College of Engineering
Daniel Guggenheim School of Aerospace Engineering

AE 2220: Dynamics

Fall 2022

Final Project - Phase 3 Report

Group 3: Aaron Xiang Trinh, Christopher Wen Wu, Dario Zaccagnino,
Nicholas Meyer-Hisey, and Alfonso Lagares de Toledo

Contents

1 Introduction

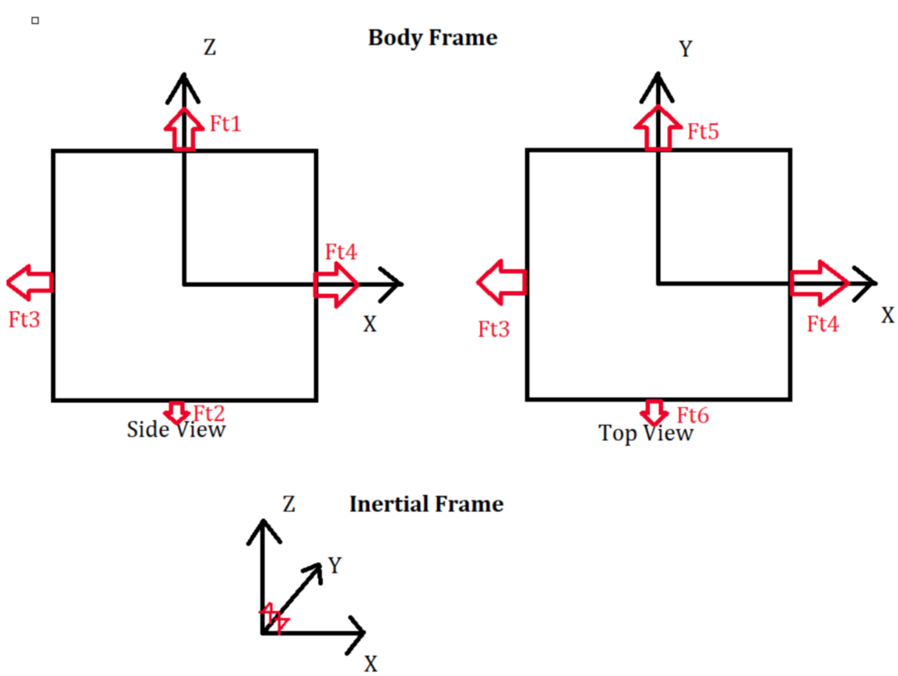
Astrobee is a robotic mission designed by NASA to reduce the amount of time astronauts in the International Space Station (ISS) spend on routine duties such as documenting experiments, moving cargo, and taking inventory of items [1]. It is meant to be a path forward for robotic technologies in space that will help propel the rapid growth of the field of space exploration. Astrobee is designed to have 6 degrees of freedom and be able to fit in a cube with side lengths of one meter. For this project, we have been tasked with developing our own version of Astrobee based on our research, ideas, and discussion as a group. Some design considerations we must address are: how does our version of Astrobee move – both translationally and rotationally – in a zero-g environment; how the robot itself is controlled; the aesthetics of the robot; and other similar factors. For Phase 3, we are discussing the three-dimensional dynamics theory of our robot, 3D simulations with the robot, case studies, and finally potential design improvements that could be made to our robot.

2 Dynamics Theory

Equations of motion for both translation and rotation have been derived specifically for how our robot will move throughout spaces such as the ISS. These have been presented in the subsections below.

2.1 Free-Body Diagram

The free-body diagram of the robot in the body frame relative to the inertial frame is shown below.



2.2 Three-Dimensional Equations of Motion: Translation

For translation, we will be using compressed air thrusters. Thus, we can model the entire robot as a rigid body and say that:

$$\mathbf{F}_{thrust_x} = m\frac{d\mathbf{v}_x}{dt} \tag{1}$$

$$\mathbf{F}_{thrust_y} = m\frac{d\mathbf{v}_y}{dt} \tag{2}$$

$$\mathbf{F}_{thrust_z} = m\frac{d\mathbf{v}_z}{dt} \tag{3}$$

Since all six faces have thrusters, it will not be necessary to rotate the robot in order to translate in any direction. For 3D motion, this means that the equations of motion for the translation of the vehicle are:

