Formally Verified Next-generation Airborne Collision Avoidance Games in ACAS X

RACHEL CLEAVELAND, STEFAN MITSCH, and ANDRÉ PLATZER, Carnegie Mellon University, USA

10

The design of aircraft collision avoidance algorithms is a subtle but important challenge that merits the need for provable safety guarantees. Obtaining such guarantees is nontrivial given the unpredictability of the interplay of the intruder aircraft decisions, the ownship pilot reactions, and the subtlety of the continuous motion dynamics of aircraft. Existing collision avoidance systems, such as TCAS and the Next-Generation Airborne Collision Avoidance System ACAS X, have been analyzed assuming severe restrictions on the intruder's flight maneuvers, limiting their safety guarantees in real-world scenarios where the intruder may change its course.

This work takes a conceptually significant and practically relevant departure from existing ACAS X models by generalizing them to hybrid games with first-class representations of the ownship and intruder decisions coming from two independent players, enabling significantly advanced predictive power. By proving the existence of winning strategies for the resulting Adversarial ACAS X in differential game logic, collision-freedom is established for the rich encounters of ownship and intruder aircraft with independent decisions along differential equations for flight paths with evolving vertical/horizontal velocities. We present three classes of models of increasing complexity: single-advisory infinite-time models, bounded time models, and infinite time, multi-advisory models. Within each class of models, we identify symbolic conditions and prove that there then always is a possible ownship maneuver that will prevent a collision between the two aircraft.

CCS Concepts: • **Theory of computation** → **Timed and hybrid models**; **Modal and temporal logics**; **Programming logic**; • **Computer systems organization** → *Embedded systems;*

Additional Key Words and Phrases: Airborne Collision Avoidance, ACAS X, theorem proving, hybrid games, differential game logic

ACM Reference format:
Rachel Cleaveland, Stefan Mitsch, and André Platzer. 2022. Formally Verified Next-generation Airborne Collision Avoidance Games in ACAS X. *ACM Trans. Embedd. Comput. Syst.* 22, 1, Article 10 (October 2022), 30 pages.
https://doi.org/10.1145/3544970

1 INTRODUCTION

Mid-air aircraft collisions are a fundamental responsibility of pilots and air traffic controllers to avoid, but their likelihood only increases as air space gets more congested and Unmanned Aerial Vehicles become more prevalent. The first onboard collision avoidance system, known as **Traffic Alert and Collision Avoidance System (TCAS)**, was developed in the 1970s and has successfully

This research was sponsored by the AFOSR under grant number FA9550-16-1-0288.
Authors' address: R. Cleaveland, S. Mitsch, and A. Platzer, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, 15213; emails: rcleavel@andrew.cmu.edu, {smitsch, aplatzer}@cs.cmu.edu.

1539-9087/2022/10-ART10
https://doi.org/10.1145/3544970

prevented several mid-air collisions. However, this system is not perfect; one particular failure of TCAS occurred in the 2002 Überlingen crash, where two airplanes collided despite having received instructions by their TCAS systems onboard. Tragedies like this underscore the importance of continued research into developing and formally verifying onboard collision avoidance systems.

Most of the time, when aircraft are on a collision course, they are detected and resolved in advance by either the pilots or flight directors of the Air Route Traffic Control Centers. However, in rare scenarios where conflicting flight paths were not detected early enough and two aircraft are on an immediate collision course, collision avoidance maneuvers must be performed as a last resort. With little time to determine and perform the necessary maneuvers to avoid collision, it is imperative to ensure the safety of these collision avoidance maneuvers in advance using formal verification under all reasonable flight circumstances to ensure that no mid-air collisions happen.

The TCAS, and the more recent ACAS X, collision avoidance systems developed by the **Federal Aviation Administration (FAA)** give vertical ascent/descent advisories when an aircraft is encountering an intruder with which it is at risk of colliding [15]. The goal of ACAS X is to prevent **Near Mid-Air Collisions (*NMACs*)**, dangerous situations where two aircraft come within $r_p = 500$ ft horizontally and $h_p = 100$ ft vertically of each other [15]. These variables r_p and h_p describe the radius and height, respectively, of a puck surrounding the aircraft, into which no other aircraft should enter.

Previous work explores formal verification of ACAS X when the intruder aircraft is moving at a constant horizontal and vertical velocity [11]. This assumption is rigid and does not take into account the potential maneuvers that the intruder may perform. This article takes a conceptually significant departure by generalizing formally verified ACAS X models from hybrid *systems* to *hybrid games*, owing to the fundamental observation that, despite best intent, the actions of the ownship and intruder aircraft may interfere with one another, since they are resolved by different pilots with different situational awareness facing a challenging safety hazard. This generalization is of practical relevance for the predictive power of verified ACAS X models but requires a fundamental shift in reasoning using differential game logic for hybrid games [24, 26, 27]. Hybrid systems are fundamentally single-player. Only hybrid games can faithfully represent a dynamics where different pilots of different aircraft may independently reach different decisions at different times with different consequences on the flight of the two aircraft. While all pilots share the intent of avoiding collisions, only hybrid games accurately reflect that their decisions may, nevertheless, interfere, because the pilots chose different means to avoid collisions that may conflict.

1.1 Airborne Collision Avoidance System ACAS X

ACAS X tracks the position and velocity of the ownship and intruders in its vicinity using a variety of sensors to compute its collision avoidance advisories [15]. An advisory alerts the pilot with an audio-visual message and requests that she either maintain her vertical speed or accelerate towards a new desired vertical speed. An advisory is issued only when a potential collision is identified, otherwise the system stays quiet to avoid distracting the pilot [15]. Advisories apply only in the vertical direction, not the horizontal direction, and only apply to the aircraft's climb rate.

Table 1 gives all of the 16 possible advisories issued by ACAS X, plus **Clear-of-Conflict (COC)**, which indicates that no action is necessary. These advisories vary in the extremeness of the action; a less extreme advisory like **Do Not Descend (DND)** only requires that the ownship does not, as the name suggests, descend past its current altitude. A more extreme advisory like SCL2500 requires the ownship to reach a climb rate of at least 2,500 ft/min. Advisories can also be either lower bounds, like SCL2500, or upper bounds, like DNC2000, which requires that the ownship not exceed a climb rate of more than 2,000 ft/min. The FAA assumes that for the pilot to achieve the

Table 1. ACAS X Advisories and Their Parameters as Summarized Elsewhere [11]

		ACAS X Specification [14]				Model [11]	
		Vertical 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	$-\infty$	+2,000	$g/4$	5	−1	+2,000
DND2000	Do Not Descend at more than 2,000 ft/min	−2,000	$+\infty$	$g/4$	5	+1	−2,000
DNC1000	Do Not Climb at more than 1,000 ft/min	$-\infty$	+1,000	$g/4$	5	−1	+1,000
DND1000	Do Not Descend at more than 1,000 ft/min	−1,000	$+\infty$	$g/4$	5	+1	−1,000
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 Descent at at least current rate	$-\infty$	current	$g/4$	5	−1	current
MCL	Maintain Climb at at least current rate	current	$+\infty$	$g/4$	5	+1	current
DES1500	Descend at at least 1,500 ft/min	∞	−1,500	$g/4$	5	−1	−1,500
CL1500	Climb at at least 1,500 ft/min	+1,500	$+\infty$	$g/4$	5	+1	+1,500
SDES1500	Strengthen Descent to at least 1,500 ft/min	$-\infty$	−1,500	$g/3$	3	−1	−1,500
SCL1500	Strengthen Climb to at least 1,500 ft/min	+1,500	$+\infty$	$g/3$	3	+1	+1,500
SDES2500	Strengthen Descent to at least 2,500 ft/min	$-\infty$	−2,500	$g/3$	3	−1	−2,500
SCL2500	Strengthen Climb to at least 2,500 ft/min	+2,500	$+\infty$	$g/3$	3	+1	+2,500
COC	Clear of Conflict	$-\infty$	$+\infty$	NA	NA	NA	NA
MTLO	Multi-Threat Level-Off	NA	NA	NA	NA	NA	NA

desired climb or descend rate, she does so by following a vertical acceleration of strength at least $g/4$ (referred to as the positive constant a_{lo}) [16], an assumption that will be pertinent later.

These advisories result from an estimation of the pilot's optimal course of action, calculated by linearly interpolating a precomputed table of scores for various actions. The domain of this table includes parameters describing the state of the encounter, while its range gives scores for each possible action [16]. This table is constructed from a Markov Decision Process, which approximates the dynamics of the system on a discretized grid of the state space. From there, dynamic programming is used to optimize the table through maximizing the expected value of each event over all future outcomes for each action [16]. These expected values approximately map to different outcomes: **Near Mid-Air Collisions (NMACs)**, for example, correspond to large negative values, while issuing advisories corresponds to small negative values. The ACAS X system then uses a multilinear interpolation of grid points and heuristics to choose the action with the greatest expected value given the particular circumstances surrounding the ownship's current flight conditions.

1.2 Formally Verified Safe Regions and Hybrid Game Logic

Previous work [10, 11] applied hybrid systems to the formal verification process, a natural application given the combination of discrete advisories and continuous dynamics of an aircraft using ACAS X. While direct verification of the ACAS X implementation is infeasible given the complexity of ACAS X (whose core lookup table defines 29,212,664 interpolation regions in a five-dimensional state-space giving rise to at least half a trillion cases to consider), References [10, 11] cut down the complexity with the concept of *safe regions*. A region is proven safe if for all possible ownship positions and velocities within the region, an NMAC with the intruder will never occur. Thus, if ACAS X issues an advisory, and following this advisory in any permitted way always keeps the ownship within the safe region, then this advisory is guaranteed to maintain safety. These regions comprise fully symbolic parameters like a_{lo}, making them easily adaptable to new ACAS X versions.

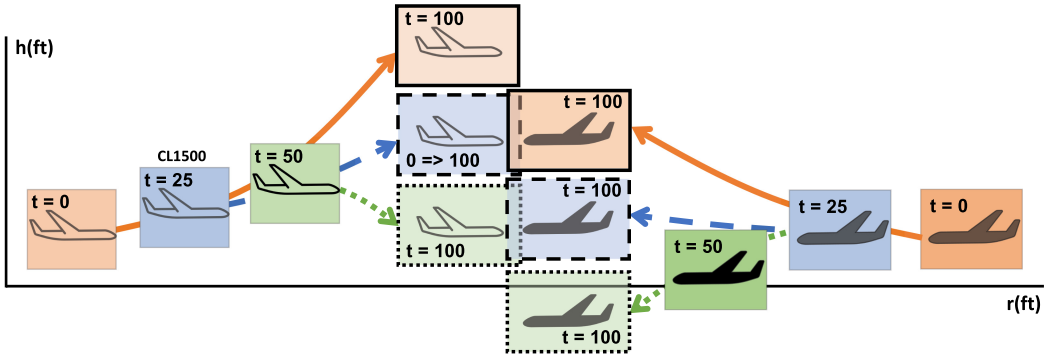


Fig. 1. An encounter between the ownship and example reactions to an intruder that interferes (orange solid trajectories), is somewhat cooperative (blue dashed trajectories), or helps resolve the conflict (green dotted trajectories).

In this work, we continue with the identification of safe regions to prove safety of the overall system, but we make the important change of applying *hybrid games*, rather than *hybrid systems*, in our formal verification of ACAS X. This change is motivated by the goal to model scenarios in which the intruder is maneuvering, such as being able to change its horizontal direction or vertical velocity. Where hybrid systems only allow *one* actor in the system to resolve decisions, hybrid games give *multiple* actors *independent* decision-making ability.

More specifically, the models presented in this article are rephrased using Differential Game Logic dGL [24, 26, 27] from the Differential Dynamic Logic dL [21–23, 25, 27] of previous work. dGL is an extension of dL, so it also supports discrete assignments, control structures, and following of differential equations to represent pilot decisions, trajectory requirements, and aircraft dynamics, respectively. However, dGL can also represent adversarial dynamics, meaning dGL can express two different players in a game scenario making independent decisions that may interfere. We make crucial use of this multi-player dynamics in our ACAS X game models to enable both aircraft to maneuver independently. Contrast this flexibility with hybrid systems models of ACAS X [10, 11], which are necessarily limited to a single fixed policy for the intruder (the intruder cannot maneuver but is assumed to follow a straight line trajectory in prior ACAS X work [10, 11]).

In the context of collision avoidance, one can think of the ownship as being a good-faith actor attempting to avoid collision, while the intruder is able to act independently in ways that, perhaps out of confusion, may interfere with the safety of the system. Since these two players follow independent intent, dGL works perfectly in this scenario to express these adversarial dynamics.

Of course, in the real world, an intruder will not actively attempt to collide with the ownship, but if the ownship's goal is to avoid collision no matter the actions of the intruder, then it is important to consider even the worst-case maneuvers that the intruder may perform. Notably, our ACAS X game model considers the case where the intruder's actions *may* interfere with the safety of the system but does not assume they will. Indeed, the actions that the ownship pilot's winning strategies for the ACAS X game needs to take to avoid collision are less extreme when the intruder pilot reaches helpful decisions and more extreme otherwise (see Figure 1). dGL is implemented in the theorem prover KeYmaera X [4], with which we verify our safe regions with respect to our models.

