Formal Verification of Collision Avoidance for Turning Maneuvers in UAVs

Eytan Adler* and Jean-Baptiste Jeannin†
University of Michigan, Ann Arbor, MI, 48109

With the predicted future proliferation of Unmanned Aerial Vehicles (UAVs) in the airspace, it is crucial to guarantee the absence of collision between aircraft. We explore the problem of formally verifying collision avoidance for UAVs performing horizontal avoidance maneuvers such as turning right or left, while staying in the same horizontal plane. Due to operational constraints and pilot preferences, manned aircraft equipped with collision avoidance such as TCAS and ACAS Xa typically only use vertical maneuvers (climbing and descending), but UAVs may also use horizontal maneuvers. A large part of the previous formal verification work has focused on manned aircraft and thus vertical maneuvers, with comparatively less work focusing on horizontal maneuvers. In this paper, we are particularly interested in fixed-wing Unmanned Aerial Vehicles (either fixed objects or stationary hovering intruders).

In this work, we consider horizontal corrective maneuvers for fixed-wing UAVs, consisting of 4 possibilities: turn right (at 1.5 degree per second), hard turn right (3 degrees per second), and symmetrically turn left and hard turn left [1, 2]. For a given maneuver and ownship position, we call the safe region the set of positions and velocities where an obstacle or intruder can safely be and never collide with the ownship. We geometrically derive safe regions for each maneuver, and we formally verify those safe regions with respect to the UAV dynamics in an interactive theorem prover. Those safe regions can then be compared to the decisions made by industrial collision avoidance systems, thereby formally verifying and validating UAV-mounted systems using horizontal maneuvers. Finally, formal verification of horizontal maneuvers provide the basis for the future verification of hybrid horizontal and vertical corrective maneuvers, such as "climb at 1500 feet per minute while turning right".

I. Introduction

As airspace becomes more crowded, especially with unmanned aerial vehicles (UAVs) and other autonomous aircraft, new collision avoidance systems will be needed to prevent near mid-air collisions (NMACs). Historically, advisories have taken the form of climb or descend instruction because pilots tend not to like changing heading, but with new unmanned systems, lateral turning advisories become feasible. These next-generation collision avoidance systems may give turning advisories to aircraft, which, if followed, should not result in a collision.

A typical situation to which this system would be applied involves two aircraft: the ownship where the collision avoidance system is installed, and another aircraft at risk of a NMAC with the ownship called the intruder. In this paper we assume a collision avoidance gives horizontal maneuver advisories to the ownship that direct it to change its heading by a specified amount to avoid the intruder. For example, an advisory could instruct the ownship to turn left 40 degrees.

A second motivation for this work is the convergence of two recent research directions: Deep Neural Networks have been proposed for Aircraft and UAV Collision Avoidance [1, 2], while tools such as Reluplex [3] for verifying properties of Deep Neural Networks have recently been developed and applied to UAV collision avoidance. However, the properties checked in [3] are fairly weak properties that do not imply absence of collision. Julian *et al.* [4] recently used Reluplex [3] to prove stronger properties of absence of collision, but only in the case of safe regions for vertical maneuvers available. This is because the vertical safe regions were available from Jeannin *et al.*, but the horizontal safe regions are not. In this paper we develop and prove correct those horizontal safe regions.

The long term goal is to be able to verify that if the advisories given by a collision avoidance system are obeyed, a NMAC will not occur. We define a NMAC to occur if the intruder is closer to the ownship than some arbitrary safety buffer radius, r_p. To determine if a NMAC occurs, we must be able easily check whether the position of the intruder ever intersects with the path of the ownship and its safety buffer during and after the ownship follows the advisory. We

*Undergraduate Aerospace Engineering, Computer Science Minor, Class of 2020, eytana@umich.edu.
†Assistant Professor, Department of Aerospace Engineering, jeannin@umich.edu

call the region that is at all times outside of the ownship's safety buffer the safe region. To formally verify the collision avoidance system, we compare the safe region for a given advisory to the region that the collision avoidance system returns the same advisory.

In this paper we derive the equations describing the trajectory and associated safe region in Sec. II, formally verify the correctness of these equations in Sec. III, and discuss the how this work applies to future work in Sec. IV.