$$\ddot{\mathbf{x}} = \frac{\mathbf{F}_{thrust_x}}{m} \tag{4}$$

$$\ddot{\mathbf{y}} = \frac{\mathbf{F}_{thrust_y}}{m} \tag{5}$$

$$\ddot{\mathbf{z}} = \frac{\mathbf{F}_{thrust_z}}{m} \tag{6}$$

2.3 Three-Dimensional Equations of Motion: Rotation

For rotation, we are using control moment gyroscopes. These operate based on the principle that moving a rotating disk via a gyroscope will change the direction of the rotating disk's angular momentum, and by conservation of momentum, this will impart a net torque on the body of the robot. When rotating the reaction control wheel along its axis on any plane, the resultant x, y, or z-component of the angular momentum can be written as:

$$\mathbf{H} = \mathbf{I}\boldsymbol{\omega} \tag{7}$$

where $\mathbf{I}$ is the moment of inertia tensor of the robot and $\boldsymbol{\omega}$ is the angular velocity.

Now, for three-dimensional rotation, we need to consider not only the angular momentum in the z-axis but also that of the x and y directions. Below is a diagram that depicts the different angle variations for both the x- and y-axis angular momenta. θ denotes the angle for the x-axis angular momentum and ψ denotes the angle for the y-axis angular momentum. The blue rectangles situated at each corner of the mount denote the control moment gyroscopes.

Looking at the figure above, we can derive an expression for the net torque in the x and y directions. For the x-axis:

$$\Sigma H_x = 2\mathbf{H}(\sin(\theta(t)) - \cos(\theta(t))) \tag{8}$$

and for the y-axis:

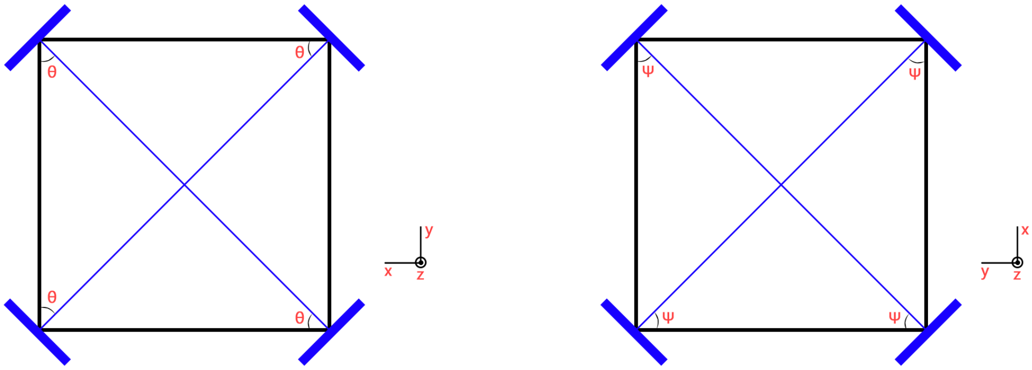


Figure 1: Depiction of angle θ and angle ψ

$$\Sigma H_y = 2\mathbf{H}(\sin(\psi(t)) - \cos(\psi(t))) \tag{9}$$

And recalling from the Phase 2 report:

$$\Sigma H_z = 4\mathbf{H}\sin(\phi(t)) \tag{10}$$

Thus, we can write the net angular momentum as:

$$\Sigma\mathbf{H} = \begin{bmatrix} 2\mathbf{H}(\sin(\theta(t)) - \cos(\theta(t))) \\ 2\mathbf{H}(\sin(\psi(t)) - \cos(\psi(t))) \\ 4\mathbf{H}\cos(\phi(t)) \end{bmatrix} \tag{11}$$

Again, by Euler's Second Law, we know that the torque on the body caused by the rotation of one of the control moment gyroscopes is given as:

$$\boldsymbol{\tau} = \frac{d\mathbf{H}}{dt} = \omega_{max} \cdot \begin{bmatrix} 2\mathbf{H}(\cos(\theta(t)) + \sin(\theta(t))) \\ 2\mathbf{H}(\cos(\psi(t)) + \sin(\psi(t))) \\ -4\mathbf{H}\sin(\phi(t)) \end{bmatrix} \tag{12}$$

Regarding the quantity ω_{max}, this is a positive value that represents the maximum angular velocity for which the motor turning the gyroscope assembly is rated. For our purposes, we will assume that as soon as the gyroscope begins to turn, it immediately reaches its maximum angular velocity. This simplifies the model since ω_{max} is assumed to be constant with respect to time. ω_{max} is taken to be the maximum angular velocity of the motors used as a guideline for the vehicle design in Phase 1, which was 6720 rpm.