As far as we know, this is the first work to apply hybrid games to the problem of aircraft collision avoidance. Hybrid games enrich the fidelity of the safety analysis for collision-avoidance algorithms, because they capture the important phenomenon that the respective pilots of intruder

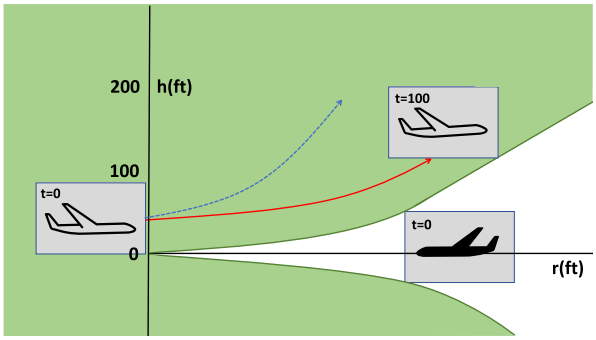


Fig. 2. Nominal trajectory (solid red) within the safe region (green) of an ownship accelerating towards an upsense advisory, with an example of a compliant trajectory (dashed blue).

and ownship aircraft reach their decisions independently, while, at the same time, being faithful to the advisories of ACAS X. Given the range of trajectories that either pilot could follow as they react to their mutual responses during an encounter, a game theory perspective on collision avoidance greatly expands the scenarios that can be modeled and proven.

The article is organized as follows: In Section 2, we give an overview of the structure that the models have in common. In Sections 3, 4, and 5, we introduce infinite-horizon safe models in which the intruder is given no maneuverability, vertical maneuverability, and horizontal maneuverability, respectively. In Sections 6 and 7, we introduce finite-horizon safe models to act as a stepping stone to the infinite-horizon models in Sections 8 and 9. Section 8 uses the concept of *safeability* from previous work [11], in which the ownship can follow an initial advisory for finite time and a subsequent advisory forever after. Section 9 adds intruder maneuverability to this scenario.

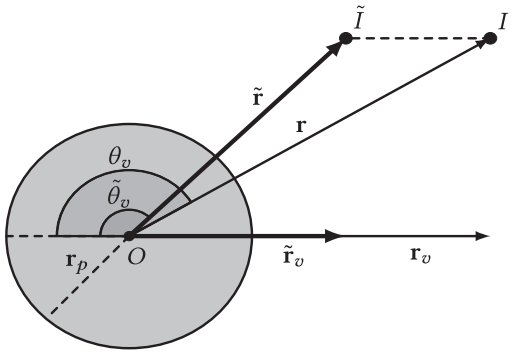
The models we consider come in three categories: infinite-time models, bounded-time models, and safeable models, which increase in complexity. Each category introduces a model that does not grant the intruder any maneuverability to establish intuition, before introducing the model(s) in which the intruder may maneuver. *The KeYmaera X models and proofs of all theorems are online.*[1]

2 OVERVIEW OF THE ACAS X MODELING APPROACH

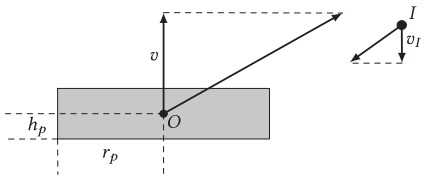
To establish intuition for our modeling approach of these flight scenarios, consider a scenario in which an ownship and an intruder are in the same flight space. The intruder at any point in time has the option to change its trajectory within a reasonable bound; the union of all of these possible trajectories at any future point in time describes the unsafe region for the ownship. If at a point in time, the ownship puck overlaps with a possible position that the intruder could be at that time, then we know that such an ownship trajectory is not provably safe, because there is a series of ownship and intruder actions that could lead to an NMAC. Therefore, if the ownship is outside of this region, then an NMAC cannot possibly occur, and the ownship is safe.

Figure 2 exemplifies a head-on encounter with the associated safe region for the intruder when the ownship follows a CL1500 advisory per Table 1. The coordinate system is fixed at the intruder and centered on the initial position of the ownship. The ownship starts at a relative vertical separation of 0, but a large horizontal separation from the intruder. Upon receiving the CL1500 advisory, it accelerates upwards with acceleration at least a_{lo}, but within the aircraft limits $a_{\max}$. Once it reaches a vertical velocity of at least 1,500 ft/min, it follows a linear path upwards until

[1] All KeYmaera X models and proofs are at https://github.com/LS-Lab/KeYmaeraX-projects/tree/master/acasx/acasx-games.



(a) Top view of the encounter: intruder adjusts $\mathbf{r}_v$ to $\tilde{\mathbf{r}}_v$, which corresponds to a change in angle θ_v between the ownship and the intruder



(b) Side view of the encounter

Fig. 3. An encounter scenario between ownship O and intruder I, with encasing puck shown in gray, adapted from Reference [11].

clearing the intruder aircraft. The green region is the region of safety that guarantees no NMAC (as long as the ownship follows the advisory), and the red line is the *nominal trajectory* representing minimal compliance with the advisory; the ownship can always choose to accelerate more than a_{lo} or reach a final upward velocity that is greater than the advisory, and this still qualifies as following the advisory giving uncountably many possible flight trajectories. An orthogonal question of equal impact on the safety of the outcome is the sequence of choices of the intruder aircraft.

2.1 Dynamics

Figure 3 shows one encounter between the ownship O and an intruder I. We follow conventions established in the ACAS X community [16], letting $r = \|\mathbf{r}\|$ be the horizontal distance and h be the vertical separation between the two aircraft. Both positions of the intruder are relative to the ownship.

In this article, we relax assumptions from previous work [10, 11] to give more maneuverability to the intruder. First, we do not necessarily assume that the horizontal rate of closure r_v between the two aircraft is constant. Specifically, in the model in Section 5, we grant the intruder limited control over this value. This corresponds to the intruder being able to change direction in the horizontal plane during the encounter, represented in Figure 3 by the θ_v angle between r_v and r.

Second, in the vertical direction, we not only allow the ownship's vertical velocity v to change at any moment, as in previous work, but we also grant the intruder limited control over its own vertical velocity v_I (Sections 4, 7, and 9). In all encounters, we assume that the vertical acceleration of the intruder cannot exceed constant c and that of the ownship cannot exceed $a_{\max}$. Any aircraft will have a rate of vertical acceleration that it cannot exceed due to the physical maneuverability limitations of the aircraft, and it is reasonable to assume for the ownship to have access to this value given the aircraft type of the intruder.

While these assumptions still limit the possible trajectories of each aircraft about which we will prove safety properties, they are necessary in the modeling and verification process. For instance, while it would be excellent to prove that the ownship can strategically wiggle out of a collision with any aircraft, this is just not possible if the intruder aircraft is strictly more maneuverable than the ownship. Thus, the c constant is necessary to prove meaningful safety properties, even if it limits the types of encounters to which these safety properties apply.

2.2 Advisories

ACAS X advisories (except for the Clear-of-Conflict and Multi-Threat Level-Off advisories) have two components: a target velocity v_{lo} and the direction of the target $w = \pm 1$. For example, the advisory CL1500 specifies that the pilot should achieve a climb rate of at least 1,500 ft/min, meaning the target velocity is 1,500 and the direction is upwards ($w = +1$) allowing larger climb rates. For the DNC2000 advisory, the pilot is advised not to climb more than 2,000 ft/min. This would make $v_{\text{lo}} = 2{,}000$ and $w = -1$. The w and v_{lo} values of the ACAS X advisories are in Table 1.

2.3 Model Overview

We present a high-level model whose basic structure other models in this article follow.

$$
\begin{array}{rll}
\text{P} & 1 & \text{init}(r_p, h_p, w, a_{\text{lo}}, a_{\max}, c) \land R(r, h, v, w, v_{\text{lo}}) \rightarrow \\
\text{advisory} & 2 & \Big[\Big(((w, v_{\text{lo}}) := *;\ ?R(r, h, v, w, v_{\text{lo}});\ \text{advisory} := (w, v_{\text{lo}}))\\
\text{ownship} & 3 & \quad \Big(a_o := \text{ownship}(\text{advisory}); ?(-a_{\max} \le a_o \le a_{\max})\Big)^d \\
\text{intruder} & 4 & \quad \boldsymbol{(a_i := *; ? -c < a_i < c;} \\
\text{motion} & 5 & \quad \{r' = -r_v, h' = -v, v' = a_o - a_i \,\&\, \text{EDC}(v, v_{\text{lo}}, a_o, a_i, a_{\text{lo}})\} \\
 & 6 & \quad \boldsymbol{)^*} \\
\neg\text{NMAC} & 7 & \Big)^*\Big]\Big(|r| > r_p \lor |h| > h_p\Big)
\end{array}
\tag{1}
$$

This dGL formula (1) of the shape $P \rightarrow [\alpha]\neg\text{NMAC}$ says that there is a winning strategy for the ownship in the hybrid game α starting in any state satisfying logical formula P to end up in a state satisfying $\neg$NMAC. The preconditions P ensure both nature-imposed and safety-imposed conditions about puck radius r_p and height h_p, upsense/downsense flag w, and we will develop relationships between minimum advisory compliant climb rate a_{lo}, ownship climb rate limits $a_{\max}$, and intruder climb rate limits c in each model. We also assume the ownship is initially in a safe region R for *some* initial advisory (w, v_{lo}), otherwise, we cannot conclude that it will be safe in the future. This is symbolically represented above by the formula $R(r, h, v, w, v_{\text{lo}})$, but this region is both specific to the model being studied and critical to proving safety, and will therefore be developed and explained in great detail in each section.

The game on lines 2–4 encodes the sequence of discrete choices made, followed by the evolution of the continuous dynamics on line 5. Specifically, an advisory is computed ($(w, v_{\text{lo}}) := *$) and issued on line 2 that must satisfy our safe region $?R(r, h, v, w, v_{\text{lo}})$, after which the ownship is allowed in line 3 to choose the particular acceleration a_o within the climb rate limits of the aircraft ($-a_{\max} \le a_o \le a_{\max}$) that it wants to follow during the encounter. This choice of a_o can access the advisory from line 2, but not the specific intruder choice a_i from line 4 after it. Note that an important switch in coordinates over Table 1 occurs with respect to v_{lo}. Table 1 uses v_{lo} to refer to the climb rate requested from the ownship pilot, while in all our models v_{lo} refers to an advisory in terms of *relative* climb rate; the coordinate transformation to the non-relative advisory is assumed to occur in advisory $:= (w, v_{\text{lo}})$ from relative v_{lo} and intruder velocity v_I at the time of issuing the advisory. Then on line 4, the intruder can change its own control decision a_i within climb rate limits c, which we assume in init to be strict enough so the ownship can overcome the worst-case intruder maneuvers given its own bounds $a_{\max}$. The differential equations of motion combine ownship and intruder acceleration to affect the relative climb rate v, and in turn the vertical separation h, while the horizontal separation r is affected by the relative horizontal speed r_v. The differential equations

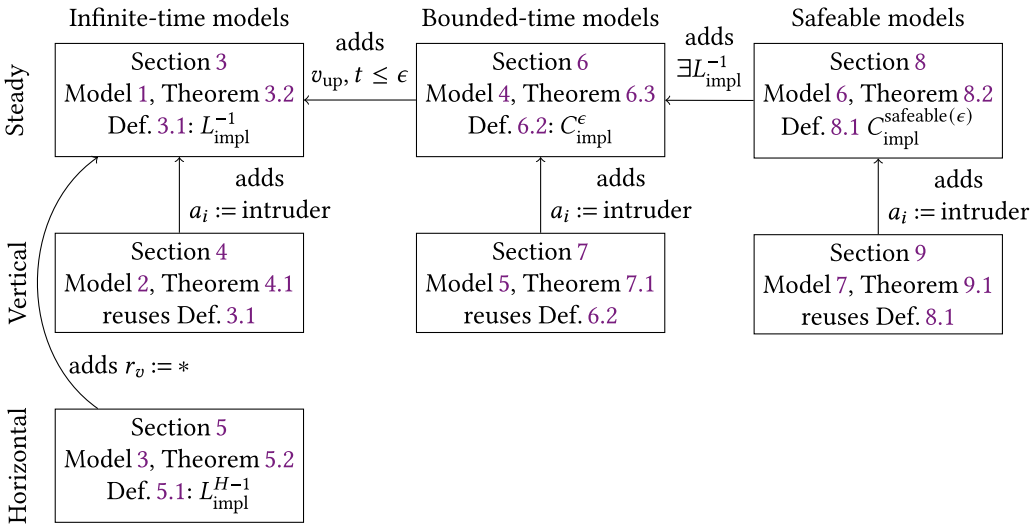


Fig. 4. Overview of the relationship between models, safe regions, and theorems.

are followed for any duration of time, as long as the evolution domain constraint $\text{EDC}(v, v_{\text{lo}}, a, a_{\text{lo}})$ is true. The evolution domain constraints vary, depending on the model, and will be discussed in later sections. The * operator on line 6 indicates that the inner loop from lines 4 to 6 can be repeated any number of times so the intruder can change decisions more often than the ownship. The boldface intruder choice on line 4 and the inner loop operator on line 6 are omitted from models that do not allow the intruder control over its trajectory.

