AERE 407X Project Proposal

Meaghan McCleary

October 22, 2020

1. I will be doing this project individually. I will be model checking a piece of flight path planning software I wrote during a summer internship to evaluate the current feasibility guards and suggest others if they are necessary.

2. Group members: Just me

3. Group name: Flight Path Planner Verification (FPPV)

4. I will be using explicit model checking.

5. I am looking to analyze a specific part the flight path planning software to ensure there are guards in place to prevent a user from creating a flight path that is physically impossible to execute. I have kept the scope of the project purposely limited, as there are a great many areas of the tool that could be verified. The situations I will verify as impossible are as follows:

   (a) The aircraft cannot complete one linear segment and immediately transition into a segment at greater than a fifteen degree angle to the original heading. As an example, the LTL equivalent of this might appear as follows:

   $$(seglin \wedge segend) \rightarrow \neg(\mathcal{X}(deltaH > 15))$$

   where seglin is true if the segment is linear, segend is true if the time step represents the end of a segment, and deltaH is the change in heading from the previous time step.

   (b) The aircraft cannot complete one segment of linear or circular nature and immediately transition into a segment that begins at a point farther away than is feasible in one time step, i.e. teleportation.

   (c) There are other examples of these specification dealing with aspects such as bank angle and airspeed, and I will continue to develop this list as the project proceeds.

6. As mentioned, the system I will be model checking is a piece of flight path planning software. It is part of a much larger system of mission planning and simulation software used by a group at NASA Langley Research center that focuses on flight operations and test imagery services. I have been given permission to access this portion of the software suite. The flight path planning software allows the user to create the mission critical portion of the flight path of an aircraft carrying in-flight imagery assets. For example, the software could be used to plan the path of an aircraft such that is in the right flight pattern at the right time to capture imagery of the reentry and recovery of a spacecraft. The coordinate- and time-based output of this tool is then used to visually model the portion of the spacecraft's trajectory seen by the camera if the aircraft were to take that path.

The system functions by allowing the user control over the various linear and curved segments of the aircraft's flight path. By creating a series of such segments indexed to the master timeline of the mission (which includes other events related to the imagery target, such as a reentry interface or parachute deployment), the portion of flight path relevant to the imagery can be completed. This information then informs decisions such as aircraft take-off times. I have included a graphic of the user interface below as a visual aid.

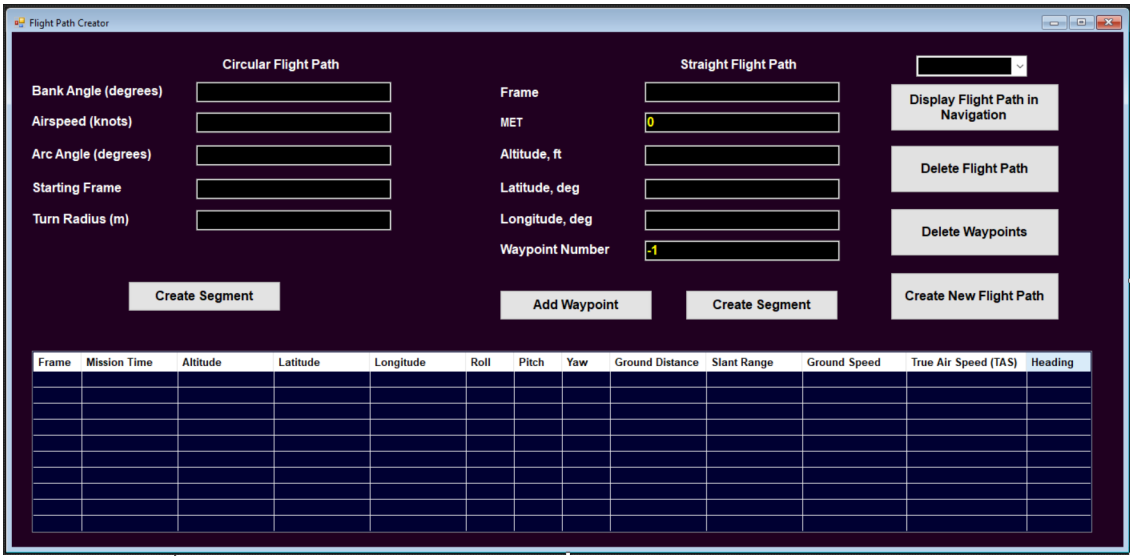

Figure 1. Flight Path Creator GUI

7. The following deliverables are required to be considered successful:
   (a) a model of the relevant part of the system, most likely written in Spin, but the choice of model checker may change as I explore the project further
   (b) validation of the model by state reachability analysis and code inspection
   (c) a set of LTL specifications corresponding to the list from question 5
   (d) model checking runs proving or disproving their accuracy
   (e) recommendations for where safe guards may need to be added or changed to improve the program
8. (a) I will use as benchmarks ranges of reasonable values for aircraft performance provided to me by a former coworker. The group uses a variety of aircraft, so these values must take into account various scenarios in which the tool may be used.

(b) I will demonstrate my analysis by the display of Spin's countertraces demonstrating the results of specification checking.

(c) My aim is to produce either proof of proper functionality or suggestions for improvement, so my results will be measured by these outcomes.

9. I would anticipate my repository structure appearing as follows:

   README.md - Description of the project and its goals
   Proposal.pdf - This document, with any edits made following professor feedback
   Report/ - PDF and .tex file for final report
   Model/ - Promela files involved in the system model
   Spec/ - LTL specification files used for model checking and a master list of specifications

   I have set aside approximately sevn hours per week until the completion of this project for related tasks, and I will allocate more time if needed.

10. Schedule:

    - Week 1:
      - Determine whether Spin the most appropriate tool
      - Develop full list of specification (English and LTL)
      - Begin creation of model
    - Week 2:
      - Finish creation of model
      - Validate model with the checks previously mentioned and debug as needed
      - Create LTL files for specifications
    - Week 3:
      - Run checks of specifications
      - Use results to develop recommendations for tool improvements
    - Week 4:
      - Finish last model checking tasks if necessary
      - Final report
      - Final presentation preparation