II. Safe Region Derivation

To model the problem, we consider a fixed wing Unmanned Aerial Vehicle (UAV), referred to as the ownship, represented as a point with a circular safety buffer around it. The ownship's location is expressed in a Cartesian coordinate frame representing the horizontal plane. This coordinate space is broken up into two regions: the unsafe and safe regions. The unsafe region is the region over which the safety buffer will pass as the ownship follows its trajectory. The safe region is the region that never touches the safety buffer—the region that a stationary obstacle, called the intruder, can occupy. In this paper, we limited the safe region to a left turn, but adding right turn is a trivial symmetrical case. To develop the safe regions, the following simplifying assumptions are made:

1) The intruder is stationary.
2) The ownship remains at constant speed throughout all the turning maneuvers.
3) The ownship instantaneously transitions from one turn radius to another (infinite jerk).
4) The ownship begins the turn immediately after receiving the advisory.
5) The ownship follows a turn with a constant radius and an inner angle of less than 90 degrees.
6) After the ownship turns, it continues straight indefinitely.

A. Nomenclature

The origin of the coordinate frame is defined to be at the center of the circle about which the the UAV turns and the x axis points to the right—out the direction of the right wing at the ownship's initial position. The other variables used to represent the geometry of the basic safe region are as follows:

R = turn radius
r_p = buffer radius
θ = turn angle
A = initial position of ownship
B = position of ownship at end of turn
n = position of ownship at arbitrary point along trajectory

B. Derivation

Following the techniques developed by Jeannin *et al.* [5], we develop two different descriptions of the same safe region that make the formal verification step more straightforward. The first, the implicit derivation, describes the safe region by defining the trajectory of the ownship and saying the safe region is the area always outside of the safety buffer as it travels along the trajectory. The second, the explicit derivation, describes the safe region by explicitly defining the path of the outermost points on the safety buffer. The explicit derivation is the one that allows us to easily check whether the intruder is in a safe or unsafe region for a given advisory from the collision avoidance system.

1. Implicit Region

The safe region derivation begins with an analysis of the dynamics of the aircraft. We make the assumption that the radial acceleration, the acceleration pointing toward the center of the turn, is constant, resulting in a constant radius turn as stated above. This means the turn is represented by a circular arc.

By making the assumption that the turn begins immediately, the arc with a radius R begins on the x axis and continues until the ownship has turned an angle θ. After turning an angle θ, the ownship continues straight indefinitely. We define the equations for the trajectory of the ownship using the coordinates (x_n, y_n) since n defines an arbitrary point along the trajectory, written in Eq. 1.

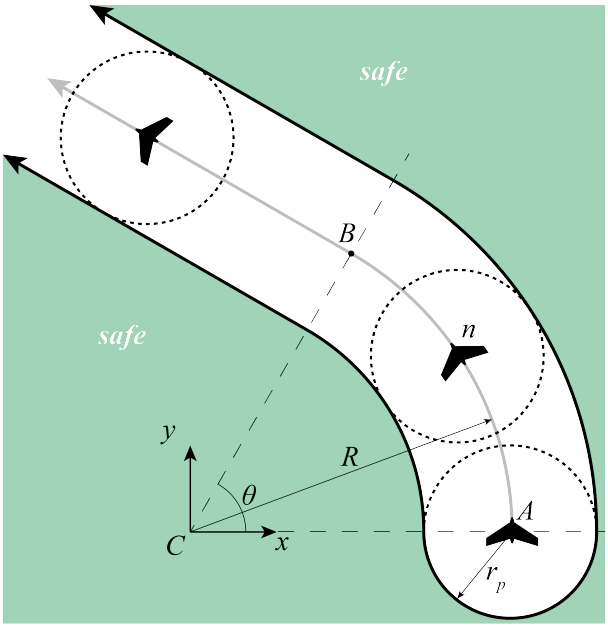

Fig. 1 The safe region represents the path of the edges of the safety buffer around the ownship.

$$trajectory(x_n, y_n) = \begin{cases} x_n^2 + y_n^2 = R^2, & y_n < x_n \tan\theta \\ y_n = -\frac{1}{\tan\theta}(x_n - R\cos\theta) + R\sin\theta, & y_n \geq x_n \tan\theta \end{cases} \tag{1}$$