Crucially, the ownship choice on line 3 is contained within the $(^d)$ operator, which represents the difference in choice between two players, so between the ownship and intruder. The dGL formula $P \rightarrow [\alpha]\neg\text{NMAC}$ states that given preconditions P, *there is a winning strategy for the ownship* that wins by successfully achieving $\neg$NMAC *for all intruder responses* when playing game α. All choices within the $(^d)$ game are resolved by our helpful player, so we need only show that there exists some run of this subgame such that for all runs of the game outside the operator, $\neg$NMAC is satisfied. Within the context of this model, this means that there need only be one choice of a_o that ultimately allows the ownship to avoid collision. Crucially, we will prove that for *any* advisory that satisfies our safe region and for *any* set of intruder actions, the pilot *can* strategically pick her acceleration a_o such that an NMAC does not occur. By contrast, if this choice of a_o were outside the $(^d)$ operator, then the model would be conjecturing that all of the infinitely many choices for ownship acceleration a_o would have to satisfy the postcondition, which it simply does not if the pilot does not pay attention.

The last * operator in line 7 is the outer loop around the entire program, which means that the aircraft encounter game can be repeated any number of times. More specifically, the pilot can be given any number of advisories by the aircraft, and our postcondition guarantees that *any sequence* of advisories that satisfy our safe region will guarantee collision freedom at all points in time.

2.4 Formalization and Verification Overview

The models and theorems in the following sections build upon the general shape (1) in an incremental fashion. An overview of the relationship between the models, definitions of safe regions, and safety theorems is given in Figure 4.

3 INFINITE-TIME SAFETY FOR A NON-MANEUVERING INTRUDER

In this section, we establish a baseline model in which the ownship and intruder are approaching each other at a constant horizontal velocity r_v. In future sections, the intruder will have control over its vertical acceleration or the horizontal rate of closure to present a greater challenge to collision avoidance, but for now, the model is kept simple to establish a baseline understanding.

In this section, we define an advisory to be safe only if it is safe indefinitely, meaning that no further advisory is needed to provide long-term safety of the ownship. This is too restrictive for reasons discussed in Section 6, but it allows for a simple model with which to start our investigation.

3.1 Model

The formula $L^{-1}_{\text{impl}}(r, h, v, w, v_{\text{lo}})$ on line 2 is the safe region of this model: An ownship originally separated from an intruder by r horizontally and h vertically will avoid collision given the advisory (w, v_{lo}). This crucial region will be developed and explained in Section 3.2. The postcondition on line 9 expresses the desire that there must always be a separation of the puck distance between the two aircraft in either the horizontal or vertical direction, so no NMAC ever occurs.

Lines 5–6 encode the advisory. The nondeterministic operator $\cup$ encodes that the pilot has two options: either to continue with her current advisory, in which case only the $(?true)$ condition must be satisfied, or follow a new advisory. The ? operator discards any runs of the system in which the condition after the ? is false, so $(?true)$ is always trivially satisfied. However, this $(?true)$ condition is important to ensure that the system always has a valid choice for an advisory (i.e., keep the previous advisory) and will not get stuck without any safe advisories.

The requirement on line 6 is that the choice of target velocity and up $(w = 1)$ or down $(w = -1)$ advisory will keep the ownship within the safe region indefinitely, as encoded by the condition $?(L^{-1}_{\text{impl}}(r, h, v, w, v_{\text{lo}}))$, given the minimum acceleration a_{lo} from Section 1.1. Without this assumption of a minimum acceleration, there is no guarantee in how quickly an aircraft would reach its target velocity, and therefore no safety guarantees.

Line 7 expresses the pilot's choice of her own vertical acceleration $(a_o := *)$. While this choice at first seems arbitrary in the context of the model, the proof will need to make strategic choices for a_o within the climb rate limits $a_{\max}$ of the aircraft to pass the subsequent test and avoid NMACs. Crucially, the choice of a_o and test in line 7 are within a $(^d)$ operator, since the choice of a_o and the responsibility of staying within the climb rate limits of the ownship are up to the ownship pilot.

The two aircraft then follow the differential equations on line 8. The differential equations express that r, v, and h of the ownship evolve according to r_v, the rate of horizontal closure, as well as the acceleration a_o chosen by the pilot based on the advisory issued by the system.

3.2 Implicit Formulation of the Safe Region

As previously stated, the safety of this model hinges on proving that an aircraft following an advisory from line 6 will stay within the safe region throughout the encounter with an intruder. This begs the question of what regions of flight guarantee safety for the ownship. Just like the previous work, we represent this region with the formula $L^{-1}_{\text{impl}}(r, h, v, w, v_{\text{lo}})$. This region is fixed at the intruder with its origin at the initial position of the ownship. We will explain one case in detail and provide a generalization of this region after.

First case: Consider the case of an upsense advisory $w = +1$, where the ownship has not yet reached the target velocity $(v \leq v_{\text{lo}})$. We use the concept of a nominal trajectory [11] (denoted by $\mathcal{N}(t)$), an example of which is shown in Figure 2 in red. In this figure, the ownship follows one possible trajectory adhering to all the requirements of ACAS X. This is just one of the uncountably

Model 1: Infinite-time safety for a non-maneuvering intruder

init	1	$r_p \geq 0 \wedge h_p > 0 \wedge r_v \geq 0 \wedge a_{\text{lo}} > 0 \wedge (w = -1 \vee w = 1) \wedge a_{\max} \geq a_{\text{lo}} \wedge$
R	2	$L^{-1}_{\text{impl}}(r, h, v, w, v_{\text{lo}})$
	3	$\rightarrow$
	4	$\big[\big($
advisory	5	$\big($ $?true$
	6	$\cup\ (w := 1 \cup w := -1); v_{\text{lo}} := *; ?L^{-1}_{\text{impl}}(r, h, v, w, v_{\text{lo}}); \text{advisory} := (w, v_{\text{lo}})\big)$;
ownship	7	$\big(a_o := *; ?(-a_{\max} \leq a_o \leq a_{\max})\big)^{\text{d}}$;
motion	8	$\{r' = -r_v, h' = -v, v' = a_o\}$
¬NMAC	9	$\big)^*\big]\big(|r| > r_p \vee |h| > h_p\big)$

infinitely many safe trajectories of the ownship. Upon receiving an advisory from ACAS X, the ownship begins climbing at an acceleration of a_{lo} and continues climbing along a parabola until it reaches the advised velocity v_{lo}. It then stops climbing and continues at the vertical velocity v_{lo} in a straight line. Through integration of the equations of motion from Model 1 line 8, we can get the coordinates (r_n, h_n) of the ownship along this nominal trajectory as a function of time:

$$\mathcal{N}(t) = (r_n, h_n) = \begin{cases} \left(r_v t, \frac{a_{\text{lo}}}{2} t^2 + vt\right) & \text{if } 0 \leq t < \frac{v_{\text{lo}} - v}{a_{\text{lo}}} \\ \left(r_v t, v_{\text{lo}} t - \frac{(v_{\text{lo}} - v)^2}{2a_{\text{lo}}}\right) & \text{if } \frac{v_{\text{lo}} - v}{a_{\text{lo}}} \leq t \end{cases}.$$

While these equations describe the minimally compliant nominal trajectory, the ownship pilot's choice of a_o in line 7 can be different than a_{lo} (and may exceed it while necessary). The actual coordinates of the ownship could be anywhere above this safe nominal trajectory, creating a region in which the ownship is guaranteed to be. Thus, the ownship is safe if it is separated horizontally from the nominal trajectory by at least the puck width ($|r - r_n| > r_p$) or it is above the nominal trajectory by at least puck height ($h_n - h > h_p$), that is:

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

This will be referred to as the implicit formulation of the safe region. It is an implicitly defined region because it uses quantifiers as opposed to explicit inequalities to define the nominal trajectory.

Generalization: The same reasoning applies to the $w = -1$ case where the pilot is told to descend to avoid collision as well as the $v > v_{\text{lo}}$ case, where the ownship has already achieved the target velocity and is now following the straight-line trajectory.

Going back to Model 1, the test $?L^{-1}_{\text{impl}}(r, h, v, w, v_{lo})$ on line 6 guarantees that following the nominal trajectory keeps the ownship safe, and thus, we can prove that if the pilot accelerates at least as fast as the minimum acceleration a_{lo} or has already reached the target velocity, then she is above the nominal trajectory and is therefore safe as well. The test also allows the ACAS X system the flexibility to give *any* advisory that results in a safe nominal trajectory, and the pilot the flexibility to choose arbitrary accelerations that keep the plane in the implicit region. This safe region is used to prove the safety postcondition of Model 1, making it sufficient to reason about this region to guarantee the safety of the ownship. The implicit formulation $L^{-1}_{\text{impl}}(r, h, v, w, v_{\text{lo}})$

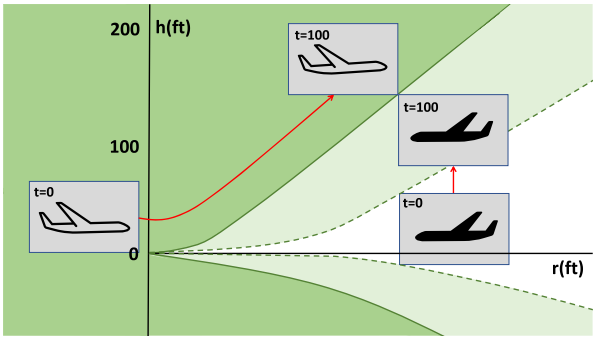


Fig. 5. Nominal trajectory (red) and safe region (dark green with a solid border) of an ownship accelerating towards an upsense advisory, where the intruder is accelerating in the same direction. The larger region that would be safe *without* intruder maneuverability is shown in light green with a dashed border.

follows Reference [11, Figure 3], is listed in Definition 3.1, and used in Theorem 3.2, which has been verified using KeYmaera X.

Definition 3.1 (Implicit Infinite-Time Safe Region).

$$T_{\text{lo}}(v, w, v_{\text{lo}}) \equiv \frac{\max(0, w(v_{\text{lo}} - v))}{a_{\text{lo}}}$$

$$A_{\text{lo}}(v, w, v_{\text{lo}}, h_n, t) \equiv \left(0 \le t < T_{\text{lo}}(v, w, v_{\text{lo}}) \wedge h_n = \frac{w a_{\text{lo}}}{2} t^2 + vt\right)$$
$$\vee \left(t \ge T_{\text{lo}}(v, w, v_{\text{lo}}) \wedge h_n = v_{\text{lo}} t - \frac{w \max(0, w(v_{\text{lo}} - v))^2}{2 a_{\text{lo}}}\right)$$

$$L^{-1}_{\text{impl}}(r, h, v, w, v_{\text{lo}}) \equiv \forall t \, \forall r_n \, \forall h_n \, (r_n = r_v t \wedge A_{lo}(v, w, v_{\text{lo}}, h_n, t)$$
$$\rightarrow (|r - r_n| > r_p \vee w(h_n - h) > h_p)).$$

Theorem 3.2 (Non-maneuvering Intruder: Correctness of Implicit Safe Regions). *The* dGL *formula given in Model 1 is valid. That is, as long as the advisories followed obey formula* $L^{-1}_{impl}(r, h, v, w, v_{lo})$ *from Definition 3.1, the winning strategy will avoid NMAC.*

Proof. The KeYmaera X proof develops a winning strategy for choosing ownship control a_o:

$$a_o = \begin{cases} w a_{\text{lo}} & \text{if } wv < w v_{\text{lo}} \\ 0 & \text{if } wv \ge w v_{\text{lo}} \end{cases}.$$

We use minimal vertical acceleration a_{lo} in direction w to adjust the climb rate towards the advisory. When the advisory is met (when $wv \ge w v_{\text{lo}}$), we keep the climb rate steady by picking $a_o = 0$. □

4 INFINITE-TIME SAFETY FOR A VERTICALLY MANEUVERING INTRUDER

Now that we have established a baseline model for this encounter that includes the important $(^d)$ duality operator for player selection in hybrid games, we can continue on to more expressive models. Our next model, Model 2, accounts for vertical intruder maneuvers, and the model expresses the notion that the ownship should always have a way to overcome any reasonable intruder action. In the example shown in Figure 5, even though the intruder begins accelerating in the same direction as the pilot, the pilot can still overcome the intruder action and navigate to safety.

4.1 Model

In this section, the setup is very similar to the last. The intruder and ownship still approach one another with a constant horizontal rate of closure r_v, and we have the same relative coordinate system with r for relative horizontal distance between the aircraft and h for relative vertical distance.

The difference in Model 2, highlighted in **bold**, is that we add the variable a_i representing the intruder's vertical acceleration, as well as a constant c, which represents the maximum magnitude of intruder acceleration, discussed previously in Section 2.1. On line 8, the intruder is now able to nondeterministically change its acceleration a_i within $-c$ and c. The differential equations on line 9 have also been modified to incorporate the new dynamics of the intruder. The rate of vertical closure between the two aircraft v now evolves at a rate of $(a_o - a_i)$ to reflect that *both* the intruder and ownship acceleration affect the rate of vertical closure. We choose to treat the rate of vertical closure v as a relative rate as opposed to creating two separate absolute vertical velocities to minimize the number of variables needed in this model even if a_o and a_i are independent.

