Noname manuscript No.
(will be inserted by the editor)

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

Jean-Baptiste Jeannin · Khalil Ghorbal · Yanni Kouskoulas · Aurora Schmidt · Ryan Gardner · Stefan Mitsch · André Platzer

the date of receipt and acceptance should be inserted later

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 consider subsequent advisories and show how to adapt our formal verification to take them into account. We examine 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 systems proving approaches are helping to 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.

Jean-Baptiste Jeannin*
Samsung Research America, Mountain View, CA, USA

Khalil Ghorbal*
Inria, Rennes, France

Yanni Kouskoulas
The Johns Hopkins University Applied Physics Laboratory, USA

Aurora Schmidt
The Johns Hopkins University Applied Physics Laboratory, USA

Ryan Gardner
The Johns Hopkins University Applied Physics Laboratory, USA

Stefan Mitsch*
Johannes Kepler University, Linz, Austria

André Platzer
Carnegie Mellon University, Pittsburgh, PA, USA

* This work was performed at Carnegie Mellon University

1 Introduction

With growing air traffic, the airspace becomes 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 had great success, and prevented many mid-air collisions over the years. Some accidents still happened; for example, a collision over Überlingen in 2002 occurred due to conflicting advice between TCAS and air traffic control. Airspace management will evolve significantly over the next decade with the introduction of the next-generation air traffic management system; this will create new requirements for collision avoidance. 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 [5,12,16].

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 aircraft called the *intruder* that is at risk of colliding with the ownship. ACAS X is designed to avoid *Near Mid-Air Collisions* (*NMACs*), situations where two aircraft come within $r_p = 500\,\text{ft}$ horizontally and $h_p = 100\,\text{ft}$ vertically [16] of each other. The NMAC definition describes a volume centered around the ownship, shaped like a hockey *puck* of radius r_p and half-height h_p.

In order to be accepted by pilots, and thus operationally suitable, ACAS X needs to strike a balance between giving advice to help pilots avoid collisions but also minimizing interruptions. These goals drive the design in opposite directions, and cannot both be perfectly met in the presence of unknown pilot behavior. As part of the ACAS X development process, this work focuses on precisely characterizing

Table 1 ACAS X advisories and their modeling variables

		ACAS X Specification [14]				Our model	
		Vertical Rate Range		Strength	Delay	Sign	Advisory
Advisory	Description	Min (ft/min)	Max (ft/min)	a_{lo}	δ (s)	w	v_{lo} (ft/min)
DNC2000	Do Not Climb at more than 2,000 ft/min	$-\infty$	$+2000$	$g/4$	5	-1	$+2000$
DND2000	Do Not Descend at more than 2,000 ft/min	-2000	$+\infty$	$g/4$	5	$+1$	-2000
DNC1000	Do Not Climb at more than 1,000 ft/min	$-\infty$	$+1000$	$g/4$	5	-1	$+1000$
DND1000	Do Not Descend at more than 1,000 ft/min	-1000	$+\infty$	$g/4$	5	$+1$	-1000
DNC500	Do Not Climb at more than 500 ft/min	$-\infty$	$+500$	$g/4$	5	-1	$+500$
DND500	Do Not Descend at more than 500 ft/min	-500	$+\infty$	$g/4$	5	$+1$	-500
DNC	Do Not Climb	$-\infty$	0	$g/4$	5	-1	0
DND	Do Not Descend	0	$+\infty$	$g/4$	5	$+1$	0
MDES	Maintain at least current Descent rate	$-\infty$	current	$g/4$	5	-1	current
MCL	Maintain at least current Climb rate	current	$+\infty$	$g/4$	5	$+1$	current
DES1500	Descend at at least 1,500 ft/min	$-\infty$	-1500	$g/4$	5	-1	-1500
CL1500	Climb at at least 1,500 ft/min	$+1500$	$+\infty$	$g/4$	5	$+1$	$+1500$
SDES1500	Strengthen Descent to at least 1,500 ft/min	$-\infty$	-1500	$g/3$	3	-1	-1500
SCL1500	Strengthen Climb to at least 1,500 ft/min	$+1500$	$+\infty$	$g/3$	3	$+1$	$+1500$
SDES2500	Strengthen Descent to at least 2,500 ft/min	$-\infty$	-2500	$g/3$	3	-1	-2500
SCL2500	Strengthen Climb to at least 2,500 ft/min	$+2500$	$+\infty$	$g/3$	3	$+1$	$+2500$
COC	Clear of Conflict	$-\infty$	$+\infty$	Not applicable			
MTLO	Multi-Threat Level-Off	Not applicable					

the circumstances in which ACAS X gives safe advice, and where safety is traded off for operational suitability, helping to identify modifications that improve its safety and performance.

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 intruders [6]. It computes its estimate of the best pilot action by linearly interpolating a precomputed *table* of scores for actions, and, if appropriate, issuing an *advisory* to avoid potential collisions [7] through a visual display 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, and never request any horizontal maneuvering. Table 1 shows the advisories ACAS X can issue. For example, Do-Not-Climb (DNC) requests the pilot to not climb, and Climb-1500 (CL1500) requests the pilot to climb at more than 1500 ft/min. The current version of ACAS X can issue a total of 16 different advisories plus Clear-of-Conflict (COC), which indicates that no action is necessary, and Multi-Threat-Level-Off (MTLO), which is used in the case of multiple intruders. To comply with an advisory, the pilot must adjust her vertical rate to fall within the advised vertical rate range. Based on previous research [14], the pilot is assumed to do so using a vertical acceleration of strength at least a_{lo} starting after a delay of at most δ after the advisory has been announced by ACAS X.

At the heart of ACAS X is a table whose domain describes the current state of an encounter, and whose range is a set of scores for each possible action [14, 17]. The table is obtained from a Markov Decision Process (MDP) approximating the dynamics of the system in a discretization of the state-space, and optimized using dynamic programming to maximize the expected value of events over all future paths for each action [14]. Near Mid-Air Collision events, for example, are associated with large negative values and issuing an advisory is associated with a small negative value. The policy is to choose the action with the highest expected 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 and dynamic conditions in which it finds itself.

1.2 Identifying Formally Verified Safe Regions

Since ACAS X involves both *discrete* advisories to the pilot and *continuous* dynamics of aircraft, it is natural to formally verify it using hybrid systems. However the complexity of ACAS X, which uses at its core 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 we prove formally that a particular advisory, if followed, prevents all possible NMACs from any aircraft position and velocity in that safe region. Then it *compares* these regions to the configurations where the ACAS X table returns this same advisory. Moreover our safe regions are *symbolic* in their parameters, and can thus be easily adapted to new parameters or new versions of ACAS X.

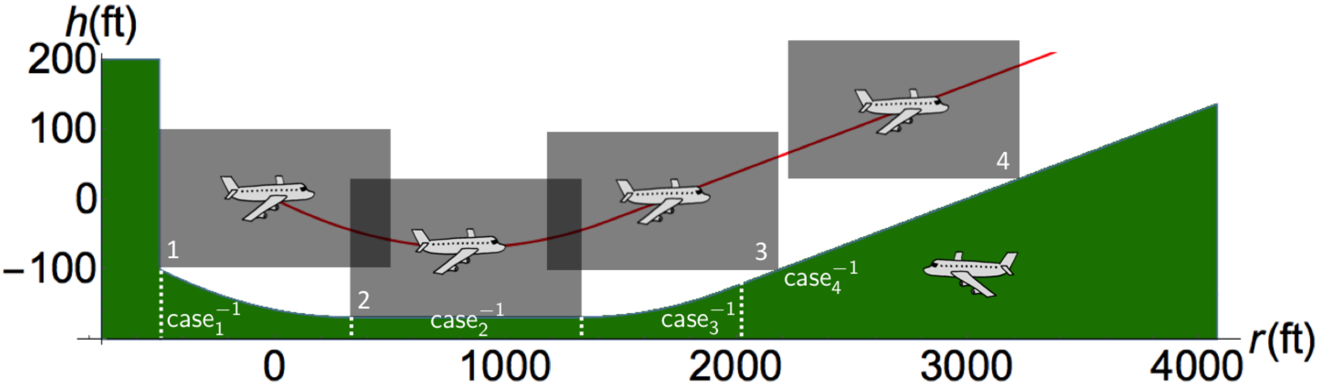


Fig. 1 Nominal trajectory of the ownship (red) and safe region for the intruder (green), immediate response

Going beyond the results of previous work [13], this paper devises and formally proves safety regions for advisories that can be corrected later on. In that context, an advisory need not be safe on its own to be considered acceptable, but the system needs to be able to correct it with a *subsequent* advisory. This is particularly useful to assess the safety of preventive advisories, and leads to the discovery of very relevant unexpected behaviors of the system.

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 precise hybrid systems 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 first formally deriving safe regions and 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. 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, consider more liberal safe regions to tolerate advisories that are only safe if followed up by suitable subsequent advisories in Sect. 5, and extend the results to a three-dimensional model in Sect. 6. Relationships and extensions are discussed in Sect. 7. In Sect. 8, 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. This progressive development of the models is helpful to support the intuition behind their geometric constructions and makes it possible to focus on one aspect at a time.

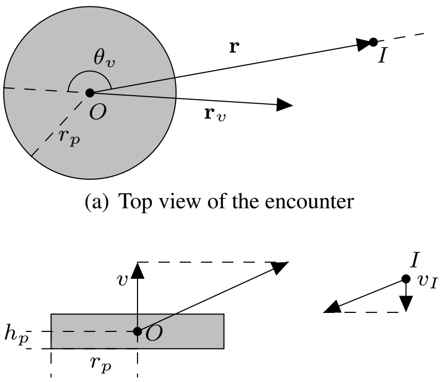
2 Overview of the ACAS X Modelling Approach

For the sake of intuition of how a safe region of an advisory for an aircraft could be constructed, 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; for each point there exists a trajectory that results in an NMAC. If the intruder is outside this set, i.e., in the safe region, an NMAC cannot occur in the model.

Fig. 1 depicts an example of a head-on encounter and its associated safe region for the advisory CL1500, projected in a vertical plane with both aircraft. It is plotted in a *coordinate system fixed to the intruder* and centered at the initial position of the ownship. The ownship, surrounded by the puck, starts at position 1 and traces out a trajectory following the red curve. It first accelerates vertically with $g/4$ until reaching the desired vertical velocity of $+1500$ ft/min at position 3. It then continues to climb at $+1500$ ft/min, respecting the specification of Table 1. The green safe-region indicates starting points in the state space for which the aircraft will remain safe for the duration of the encounter when following the CL1500 advisory in any way. Note that no safe region exists above the trajectory since the ownship is allowed to accelerate vertically at greater than $g/4$ or climb more than $+1500$ ft/min, in accordance with Table 1.

2.1 Model of Dynamics

Let us consider an encounter between two aircraft — ownship O and intruder I, as portrayed in Fig. 2. Following the notation of the ACAS X community [14], let $r = \|\mathbf{r}\|$ be the



(a) Top view of the encounter

(b) Side view of the encounter

Fig. 2 An encounter between ownship O and intruder I, with NMAC puck in gray of horizontal radius r_p and vertical radius h_p

horizontal distance between the aircraft (called *range*) and h the height of the intruder relative to the ownship. We assume that the relative horizontal velocity $\mathbf{r_v}$ of the intruder with respect to the ownship is constant throughout the encounter. I.e., from a top view, the planes follow straight-line trajectories. Let θ_v be the non-directed angle between $\mathbf{r_v}$ and the line segment $\mathbf{r}$. In the vertical dimension, we assume that the ownship's vertical velocity v can vary at any moment, while the intruder's vertical velocity v_I is fixed throughout the encounter. Moreover, we assume that the magnitude of the vertical acceleration of the ownship cannot exceed a^{d} in absolute value. These assumptions restrict the potential trajectories of the aircraft and, while making our modeling and verification possible, they weaken the overall safety guarantees.

Our analysis considers all these as symbolic parameters and is, thus, valid for any value they might have. For a typical encounter, r varies between 0 nmi and 7 nmi,[1] h between $-4{,}000\,\mathrm{ft}$ and $4{,}000\,\mathrm{ft}$, $r_v = \|\mathbf{r_v}\|$ between 0 kts and 1,000 kts, and v and v_I between $-5{,}000\,\mathrm{ft/min}$ and $+5{,}000\,\mathrm{ft/min}$. The acceleration a^{d} is usually $g/2$, where g is Earth's gravitational acceleration. The NMAC *puck* of ACAS X has radius $r_p = 500\,\mathrm{ft}$ and half-height $h_p = 100\,\mathrm{ft}$.

2.2 Model of Advisories

ACAS X prevents NMACs by giving advisories to the ownship's pilot. Every advisory, except Clear-of-Conflict (COC), has a vertical rate range of the form $(-\infty, v_{\mathsf{lo}}]$ or $[v_{\mathsf{lo}}, +\infty)$ for some vertical rate v_{lo} (Table 1), which we call the *target vertical velocity*. We model any advisory by its corresponding target vertical velocity v_{lo}, and a binary variable w for its orientation, whose value is -1 if the vertical rate range of the advisory is $(-\infty, v_{\mathsf{lo}}]$ and $+1$ if it is $[v_{\mathsf{lo}}, +\infty)$. This symbolic encoding can represent many advisories and is robust to changes in the ACAS X advisory set. The only advisory that this symbolic encoding cannot handle is the recently-added Multi-Threat Level-Off (MTLO) advisory, only relevant in the presence of multiple intruders.

Following the ACAS X design [14], we assume that the ownship pilot complies with each advisory within δ seconds, and that she accelerates with acceleration of magnitude at least the lower bound a_{lo} to reach the target vertical velocity.

3 Safe Region for an Immediate Pilot Response

We present in this section a simplified version of the dynamics from Sect. 2.1. We give a hybrid model for this simplified system and prove its safety. The new assumptions will be relaxed in later sections to achieve the safety verification of the full model of Sect. 2.1.

3.1 Model

In this section, we assume that the ownship and intruder are flying head-on ($\theta_v = 180°$). We assume that the pilot reacts immediately to any advisory ($\delta = 0\,\mathrm{s}$), and that the advisory COC is not allowed. These assumptions will be relaxed in Sect. 4 and Sect. 6. The model in this section permits updates to the resolution advisory but, unlike in Sect. 5, each advisory issued has to be safe, i.e., it has to prevent any NMAC at any future time, even if followed forever. 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. Both cases could require an advisory. Since 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 (gray in Fig. 1).

3.2 Differential Dynamic Logic and KeYmaera X

To model our system, we use Differential Dynamic Logic d$\mathcal{L}$ [20,21,22,23], a logic for reasoning about hybrid systems. The logic d$\mathcal{L}$ supports discrete assignments, control structures, and execution of differential equations. It is implemented in the theorem prover KeYmaera X [9], that we use to verify our safe regions with respect to our models. All the KeYmaera X models and proofs of this paper can be found online.[2]

[1] 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).

[2] `http://www.ls.cs.cmu.edu/pub/AcasX-long.zip`

The d$\mathcal{L}$ formula for the ACAS X model that we use in this section is given in Eq. (1) below. We use the notation L^{-1}_{impl} for the safe region: the letter L stands for *lower* bound (for $w = 1$; it is an upper bound for $w = -1$); the subscript impl stands for *implicit* safe region, as described in Sect. 3.3; and the superscript -1 indicates that the region is safe for unbounded time; the rationale behind its use will become clear in Sect. 5.2.

```
1 $r_p \geq 0 \land h_p > 0 \land r_v \geq 0 \land a_{\mathsf{lo}} > 0 \land$
2 $(w = -1 \lor w = 1) \land L^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}}) \to$
3 $[($ $($ $?\mathsf{true} \cup$
4       $(w := -1 \cup w := 1); v_{\mathsf{lo}} := *;$
5       $?L^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}}); \mathsf{advisory} := (w, v_{\mathsf{lo}})$ $);$
6    $a := *;$
7    $\{r' = -r_v, h' = -v, v' = a \;\&\; wv \geq wv_{\mathsf{lo}} \lor wa \geq a_{\mathsf{lo}}\}$
8 $)^*]$ $(|r| > r_p \lor |h| > h_p)$
                                                        (1)
```

This formula of the form $p \to [\alpha]q$ says all executions of hybrid 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 $L^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})$ (which we identify below) that forces the intruder to be in a safe region for an initial advisory (w, v_{lo}). We cannot guarantee safety if the intruder starts initially in an unsafe region. The postcondition encodes absence of NMAC using absolute values. Lines 3–5 express the action of the ACAS X system. The nondeterministic choice operator $\cup$ in Line 3 expresses that the system can either continue with the same advisory by doing nothing—just testing the trivial condition $?\mathsf{true}$—this ensures it always has a valid choice and cannot get stuck. Otherwise it can choose a new $\mathsf{advisory}$ (w, v_{lo}) in Line 4 that passes the safety condition $L^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})$ in Line 5—$\mathsf{advisory}$ represents the next message to the pilot. Line 6 expresses the action of the ownship pilot, who can nondeterministically choose an arbitrary acceleration ($a := *$). The ownship and intruder aircraft then follow the continuous dynamics in Line 7. The evolution of the variables r, h and v is expressed by a differential equation, and requires (using the operator $\&$) that the ownship evolves towards its target vertical velocity v_{lo} at acceleration a_{lo} (condition $wa \geq a_{\mathsf{lo}}$), unless it has already reached vertical velocity v_{lo} (condition $wv \geq wv_{\mathsf{lo}}$). Finally, the star * on Line 8 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

In this section, we identify what formula can be used as safe region $L^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})$ to prove Eq. (1). As in Sect. 2, we use a coordinate system fixed to the intruder and with its origin at the initial position of the ownship (see Fig. 1).