To complete the derivation of the implicit region, we must enforce that the intruder is outside the safety buffer of the ownship. The safety buffer, written in Eq. 2, is a circular region in the coordinate space, (x, y), with a radius of r_p centered at an arbitrary point along the trajectory of the ownship, (x_n, y_n).

$$buffer(x, y) \equiv (x - x_n)^2 + (y - y_n)^2 \geq r_p^2 \tag{2}$$

In sum, the implicit region ensures that for every point along the trajectory, the position of the intruder (x_o, y_o) is outside the safety buffer. This is written as:

$$L_{impl} \equiv \forall x_n, \forall y_n \Big(trajectory(x_n, y_n) \;\rightarrow\; buffer(x_o, y_o) \Big)$$

2. Explicit Region

Based on the geometric idea from the implicit region, we can describe the explicit safe region, Eq. 3, by tracking the edges of the safety buffer around the circular then linear trajectory. Below the x-axis, there will be a semicircle from the buffer at the ownship's initial position. Around the arc, the buffer will sweep across a region that can be represented by an annular sector with an outer arc bound of radius $R + r_p$ and an inner arc bound of radius $R - r_p$. After the turn is completed, the buffer will travel linearly, so the safe region bounds are lines that are parallel to the trajectory offset by r_p in the direction perpendicular to the trajectory.

$$safe(x, y) = \begin{cases} (x - R)^2 + y^2 > r_p^2, & y \leq 0 \\ \left[x^2 + y^2 < (R - r_p)^2\right] \vee \left[x^2 + y^2 > (R + r_p)^2\right], & (y \leq x\tan\theta) \wedge (y > 0) \\ \left[y > \frac{-1}{\tan\theta}(x - (R + r_p)\cos\theta) + (R + r_p)\sin\theta\right] \vee & \\ \left[y < \frac{-1}{\tan\theta}(x - (R - r_p)\cos\theta) + (R - r_p)\sin\theta\right], & (y > x\tan\theta) \wedge (y > 0) \end{cases} \tag{3}$$

III. Formal Verification of Safe Region

We express the dynamics of our system in Differential Dynamic Logic [6], and formally prove the safe regions of Sec. II in its companion theorem prover KeYmaera X [7]. We follow much of the same ideas as Jeannin *et al.* [5], who formally proved safe regions for vertical collision avoidance maneuvers.

We formally verify the explicit safe region equations in two steps. Firstly, we prove that the implicit region matches the simple dynamics of the ownship with a safety buffer around the ownship's position. Secondly, we prove that the implicit and explicit equations describe the same region. The models and proofs for the following can be found at `http://jeannin.github.io/aviation19.zip`.

A. Formally Verifying the Implicit Region

The compare the implicit region to the ownship's dynamics, we must first define the dynamics of the ownship. Next, the formulation of the proof will be described.

Some additional nomenclature must be added to what was already defined the safe region derivation section to incorporate the dynamics and conform to the capabilities of the theorem prover. Some notable variables are restated from earlier for clarity.

v	=	constant velocity of the ownship
s_θ, c_θ	=	sine and cosine, respectively, of the final turn angle
(x_o, y_o)	=	position of the stationary obstacle or intruder
(x, y)	=	position of the ownship used in the dynamics
(x_n, y_n)	=	future positions of the ownship following a given advisory
s, c	=	sine and cosine, respectively, of the current angle around the turn used in the dynamics
s_n, c_n	=	sine and cosine, respectively, of the current angle around the turn used in the implicit region
t	=	current time used in the dynamics, held at zero during the turn and evolves normally after
t_n	=	time used in the implicit region, held at zero during the turn and evolves normally after