Model 2: Infinite-time safety for a vertically maneuvering intruder

init	1	$r_p \geq 0 \land h_p > 0 \land r_v \geq 0 \land a_{\text{lo}} > 0 \land \boldsymbol{c > 0} \land (w = -1 \lor w = 1) \land \boldsymbol{a_{\max} \geq a_{\text{lo}} + c}$
R	2	$L^{-1}_{\text{impl}}(r, h, v, w, v_{\text{lo}})$
	3	$\rightarrow$
	4	$\big[\big($
advisory	5	$\quad\big($ $?true$
	6	$\quad\cup\ (w := 1 \cup w := -1); v_{\text{lo}} := *; ?L^{-1}_{\text{impl}}(r, h, v, w, v_{\text{lo}}); \text{advisory} := (w, v_{\text{lo}})\big)$;
ownship	7	$\big(a_o := *; ?(-a_{\max} \leq a_o \leq a_{\max})\big)^{\text{d}}$;
intruder	8	$\boldsymbol{\big(a_i := *;\ ?(-c < a_i < c)}$;
motion	9	$\quad\{r' = -r_v, h' = -v, \boldsymbol{v' = a_o - a_i}\}$
	10	$\quad\big)^*$
¬NMAC	11	$\big)^*\big]\big(|r| > r_p \lor |h| > h_p\big)$

We add a loop around the intruder choice of acceleration a_i and the dynamics on lines 8–10. The loop contains the intruder choice of acceleration, but not the ownship's choice, meaning the intruder could change its acceleration *any* number of times during an interaction without the ownship being able to react. Recall that ACAS X tracks the position and velocity of the intruder, but not its acceleration [15], so we cannot assume that the ownship has constant knowledge of the intruder's acceleration. Therefore, the ownship must first make a choice in acceleration and stick with it while the intruder is more powerful and allowed to change its acceleration arbitrarily often.

This model comes with new strategic insights that are developed in the proof. The strategy chooses a *relative* rate of vertical separation (not the absolute ownship velocity) as the target velocity, or that the ownship is accelerating at least at the minimum velocity a_{lo} plus the maximum intruder acceleration c as a strategy to overcome any possible intruder action. Since the ownship

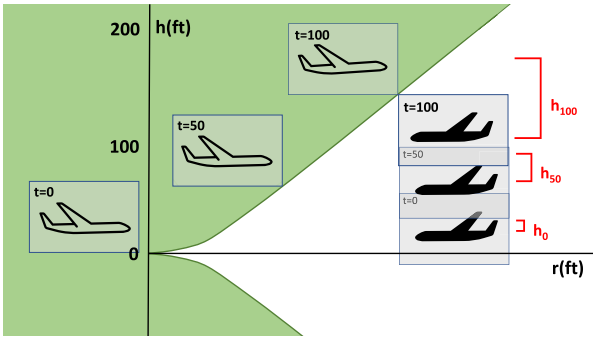


Fig. 6. An encounter between the ownship and a vertically maneuvering intruder showing absolute altitudes of each aircraft and relative separation h in red (h_0 initially, h_{50} at time $t = 50$, h_{100} at time $t = 100$).

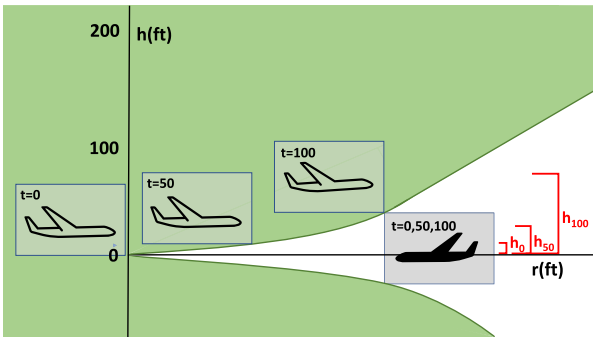


Fig. 7. An encounter between the ownship and a non-maneuvering intruder showing absolute altitudes of each aircraft and relative separation h in red (h_0 initially, h_{50} at time $t = 50$, h_{100} at time $t = 100$).

has access to the intruder's velocity and position, it can reasonably monitor the rate of vertical separation. However, since the ownship does not have continuous access to intruder acceleration, it must choose an acceleration that can withstand changes to the intruder's acceleration throughout the encounter. Therefore, if the ownship compensates by accelerating at least $a_{\text{lo}} + c$, then the intruder will not be able to catch the ownship during the encounter, even in the worst cases of the intruder accelerating at c or $-c$, because the relative acceleration difference will be at least a_{lo}.

Figure 6 shows an ownship following this new strategy from Theorem 4.1 with an intruder accelerating vertically at maximum climb rate c, and Figure 7 shows the ownship following the strategy from Theorem 3.2 with a non-maneuvering intruder. Note that the relative separation in the two figures evolves in the same way, since the ownship in Figure 6 compensates for the intruder maneuver by choosing a climb rate that accounts for the intruder maximum climb rate c.

4.2 Implicit Formulation of the Safe Region

Despite the new collision avoidance strategy, we do not need to make any changes to the implicit safe region formulation. This is because the intruder dynamics are hidden by the variables h and v, which represent *relative* vertical separation and rate of vertical closure. When the ownship strategically accelerates upwards with at least acceleration $a_{\text{lo}} + c$ until reaching a vertical rate of separation of at least v_{lo}, since the intruder cannot accelerate more than c, the relative rate of vertical acceleration is still at least a_{lo}. Therefore, our previous safe region L^{-1}_{impl} from Definition 3.1 still applies with the coordinate system still fixed at the intruder.

Theorem 4.1 (Vertically Maneuvering Intruder: Correctness of Implicit Safe Regions). *The* dGL *formula given in Model 2 is valid. That is, as long as the advisories followed obey formula* $L^{-1}_{impl}(r, h, v, w, v_{lo})$ *from Definition 3.1, the winning strategy will avoid NMAC.*

Proof. The KeYmaera X proof develops a winning strategy for choosing ownship control a_o:

$$a_o = \begin{cases} w(a_{\text{lo}} + c) & \text{if } wv < wv_{\text{lo}} \\ wc & \text{if } wv \geq wv_{\text{lo}} \end{cases}.$$

We compensate intruder maneuvers with increased minimal vertical acceleration $a_{\text{lo}}+c$ in direction w to adjust the climb rate towards the advisory. When the ownship follows the issued advisory (when $wv \geq wv_{\text{lo}}$), we pick $a_o = wc$ to compensate intruder maneuvers and maintain following the advisory. The safe region $L^{-1}_{\text{impl}}(r, h, v, w, v_{\text{lo}})$ serves as a loop invariant of the outer loop of Model 2. Since the ownship control choice $a_o = wc$ in the winning strategy is only safe when the ownship already follows the issued advisory, in this case, we additionally show that following the advisory ($wv \geq wv_{\text{lo}}$) is an invariant of the inner loop. □

5 INFINITE-TIME SAFETY FOR A HORIZONTALLY MANEUVERING INTRUDER

Where Model 2 expands on Model 1 by granting the intruder limited control over its *vertical* velocity, Model 3 in this section grants the intruder limited control over the *horizontal* rate of closure. The intruder can increase its ground velocity towards the ownship, as well as change the angle θ_v shown in Figure 3 to alter the rate of closure between the two aircraft.

5.1 Model

In Model 3, we give control of the horizontal rate of closure r_v to the intruder. We do not assume horizontal maneuverability for the ownship, but the intruder has sole control over the horizontal rate of closure, which determines angle θ_v, and can therefore nondeterministically choose r_v on line 8. This requires the new safe region L^{H-1}_{impl} on line 6. Just as in the previous section, we must assume some upper limit on the intruder maneuverability; therefore, we introduce the constant v_{max} on line 8 to represent the maximum possible horizontal rate of closure between the two aircraft. For simplicity, we also assume that the intruder's maneuvers cannot invert the rate of closure, meaning the intruder cannot fully turn around during the encounter ($\theta_v \in [90°, 270°]$), and r_v must be non-negative. This assumption ensures that the intruder is not fully adversarial and will not ascend or descend along a helix, or turn around to chase the ownship.

Besides these changes, the dynamics mirror that of Model 1. Again, the loop around the intruder choice and the dynamics allows the intruder to change the rate of closure as many times as it pleases during the encounter, but the ownship must stick with its initial choice of acceleration.

5.2 Implicit Formulation of the Safe Region

The safe region of this model must take into consideration the variable rate of closure r_v, which could vary anywhere from 0 to v_{max}. As such, we no longer have a single nominal trajectory, but infinitely many nominal trajectories $\mathcal{N}_{r_v}$ that are a function of the horizontal rate of closure r_v chosen by the intruder. Each nominal trajectory $\mathcal{N}_{r_v}$ with coordinates (r_n, h_n) at time t is the same as in Sections 3 and 4:

$$\mathcal{N}_{r_v}(t) = (r_n, h_n) = \begin{cases} \left(r_v t, \frac{a_{\text{lo}}}{2} t^2 + vt\right) & \text{if } 0 \leq t < \frac{v_{\text{lo}} - v}{a_{\text{lo}}} \\ \left(r_v t, v_{\text{lo}} t - \frac{(v_{\text{lo}} - v)^2}{2a_{\text{lo}}}\right) & \text{if } \frac{v_{\text{lo}} - v}{a_{\text{lo}}} \leq t \end{cases}.$$

Model 3: Infinite-time safety for a horizontally maneuvering intruder

init	1	$r_p \geq 0 \land h_p > 0 \land r_v \geq 0 \land a_{\text{lo}} > 0 \land \boldsymbol{v_{\max} > 0} \land (w = -1 \lor w = 1) \land$
R	2	$\boldsymbol{L_{\text{impl}}^{H-1}(r, h, v, w, v_{\text{lo}})}$
	3	$\rightarrow$
	4	$[($
advisory	5	$($ $?true$
	6	$\cup\ (w := 1 \cup w := -1); v_{\text{lo}} := *; \boldsymbol{?L_{\text{impl}}^{H-1}(r, h, v, w, v_{\text{lo}})}; \text{advisory} := (w, v_{\text{lo}}))$;
ownship	7	$(a_o := *; ?(-a_{\max} \leq a_o \leq a_{\max}))^{\text{d}}$;
intruder	8	$(\boldsymbol{r_v := *; ?(0 \leq r_v \leq v_{\max})}$;
motion	9	$\{r' = -r_v, h' = -v, v' = a_o\}$
	10	$)^*$
¬NMAC	11	$)^*](|r| > r_p \lor |h| > h_p)$

However, for our region to be safe, we must know that *each* nominal trajectory whose r_v is within $[0, v_{\max}]$ keeps a safe distance from the intruder at all future points in time. This motivates universally quantifying over all possible choices of r_v in our new safe region:

$$\forall t \forall r_v \forall r_n \forall h_n (r_v \in [0, v_{\max}] \land (r_n, h_n) \in \mathcal{N}_{r_v} \rightarrow |r - r_n| > r_p \lor h_n - h > h_p).$$

If this region holds for some advisory v_{lo}, then we can prove that for any possible horizontal rate of closure chosen by the intruder, if the ownship obeys the advisory, then it will avoid an NMAC. The implicit formulation $L_{\text{impl}}^{H-1}(r, h, v, w, v_{\text{lo}})$ is shown in Definition 5.1 and used in Theorem 5.2, which has been verified to be safe using KeYmaera X.

Definition 5.1 (Lower-bounded, Infinite Time Safe Region with Horizontally Maneuvering Intruder).

$$L_{\text{impl}}^{H-1}(r, h, v, w, v_{\text{lo}}) \equiv \forall t\, \forall r_v\, \forall r_n\, \forall h_n \Big(r_v \in [0, \max_v] \land r_n = r_v t \land A_{\text{lo}}(v, w, v_{\text{lo}}, h_n, t) \rightarrow (|r - r_n| > r_p \lor w(h_n - h) > h_p)\Big),$$

with $A_{\text{lo}}(v, w, v_{\text{lo}}, h_n, t)$ and $T_{\text{lo}}(v, w, v_{\text{lo}})$ as per Definition 3.1.

Theorem 5.2 (Horizontally Maneuvering Intruder: Correctness of Implicit Safe Regions). *The* dGL *formula given in Model 3 is valid. That is, as long as the advisories followed obey formula* $L_{\text{impl}}^{H-1}(r, h, v, w, v_{\text{lo}})$ *of Definition 5.1, the winning strategy will avoid NMAC.*

Proof. The KeYmaera X proof develops a winning strategy for choosing ownship control a_o:

$$a_o = \begin{cases} wa_{\text{lo}} & \text{if } wv < wv_{\text{lo}} \\ 0 & \text{if } wv_{\text{lo}} \leq wv \end{cases}.$$

We use minimal vertical acceleration a_{lo} in direction w to adjust the climb rate towards the advisory. When the advisory is met (when $wv \geq wv_{\text{lo}}$), we keep the climb rate steady by picking $a_o = 0$. □

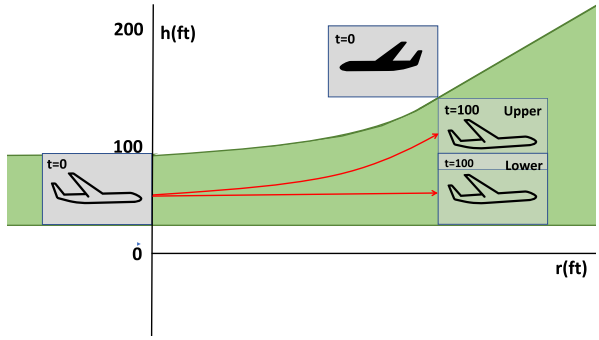


Fig. 8. An encounter between the ownship and a non-maneuvering intruder showing the two-sided region, with two compliant trajectories in red solid lines.