First case: if $w = +1$ *and* $v_{\mathsf{lo}} \geq v$. Fig. 1 shows, in red, a possible trajectory of an ownship following exactly the requirements of ACAS X. This *nominal* trajectory of the ownship is denoted by $\mathcal{N}$ and merely represents one of the infinitely many possible scenarios to consider. The pilot reacts immediately, and the ownship starts accelerating vertically with acceleration a_{lo} until it reaches the target vertical velocity v_{lo}—describing a parabola—then climbs at vertical velocity v_{lo} along a straight line. Horizontally, the relative velocity r_v remains constant. Integrating the differential equations in Eq. (1) Line 7, the ownship position (r_n, h_n) at time t along $\mathcal{N}$ is given by:

$$(r_n, h_n) = \begin{cases} \left(r_v t\,,\ \dfrac{a_{\mathsf{lo}}}{2}t^2 + vt\right) & \text{if } 0 \leq t < \dfrac{v_{\mathsf{lo}} - v}{a_{\mathsf{lo}}} \ (a) \\ \left(r_v t\,,\ v_{\mathsf{lo}} t - \dfrac{(v_{\mathsf{lo}} - v)^2}{2a_r}\right) & \text{if } \dfrac{v_{\mathsf{lo}} - v}{a_{\mathsf{lo}}} \leq t \ (b) \end{cases} \tag{2}$$

Recall that in the ACAS X specification, the ownship moves vertically with acceleration of *at least* a_{lo}, then continues with vertical velocity of *at least* v_{lo}. Therefore all possible future positions of the ownship will turn out to be *above* the red nominal trajectory. Hence, an intruder is safe if its position is always either to the side of or under any puck centered on a point in $\mathcal{N}$, that is:

$$\forall t. \forall r_n. \forall h_n. \\ \big((r_n, h_n) \in \mathcal{N} \to |r - r_n| > r_p \lor h - h_n < -h_p\big) \tag{3}$$

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 by quantifiers with respect to the nominal trajectory from Eq. (2).

Generalization. The reasoning above is generalized to the case where the target vertical velocity is exceeded ($v_{\mathsf{lo}} < v$) —which happens after the parabola part of the nominal trajectory— and symmetrically to the case of descend-type advisories ($w = -1$).

Eq. (1) gives the pilot ample flexibility in how to respond to a resolution advisory and gives ACAS X full flexibility to choose any advisories respecting $L^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})$. In particular, the pilot may not follow the nominal trajectory $\mathcal{N}$ that we used to construct this region. Nevertheless, we prove that the safe regions identified this way respect safety property Eq. (1). The implicit formulation $L^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})$ of the safe region is specified in Fig. 3 and used in Theorem 1, which is verified to be safe in KeYmaera X:

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 followed obey formula L^{-1}_{impl} there will be no NMAC.*

Implicit formulation

$$
\begin{aligned}
A(t, h_n, v, w, v_{\mathsf{lo}}) \equiv\ & \left(0 \le t < \frac{\max(0, w(v_{\mathsf{lo}} - v))}{a_{\mathsf{lo}}} \land h_n = \frac{w a_{\mathsf{lo}}}{2} t^2 + vt \right) \\
\lor & \left(t \ge \frac{\max(0, w(v_{\mathsf{lo}} - v))}{a_{\mathsf{lo}}} \land h_n = v_{\mathsf{lo}} t - \frac{w \max(0, w(v_{\mathsf{lo}} - v))^2}{2 a_{\mathsf{lo}}} \right) \\
L_{\mathsf{impl}}^{-1}(r, h, v, w, v_{\mathsf{lo}}) \equiv\ & \forall t. \forall r_n. \forall h_n. \Big(r_n = r_v t \land A(t, h_n, v, w, v_{\mathsf{lo}}) \to (|r - r_n| > r_p \lor w(h - h_n) < -h_p) \Big)
\end{aligned}
$$

Explicit formulation

$$
\begin{aligned}
\mathsf{case}_1^{-1}(r, v, w, v_{\mathsf{lo}}) &\equiv -r_p \le r < -r_p - \frac{r_v \min(0, wv)}{a_{\mathsf{lo}}} \\
\mathsf{bound}_1(r, h, v, w, v_{\mathsf{lo}}) &\equiv w {r_v}^2 h < \frac{a_{\mathsf{lo}}}{2}(r + r_p)^2 + w r_v v (r + r_p) - {r_v}^2 h_p \\
\mathsf{case}_2^{-1}(r, v, w, v_{\mathsf{lo}}) &\equiv -r_p - \frac{r_v \min(0, wv)}{a_{\mathsf{lo}}} \le r \le r_p - \frac{r_v \min(0, wv)}{a_{\mathsf{lo}}} \\
\mathsf{bound}_2(r, h, v, w, v_{\mathsf{lo}}) &\equiv wh < -\frac{\min(0, wv)^2}{2 a_{\mathsf{lo}}} - h_p \\
\mathsf{case}_3^{-1}(r, v, w, v_{\mathsf{lo}}) &\equiv r_p - \frac{r_v \min(0, wv)}{a_{\mathsf{lo}}} < r \le r_p + \frac{r_v \max(0, w(v_{\mathsf{lo}} - v))}{a_{\mathsf{lo}}} \\
\mathsf{bound}_3(r, h, v, w, v_{\mathsf{lo}}) &\equiv w {r_v}^2 h < \frac{a_{\mathsf{lo}}}{2}(r - r_p)^2 + w r_v v (r - r_p) - {r_v}^2 h_p \\
\mathsf{case}_4^{-1}(r, v, w, v_{\mathsf{lo}}) &\equiv r_p + \frac{r_v \max(0, w(v_{\mathsf{lo}} - v))}{a_{\mathsf{lo}}} < r \\
\mathsf{bound}_4(r, h, v, w, v_{\mathsf{lo}}) &\equiv (r_v = 0) \lor \left(w r_v h < w v_{\mathsf{lo}}(r - r_p) - \frac{r_v \max(0, w(v_{\mathsf{lo}} - v))^2}{2 a_{\mathsf{lo}}} - r_v h_p \right) \\
\mathsf{case}_5^{-1}(r, v, w, v_{\mathsf{lo}}) &\equiv -r_p \le r < -r_p + \frac{r_v \max(0, w(v_{\mathsf{lo}} - v))}{a_{\mathsf{lo}}} \\
\mathsf{bound}_5(r, h, v, w, v_{\mathsf{lo}}) &\equiv w {r_v}^2 h < \frac{a_{\mathsf{lo}}}{2}(r + r_p)^2 + w r_v v (r + r_p) - {r_v}^2 h_p \\
\mathsf{case}_6^{-1}(r, v, w, v_{\mathsf{lo}}) &\equiv -r_p + \frac{r_v \max(0, w(v_{\mathsf{lo}} - v))}{a_{\mathsf{lo}}} \le r \\
\mathsf{bound}_6(r, h, v, w, v_{\mathsf{lo}}) &\equiv (r_v = 0 \land r > r_p) \lor \left(w r_v h < w v_{\mathsf{lo}}(r + r_p) - \frac{r_v \max(0, w(v_{\mathsf{lo}} - v))^2}{2 a_{\mathsf{lo}}} - r_v h_p \right) \\
L_{\mathsf{expl}}^{-1}(r, h, v, w, v_{\mathsf{lo}}) &\equiv\ \left(w v_{\mathsf{lo}} \ge 0 \to \bigwedge_{i=1}^{4} (\mathsf{case}_i^{-1}(r, v, w, v_{\mathsf{lo}}) \to \mathsf{bound}_i(r, h, v, w, v_{\mathsf{lo}})) \right) \\
&\quad \land \left(w v_{\mathsf{lo}} < 0 \to \bigwedge_{i=5}^{6} (\mathsf{case}_i^{-1}(r, v, w, v_{\mathsf{lo}}) \to \mathsf{bound}_i(r, h, v, w, v_{\mathsf{lo}})) \right)
\end{aligned}
$$

Fig. 3 Implicit and explicit formulations of the safe region for an immediate response (lower bounds for $w = 1$, upper bound for $w = -1$)

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. However, because it still contains quantifiers, its use comes at the extra cost of eliminating the quantifiers, which is inefficient and impractical to repeatedly compute during the comparison part of our analysis. An efficient comparison with the ACAS X table, as described in Sect. 8, can only be achieved with a quantifier-free, *explicit formulation*, that we present in this section. We show that both formulations are equivalent. As for the implicit formulation, we derive the equations for one representative case before generalizing them.

First case: if $w = +1$, $r_v > 0$, $v < 0$ and $v_{\mathsf{lo}} \ge 0$. We are in the case shown in Fig. 1 and described in detail in Sect. 3.3. The nominal trajectory $\mathcal{N}$ is given by Eq. (2). 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. For this case, this boundary can be characterized by a set of equations (where cases 1 to 4 follow cases 1 to 4 of Fig. 3):

0. positions left of the puck's initial position ($r < -r_p$) are in the safe region;
1. 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 $-r_p \leq r < -r_p - \frac{r_v v}{a_{\mathsf{lo}}}$, the position (r, h) is safe if and only if $h < \frac{a_{\mathsf{lo}}}{2{r_v}^2}(r+r_p)^2 + \frac{v}{r_v}(r+r_p) - h_p$;

2. 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 $-r_p - \frac{r_v v}{a} \leq r \leq r_p - \frac{r_v v}{a_{\mathsf{lo}}}$, the position (r, h) is in the safe region if and only if $h < -\frac{v^2}{2a_{\mathsf{lo}}} - h_p$;
3. 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 $r_p - \frac{r_v v}{a_{\mathsf{lo}}} < r \leq r_p + \frac{r_v(v_{\mathsf{lo}} - v)}{a_{\mathsf{lo}}}$, the position (r, h) is safe if and only if $h < \frac{a_{\mathsf{lo}}}{2{r_v}^2}(r - r_p)^2 + \frac{v}{r_v}(r - r_p) - h_p$;
4. 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 $r_p + \frac{r_v(v_{\mathsf{lo}} - v)}{a_{\mathsf{lo}}} < r$, the position (r, h) is in the safe region if and only if $h < \frac{v_{\mathsf{lo}}}{r_v}(r - r_p) - \frac{(v_{\mathsf{lo}} - v)^2}{2a_r} - h_p$.

Generalization. The general case is given in the formula $L^{-1}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}})$ of Fig. 3. The cases 1-4 and their associated bounds are for the case $wv_{\mathsf{lo}} \geq 0$, whereas cases 5 and 6 and associated bounds are for $wv_{\mathsf{lo}} < 0$; both cases 5 and 6 follow the bottom left-hand corner of the puck as it is going along the nominal trajectory. We use KeYmaera X to formally prove that this explicit safe region formulation is equivalent to its implicit counterpart:

Lemma 1 (Equivalence of explicit safe regions) *If $w = \pm 1$, $r_p \geq 0$, $h_p > 0$, $r_v \geq 0$ and $a_{\mathsf{lo}} > 0$, then the conditions $L^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})$ and $L^{-1}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}})$ are equivalent.*

Since the assumptions of Lemma 1 are invariants of the model in Eq. (1), the explicit safe regions give a model that inherits safety from Theorem 1. The proof in KeYmaera X is a combination of contextual equivalence reasoning and monotonicity reasoning [23] to embed the conditional equivalence from Lemma 1 into the context of Theorem 1.

Corollary 1 (Correctness of explicit safe regions) *The d$\mathcal{L}$ formula given in Eq. (1) remains valid when replacing all occurrences of L^{-1}_{impl} with L^{-1}_{expl}. That is as long as the advisories followed obey formula L^{-1}_{expl} there will be no NMAC.*

4 Safe Region for a Delayed Pilot Response

Since the pilot will need some time to react to an advisory issued by ACAS X, 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). In Fig. 4, for example, the pilot reacts to a CL1500 advisory only after a certain reaction delay δ during which she was still in the process of initiating a descent.

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. Yet, the model now includes an initial period where there is no compliance with any advisory—the ownship accelerates non-deterministically (within limits) in the vertical direction. As before, we derive the safe regions by imagining to consider all possible positions of the ownship's puck in all possible trajectories that might evolve in the encounter. To represent pilot delay for an advisory, the model assumes an immediate advisory, and a period of non-compliance δ, representing the time it takes the pilot to respond. To represent COC, the model looks for a safe advisory it can issue after time ε in the future if necessary, where ε is the system delay—i.e., the time before the system can issue a new advisory—and shortest COC. Hence the period of non-compliance is $\varepsilon + \delta$.

$$\begin{array}{rl}
1 & r_p \geq 0 \land h_p > 0 \land r_v \geq 0 \land a_{\mathsf{lo}} > 0 \land \boldsymbol{a^{\mathsf{d}} \geq 0} \\
2 & \boldsymbol{\land \delta \geq 0 \land \varepsilon \geq 0} \land (w = -1 \lor w = 1) \\
3 & \land \boldsymbol{D^{d}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})} \rightarrow \\
4 & [\big((\mathsf{?true} \cup \\
5 & \quad (w := -1 \cup w := 1); v_{\mathsf{lo}} := *; \\
6 & \quad \boldsymbol{(d := \delta; ?D^{d}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})}; \\
7 & \qquad \mathsf{advisory} := (w, v_{\mathsf{lo}}) \cup \\
8 & \qquad \boldsymbol{d := \delta + \varepsilon; ?D^{d}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})}; \\
9 & \qquad \boldsymbol{\mathsf{advisory} := \mathsf{COC}})\big); \\
10 & \quad a := *; \boldsymbol{?(wa \geq -a^{\mathsf{d}}); t := 0;} \\
11 & \quad \{r' = -r_v, h' = -v, v' = a, \boldsymbol{d' = -1, t' = 1} \ \& \\
12 & \quad\ \boldsymbol{(t \leq \varepsilon)} \land (\boldsymbol{d \leq 0} \rightarrow wv \geq wv_{\mathsf{lo}} \lor wa \geq a_{\mathsf{lo}})\} \\
13 &)^*] \, (|r| > r_p \lor |h| > h_p)
\end{array} \qquad (4)$$

We modify the model of Eq. (1) to capture these new ideas, and obtain the model of Eq. (4), highlighting the differences in **bold**. The structure, precondition (lines 1 to 3) and postcondition (line 13) 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 4 to 9 represent the actions of the ACAS X system. As before, the system can continue with the same advisory ($\mathsf{?true}$). Otherwise it can select a safe $\mathsf{advisory}$ (w, v_{lo}) to be applied after at most delay δ; or it can safely remain silent, displaying COC, if it knows an advisory (w, v_{lo}) that is safe if it is followed after a combined pilot and system and pilot delay of $\delta + \varepsilon$. In line 10, the pilot non-deterministically chooses an acceleration ($a := *$), within some limit ($wa \geq -a^{\mathsf{d}}$). The set of differential equations in line 11 describes the system's dynamics, and the conditions in line 12 use the clock t to ensure that continuous time does not evolve longer than system delay ε without a system response ($t \leq \varepsilon$). Those conditions also ensure that when $d \leq 0$ the pilot starts complying with the advisory ($wv \geq wv_{\mathsf{lo}} \lor wa \geq a_{\mathsf{lo}}$ as before).

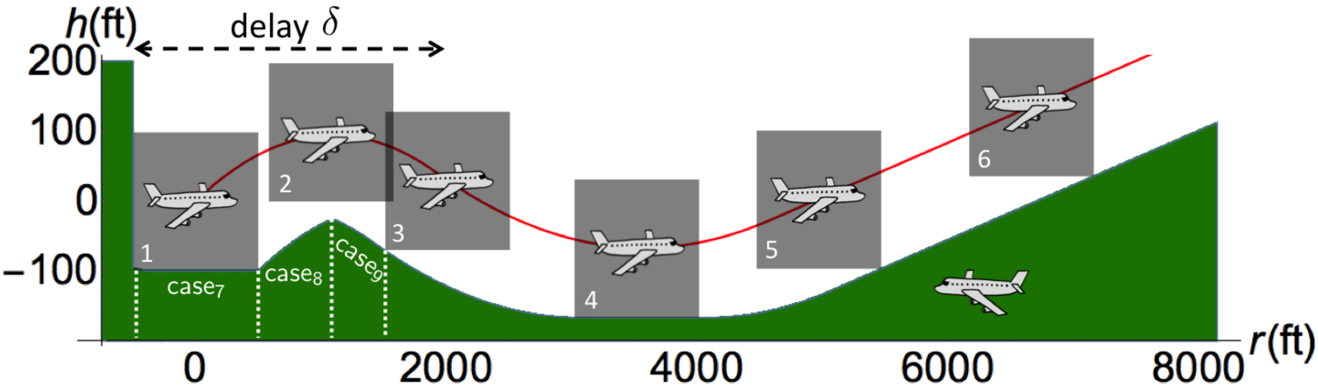


Fig. 4 Nominal trajectory of the ownship (red) and safe region for the intruder (green), *delayed* response

The model is structured so that the pilot can safely delay responding to an advisory for up to δ, and the system can additionally delay issuing an advisory associated with COC for up to ε. Because of the loop in our model (line 13), 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 ownship pilot exhibits some non-deterministic behavior in the vertical dimension.