3 Coding and Simulation

3.1 Three-Dimensional Translation

Translation in three dimensions is an extension of 2D translation, as the equations of motion in all three axes are equal. Determining a path between a set of waypoints does become more complicated, as now three dimensions need to be considered. We used the function *bsplinepolytraj* to generate spline paths between a set of waypoints. This guarantees a smooth translational motion from point A to point B, and the path of the vehicle between the two points can be manipulated using intermediate waypoints. This method for determining translation yields the intended vehicle position, velocity, and acceleration at all points of the motion. This allows the thruster actuation rate in the world axes F_r to be determined for all points in the motion using the known acceleration values a^* by computing:

$$F_r = \frac{a^*}{m} \tag{13}$$

As an example, this is a 3D trajectory modeled by our simulation. The trajectory moves the vehicle up by 4 meters while following a spiral movement generated by waypoints arranged in a square. The waypoints are shown in blue, while the final spline trajectory is shown in red.

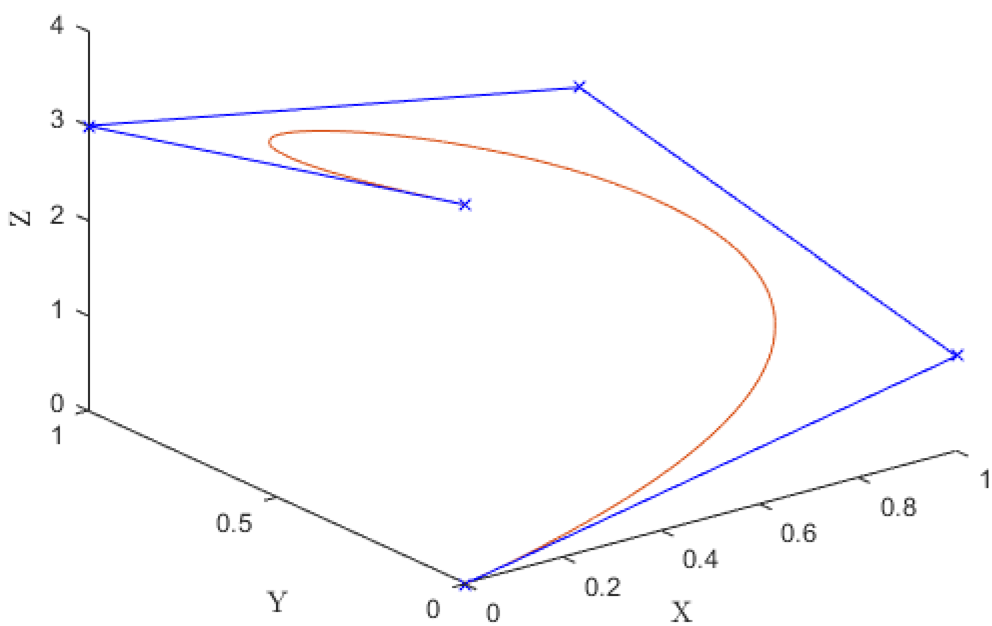


Figure 2: Spiral trajectory

The velocity and acceleration graphs for this trajectory are also generated:

