, P.: Hybrid theorem proving of aerospace systems: applications and challenges. J. Aerosp. Inf. Syst. **11**, 202–713 (2014)
5. Holland, J.E., Kochenderfer, M.J., Olson, W.A.: Optimizing the next generation collision avoidance system for safe, suitable, and acceptable operational performance. Air Traffic Control Q. **21**, 275–297 (2014)
6. Jeannin, J., Ghorbal, K., Kouskoulas, Y., Gardner, R., Schmidt, A., Zawadzki, E., Platzer, A.: Formal verification of ACAS X, an industrial airborne collision avoidance system. In: Girault, A., Guan, N. (eds.) 2015 International Conference on Embedded Software, EMSOFT 2015, Amsterdam, The Netherlands, 4–9 October 2015. ACM (2015)
7. Jeannin, J.-B., Ghorbal, K., Kouskoulas, Y., Gardner, R., Schmidt, A., Zawadzki, E., Platzer, A.: A formally verified hybrid system for the next-generation airborne collision avoidance system. In: Baier, C., Tinelli, C. (eds.) TACAS 2015. LNCS, vol. 9035, pp. 21–36. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46681-0_2
8. Kochenderfer, M.J., Chryssanthacopoulos, J.P.: Robust airborne collision avoidance through dynamic programming. Technical report ATC-371, MIT Lincoln Laboratory (2010)
9. Kochenderfer, M.J., Holland, J.E., Chryssanthacopoulos, J.P.: Next generation airborne collision avoidance system. Lincoln Lab. J. **19**(1), 17–33 (2012)
10. Lee, R., Kochenderfer, M.J., Mengshoel, O.J., Brat, G.P., Owen, M.P.: Adaptive stress testing of airborne collision avoidance systems. In: 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), p. 6C2-1. IEEE (2015)
11. Loos, S.M., Renshaw, D.W., Platzer, A.: Formal verification of distributed aircraft controllers. In: HSCC, pp. 125–130. ACM (2013). doi:10.1145/2461328.2461350
12. Muñoz, C., Narkawicz, A., Chamberlain, J.: A TCAS-II resolution advisory detection algorithm. In: Proceedings of the AIAA Guidance Navigation, and Control Conference and Exhibit 2013, AIAA-2013-4622, Boston, Massachusetts (2013)

13. Narkawicz, A., Muñoz, C.: Formal verification of conflict detection algorithms for arbitrary trajectories. Reliab. Comput. **17**, 209–237 (2012)
14. Narkawicz, A., Muñoz, C.: A formally verified conflict detection algorithm for polynomial trajectories. In: Proceedings of the 2015 AIAA Infotech@ Aerospace Conference, Kissimmee, Florida (2015)
15. Platzer, A., Clarke, E.M.: Formal verification of curved flight collision avoidance maneuvers: a case study. In: Cavalcanti, A., Dams, D.R. (eds.) FM 2009. LNCS, vol. 5850, pp. 547–562. Springer, Heidelberg (2009). doi:10.1007/978-3-642-05089-3_35
16. Tomlin, C., Pappas, G.J., Sastry, S.: Conflict resolution for air traffic management: a study in multiagent hybrid systems. IEEE Trans. Autom. Control **43**(4), 509–521 (1998)