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 Implicit Formulation of the Safe Region

As in Sect. 3.3, let us place ourselves in the coordinate system 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, v_{lo}) such that $w = +1$, and that $\delta \geq 0$. Let us focus on the period of time before the pilot reacts, which we henceforth call delay. During the delay, the ownship can take any vertical acceleration less than a^{d} in absolute value, therefore its most restrictive nominal trajectory $\mathcal{N}^{\mathsf{d}}$ is to accelerate the opposite way of the advisory, at acceleration $-a^{\mathsf{d}}$. Horizontally, its speed is constant at r_v. It thus describes a *delay parabola*, in red on Fig. 4, and its position (r_n, h_n) along the nominal trajectory for $0 \leq t < \delta$ is given by $(r_n, h_n) = \left(r_v t, -\frac{a^{\mathsf{d}}}{2}t^2 + vt\right)$.

After the delay, i.e., after time δ, the nominal trajectory $\mathcal{N}^{\mathsf{d}}$ is the same as a nominal trajectory $\mathcal{N}$ from Sect. 3, translated by time δ and by its position at time δ given by $r^{\mathsf{d}} = r_n(\delta)$ and $h^{\mathsf{d}} = h_n(\delta)$, and starting with vertical velocity $v^{\mathsf{d}} = v - a^{\mathsf{d}}\delta$. As in Sect. 3.3, we can now express the implicit formulation of the safe region:

$$\forall t. \forall r_n. \forall h_n. (r_n, h_n) \in \mathcal{N}^{\mathsf{d}} \rightarrow |r - r_n| > r_p \vee h - h_n < -h_p$$

Symmetrically, the reasoning of this section extends to the case where $w = -1$. Moreover, we can handle cases where $d < 0$, i.e., after the pilot has reacted, by replacing d by $\max(0, d)$. The generalized implicit formulation of the safe region is given as D^d_{impl} in Fig. 5. Note that it involves the expression $A(t - \max(0, d), h_n - h^{\mathsf{d}}, v^{\mathsf{d}}, w, v_{\mathsf{lo}})$ 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 v^{d}. It is proved correct in KeYmaera.

Theorem 2 (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^d_{impl} there will be no NMAC.*

4.3 Explicit Formulation of the Safe Region

Similarly as in Sect. 4, we determine a (quantifier-free) explicit formulation of the safe region, called D^d_{expl} in Fig. 5 based on Fig. 3, and prove it correct in KeYmaera.

Lemma 2 (Equivalence of delayed explicit safe regions) *If $w = \pm 1$, $r_p \geq 0$, $h_p > 0$, $r_v \geq 0$, $a_{\mathsf{lo}} > 0$, $a^{\mathsf{d}} \geq 0$, $\delta \geq 0$ and $\varepsilon \geq 0$ then the two conditions $D^d_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})$ and $D^d_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}})$ are equivalent.*

5 Safe Region for Subsequent Advisories

The safety analysis from Sect. 3 requires the system to only issue advisories that will never lead to a collision. After presenting our initial results to ACAS X designers and engineers, we received feedback that the safety advice for single advisories was too restrictive for their operational purposes. Early in an encounter, there is often enough separation between aircraft and time in the encounter so that an initial advisory, which would not be safe on its own, can still be changed or corrected to keep the aircraft safe. The rationale is that while ACAS X is designed to avoid collisions, it is also designed to avoid bothering pilots as much as possible.

Implicit formulation

$$B^d(t,h_n,v) \equiv 0 \le t < \max(0,d) \land h_n = -\frac{wa^{\mathsf{d}}}{2}t^2 + vt$$

$$\mathsf{const} \equiv h^{\mathsf{d}} = -\frac{wa^{\mathsf{d}}}{2}\max(0,d)^2 + v\max(0,d) \land v^{\mathsf{d}} - v = -wa^{\mathsf{d}}\max(0,d)$$

$$D^d_{\mathsf{impl}}(r,h,v,w,v_{\mathsf{lo}}) \equiv \forall t.\forall r_n.\forall h_n.\forall h^{\mathsf{d}}.\forall v^{\mathsf{d}}.\Big(r_n = r_v t \land (B^d(t,h_n,v) \lor \mathsf{const} \land A(t-\max(0,d),h_n - h^{\mathsf{d}},v,w,v_{\mathsf{lo}})) \to (|r-r_n| > r_p \lor w(h-h_n) < -h_p)\Big)$$

Explicit formulation

$$r^{\mathsf{d}} = r_v\max(0,d) \qquad v^{\mathsf{d}} = v - wa^{\mathsf{d}}\max(0,d) \qquad h^{\mathsf{d}} = -\frac{wa^{\mathsf{d}}}{2}\max(0,d)^2 + v\max(0,d)$$

$$\mathsf{case}_7(r) \equiv -r_p \le r \le r_p \qquad \mathsf{bound}_7(r,h) \equiv wh < -h_p$$

$$\mathsf{case}_8(r) \equiv r_p < r \le r^{\mathsf{d}} + r_p \qquad \mathsf{bound}_8(r,h) \equiv w{r_v}^2h < -\frac{a_d}{2}(r-r_p)^2 + wr_v v(r-r_p) - {r_v}^2h_p$$

$$\mathsf{case}_9(r) \equiv -r_p \le r < r^{\mathsf{d}} - r_p \qquad \mathsf{bound}_9(r,h) \equiv w{r_v}^2h < -\frac{a_d}{2}(r+r_p)^2 + wr_v v(r+r_p) - {r_v}^2h_p$$

$$D^d_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{lo}}) \equiv \left(\bigwedge_{i=7}^{9}(\mathsf{case}_i(r) \to \mathsf{bound}_i(r,h))\right) \land L^{-1}_{\mathsf{expl}}(r-r^{\mathsf{d}},h-h^{\mathsf{d}},v^{\mathsf{d}},w,v_{\mathsf{lo}})$$

Fig. 5 Implicit and explicit formulations of the safe region for a delayed response

To balance these concerns, if an encounter is not immediately threatening, ACAS X will typically first issue COC, or a preventive advisory like DNC or DND, before issuing a more disruptive advisory to the pilot. In those cases, the first advisory is often not safe in the sense of Sect. 3 : it will tend to keep the planes from flying directly towards each other immediately and will only ensure safety *for a few seconds*, but not *for the rest of the encounter*.

As a consequence, running the safety analysis on the immediate advisory leads to counterexamples considered as false alarms by the ACAS X designers. Anything is safe if you are far enough away; many of the examples of unsafe behavior we found were uninteresting because if the pilots chose a course that brought them closer, the system would issue a more disruptive, but safer follow-on advisory. This section develops a more sophisticated safety analysis taking into account follow-on advisories. More specifically, we define the concept of *safeable* advisories, and develop sufficient, easily checkable conditions for an advisory to be safeable.

Definition 1 (safeable) We say that an advisory is *safeable* if and only if it is safe or can still be made safe in the future, if necessary, via subsequent advisories.

The safeable region of a given advisory is always a superset of its safe region.

This section builds up the safeable region in three steps. We first present two-sided safe regions, providing both an upper and a lower bound to the trajectory. We then present bounded safe regions, which only ensure absence of collision for a limited amount of time ε. Bounded safe regions provide no guarantee after time ε, and the corresponding model has no liveness. Based on these important building blocks, we finally present safeable regions, which model subsequent advisories, and have a corresponding model providing liveness. This section is new, and was not presented in the conference version of this paper [13].

Throughout the section, we still assume that the ownship and intruder are flying head-on ($\theta_v = 180°$), and we use the same conventions as in Sect. 3 for r and r_v. We say that a subsequent advisory is a *reversal* if and only if it is a downsense advisory ($w = -1$) while the first advisory was upsense ($w = 1$)—or vice-versa. In the opposite case we usually call the subsequent advisory a *strengthening* or a *weakening* advisory.

5.1 Two-Sided Safe Region with Immediate Pilot Response

A first step towards the treatment of subsequent advisories is to provide an *envelop* of the trajectory of the ownship while it follows an initial advisory. Indeed, if the initial advisory is upsense with a reversal as a subsequent advisory, then it is crucial to have an estimate of the pose (position and vertical velocity) of the aircraft when the pilot starts following the second advisory. Safe regions described in Sect. 3 are not sufficient as they only provide a lower bound when $w = 1$, and an upper bound when $w = -1$.

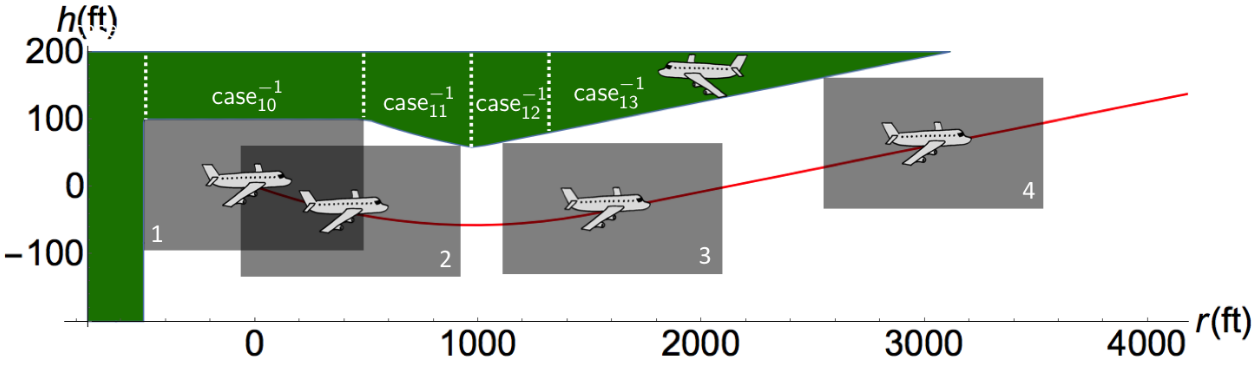


Fig. 6 Nominal trajectory of the ownship (red) and *upper* safe region for the intruder (green), immediate response

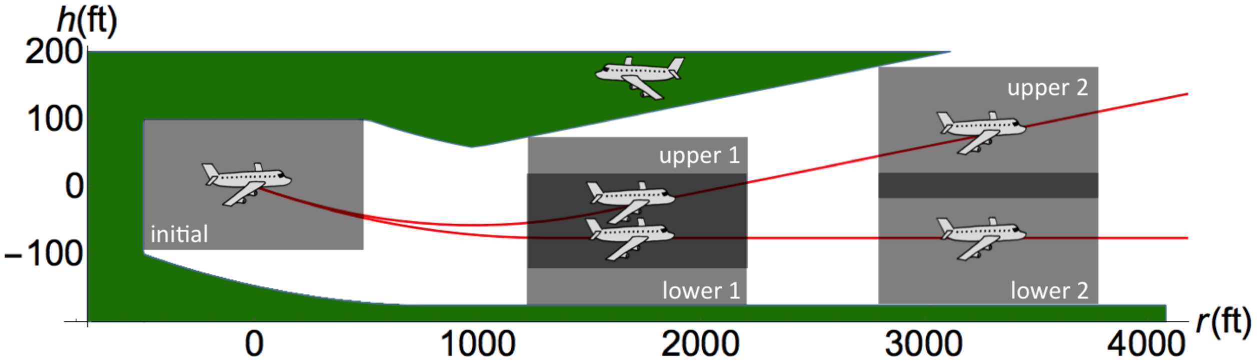


Fig. 7 Nominal trajectories of the ownship (red) and *two-sided* safe region for the intruder (green), immediate response

5.1.1 Model For simplicity, let us first consider the case of an initial upsense advisory ($w = 1$) such as CL1500 or DND; the case of the initial downsense advisory is symmetric. In Sect. 3 we argued that following the advisory (w, v_{lo}) meant that either the vertical speed of the ownship should be greater than v_{lo}, or its acceleration should be greater than a_{lo}, leading to the constraint $wv \geq wv_{\mathsf{lo}} \vee wa \geq a_{\mathsf{lo}}$. Following the same reasoning, we fix upper bounds v_{up} and a_{up} on the vertical velocity and acceleration of the ownship while following the advisory (w, v_{lo}). Those are again symbolic parameters, with typical values $a_{\mathsf{up}} = g/2$ and $v_{\mathsf{up}} = 10{,}000\,\text{ft/min}$. We update the model of Eq. (1) to capture these new ideas, and obtain the model of Eq. (5), highlighting the differences in **bold**.

$$
\begin{array}{rl}
1 & r_p \geq 0 \wedge h_p > 0 \wedge r_v \geq 0 \wedge a_{\mathsf{lo}} > 0 \;\wedge\; \boldsymbol{a_{\mathsf{up}} \geq a_{\mathsf{lo}}}, \\
2 & \wedge\, (w = -1 \vee w = 1) \wedge \boldsymbol{C_{\mathsf{impl}}^{-1}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}}) \rightarrow \\
3 & [(\;\; (\;?\mathsf{true} \cup \\
4 & \qquad (w := -1 \cup w := 1); v_{\mathsf{lo}} := *; \boldsymbol{v_{\mathsf{up}} := *}; \\
5 & \qquad ?\boldsymbol{C_{\mathsf{impl}}^{-1}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}}); \mathsf{advisory} := (w, v_{\mathsf{lo}}, \boldsymbol{v_{\mathsf{up}}})); \\
6 & \quad a := *; \\
7 & \quad \{r' = -r_v, h' = -v, v' = a \\
8 & \qquad \&\; (wv \geq wv_{\mathsf{lo}} \vee wa \geq a_{\mathsf{lo}}) \\
9 & \qquad \boldsymbol{\wedge((wv \leq wv_{\mathsf{up}} \wedge wa \leq a_{\mathsf{up}}) \vee wa \leq 0)}\} \\
10 &)^*]\; (|r| > r_p \vee |h| > h_p)
\end{array}
\tag{5}
$$

Beyond replacing the *lower* safe region L_{impl}^{-1} by a two-sided safe region C_{impl}^{-1}, we impose $a_{\mathsf{up}} \geq a_{\mathsf{lo}}$ to ensure that the pilot can always find a suitable acceleration between a_{lo} and a_{up} (line 1), and we add an upper bound, v_{up}, when a new advisory is selected (lines 4 and 5).

More interestingly, we update the evolution domain of the differential equation (lines 8 and 9). To understand what it means for the ownship to respect the new upper bounds v_{up} and a_{up}, consider an advisory for which $w = 1$, and let us distinguish two cases. If initially $v \leq v_{\mathsf{up}}$, then both upper bounds on vertical velocity and acceleration need to be respected simultaneously, leading to condition $v \leq v_{\mathsf{up}} \wedge a \leq a_{\mathsf{up}}$. Otherwise, $v > v_{\mathsf{up}}$ and the initial vertical speed of the aircraft v is strictly greater than v_{up}. Given that the pilot receives an *upsense* advisory, it would be unrealistic to assume that the aircraft would typically follow a negative acceleration to get its vertical speed to go back to v_{up}. Instead, we assume that the pilot does not accelerate up further, leading to the condition $a \leq 0$. Incorporating the symmetric case $w = -1$ leads to the general evolution domain for the upper bound $(wv_{\mathsf{lo}} \leq wv_{\mathsf{up}} \wedge wa \leq a_{\mathsf{up}}) \vee wa \leq 0$.

This analysis leads to an important realization for the upper safe region: in the case where the initial vertical velocity *overcomplies* (i.e., when $wv \geq wv_{\mathsf{up}}$), the upper target vertical velocity is not v_{up} anymore, but rather it is the initial value of v; in full generality the upper target vertical velocity becomes the *modified* upper target vertical velocity $w\max(wv_{\mathsf{up}}, wv)$. Throughout the implicit and explicit formulations of the safe region, this modified target vertical velocity will play the role simply played by v_{lo} in Sect. 3; we usually highlight it in **bold**.

5.1.2 Implicit formulation of the safe region The safe region C^{-1}_{impl} for two-sided safety consists of L^{-1}_{impl} from Fig. 3 and an additional upper bound U^{-1}_{impl}. The implicit formulation of the upper bound U^{-1}_{impl} is similar to the implicit formulation of the lower bound described in Sect. 3.3. As in Sect. 3.3, we use a coordinate system fixed to the intruder and with its origin at the initial position of the ownship.

