A Formally Verified Hybrid System for the Next-Generation Airborne Collision Avoidance System

Jean-Baptiste Jeannin[1] **Khalil Ghorbal**[1]
Yanni Kouskoulas[2] **Ryan Gardner**[2] **Aurora Schmidt**[2]
Erik Zawadzki[1] **André Platzer**[1]

October 2014
CMU-CS-14-138

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[1] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213
[2] The Johns Hopkins Applied Physics Laboratory, Laurel, MD 20723

This material is based upon work supported by the Federal Aviation Administration Traffic Alert & Collision Avoidance System (TCAS) Program Office (PO) AJM-233 and by the National Science Foundation under NSF CAREER Award CNS-1054246. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution or government. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of any sponsoring institution or government.

Keywords: Airborne Collision Avoidance; ACAS X; Hybrid Systems; Theorem Proving; Federal Aviation Administration; Aircraft; Markov Decision Processes; Cyber Physical Systems

Abstract

The next-generation Airborne Collision Avoidance System (ACAS X) is intended to be installed on all large aircraft to give advice to pilots and prevent mid-air collisions with other aircraft. It is currently being developed by the Federal Aviation Administration (FAA). In this paper we determine the geometric configurations under which the advice given by ACAS X is safe under a precise set of assumptions and formally verify these configurations using hybrid systems theorem proving techniques. We conduct an initial examination of the current version of the real ACAS X system and discuss some cases where our safety theorem conflicts with the actual advisory given by that version, demonstrating how formal, hybrid approaches are helping ensure the safety of ACAS X. Our approach is general and could also be used to identify unsafe advice issued by other collision avoidance systems or confirm their safety.

1 Introduction

With the growing air traffic, the airspace becomes ever more crowded and the risk of airborne collisions between aircraft increases. In the 1970s, after a series of mid-air collisions, the Federal Aviation Administration (FAA) decided to develop an onboard collision avoidance system, the Traffic Alert and Collision Avoidance System TCAS. This program has had great success, preventing many mid-air collisions over the years. Some accidents still happened however, for example a collision over Überlingen in 2002, due to conflicting instructions between TCAS and the air traffic controller. Airspace management will evolve significantly over the next decade with the introduction of the next-generation air traffic management system, creating new requirements for collision avoidance and requiring a costly redesign of TCAS. To meet these new requirements, the FAA has decided to develop a new system, the Next-Generation Airborne Collision Avoidance System, known as ACAS X [Fed11, HKO14, KHC12].

Like TCAS, ACAS X avoids collisions by giving vertical guidance to an aircraft's pilot. A typical scenario involves two aircraft, the *ownship* where ACAS X is installed, and another aicraft called *intruder* that is at risk of colliding with the ownship. The Collision Avoidance community defines a *Near Mid-Air Collision* (*NMAC*) when the two aircraft are within $h_p = 500$ ft horizontally and $r_p = 100$ ft vertically [KHC12]. ACAS X is designed to avoid such NMACs. This describes a volume centered around the ownship, shaped like a hockey *puck* of radius r_p and half-height h_p, such that an NMAC occurs if the intruder enters that puck.

In order to be accepted by pilots and thus operationally suitable, ACAS X needs to strike a balance between giving advice that helps pilots avoid collisions as needed yet minimizing interruptions to pilots' normal activities. These goals oppose each other, and cannot both be perfectly met in the presence of uncertainty like undetermined pilot and intruder aircraft behavior, for example. This paper focuses on precisely characterizing the circumstances in which ACAS X gives advice that is safe. An integral part of the ACAS X development process, this work is intended to help ensure that the design of ACAS X is correct, potentially by identifying ways it should be adjusted.

1.1 Airborne Collision Avoidance System ACAS X

In order to prevent an NMAC with other aircraft, ACAS X uses various sensors to determine the position of the ownship, as well as the positions of any intruder [Fed14a]. It computes its estimate of the best pilot action by linearly interpolating a precomputed *table* of actions that optimize a Markov Decision Process. If appropriate, ACAS X alerts the pilot issuing an *advisory* to avoid potential collisions [Fed14b] through a visual display in the cockpit and a voice message.

An advisory is a request to the pilot of the ownship to alter or maintain her vertical speed. ACAS X advisories are strictly vertical, no advisories request the ownship to turn. ACAS X can also communicate Clear of Conflict (COC) to the pilot when no action needs to be taken. Besides COC, ACAS X can generate any of 16 advisories, summarized in Table 1. For example, the advisory Do Not Climb (DNC) requests the pilot to not climb, and the advisory Climb 1500 (CL1500) requests the pilot to start a climb at more than 1500 ft/min. Other advisories include Maintain Climb (MCL) and Subsequent Climb 2500 (SCL2500), which always follows a previous

Table 1: Advisories and their modeling variables

	ACAS X Specification [KC10]				Our model	
	Vertical Rate Range		Strength	Delay (s)		
Advisory	Min (ft/min)	Max (ft/min)	a_r	d_p	w	$\dot{h}_f$ (ft/min)
DNC2000	$-\infty$	$+2000$	g/4	5	-1	$+2000$
DND2000	-2000	$+\infty$	g/4	5	$+1$	-2000
DNC1000	$-\infty$	$+1000$	g/4	5	-1	$+1000$
DND1000	-1000	$+\infty$	g/4	5	$+1$	-1000
DNC500	$-\infty$	$+500$	g/4	5	-1	$+500$
DND500	-500	$+\infty$	g/4	5	$+1$	-500
DNC	$-\infty$	0	g/4	5	-1	0
DND	0	$+\infty$	g/4	5	$+1$	0
MDES	$-\infty$	current	g/4	5	-1	current
MCL	current	$+\infty$	g/4	5	$+1$	current
DES1500	$-\infty$	-1500	g/4	5	-1	-1500
CL1500	$+1500$	$+\infty$	g/4	5	$+1$	$+1500$
SDES1500	$-\infty$	-1500	g/3	3	-1	-1500
SCL1500	$+1500$	$+\infty$	g/3	3	$+1$	$+1500$
SDES2500	$-\infty$	-2500	g/3	3	-1	-2500
SCL2500	$+2500$	$+\infty$	g/3	3	$+1$	$+2500$
COC	$-\infty$	$+\infty$	Not applicable			

advisory. To comply with an advisory, the pilot needs to adjust her vertical rate to match the corresponding vertical rate range specified in Table 1. Based on previous research [KC10], the pilot is assumed to do so using a vertical acceleration of strength at least a_r starting after a delay of at most d_p after the advisory has been announced by ACAS X.

At the heart of the system is the ACAS X table whose domain describes a grid of possible configurations for the current state, and whose range is a set of scores for each action that can be taken [KC10, KM13]. The table is obtained from a Markov Decision Process (MDP) approximating the dynamics of the system by a discretization on that grid, and optimized using dynamic programming to maximize the combined value of events over all future paths for each action [KC10]. States representing a near mid-air collision (NMAC) are given large negative weights and actions are given small negative weights. The policy is then to choose the actions with highest value from a multilinear interpolation of grid points in this table. ACAS X uses this table, along with some heuristics, to determine the best action to take for the geometry in which it finds itself.

1.2 Identifying Formally Verified Safe Regions

ACAS X involves both *discrete* advisories to the pilot and *continuous* dynamics of aircraft, it thus seems natural to formally verify it using hybrid systems. However the complexity of ACAS X, and in particular the core use of a large lookup table—defining 29,212,664 interpolation regions within

a 5-dimensional state-space—makes the direct use of hybrid systems verification techniques intractable. Our approach is different. It identifies *safe regions* in the state space of the system where the current positions and velocities of the aircraft ensure that a particular advisory, if followed, prevents all possible NMACs. Then it *compares* these regions to the configurations where the ACAS X table returns this same advisory.

Our results provide independent characterizations of the ACAS X behavior to provide a clear and complete picture of its performance. Our method can be used by the ACAS X development team in two ways. It provides a mathematical proof—with respect to a model—that ACAS X is absolutely safe for some configurations of the aircraft. Additionally, when ACAS X is not safe, it is able to identify unsafe or unexpected behaviors and suggests ways of correcting them.

Our approach of formally deriving safe regions then comparing them to the behavior of an industrial system is, as far as we are aware, the first of its kind in the formal verification of hybrid systems. The approach may be valuable for verifying or assessing properties of other systems with similar complexities, or also using large lookup tables, which is a common challenge in practice. Finally, the constraints we identified for safety are fairly general and could be used to analyze other collision avoidance systems.

The paper is organized as follows. We first create a simple, independent model of aircraft dynamics using continuous state variables and determine the safe regions for which an advisory is absolutely safe within that model. After an overview of the method in Sect. 2, we start with a simple two-dimensional model assuming immediate reaction of the pilot in Sect. 3. We extend the model to account for the reaction time of the pilot in Sect. 4, and extend the results to a three-dimensional model in Sect. 5. In Sect. 6, we conduct an initial analysis of ACAS X whereby we compare the advisory recommended by a core component of ACAS X with our safe regions, identifying the circumstances where safety of those ACAS X advisories is guaranteed within our model.

2 Overview of the ACAS X Modelling Approach

To construct a safe region of an advisory for an aircraft, imagine following all allowable trajectories of the ownship relative to the intruder, accounting for every possible position of the ownship and its surrounding puck at every future moment in time. The union of all such positions of the puck describes a potentially unsafe region since for each point there exists a trajectory of the ownship that may cause an NMAC. If the intruder is outside this set, then no NMAC is possible. Dually, the safe region describes a zone where no circumstances of the encounter can lead to an NMAC in the model.