6 BOUNDED-TIME SAFETY FOR A NON-MANEUVERING INTRUDER

Up to this point, the models described in this article have all conveyed the notion that advisories issued by the ACAS X system must be safe *indefinitely*; that is, while the system *can* issue new advisories at any point, the aircraft must avoid collision *whether or not* the system issues a new advisory to pass our rigorous notion of safety. In the context of the above models, this need comes from the lack of a time bound on the differential equations, so *every* duration of the differential equation must guarantee safety for our model to be safe, even without another advisory change.

In a realistic scenario, this idea is limiting, because it typically forces the system to give fairly strong advisories too early on and too frequently. While this is acceptable from a safety perspective, it is unacceptable from a pilot's perspective. A system that gives overly frequent advisories can be distracting to the pilot and may lead the pilot to ignore the system warnings altogether. Thus, if an encounter is not immediately threatening, then the ACAS X implementation will actually issue COC or a preventative advisory such as DNC or DND and will later issue a more disruptive advisory as the threat level increases. In cases like this, the preliminary advisory may not be safe indefinitely, but safety can later be restored with a subsequent advisory (as will be the case in Figure 10). We represent this idea through the concept of safeability from previous ACAS X work [11, Definition 1].

Definition 6.1 (Safeable). 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.

In other words, we need only ensure the safety of an advisory for a few seconds as long as *some* followup advisory exists that will keep the ownship safe forever. Once again, the safeable region of a given advisory is always a superset of its safe region, so safeability is a more robust definition.

To develop the safeability models, we assume that the ownship and intruder are approaching at a constant horizontal rate of closure r_v, and r and h represent the relative horizontal and vertical separation, respectively. A subsequent advisory is called a *reversal* when the sign of w reverses from the initial advisory. If w does not change, then the subsequent advisory is a *strengthening* or *weakening*.

Given the complexity of proving safeability, we first prove bounded-time, upper and lower bounded safe regions as building blocks to proving safeability. All previous safe regions constructed in this article have only been lower-bounded for $w = 1$ (or only upper-bounded for $w = -1$), meaning the ownship must achieve some minimum acceleration or velocity to stay within the region. We now incorporate two-sided bounds on the ownship's acceleration and

vertical velocity to construct a region that is bounded from above *and* below (see Figure 8). This is important, because in a safeable scenario, the ownship could receive a reversal advisory, meaning it must reverse its direction of flight between the first and second advisories. In this case, it is imperative that we know both an upper and lower bound on the trajectory of the ownship in the first phase of flight to reason about its position and velocity in the second phase of flight.

The following models prove safety of the two-sided safe regions only up to some time ϵ. While it is *not sufficient* only to prove safety for some bounded ϵ amount of time, this step lays important groundwork for proving the fully safeable region in Section 8. The safeable proofs require that an advisory be provably safe only for ϵ time and that some followup advisory is safe indefinitely, so the bounded-time proof is a necessary stepping stone to ultimately proving the safeable region.

We, again, start with the model that allows no maneuverability to the intruder before moving on to the model that allows the intruder to accelerate vertically.

6.1 Model

Model 4 highlights differences to the infinite-time Model 1 in **bold**. First, we add symbolic upper bounds on the ownship acceleration a_{up} and vertical velocity v_{up} (usually $g/2$ and 10,000 ft/min, respectively). The upper bound v_{up} increases the complexity of the ownship strategy: The strategy in Theorem 3.2 picked acceleration $a_o \geq wa_{\text{lo}}$ towards satisfying $wv \geq wv_{\text{lo}}$, but now we must ultimately adjust this choice again before violating $wv \leq wv_{\text{up}}$. We address this with an *event-triggered* design in lines 9–10 with evolution domain constraints to monitor when wv crosses wv_{up}. To not discard behavior for the sake of detecting this event, we follow the standard pattern [29] to model event monitoring with overlapping evolution domain constraints.[2] We get notified that an event occurred exactly at the boundary where the two evolution domain constraints $wa_o \leq 0 \lor wv \leq wv_{\text{up}}$ and $wa_o \geq 0 \land wv \geq wv_{\text{up}}$ overlap. Through the loop on lines 8–11, the ownship can react to the event by selecting a new acceleration a_o in line 8.

To model the bounded-time safety, we add variable t, which acts as a timer for the duration of the dynamics. It is reset to 0 on line 7 at the beginning of each new advisory and evolves at a constant rate in the dynamics equation on line 9. Once this variable reaches ϵ, the differential equations are stopped from evolving further, and a new advisory must be issued. This time bound $t \leq \epsilon$ guarantees that the ownship will be given a new advisory and update its acceleration after at most ϵ-time has passed. The time-unbounded case is represented with the condition $\epsilon < 0$.

Another important difference introduced in this model is the removal of the test $?true$ from line 5. This test has been removed to disallow the ownship from blindly continuing with the old advisory after ϵ time. While the same advisory can be given on each iteration of the loop, the removal of the test $?true$ ensures that new advisories will definitely be considered after each ϵ time step.

Even though a new advisory is given after ϵ time elapses, this does not mean that a safe advisory *exists*. If no safe advisory exists, then the test on line 5 would fail, and the model would be vacuously true, since no possible runs of the system exist (and therefore all runs satisfy the postcondition). Thus, our model is only meaningful for ϵ time, after which we cannot draw any meaningful conclusions about safety. However, we address this issue of liveness in the models in Sections 8 and 9.

[2]The notation $\{x' = f(x)\ \&\ P(x) \land (Q_1(x) \cup Q_2(x) \cup \ldots \cup Q_n(x))\}$ is shorthand notation for the nondeterministic choice between ODEs that only differ in their evolution domain constraints: $\{x' = f(x)\ \&\ P(x) \land Q_1(x)\} \cup \{x' = f(x)\ \&\ P(x) \land Q_2(x)\} \cup \ldots \{x' = f(x)\ \&\ P(x) \land Q_n(x)\}$ and the evolution domain constraints Q_i are jointly exhaustive, i.e., $\bigvee_i Q_i(x)$ is valid. Crucially, $\{x' = f(x)\ \&\ P(x) \land (Q_1(x) \cup Q_2(x) \cup \ldots \cup Q_n(x))\}$ differs from $\{x' = f(x)\ \&\ P(x) \land (Q_1(x) \lor Q_2(x) \lor \ldots \lor Q_n(x))\}$ in its ability to detect the events of handover between neighboring regions Q_i and Q_{i+1} (e.g., from $Q_1(x)$ to $Q_2(x)$ or vice versa).

Model 4: Bounded-time safety for a non-maneuvering intruder

$$
\begin{array}{rll}
\text{init} & 1 & r_p \geq 0 \wedge h_p > 0 \wedge r_v \geq 0 \wedge a_{\text{lo}} > 0 \wedge (w = -1 \vee w = 1) \wedge \boldsymbol{a_{\text{up}} > a_{\text{lo}}} \wedge \\
\text{R} & 2 & \boldsymbol{C^{\epsilon}_{\text{impl}}(r, h, v, w, v_{\text{lo}}, v_{\text{up}})} \\
& 3 & \rightarrow \\
& 4 & \big[\big(\\
\text{advisory} & 5 & \quad \big((w := 1 \cup w := -1); v_{\text{lo}} := *; \boldsymbol{v_{\text{up}} := *}; ?\boldsymbol{C^{\epsilon}_{\text{impl}}(r, h, v, w, v_{\text{lo}}, v_{\text{up}})}; \\
& 6 & \qquad\qquad\qquad\qquad \text{advisory} := (w, v_{\text{lo}}, \boldsymbol{v_{\text{up}}})\big); \\
& 7 & \quad \boldsymbol{t := 0}; \\
\text{ownship} & 8 & \quad \big(\big(a_o := *; ?(-a_{\max} \leq a_o \leq a_{\max})\big)^{\text{d}}; \\
\text{motion} & 9 & \qquad \{r' = -v_v, h' = -v, v' = a_o, \boldsymbol{t' = 1} \,\&\, \boldsymbol{(t \leq \epsilon \vee \epsilon < 0)} \\
& 10 & \qquad\qquad \boldsymbol{\wedge ((w a_o \leq 0 \vee w v \leq w v_{\text{up}}) \cup (w a_o \geq 0 \wedge w v \geq w v_{\text{up}}))}\} \\
& 11 & \quad \big)^* \\
\neg\text{NMAC} & 12 & \big)^*\big]\big(|r| > r_p \vee |h| > h_p\big)
\end{array}
$$

6.2 Implicit Formulation of the Safe Region

The safe region of this model now consists of two separate safe regions: the lower bounded region $L^{\epsilon}_{\text{impl}}$ and the new upper region $U^{\epsilon}_{\text{impl}}$, listed in Definition 6.2.

The following observation is important for understanding the safe regions: In the event that the ownship is already exceeding v_{up} when receiving the v_{up} advisory, it is unrealistic for the pilot to accelerate downwards to reach the upper bound v_{up} from above. Instead, we assume the pilot will not accelerate further if she has already exceeded the target v_{up}. Therefore, on an initial overcompliance in vertical velocity ($wv \geq wv_{\text{up}}$), the target velocity becomes v as opposed to v_{up}.

First case: Consider $w = +1$ and $v_{\text{up}} \geq v$. Just like for the lower safe region, we consider a nominal trajectory $\mathcal{N}_{up}$ to characterize the upper safe region. In this region, we again accelerate upwards at the upper acceleration a_{up} until reaching v_{up} (or v in the case of initial overcompliance) and then continue linearly at velocity $\max(v_{\text{up}}, v)$. As before, we continue at a constant horizontal velocity r_v. We give the position along the nominal trajectory $\mathcal{N}_{up}$ as a function of time:

$$
\mathcal{N}_{up}(t) = (r_n, h_n) = \begin{cases} \left(r_v t, \frac{a_{\text{up}}}{2} t^2 + vt\right) & \text{if } 0 \leq t < \frac{v_{\text{up}} - v}{a_{\text{up}}} \\ \left(r_v t, v_{\text{up}} t - \frac{(v_{\text{up}} - v)^2}{2 a_{\text{up}}}\right) & \text{if } \frac{v_{\text{up}} - v}{a_{\text{up}}} \leq t \end{cases}.
$$

In the same way that we knew that the ownship is above the nominal trajectory in the lower-bounded region, we know that the ownship will be below this upper nominal trajectory because $a_o \leq a_{\text{up}}$ and $v \leq v_{\text{up}}$ (or $a_o \leq 0$). Therefore, to ensure safety of the ownship, we need:

$$
\forall t \, \forall r_n \, \forall h_n \left((t \leq \epsilon \vee \epsilon < 0) \wedge (r_n, h_n) \in \mathcal{N}_{up} \rightarrow |r - r_n| > r_p \vee h - h_n > h_p\right).
$$

Now that we have the regions $L^{\epsilon}_{\text{impl}}$ and $U^{\epsilon}_{\text{impl}}$ (Definition 6.2), we can characterize the two-sided safe region $C^{\epsilon}_{\text{impl}}$ as their disjunction. Given the fact that $a_{\text{lo}} \leq a_o \leq a_{\text{up}}$, we know that the ownship

stays between the two nominal trajectories, so its flight is properly upper and lower bounded; thus, as long as either the lower nominal trajectory or the upper nominal trajectory avoids collision, the ownship will be safe. This is the motivation for the disjunction of the two safe regions: Since the ownship is between the two trajectories, only one trajectory needs to be safe for the ownship to be safe as well. The implicit formulation of $C^{\epsilon}_{\text{impl}}$ is shown in Definition 6.2 and used in Theorem 6.3, which has been verified to be safe using KeYmaera X.

Definition 6.2 (Implicit Formulation of the Two-sided Safe Region).

$$T_{\text{up}}(v, w, v_{\text{up}}) \equiv \frac{\max(0, w(v_{\text{up}} - v))}{a_{\text{up}}}$$

$$A_{\text{up}}(v, w, v_{\text{up}}, h_n, t) \equiv \left(0 \leq t < T_{\text{up}} \land h_n = \frac{wa_{\text{up}}}{2}t^2 + vt\right) \lor \left(t \geq T_{\text{up}} \land h_n = w\max(wv_{\text{up}}, wv)t - \frac{w\max(0, w(v_{\text{up}} - v))^2}{2a_{\text{up}}}\right)$$

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

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

$$C^{\epsilon}_{\text{impl}}(r, h, v, w, v_{\text{lo}}, v_{\text{up}}) \equiv wv_{\text{lo}} \leq wv_{\text{up}} \land \left(L^{\epsilon}_{\text{impl}}(r, h, v, w, v_{\text{lo}}) \lor U^{\epsilon}_{\text{impl}}(r, h, v, w, v_{\text{up}})\right),$$

with $A_{\text{lo}}(v, w, v_{\text{lo}}, h_n, t)$ and $T_{\text{lo}}(v, w, v_{\text{lo}})$ per Definition 3.1.