To begin the proof, we first describe the implicit region formally. This is done as follows:

$$
\begin{aligned}
L_{impl} \equiv\ & \forall t_n, \forall x_n, \forall y_n, \forall s_n, \forall c_n \Bigg(\\
& \Big(\big(t_n = 0 \ \wedge\ c_n \geq c_\theta \ \wedge\ s_n \geq 0 \ \wedge\ s_n^2 = 1 - c_n^2 \ \wedge\ x_n = R c_n \ \wedge\ y_n = R s_n \big) \\
& \vee\ \big(t_n \geq 0 \ \wedge\ c_n = c_\theta \wedge\ s_n \geq 0 \ \wedge\ s_n^2 = 1 - c_n^2 \ \wedge\ x_n = R c_n - v t_n s_n \ \wedge\ y_n = R s_n + v t_n c_n \big) \Big) \\
& \rightarrow\ \Big((x_n - x_o)^2 + (y_n - y_o)^2 \geq r_p^2 \Big) \Bigg)
\end{aligned}
$$

The strategy now becomes to derive the dynamics and prove that, given some bounds on the constants and initial conditions, $L_{impl} \rightarrow dynamics$. To derive the dynamics, two cases are considered: the circular turning case and the linear case.

For the turning case, we must update s and c since we avoid sines and cosines in KeYmaera X. Using chain rule, the derivative of $\sin\theta$ is $\dot{\theta}\cos\theta$ where $\dot{\theta} = \frac{v}{R}$ using the basic angular velocity equation. By the same idea, $\dot{\cos}\,\theta = -\sin\theta\frac{v}{R}$. With the calculated s and c, we can define $\dot{x} = -vs$ and $\dot{y} = vc$. The s and c are flipped from where they would intuitively go because the trajectory of the ownship is perpendicular to the vector with which θ is defined. Just like in the implicit region, the time is constrained to $t = 0$ and the condition $c \geq c_\theta$ is added to determine when to use the dynamics from the turning case versus the linear case's dynamics.

For the linear case, time now evolves as $\dot{t} = 1$. The sine and cosine parameters no longer evolve as the angle of the trajectory is held constant. The dynamics for $\dot{x}$ and $\dot{y}$ remain the same.

Finally, we must define the constraints on the constants and initial conditions at which to begin the evolution of the dynamics. The turn angle is confined to be less than or equal to 90 degrees via $c_\theta \geq 0 \ \wedge\ c_\theta < 1$. The initial condition on (x, y) is set to its initial position as shown in the figure at $(R, 0)$ and the associated sine and cosine of the position are set, $s = 0, c = 1$.

At this point, we can write the full model used to formally verify the implicit region as:

$$
\begin{aligned}
&\big(R > 0 \ \wedge\ r_p \geq 0 \ \wedge\ v > 0 \ \wedge\ c_\theta \geq 0 \ \wedge\ c_\theta < 1 \ \wedge\ x = R \ \wedge\ y = 0 \ \wedge\ s = 0 \ \wedge\ c = 1 \ \wedge\ t = 0\big) \wedge \\
&\quad L_{impl} \rightarrow \\
&\qquad \left[\left(\left(\dot{s} = c\frac{v}{R},\ \dot{c} = -s\frac{v}{R},\ \dot{x} = -vs,\ \dot{y} = vc \ \&\ t = 0 \ \wedge\ c \geq\ c_\theta\right)\right.\right. \\
&\qquad \left.\left. \cup\ \left(\dot{t} = 1,\ \dot{x} = -vs,\ \dot{y} = vc \ \&\ c = c_\theta\right)\right)^*\right] \left((x_n - x_o)^2 + (y_n - y_o)^2 \geq r_p^2\right)
\end{aligned}
$$

This model was successfully proven in KeYmaera X.

B. Formally Verifying the Explicit Region

The explicit formulation of the safe region is written as follows:

$$
\begin{aligned}
case_1 &\equiv y_o \leq 0 \\
bound_1 &\equiv (x_o - R)^2 + y_o^2 > r_p^2
\end{aligned}
$$

$$
\begin{aligned}
case_2 &\equiv y_o \leq x\frac{s_\theta}{c_\theta} \ \wedge\ y_o > 0 \\
bound_2 &\equiv x_o^2 + y_o^2 < (R - r_p)^2 \ \vee\ x_o^2 + y_o^2 > (R + r_p)^2
\end{aligned}
$$