Fig. 1 represents an example of a head-on encounter and its associated safe region for the advisory CL1500, projected in a vertical plane containing both aircraft.[1] Fig. 1 is plotted in a frame fixed to the intruder and centered at the initial position of the ownship. At the beginning of the encounter, the ownship starts at the origin and the intruder starts at some other location; as time progresses, the ownship traces out a trajectory following the red curve. The intruder remains fixed in this reference frame. The ownship, surrounded by the puck, starts on the left at position number

[1]Throughout this paper, the aircraft drawings are courtesy of `http://sweetclipart.com`

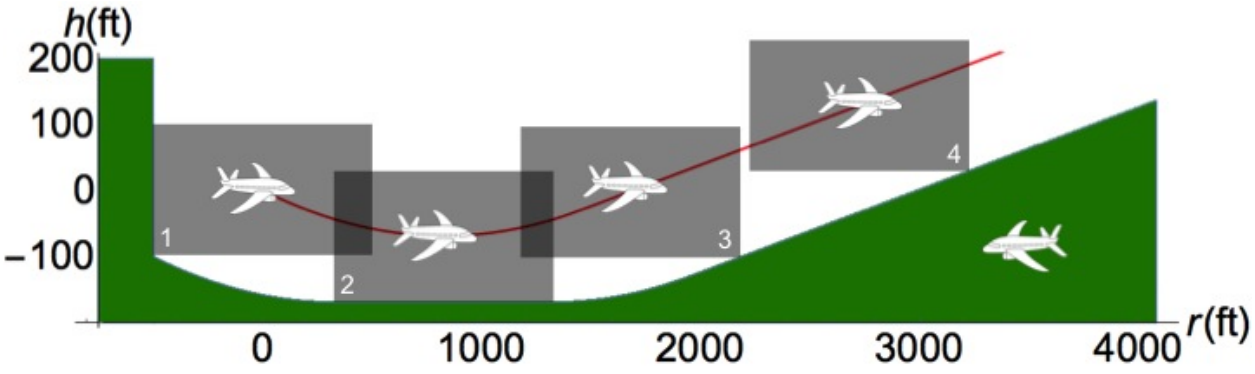

Figure 1: Trajectory of ownship (red) and safe region for the intruder (green), immediate response

1. It first accelerates vertically with $g/4$ until reaching the desired vertical velocity of $+1500$ ft/min at position number 3, then climbs at $+1500$ ft/min, thus respecting the specification of Table 1. The green safe region indicates starting points in the state space for which the intruder will remain safe for the entire duration of the encounter. Note that there is no safe region above the trajectory, since according to the specification of ACAS X in Table 1, the ownship could accelerate vertically at more than $g/4$ or past $+1500$ ft/min.

2.1 Model of Dynamics

Let us consider an encounter between two planes—ownship O and intruder I, as portrayed in Fig. 2. Following the notations of [KC10] and the ACAS X community, let r be the horizontal distance between the aircraft, and h the relative height of the intruder with respect to the ownship. We assume that the relative horizontal velocity $\vec{r_v}$ of the intruder with respect to the ownship is constant throughout the encounter, so from a top view, the planes are both following straight-line trajectories. (They do not have to be coming straight at each other, but they are not allowed to turn left or right during the encounter.) Let θ_v be the non-directed angle between the relative speed $\vec{r_v}$ of the aircraft and the line segment $\vec{r}$. In the vertical dimension, we assume that the ownship's vertical velocity $\dot{h}_0$ can vary at any moment, while the intruder's vertical velocity $\dot{h}_1$ is fixed throughout the encounter: the intruder has no vertical acceleration. Moreover, we assume that the magnitude of the vertical acceleration of the ownship cannot exceed a_d in absolute value. Finally, recall that an NMAC is defined as the two aircraft being within r_p horizontally and h_p vertically.

For a typical encounter, r varies between 0 nmi and 7 nmi,[2] h between $-4,000$ ft and $4,000$ ft, r_v between 0 kts and $1,000$ kts, and $\dot{h}_0$ and $\dot{h}_1$ between $-5,000$ ft/min and $+5,000$ ft/min. The acceleration a_d is usually $g/2$, where g is Earth's gravitational acceleration. The NMAC *puck* has radius $r_p = 500$ ft and half-height $h_p = 100$ ft.

[2]We use units most common in the aerospace community, even though they are not part of the international system, including nautical miles nmi ($1,852$ metres), knots kts (nautical miles per hour), feet ft (0.3048 meter) and minutes min (60 seconds).

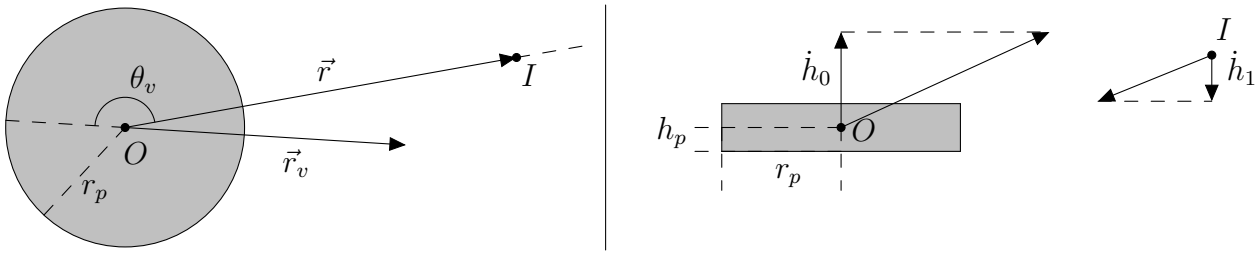

Figure 2: Top view (left) and side view (right) of an encounter

2.2 Model of Advisories

Recall that ACAS X prevents an NMAC by giving an advisory to the pilot of the ownship. From Table 1 every advisory, except the clear of conflict (COC), has a vertical rate range of the form $(-\infty, \dot{h}_f]$ or $[\dot{h}_f, +\infty)$ for some vertical rate $\dot{h}_f$, which we call the *target vertical velocity*. We therefore model any advisory by its corresponding target vertical velocity $\dot{h}_f$, and a binary variable w, whose value is -1 if the vertical rate range of the advisory is $(-\infty, \dot{h}_f]$ and $+1$ if it is $[\dot{h}_f, +\infty)$. Note that this symbolic encoding makes it possible to represent much more advisories and is therefore robust to changes in the set of advisories that ACAS X allows.

Following the specification of ACAS X [KC10], we assume that the ownship pilot complies with each advisory within d_p seconds, and that they accelerate with at least acceleration a_r to bring the relative vertical velocity in compliance with the advisory.

3 Safe Region for an Immediate Pilot Response

To simplify the presentation, we present a simplified version of the dynamics from Sect. 2.1 in this section. We then give a hybrid model for the system and prove its safety.

3.1 Model

In this section, we assume that the ownship and intruder are flying head-on ($\theta_v = 180°$). We also assume that the pilot reacts immediately to any advisory ($d_p = 0$ s), and that the advisory COC is not allowed. The last assumptions will be relaxed in Sect. 4, and the first one in Sect. 5. Finally, we assume that r is a scalar: if $r \geq 0$ then the ownship is flying towards the intruder, otherwise it is flying away from it.

Since we assume that the ownship and intruder are flying head-on with straight line trajectories, there exists a vertical plane containing both their trajectories. In this plane, the puck becomes a rectangle centered around the ownship, of width $2r_p$ and height $2h_p$, and there is an NMAC if and only if the intruder is in this rectangle (Fig. 1).

3.2 Differential Dynamic Logic and KeYmaera

We model our system using Differential Dynamic Logic d$\mathcal{L}$ [Pla08, Pla10, Pla12], a logic for reasoning about hybrid programs. d$\mathcal{L}$ allows discrete assignments, control structures, and execution of differential equations. d$\mathcal{L}$ is implemented in the theorem prover KeYmaera [PQ08], that we use to verify our safe regions with respect to our models. Our KeYmaera models and proofs can be found at http://www.cs.cmu.edu/~jeannin/acasx.zip, and statistics in Appendix A.

The d$\mathcal{L}$ formula for the model that we use in this section is given in Eq. (1).

$$
\begin{array}{l}
1\ \ h_p > 0 \land r_p \geq 0 \land r_v \geq 0 \land a_r > 0 \land (w = -1 \lor w = 1) \land C_{\mathsf{impl}}(r, h, \dot{h}_0) \rightarrow \\
2\ \ [(\ ?\mathsf{true} \cup (\ \dot{h}_f := *; (w := -1 \cup w := 1); ?C_{\mathsf{impl}}(r, h, \dot{h}_0); \mathsf{advisory} := (w, \dot{h}_f)\ ); \\
3\ \ \quad a := *; \{r' = -r_v, h' = -\dot{h}_0, \dot{h}_0' = a\ \&\ w\dot{h}_0 \geq w\dot{h}_f \lor wa \geq a_r\} \\
4\ \ )^*]\ (|r| > r_p \lor |h| > h_p)
\end{array}
\tag{1}
$$

This formula of the form $p \rightarrow [\alpha]q$ says all executions of program α starting in a state satisfying logical formula p end up in a state satisfying q. It is akin to the Hoare triple $\{p\}\alpha\{q\}$ with precondition p and postcondition q. The precondition in Eq. (1) imposes constraints on several constants, as well as the formula $C_{\mathsf{impl}}(r, h, \dot{h}_0)$ (defined below) that forces the intruder to be in a safe region for an initial advisory $(w, \dot{h}_f)$. We cannot guarantee safety if the intruder starts initially in an unsafe region. The postcondition encodes absence of NMAC. Line 2 expresses the action of the ACAS X system. The nondeterministic choice operator $\cup$ expresses that the system can either continue with the same advisory by doing nothing—just testing $?\mathsf{true}$—this ensures it always has a valid choice and cannot get stuck. Otherwise it can choose a new $\mathsf{advisory}$ $(w, \dot{h}_f)$ that passes the safety condition $C_{\mathsf{impl}}(r, h, \dot{h}_0)$. Line 3 expresses the action of the ownship, first nondeterministically choosing an arbitrary acceleration ($a := *$) then following the continuous dynamics. This line characterizes the evolutions of the variables r, h and $\dot{h}_0$ by a differential equation, and requires (using the operator $\&$) that the ownship evolves towards its target vertical velocity $\dot{h}_f$ at acceleration a_r ($wa \geq a_r$), unless it has already reached it ($w\dot{h}_0 \geq w\dot{h}_f$). Finally, the star * on line 4 indicates that the program can be repeated any number of times, allowing the system to go through several advisories.

3.3 Implicit Formulation of the Safe Region

As explained in Sect. 2, we use a frame fixed to the intruder and centered at the original position of the ownship (see Fig. 1).