Theorem 6.3 (Bounded-time Non-maneuvering Intruder: Correctness of Two-sided Bounded-time Safe Regions). *The* dGL *formula given in Model 4 is valid. That is, as long as the advisories obey formula C^{ϵ}_{impl} in Definition 6.2, the winning strategy will avoid NMAC.*

Proof. The KeYmaera X proof develops a winning strategy for choosing ownship control a_o:

$$a_o = \begin{cases} wa_{\text{lo}} & \text{if } wv < wv_{\text{lo}} \\ 0 & \text{if } wv_{\text{lo}} \leq wv \leq wv_{\text{up}} \\ 0 & \text{if } wv_{\text{up}} < wv \end{cases}.$$

We pick wa_{lo} to accelerate towards the advisory if the ownship is not yet in compliance $wv < wv_{\text{lo}}$. This case crucially relies on the event-triggered design, which notifies the ownship of a required change in strategy before violating $wv \leq wv_{\text{up}}$. When in compliance $wv_{\text{lo}} \leq wv \leq wv_{\text{up}}$ or in overcompliance $wv_{\text{up}} < wv$, we simply pick $a_{\text{lo}} = 0$ to maintain the current climb rate. □

7 BOUNDED-TIME SAFETY FOR A VERTICALLY MANEUVERING INTRUDER

With the groundwork laid in Section 6, we expand Model 4 for bounded-time safety to Model 5, where we allow the intruder to control its vertical velocity.

7.1 Model

Again, the variables t and v_{up} as well as the constants a_{up} and ϵ represent the upper safe region and our time bound, the variable a_i tracks the intruder's acceleration, and we allow it to affect the relative vertical velocity v. The main update in Model 5 is how the ownship reacts to intruder behavior. In line 7, the ownship estimates a bound c_o for the upcoming intruder acceleration (e.g., the worst-case bound c or a less permissive bound). The ownship strategy can take this estimate

Model 5: Bounded-time vertically maneuvering intruder

init	1	$r_p \geq 0 \wedge h_p > 0 \wedge r_v \geq 0 \wedge a_{\text{lo}} > 0 \wedge a_{\text{up}} > a_{\text{lo}} \wedge \boldsymbol{c > 0} \wedge (w = -1 \vee w = 1) \wedge$
R	2	$C^{\epsilon}_{\text{impl}}(r, h, v, w, v_{\text{lo}}, v_{\text{up}})$
	3	$\rightarrow$
	4	$[($
advisory	5	$\big((w := 1 \cup w := -1); v_{\text{lo}} := *; v_{\text{up}} := *; ?C^{\epsilon}_{\text{impl}}; \text{advisory} := (w, v_{\text{lo}}, v_{\text{up}})\big)\big);$
	6	$\boldsymbol{t := 0};$
estimate	7	$\big(\boldsymbol{(c_o := *; ?c_o \geq 0)^{\text{d}}};$
ownship	8	$\big((a_o := *; ?(-a_{\max} \leq a_o \leq a_{\max}))^{\text{d}};$
intruder	9	$\big(a_i := *; ?(\boldsymbol{-c_o} < a_i < \boldsymbol{c_o});$
motion	10	$\{r' = -r_v, h' = -v, \boldsymbol{v' = a_o - a_i}, t' = 1 \;\&\; (t \leq \epsilon \vee \epsilon < 0)$
	11	$\wedge (\boldsymbol{wv \leq wv_{\text{lo}} \cup wv_{\text{lo}} \leq wv \leq wv_{\text{up}} \cup wv_{\text{up}} \leq wv})\}$
	12	$\big)^*$
	13	$\big)^*$
	14	$\big)^*$
¬NMAC	15	$\big)^*]\big(\lvert r\rvert > r_p \vee \lvert h\rvert > h_p\big)$

into account when picking acceleration a_o. The intruder then, in line 9, gets to select intruder acceleration a_i: If that choice happens to fit to the ownship estimate, then the test $?(-c_o < a_i < c_o)$ passes and the model continues with motion on lines 10–11. Otherwise, the test fails and the only runs either have the intruder change its acceleration choice through the loop in lines 9–12 or have control return to the ownship via the loop on lines 7–14 to update the estimate c_o. This model allows for a variety of system implementations over a range of interaction requirements:

- no interaction between intruder and ownship is required when the ownship uses the worst-case acceleration c as its estimate c_o;
- the ownship may detect when the intruder acceleration exceeds the estimate c_o and change its strategy in return;
- the ownship and intruder may cooperate to pick the bound c_o;
- the ownship may announce the bound c_o as a requirement to the intruder.

The evolution domain constraint is again modified to detect when the relative climb rate falls below wv_{lo} or exceeds the target wv_{up}, so the ownship can change its strategy for overcoming the intruder motion. The event detection mechanism of Model 4 is extended in line 11 to detect when the ownship is about to no longer satisfy $wv_{\text{lo}} \leq wv \leq wv_{\text{up}}$, and so includes the choice between overlapping evolution domain constraints $wv \leq wv_{\text{lo}} \cup wv_{\text{lo}} \leq wv \leq wv_{\text{up}} \cup wv_{\text{up}} \leq wv$.

Figure 9 shows an example comparison of the current and former safe regions given the new intruder maneuverability. For simplicity, we explain only the strategy for the upsense case $w = +1$ and the worst-case intruder acceleration c. To continue evolving, just as in the infinite-time Model 2

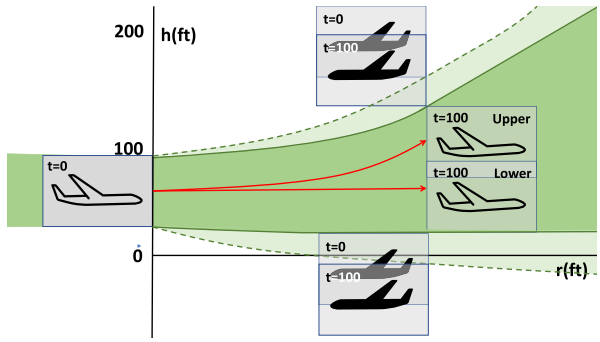


Fig. 9. An encounter between the ownship and a vertically maneuvering intruder showing the two-sided region in dark green with a solid border, with two compliant trajectories in red solid lines. The larger region that would be safe without intruder maneuverability is shown in light green with a dashed border.

from Section 4, the ownship acceleration must be at least $a_{\text{lo}} + c$ or the ownship velocity is at least v_{lo}. Simultaneously, the ownship either initially overcomplies in terms of velocity ($wv_{\text{up}} \leq wv$), or its velocity must not exceed v_{up} and its acceleration must not exceed $a_{\text{up}} - c$. In this first case of overcompliance, it is no longer enough to stop accelerating upwards. The ownship now needs to compensate for the fact that the intruder could be accelerating downwards with acceleration at most c by accelerating downwards with acceleration c as well.

7.2 Implicit Formulation of the Safe Region

The safe region of this model is exactly the same as that from Section 6. Due to the full relativization of both the rate of vertical closure and vertical separation, as well as the requirement that $a_{\text{lo}} + c \leq a_o \leq a_{\text{up}} - c$, the $C^{\epsilon}_{\text{impl}}$ with coordinate system fixed at the intruder still applies to this model.

Theorem 7.1 (Bounded-time Vertically Maneuvering Intruder: Correctness of Two-sided Bounded-time Safe Regions). *The* dGL *formula given in Model 5 is valid. That is, as long as the advisories obey formula C^{ϵ}_{impl} from Definition 6.2, the winning strategy will avoid NMAC.*

Proof. The KeYmaera X proof develops a winning strategy for choosing ownship control a_o in reaction to the worst-case intruder acceleration estimate $c_o = c$:

$$a_o = \begin{cases} w(a_{\text{lo}} + c) & \text{if } wv \leq wv_{\text{lo}} \\ 0 & \text{if } wv_{\text{lo}} \leq wv \leq wv_{\text{up}} \\ -wc & \text{if } wv_{\text{up}} \leq wv \end{cases}.$$

If the ownship is not yet in compliance $wv \leq wv_{\text{lo}}$, then we pick $a_o = w(a_{\text{lo}} + c)$, which is the minimum compliant acceleration a_{lo} compensated for worst-case intruder acceleration c. When in compliance $wv_{\text{lo}} \leq wv \leq wv_{\text{up}}$, we pick $a_o = 0$ for proof simplicity, but any acceleration $a_o \leq w(a_{\text{up}} - c)$ that does not exceed the upper acceleration a_{up} compensated for worst-case intruder acceleration c would work as well. Finally, in overcompliance $wv_{\text{up}} \leq wv$, we decelerate downwards with $a_o = -wc$ to compensate for worst-case intruder acceleration $-c$. □

Unlike in the non-maneuvering case Theorem 6.3, in the presence of a vertically maneuvering intruder there exists no choice of a_o that keeps the relative climb rate constant, since the intruder is allowed to change its acceleration a_i arbitrarily often. As a result, in Theorem 7.1 the choice

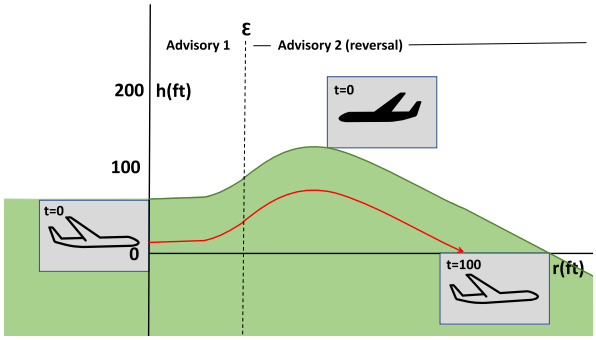


Fig. 10. An encounter between the ownship and an intruder showing a safeable region with reversal.

of $a_o = 0$, and any other choice in $a_o \leq w(a_{\text{up}} - c)$, crucially relies on the event detection that informs the ownship when $wv_{\text{lo}} \leq wv \leq wv_{\text{up}}$ is about to be violated and a change in strategy is required.

8 SAFEABILITY FOR A NON-MANEUVERING INTRUDER

In this section, we combine bounded-time safety with infinite-time safety to the notion of safeability: It is safe to follow a bounded-time region if there exists an infinite-time follow-up advisory. The intuition behind the safeable region is shown in Figure 10. We consider all the possible positions and speeds that the ownship could end up after ϵ time, in particular the lowest and highest such speeds and positions. At the lowest position, the most extreme strengthening is most critical, and at the highest position the most extreme reversal is most critical. The two safe regions achieved by these strengthenings and reversals give us the regions of ownship position that we can achieve by acting at time ϵ. This is precisely the safeable region: the safe region achieved by following a weaker advisory until time ϵ and then following a stronger advisory indefinitely.

8.1 Model

Model 6 extends the bounded-time Model 4 with changes highlighted in **bold**. We now require ϵ to be non-negative so the initial ϵ-time region is finitely bounded, and we use region $C_{\text{impl}}^{\text{safeable}(\epsilon)}$ on lines 2 and 5. The extensive changes to the safe region are discussed next.

8.2 Implicit Formulation of the Safe Region

The region $C_{\text{impl}}^{\text{safeable}(\epsilon)}$ again consists of a lower bound and an upper bound: regions $L_{\text{impl}}^{\text{safeable}(\epsilon)}$ and $U_{\text{impl}}^{\text{safeable}(\epsilon)}$ now combine two separate conditions: one region up to time ϵ and one region from time ϵ onward. Up to time ϵ, we follow bounded-time $L_{\text{impl}}^{\epsilon}$ and $U_{\text{impl}}^{\epsilon}$ from Definition 6.2. To be allowed to issue an advisory, the region now encodes that from time ϵ onward there must actually exist a new advisory given the potential ownship velocity and vertical separation at time ϵ under which the infinite region L_{impl}^{-1} is satisfied. This model is similar to the previous bounded-time Model 4 but also proves liveness: After the ϵ time bound, we guarantee the existence of another advisory (weakening or strengthening with w passed to L_{impl}^{-1}, reversal with $-w$) that keeps the ownship safe. The final region $C_{\text{impl}}^{\text{safeable}(\epsilon)}$ in Definition 8.1 is a disjunction of $L_{\text{impl}}^{\text{safeable}(\epsilon)}$ and $U_{\text{impl}}^{\text{safeable}(\epsilon)}$.

