Drone Ship Lander

15-624 Final Project Paper

David Kyle (dkyle@andrew.cmu.edu)

Space Explorations Technologies Inc. (SpaceX) is amidst tests to bring the first stage of their Falcon 9 rocket to a safe landing back on Earth. Their landing attempts involve sending a payload to orbit, then using a spare fuel margin to propulsively land either back at the launch site, or on an autonomous drone ship out to sea. To date, they have made 1 attempt to land at the launch site, which succeeded, and 5 attempts to land on the drone ship, the latest of which succeeded. The prior attempts all reached the ship, but failed to land successfully. In this project, I will attempt to model the latter, and prove correctness of a model controller.

Related Work

In *Returning Rocket, Friend or Foe?* [1], David Franklin and Phillip Massey analyze this same situation. They focus on treating the rocket as an inverted pendulum, which the controller must stabilize. I will consider this a solved problem, and assume the presence of this correct stabilization. Their descent control is relatively straightforward, as they ignore changing acceleration due to fuel mass loss. They justify this with a claim that this only produces "extra" deceleration, which is inherently safe. This is only true given that they also ignore the tight constraints on throttling the Falcon 9 booster faces. The engine cannot throttle arbitrarily low, and a premature cutoff may result in the booster crashing before the engine can start up again (if it could at all). Therefore, coming to a stop can be just as deadly as failing to stop in time. In this work, I intend to accurately model the acceleration the Falcon 9 boost experiences while trying to land.

In *Asteroid Approach* [2], Kerry Snyder analyzes a probe approaching and rendezvousing with an asteroid. This analysis uses an accurate model of mass changing over the course of the burn, however the controller assumes that using the "worst case" acceleration to compute actions is safe. In this case, the probe simple stops a little farther from the asteroid, but for the Falcon 9 booster, stopping may not be safe.

Goals

Even though the *Returning Rocket* paper is more thematically close to this paper's subject, the *Asteroid Approch* paper provides a model that is closer to what I wish to prove. It already attempts to account for changing acceleration as fuel burns away.

Thus, my first goal was to reimplement that model in a Keymaera X compatible form. Once this was completed I intended to accomplish the following incremental improvements:

Stopping at Surface Enforcement

The existing model merely attempts to stop before hitting the asteroid; it does not guarantee landing on it. For a landing booster, this model isn't sufficient. Stopping in mid-air would not be safe, as it would then be forced to cut engines (because it cannot throttle low enough to hover), and likely would not be able to restart them for the actual landing.

Fuel Budget Enforcement

The existing model assumes sufficient fuel. The ODE's domain constraint guarantees that fuel will not go below 0, and thus worlds in which this can happen effectively don't exist. The model could be improved by adding fuel-related safety properties, and adjusting the domain constraint.

Model Ocean Dynamics

The drone ship has stabilization engines, but they're not perfect. Some wave motion is inevitable. The model could account for this through sinusoidal changes to the distance to ground, and a "maximum safe velocity" for landing.

Initial Work

I started by implementing a model inspired by *Asteroid Approach*, intending that the same proof would work. Unfortunately, it did not. I attempted to investigate where mine differed significantly from the original, but I couldn't find a fix.

I then attempted a direct port of the original model, making only the minimal changes necessary for Keymeara X support. Mostly simple syntax changes were necessary, but an if/then/else construct needed to be replaced by a union operator; the conditional became a test for the first choice, and the negation the test for the second choice.

In spite of the straightforwardness of this conversion, it would still not prove in Keymaera X using the same approach used in the original paper. Upon looking closer at the proof technique used, I noticed that the differential invariant was surprisingly simple. Looking closer at the ODE, I noticed a major problem: the velocity assigned to the derivative of position had the wrong sign. The probe was being simulated as moving away from the asteroid, not towards it. Obviously, the safety property in that case is much easier than the correct model.

While I still don't know why the proof works in Keymaera and not Keymaera X, it was apparent that the original proof was not useful for the fixed model. Therefore, my objective shifted simply to implementing and proving a model of scope similar to the one provided in *Asteroid Approach*.

The goals listed above are kept as a record of my original intentions, and to serve as potential inspiration for followup work.

A New Model

I modified the model significantly. In addition to fixing the sign of the velocity, I switched it to event based, in hopes that it would be easier to prove. Were I to have time, I might have switched it back to time-based, but I did not. The event-based model should still be useful. I also refactored some of the computation out of the ODE itself to simplify attempts to solve it.

The revised model is available in the fuel.kyx file in the project files distribution.

A New Proof

The old proof approach was not capable of working. It attempted to cut in a differential invariant that was the same as the safety property itself. It likely worked in Keymaera purely due to the trivial nature of the erroneous model.

For this new proof, I employed differential ghosts to bound the real dynamics according to simplified dynamics which could be solved directly. I added p0, v0, and a0 ghost variables (to the model itself; I was unable to determine how to add them in the course of proving using Keymaera X), and gave them the dynamics where the change in acceleration due to change in fuel mass is ignored. Because the real dynamics are guaranteed to stop shorter than these simplified ones, they can be employed as differential invariant.

Ultimately, I arrive at an ODE where the “real” dynamics, and simplified dynamics coexist, and the real dynamics are bounded by the simplified dynamics, and with a safety property based solely on the simplified dynamics. Even though the real dynamics no longer have any impact on the hybrid program’s correctness, they still prevent Keymaera X from solving the ODE. To work around this, I produced a second model which is just this ODE, and it’s context, which can be loaded and proved after partially proving the main model.

This “closer” model is available in the fuel-closer.kyx file in the project files distribution.

Tactics are also provided (“fuel.kyt” and “fuel-closer.kyt”), however the former does not work if executed as a complete block, due to some issue with the partial keyword (according to the error log). I have tried putting it everywhere that makes any sense, but it doesn’t seem to work.

Conclusion

While this project did not accomplish the goals I had initially hoped to achieve, I hope it is still of value. Repetition of results in computer science tends to be rare, and in this case, I was able to find a subtle, but important issue, and fix it.

Works Cited

[1] D. F. a. P. Massey, "The Returning Rocket: Friend or Foe?," 10 December 2014. [Online]. Available:

http://symbolaris.com/course/fcps14/projects/dfranklipmassey.pdf.

[2] K. Snyder, "Asteroid Approach," 10 December 2014. [Online]. Available: http://symbolaris.com/course/fcps14/projects/kdsnyder.pdf.