First case: if $w = +1$ and $\dot{h}_f \geq \dot{h}_0$. Fig. 1 shows, in red, a possible trajectory of an ownship following exactly the requirements of ACAS X. This trajectory is the *nominal* trajectory of the ownship and will be denoted by $\mathcal{N}$. The pilot reacts immediately, and the ownship starts accelerating vertically with acceleration a_r until reaching the target vertical velocity $\dot{h}_f$—describing a parabola—then climbs at vertical velocity $\dot{h}_f$ along a straight line. Horizontally, the relative velocity r_v remains constant. The ownship position (r_t, h_t) along the nominal trajectory $\mathcal{N}$ is given

by:

$$(r_t, h_t) = \begin{cases} \left(r_v t\ ,\ \frac{a_r}{2}t^2 + \dot{h}_0 t\right) & \text{if } 0 \le t < \frac{\dot{h}_f - \dot{h}_0}{a_r} \qquad (a) \\ \left(r_v t\ ,\ \dot{h}_f t - \frac{(\dot{h}_f - \dot{h}_0)^2}{2a_r}\right) & \text{if } \frac{\dot{h}_f - \dot{h}_0}{a_r} \le t \qquad (b) \end{cases} \tag{2}$$

Recall that in the ACAS X specification, the ownship accelerates with vertical acceleration *at least* a_r, then continues at vertical velocity of *at least* $\dot{h}_f$. Therefore all possible future positions of the ownship are *above* the red nominal trajectory. An intruder is safe if it is always to the side or under any puck centered on a point of the nominal trajectory $\mathcal{N}$, or in other words, if: for all (r_t, h_t) along the nominal trajectory $\mathcal{N}$ of the ownship, the intruder (r, h) is either strictly to the right or to the left of the puck ($|r - r_t| > r_p$), or it is under the puck ($h - h_t < -h_p$). That is,

$$\forall t.\forall r_t.\forall h_t.(r_t, h_t) \in \mathcal{N} \Rightarrow |r - r_t| > r_p \vee h - h_t < -h_p \tag{3}$$

We call this formulation the *implicit formulation of the safe region*. It does not give explicit equations for the safe region border, but expresses them instead implicitly with respect to the nominal trajectory.

Generalization. The reasoning above is generalized to the case where $\dot{h}_f < \dot{h}_0$, and symmetrically to the case $w = -1$. The most general implicit formulation of the safe region is C_{impl} in Fig. 3.

Theorem 1 (Correctness of implicit safe regions)**.** *The d$\mathcal{L}$ formula given in Eq.* (1) *is valid. That is as long as the advisories obey formula C_{impl} there will be no NMAC.*

3.4 Explicit Formulation of the Safe Region

The implicit formulation of the safe region gives an intuitive understanding of where it is safe for the intruder to be. Because it still contains quantifiers, its use comes at the extra cost of eliminating the quantifiers. In this section we derive a quantifier-free *explicit formulation* of the safe region. We show that both formulations are equivalent in our setting. As for the implicit formulation, we derive the equations for one representative case before generalizing them.

First case: if $w = +1$, $r_v > 0$, $\dot{h}_0 < 0$ and $\dot{h}_f \ge 0$. We are in the case shown in Fig. 1 and described in details above. The nominal trajectory $\mathcal{N}$ is given by Eq. (2)(a) and Eq. (2)(b). The boundary of the (green) safe region in Fig. 1 is drawn by either the bottom left hand corner, the bottom side or the bottom right hand corner of the puck. This boundary can be characterized by a set of equations:

- positions left of the puck's initial position ($r < -r_p$) are in the safe region;
- then the boundary follows the bottom left hand corner of the puck as it is going down the parabola of Eq. (2)(a); therefore for $-r_p \le r < -r_p - \frac{r_v \dot{h}_0}{a_r}$, the position (r, h) is safe if and only if $h < \frac{a_r}{2r_v^2}(r + r_p)^2 + \frac{\dot{h}_0}{r_v}(r + r_p) - h_p$;

Implicit formulation

$$A(t,h_t,\dot{h}_0) \equiv \left(\begin{array}{rcl} 0 \le t < \dfrac{\max(0,w(\dot{h}_f-\dot{h}_0))}{a_r} & \wedge & h_t = \dfrac{wa_r}{2}t^2+\dot{h}_0 t \\ \vee & & \\ t \ge \dfrac{\max(0,w(\dot{h}_f-\dot{h}_0))}{a_r} & \wedge & h_t = \dot{h}_f t - \dfrac{w\max(0,w(\dot{h}_f-\dot{h}_0))^2}{2a_r} \end{array}\right)$$

$$C_{\mathsf{impl}}(r,h,\dot{h}_0) \equiv \forall t.\forall r_t.\forall h_t.\left(r_t = r_v t \wedge A(t,h_t,\dot{h}_0) \Rightarrow (|r-r_t| > r_p \vee w(h-h_t) < -h_p)\right)$$

Explicit formulation

$$\mathsf{case}_1(r,\dot{h}_0) \equiv -r_p \le r < -r_p - \frac{r_v \min(0,w\dot{h}_0)}{a_r}$$

$$\mathsf{bound}_1(r,h,\dot{h}_0) \equiv wr_v^2 h < \frac{a_r}{2}(r+r_p)^2 + wr_v\dot{h}_0(r+r_p) - r_v^2 h_p$$

$$\mathsf{case}_2(r,\dot{h}_0) \equiv -r_p - \frac{r_v \min(0,w\dot{h}_0)}{a_r} \le r \le r_p - \frac{r_v \min(0,w\dot{h}_0)}{a_r}$$

$$\mathsf{bound}_2(r,h,\dot{h}_0) \equiv wh < -\frac{\min(0,w\dot{h}_0)^2}{2a_r} - h_p$$

$$\mathsf{case}_3(r,\dot{h}_0) \equiv r_p - \frac{r_v \min(0,w\dot{h}_0)}{a_r} < r \le r_p + \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))}{a_r}$$

$$\mathsf{bound}_3(r,h,\dot{h}_0) \equiv wr_v^2 h < \frac{a_r}{2}(r-r_p)^2 + wr_v\dot{h}_0(r-r_p) - r_v^2 h_p$$

$$\mathsf{case}_4(r,\dot{h}_0) \equiv r_p + \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))}{a_r} < r$$

$$\mathsf{bound}_4(r,h,\dot{h}_0) \equiv (r_v = 0) \vee \left(wr_v h < w\dot{h}_f(r-r_p) - \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))^2}{2a_r} - r_v h_p\right)$$

$$\mathsf{case}_5(r,\dot{h}_0) \equiv -r_p \le r < -r_p + \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))}{a_r}$$

$$\mathsf{bound}_5(r,h,\dot{h}_0) \equiv wr_v^2 h < \frac{a_r}{2}(r+r_p)^2 + wr_v\dot{h}_0(r+r_p) - r_v^2 h_p$$

$$\mathsf{case}_6(r,\dot{h}_0) \equiv -r_p + \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))}{a_r} \le r$$

$$\mathsf{bound}_6(r,h,\dot{h}_0) \equiv (r_v = 0 \wedge r > r_p) \vee \left(wr_v h < w\dot{h}_f(r+r_p) - \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))^2}{2a_r} - r_v h_p\right)$$

$$C_{\mathsf{expl}}(r,h,\dot{h}_0) \equiv \left(w\dot{h}_f \ge 0 \Rightarrow \bigwedge_{i=1}^{4}(\mathsf{case}_i(r,\dot{h}_0) \Rightarrow \mathsf{bound}_i(r,h,\dot{h}_0))\right) \wedge \left(w\dot{h}_f < 0 \Rightarrow \bigwedge_{i=5}^{6}(\mathsf{case}_i(r,\dot{h}_0) \Rightarrow \mathsf{bound}_i(r,h,\dot{h}_0))\right)$$

Figure 3: Implicit and explicit formulations of the safe region for an immediate response

- following this, the boundary is along the bottom side of the puck as it is at the bottom of the parabola of Eq. (2)(a); therefore for $-r_p - \frac{r_v \dot{h}_0}{a} \leq r \leq r_p - \frac{r_v \dot{h}_0}{a_r}$, the position (r, h) is in the safe region if and only if $h < -\frac{\dot{h}_0^2}{2a_r} - h_p$;
- then the boundary follows the bottom right hand corner of the puck as it is going up the parabola of Eq. (2)(a); therefore for $-r_p \leq r < -r_p - \frac{r_v \dot{h}_0}{a_r}$, the position (r, h) is safe if and only if $h < \frac{a_r}{2r_v^2}(r - r_p)^2 + \frac{\dot{h}_0}{r_v}(r - r_p) - h_p$;
- finally the boundary follows the bottom right hand corner of the puck as it is going up the straight line of Eq. (2)(b); therefore for $r_p + \frac{r_v(\dot{h}_f - \dot{h}_0)}{a_r} < r$, the position (r, h) is in the safe region if and only if $h < \frac{\dot{h}_f}{r_v}(r - r_p) - \frac{(\dot{h}_f - \dot{h}_0)^2}{2a_r} - h_p$.

Generalization. The general case is given in the formula C_{expl} of Fig. 3. The cases 1-4 and their associated bounds are for the case $w\dot{h}_f \geq 0$, whereas cases 5 and 6 and associated bounds are for $w\dot{h}_f < 0$. We use KeYmaera to formally prove that this explicit safe region formulation is equivalent to its implicit counterpart.

Lemma 2 (Correctness of explicit safe regions)**.** *If $w = \pm 1$, $r_p \geq 0$, $h_p > 0$, $r_v \geq 0$ and $a_r > 0$, then conditions $C_{\mathsf{impl}}(r, h, \dot{h}_0)$ and $C_{\mathsf{expl}}(r, h, \dot{h}_0)$ are equivalent.*

4 Safe Region for a Delayed Pilot Response

We generalize the model of Sect. 3 to account for a non-deterministic, non-zero pilot delay, and for periods of time where the system does not issue an advisory (i.e., COC).