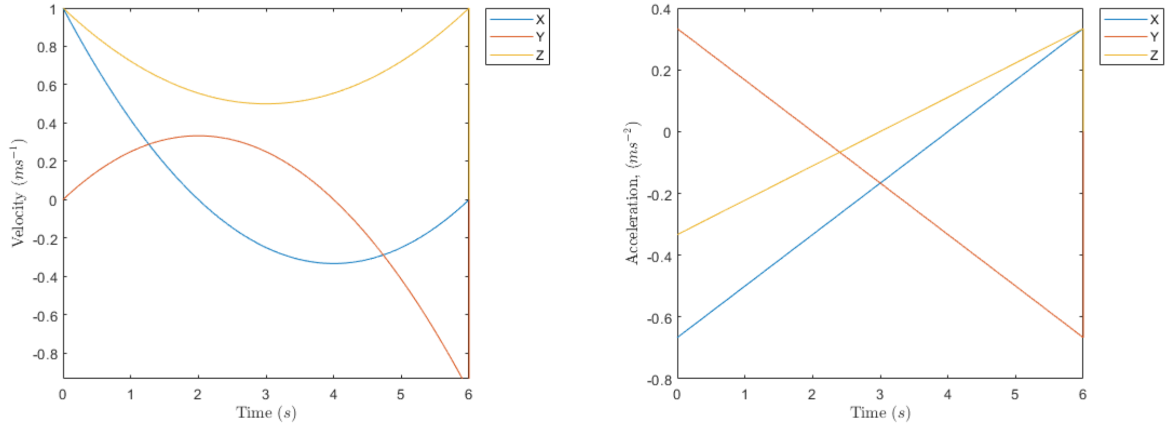


Figure 3: Velocity and acceleration along a spiral trajectory

Finally, thruster actuation can be found using equation 13

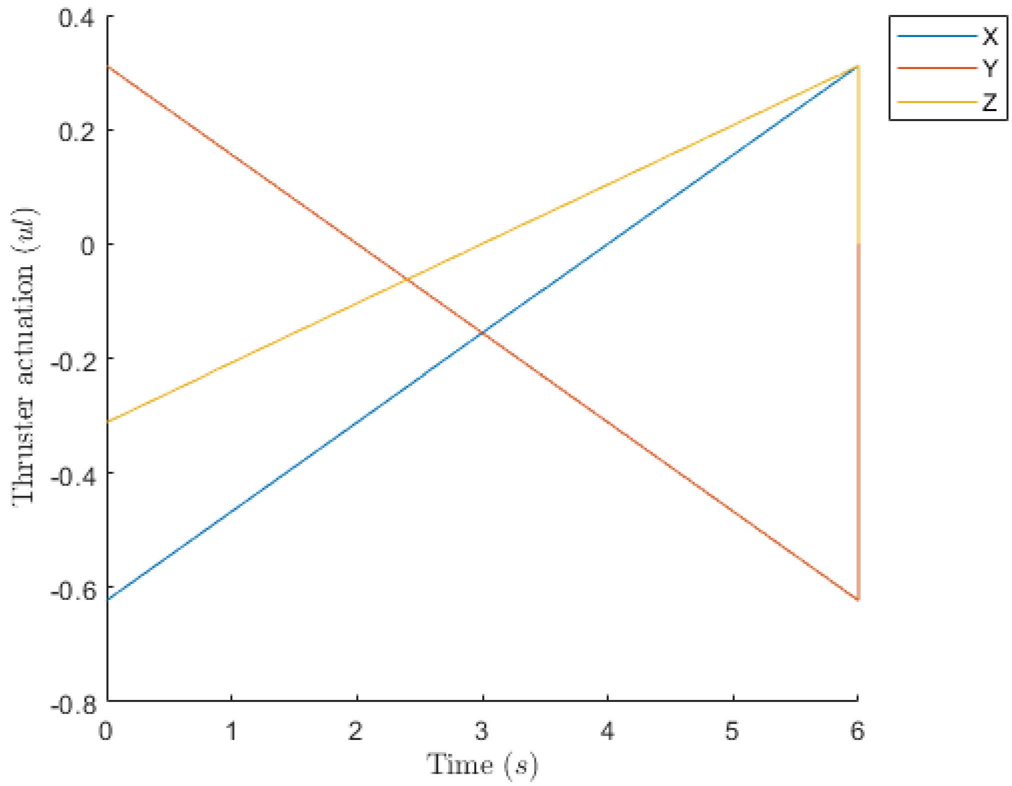


Figure 4: Thruster actuation in world reference frame along motion

We can calculate the error between an ideal straight trajectory along the waypoints and the computed spline trajectory by finding the absolute error between the straight trajectory and the spline path using the following formula:

$$\boldsymbol{E}_a bs = \sum_{i=1}^{n} |\vec{x}_{spline} - \vec{x}_{straight}| \tag{14}$$

By applying a color gradient to the path in terms of each point's absolute error, we can find the path sections that deviate the most from the straight path.