$$
\begin{aligned}
case_3 &\equiv y_o > x_o\frac{s_\theta}{c_\theta} \ \wedge\ y_o > 0 \\
bound_3 &\equiv y_o > -\frac{c_\theta}{s_\theta}(x_o - (R + r_p)c_\theta) + (R + r_p)s_\theta \\
&\qquad \vee\ y_o < -\frac{c_\theta}{s_\theta}(x_o - (R - r_p)c_\theta) + (R - r_p)s_\theta
\end{aligned}
$$

$$
L_{expl} \equiv (\wedge_{i=1}^{3}(case_i \rightarrow bound_i))
$$

As before, we impose constraints on the constants. Now that s_θ and c_θ are added the relationship $s_\theta^2 + c_\theta^2 = 1$ must be introduced. Putting everything together, we must show that the constraints on the constants imply that the implicit formulation and explicit formulation are equivalent. This is written as follows:

$$
\begin{aligned}
&\big(R > 0 \ \wedge\ r_p \geq 0 \ \wedge\ r_p < R \ \wedge\ v > 0 \ \wedge\ c_\theta > 0 \ \wedge\ c_\theta < 1 \\
&\wedge\ s_\theta < 1 \ \wedge\ s_\theta > 0 \ \wedge\ s_\theta^2 + c_\theta^2 = 1) \ \rightarrow \\
&\qquad L_{impl} \longleftrightarrow L_{expl}
\end{aligned}
$$

The proof of this model is done geometrically, and a formal proof in KeYmaera X is in progress at the time of writing of this paper.

IV. Discussion

The safe region described in this paper provides an excellent place to begin to understand how to represent turning maneuvers in a horizontal plane. The equations also provide explicit, formally verified equations that describe the trajectory and surrounding path of the buffer.

Although the safe region offers an intuitive starting point, it lacks a few more advanced capabilities that should be addressed by a more nuanced model. The most notable limitation is that the collision avoidance systems of interest, specifically ACAS Xu, give turn rate advisories and no absolute angle that defines when the turn is complete. ACAS Xu

is a next-generation collision avoidance system in development that is designed for UAVs and gives horizontal maneuver advisories [1, 2]. ACAS Xu would give an advisory in the form "turn left at 1.5 degrees per second" as opposed to "turn left 30 degrees" or "turn left at 1.5 degrees per second until heading has changed by 30 degrees." The basic safe region derived in this section does not handle this well because an inner angle, θ, must be explicitly defined to construct the equations.

The second limitation is that the delay and imperfections of the pilot or autopilot are not taken into account. The basic safe region assumes the turn is begun immediately when the advisory is received. It also makes the assumption that the path of the circular arc and then linear trajectory are followed exactly.

Given these limitations, we intend to fully develop and formally verify a refined version of this safe region called the safeable region, based off of work done to formally verify ACAS X for vertical advisories [8]. The safeable region describes where, if a stationary intruder is located, a collision with the intruder is avoidable. The safeable region provides a solution to the basic safe region's major limitation: the difference between how the safe region is defined and the form of the collision avoidance system's advisories. By the way the safeable region is defined, it also solves the second limitation of the basic safe region that assumed the pilot/autopilot responded immediately and perfectly followed the trajectory.

The safeable region can be thought of as having two distinct regions: the initial delay period and the time after action is taken based on the advisory.

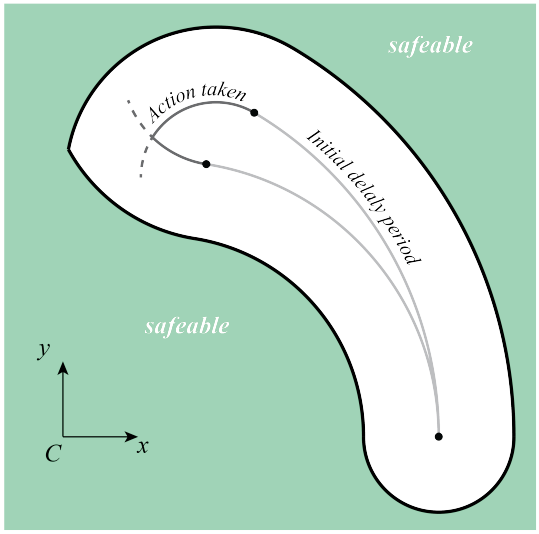

Fig. 2 The safeable region solves the big problems of the safe region.