4.1 Model

In this section, we still assume that the ownship and intruder are flying head-on ($\theta_v = 180°$). We use the same conventions as in Sect. 3 for r and r_v. To model the pilot reaction delay, we add an initial period of time d_p to our trajectory where the ownship accelerates non-deterministically (within limits) in the vertical direction. We derive the safe regions by considering all possible positions of the ownship's puck in all possible trajectories that might evolve in the encounter. We also use this delay to reason about the safety of the system displaying COC; for this to be safe, ACAS X has to be able to generate a safe advisory after a time d_ℓ, corresponding to the system delay. This corresponds to finding a safe advisory with a delay equal to the system delay plus the

maximum pilot's delay, i.e. $d_p + d_\ell$.

$$
\begin{array}{ll}
1 & r_p \geq 0 \wedge h_p > \ 0 \wedge r_v \geq 0 \wedge a_r > 0 \wedge a_d \geq 0 \wedge d_p \geq 0 \wedge d_\ell \geq 0 \\
2 & \wedge (w = -1 \vee w = 1) \wedge D_{\mathsf{impl}}(r, h, \dot{h}_0) \rightarrow \\
3 & [(?\mathsf{true} \cup \\
4 & \ \ (\dot{h}_f := *; (w := -1 \cup w := 1); \\
5 & \ \ \ (d := d_p; \mathsf{advisory} := (w, \dot{h}_f) \cup d := d_p + d_\ell; \mathsf{advisory} := \mathsf{COC}); ?D_{\mathsf{impl}}(r, h, \dot{h}_0)); \qquad (4) \\
6 & \ \ a := *; ?(wa \geq -a_d); t_\ell := 0; \\
7 & \ \ \{\dot{r} = -r_v, \dot{h} = -\dot{h}_0, \dot{\dot{h}}_0 = a, \dot{d} = -1, \dot{t_\ell} = 1 \ \& \\
8 & \ \ \ (t_\ell \leq d_\ell) \wedge (d \leq 0 \rightarrow w\dot{h}_0 \geq w\dot{h}_f \vee wa \geq a_r)\} \\
9 & \ )^*] \ (|r| > r_p \vee |h| > h_p)
\end{array}
$$

We modify the model of Eq. (1) to capture these new ideas, and obtain the model of Eq. (4). The structure, precondition (lines 1 and 2) and postcondition (line 9) are similar. The clock d, if positive, represents the amount of time until the ownship pilot must respond to the current advisory to remain safe. Lines 3 to 5 represent the actions of the ACAS X system. As before, the system can continue with the same advisory (?true). Otherwise it can select a safe advisory $(w, \dot{h}_f)$ to be applied after at most delay d_p; or it can safely remain silent, displaying COC, if it knows an advisory $(w, \dot{h}_f)$ that is safe if applied after delay $d_p + d_\ell$. In line 6, the pilot non-deterministically chooses an acceleration, within some limit. The set of differential equations in line 7 describes the system's dynamics, and the conditions in line 8 use the clock t_ℓ to ensure that continuous time does not evolve longer than d_ℓ without a system response. Those conditions also ensure that when $d \leq 0$ the pilot starts complying with the advisory. The model is structured so that the pilot can safely delay responding to an advisory for up to d_p, and to an advisory associated with COC for up to $d_p + d_\ell$. Because of the loop in our model (line 9), the safety guarantees of this theorem apply to encounters whose advisories change as the encounter evolves, encounters with periods of no advisory, and encounters where the pilot exhibits a range of non-deterministic behavior and delay.

In the rest of the section we use the same approach as in Sect. 3: we first derive an implicit formulation, then an equivalent explicit formulation of the safe region, and prove that the safe region guarantees that the intruder cannot cause an NMAC.

4.2 Formulations of the Safe Region

As in Sect. 3.3, let us place ourselves in the referential centered on the current position of the ownship and where the intruder is fixed, and let us first assume that the ownship receives an advisory $(w, \dot{h}_f)$ such that $w = +1$, and that $d \geq 0$. We focus on the period of the reaction time of the pilot, which we henceforth call delay. During the delay, the ownship can take any vertical acceleration less than a_d in absolute value, therefore its nominal trajectory $\mathcal{N}_d$ is to accelerate the opposite way of the advisory, at acceleration $-a_d$. Horizontally, its speed is constant at r_v. It thus describes a *delay parabola*, in red on Fig. 4, and its position (r_t, h_t) along the nominal trajectory for $0 \leq t < d$ is given by $(r_t, h_t) = \left(r_v t, -\frac{a_d}{2}t^2 + \dot{h}_0 t\right)$.

After the delay, i.e., after time d, the nominal trajectory $\mathcal{N}_d$ is the same as a nominal trajectory $\mathcal{N}$ from Sect. 3, translated by time d and by its position at time d given by $r_d = r_t(d)$ and $h_d =$

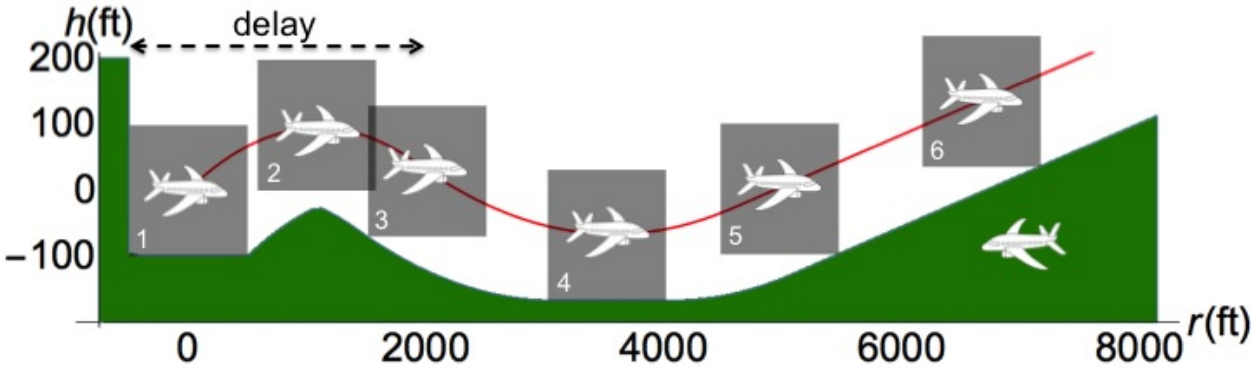

Figure 4: Trajectory of the ownship (red) and safe region for the intruder (green), delayed response

$h_t(d)$, and starting with vertical velocity $\dot{h}_d = \dot{h}_0 - a_d d$. As in Sect. 3.3, we can now express the implicit formulation of the safe region:

$$\forall t.\forall r_t.\forall h_t.(r_t, h_t) \in \mathcal{N}_d \Rightarrow |r - r_t| > r_p \vee h - h_t < -h_p$$

Symmetrically, the reasoning of this section extends to the case where $w = -1$. Moreover, we can handle cases beyond the reaction time of the pilot where $d < 0$ by replacing d by $\max(0, d)$. The generalized implicit formulation of the safe region is given as D_{impl} in Fig. 5. Note that it involves the expression $A(t - \max(0, d), h_t - h_d, \dot{h}_d)$ from Fig. 3 capturing the implicit safe region of Sect. 3.3 translated by time $\max(0, d)$, vertical height h_d, and starting at vertical speed $\dot{h}_d$. It is proved correct in KeYmaera.

Theorem 3 (Correctness of delayed safe regions)**.** *The d$\mathcal{L}$ formula given in Eq.* (4) *is valid. That is as long as the advisories obey formula D_{impl} there will be no NMAC.*

Similarly as in Sect. 4, we determine an explicit formulation of the safe region, called D_{expl} in Fig. 5 based on Fig. 3, and prove it correct in KeYmaera.

Lemma 4 (Correctness of delayed explicit safe regions)**.** *If $w = -1$ or $w = +1$, $r_p \geq 0$, $h_p > 0$, $r_v \geq 0$, $a_r > 0$, $a_d \geq 0$, $d_p \geq 0$ and $d_\ell \geq 0$ then the two conditions $D_{\mathsf{impl}}(r, h, \dot{h}_0)$ and $D_{\mathsf{expl}}(r, h, \dot{h}_0)$ are equivalent.*

5 Reduction from 3D Dynamics to 2D Dynamics

In this section, we show that, with respect to our assumptions, any 3-dimensional encounter can be reduced to a 2-dimensional encounter without loss of generality.

For the sake of clarity, let us this time work in a reference frame $(O, \vec{i}, \vec{j}, \vec{k})$ fixed to the ownship (O). In that reference frame, the position of an intruder I is represented by the tuple (x, y, h), and the differential equation system that governs its motion is given by $\dot{x} = r_x$, $\dot{y} = r_y$, $\ddot{h} = a$, where r_x, r_y and a remain constant as time evolves. Therefore, the motion of the encounter can be decoupled into a 2-dimensional horizontal encounter in the reference frame $(O, \vec{i}, \vec{j})$ (horizontal

Implicit formulation

$$
\begin{aligned}
B(t, h_t, \dot{h}_0) &\equiv 0 \leq t < \max(0, d) \wedge h_t = -\frac{wa_d}{2} t^2 + \dot{h}_0 t \\
\mathsf{const} &\equiv h_d = -\frac{wa_d}{2} \max(0, d)^2 + \dot{h}_0 \max(0, d) \wedge \dot{h}_d - \dot{h}_0 = -wa_d \max(0, d) \\
D_{\mathsf{impl}}(r, h, \dot{h}_0) &\equiv \forall t. \forall r_t. \forall h_t. \forall h_d. \forall \dot{h}_d. \\
&\quad \Big(r_t = r_v t \wedge (B(t, h_t, \dot{h}_0) \vee \mathsf{const} \wedge A(t - \max(0, d), h_t - h_d, \dot{h}_d)) \\
&\quad \Rightarrow (|r - r_t| > r_p \vee w(h - h_t) < -h_p) \Big)
\end{aligned}
$$

Explicit formulation

$$
\begin{aligned}
r_d &= r_v \max(0, d) & \dot{h}_d &= \dot{h}_0 - wa_d \max(0, d) \\
h_d &= -\frac{wa_d}{2} \max(0, d)^2 + \dot{h}_0 \max(0, d) \\
\mathsf{case}_7(r) &\equiv -r_p \leq r \leq r_p & \mathsf{bound}_7(r, h) &\equiv wh < -h_p \\
\mathsf{case}_8(r) &\equiv r_p < r \leq r_d + r_p & \mathsf{case}_9(r) &\equiv -r_p \leq r < r_d - r_p \\
\mathsf{bound}_8(r, h) &\equiv wr_v^2 h < -\frac{a_d}{2}(r - r_p)^2 + wr_v \dot{h}_0 (r - r_p) - r_v^2 h_p \\
\mathsf{bound}_9(r, h) &\equiv wr_v^2 h < -\frac{a_d}{2}(r + r_p)^2 + wr_v \dot{h}_0 (r + r_p) - r_v^2 h_p \\
D_{\mathsf{expl}}(r, h, \dot{h}_0) &\equiv \left(\bigwedge_{i=7}^{9} (\mathsf{case}_i(r) \Rightarrow \mathsf{bound}_i(r, h)) \right) \wedge C_{\mathsf{expl}}(r - r_d, h - h_d, \dot{h}_d)
\end{aligned}
$$

Figure 5: Implicit and explicit formulations of the safe region for a delayed response

plane) and a 1-dimensional vertical encounter in the reference frame $(O, \vec{k})$. In what follows, we reduce the horizontal encounter from a 2-dimensional motion to a 1-dimensional motion, thereby simplifying the problem conceptually and computationally by reducing its number of variables.