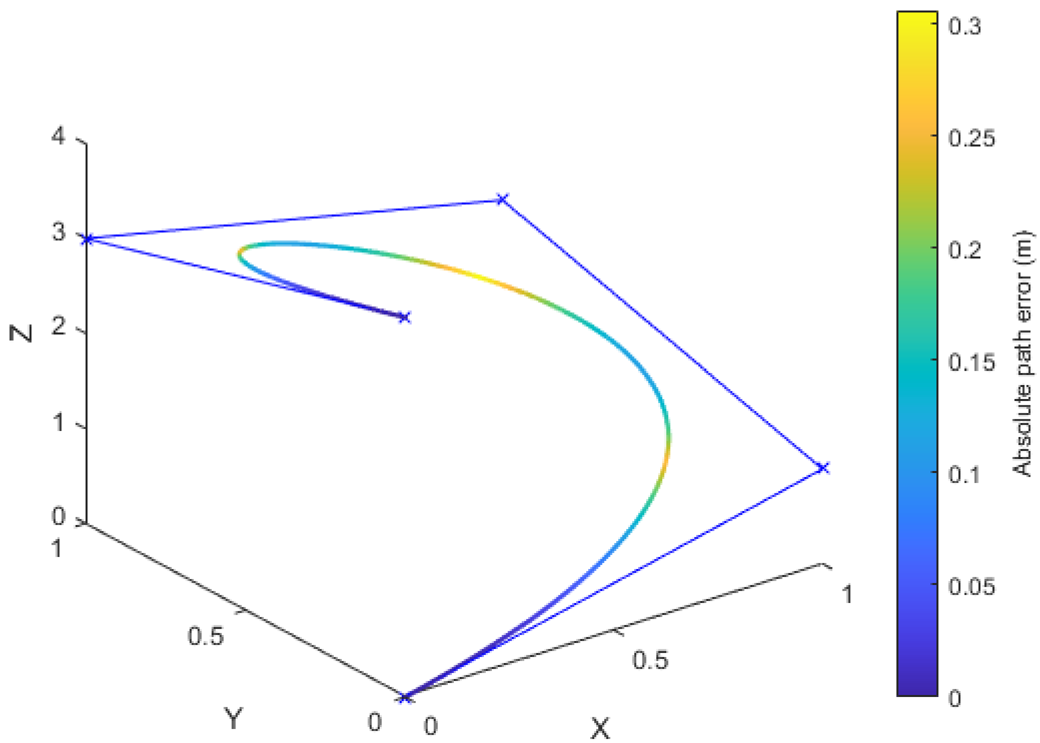


Figure 5: Absolute error between spline and straight path

3.2 Three-Dimensional Rotation

Rotation is more complicated to model due to the vehicle having asymmetric torque capabilities. This means that the vehicle is able to impart torque on itself more in the z axes than in the x or y body axes. This is because when rotating about the z axes all four reaction control wheels provide torque in the same axes while in the x and y axes the torque of two reaction wheels counteract the torque of the other two. Net torque is achieved in the x and y axes by causing an imbalance in the torques applied by each pair of wheels.

To make an analysis of the rotational dynamics of the vehicle simpler, we chose our Euler angles to be ψ, θ, ϕ, each corresponding to rotations about the z, y, and x axes. This means that for an Euler angle orientation adjustment the reaction control wheels will first rotate about their pitch axis to match the desired yaw rate. This makes the subsequent pitch and roll operations independent of the yaw operation, as the reaction control wheels all stay in the same plane.

We chose to drive our rotations using a sigmoid function, as it provides smooth acceleration throughout the motion. The resultant angular velocity curve, which ends up being the positive section of a sine curve, can be followed by the reaction control gyroscopes to provide accurate rotation. Once we find the angular velocity we want to meet, the angular momentum needed is calculated and the resulting actuation angle for the reaction control wheels is found.