The initial delay period is the region where the ownship is continuing on with its trajectory from before it received the advisory and continues until action regarding the advisory is taken. This region has an inner and outer radius to account for the ownship not following a perfectly circular trajectory, putting a bounded uncertainty on the ownship's location.

The time after action has been taken is represented by turning at the maximum possible rate in either direction. If the ownship was on the outside path, it would turn harder to a turn that is as sharp as possible. If the ownship was on the inside path, it would turn as sharp as it can in the opposite direction. This is meant to represent the limits of what can be avoided from the worst possible uncertainty case. This safeable region describes the possible locations of an intruder that avoid a NMAC if an imperfect pilot or autopilot makes a sharp turn after given time to respond to the advisory and would allow for a more realistic formal verification of ACAS Xu.

V. Related Work

Jeannin *et al.* develop similar safe regions designed for vertical ACAS X advisories [5]. Their paper models a moving intruder as opposed to the stationary intruder modeled in this paper.

Horizontal maneuvers for collision avoidance are also applied to hybrid systems in work by Tomlin *et al.* [9], Platzer and Clarke [10], Loos *et al.* [11] and Ghorbal *et al.* [12]. Dowek *et al.* [13] and Galdino *et al.* [14] developed a collision avoidance system called KB3D [15] and verified their work using PVS, another theorem prover.

Fulton and Platzer go into detail about the logic behind Differential Dynamic Logic and explicit representations of proofs in their paper [16].

Narkawicz *et al.* describe a different method using kinematic-based horizontal maneuvers to model close-proximity encounters to avoid NMACs where the instantaneous maneuver assumption cannot be made [17].

Muñoz *et al.* developed a criteria standard to resolve conflicts that differs from distributed systems in that the two aircraft involved do not necessarily need to execute the same conflict resolution algorithm [18]. Their work, guaranteed through formal verification in PVS, allows one or both aircraft to maneuver. Narkawicz *et al.* go into more detail to analytically define safety buffers in both lateral and temporal dimensions [19].

Lin and Saripalli propose a different method of avoiding UAV collisions by using a rapidly-exploring random tree to explore possible paths and suggest the shortest path with no collision [20]. Their method was not formally verified, only experimentally validated with simulated and actual flight testing.

More details about KeYmaera X, the theorem prover used for the work in this paper, can be found in its paper authored by Fulton *et al.* [7] and Platzer's book [6].

Acknowledgments

The safe region and its associated formal proof, done in KeYmaera X, are supported and based off work by Abhishek *et al.* [21]. The safe region and proof method is heavily influenced by Jeannin *et al.* [5, 8].

References

[1] Julian, K., Lopez, J., S. Brush, J., Owen, M., and Kochenderfer, M., "Policy compression for aircraft collision avoidance systems," 2016, pp. 1–10. doi:10.1109/DASC.2016.7778091.

[2] Julian, K. D., Kochenderfer, M. J., and Owen, M. P., "Deep neural network compression for aircraft collision avoidance systems," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 3, 2019, pp. 598–608. doi:10.2514/1.G003724, URL `https://arxiv.org/abs/1810.04240`.

[3] Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J., "Reluplex: An efficient SMT solver for verifying deep neural networks," *International Conference on Computer Aided Verification*, Springer, 2017, pp. 97–117.

[4] Julian, K. D., Sharma, S., Jeannin, J.-B., and Kochenderfer, M. J., "Verifying aircraft collision avoidance neural networks through linear approximations of safe regions," *AIAA Spring Symposium*, 2019. URL `https://arxiv.org/abs/1903.00762`.

[5] Jeannin, J.-B., Ghorbal, K., Kouskoulas, Y., Gardner, R., Schmidt, A., Zawadzki, E., and Platzer, A., "A Formally Verified Hybrid System for the Next-Generation Airborne Collision Avoidance System," *Tools and Algorithms for the Construction and Analysis of Systems*, edited by C. Baier and C. Tinelli, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 21–36.

[6] Platzer, A., *Logical Foundations of Cyber-Physical Systems*, Springer International Publishing, 2018. doi:10.1007/978-3-319-63588-0.