Fig. 6 depicts a top view of a generic encounter. We denote by $\vec{r}$ the position, and $\vec{r}_v$ the velocity, of the intruder relative to the ownship, and by $r_v \geq 0$ the norm of $\vec{r}_v$.

First suppose $r_v > 0$. The idea is to choose a reference frame $(O', \vec{i'}, \vec{j'})$ in which one axis $\vec{i'}$ is aligned with $\vec{r}_v$, such that no relative motion happens in the other direction $\vec{j'}$. Its fixed center O' is defined as the orthogonal projection of point O onto the direction of $\vec{r}_v$. The unit vector $\vec{i'}$ is defined as $\frac{\vec{r}_v}{r_v}$, and $\vec{j'}$ is a unit such that $(O', \vec{i'}, \vec{j'})$ is positively oriented.

Let $\vec{v}_{|O}$ (resp. $\vec{v}_{|O'}$) denote the coordinates of a vector $\vec{v}$ relative to the reference frame $(O, \vec{i}, \vec{j})$ (resp. $(O', \vec{i'}, \vec{j'})$). Then, the coordinates for $\vec{r}$ and $\vec{r}_v$ are: $\vec{r}_{|O} = (x, y)$, $\vec{r}_{v_{|O}} = (r_x, r_y)$, $\vec{r}_{|O'} = (s, n)$ and $\vec{r}_{v_{|O'}} = (-r_v, 0)$. The scalar product $\vec{r} \cdot \vec{r}_v$ and the cross product $\vec{r} \times \vec{r}_v$ are independent of the horizontal reference frame, therefore:

$$
xr_x + yr_y = -sr_v \qquad xr_y - yr_x = nr_v \tag{5}
$$

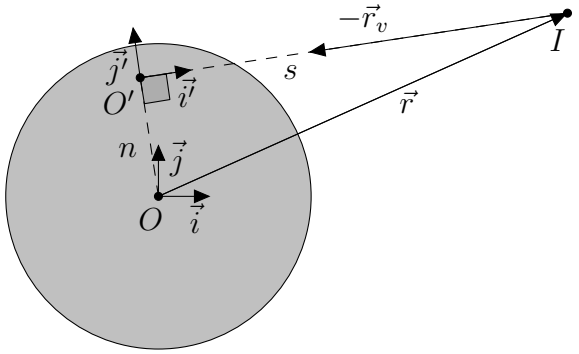

Figure 6: Top view of the two reference frames

Given r_x and r_y, Eqns. (5) imply that the coordinates (x, y) are uniquely determined by the choice of (s, n), as long as $r_v \neq 0$ (with $r_v^2 = r_x^2 + r_y^2$). For any 2-dimensional configuration, the encounter can thus be considered a head-on encounter where s plays the role of r and where a new puck radius, denoted s_p, plays the role of r_p.

Let us now determine the radius s_p of the dimension-reduced encounter, and prove that the absence of NMAC in $(O, \vec{i}, \vec{j})$—characterized by $r^2 > r_p^2$—is equivalent to the absence of NMAC in $(O', \vec{i}', \vec{j}')$—characterized by $s^2 > s_p^2$. Using (5):

$$r_v^2 r^2 = r_v^2(x^2 + y^2) = (xr_x + yr_y)^2 + (xr_y - yr_x)^2 = r_v^2(s^2 + n^2) \ .$$

Since $r_v \neq 0$, this implies $r^2 = s^2 + n^2$. Therefore, $r^2 > r_p^2$ if and only if $s^2 + n^2 > r_p^2$ or equivalently $s^2 > r_p^2 - n^2$. If $r_p^2 - n^2 < 0$, the direction of the vector $\vec{r}_v$ does not intersect the puck, the inequality $s^2 > r_p^2 - n^2$ is trivially true, and the encounter is safe. If $r_p^2 - n^2 \geq 0$, we choose the new puck radius s_p for the dimension-reduced encounter as $s_p = \sqrt{{r_p}^2 - n^2} \geq 0$, and the safety condition in $(O', \vec{i}', \vec{j}')$ becomes $s^2 \geq s_p^2$. When $\theta_v = 180°$, one has $s = r$, $n = 0$ and $s_p = r_p$ as used in the previous sections.

As the encounter evolves in $(O, \vec{i}, \vec{j})$ along $\dot{x} = r_x, \dot{y} = r_y$, its dimension-reduced version evolves in $(O', \vec{i}', \vec{j}')$ along the differential equations $\dot{s} = -r_v, \dot{n} = 0$, obtained by differentiating Eqns. (5) and canceling r_v. The following proposition, proved in KeYmaera, combines both dynamics and shows that the absence of an NMAC of radius r_p in $(O, \vec{i}, \vec{j})$ is equivalent to the absence of an NMAC of radius s_p in $(O', \vec{i}', \vec{j}')$.

Proposition 5 (Horizontal Reduction)**.** *The following dℒ formula is valid*

$$\begin{aligned}\big(xr_x + yr_y = -sr_v \wedge xr_y - yr_x = nr_v \wedge x^2 + y^2 = n^2 + s^2 \wedge r_v^2 = r_x^2 + r_y^2\big) \\ \longrightarrow [\dot{x} = r_x, \dot{y} = r_y, \dot{s} = -r_v, \dot{n} = 0] \left(x^2 + y^2 > r_p^2 \longleftrightarrow s^2 > r_p^2 - n^2\right)\end{aligned} \quad (6)$$

Observe that the (horizontal) NMAC condition in $(O', \vec{i}', \vec{j}')$ only depends on the change of one variable rather than two. The proposition also applies to the special case $r_v = 0$. In this case, the origin O' is no longer defined, Eqns. (5) are trivially true. The variables s and n are constants

Table 2: Summary of the points of the state space at which we examined ACAS X.

	Range r (ft)	Relative speed r_v (ft/s)	Angle θ_v (degrees)	Relative altitude h (ft)	Vertical rates $\dot{h}_0$, $\dot{h}_1$ (ft/s)	Previous advisory
Min value	1,500	100	$180°$	-4,000	-41.67	None
Max value	200,000	2,200	$180°$	4,000	41.67	None
# values	80	10	1	33	13^2	1

($\dot{s} = 0, \dot{n} = 0$), their initial values are only restricted by the condition $n^2 + s^2 = x^2 + y^2$ in the assumption of the proposition, but they are not unique. When the relative position between the ownship and the intruder does not evolve over time, if the intruder is at a safe distance initially, the encounter is safe for all time.

6 Initial Examination of the Safety of ACAS X

In this section, we use Theorem 1 to check the safety of the first advisory that ACAS X would give for the same geometrical configuration of the encounter. More precisely, we focus on Run 12 (July 2014) of the optimized logic tables, a core component of ACAS X. The full policy of the system is built on these lookup tables and incorporates additional components to accommodate various operational scenarios.

We compare the ACAS X table to the (explicit) safe regions where the pilot reacts immediately (Sect. 3). For a given initial state of an encounter, we query the *first* advisory issued by ACAS X and check its safety as identified in Theorem 1. In a real scenario, the ACAS X logic could later strengthen or reverse the first advisory as the encounter evolves. The safety of the first advisory is however critical from an operational prospective as later changes of advisories are undesirable.

Our initial analysis considers a nominal set of discrete states—summarized in Table 2—of the ACAS X MDP model where no advisory has yet been issued. All examined states are head-on encounters: in a sense, they are the most obviously dangerous configurations. For those states, the ACAS X advisories are compared against the safe regions stated in Fig. 3. Overall, $4,461,600$ discrete states were examined, among which $44,306$ states (1.2%) did not meet the conditions of Fig. 3: $11,524$ of these were unresolvable, i.e., the intruder was too close for any advisory to avoid NMAC; while $32,782$ could have been resolved with a different safe advisory that satisfies Theorem 1.

Analyzing these encounters, we identified unexpected behavior in the ACAS X lookup tables. In some cases, the ACAS X advisory seems to *induce* an NMAC (Fig. 7), i.e., if the initial advisory is not strengthened or reverted later, an NMAC will occur. In other cases, the advisory does not seem to have any benefit, that is flying at vertical rates disallowed by the advisory would actually avoid NMAC while not all allowed vertical rates are safe. Notice that these behaviors are not necessarily all deemed undesirable, as ACAS X tends to minimize alerting the pilot unless it has to do so; for some cases, ACAS X will strengthen the advisory later and hence does not issue a disruptive alert immediately. Fig. 7 depicts a typical example where the ACAS X advisory

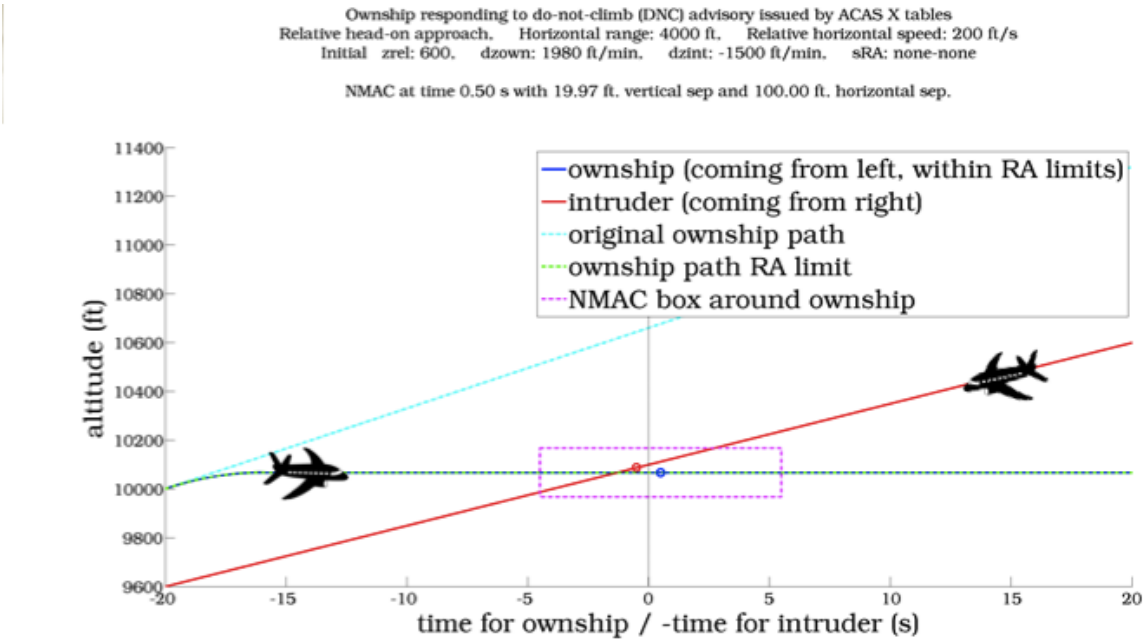

Figure 7: Original ownship path (cyan) and intruder path (red) vs. ownship responding to a do-not-climb (DNC) advisory (green dotted line) issued by the ACAS X tables in starting state: $r = 4,000$ ft, $r_v = 200$ ft/s, $\theta_v = 180°$, $h = 600$ ft, $\dot{h}_0 = 1,980$ ft/min, $\dot{h}_1 = -1,500$ ft/min.

induces an NMAC. The ownship is flying from the left and the intruder from the right. As time counts down, the intruder evolves towards the ownship and an NMAC happens at $t = 0$. The original path of the ownship does not lead to an NMAC. However, ACAS X gives a Do-Not-Climb advisory. If the pilot, following this advisory, decides to stop climbing, its trajectory will cause an NMAC. (Other examples are illustrated in Appendix D.)