4 Case Studies

4.1 Euler Angle Orientation Adjustment

To complete a rotation by a series of Euler angles, the vehicle performs sigmoid-driven yaw, pitch, and roll along its principal axes. For example, the Euler angles, angular velocity, and reaction control gyroscope actuation through sequential 180-degree rotations along the principal axes are:

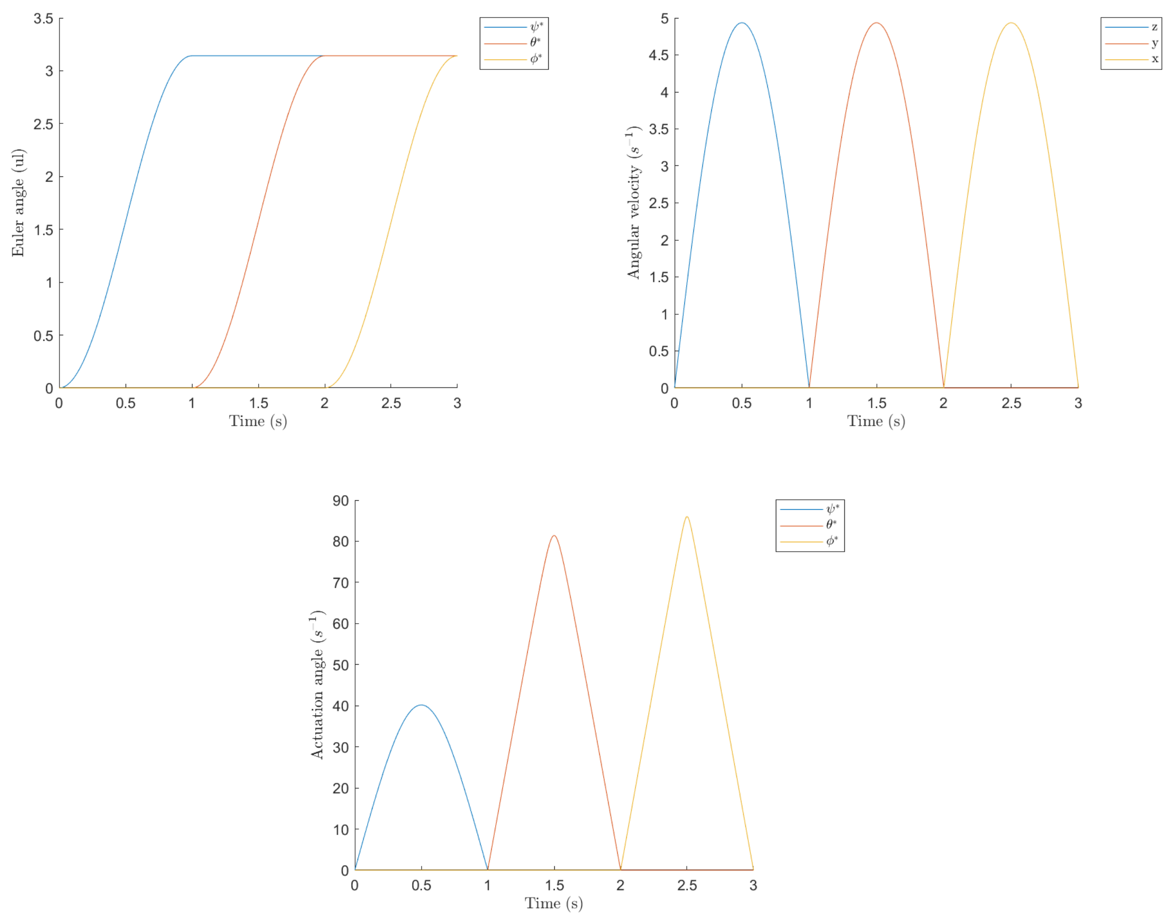


Figure 6: Euler angles, angular velocity, and RCG actuation angles for a 180 yaw-pitch-roll maneuver.

4.2 Translation and Rotation

Combining the resulting translation and rotation techniques, we can simulate complex 3D trajectories. For this test case, we simulated the vehicle completing a total of five spiraling turns around the z-axis while raising a total of 25 meters. The vehicle is expected to complete this path in 90 seconds. While doing so, the vehicle performs a complete 360-degree yaw, pitch, and roll rotation, each lasting 30 seconds. The resulting plots are not shown here due to a lack of space. They are included in Annex A. Note that the reported thruster actuation rates are given in the world axis and not the body axis of the vehicle. The corresponding body axis thruster actuation rates can be found using direction cosine matrices, where the yaw, pitch, and roll values are given by the Euler rotation angles of the vehicle at any given time.

5 Potential Design Improvements

When simulating the vehicle design, we realized the actuators used were oversized for the vehicle size and safety requirements inside the ISS. Thruster force was able to accelerate the vehicle to its maximum safe operating speed of $0.5ms^{-1}$ in only 0.0467 seconds. For our next design, we must be either more conservative with our thrust estimates or find a less powerful thruster technology to ensure safety when operating inside confined spaces.

