Formal Verification of Distributed Aircraft Controllers

Sarah M. Loos, David Renshaw, and André Platzer

Computer Science Department, Carnegie Mellon University

ABSTRACT

As airspace becomes ever more crowded, air traffic management must reduce both space and time between aircraft to increase throughput, making on-board collision avoidance systems ever more important. These safety-critical systems must be extremely reliable and work properly under every circumstance. In tough scenarios where a large number of aircraft must execute a collision avoidance maneuver, a human pilot under stress is not necessarily able to understand the complexity of the distributed system and may not take the right course, especially if actions must be taken quickly. We consider a class of distributed collision avoidance controllers designed to work even in environments with arbitrarily many aircraft. We prove the controllers never allow aircraft to get too close to one another, even when new planes approach an in-progress avoidance maneuver. Because these safety guarantees always hold, the aircraft are protected against unexpected emergent behavior which simulation and testing may miss.

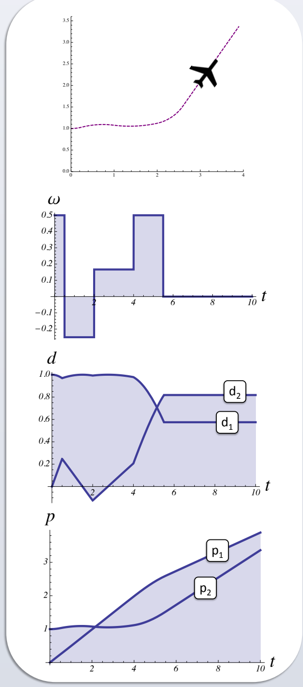


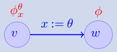
Figure 1. In a hybrid program, discrete changes in acceleration, for example, result in continuous changes in position and velocity.

METHODS

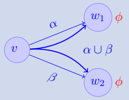
Distributed aircraft controllers are distributed hybrid systems, which we model and verify using quantified differential dynamic logic (QdL) [1]. Below are a few selected rules from the QdL proof calculus, categorized loosely as discrete or continuous transitions.

Discrete Transitions

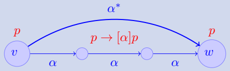
$$\frac{\phi_x^\theta}{[x:=\theta]\phi}$$



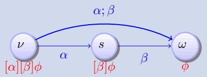
$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$



$$\frac{\vdash p \quad \vdash (p \to [\alpha]p)}{\vdash [\alpha^*]p}$$

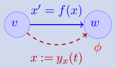


$$\frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi}$$



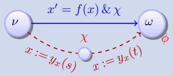
Continuous Transitions

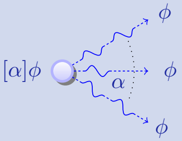
$$\frac{\exists t{\ge}0\, \langle x := y_x(t)\rangle\phi}{\langle x' = f(x)\rangle\phi}$$



$$\frac{\exists t{\ge}0\, (\bar\chi \land \langle x := y_x(t)\rangle\phi)}{\langle x' = f(x)\,\&\,\chi\rangle\phi}$$

$$\bar\chi \equiv \forall 0{\le}s{\le}t\, \langle x := y_x(s)\rangle\chi$$





CASE STUDY

Verified Big Disc Protocol

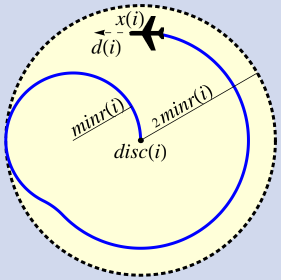


Figure 2. We illustrate one possible trajectory of a collision avoidance maneuver under the Big Disc protocol. The current direction of flight of aircraft *i* is given by the indexed variable *d(i)* as a 2D unit vector. The variable *disc(i)* is the position of the center of *i*'s buffer disc. The aircraft need not always turn at its maximum angular velocity; we require only that the aircraft remain within the disc by circling in its original direction.

Verified Small Discs Protocol

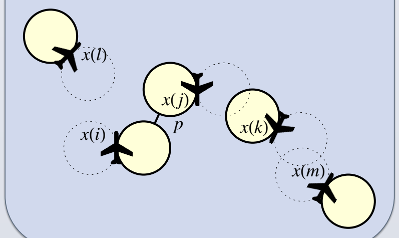


Figure 3. The buffer zone in this protocol is a disc of radius *minr(i)* centered at a point distance *minr(i)* away from *x(i)*, in a direction perpendicular to *i*'s motion either to the left or the right. Thus the aircraft is always on the edge of its disc, and during collision avoidance, the aircraft follows the circumference of its disc. An aircraft may flip its circling direction during free flight, causing the disc to jump to the other side of the aircraft; however, before an aircraft can flip its circling direction, it must check that it may do so safely.

TOOLS

KeYmaera is our automated theorem-prover for differential dynamic logic, [2]. KeYmaeraD is a distributed theorem-prover for quantified differential dynamic logic, which handles distributed hybrid systems, [3].

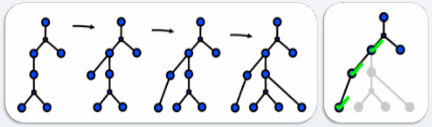


Figure 4a. Application of different rules at the same node allows parallelization of the proof search. These or-branches are shown here as circle nodes with two children. **Figure 4b**. Closing an or-branch.

CONTRIBUTIONS

- We provide *first formally* verified distributed system of aircraft with *curved flight* dynamics.
- Our controller requires only *flyable* aircraft trajectories with no corners or instantaneous changes of ground speed.
- We prove our controller is safe for an *arbitrarily large number of aircraft*. This guarantee is necessary for high-traffic applications such as crowded commercial airspace, UAV maneuvers, and robotic swarms.
- Other aircraft may enter an avoidance maneuver already *in progress* and safety for all aircraft is guaranteed still.
- We use *arithmetic coding* to reduce proof complexity and branching.
- We prove that even when the interactions of many aircraft cause unexpected *emergent behaviors*, all resulting control choices are still safe.
- We present *hierarchical and compositional* techniques to reduce a very complex system into smaller, provable pieces.

REFERENCES

This poster is based on research presented in full in:

Sarah M. Loos, David Renshaw, and André Platzer. Formal Verification of Distributed Aircraft Controllers. In Calin Belta and Franjo Ivancic, editors, Hybrid Systems: Computation and Control (HSCC), 2013.

[1] André Platzer. Quantified differential dynamic logic for distributed hybrid systems. In Computer Science Logic. Volume 6247 of LNCS. Springer, 2010.

[2] André Platzer and Jan-David Quesel. KeYmaera: A hybrid theorem prover for hybrid systems. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *IJCAR*, volume 5195 of *LNCS*, pages 171-178. Springer, 2008

[3] David Renshaw, Sarah M. Loos, and André Platzer. Distributed theorem proving for distributed hybrid systems. In the International Conference on Formal Engineering Methods, ICFEM'11, Durham, United Kingdom, Proceedings, LNCS. Springer, 2011.