The development of the safe regions gave an insight into possible improvements for the ACAS X system. Although we are not analyzing the complete system, nor the subsequent advisories, we automatically pointed out some subregions of the state space worth looking at. Some of those problems were independently identified by the ACAS X team using simulation-based testing, and will be addressed in subsequent revisions of the system. When extended to check contiguous regions of the state space, our approach will have the potential for a complete analysis of the system over all potential encounter configurations, thereby reducing vulnerability to the sampling of encounter scenarios.

7 Related Work

In [KC10] Kochenderfer and Chryssanthacopoulos describe the design of the ACAS X lookup-tables. Their principled approach, based on optimizing an MDP, guarantees the selection of optimal advisories according to a cost model. The state space as well as the dynamics are discretized. Their notion of optimality depends on costs assigned to various events. In contrast, we use continuous

dynamics to assess when the system meets a clear, specific safety property.

Von Essen and Giannakopoulou [vEG14] used probabilistic model-checking to analyze a simpler MDP based on [KC10]. They study the impact of varying the granularity of the discretization used to generate the lookup table, and investigate the probability of several undesirable events occurring. Because they ostensibly analyze an MDP, their work inherits many of the same assumptions of ACAS X, including discretized dynamics. Moreover, their analysis depends heavily on the MDP considered and thus needs to be redone on every version of the actual MDP used for ACAS X. Our approach, however, generates conservative safe regions that assess the safety of an advisory with respect to the assumed dynamics. Therefore, only the comparison part needs to be redone for the future versions of the systems, the safety regions remain unaltered.

Lygeros and Lynch [LL97] use hybrid techniques to formally verify the TCAS conflict resolution algorithms. TCAS is the industrial aircraft collision avoidance system that was developed in the 1970s, a few decades before ACAS X. They consider the simplified case of two aircraft, both TCAS-equipped, with no horizontal acceleration and both pilots complying after some delay; this differs from us in that we only require the ownship to be equipped with ACAS X. More importantly, they assume—rather than prove—that the TCAS system ends up in a state where one aircraft has a climbing advisory and the other a descending advisory. Under those assumptions, they prove (by hand) a lower bound—dependent on initial conditions—on the vertical separation of both aircraft at the point of closest approach. In contrast, we construct universal safe regions that we compare to the ACAS X system's decisions; and we do not need to assume anything about those decisions, but rather determine which decisions are safe. Moreover we prove separation using the puck, whose dimensions are fixed independently from initial conditions; this separation is valid for all times—not just at the point of closest approach—including during the delay and acceleration phases, which significantly complicates our equations. Finally, our proofs are not only by hand but also mechanized in the KeYmaera theorem prover.

Holland *et al.* [HKO14] and Chludzinski [Chl09] simulate large numbers of encounters, including tracks from recorded flight data, to evaluate the performance of ACAS X. These simulations account for high-fidelity details of an encounter, but they only cover a finite set of the continuous state space with no formal guarantees.

Tomlin *et al.* [TPS98], Platzer and Clarke [PC09], Loos *et al.* [LRP13] and more recently Ghorbal *et al.* [GJZ$^+$14] use hybrid systems approaches to design safe horizontal maneuvers for collision avoidance. Dowek *et al.* [DMC05] and Galdino *et al.* [GMA07] describe and verify in the PVS theorem prover a collision avoidance system of their design called KB3D.

Overall, our approach is different from previous complementary work in that:

- unlike [vEG14, KC10], we rely on an independent model from the one used to design ACAS X;
- unlike [DMC05, GMA07, LRP13, PC09, TPS98, GJZ$^+$14] we analyze an independent industrial system and not a safe-by-design system;
- unlike [DMC05, vEG14, GMA07] our analysis uses realistic, continuous dynamics;
- unlike [Chl09, HKO14, KEKG08, LL97, TPS98], we provide formal, mechanized proofs for the correctness of our model;

- unlike [LL97] who verify TCAS, we verify the ACAS X conflict resolution algorithms.

8 Conclusion and Future Work

We developed a general strategy for analyzing the safety of complicated, real-world collision avoidance systems, and applied it to ACAS X, currently under development. This strategy identifies conditions on resolution advisories for each geometric configuration that are proved to always keep the aircraft clear of NMAC as long as the considered assumptions hold. We identified discrete states where ACAS X is provably safe, and fed back others showing unexpected behaviors to the ACAS X development team. The identified safe regions are independent from the actual version of ACAS X and could be used to assess the safety of future versions. In future, we plan to extend our hybrid model to account for curved trajectories of both aircraft as well as vertical acceleration of the intruder.

Acknowledgments

The authors would like to warmly thank Stefan Mitsch and Jan-David Quesel for their availability and support of the KeYmaera tool. The authors would also like to thank Jeff Brush, Jessica Holland, Robert Klaus, Barbara Kobzik-Juul, Mykel Kochenderfer, Ted Londner, Sarah Loos, Ed Morehouse, Wes Olson, Michael Owen, Joshua Silbermann, Neal Suchy, the Logical Systems Lab at Carnegie Mellon University, and the ACAS X development team for interesting discussions and remarks.

References

[Chl09] Barbara J. Chludzinski. Evaluation of TCAS II version 7.1 using the FAA fast-time encounter generator model. Technical Report ATC-346, Massachusetts Institute of Technology Lincoln Laboratory, April 2009.

[DMC05] Gilles Dowek, César Muñoz, and Víctor Carreño. Provably safe coordinated strategy for distributed conflict resolution. In *AIAA Guidance Navigation, and Control Conference and Exhibit*, San Francisco, California, 2005. doi:10.2514/6.2005-6047.

[Fed11] Federal Aviation Administration. Introduction to TCAS II. Version 7.1, February 2011.

[Fed14a] Federal Aviation Administration Traffic Alert & Collision Avoidance System (TCAS) Program Office (PO). Algorithm design description for the surveillance and tracking module of ACAS X. Run12, July 2014.

[Fed14b] Federal Aviation Administration Traffic Alert & Collision Avoidance System (TCAS) Program Office (PO). Algorithm design description for the threat resolution module of ACAS X. Version 3 Rev. 1, May 2014.

[GJZ+14] Khalil Ghorbal, Jean-Baptiste Jeannin, Erik W. Zawadzki, André Platzer, Geoffrey J. Gordon, and Peter Capell. Hybrid theorem proving of aerospace systems: Applications and challenges. *Journal of Aerospace Information Systems*, 2014. doi:10.2514/1.I010178.

[GMA07] André Galdino, César Muñoz, and Mauricio Ayala. Formal verification of an optimal air traffic conflict resolution and recovery algorithm. In D. Leivant and R. de Queiroz, editors, *Proceedings of the 14th Workshop on Logic, Language, Information and Computation*, volume 4576 of *LNCS*, pages 177–188, Rio de Janeiro, Brazil, July 2007. Springer-Verlag. doi:10.1007/978-3-540-73445-1_13.

[HKO14] Jessica E. Holland, Mykel J. Kochenderfer, and Wesley A. Olson. Optimizing the next generation collision avoidance system for safe, suitable, and acceptable operational performance. *Air Traffic Control Quarterly*, 2014. doi:10.1.1.352.4336.

[KC10] Mykel J. Kochenderfer and James P. Chryssanthacopoulos. Robust airborne collision avoidance through dynamic programming. Technical Report ATC-371, Massachusetts Institute of Technology Lincoln Laboratory, January 2010.

[KEKG08] M. J. Kochenderfer, L. P. Espindle, J. K. Kuchar, and J. D. Griffith. Correlated encounter model for cooperative aircraft in the national airspace system version 1.0. Technical Report ATC-344, Massachusetts Institute of Technology Lincoln Laboratory, October 2008.

[KHC12] Mykel J. Kochenderfer, Jessica E. Holland, and James P. Chryssanthacopoulos. Next generation airborne collision avoidance system. *Lincoln Laboratory Journal*, 19(1):17–33, 2012.

[KM13] Mykel J. Kochenderfer and Nicholas Monath. Compression of optimal value functions for Markov decision processes. In *Data Compression Conference*, Snowbird, Utah, 2013. doi:10.1109/DCC.2013.81.

[LL97] John Lygeros and Nancy Lynch. On the formal verification of the TCAS conflict resolution algorithms. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 2, pages 1829–1834. IEEE, 1997. doi:10.1109/CDC.1997.657846.

[LRP13] Sarah M. Loos, David W. Renshaw, and André Platzer. Formal verification of distributed aircraft controllers. In *HSCC*, pages 125–130. ACM, 2013. doi:10.1145/2461328.2461350.

[PC09] André Platzer and Edmund M. Clarke. Formal verification of curved flight collision avoidance maneuvers: A case study. In *FM*, volume 5850 of *LNCS*, pages 547–562. Springer, 2009. doi:10.1007/978-3-642-05089-3_35.

[Pla08] André Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reas.*, 41(2):143–189, 2008. doi:10.1007/s10817-008-9103-8.

[Pla10] André Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer, 2010. doi:10.1007/978-3-642-14509-4.

[Pla12] André Platzer. Logics of dynamical systems. In *LICS*, pages 13–24. IEEE, 2012. doi:10.1109/LICS.2012.13.

[PQ08] André Platzer and Jan-David Quesel. KeYmaera: A hybrid theorem prover for hybrid systems. In *IJCAR*, volume 5195 of *LNCS*, pages 171–178. Springer, 2008. doi:10.1007/978-3-540-71070-7_15.

