eVTOL Simulation

By Devin Connell

# Introduction

The eVTOL simulation was implemented according to the requirements and design document associated with this project. The simulation provides a comparison of different types of eVTOL aircraft and reports on flight and charging statistics of those aircraft. The simulation is written in C++ using the Microsoft Visual Studio IDE. Implementing in Linux was a non-starter since a Linux environment was not readily available. It would have taken extra time to set up the environment.

# Simulation Description

The simulation is an object oriented implementation that follows the class diagram in the Simulation Design Document. The simulation is initialized at the main entry point of the program, and a set of parameters are read from a configuration file. When the simulation is ready a main loop is executed that runs until the duration of the sim expires. At this point a statistics report is generated and output to a file.

## Configuration Files

The configuration file contains key items that determine the behavior of the simulation. These items include simulation duration, simulation time step, number of vehicles, and number of chargers. The intention of the configuration file is to provide a quick method for changing parameters of the simulation to explore different vehicle behaviors and performance in a variety of conditions. The configuration file is an item where additional work could be done to provide and even wider variety of simulation options. There is more about improving the configuration file in the Future Work section.

## Execution

The simulation executes incrementally over a time step. Since the simulation duration is measured in hours the time step is also measured in those units, although broken down into fractions of an hour, i.e., if the configured time step is 1 second the value used is 1/3600th of an hour. At each time step the state of each vehicle in the simulation is updated. The vehicle state data are parameters such as the current battery level, whether the vehicle is charging, waiting for a charger, or done charging. Statistical data such as Flight Distance, Passenger Miles, Flight Time, and Charge Time are also updated at this time step. Once the time increment has reached the simulation duration the simulation ends.

## Statistic Data Computation and Reporting

At the conclusion of the simulation the statistical data of all the vehicles is collected. The data for each type of vehicle is computed in one place and then a report is output to a file to show how each type of vehicle performed. The report provides the number of each type of vehicle and the total and averages of the following data for each vehicle type: Flight Distance, Passenger Miles, Flight Time, Charge Time, Charger Waiting Time, and Number of Faults. A sample report is provided in the Appendix.

# Known Issues

In addition to the known design constraints detailed in the design document, the primary issue in the implementation is in the configurable time step. With a configurable time step it is possible to select an increment of time that will skip over the true moment when a vehicle battery reaches zero, or full capacity. This will cause additional time to be attributed to flight time and/or charging time. Though the amount of time would be small, a sufficiently long simulation or a simulation with many more vehicles would skew all results relying on the time a vehicle transitions from flight to charging and vice versa.

The other known issue are the arbitrary maximums selected for the configuration items. There is no requirement to allow these parameters, such as the simulation duration, to vary. The maximum values are arbitrary and provide no additional benefit except to prevent a user from running a simulation lasting 232 hours with 232 vehicles.

When using the “Double Chargers” preset value, the value contained in the “Chargers” field is doubled. Since this value can vary, double the number of chargers means double the number specified. It does not mean double the default number. A simple change to use a hard coded value for the default number of chargers will fix this issue.

# Future Work

## Simulation Presets

Built into the simulation is an opportunity for expanding the simulation to test specific behaviors or conditions. At the top of the configuration file and included in the code are the simulation presets. Currently only the default settings and the “Double Chargers” setting are implemented as part of the simulation. There are many other conditions that may want to be explored. These presets would offer a way to quickly alter the simulation. Some preset examples that would be interesting experiments are: allowing vehicles to charge early if there is no wait for a charger, providing an equal number of each vehicle type, and oversaturating the simulation with a specific type of vehicle.

## Charger Queuing

Currently when a vehicle requires a charger it selects the first charger available, or the charger with the shortest number of vehicles waiting. If there are two chargers with the same number of vehicles waiting the vehicle selects the first one it encounters. Instead the vehicle could assess and estimate the remaining wait time and select the queue with the shortest wait time. A queue with 3 vehicles that charge in 15 minutes each will have a shorter wait time than a queue with 2 vehicles that charge in 45 minutes each.

## Random Number Generation

The simulation uses the simple built in C-style random number generator function rand(). This random number generation could be upgraded to a C++ standard library random number generator. Another option is to write a set of utility functions that provide a random number over a range of values specified as function arguments. The utility functions could provide different types of random values as well, e.g. unsigned integers, floating point values, and double precision values. The user could then select the type needed for the random number.

## Aggregation and Analysis of Simulation Results

Running one simulation provides some details about the behaviors and performance of each vehicle type. Having a large number of simulation results would provide trend data on how the vehicle types perform across many simulations. A method for running and generating statistical data for many simulations would allow for this type of analysis to occur. An additional external analysis tool would be required, such as Matlab, for gathering all the data and generating plots or other statistical data. Having this sort of analysis makes the simulation configuration file even more powerful. The simulation behavior can be changed very quickly to obtain large amounts of data to show vehicle performance. The configuration maximums could be altered or removed to allow for simulations with hundreds or thousands of vehicles to generate performance data over thousands of hours of time. The data would then be reduced to simple graphical format from an analysis script written in Matlab.

# Appendix

Vehicle Type: ALPHA

Num Vehicles: 2

Flight Distance: 463.867 miles, with an average of: 231.933 miles.

Passenger Miles: 1855.47 miles, with an average of: 927.733 miles.

Flight Time: 3.86556 hours, with an average of: 1.93278 hours.

Charge Time: 1.2 hours, with an average of: 0.6 hours.

Wait Time: 0.961389 hours, with an average of: 0.480694 hours.

Num Faults: 4, with an average of: 2 per vehicle

Vehicle Type: BRAVO

Num Vehicles: 6

Flight Distance: 866.667 miles, with an average of: 144.444 miles.

Passenger Miles: 4333.33 miles, with an average of: 722.222 miles.

Flight Time: 8.66667 hours, with an average of: 1.44444 hours.

Charge Time: 1.2 hours, with an average of: 0.2 hours.

Wait Time: 7.93306 hours, with an average of: 1.32218 hours.

Num Faults: 3, with an average of: 0.5 per vehicle

Vehicle Type: CHARLIE

Num Vehicles: 2

Flight Distance: 400.178 miles, with an average of: 200.089 miles.

Passenger Miles: 1200.53 miles, with an average of: 600.267 miles.

Flight Time: 2.50111 hours, with an average of: 1.25056 hours.

Charge Time: 1.6 hours, with an average of: 0.8 hours.

Wait Time: 1.89889 hours, with an average of: 0.949444 hours.

Num Faults: 0, with an average of: 0 per vehicle

Vehicle Type: DELTA

Num Vehicles: 4

Flight Distance: 600.1 miles, with an average of: 150.025 miles.

Passenger Miles: 1200.2 miles, with an average of: 300.05 miles.

Flight Time: 6.66778 hours, with an average of: 1.66694 hours.

Charge Time: 2.48 hours, with an average of: 0.62 hours.

Wait Time: 4.2275 hours, with an average of: 1.05688 hours.

Num Faults: 2, with an average of: 0.5 per vehicle

Vehicle Type: ECHO

Num Vehicles: 6

Flight Distance: 293.083 miles, with an average of: 48.8472 miles.

Passenger Miles: 586.167 miles, with an average of: 97.6944 miles.

Flight Time: 9.76944 hours, with an average of: 1.62824 hours.

Charge Time: 1.8 hours, with an average of: 0.3 hours.

Wait Time: 6.43056 hours, with an average of: 1.07176 hours.

Num Faults: 10, with an average of: 1.66667 per vehicle