First case: if $w = +1$ and $v_{\mathsf{up}} \geq v$. We again consider a (different) upper nominal trajectory $\mathcal{N}_{\mathsf{up}}$, represented in red on Fig. 6. This nominal trajectory accelerates vertically with acceleration a_{up} until reaching the modified target vertical velocity (which, here, is $v_{\mathsf{up}} = \max(wv_{\mathsf{up}}, wv)$), describing a parabola; it then continues at the vertical velocity v_{up} along a straight line. As before, the horizontal velocity remains constant at r_v. The ownship position (r_n, h_n) at time t along this nominal trajectory is, thus, given by:

$$(r_n, h_n) = \begin{cases} \left(r_v t\,,\ \dfrac{a_{\mathsf{up}}}{2}t^2 + vt\right) \text{ if } 0 \leq t < \dfrac{v_{\mathsf{up}} - v}{a_{\mathsf{up}}} & (a) \\ \left(r_v t\,,\ v_{\mathsf{up}}t - \dfrac{(v_{\mathsf{up}} - v)^2}{2a_{\mathsf{up}}}\right) & \\ \qquad\qquad \text{if } \dfrac{v_{\mathsf{up}} - v}{a_{\mathsf{up}}} \leq t & (b) \end{cases} \tag{6}$$

Recall that the specification is that the ownship moves vertically with acceleration of *at most* a_{up}, then continues with vertical velocity of *at most* $\max(v_{\mathsf{up}}, v)$. Therefore all possible future positions of the ownship will turn out to be *below* the red upper nominal trajectory. Therefore, an intruder is now safe if its position (r, h) is always either to the side of *or above* any puck centered on a point in $\mathcal{N}_{\mathsf{up}}$, that is:

$$\begin{aligned} &\forall t.\forall r_n.\forall h_n.\big((r_n, h_n) \in \mathcal{N}_{\mathsf{up}} \\ &\qquad \rightarrow |r - r_n| > r_p \lor \boldsymbol{h - h_n > h_p}\big) \end{aligned} \tag{7}$$

We call this formulation the implicit formulation of the upper safe region.

Generalization. The reasoning above is generalized to the case $w = -1$, leading to fully general equations for the implicit formulation of the upper safe region presented in Fig. 8.

Finally, the condition for the two-sided advisory C^{-1}_{impl} is built as a disjunction of the lower safety advisory L^{-1}_{impl} and upper safety advisory U^{-1}_{impl}. Although we cannot assume that the ownship will follow either nominal trajectory, we show that an ownship following the model of Eq. (5), thus respecting the two-sided condition C^{-1}_{impl}, stays between both nominal trajectories, keeping it safe. The proof of safety is verified in KeYmaera X:

Theorem 3 (Correctness of two-sided safe regions) *The $\mathsf{d}\mathcal{L}$ formula given in Eq.* (5) *is valid. That is as long as the advisories obey formula C^{-1}_{impl} there will be no NMAC.*

5.1.3 Explicit formulation of the safe region Constructing the explicit safety condition for the upper bound U^{-1}_{expl} follows similar motivation and methods as in Sect. 3.4. But instead of distinguishing cases upon the target vertical velocity v_{lo}, it distinguishes them upon the modified upper target vertical velocity $w\max(wv_{\mathsf{up}}, wv)$.

First case: if $w = +1$, $r_v > 0$, $v \leq 0$ and $v_{\mathsf{up}} > 0$. In particular $v_{\mathsf{up}} > v$, therefore the modified upper target vertical velocity is $\max(v_{\mathsf{up}}, v) = v_{\mathsf{up}}$. This is the case described in Fig. 6, and the nominal trajectory $\mathcal{N}_{\mathsf{up}}$ is given by Eq. (7). The boundary of the (green) safe region in Fig. 6 is drawn by either the top side, the top left hand corner or the top right hand corner of the puck. This explicit formulation is a little bit less intuitive than the formulation for the lower safe region of Sect. 3.4 because the different cases overlap. It can nonetheless be described by a set of equations (where cases 10 to 13 are similar to cases 10 to 13 of Fig. 8):

0. positions left of the puck's initial position ($r < -r_p$) are in the safe region;
10. up to $r = r_p$, the boundary is horizontal along the top side of the puck at its initial position; therefore for $-r_p \leq r \leq r_p$, the position (r, h) is in the safe region if and only if $h > h_p$;
11. then the boundary can follow the top right-hand corner of the puck as it is going down the parabola of Eq. (6)(a); therefore for $r_p < r \leq r_p + \frac{r_v(v_{\mathsf{up}}-v)}{a_{\mathsf{up}}}$, the position (r, h) is safe if and only if $h > \frac{a_{\mathsf{up}}}{2{r_v}^2}(r - r_p)^2 + \frac{v}{r_v}(r - r_p) + h_p$;
12. the boundary can also follow the top left-hand corner of the puck as it is going up the parabola of Eq. (6)(a); therefore for $-r_p \leq r < -r_p + \frac{r_v(v_{\mathsf{up}}-v)}{a_{\mathsf{up}}}$, the position (r, h) is safe if and only if $h > \frac{a_{\mathsf{up}}}{2{r_v}^2}(r + r_p)^2 + \frac{v}{r_v}(r + r_p) + h_p$; note that this case can overlap with case 10;
13. finally the boundary follows the top left-hand corner of the puck as it is going up the straight line of Eq. (6)(a); therefore for $-r_p + \frac{r_v(v_{\mathsf{up}}-v)}{a_{\mathsf{up}}} \leq r$, the position (r, h) is in the safe region if and only if $h > \frac{v_{\mathsf{up}}}{r_v}(r - r_p) - \frac{(v_{\mathsf{up}}-v)^2}{2a_{\mathsf{up}}} + h_p$.

Generalization The general case is given in formula U^{-1}_{expl} of Fig. 8. The cases 10-13, described above for a specific situation, are for the case $\max(wv_{\mathsf{up}}, wv) > 0$, whereas cases

Implicit formulation

$$
\begin{aligned}
A_{\mathsf{up}}(t, h_n, v, w, v_{\mathsf{up}}) \equiv\ & \left(0 \le t_n < \frac{\max(0, w(v_{\mathsf{up}} - v))}{a_{\mathsf{up}}} \land h_n = \frac{w a_{\mathsf{up}}}{2} {t_n}^2 + v t_n\right) \\
& \lor \left(t_n \ge \frac{\max(0, w(v_{\mathsf{up}} - v))}{a_{\mathsf{up}}} \land h_n = w \max(w v_{\mathsf{up}}, wv) t_n - \frac{w \max(0, w(v_{\mathsf{up}} - v))^2}{2 a_{\mathsf{up}}}\right) \\
U^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{up}}) \equiv\ & \forall t_n . \forall r_n . \forall h_n . \Big(r_n = r_v t_n \land A_{\mathsf{up}}(t, h_n, v, w, v_{\mathsf{up}}) \rightarrow (|r - r_n| > r_p \lor w(h - h_n) > h_p) \Big) \\
C^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}}) \equiv\ & L^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}}) \lor U^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{up}})
\end{aligned}
$$

Explicit formulation

$$
\begin{aligned}
\mathsf{case}^{-1}_{10}(r, v, w, v_{\mathsf{up}}) &\equiv -r_p \le r \le r_p \\
\mathsf{bound}_{10}(r, h, v, w, v_{\mathsf{up}}) &\equiv wh > h_p \\
\mathsf{case}^{-1}_{11}(r, v, w, v_{\mathsf{up}}) &\equiv r_p < r \le r_p + \frac{r_v \max(0, w(v_{\mathsf{up}} - v))}{a_{\mathsf{up}}} \\
\mathsf{bound}_{11}(r, h, v, w, v_{\mathsf{up}}) &\equiv w {r_v}^2 h > \frac{a_{\mathsf{up}}}{2}(r - r_p)^2 + w r_v v (r - r_p) + {r_v}^2 h_p \\
\mathsf{case}^{-1}_{12}(r, v, w, v_{\mathsf{up}}) &\equiv -r_p \le r < -r_p + \frac{r_v \max(0, w(v_{\mathsf{up}} - v))}{a_{\mathsf{up}}} \\
\mathsf{bound}_{12}(r, h, v, w, v_{\mathsf{up}}) &\equiv w {r_v}^2 h > \frac{a_{\mathsf{up}}}{2}(r + r_p)^2 + w r_v v (r + r_p) + {r_v}^2 h_p \\
\mathsf{case}^{-1}_{13}(r, v, w, v_{\mathsf{up}}) &\equiv -r_p + \frac{r_v \max(0, w(v_{\mathsf{up}} - v))}{a_{\mathsf{up}}} \le r \\
\mathsf{bound}_{13}(r, h, v, w, v_{\mathsf{up}}) &\equiv (r_v = 0 \land r > r_p) \lor \left(w r_v h > \max(w v_{\mathsf{up}}, wv)(r + r_p) - \frac{r_v \max(0, w(v_{\mathsf{up}} - v))^2}{2 a_{\mathsf{up}}} + r_v h_p \right) \\
\mathsf{case}^{-1}_{14}(r, v, w, v_{\mathsf{up}}) &\equiv r_p + \frac{r_v \max(0, w(v_{\mathsf{up}} - v))}{a_{\mathsf{up}}} < r \\
\mathsf{bound}_{14}(r, h, v, w, v_{\mathsf{up}}) &\equiv (r_v = 0) \lor \left(w r_v h > \max(w v_{\mathsf{up}}, wv)(r - r_p) - \frac{r_v \max(0, w(v_{\mathsf{up}} - v))^2}{2 a_{\mathsf{up}}} + r_v h_p \right) \\
U^{-1}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{up}}) &\equiv \left(\max(w v_{\mathsf{up}}, wv) > 0 \rightarrow \bigwedge_{i=10}^{13} (\mathsf{case}^{-1}_i(r, v, w, v_{\mathsf{up}}) \rightarrow \mathsf{bound}_i(r, h, v)) \right) \\
&\quad \land \left(\max(w v_{\mathsf{up}}, wv) \le 0 \rightarrow \bigwedge_{i \in \{10,11,14\}} (\mathsf{case}^{-1}_i(r, v, w, v_{\mathsf{up}}) \rightarrow \mathsf{bound}_i(r, h, v)) \right) \\
C^{-1}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}}) &\equiv L^{-1}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}}) \lor U^{-1}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{up}})
\end{aligned}
$$

Fig. 8 Implicit and explicit formulations of the safe region for an immediate response (upper bounds for $w = 1$, lower bound for $w = -1$)

10, 11 and 14 are used for the case $\max(w v_{\mathsf{up}}, wv) \le 0$. C ase 14 follows the top left-hand corner of the puck.

Finally, the explicit condition for the two-sided advisory C^{-1}_{expl} is built as a disjunction of the lower and upper safety advisories, as shown in Fig. 8. A graphic representation of C^{-1}_{expl} (in green) along with its associated nominal trajectories is shown in Fig. 7. We again use KeYmaera X to formally prove that this explicit two-sided safe region formulation is equivalent to its implicit counterpart:

Lemma 3 (Equivalence of two-sided explicit safe regions) *If* $w = \pm 1$, $r_p \ge 0$, $h_p > 0$, $r_v \ge 0$, $a_{\mathsf{lo}} > 0$, $a_{\mathsf{up}} \ge a_{\mathsf{lo}}$ *then the conditions* $C^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})$ *and* $C^{-1}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}})$ *are equivalent.*

The assumptions of Lemma 3 are invariants of the model in Eq. (5) . As a consequence, a model of explicit safe regions inherits the safety property from Theorem 3 which is also proved in KeYmaera X by conditional congruence reasoning.

Corollary 2 (Correctness of two-sided explicit safe regions) *The d$\mathcal{L}$ formula given in Eq.* (5) *remains valid when replacing all occurrences of* $C^{-1}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}})$ *with* $C^{-1}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}})$. *That is, as long as the advisories followed obey formula* $C^{-1}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}})$ *there will be no NMAC.*

5.2 Bounded-Time Safe Regions

We build on the two-sided safe region to build a model and safe regions for bounded-time safety, i.e., regions only guaranteeing safety of the ownship up to some time ε. Flying

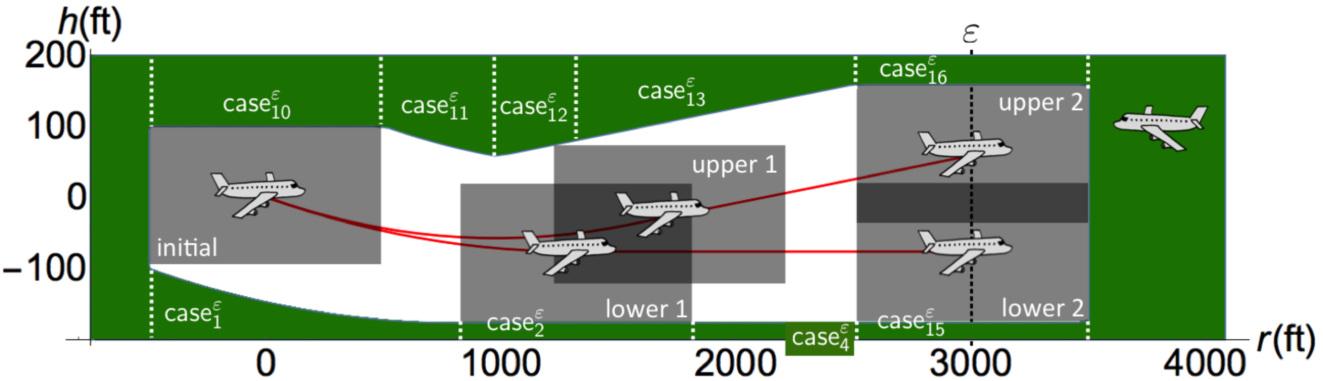


Fig. 9 Nominal trajectories of the ownship (red) and *bounded-time safe* region for the intruder (green), immediate response

aircraft in ways that are merely safe for a bounded time ε is inherently unsafe. It is, nevertheless, a critical building block toward constructing safeable regions, since those feature advisories that are acceptable for some time ε and can be followed up with safe subsequent advisories. This section studies only the former aspect of safety for bounded time. An intuitive understanding of bounded-time safe regions can be gathered from Fig. 9: the nominal trajectories stop at time ε, beyond which the safe region provides no guarantee at all. The corresponding safe regions are truncated vertically at $r = r_v\varepsilon + r_p$.

We call the corresponding conditions $L^\varepsilon_{\mathsf{impl}}$ and $L^\varepsilon_{\mathsf{expl}}$ for lower bounded-time safety, $U^\varepsilon_{\mathsf{impl}}$ and $U^\varepsilon_{\mathsf{expl}}$ for upper bounded-time safety, as well as $C^\varepsilon_{\mathsf{impl}}$ and $C^\varepsilon_{\mathsf{expl}}$ for two-sided bounded-time safety. By convention, a negative $\varepsilon < 0$ signifies an unbounded region, which fits to the notations L^{-1}_{impl} and L^{-1}_{expl}, U^{-1}_{impl} and U^{-1}_{expl}, C^{-1}_{impl} and C^{-1}_{expl} used in Sect. 3 and 5.1.

5.2.1 Model We modify the model of Eq. (5) to reflect the ideas of safety for up to time ε and obtain the model of Eq. (8), highlighting differences in **bold**:

$$\begin{array}{ll}
1 & r_p \geq 0 \land h_p > 0 \land r_v \geq 0 \land a_{\mathsf{lo}} > 0 \ \land\ a_{\mathsf{up}} \geq a_{\mathsf{lo}} \\
2 & \land\, (w = -1 \lor w = 1) \land \boldsymbol{C^\varepsilon_{\mathsf{impl}}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}}) \rightarrow \\
3 & [(\ ((w := -1 \cup w := 1); v_{\mathsf{lo}} := *; v_{\mathsf{up}} := *; \\
4 & \quad ?\boldsymbol{C^\varepsilon_{\mathsf{impl}}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}});\ \mathsf{advisory} := (w, v_{\mathsf{lo}}, v_{\mathsf{up}})); \\
5 & \quad \boldsymbol{t := 0;} \\
6 & \quad (\ a := *; \\
7 & \quad\ \ \{r' = -r_v, h' = -v, v' = a, \boldsymbol{t' = 1} \\
8 & \quad\quad \&\ \boldsymbol{(t \leq \varepsilon \lor \varepsilon < 0)} \\
9 & \quad\quad \land (wv \geq wv_{\mathsf{lo}} \lor wa \geq a_{\mathsf{lo}}) \\
10 & \quad\quad \land ((wv \leq wv_{\mathsf{up}}\ \land\ wa \leq a_{\mathsf{up}})\ \lor\ wa \leq 0) \\
11 & \quad\ \ \}\ \boldsymbol{)^*} \\
12 &)^*]\ (|r| > r_p \lor |h| > h_p)
\end{array} \tag{8}$$

Beyond replacing the condition C^{-1}_{impl} by $C^\varepsilon_{\mathsf{impl}}$ at lines 2 and 4, the most notable difference is the disappearance of the ?true case in the system decision (line 3 of Eq. (5)): since an advisory can only be followed during at most time ε, we disallow the model to loop and continue following the same advisory. However, we need to still allow the pilot to use several accelerations while she is following a given advisory; to model this we add a loop (*) around the pilot decisions on lines 6 to 11; in Eq. (5) this second loop was not necessary thanks to the ?true case. Finally, we add an explicit clock variable t to model time since the last advisory was issued. The variable t is initialized to 0 at each initial advisory (line 5), evolves with derivative 1 and enforces that the differential equation does not execute for longer than time bound ε ($t \leq \varepsilon$ in line 7) unless time is unbounded (encoded by $\varepsilon < 0$). Note that t is only reset on line 5 before the pilot's loop lines 6–10, so beyond time $t = \varepsilon$, only repetitions of the outer loop lines 3–11 in Eq. (8) make any progress, which will first issue an updated ACAS X advisory in lines 3–4 for the pilot to comply with from then on.

5.2.2 Implicit formulation of the bounded-time safe region The implicit and explicit formulations of the bounded-time safe regions modify the different cases presented in Sect. 5.1 to take into account the time bound ε. The general philosophy is to have the bounded-time equations be an extension of the equations presented in Sect. 5.1: to achieve that all supplemental restrictions are of the form ($\varepsilon < 0 \lor$ restriction), which trivially evaluates to true when considering an unbounded time condition (represented by $\varepsilon < 0$). Full equations are presented in Fig. 10.

