Midterm Progress Report – Scott Blender

Project Summary:

This project aims to create a software suite that solves a nominal trajectory given initial and final conditions. Initial and final conditions include three positions (x, y, z) and three velocities (vx, vy, vz). The control states will be determined using the nonlinear least squares minimization determined by the optimality and transversality constraint equations using the Levenberg-Marquardt algorithm to solve the nominal trajectory. After the nominal trajectory is solved, a bundle of trajectories perturbed from the nominal will be solved by perturbing the final mass of the spacecraft to determine new control states. To determine the nominal and bundled end-to-end trajectories, once the control states are determined, the system will be propagated forward and backward in time using the fourth-order Runge-Kutta algorithm to solve the two-point boundary value problems.

Code Outline:

The first task completed was to make a scaffolding of the codebase structure. I have previously written similar code in Python, so a specific collection of auxiliary functions must be written in addition to the two main scripts.

1. Write a function to wrap an angle to the range [0, 2π]
2. Write a function to convert Earth-Centered Inertial (ECI) position and velocity to Modified Equinoctial Elements (MEE).
3. Write a function to perform nonlinear least squares minimization using the Levenberg-Marquardt algorithm given dense Eigen matrices and a residual functor.
4. Write a custom residual functor based on optimality and transversality constraint equations for the Lagrange multipliers of time-optimal trajectory dynamics.
5. Write a script to use the Levenberg- Marquardt algorithm to solve for the control states of the time-optimal trajectory.
6. Write a function to perform fourth-order Runge-Kutta numerical integration.
7. Write a script to integrate the nominal trajectory, solving a two-point boundary-value problem based on initial and final conditions.
8. Write a custom residual functor based on optimality and transversality conditions for trajectory bundle control state determination.
9. Write a script to perform nonlinear least squares minimization to solve for the control states of each bundle, perturbing the final mass constraint to generate new trajectories.
10. Write a script to integrate each new trajectory using the fourth-order Runge-Kutta scheme.
11. Write a user interface to select initial and final conditions.
12. Write a script to export data to be plotted in external software.

Code Progress:

The first two functions and corresponding header files are implemented right now. I also installed the Eigen library. I might make the auxiliary functions into a library to leverage CMake when working on the project.