[TPS98] Claire Tomlin, George J Pappas, and Shankar Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *Automatic Control, IEEE Transactions on*, 43(4):509–521, 1998. doi:10.1109/9.664154.

[vEG14] Christian von Essen and Dimitra Giannakopoulou. Analyzing the next generation airborne collision avoidance system. In *TACAS*, volume 8413 of *LNCS*, pages 620–635. Springer, 2014. doi:10.1007/978-3-642-54862-8_54.

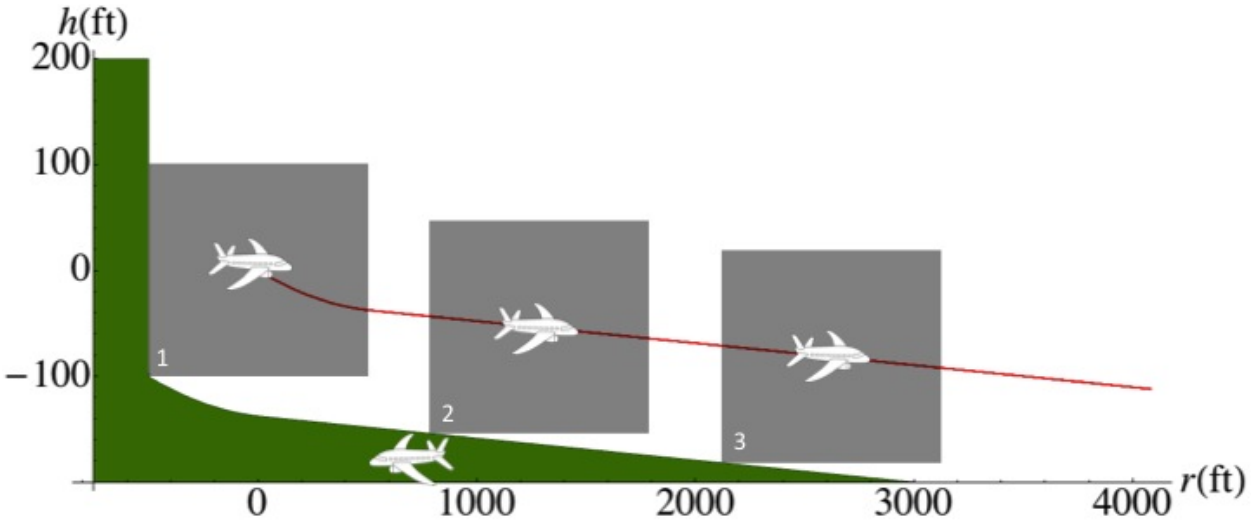

Figure 8: Trajectory of the ownship (red) and safe region for the intruder (green)

A KeYmaera statistics

This appendix shows some statistics on our different KeYmaera proofs. The proof are available at http://www.cs.cmu.edu/~jeannin/acasx.zip.

	Time (s)	Memory (MB)	Steps	Dimensions
Implicit regions, no delay (Theorem 1)	76.2	104.8	310	10
Explicit regions, no delay (Lemma 2)	68.0	79.1	749	13
Implicit regions, with delay (Theorem 3)	214.5	347.1	1835	16
Explicit regions, with delay (Lemma 4)	181.4	127.2	2118	16
Horizontal reduction (Prop. 5)	2.3	40.3	17	8

B Safe Region for a Delayed Pilot Response

In this appendix we give more intuition for the explicit formulation of the safe regions presented in Fig. 3.

Second case: if $w = +1$**,** $r_v > 0$ **and** $\dot{h}_0 < \dot{h}_f < 0$ This case is represented in Fig. 8, and the nominal trajectory $\mathcal{N}$ is still given by Eq. (2)(a) and Eq. (2)(b). In Fig. 8, the green safe region's boundary is drawn by the bottom left hand corner of the puck, and can be characterized by the following set of equations:

- as in the first case, all the positions left of the initial position of the puck ($r < -r_p$) are in the safe region;

- then the boundary follows the bottom left hand corner of the puck as it goes down the parabola of Eq. (2)(a); therefore for $-r_p \leq r \leq -r_p + \frac{r_v(\dot{h}_f - \dot{h}_0)}{a_r}$, the position (r, h) is safe if and only if $h < \frac{a_r}{2r_v^2}(r + r_p)^2 + \frac{\dot{h}_0}{r_v}(r + r_p) - h_p$;
- finally the boundary follows the bottom left hand corner of the puck as it is going down the straight line of Eq. (2)(b); therefore for $-r_p + \frac{r_v(\dot{h}_f - \dot{h}_0)}{a_r} < r$, the position (r, h) is in the safe region if and only if $h < \frac{\dot{h}_f}{r_v}(r + r_p) - \frac{(\dot{h}_f - \dot{h}_0)^2}{2a} - h_p$.

C Safe Region for an Immediate Pilot Response

In this appendix we give some intuition for the explicit formulation of the delayed safe regions presented in Fig. 5.

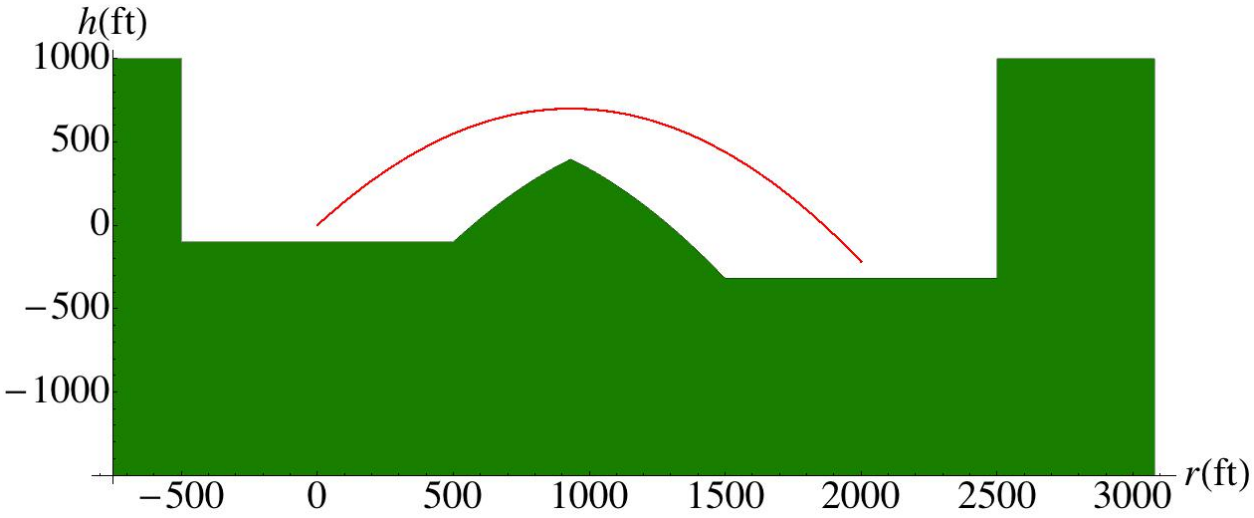

Figure 9: The most common configuration of the delay cusp.

As in Sect. 3.4, let us derive the explicit formulation of the safe zone during the delay. Again, let us assume that $w = +1$ and $d \geq 0$, then generalize. During the delay, the nominal trajectory is given by $(r_t, h_t) = \left(r_v t, -\frac{a_d}{2}t^2 + \dot{h}_0 t\right)$, and looking at Fig. 9, the safe region describes a *cusp* and its boundary can be characterized by a set of equations:

- all the positions left of the initial position of the puck ($r < -r_p$) are in the safe region;
- a position (r, h) is in the safe region only if it is to the right, left, or under the puck at time 0; therefore for $-r_p \leq r \leq r_p$, a position (r, h) is in the safe region only if $h < -h_p$;
- a position (r, h) is in the safe region only if it is to the right, left, or under the trajectory drawn by the bottom right hand corner of the puck during the climbing phase of the delay parabola; therefore for $r_p < r \leq r_d + r_p$, a position (r, h) is in the safe region only if $r_v^2 h < -\frac{a_d}{2}(r - r_p)^2 + r_v \dot{h}_0 (r - r_p) - r_v^2 h_p$;

- a position (r, h) is in the safe region only if it is to the right, left, or under the trajectory drawn by the bottom left hand corner of the puck during the descending phase of the delay parabola; for $-r_p < r \leq r_d - r_p$, a position (r, h) is in the safe region only if $r_v^2 h < -\frac{a_d}{2}(r + r_p)^2 + r_v \dot{h}_0 (r + r_p) - r_v^2 h_p$;
- a position (r, h) is in the safe region only if it is to the right, left, or under the puck at time d; therefore for $r_d - r_p \leq r \leq r_d + r_p$, a position (r, h) is in the safe region only if $h - h_d < -h_p$.

We can again generalize those equations to the cases where $w = -1$ and $d < 0$. We put together the different explicit boundaries according to their respective cases, and link it to the formula C_{expl} of Fig. 3 translated by horizontal distance r_d, height h_d and starting at vertical speed $\dot{h}_d$. Doing this we realize that the last condition of $h - h_d < -h_p$ when $r_d - r_p \leq r \leq r_d + r_p$ is unnecessary because already implied by $C_{\mathsf{expl}}(r - r_d, h - h_d, \dot{h}_d)$. We thus create formula D_{expl} of Fig. 5. Finally, we formally prove using KeYmaera that this explicit formulation of the safe region is equivalent to its implicit counterpart (Lemma 4).

Note that, contrarily to the safe region for an immediate reaction of the pilot, the different cases of the safe region for a delayed reaction of the pilot overlap. This is due to a number of possible configurations, happening especially when r_v is small. Examples are shown in Fig. 10 and 11.