The implicit formulations $L^\varepsilon_{\mathsf{impl}}$ and $U^\varepsilon_{\mathsf{impl}}$ are very similar to the ones presented in Sect. 5.1: when considering a bounded nominal lower or upper trajectory, we only add a condition $t_n \leq \varepsilon$ whenever $\varepsilon \geq 0$, to truncate the nominal trajectory at time $t_n = \varepsilon$. As usual, the two-sided implicit formulation $C^\varepsilon_{\mathsf{impl}}$ is the disjunction of $L^\varepsilon_{\mathsf{impl}}$ and $U^\varepsilon_{\mathsf{impl}}$.

We verify the proof of safety in KeYmaera X:

Theorem 4 (Correctness of bounded-time implicit safe regions) *The d$\mathcal{L}$ formula given in Eq.* (8) *is valid. That is*

Implicit formulation

$$L^{\varepsilon}_{\mathsf{impl}}(r,h,v,w,v_{\mathsf{lo}}) \equiv \forall t_n.\forall r_n.\forall h_n.\Big(\boldsymbol{(\varepsilon < 0 \lor t_n \le \varepsilon)} \land r_n = r_v t_n \land A_{\mathsf{lo}}(t,h_n,v,w,v_{\mathsf{lo}}) \rightarrow (|r-r_n| > r_p \lor w(h-h_n) < h_p)\Big)$$

$$U^{\varepsilon}_{\mathsf{impl}}(r,h,v,w,v_{\mathsf{up}}) \equiv \forall t_n.\forall r_n.\forall h_n.\Big(\boldsymbol{(\varepsilon < 0 \lor t_n \le \varepsilon)} \land r_n = r_v t_n \land A_{\mathsf{up}}(t,h_n,v,w,v_{\mathsf{up}}) \rightarrow (|r-r_n| > r_p \lor w(h-h_n) > h_p)\Big)$$

$$C^{\varepsilon}_{\mathsf{impl}}(r,h,v,w,v_{\mathsf{lo}},v_{\mathsf{up}}) \equiv L^{\varepsilon}_{\mathsf{impl}}(r,h,v,w,v_{\mathsf{lo}}) \lor U^{\varepsilon}_{\mathsf{impl}}(r,h,v,w,v_{\mathsf{up}})$$

Explicit formulation

$$\begin{aligned}
\mathsf{case}^{\varepsilon}_1(r,v,w,v_{\mathsf{lo}}) &\equiv \mathsf{case}^{-1}_1(r,v,w,v_{\mathsf{lo}}) \land (\boldsymbol{\varepsilon < 0 \lor r \le -r_p + r_v\varepsilon})\\
\mathsf{case}^{\varepsilon}_2(r,v,w,v_{\mathsf{lo}}) &\equiv \mathsf{case}^{-1}_2(r,v,w,v_{\mathsf{lo}}) \land \left(\boldsymbol{\varepsilon < 0 \lor -\frac{\min(0,wv)}{a_{\mathsf{lo}}} \le \varepsilon}\right)\\
\mathsf{case}^{\varepsilon}_3(r,v,w,v_{\mathsf{lo}}) &\equiv \mathsf{case}^{-1}_3(r,v,w,v_{\mathsf{lo}}) \land (\boldsymbol{\varepsilon < 0 \lor r \le r_p + r_v\varepsilon})\\
\mathsf{case}^{\varepsilon}_4(r,v,w,v_{\mathsf{lo}}) &\equiv \mathsf{case}^{-1}_4(r,v,w,v_{\mathsf{lo}}) \land (\boldsymbol{\varepsilon < 0 \lor r \le r_p + r_v\varepsilon})\\
\mathsf{case}^{\varepsilon}_5(r,v,w,v_{\mathsf{lo}}) &\equiv \mathsf{case}^{-1}_5(r,v,w,v_{\mathsf{lo}}) \land (\boldsymbol{\varepsilon < 0 \lor r \le -r_p + r_v\varepsilon})\\
\mathsf{case}^{\varepsilon}_6(r,v,w,v_{\mathsf{lo}}) &\equiv \mathsf{case}^{-1}_6(r,v,w,v_{\mathsf{lo}}) \land (\boldsymbol{\varepsilon < 0 \lor r \le -r_p + r_v\varepsilon})\\
\mathsf{case}^{\varepsilon}_{10}(r,v,w,v_{\mathsf{up}}) &\equiv \mathsf{case}^{-1}_{10}(r,v,w,v_{\mathsf{up}})\\
\mathsf{case}^{\varepsilon}_{11}(r,v,w,v_{\mathsf{up}}) &\equiv \mathsf{case}^{-1}_{11}(r,v,w,v_{\mathsf{up}}) \land (\boldsymbol{\varepsilon < 0 \lor r \le r_p + r_v\varepsilon})\\
\mathsf{case}^{\varepsilon}_{12}(r,v,w,v_{\mathsf{up}}) &\equiv \mathsf{case}^{-1}_{12}(r,v,w,v_{\mathsf{up}}) \land (\boldsymbol{\varepsilon < 0 \lor r \le -r_p + r_v\varepsilon})\\
\mathsf{case}^{\varepsilon}_{13}(r,v,w,v_{\mathsf{up}}) &\equiv \mathsf{case}^{-1}_{13}(r,v,w,v_{\mathsf{up}}) \land (\boldsymbol{\varepsilon < 0 \lor r \le -r_p + r_v\varepsilon})\\
\mathsf{case}^{\varepsilon}_{14}(r,v,w,v_{\mathsf{up}}) &\equiv \mathsf{case}^{-1}_{14}(r,v,w,v_{\mathsf{up}}) \land (\boldsymbol{\varepsilon < 0 \lor r \le r_p + r_v\varepsilon})
\end{aligned}$$

$$\begin{aligned}
L^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{lo}}) \equiv\ & \left(wv_{\mathsf{lo}} \ge 0 \rightarrow \bigwedge_{i=1}^{4}(\mathsf{case}^{\varepsilon}_i(r,v,w,v_{\mathsf{lo}}) \rightarrow \mathsf{bound}_i(r,h,v,w,v_{\mathsf{lo}}))\right)\\
\land & \left(wv_{\mathsf{lo}} < 0 \rightarrow \bigwedge_{i=5}^{6}(\mathsf{case}^{\varepsilon}_i(r,v,w,v_{\mathsf{up}}) \rightarrow \mathsf{bound}_i(r,h,v,w,v_{\mathsf{up}}))\right)\\
U^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{up}}) \equiv\ & \left(\max(wv_{\mathsf{up}},wv) > 0 \rightarrow \bigwedge_{i=10}^{13}(\mathsf{case}^{\varepsilon}_i(r,v,w,v_{\mathsf{up}}) \rightarrow \mathsf{bound}_i(r,h,v,w,v_{\mathsf{up}}))\right)\\
\land & \left(\max(wv_{\mathsf{up}},wv) \le 0 \rightarrow \bigwedge_{i\in\{10,11,14\}}(\mathsf{case}^{\varepsilon}_i(r,v,w,v_{\mathsf{up}}) \rightarrow \mathsf{bound}_i(r,h,v,w,v_{\mathsf{up}}))\right)\\
C^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{lo}},v_{\mathsf{up}}) \equiv\ & L^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{lo}}) \lor U^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{up}})
\end{aligned}$$

Special cases of the bounded-time explicit formulation

$$\begin{aligned}
\mathsf{case}^{\varepsilon}_{15}(r,v,w,v_{\mathsf{lo}}) &\equiv \mathsf{case}^{\varepsilon}_{16}(r,v,w,v_{\mathsf{lo}}) \equiv \varepsilon \ge 0 \land -r_p + r_v\varepsilon \le r \le r_p + r_v\varepsilon\\
\mathsf{bound}^{\varepsilon}_{15}(r,h,v,w,v_{\mathsf{lo}}) &\equiv \left(\varepsilon \le \frac{\max(0,w(v_{\mathsf{lo}}-v))}{a_{\mathsf{lo}}} \rightarrow wh < \frac{a_{\mathsf{lo}}}{2}\varepsilon^2 + wv\varepsilon - h_p\right)\\
&\land \left(\varepsilon > \frac{\max(0,w(v_{\mathsf{lo}}-v))}{a_{\mathsf{lo}}} \rightarrow wh < wv\varepsilon - \frac{\max(0,w(v_{\mathsf{lo}}-v))^2}{2a_{\mathsf{lo}}} - h_p\right)\\
\mathsf{bound}^{\varepsilon}_{16}(r,h,v,w,v_{\mathsf{up}}) &\equiv \left(\varepsilon \le \frac{\max(0,w(v_{\mathsf{up}}-v))}{a_{\mathsf{up}}} \rightarrow wh > \frac{a_{\mathsf{up}}}{2}\varepsilon^2 + wv\varepsilon + h_p\right)\\
&\land \left(\varepsilon > \frac{\max(0,w(v_{\mathsf{up}}-v))}{a_{\mathsf{up}}} \rightarrow wh > \max(wv_{\mathsf{up}},wv)\varepsilon - \frac{\max(0,w(v_{\mathsf{up}}-v))^2}{2a_{\mathsf{up}}} + h_p\right)\\
\hat{L}^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{lo}}) &\equiv L^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{lo}}) \land (wv_{\mathsf{lo}} < 0 \rightarrow \mathsf{case}^{\varepsilon}_{15}(r,v,w,v_{\mathsf{lo}}) \rightarrow \mathsf{bound}_{15}(r,h,v,w,v_{\mathsf{lo}}))\\
\hat{U}^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{up}}) &\equiv U^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{up}}) \land (\max(wv_{\mathsf{up}},wv) \le 0 \rightarrow \mathsf{case}^{\varepsilon}_{16}(r,v,w,v_{\mathsf{up}}) \rightarrow \mathsf{bound}_{16}(r,h,v,w,v_{\mathsf{up}}))\\
\hat{C}^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{lo}},v_{\mathsf{up}}) &\equiv \hat{L}^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{lo}}) \lor \hat{U}^{\varepsilon}_{\mathsf{expl}}(r,h,v,w,v_{\mathsf{up}})
\end{aligned}$$

Fig. 10 Implicit and explicit formulations of the safe region for bounded time

as long as the advisories obey formula $C^{\varepsilon}_{\mathsf{impl}}$ there will be no NMAC for time up to ε if $\varepsilon \geq 0$, and forever if $\varepsilon < 0$. There are no guarantees beyond time ε if $\varepsilon \geq 0$.

The loop invariant used to prove Eq. (8) has a subtle difference compared to the previous theorems. Unlike in all previous theorems, $C^{\varepsilon}_{\mathsf{impl}}$ is *not* an invariant of the corresponding model Eq. (8) (but almost). To turn the implicit conditions of Fig. 10 into an invariant, we capture the *remaining time* that we must follow an advisory by simply turning ε into $(\varepsilon - t)$. That is after having followed an advisory for duration t we have to follow it for the remaining duration $\varepsilon - t$. The condition $\varepsilon < 0$ encodes advisories that must be followed forever, and remains unchanged in the invariant. So $\varepsilon < 0 \vee t_n \leq \varepsilon$ turns into $\varepsilon < 0 \vee t_n \leq \varepsilon - t$ in both $L^{\varepsilon}_{\mathsf{impl}}$ and $U^{\varepsilon}_{\mathsf{impl}}$ to obtain the invariant.

5.2.3 Explicit formulation of the bounded-time safe region The explicit formulation of the bounded-time safe region also builds on its unbounded-time counterpart from Sect. 5.1. In cases 1 to 6 and 10 to 14, and whenever $\varepsilon \geq 0$, only the following cases need to be modified:

- for a case that follows the bottom or top *left*-hand corner of the puck, the corresponding boundary of the safe region should now stop when the puck reaches time ε, i.e., when the corner reaches $-r_p + r_v\varepsilon$. Therefore we add the condition $r \leq -r_p + r_v\varepsilon$. This is for $\mathsf{case}^{\varepsilon}_1$, $\mathsf{case}^{\varepsilon}_5$, $\mathsf{case}^{\varepsilon}_6$, $\mathsf{case}^{\varepsilon}_{12}$ and $\mathsf{case}^{\varepsilon}_{13}$;
- for a case that follows the bottom or top *right*-hand corner of the puck, the corresponding boundary of the safe region should now stop when the puck reaches time ε, i.e., when the corner reaches $r_p + r_v\varepsilon$. Therefore we add the condition $r \leq r_p + r_v\varepsilon$. This is for $\mathsf{case}^{\varepsilon}_3$, $\mathsf{case}^{\varepsilon}_4$, $\mathsf{case}^{\varepsilon}_{11}$, and $\mathsf{case}^{\varepsilon}_{14}$;
- $\mathsf{case}^{\varepsilon}_{10}$ models the boundary above the puck at time 0 and is unaffected by bounded time;
- $\mathsf{case}^{\varepsilon}_2$ should only appear if the puck ever reaches the bottom of the parabola Eq. (6)(a), that is, only in the case where $-\frac{\min(0,wv)}{a_{\mathsf{lo}}} \leq \varepsilon$, which is exactly the condition we added.

The formulas for $L^{\varepsilon}_{\mathsf{expl}}$, $U^{\varepsilon}_{\mathsf{expl}}$ and $C^{\varepsilon}_{\mathsf{expl}}$ are constructed from these modified cases as before.

However, those changes alone are not enough. In the expression of $L^{\varepsilon}_{\mathsf{expl}}$ and when $wv_{\mathsf{lo}} \geq 0$, there is a missing explicit boundary along the bottom side of the puck at time ε; we add it explicitly as $\mathsf{case}_{15} \rightarrow \mathsf{bound}_{15}$ to form $\hat{L}^{\varepsilon}_{\mathsf{expl}}$. Similarly, in the expression of $U^{\varepsilon}_{\mathsf{expl}}$ and when we have $\max(wv_{\mathsf{up}}, wv) \leq 0$, there is a missing explicit boundary along the top side of the puck at time ε; we add it explicitly as $\mathsf{case}_{16} \rightarrow \mathsf{bound}_{16}$ to form $\hat{U}^{\varepsilon}_{\mathsf{expl}}$. We still define $\hat{C}^{\varepsilon}_{\mathsf{expl}}$ as the disjunction $\hat{L}^{\varepsilon}_{\mathsf{expl}} \vee \hat{U}^{\varepsilon}_{\mathsf{expl}}$. These extra cases 15 and 16 are inconsequential for the safeable result ultimately obtained and are, thus, kept in the separate expression $\hat{C}^{\varepsilon}_{\mathsf{expl}}$.

Lemma 4 (Equivalence of bounded-time explicit safe regions) *If $w = \pm 1$, $r_p \geq 0$, $h_p > 0$, $r_v \geq 0$, $a_{\mathsf{lo}} > 0$, $a_{\mathsf{up}} \geq a_{\mathsf{lo}}$ then the two conditions $C^{\varepsilon}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}})$ and $\hat{C}^{\varepsilon}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}})$ are equivalent.*

To prove this lemma we first prove that $L^{\varepsilon}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}})$ and $\hat{L}^{\varepsilon}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}})$ are equivalent, then that conditions $U^{\varepsilon}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{up}})$ and $\hat{U}^{\varepsilon}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{up}})$ are equivalent.

The safety of explicit safe regions follows from Theorem 4 and Lemma 4 by conditional congruence reasoning.

Corollary 3 (Correctness of bounded-time explicit safe regions) *The d$\mathcal{L}$ formula in Eq. (8) remains valid when replacing all occurrences of $C^{\varepsilon}_{\mathsf{impl}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}})$ with the formula $\hat{C}^{\varepsilon}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}})$. That is, as long as the advisories followed obey formula $\hat{C}^{\varepsilon}_{\mathsf{expl}}(r, h, v, w, v_{\mathsf{lo}}, v_{\mathsf{up}})$ there will be no NMAC.*

5.3 Safeable region

Putting together the building blocks we have presented, we finally present safeable regions, in implicit $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{impl}}$ and explicit form $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{expl}}$. The intuition behind the construction of the safeable region is captured in Fig. 11: we consider all the positions and speeds at which the ownship can end up at time ε, and in particular the lowest such position and speed (position lower 1), and the highest such position and speed (position upper 1). At the lowest position, we look at the most extreme strengthening available; and at the highest position, we look at the most extreme reversal available. The disjunction of the two safe regions of this strengthening and of this reversal corresponds to intruder positions that can be avoided by an appropriate action at time ε: this is the safeable region. Another way of seeing safeable is that it is a subset of bounded-time safe that also provides liveness of the model: it ensures that the ownship does not get stuck without safe followups at time ε.

The safeable formulation is presented in Fig. 12, and a graphic representation in Fig. 11. Throughout this section we suppose that $\varepsilon \geq 0$, i.e., all the safe regions not explicitly labelled as non-bounded-time (with superscript -1) have a finite time bound.