Model 6: Safeability for a non-maneuvering intruder

$$
\begin{array}{rll}
\text{init} & 1 & r_p \geq 0 \wedge h_p > 0 \wedge r_v \geq 0 \wedge a_{\text{lo}} > 0 \wedge (w = -1 \vee w = 1) \wedge a_{up} > a_{lo} + 2c \wedge \boldsymbol{\epsilon \geq 0} \wedge \\
\text{R} & 2 & \boldsymbol{C^{\text{safeable}(\epsilon)}_{\text{impl}}(r, h, v, w, v_{\text{lo}}, v_{\text{up}})} \\
& 3 & \rightarrow \\
& 4 & \big[\big(\\
\text{advisory} & 5 & \quad \big((w := 1 \cup w := -1); v_{\text{lo}} := *; v_{\text{up}} := *; ?\boldsymbol{C^{\text{safeable}(\epsilon)}_{\text{impl}}(r, h, v, w, v_{\text{lo}}, v_{\text{up}})}; \\
& 6 & \qquad\qquad\qquad\qquad \text{advisory} := (w, v_{\text{lo}}, v_{\text{up}})\big); \\
& 7 & \quad t := 0; \\
\text{ownship} & 8 & \quad \big(\big(a_o := *; ?(-a_{\max} \leq a_o \leq a_{\max})\big)^{\text{d}}; \\
\text{motion} & 9 & \qquad \{r' = -r_v, h' = -v, v' = a_o, t' = 1 \;\&\; \boldsymbol{t \leq \epsilon} \\
& 10 & \qquad\qquad \wedge ((w a_o \leq 0 \vee w v \leq w v_{\text{up}}) \cup (w a_o \geq 0 \wedge w v \geq w v_{\text{up}}))\} \\
& 11 & \quad \big)^* \\
\neg\text{NMAC} & 12 & \big)^*\big](|r| > r_p \vee |h| > h_p)
\end{array}
$$

Definition 8.1 (Implicit Two-sided Safeable Region).

$$
\begin{aligned}
L^{\text{safeable}(\epsilon)}_{\text{impl}}(r, h, v, w, v_{\text{lo}}) \equiv\; & L^{\epsilon}_{\text{impl}}(r, h, v, w, v_{\text{lo}}) \wedge \\
& \forall h^{\text{ex}}_L \forall v^{\text{ex}}_L \Big(\Big(0 \leq \epsilon < T_{\text{lo}}(v, w, v_{\text{lo}}) \wedge v^{\text{ex}}_L = w a_{\text{lo}} \epsilon + v \wedge h^{\text{ex}}_L = \frac{w a_{\text{lo}}}{2}\epsilon^2 + v\epsilon \\
& \qquad \vee \epsilon \geq T_{\text{lo}}(v, w, v_{\text{lo}}) \wedge v^{\text{ex}}_L = v_{\text{lo}} \wedge h^{\text{ex}}_L = v^{\text{ex}}_L \epsilon - \frac{w \max(0, w(v_{\text{lo}} - v))^2}{2a_{\text{lo}}}\Big) \\
& \qquad \rightarrow \exists v^{\text{ex}}_{\text{lo}} L^{-1}_{\text{impl}}(r - r_v \epsilon, h - h^{\text{ex}}_L, v^{\text{ex}}_L, w, v^{\text{ex}}_{\text{lo}})\Big)
\end{aligned}
$$

$$
\begin{aligned}
U^{\text{safeable}(\epsilon)}_{\text{impl}}(r, h, v, w, v_{\text{up}}) \equiv\; & U^{\epsilon}_{\text{impl}}(r, h, v, w, v_{\text{up}}) \wedge \\
& \forall h^{\text{ex}}_U \forall v^{\text{ex}}_U \Big(\Big(0 \leq \epsilon < T_{\text{up}}(v, w, v_{\text{up}}) \wedge v^{\text{ex}}_U = w a_{\text{up}} \epsilon + v \wedge h^{\text{ex}}_U = \frac{w a_{\text{up}}}{2}\epsilon^2 + v\epsilon \\
& \qquad \vee \epsilon \geq T_{\text{up}}(v, w, v_{\text{up}}) \wedge v^{\text{ex}}_U = w \max(w v_{\text{up}}, w v) \\
& \qquad\qquad \wedge h^{\text{ex}}_U = v^{\text{ex}}_U \epsilon - \frac{w \max(0, w(v_{\text{up}} - v))^2}{2a_{\text{up}}}\Big) \\
& \qquad \rightarrow \exists v^{\text{ex}}_{\text{up}} L^{-1}_{\text{impl}}(r - r_v \epsilon, h - h^{\text{ex}}_U, v^{\text{ex}}_U, -w, v^{\text{ex}}_{\text{up}})\Big)
\end{aligned}
$$

$$
C^{\text{safeable}(\epsilon)}_{\text{impl}}(r, h, v, w, v_{\text{lo}}, v_{\text{up}}) \equiv w v_{\text{lo}} \leq w v_{\text{up}} \wedge \Big(L^{\text{safeable}(\epsilon)}_{\text{impl}}(r, h, v, w, v_{\text{lo}}) \vee U^{\text{safeable}(\epsilon)}_{\text{impl}}(r, h, v, w, v_{\text{up}})\Big)
$$

with L^{-1}_{impl}, T_{lo} per Definition 3.1, and $L^{\epsilon}_{\text{impl}}$, $U^{\epsilon}_{\text{impl}}$, T_{up} per Definition 6.2.

Theorem 8.2. *The* dGL *formula given in Model 6 is valid. That is, as long as the advisories obey formula* $C^{safeable(\epsilon)}_{impl}(r, h, v, w, v_{lo}, v_{up})$ *in Definition 8.1, the winning strategy will avoid NMAC.*

Proof. The KeYmaera X proof develops a winning strategy for choosing ownship control a_o:

$$a_o = \begin{cases} wa_{\text{lo}} & \text{if } wv < wv_{\text{lo}} \\ 0 & \text{if } wv_{\text{lo}} \leq wv < wv_{\text{up}} \\ 0 & \text{if } wv_{\text{up}} \leq wv \end{cases}.$$

The proof is more involved, because the theorem is stronger, but the strategy is as in the bounded-time case. We pick wa_{lo} to accelerate towards the advisory if the ownship is not yet in compliance $wv < wv_{\text{lo}}$. We can keep this strategy until the event-trigger that the ownship has reached the advisory wv_{lo}. When $wv_{\text{lo}} \leq wv < wv_{\text{up}}$, we pick $a_o = 0$ so wv stays within wv_{lo} and wv_{up}. The ownship can choose an advisory that does not keep it within this range indefinitely, but it then relies on the event-trigger to notify when it is no longer in compliance, necessitating a new choice of acceleration. In the overcompliance case, we again choose 0 for wa_{lo}. The safe region $C^{\text{safeable}(\epsilon)}_{\text{impl}}$ is provably a loop invariant. From this invariant, it follows that a future advisory always exists such that the ownship will be safe indefinitely after using these strategies for the initial ϵ time. □

9 SAFEABILITY FOR A VERTICALLY MANEUVERING INTRUDER

Taking all of the groundwork laid by the previous sections, we finally present the model and safeable regions generalized for the case where the intruder can change its vertical acceleration.

9.1 Model

Model 7 extends bounded-time Model 5 with changes highlighted in **bold**. Again, we represent the intruder choice for a_i at line 11 and the bound c_o on this choice, as well as the loop around the intruder choice and dynamics to reflect that the intruder can change its acceleration throughout the encounter without the ownship being alerted to these changes.

The safe region for Model 7 follows Definition 8.1 due to the relativization of the vertical rate of closure and vertical separation.

Theorem 9.1. *The* dGL *formula given in Model 7 is valid. That is, as long as the advisories obey formula* $C^{safeable(\epsilon)}_{impl}(r, h, v, w, v_{lo}, v_{up})$ *from Definition 8.1, the winning strategy will avoid NMAC.*

Proof. The KeYmaera X proof develops a winning strategy for choosing ownship control a_o in reaction to the worst-case intruder acceleration estimate $c_o = c$:

$$a_o = \begin{cases} w(a_{\text{lo}} + c) & \text{if } wv \leq wv_{\text{lo}} \\ 0 & \text{if } wv_{\text{lo}} \leq wv \leq wv_{\text{up}} \\ -wc & \text{if } wv_{\text{up}} \leq wv \end{cases}.$$

This strategy matches that of the bounded time case. Again, if the ownship is not yet in compliance $wv \leq wv_{\text{lo}}$, then we pick $a_o = w(a_{\text{lo}} + c)$. When the ownship is compliant, with $wv_{\text{lo}} \leq wv \leq wv_{\text{up}}$, we pick $a_o = 0$, and in the overcompliance case, we accelerate downwards with $a_o = -wc$. With the safe region $C^{\text{safeable}(\epsilon)}_{\text{impl}}$ as the loop invariant, we know that a future advisory exists that will keep the ownship safe indefinitely, assuming the ownship follows these strategies up to ϵ time. □

10 DISCUSSION

Each of the proofs of the theorems presented in this article were completed in KeYmaera X [4] with support of Mathematica's implementation of real quantifier elimination [2]. Besides designing the hybrid game model and identifying the safe regions for its advisories, the main insights

Model 7: Safeability for a vertically maneuvering intruder

init
$r_p \geq 0 \land h_p > 0 \land r_v \geq 0 \land c > 0 \land a_{\text{lo}} > 0 \land (w = -1 \lor w = 1) \land$
$a_{\mathbf{up}} > a_{\mathbf{lo}} + 2c \land a_{\max} \geq a_{\text{lo}} + c \land \epsilon \geq 0 \land$
R $C^{\mathbf{safeable}(\epsilon)}_{\mathbf{impl}}(\boldsymbol{r, h, v, w, v_{\text{lo}}, v_{\text{up}}})$
$\rightarrow$
$[($
advisory $((w := 1 \cup w := -1); v_{\text{lo}} := *; v_{\text{up}} := *; ?C^{\mathbf{safeable}(\epsilon)}_{\mathbf{impl}}(\boldsymbol{r, h, v, w, v_{\text{lo}}, v_{\text{up}}});$
$\text{advisory} := (w, v_{\text{lo}}, v_{\text{up}}))^*;$
$t := 0;$
estimate $((c_o := *; ?c_o \geq 0)^{\text{d}};$
ownship $((a_o := *; ?(-a_{\max} \leq a_o \leq a_{\max}))^{\text{d}};$
intruder $(a_i := *; ?(-c_o < a_i < c_o);$
motion $\{r' = -r_v, h' = -v, v' = a_o - a_i, t' = 1 \;\&\; \boldsymbol{t \leq \epsilon}$
$\land (wv \leq wv_{\text{lo}} \cup wv_{\text{lo}} \leq wv \leq wv_{\text{up}} \cup wv_{\text{up}} \leq wv)\}$
$)^*$
$)^*$
$)^*$
¬NMAC $)^*](|r| > r_p \lor |h| > h_p)$

in the proofs are the invariants and winning strategies. The verified models crucially extend previous work with a rich representation of intruder behavior and its potentially adversarial nature. The benefit of attributing different actions to different players in the game is that we can now represent interactions between the ownship and the intruder, including worst-case non-cooperative interactions, as well as interactions in which the intruder and ownship react to one another.

Hybrid games verification, however, leads to even more complicated real arithmetic decision problems, which, even if decidable, may need time-consuming, handmade proof simplification for the proof to close. The arithmetic complexity of the safe regions required an intimate knowledge to simplify the proof into pieces digestible to Mathematica. This was particularly relevant compared to hybrid systems ACAS X verification [11] due to the addition of quantifier alternation stemming from games and new variables and constants to represent the intruder's maneuverability, as well as the introduction of more relationships between these values in the preconditions. The majority of manual arithmetic simplifications are case splitting (e.g., between $w = -1$ or $w = 1$), providing witnesses to quantifiers according to the winning strategy to help solvers, abbreviating terms to make transitivity of inequalities obvious, and selecting the relevant assumptions from the list of all assumptions. Table 2 compares our proofs in terms of tactic size as an indicator for relative manual proof effort, and proof checking duration as an indicator of arithmetic complexity. Its content has to be taken with a grain of salt, because there is considerable freedom in designing proofs and

Table 2. Proof Statistics: Tactic Size and Total Proof Checking Duration, Duration of All Proof Attempts at Real Arithmetic Proof Obligations (Column QE), and Duration of the Successful Attempts (Column RCF)

	Model	Intruder		Tactic steps	Proof checking [s]		
		Vert.	Horiz.		Total	QE	RCF
Infinite	Model 1, Theorem 3.2			163	18	15	12
	Model 2, Theorem 4.1	✓		196	35	30	26
	Model 3, Theorem 5.2		✓	242	35	29	23
Bounded	Model 4, Theorem 6.3			540	112	97	79
	Model 5, Theorem 7.1	✓		724	319	298	177
Safeable	Model 6, Theorem 8.2			3,402	1,786	886	279
	Model 7, Theorem 9.1	✓		2,782	1,404	1,163	247

in making the tradeoff between manual steps and duration of arithmetic proof obligations. We observe a few general trends across models:

- Even with manual simplification, the real arithmetic proof obligations are responsible for a considerable portion of the proof-checking duration;
- Infinite-time models are considerably easier (smaller tactic size, faster proof) than bounded-time models, which are in turn considerably easier than safeable models;
- Intruder maneuverability increases proof complexity, but experience from controlling the branching when proving the many ways of ownship and intruder interaction in the non-maneuvering safeable model helped find a smaller tactic design and faster proof in the vertical safeable model;
- The large number of combinations of ownship and intruder interactions in the safeable models makes it infeasible to manually simplify all the arithmetic proof obligations, which results in considerably longer proof checking duration; the large difference in the proofs of Models 6 and 7 between the duration of all proof attempts in the tactic QE and the duration of its successful attempt in column RCF indicates a potential for improving tactic heuristics or parallelization.

The increased complexity that each new variable introduces motivated our decision to relativize the rates of closure. This simplified the model and arithmetic and allowed for the same region from the corresponding non-maneuvering intruder to be used. We cannot further relativize the models beyond the rates of closure, however; it is important that each actor in the encounter has sole control over its own acceleration independently to properly model that both actors can affect the outcome of an encounter in this aircraft game scenario.

In terms of proof construction, the main challenge was in the development of the winning strategies for the ownship. This is a particular challenge in proving hybrid games; the ownship must pick a strategy without any knowledge of what the intruder may end up choosing, only knowledge of the broad limitations of the intruder's maneuvering capabilities. In the context of ACAS X, this is complicated by the fact that the ownship does not have access to the acceleration of the intruder, so it cannot react to the intruder's choice of acceleration to update its own choice. Therefore, it was imperative to have a proper strategy in choosing the pilot's acceleration to prove our intruder-maneuverability models.