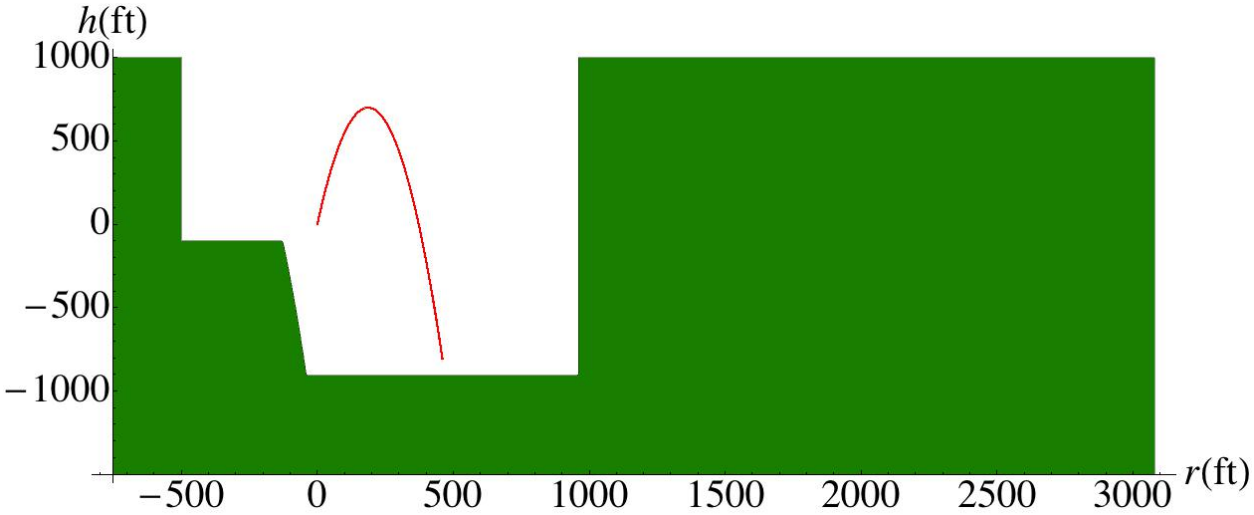

Figure 10: An example of a degenerate cusp, caused by a very small relative velocity. Final height is below the starting point.

D Examples of Identified ACAS X Behavior

Fig. 7 from Sect. 6 illustrated a geometry where the comparison of our safety theorem to ACAS X identified behavior that may induce an NMAC.

Fig. 12 depicts a different case, where the benefit of the advisory issued by the ACAS X lookup tables is unclear and may reduce safety although it does not directly induce an NMAC. In this

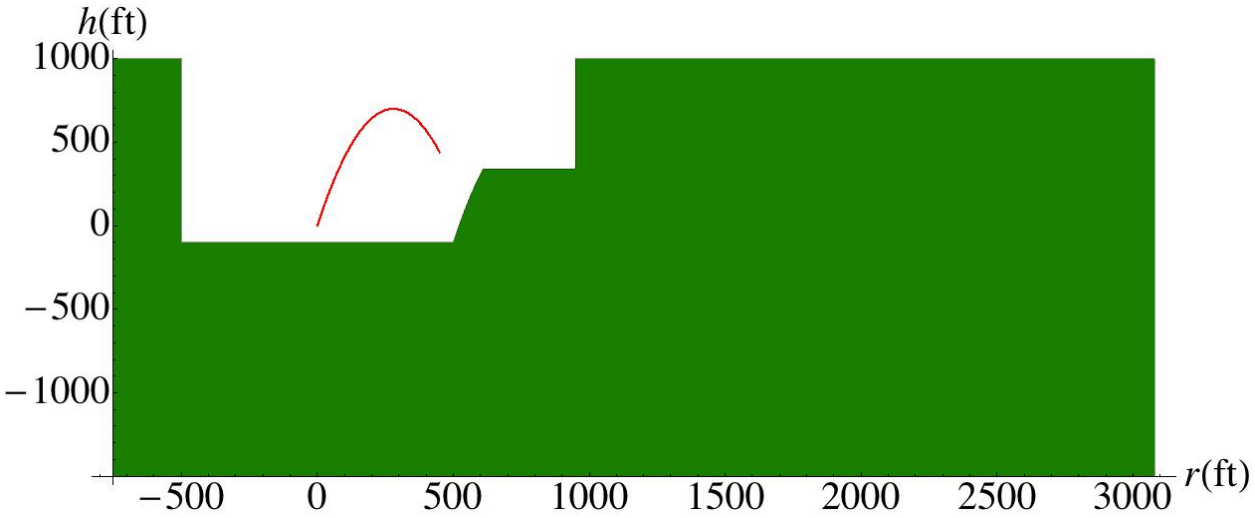

Figure 11: An example of a degenerate cusp, caused by a very small relative velocity. Final height is above the starting point.

scenario, the lookup tables issue an alert not to descend at more than 2,000 ft/min (DND2000). However, the ownship can still fly within the limits of this advisory and cause an NMAC, as illustrated in the figure where the ownship climbs at 1,180 ft/min. If the ownship were restricted not to climb (DNC) an NMAC would not occur under straight-line assumptions. Thus, an NMAC would also not occur if the ownship were restricted to rates lower than -2,000 ft/min, the range of rates being disallowed.

Fig. 13 depicts a case where the advice issued by the ACAS X tables did not meet the conditions of our safety theorem, but it may be desirable based on the goals of ACAS X. ACAS X issues a do-not-climb (DNC) approximately 28 seconds from potential collision. This advice is less disruptive to the pilot and flight path than something stronger like descend-1500 (DES1500). A stronger advisory may not be necessary because the intruder may reduce its rate of descent. At the same time, ACAS X may be able to effectively issue a stronger advisory in a few seconds if necessary.

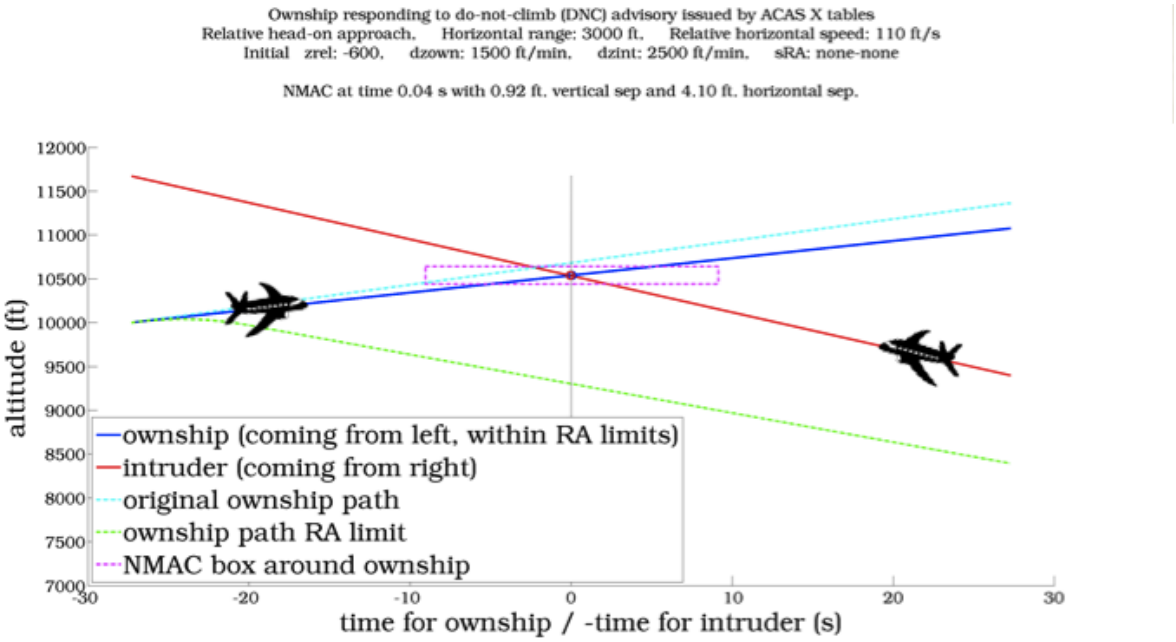

Figure 12: Ownship flying within the limits of a do-not-descend-2000 (DND2000) advisory issued by the ACAS X tables in starting state: $r = 3,000$ ft, $r_v = 110$ ft/s, $\theta_v = 180°$, $h = -600$ ft, $\dot{h}_0 = 1,500$ ft/min, $\dot{h}_1 = 2,500$ ft/min, no previous advisory.

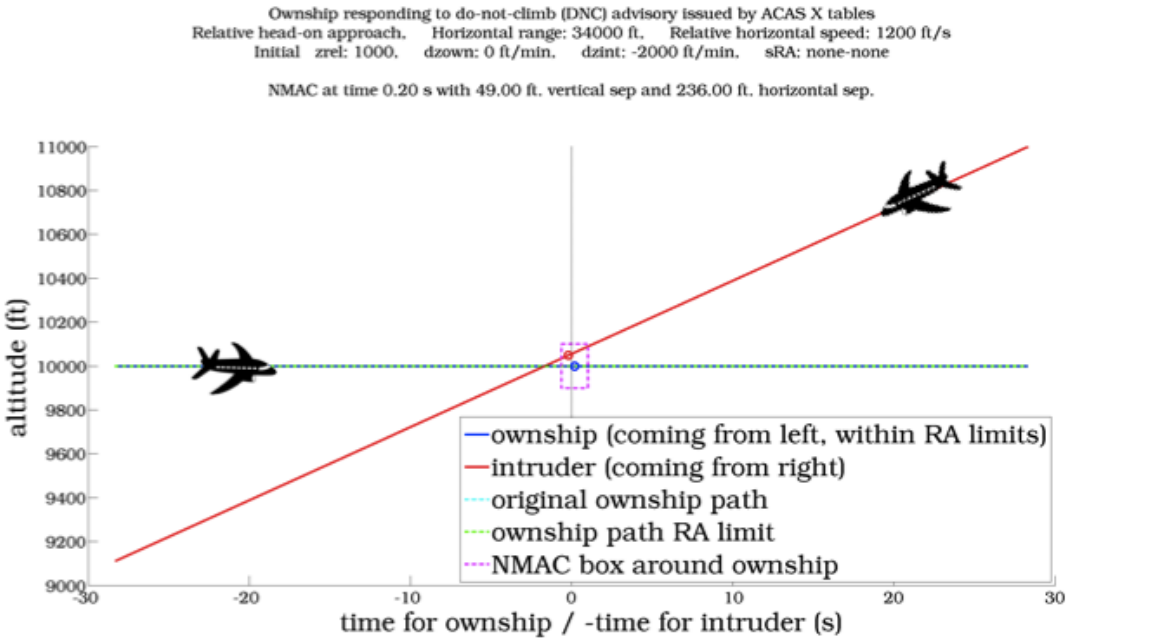

Figure 13: Ownship complying with a do-not-climb (DNC) advisory issued by the ACAS X tables in starting state: $r = 34,000$ ft, $r_v = 1,200$ ft/s, $\theta_v = 180°$, $h = 1,000$ ft, $\dot{h}_0 = 0$ ft/min, $\dot{h}_1 = -2,000$ ft/min, no previous advisory.