[7] Fulton, N., Mitsch, S., Quesel, J.-D., Völp, M., and Platzer, A., "KeYmaera X: An Axiomatic Tactical Theorem Prover for Hybrid Systems," *CADE*, LNCS, Vol. 9195, edited by A. P. Felty and A. Middeldorp, Springer, 2015, pp. 527–538. doi:10.1007/978-3-319-21401-6_36.

[8] Jeannin, J.-B., Ghorbal, K., Kouskoulas, Y., Schmidt, A., Gardner, R., Mitsch, S., and Platzer, A., "A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system," *International Journal on Software Tools for Technology Transfer*, Vol. 19, No. 6, 2017, pp. 717–741. doi:10.1007/s10009-016-0434-1, URL `https://doi.org/10.1007/s10009-016-0434-1`.

[9] Tomlin, C., Pappas, G. J., and Sastry, S., "Conflict resolution for air traffic management: a study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, 1998, pp. 509–521. doi:10.1109/9.664154.

[10] Platzer, A., and M. Clarke, E., "Formal Verification of Curved Flight Collision Avoidance Maneuvers: A Case Study," 2009, pp. 547–562. doi:10.1007/978-3-642-05089-3_35.

[11] M. Loos, S., Renshaw, D., and Platzer, A., "Formal verification of distributed aircraft controllers," 2013, pp. 125–130. doi:10.1145/2461328.2461350.

[12] Ghorbal, K., Jeannin, J.-B., Zawadzki, E., Platzer, A., J. Gordon, G., and Capell, P., "Hybrid Theorem Proving of Aerospace Systems: Applications and Challenges," *Journal of Aerospace Information Systems*, Vol. 11, 2014, pp. 702–713. doi: 10.2514/1.I010178.

[13] Dowek, G., and Muñoz, C., "Provably Safe Coordinated Strategy for Distributed Conflict Resolution," *AIAA Guidance, Navigation, and Control Conference*, 2005. doi:10.2514/6.2005-6047.

[14] Galdino, A., Muñoz, C., and Ayala-Rincon, M., "Formal Verification of an Optimal Air Traffic Conflict Resolution and Recovery Algorithm," 2007, pp. 177–188. doi:10.1007/978-3-540-73445-1_13.

[15] Muñoz, C., Siminiceanu, R., Carreño, V., and Dowek, G., "KB3D Reference Manual - Version 1.a," Technical Memorandum NASA/TM-2005-213679, NASA Langley, NASA LaRC,Hampton VA 23681-2199, USA, June 2005.

[16] Fulton, N., and Platzer, A., "A Logic of Proofs for Differential Dynamic Logic," 2016. doi:10.1145/2854065.2854078.

[17] Narkawicz, A., Muñoz, C., and Hagen, G., "An Independent and Coordinated Criterion for Kinematic Aircraft Maneuvers," *Proceedings of the 14th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Georgia, Atlanta, USA, 2014. doi:10.2514/6.2014-2859, URL `http://arc.aiaa.org/doi/abs/10.2514/6.2014-2859`.

[18] Muñoz, C., Butler, R., Narkawicz, A., Maddalon, J., and Hagen, G., "A Criteria Standard for Conflict Resolution: A Vision for Guaranteeing the Safety of Self-Separation in NextGen," Technical Memorandum NASA/TM-2010-216862, NASA, Langley Research Center, Hampton VA 23681-2199, USA, October 2010.

[19] Narkawicz, A., Muñoz, C., Herencia-Zapana, H., and Hagen, G., "Formal Verification of Lateral and Temporal Safety Buffers for State-Based Conflict Detection," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 227, No. 9, 2013, pp. 1412–1424. doi:10.1177/0954410012456495, URL `http://dx.doi.org/10.1177/0954410012456495`.

[20] Lin, Y., and Saripalli, S., "Sampling-based path planning for UAV collision avoidance," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, No. 11, 2017, pp. 3179–3192.

[21] Abhishek, A., Sood, H., and Jeannin, J.-B., "Formal Verification of Swerving Maneuvers for Car Collision Avoidance," 2019.