A final challenge in hybrid game modeling is in the correct assignment of player responsibility for a given action. It is vital to the fidelity of the model that the choices in the model be resolved by the correct player to prevent one player having an unfair advantage. For instance, if the choice of advisory were resolved by the ownship, then the ownship could choose an optimal advisory to follow. However, the choice of advisory needs to be resolved by the intruder, because that proves that *all* of the advisories that satisfy the safe region can be chosen and are shown safe.

11 RELATED WORK

Kochenderfer and Chryssanthacopoulos [16] design the ACAS X lookup tables, the verification of which motivates this work (see Section 1.1). Von Essen and Giannakopoulou [32] use probabilistic model-checking in their analysis of a similar Markov Decision Process [16] to explore the probability of occurrence of various unfavorable events. The outcome is limited due to its discretization of the continuous dynamics in analyzing the system and the implausible assumption that the intruder follows a random walk in the decision space.

Holland et al. [8] and Chludzinski [1] simulate encounters, many of which come from recorded flight data. Lee et al. [18] develop a technique called differential adaptive stress testing to find scenarios in which TCAS does not result in an NMAC, but ACAS X does. These simulations provide interesting evaluations of the performance of ACAS X, but only explore a finite set of the state space and therefore cannot allow any conclusions about the infinitely many other possible behaviors.

Julian and Kochenderfer [13] train a deep neural network to approximate the ACAS X lookup table to reduce the storage needs and runtime of the system, and Irfan et al. [9] and Julian and Kochenderfer [12] explore applying formal methods to verify such neural networks. One drawback to this approach is that SMT does not support continuous dynamics, and all queries to the SMT solver must be in the form of discrete, linear regions. The inherent nonlinearity of the relevant regions when verifying ACAS X severely limits the verifying ability of this approach.

Kouskoulas et al. [17] develop a formally verified, quantifier-free predicate that, given a specific sequence of timed ownship and intruder maneuvers, checks whether or not an NMAC may occur. They do so by establishing envelopes around either aircraft that contain all altitudes reachable by each aircraft through each maneuver. While this work verifies the safety of pre-determined encounters between an ownship and intruder accelerating nondeterministically, it requires prior knowledge of the full sequence of maneuvers that the intruder will perform. Our work instead proves safety assuming no knowledge of the sequence of intruder maneuvers, better capturing the uncertain adversarial abilities of the intruder.

Lygeros and Lynch [20] explore verification of the conflict resolution algorithms used in TCAS, a predecessor of ACAS X, with hybrid techniques. This work is limited in its overzealous use of assumptions, for instance in assuming that two aircraft both using TCAS will ultimately be given opposite advisories in an encounter. Our work does not make assumptions about the actions of the intruder and ownship relative to one another; the decisions of one aircraft are independent from those of the other as are best expressed with game models.

Tomlin et al. [31] present a methodology with which to develop safe collision-avoidance maneuvers using hybrid systems. Platzer and Clarke [28], Loos et al. [19], and Ghorbal et al. [6] also use hybrid systems to design and verify their own horizontal collision avoidance maneuvers. Dowek et al. [3] and Galdino et al. [5] design their own algorithms for collision avoidance, known as KB3D and KB2D, respectively, and verify their geometry using the PVS theorem prover.

Other approaches for hybrid games have limited real-world applications due to the overly restrictive assumptions that they place on the systems that they represent. Henzinger et al. [7] work with rectangular hybrid games, which require strict numeric upper and lower bounds on the

continuous dynamics of the system and forgetful transitions. This work is one of the first case studies for hybrid games verification, the only example that we are aware of coming from Quesel and Platzer [30] in their feasibility study involving an abstract robot in a factory.

12 CONCLUSION AND FUTURE WORK

We applied hybrid games to the verification of ACAS X to prove that under limited horizontal and vertical maneuverability of an adversarial intruder, an ownship given a safe advisory from ACAS X always has a strategy to find a trajectory that avoids an NMAC despite subsequent intruder maneuvering. This work employed the principle of previous work [11] to identify regions of safety, whereby following any advisory in the safe region has an ownship strategy that will keep the aircraft clear of an NMAC. The safe regions and intruder capabilities are symbolic such that these models can be reused for future versions of ACAS X and apply to any intruder that is less maneuverable than the ownship. While increased arithmetic and proof complexity is the downside of working with hybrid games, the advantage is the significantly increased resulting predictive power, because collision freedom can be proved even if the intruder is maneuvering, which it will, in reality. In future work, we plan to explore verifying more complex ownship maneuvers in the horizontal direction and apply horizontal intruder maneuverability to the safeable model.

ACKNOWLEDGMENTS

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government, or any other entity.

REFERENCES

[1] Barbara J. Chludzinski. 2009. *Evaluation of TCAS II Version 7.1 Using the FAA Fast-time Encounter Generator Model.* Technical Report. MIT Lincoln Laboratory.

[2] George E. Collins. 1976. Quantifier elimination for real closed fields by cylindrical algebraic decomposition: A synopsis. *SIGSAM Bull.* 10, 1 (1976), 10–12. DOI: https://doi.org/10.1145/1093390.1093393

[3] Gilles Dowek and César Muñoz. 2005. Provably safe coordinated strategy for distributed conflict resolution. *AIAA Guidance, Navigation, and Control Conference* (08 2005). DOI: https://doi.org/10.2514/6.2005-6047

[4] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer. 2015. KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In *25th International Conference on Automated Deduction (LNCS)*, Amy P. Felty and Aart Middeldorp (Eds.), Vol. 9195. Springer, Cham, 527–538. DOI: https://doi.org/10.1007/978-3-319-21401-6_36

[5] André Luiz Galdino, César A. Muñoz, and Mauricio Ayala-Rincón. 2007. Formal verification of an optimal air traffic conflict resolution and recovery algorithm. In *14th International Workshop on Logic, Language, Information and Computation (LNCS)*, Daniel Leivant and Ruy J. G. B. de Queiroz (Eds.), Vol. 4576. Springer, Berlin, 177–188. DOI: https://doi.org/10.1007/978-3-540-73445-1_13

[6] Khalil Ghorbal, Jean-Baptiste Jeannin, Erik Zawadzki, André Platzer, Geoffrey J. Gordon, and Peter Capell. 2014. Hybrid theorem proving of aerospace systems: Applications and challenges. *J. Aerosp. Inf. Syst.* 11, 10 (2014), 702–713. DOI: https://doi.org/10.2514/1.I010178

[7] Thomas A. Henzinger, Benjamin Horowitz, and Rupak Majumdar. 1999. Rectangular hybrid games. In *10th International Conference on Concurrency Theory (LNCS)*, Jos C. M. Baeten and Sjouke Mauw (Eds.), Vol. 1664. Springer, 320–335. DOI: https://doi.org/10.1007/3-540-48320-9_23

[8] Jessica Holland, Mykel Kochenderfer, and Wesley Olson. 2013. Optimizing the next generation collision avoidance system for safe, suitable, and acceptable operational performance. *Proce. 10th USA/Eur. Air Traff Manag. Res. Devel. Semin.* 21 (07 2013). DOI: https://doi.org/10.2514/atcq.21.3.275

[9] Ahmed Irfan, Kyle D. Julian, Haoze Wu, Clark Barrett, Mykel J. Kochenderfer, Baoluo Meng, and James Lopez. 2020. Towards verification of neural networks for small unmanned aircraft collision avoidance. In *AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*. 1–10. DOI: https://doi.org/10.1109/DASC50938.2020.9256616

[10] Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Ryan W. Gardner, Aurora Schmidt, Erik Zawadzki, and André Platzer. 2015. A formally verified hybrid system for the next-generation airborne collision avoidance system.

In *21st International Conference on Tools and Algorithms for the Construction and Analysis of Systems, held as part of the European Joint Conferences on Theory and Practice of Software (LNCS)*, Christel Baier and Cesare Tinelli (Eds.), Vol. 9035. Springer, Berlin, 21–36. DOI: https://doi.org/10.1007/978-3-662-46681-0_2

[11] Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Aurora Schmidt, Ryan W. Gardner, Stefan Mitsch, and André Platzer. 2017. A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system. *Int. J. Softw. Tools Technol. Transf.* 19, 6 (2017), 717–741. DOI: https://doi.org/10.1007/s10009-016-0434-1

[12] Kyle D. Julian and Mykel J. Kochenderfer. 2021. Reachability analysis for neural network aircraft collision avoidance systems. *J. Guid. Contr. Dynam.* 44, 6 (2021), 1132–1142. DOI: https://doi.org/10.2514/1.G005233

[13] Kyle D. Julian, Mykel J. Kochenderfer, and Michael P. Owen. 2018. Deep neural network compression for aircraft collision avoidance systems. *CoRR* abs/1810.04240 (2018), 598–608.

[14] Mykel Kochenderfer and James Chryssanthacopoulos. 2011. *Robust Airborne Collision Avoidance through Dynamic Programming*. Technical Report ATC-371. MIT Lincoln Laboratories.

[15] Mykel Kochenderfer, Jessica Holland, and James Chryssanthacopoulos. 2012. Next generation airborne collision avoidance system. *Lincoln Lab. J.* 19 (01 2012), 17–33.

[16] Mykel J. Kochenderfer, L. P. Espindle, James K. Kuchar, and John Daniel Griffith. 2008. Correlated encounter model for cooperative aircraft in the national airspace system version 1.0. Technical Report ATC-344. MIT Lincoln Laboratory, Cambridge, Massachusetts.

[17] Yanni Kouskoulas, Daniel Genin, Aurora Schmidt, and Jean-Baptiste Jeannin. 2017. Formally verified safe vertical maneuvers for non-deterministic, accelerating aircraft dynamics. In *8th International Conference on Interactive Theorem Proving (LNCS)*, Mauricio Ayala-Rincón and César A. Muñoz (Eds.), Vol. 10499. Springer, 336–353. DOI: https://doi.org/10.1007/978-3-319-66107-0_22

[18] Ritchie Lee, Ole Mengshoel, Anshu Saksena, Ryan Gardner, Daniel Genin, Jeffrey Brush, and Mykel J. Kochenderfer. 2018. *Differential Adaptive Stress Testing of Airborne Collision Avoidance Systems*. DOI: https://doi.org/10.2514/6.2018-1923

[19] Sarah M. Loos, David W. Renshaw, and André Platzer. 2013. Formal verification of distributed aircraft controllers. In *16th International Conference on Hybrid Systems: Computation and Control*. ACM, 125–130. DOI: https://doi.org/10.1145/2461328.2461350

[20] J. Lygeros and N. Lynch. 1997. On the formal verification of the TCAS conflict resolution algorithms. In *36th IEEE Conference on Decision and Control*. 1829–1834. DOI: https://doi.org/10.1109/CDC.1997.657846

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

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

[23] André Platzer. 2012. Logics of dynamical systems. In *Logic in Computer Science Symposium*. IEEE, 13–24. DOI: https://doi.org/10.1109/LICS.2012.13

[24] André Platzer. 2015. Differential game logic. *ACM Trans. Comput. Log.* 17, 1 (2015), 1:1–1:51. DOI: https://doi.org/10.1145/2817824

[25] André Platzer. 2017. A complete uniform substitution calculus for differential dynamic logic. *J. Autom. Reas.* 59, 2 (2017), 219–265. DOI: https://doi.org/10.1007/s10817-016-9385-1

[26] André Platzer. 2017. Differential hybrid games. *ACM Trans. Comput. Log.* 18, 3 (2017), 19:1–19:44. DOI: https://doi.org/10.1145/3091123

[27] André Platzer. 2018. *Logical Foundations of Cyber-physical Systems*. Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-63588-0

[28] André Platzer and Edmund M. Clarke. 2009. Formal verification of curved flight collision avoidance maneuvers: A case study. In *International Symposium on Formal Methods (LNCS)*, Ana Cavalcanti and Dennis Dams (Eds.), Vol. 5850. Springer, 547–562. DOI: https://doi.org/10.1007/978-3-642-05089-3_35

[29] Jan-David Quesel, Stefan Mitsch, Sarah Loos, Nikos Aréchiga, and André Platzer. 2016. How to model and prove hybrid systems with KeYmaera: A tutorial on safety. *Int. J. Softw. Tools Technol. Transf.* 18, 1 (2016), 67–91. DOI: https://doi.org/10.1007/s10009-015-0367-0

[30] Jan-David Quesel and André Platzer. 2012. Playing hybrid games with KeYmaera. In *International Joint Conference on Automated Reasoning (LNCS)*, Bernhard Gramlich, Dale Miller, and Ulrike Sattler (Eds.), Vol. 7364. Springer, 439–453. DOI: https://doi.org/10.1007/978-3-642-31365-3_34

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

[32] Christian von Essen and Dimitra Giannakopoulou. 2014. Analyzing the next generation airborne collision avoidance system. In *20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, held as part of the European Joint Conferences on Theory and Practice of Software (LNCS)*, Erika Ábrahám and Klaus Havelund (Eds.), Vol. 8413. Springer, 620–635. DOI: https://doi.org/10.1007/978-3-642-54862-8_54

Received 3 June 2021; revised 17 January 2022; accepted 23 May 2022