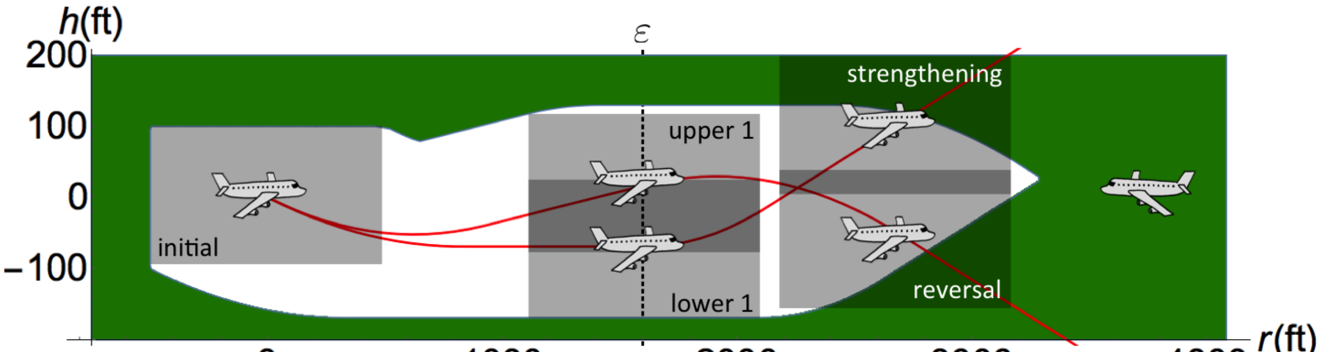


Fig. 11 Nominal trajectories of the ownship (red) and *safeable* region for the intruder (green), immediate response

5.3.1 Model The model is presented in Eq. (9), and builds on the bounded-time model Eq. (8), with very few changes again highlighted in **bold**.

```
1  $r_p \geq 0 \land h_p > 0 \land r_v \geq 0 \land a_{\mathsf{lo}} > 0 \ \land\ a_{\mathsf{up}} \geq a_{\mathsf{lo}}$
2  $\land\, \boldsymbol{\varepsilon \geq 0} \land\ (w = -1 \lor w = 1)$
3  $\land\, \boldsymbol{C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{impl}}}(r, h, v, w, v_{\mathsf{lo}}, v^{\mathsf{ex}}_{\mathsf{lo}}, v_{\mathsf{up}}, v^{\mathsf{ex}}_{\mathsf{up}}) \to$
4  $[(\ (\ (w := -1 \cup w := 1); v_{\mathsf{lo}} := *; v_{\mathsf{up}} := *;$
5      $?\boldsymbol{C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{impl}}}(r, h, v, w, v_{\mathsf{lo}}, v^{\mathsf{ex}}_{\mathsf{lo}}, v_{\mathsf{up}}, v^{\mathsf{ex}}_{\mathsf{up}});$
6      $\mathsf{advisory} := (w, v_{\mathsf{lo}}, v_{\mathsf{up}})\ );$
7    $t := 0;$                                                    (9)
8    $(\ a := *;$
9      $\{r' = -r_v, h' = -v, v' = a, t' = 1 \ \&\ t \leq \varepsilon$
10       $\land\, (wv \geq wv_{\mathsf{lo}} \lor wa \geq a_{\mathsf{lo}})$
11       $\land\, ((wv \leq wv_{\mathsf{up}} \ \land\ wa \leq a_{\mathsf{up}}) \ \lor\ wa \leq 0)$
12     $\}\ )^*$
13 $)^*]\ (|r| > r_p \lor |h| > h_p)$
```

In fact, we are only changing the conditions to $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{impl}}$ on lines 2 and 4. But that makes a big difference: informally, instead of having a model that gets stuck at time ε, we now have a model that can always find a safeable advisory (although we don't formally prove that last fact yet).

5.3.2 Implicit and explicit formulations of the safeable regions The formulations presented in Fig. 12 use the formulations of the bounded-time safe regions as their building blocks. The implicit and explicit formulations are built in very similar ways.

As shown in Fig. 11, the nominal lower bound trajectory consists of a bounded-time lower bound trajectory starting at time 0, followed by an unbounded-time lower bound trajectory starting at time ε; this nominal trajectory is at height h^{ex} and vertical velocity v^{ex} at time ε. Therefore the safeable lower bound consists of one bounded-time lower bound up to time ε, followed by an unbounded-time lower bound starting at time ε, height h^{ex}_L and vertical velocity v^{ex}_L.

The nominal upper bound trajectory consists, however, of a bounded-time upper bound trajectory starting at time 0, followed by an unbounded time reversed (i.e., taking $-w$) lower bound trajectory starting at time ε; this nominal trajectory is at height h^{ex} and vertical velocity v^{ex} at time ε. Therefore the safeable upper bound consists of one bounded-time lower bound up to time ε, followed by an unbounded-time lower bound starting at time ε, height h^{ex}_U and vertical velocity v^{ex}_U.

As usual, the proof of safety is verified in KeYmaera X:

Theorem 5 (Correctness of implicit safeable regions) *The d$\mathcal{L}$ formula given in Eq. (9) is valid. That is as long as the advisories followed obey formula $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{impl}}$ there will be no NMAC.*

Before proving the equivalence of conditions $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{impl}}$ and $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{expl}}$, we prove a lemma allowing us to simplify cases 15 and 16 presented in Sect. 5.2, in the safeable case. Intuitively, cases 15 and 16 correspond to ensuring that the intruder is outside of the puck at time ε. However, in the safeable case, this is already ensured by ensuring that the intruder is outside of the initial time of the subsequent trajectory.

Lemma 5 (Simplification of cases 15 and 16) *If $w = \pm 1$, $r_p \geq 0$, $h_p > 0$, $r_v \geq 0$, $a_{\mathsf{lo}} > 0$, $a_{\mathsf{up}} \geq a_{\mathsf{lo}}$ and $\varepsilon \geq 0$:*

- *given h^{ex}_L and v^{ex}_L as defined in the definition of $L^{\mathsf{safeable}(\varepsilon)}_{\mathsf{expl}}$ in Fig. 12, then: $L^{-1}_{\mathsf{expl}}(r - r_v\varepsilon, h - h^{\mathsf{ex}}, v^{\mathsf{ex}}, w, v^{\mathsf{ex}}_{\mathsf{lo}}) \to \mathsf{case}^{\varepsilon}_{15}(r, v, w, v_{\mathsf{lo}}) \to \mathsf{bound}^{\varepsilon}_{15}(r, h, v, w, v_{\mathsf{lo}})$;*
- *given h^{ex}_U and v^{ex}_U as defined in the definition of $U^{\mathsf{safeable}(\varepsilon)}_{\mathsf{expl}}$ in Fig. 12, then: $L^{-1}_{\mathsf{expl}}(r - r_v\varepsilon, h - h^{\mathsf{ex}}, v^{\mathsf{ex}}, -w, v^{\mathsf{ex}}_{\mathsf{lo}}) \to \mathsf{case}^{\varepsilon}_{16}(r, v, w, v_{\mathsf{up}}) \to \mathsf{bound}^{\varepsilon}_{16}(r, h, v, w, v_{\mathsf{up}})$.*

Lemma 6 (Equivalence of explicit safeable regions) *If $w = \pm 1$, $r_p \geq 0$, $h_p > 0$, $r_v \geq 0$, $a_{\mathsf{lo}} > 0$, $a_{\mathsf{up}} \geq a_{\mathsf{lo}}$ and $\varepsilon \geq 0$, then the two conditions $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{impl}}$ and $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{expl}}$ are equivalent.*

Implicit formulation

$$
\begin{aligned}
&L_{\mathsf{impl}}^{\mathsf{safeable}(\varepsilon)}(r,h,v,w,v_{\mathsf{lo}},v_{\mathsf{lo}}^{\mathsf{ex}}) \equiv L_{\mathsf{impl}}^{\varepsilon}(r,h,v,w,v_{\mathsf{lo}}) \wedge \\
&\quad \Bigg(\forall h_L^{\mathsf{ex}}.\forall v_L^{\mathsf{ex}}.\Bigg(0 \le \varepsilon < \frac{\max(0,w(v_{\mathsf{lo}}-v))}{a_{\mathsf{lo}}} \wedge h_L^{\mathsf{ex}} = \frac{w a_{\mathsf{lo}}}{2}\varepsilon^2 + v_{\mathsf{lo}}\varepsilon \wedge v_L^{\mathsf{ex}} = w a_{\mathsf{lo}}\varepsilon + v \\
&\qquad\qquad \vee \;\; \varepsilon \ge \frac{\max(0,w(v_{\mathsf{lo}}-v))}{a_{\mathsf{lo}}} \wedge h_L^{\mathsf{ex}} = v_{\mathsf{lo}}\varepsilon - \frac{w\max(0,w(v_{\mathsf{lo}}-v))^2}{2a_{\mathsf{lo}}} \wedge v_L^{\mathsf{ex}} = v_{\mathsf{lo}}\Bigg) \\
&\qquad \rightarrow L_{\mathsf{impl}}^{-1}(r - r_v\varepsilon, h - h_L^{\mathsf{ex}}, v_L^{\mathsf{ex}}, w, v_{\mathsf{lo}}^{\mathsf{ex}})\Bigg) \\
&U_{\mathsf{impl}}^{\mathsf{safeable}(\varepsilon)}(r,h,v,w,v_{\mathsf{up}},v_{\mathsf{up}}^{\mathsf{ex}}) \equiv U_{\mathsf{impl}}^{\varepsilon}(r,h,v,w,v_{\mathsf{up}}) \wedge \\
&\quad \Bigg(\forall h_U^{\mathsf{ex}}.\forall v_U^{\mathsf{ex}}.\Bigg(0 \le \;\varepsilon < \frac{\max(0,w(v_{\mathsf{up}}-v))}{a_{\mathsf{up}}} \wedge h_U^{\mathsf{ex}} = \frac{w a_{\mathsf{up}}}{2}\varepsilon^2 + v_{\mathsf{up}}\varepsilon \wedge v_U^{\mathsf{ex}} = w a_{\mathsf{up}}\varepsilon + v \\
&\qquad\qquad \vee \;\; \varepsilon \ge \frac{\max(0,w(v_{\mathsf{up}}-v))}{a_{\mathsf{up}}} \wedge h_U^{\mathsf{ex}} = w\max(wv_{\mathsf{up}},wv)\varepsilon - \frac{w\mathrm{max}(0,w(v_{\mathsf{up}}-v))^2}{2a_{\mathsf{up}}} \wedge v_U^{\mathsf{ex}} = w\max(wv_{\mathsf{up}},wv)\Bigg) \\
&\qquad \rightarrow L_{\mathsf{impl}}^{-1}(r - r_v\varepsilon, h - h_U^{\mathsf{ex}}, v_U^{\mathsf{ex}}, -w, v_{\mathsf{up}}^{\mathsf{ex}})\Bigg) \\
&C_{\mathsf{impl}}^{\mathsf{safeable}(\varepsilon)}(r,h,v,w,v_{\mathsf{lo}},v_{\mathsf{lo}}^{\mathsf{ex}},v_{\mathsf{up}},v_{\mathsf{up}}^{\mathsf{ex}}) \equiv L_{\mathsf{impl}}^{\mathsf{safeable}(\varepsilon)}(r,h,v,w,v_{\mathsf{lo}},v_{\mathsf{lo}}^{\mathsf{ex}}) \vee U_{\mathsf{impl}}^{\mathsf{safeable}(\varepsilon)}(r,h,v,w,v_{\mathsf{up}},v_{\mathsf{up}}^{\mathsf{ex}})
\end{aligned}
$$

Explicit formulation

$$
\begin{aligned}
&L_{\mathsf{expl}}^{\mathsf{safeable}(\varepsilon)}(r,h,v,w,v_{\mathsf{lo}},v_{\mathsf{lo}}^{\mathsf{ex}}) \equiv L_{\mathsf{expl}}^{\varepsilon}(r,h,v,w,v_{\mathsf{lo}}) \wedge L_{\mathsf{expl}}^{-1}(r - r_v\varepsilon, h - h_L^{\mathsf{ex}}, v_L^{\mathsf{ex}}, w, v_{\mathsf{lo}}^{\mathsf{ex}}) \\
&\qquad \text{where} \begin{cases} h_L^{\mathsf{ex}} = \dfrac{w a_{\mathsf{lo}}}{2}\varepsilon^2 + v_{\mathsf{lo}}\varepsilon & \text{and} \quad v_L^{\mathsf{ex}} = w a_{\mathsf{lo}}\varepsilon + v \quad \text{if } 0 \le \varepsilon < \dfrac{\max(0,w(v_{\mathsf{lo}}-v))}{a_{\mathsf{lo}}} \\ h_L^{\mathsf{ex}} = v_{\mathsf{lo}}\varepsilon - \dfrac{w\max(0,w(v_{\mathsf{lo}}-v))^2}{2a_{\mathsf{lo}}} & \text{and} \quad v_L^{\mathsf{ex}} = v_{\mathsf{lo}} \quad \text{if } \varepsilon \ge \dfrac{\max(0,w(v_{\mathsf{lo}}-v))}{a_{\mathsf{lo}}} \end{cases} \\
&U_{\mathsf{expl}}^{\mathsf{safeable}(\varepsilon)}(r,h,v,w,v_{\mathsf{up}},v_{\mathsf{up}}^{\mathsf{ex}}) \equiv U_{\mathsf{expl}}^{\varepsilon}(r,h,v,w,v_{\mathsf{up}}) \wedge L_{\mathsf{expl}}^{-1}(r - r_v\varepsilon, h - h_U^{\mathsf{ex}}, v_U^{\mathsf{ex}}, -w, v_{\mathsf{up}}^{\mathsf{ex}}) \\
&\qquad \text{where} \begin{cases} h_U^{\mathsf{ex}} = \dfrac{w a_{\mathsf{up}}}{2}\varepsilon^2 + v_{\mathsf{up}}\varepsilon & \text{and} \quad v_U^{\mathsf{ex}} = w a_{\mathsf{up}}\varepsilon + v \quad \text{if } 0 \le \varepsilon < \dfrac{\max(0,w(v_{\mathsf{up}}-v))}{a_{\mathsf{up}}} \\ h_U^{\mathsf{ex}} = w\max(wv_{\mathsf{up}},wv)\varepsilon - \dfrac{w\mathrm{max}(0,w(v_{\mathsf{up}}-v))^2}{2a_{\mathsf{up}}} & \text{and} \quad v_U^{\mathsf{ex}} = w\max(wv_{\mathsf{up}},wv) \quad \text{if } \varepsilon \ge \dfrac{\max(0,w(v_{\mathsf{up}}-v))}{a_{\mathsf{up}}} \end{cases} \\
&C_{\mathsf{expl}}^{\mathsf{safeable}(\varepsilon)}(r,h,v,w,v_{\mathsf{lo}},v_{\mathsf{lo}}^{\mathsf{ex}},v_{\mathsf{up}},v_{\mathsf{up}}^{\mathsf{ex}}) \equiv L_{\mathsf{expl}}^{\mathsf{safeable}(\varepsilon)}(r,h,v,w,v_{\mathsf{lo}},v_{\mathsf{lo}}^{\mathsf{ex}}) \vee U_{\mathsf{expl}}^{\mathsf{safeable}(\varepsilon)}(r,h,v,w,v_{\mathsf{up}},v_{\mathsf{up}}^{\mathsf{ex}})
\end{aligned}
$$

Fig. 12 Implicit and explicit formulations of the safeable region

6 Reduction from 3D Dynamics to 2D Dynamics

In this section, we show that, with respect to our assumptions, any 3-dimensional encounter (Sect. 2) can be reduced to a 2-dimensional encounter (Sect. 3) without loss of generality. This could be done using a change of reference frame and a dimension reduction.

For the sake of clarity, let us put ourselves in a reference frame $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$ fixed to the ownship (O). In this 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 $x' = r_x$, $y' = r_y$, $(h')' = a$, where r_x, r_y and a remain constant as time evolves. The motion of the encounter can be decoupled into a 2-dimensional horizontal encounter in the reference frame $(O, \mathbf{i}, \mathbf{j})$ (horizontal plane) and a 1-dimensional vertical encounter in the reference frame $(O, \mathbf{k})$. 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.

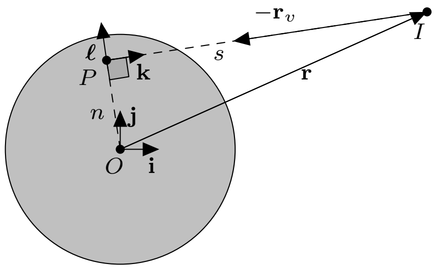


Fig. 13 Top view of the two reference frames

Fig. 13 depicts a top view of a generic encounter. We denote by $\mathbf{r}$ the position, and $\mathbf{r}_v$ the velocity, of the intruder relative to the ownship, and by $r_v \ge 0$ the norm of $\mathbf{r}_v$.

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

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

$$xr_x + yr_y = -sr_v \qquad xr_y - yr_x = nr_v \tag{10}$$

Given r_x and r_y, Eqns. (10) imply that the coordinates (x, y) are uniquely determined by the choice of (s, n), as long as $r_v \neq 0$ (using ${r_v}^2 = r_x^2 + r_y^2$). 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.

Next, we determine the radius of the dimension-reduced encounter, and prove that the absence of NMAC in $(O, \mathbf{i}, \mathbf{j})$, characterized by $r^2 > r_p^2$, is equivalent to the absence of NMAC in $(P, \mathbf{k}, \boldsymbol{\ell})$, characterized by $s^2 > s_p^2$. Using (10):

$$\begin{aligned} {r_v}^2 r^2 = {r_v}^2(x^2 + y^2) &= (xr_x + yr_y)^2 + (xr_y - yr_x)^2 \\ &= {r_v}^2(s^2 + n^2) \ . \end{aligned}$$

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 $\mathbf{r}_v$ 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 $(P, \mathbf{k}, \boldsymbol{\ell})$ becomes $s^2 \geq s_p^2$. When $\theta_v = 180°$, one has $s = r$, $n = 0$ and $s_p = r_p$ as in Sect. 3–4.

As the encounter evolves in $(O, \mathbf{i}, \mathbf{j})$ along $x' = r_x, y' = r_y$, its dimension-reduced version evolves in $(P, \mathbf{k}, \boldsymbol{\ell})$ along the differential equations $s' = -r_v, n' = 0$, obtained by differentiating Eqns. (10) 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 $(O, \mathbf{i}, \mathbf{j})$ is equivalent to the absence of an NMAC of radius s_p in $(P, \mathbf{k}, \boldsymbol{\ell})$.

Proposition 1 (Horizontal Reduction) *The following $d\mathcal{L}$ formula is valid*

$$\begin{aligned} &\big(xr_x + yr_y = -sr_v \land xr_y - yr_x = nr_v \land \\ &\ x^2 + y^2 = n^2 + s^2 \land {r_v}^2 = r_x^2 + r_y^2\big) \\ &\to [x' = r_x, y' = r_y, s' = -r_v, n' = 0] \\ &\qquad\qquad \big(x^2 + y^2 > r_p^2 \leftrightarrow s^2 > r_p^2 - n^2\big) \end{aligned} \tag{11}$$

Observe that the horizontal NMAC condition in $(P, \mathbf{k}, \boldsymbol{\ell})$ 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 P is no longer defined, and Eqns. (10) are trivially true. The variables s and n are constants ($s' = 0, n' = 0$), 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 two aircraft does not evolve over time, if the intruder is at a safe distance initially, the encounter is still safe for all time.

7 Discussion

7.1 Verification Process and Lessons Learned

The formal proofs in this paper were conducted while the KeYmaera X was still in active development. They use manually written tactics for differential dynamic logic decompositions [22] as well as Mathematica's implementation of real quantifier elimination [2]. The core parts of the proofs justify correctness of the safety regions (e.g., those in Fig. 12), which requires a deep understanding of the semantics of the different components of the hybrid model as well as their interactions. Due the high computational complexity of quantifier elimination (here 18 variables), the primary proof effort involved arithmetic simplifications.

A major lesson of the formal verification task carried out in this paper has been to use *both* implicit and explicit formulations of the safety conditions. The implicit formulation is better-suited for proving safety with respect to a model. The explicit formulation is better-suited for explicit testing and comparison to the ACAS X lookup table. Both formulations are formally proved equivalent. We initially proved the safe regions (Corollary 1) using the explicit formulation directly, which was significantly more difficult. We stress the fact that the explicit formulation could be theoretically obtained by quantifier elimination from the implicit one. Nevertheless, considering the current state-of-the-art of such automated algorithms, none is capable of automatically performing such task in a reasonable amount of time. Even if it were, the manually constructed explicit region are more readable since they follow the geometrical intuitions underlying the hybrid model.

7.2 Tightness of Conditions

The conditions L_{impl}^{-1} and L_{expl}^{-1} in Fig. 3, D_{impl}^{d} and D_{expl}^{d} in Fig. 5, and $C_{\mathsf{impl}}^{\mathsf{safeable}(\varepsilon)}$ and $C_{\mathsf{expl}}^{\mathsf{safeable}(\varepsilon)}$ in Fig. 12 specify conditions we have derived for safety under varying assumptions. While we have formally proved that each of these conditions is sufficient to guarantee safety (soundness) of the relevant models (cf. Theorem 1, Corollary 1, Theorem 2, Lemma 2, Theorem 5, and Lemma 6), we have not proved that these conditions are necessary for safety. This is often referred to as *completeness* of the conditions and measures how tight the used conditions overapproximate the actual exact safe regions. In other words, if an advisory and aircraft geometry meet the safety conditions, then the aircraft are guaranteed to be safe under the relevant assumptions. However, we have not proved that advisories that do not meet the conditions of the associated geometry can cause collisions.

In some cases, our conditions are strict overapproximations. For the conditions that do not account for subsequent advisories (*safe* conditions), L^{-1}_{impl}, L^{-1}_{expl}, D^{d}_{impl} and D^{d}_{expl}, consider the following geometry. The aircraft are diverging horizontally (e.g., $\theta_v = 0$ and $r_v > 0$), the intruder is sufficiently above the ownship in altitude, i.e., more than h_p above the ownship ($h > h_p$), and the aircraft are horizontally separated by exactly the radius of the puck, i.e., $r = r_p$. Intuitively, the intruder is directly above the left edge of the gray box in Fig. 4. If considering an up-sense advisory, this geometry does not pass L^{-1}_{expl} or D^{d}_{expl} because the conditions have no exception for intruders over the exact edge of the puck. However, an NMAC would only happen if the ownship were to accelerate upward at an infinite rate, so NMAC is not possible.

There are cases where advisories fail to meet the conditions for subsequent advisories (*safeable* conditions), but are safe under the relevant assumptions as well. Conditions $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{impl}}$ and $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{expl}}$ are built from a lower-bound trajectory and an upper-bound trajectory where, e.g., the lower-bound trajectory ends with an unbounded-time trajectory corresponding to the strongest possible upward subsequent advisory (vertical velocity v^{ex}). Such worst-case consideration leads to a reasonable overapproximation under the intuition that if the strongest upward subsequent advisory makes the lower-bound initial-trajectory safe, that subsequent advisory would also make any other initial-trajectory safe. An analogous reasoning supports the construction of the upper-bound trajectory. This simple analysis limits completeness as it implicitly assumes that the subsequent advisory is fixed, or determined at the time of the first advisory. That is, it asks if there exists *one* subsequent advisory *now* (at least either the most extreme upward or downward advisory) that can guarantee safety in the future. In reality, ACAS X chooses the subsequent advisory later in time, with some knowledge of the initial portion of the trajectory. In some cases, it is advantageous, for example, to choose the most extreme downward advisory for lower initial trajectories and to choose the most extreme upward advisory for upper initial trajectories. The result of this overapproximation is that ACAS X could choose a safe subsequent advisory for some geometries that cannot be concluded safeable by $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{impl}}$ or $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{expl}}$.

8 Comparison of Safety Theorems to ACAS X

The preceding theorems about safety are independent of the tested system. To characterize the safety of ACAS X itself, we check whether advisories indicated by the core component of ACAS X, consisting of the timing and logic tables, meet the conditions of the theorems. The timing and logic tables of ACAS X contain scores for each advisory for a set of discrete sample states. Each score quantifies the desirability of issuing the corresponding advisory when in the corresponding state and is the result of optimizing a Markov decision process (MDP). In practice, ACAS X multilinearly interpolates advisory scores from the values stored in the tables, given estimates of aircraft states. Although these estimates model uncertainty in practice, we check behavior for the sample points themselves, assuming perfect knowledge of state, to compare system behavior under best-case circumstances. Table 2 shows the range of sample state points, called *cut-points*, for each of the 7 dimensions of the logic tables. These non-uniformly sampled cut-points were chosen by the ACAS X designers to maximize system performance for realistic encounters while keeping the size of the tables modest. The previous advisory state includes the previously-issued advisory and information about whether the pilot was acting to comply with the advisory at the previous time step. This information is used in the MDP state for the multi-step optimization. Together these samples make up over 648 billion state combinations for which scores are explicitly stored in the ACAS X logic tables.

Corollary 1 and Lemma 6 along with Theorem 5 reduce the safety of all future trajectories to a static condition on the current state: L^{-1}_{expl} or $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{expl}}$, respectively. We examine the advisory specified by the logic tables at each of the 648 billion state combinations for which scores are explicitly stored and check the advisory against the respective *safe* or *safeable* condition for that advisory.

One limitation of the comparison is that it assumes an omniscient observer, with full exact knowledge of the positions and velocities of both aircraft. In practice, this is unrealistic because of sensor inaccuracies. Furthermore, the sampling-based approach presented in this section constitutes a (thorough) test of the ACAS X logic tables, but not a verification, as we provide no guarantees on values interpolated between cut-points. We plan to address these limitations in future work.

Approach. To perform the comparison we use a 10-node cluster, each with 48 cores and 128 GB RAM. Checks for the relevant conditions are implemented in Julia [3] and run on the 648 billion cut-point combinations in parallel, taking approximately 8 days. We first define what we mean by a *counterexample*.

[3] `http://julialang.org/`.

parameter	**samples**	**range of values**
relative alt. h (ft)	45	[-8000, -4000, -3600, -3200, -2800, -2400, -2000, ..., -100, -50, 0, 50, ..., 4000, 8000]
ownship alt. rate v (ft/s)	25	[-166.7, -83.3, -75.0, -66.7, -58.3, -50, ..., -16.7, -8.3, -4.2, 0, 4.2, ..., 83.3, 166.7]
intruder alt. rate v_I (ft/s)	25	[-166.7, -83.3, -75.0, -66.7, -58.3, -50, ..., -16.7, -8.3, -4.2, 0, 4.2, ..., 83.3, 166.7]
previous advisory	33	{ None-None, DNC2000-None, DNC2000-DNC2000, ..., SCL2500-None, SCL2500-SCL2500 }
range r (ft)	101	[0, 50, 100, 150, 200, 250, 300, ..., 1000, 1500, 2000, ..., 39500, 40000, 100000, 200000]
relative velocity r_v (ft/s)	187	[0, 10, 20, 30, 40, 50, 60, 70, 80, 90 ..., 1700, 1750, 1800, ..., 2350, 2400, 2450, 2500]
velocity angle θ_v (deg)	37	[0, 5, 10, 15, 20, 25, 30, 35, 40, 45, ..., 145, 150, 155, 160, 165, 170, 175, 180]

Table 2 Discrete parameter ranges and number of cut-points specified by ACAS X run 13 and checked during comparison

(δ, ε) *in (seconds, seconds)*	$(a^{\mathsf{d}}, \varDelta a = a_{\mathsf{up}} - a_{\mathsf{lo}})$
$(0, 1)$	$(g/10^6, 0)$
$(1, 2)$	$(g/6, g/12)$
$(3, 5)$	$(g/3, g/6)$
$(5, 7)$	$(g/2, g/4)$

Table 3 Delay and Acceleration Parameters Swept During Logic Comparison. The 16 combinations of these pairs are tested.

Definition 2 (Counterexample) We say that a state is a *counterexample* for the safe (resp. safeable) conditions if the advisory given by the ACAS X logic tables for that state violates L^{-1}_{expl} given in Fig. 3 (resp. $C^{\mathsf{safeable}(\varepsilon)}_{\mathsf{expl}}$ given in Fig. 12) and for which there exists an alternative advisory that does satisfy the respective safety conditions.

Note that since some conditions are not tight (Sect. 7.2), not all counterexamples constitute a safety threat. We say a state is *non-safe*, or *non-safeable* if it is a *counterexample* for the safe or safeable conditions respectively.

We selected a set of parameters for acceleration and delay ranges, shown in Table 3. The delays are parameterized by two values δ and ε. The first, δ, is the delay from time 0 to the time at which the pilot begins adjusting vertical acceleration to follow the first advisory. The second delay parameter, ε, is the delay from time 0 to the time at which the second issued advisory begins to be followed. Thus the first advisory is followed for $\varepsilon - \delta$ seconds and we require that $\varepsilon > \delta$. The selections of delay parameters in Table 3 have ε values that are at least 1 second greater than δ to allow a period of 1 second or more of compliance with a first advisory. The right side of Table 3 gives the parameters governing the limits of ownship acceleration. The free acceleration, a^{d}, as previously defined, is the maximum absolute acceleration of the pilot during delay or COC; it is swept between almost zero acceleration and $g/2$. (We used $g/10^6$ for almost zero to avoid divisions by zero in our conditions.) The second parameter, $\varDelta a$, the overcompliance acceleration, is the difference between the upper limit of acceleration under the advisory, a_{up}, and the lower compliance limit, a_{lo}. It is tied in each case to one half of the free acceleration a^{d}. The delay parameter pairs are also tied together during each query, meaning that during the first query $(\delta, \varepsilon) = (0, 1)$ and for the second query $(\delta, \varepsilon) = (0, 2)$, and so on. All 16 combinations of delay pairs and acceleration pairs are tested against the total 648 billion cut-point combinations summarized in Table 2.

Safe Counterexamples				
(in billions)	delay (δ, ε) *in seconds*			
acceleration $(a^{\mathsf{d}}, \varDelta a)$	(0, 1)	(1, 2)	(3, 5)	(5, 7)
$(g/10^6, 0)$	15.38	15.05	14.85	14.79
$(g/6, g/12)$	15.40	15.03	14.87	14.88
$(g/3, g/6)$	15.43	15.02	14.95	15.14
$(g/2, g/4)$	15.45	15.01	15.18	15.33
Unresolvable States				
(in billions)	delay (δ, ε) *in seconds*			
acceleration $(a^{\mathsf{d}}, \varDelta a)$	(0, 1)	(1, 2)	(3, 5)	(5, 7)
$(g/10^6, 0)$	10.42	10.64	10.84	10.96
$(g/6, g/12)$	10.42	10.75	11.16	11.51
$(g/3, g/6)$	10.42	10.88	11.59	12.36
$(g/2, g/4)$	10.42	11.04	12.13	13.46

Table 4 Number of *Safe* counterexamples (top) and number of unresolvable states (bottom) for each delay and acceleration parameter set

Safeable				
Counterexamples				
(in millions)	delay (δ, ε) *in seconds*			
acceleration $(a^{\mathsf{d}}, \varDelta a)$	(0, 1)	(1, 2)	(3, 5)	(5, 7)
$(g/10^6, 0)$	266.3	99.7	39.2	31.6
$(g/6, g/12)$	310.0	117.2	101.4	164.9
$(g/3, g/6)$	353.2	138.6	277.7	544.7
$(g/2, g/4)$	398.9	163.6	668.8	898.7
Unresolvable States				
(in billions)	delay (δ, ε) *in seconds*			
acceleration $(a^{\mathsf{d}}, \varDelta a)$	(0, 1)	(1, 2)	(3, 5)	(5, 7)
$(g/10^6, 0)$	11.03	11.27	11.51	11.64
$(g/6, g/12)$	11.03	11.40	11.85	12.23
$(g/3, g/6)$	11.03	11.54	12.30	13.09
$(g/2, g/4)$	11.03	11.71	12.86	14.23

Table 5 Number of *Safeable* counterexamples (top) and unresolvable states (bottom) for each delay and acceleration parameter set

Counterexamples. Tables 4 and 5 summarize the number of counterexamples (Def. 2) we found. The lower portions of the tables show the number of initial states tested that were not resolvable with any of the available safe or safeable actions. That is, for those unresolvable states, there are no available advisories that pass the conditions we formally proved. For the *safe* comparison, the checks we perform are limited to states in the ACAS X tables where COC is the previous advisory (about 196 billion states). In addition, the parameters ε and Δa are not used in the safe conditions: ε is fixed to -1 as we assume that the pilot will follow the issued advisory forever; Δa is not used as we assume that the vertical acceleration has only to respect a minimal rate (a_{lo}) to satisfy the advisory. Thus, only δ and a^{d} vary. One can notice that taking into account the pilot delay in the formal model reduces the number of counterexamples. This may seem at first counterintuitive as one expects less unsafe advisories if the pilot responds immediately. However, given that the previous advisory is a COC, the ACAS X tables are designed to not necessarily issue a disruptive advisory right away and will rather either issue a preventive advisory (such as DNC or DND) or simply a MAINTAIN before actually strengthening those advisories in the future if needed. Those first advisories may indeed violate the safe conditions although the system is able to resolve the potential encounters. We also observe, on the contrary, that increasing the free acceleration a^{d} results in more counterexamples and more unresolvable encounters. This is a straightforward effect of our worst-case analysis: if we allow $g/2$, say, as worst-case vertical acceleration, then we have to consider that the pilot will actually accelerate at $g/2$ during the delay, which may be unrealistic.

Overall, as shown in Table 4, we found billions of counterexamples to Corollary 1 (*safe* regions). Many were used to create test encounters and tested in the full system as a means of targeted stress testing. As alluded to earlier, the ACAS X system was able to resolve many of those by issuing subsequent advisories, which motivated the safeable extension we developed in this paper. Indeed, *safeable* counterexamples represent states where the advice given by ACAS X may not be correctable by subsequent advisories, although an alternative guaranteed safe sequence exists for our assumptions according to Lemma 6.

We can see from Table 5 that there are considerably fewer safeable counterexamples than safe, hundreds of millions instead of tens of billions. Theorem 5 (*safeable* regions) is designed to detect points of no return. As with safe regions, we see that the number of counterexamples tend to have an inverse relationship with pilot response delay although an exception to this trend occurs when delay is exactly 0. We believe this exception is caused by the fact that many states are resolvable when there is no pilot response delay, i.e., when the pilot responds immediately to the issued advice. Once a moderate response delay is assumed, there are many fewer resolvable initial states, resulting in fewer total counterexamples, albeit still a few hundred million. These *safeable* counterexamples are also being used to generate stressing short-time encounters that may be used for robustness testing.

Safety Analysis. The comparison of the system to *safe* condition D^{-1}_{expl} (valid by Theorem 2) gave insight into possible improvements for ACAS X. Our analysis led to the identification of unexpected behavior in the ACAS X run 13 (i.e., version 13) lookup tables. In some cases, the ACAS X advisory seems to *induce* an NMAC, i.e., if the initial advisory is followed and not strengthened or reverted later, an NMAC will occur when it would not have occurred if the aircraft continued flying straight. A typical example, found during checking against Corollary 1 (*safe* regions) with $\delta = 0$ and $a^{\mathsf{d}} = g/2$, is shown in Fig. 14. The ownship is flying from the left and the intruder from the right. The time 0 corresponds to the time of closest horizontal approach. As time progresses, the intruder flies towards the ownship and an NMAC happens near the time $t = 0$. The original path of the ownship does not lead to an NMAC. However, ACAS X issues a Do-Not-Climb advisory. If the pilot follows this advisory immediately and stops climbing, and if the initial advisory is not subsequently strengthened or reversed, an NMAC will occur.

In other cases of counterexamples to Corollary 1 (*safe* regions), the advisory does not seem to have any benefit. In those cases, flying at the vertical rates disallowed by the advisory would actually avoid NMAC, while not all allowed vertical rates of the advisory are safe.

Some safe counterexamples are tolerated, as ACAS X tries to minimize alerting the pilot unless it has to do so; for these cases, ACAS X will issue or strengthen an advisory later to avoid issuing a disruptive alert immediately. Additionally, the assumption of straight vertical flight is not always valid. E.g., aircraft may actually be more likely to level-off than continue at high vertical rates in some cases. The particular unexpected behavior shown in Fig. 14 was independently identified by the ACAS X team using simulation-based testing, and is being addressed in a subsequent revision of the system.

Safeable Analysis. In Fig. 15, we see an automatically discovered *safeable* counterexample. The pilot is assumed to start complying with the initially issued advisory at 5 seconds and will only begin complying with a potential subsequent advisory after 7 seconds from the initial time (the first advisory will thus be followed for 2 seconds). For this state, the ACAS X tables issue the advisory MDES, maintain descent. The upper panel shows the most extreme upper and lower paths that a subsequent advisory can restrict the ownship's motion to. Neither of these can avoid NMAC with the intruder. This is a relatively close range, slow closing encounter, as the initial horizontal separation is $1,500$ ft and

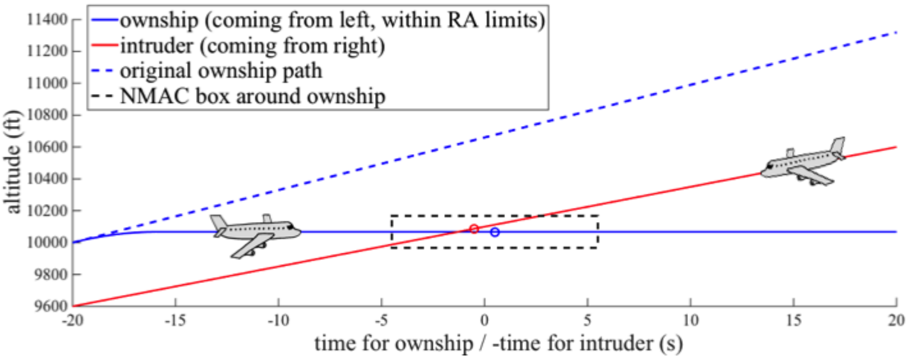


Fig. 14 *Safe* Counterexample. Original ownship path (cyan) and intruder path (red) vs. ownship responding to a do-not-climb (DNC) advisory issued by the ACAS X tables in starting state: $r = 4{,}000\,\mathrm{ft}$, $r_v = 200\,\mathrm{ft/s}$, $\theta_v = 180°$, $h = 600\,\mathrm{ft}$, $v = 1{,}980\,\mathrm{ft/min}$, $v_I = -1{,}500\,\mathrm{ft/min}$. Time is shown counting from -20 s to time 0; the time of closest horizontal approach. The 2D projection of the NMAC cylinder is shown centered around the ownship as a dashed rectangle; the intruder intersecting with the NMAC region is shown by a red circle.

horizontal relative closing speed is only 90 ft/second. Both aircraft begin descending at moderate rates, offset vertically by 300 ft. The problem occurs because the most extreme descend advisory available has a rate of only 2500 ft/min and with the ownship already descending at 2000 ft/min the increase in descent rate cannot avoid the intruder. In this slow closing geometry, the two aircraft will be in horizontal conflict range for an extended period, which at this geometry lasts for 7 seconds starting at $t = -3.5$ to $t = 3.5$ seconds. In this case, Lemma 6 tells us that *safeable* actions include Do-Not-Descend and Climb-1500. The lower panel of Fig. 15 shows that if the ownship were, instead, told to climb at the maximum rates allowed by a sequence of up-sense advisories, this situation would have been resolved, and so is indeed a counterexample.

The smaller number of *safeable* counterexamples are of special interest to system verification, as they are situations that cannot even be corrected by subsequent advisories. To aid in robustness testing and tuning of the actual system, we have created a set of short-time encounters based on the counterexamples found using the *safeable* analysis and are sharing these encounters with the ACAS X designers.

Overall, we have begun analyzing ACAS X using our theorem and are identifying numerous valuable outcomes. These results either help us characterize tradeoffs being made or help us identify undesirable behaviors in the system. As one of our next steps, we aim to prove that ACAS X gives safe advice for continuous regions of the state space. When comparisons are 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.

9 Related Work

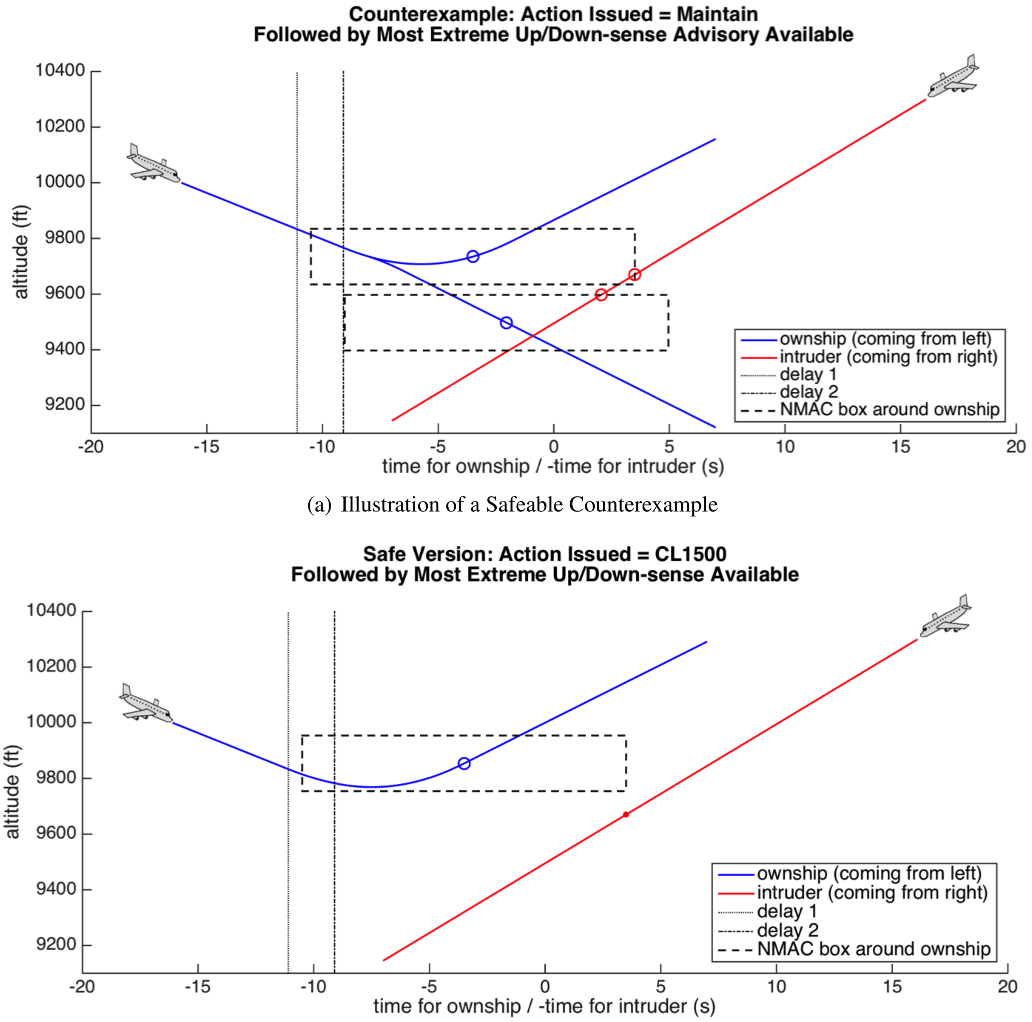
Kochenderfer and Chryssanthacopoulos [14] describe the design of the ACAS X lookup-tables. Their principled approach, based on optimizing an MDP, guarantees the selection of optimal advisories according to a cost model. The state space and dynamics are discretized. Their notion of optimality depends on costs assigned to various events.

Holland *et al.* [12] and Chludzinski [1] 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.

Von Essen and Giannakopoulou [4] use probabilistic model-checking to analyze an MDP based on [14]. They investigate the probability of several undesirable events occurring. Because they ostensibly analyze an MDP, their work inherits many of the assumptions of ACAS X, including errors due to discretized dynamics. Their analysis depends heavily on the MDP considered and thus needs to be redone in full on every version of ACAS X.

Lygeros and Lynch [19] use hybrid techniques to formally verify the TCAS conflict resolution algorithms. They assume—rather than prove—that TCAS ends up in a state where one aircraft has a climbing advisory and the other a descending advisory. They then prove (by hand) a lower bound on the vertical separation of both aircraft at the point of closest approach. In contrast, we do prove as opposed to assume that and when advisories are safe.

Tomlin *et al.* [25], Platzer and Clarke [24], Loos *et al.* [18] and Ghorbal *et al.* [11] use hybrid systems approaches to design safe horizontal maneuvers for collision avoidance. Dowek *et al.* [3] and Galdino *et al.* [10] describe and verify in the PVS theorem prover a collision avoidance system of their design called KB3D.



(a) Illustration of a Safeable Counterexample

(b) The Counterexample Under a Safeable First Advisory

Fig. 15 *Safeable* counterexample, where "delay 1" $(\delta) = 5$ s, "delay 2" $(\varepsilon) = 7$ s, $a^{\mathrm{d}} = g/10^6$, and $\varDelta a = 0$. Action issued is "maintain" for the initial state $r = 1,500$ ft, $r_v = 90$ ft/s, $\theta_v = 2.88$ rad, $h = 300$ ft, $v = -33.33$ ft/s, $v_I = -50$ ft/s, `previous advisory` $=$ None. Plots show absolute altitude of ownship vs. time and intruder vs. negative time; ownship travels left to right and the intruder right to left. Time is shown counting from -16 s to time 0; the time of closest horizontal approach. The delay times are shown in vertical dashed lines. The 2D projection of the NMAC cylinder is shown centered around the ownship as a dashed rectangle. When the intruder intersects with the NMAC region, it is shown by a red circle, and when the intruder misses the NMAC region it is shown as a red dot.

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

- unlike [4,14], we rely on an independent model from the one used to design ACAS X;
- unlike [3,10,11,18,24,25] we analyze an independent industrial system and not a safe-by-design system;
- unlike [3,4,10] our analysis uses realistic, continuous dynamics;
- unlike [19,25,4] we provide universal safe regions that can be reused for new versions of ACAS X or new systems;
- unlike [1,12,15,19,25], we provide mechanized rigorous proofs of correctness of our model.

10 Conclusion and Future Work

We developed a general strategy for analyzing the safety of complicated, real-world aircraft collision avoidance systems, and applied it to ACAS X. Our strategy identifies safe regions where an advisory is proved to always keep the aircraft clear of NMAC, under the assumptions of the model. We identified states where ACAS X is provably safe, and delivered others showing unexpected behaviors back to the ACAS X development team. The identified safe regions are independent from the version of ACAS X and can thus be reused for future versions. In future work, we plan to extend our hybrid model to account for curved trajectories of both aircraft, vertical acceleration of the intruder, and sensor uncertainties. We would also like to extend our table comparison to provide safety guarantees on interpolated values.

Acknowledgments. This research was conducted under the sponsorship of the Federal Aviation Administration Traffic Alert & Collision Avoidance System (TCAS) Program Office (PO) AJM-233 under contract number DTFAWA-11-C-00074. Additionally, support for the basic verification technology used as a foundation for this research was provided by the National Science Foundation under NSF CAREER Award CNS-1054246.

The authors would like to warmly thank Neal Suchy for his lead of the ACAS X project and support of this work, as well as Nathan Fulton, Ran Ji and Jan-David Quesel for their support of the KeYmaera X tool. The authors would also like to thank Jeff Brush, Barbara Chludzinski, Dane Fichter, Jessica Holland, Robert Klaus, Barbara Kobzik-Juul, Mykel Kochenderfer, Ted Londner, Sarah Loos, Jessica Lopez, Ed Morehouse, Wes Olson, Michael Owen, Anshu Saksena, Joshua Silbermann, the ACAS X development team, and the anonymous reviewers for their interesting feedback.

References

1. Chludzinski, B.J.: Evaluation of TCAS II version 7.1 using the FAA fast-time encounter generator model. Tech. Rep. ATC-346, MIT Lincoln Laboratory (2009)
2. Collins, G.E.: Hauptvortrag: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In: Automata Theory and Formal Languages, *LNCS*, vol. 33, pp. 134–183. Springer (1975)
3. Dowek, G., Muñoz, C., Carreño, V.: Provably safe coordinated strategy for distributed conflict resolution. In: AIAA Guidance Navigation, and Control Conference and Exhibit (2005)
4. von Essen, C., Giannakopoulou, D.: Analyzing the next generation airborne collision avoidance system. In: TACAS, *LNCS*, vol. 8413, pp. 620–635. Springer (2014).
5. Federal Aviation Administration: Introduction to TCAS II (2011). Version 7.1
6. Federal Aviation Administration TCAS Program Office: Algorithm design description for the surveillance and tracking module of ACAS X (2014). Run12
7. Federal Aviation Administration TCAS Program Office: Algorithm design description for the threat resolution module of ACAS X (2014). Version 3 Rev. 1
8. Felty, A., Middeldorp, A. (eds.): International Conference on Automated Deduction, CADE'15, Berlin, Germany, Proceedings, *LNCS*, vol. 9195. Springer (2015)
9. Fulton, N., Mitsch, S., Quesel, J.D., Völp, M., Platzer, A.: KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In: Felty and Middeldorp [8], pp. 527–538.
10. Galdino, A., Muñoz, C., Ayala, M.: Formal verification of an optimal air traffic conflict resolution and recovery algorithm. In: WoLLIC, *LNCS*, vol. 4576. Springer (2007)
11. Ghorbal, K., Jeannin, J.B., Zawadzki, E., Platzer, A., Gordon, G.J., Capell, P.: Hybrid theorem proving of aerospace systems: Applications and challenges. Journal of Aerospace Information Systems (2014)
12. 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 Quarterly (2014)
13. 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: C. Baier, C. Tinelli (eds.) TACAS, *LNCS*, vol. 9035, pp. 21–36. Springer (2015).
14. Kochenderfer, M.J., Chryssanthacopoulos, J.P.: Robust airborne collision avoidance through dynamic programming. Tech. Rep. ATC-371, MIT Lincoln Laboratory (2010)
15. Kochenderfer, M.J., Espindle, L.P., Kuchar, J.K., Griffith, J.D.: Correlated encounter model for cooperative aircraft in the national airspace system version 1.0. Tech. Rep. ATC-344, MIT Lincoln Laboratory (2008)
16. Kochenderfer, M.J., Holland, J.E., Chryssanthacopoulos, J.P.: Next generation airborne collision avoidance system. Lincoln Laboratory Journal **19**(1), 17–33 (2012)
17. Kochenderfer, M.J., Monath, N.: Compression of optimal value functions for Markov decision processes. In: Data Compression Conference. Snowbird, Utah (2013)
18. Loos, S.M., Renshaw, D.W., Platzer, A.: Formal verification of distributed aircraft controllers. In: HSCC, pp. 125–130. ACM (2013).
19. Lygeros, J., Lynch, N.: On the formal verification of the TCAS conflict resolution algorithms. In: IEEE Decision and Control, vol. 2, pp. 1829–1834. IEEE (1997).
20. Platzer, A.: Differential dynamic logic for hybrid systems. J. Autom. Reas. **41**(2), 143–189 (2008).
21. Platzer, A.: Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics. Springer (2010).
22. Platzer, A.: Logics of dynamical systems. In: LICS, pp. 13–24. IEEE (2012).
23. Platzer, A.: A uniform substitution calculus for differential dynamic logic. In: Felty and Middeldorp [8], pp. 467–481.
24. Platzer, A., Clarke, E.M.: Formal verification of curved flight collision avoidance maneuvers: A case study. In: FM, *LNCS*, vol. 5850, pp. 547–562. Springer (2009).
25. Tomlin, C., Pappas, G.J., Sastry, S.: Conflict resolution for air traffic management: A study in multiagent hybrid systems. Automatic Control, IEEE Transactions on **43**(4), 509–521 (1998)