The reaction control gyroscopes designed also suffered from excessive torque requirements. Although their corrective power would be beneficial if the vehicle experience any external torque, the high moment of inertia of the system would require very precise control of the angle to which they are rotated. This control would be difficult to achieve, so over-torque events are likely. In order to not endanger the crew or the ISS, lighter reaction control gyroscopes with lower moments of inertia would be beneficial. Another option would be to slow down the rotation speed of the reaction control gyroscopes in order to reduce their moment of inertia.

Another major issue faced by our vehicle, specifically with regard to its control algorithm, is that it lacks a feedback loop. This means small errors in the translation or rotation of the vehicle or random external inputs will make the vehicle drift and lose its ability to control its attitude in space. Including a feedback control loop would help to minimize such drift, if not eliminate it altogether.

6 Annex A: Translation and Rotation Case Study Figures

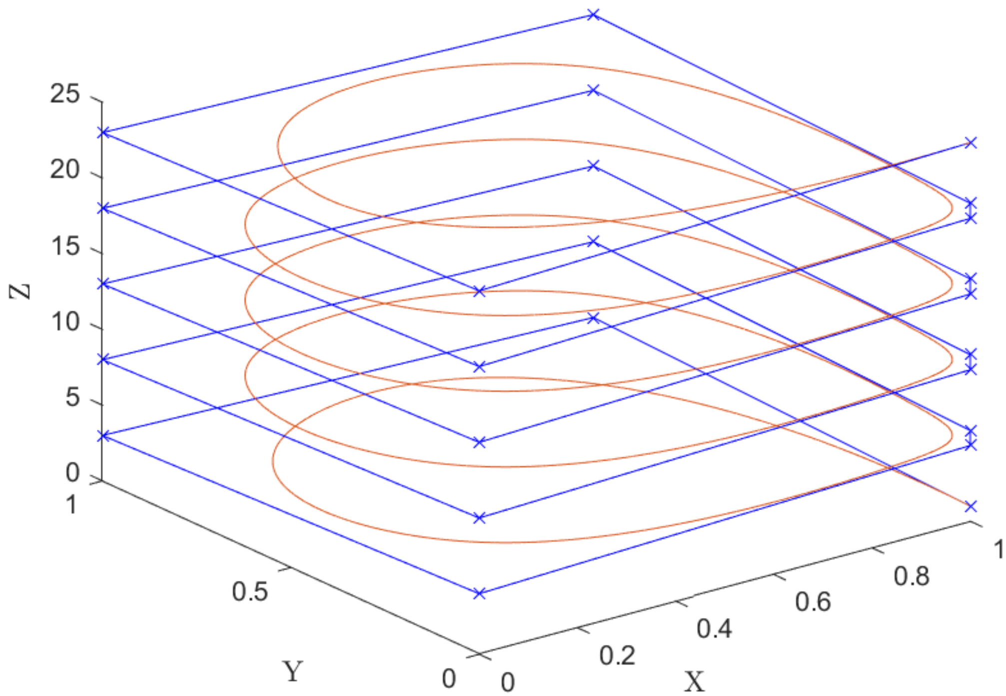


Figure 7: Vehicle Trajectory

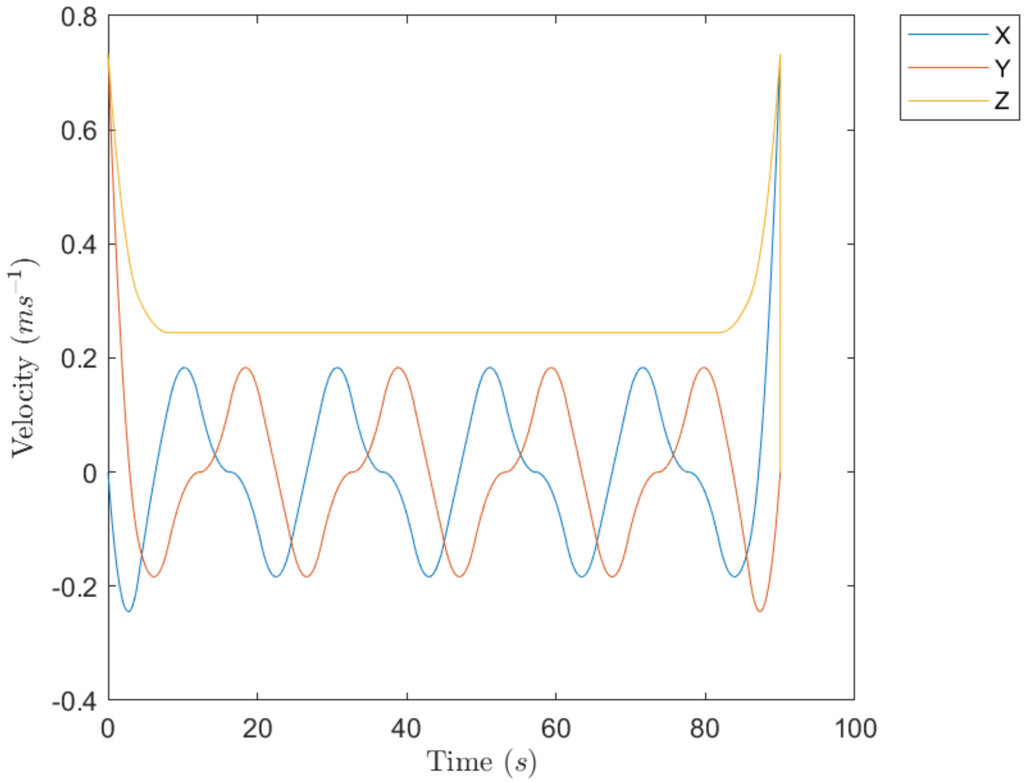


Figure 8: Vehicle Velocity

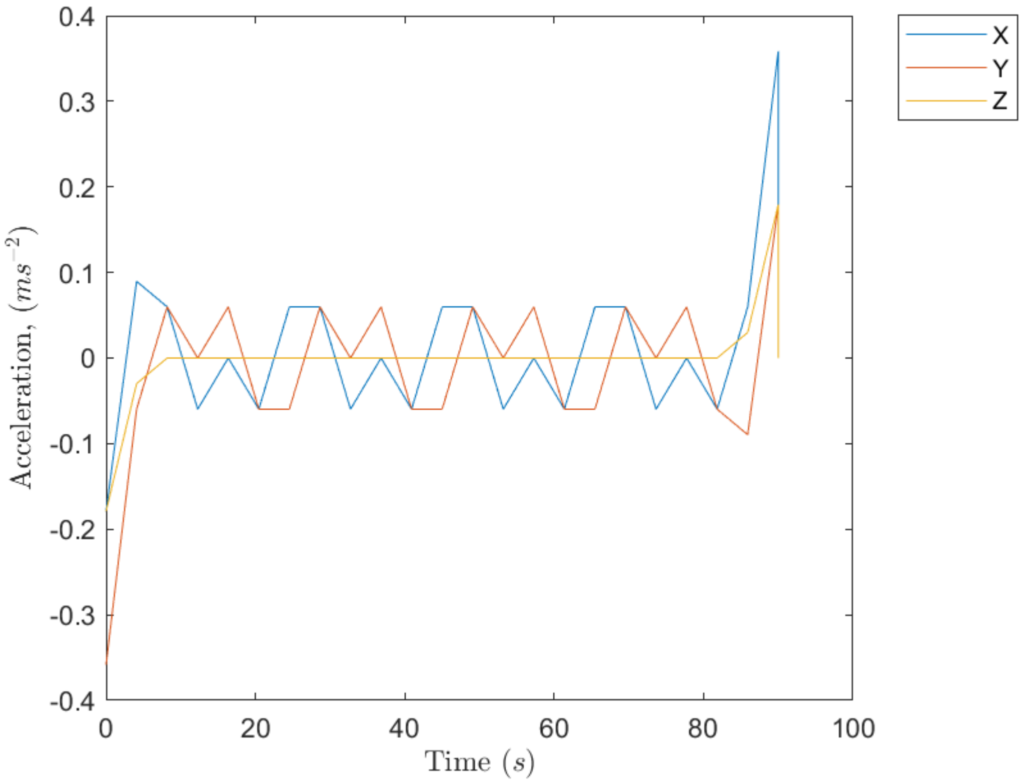


Figure 9: Vehicle Acceleration

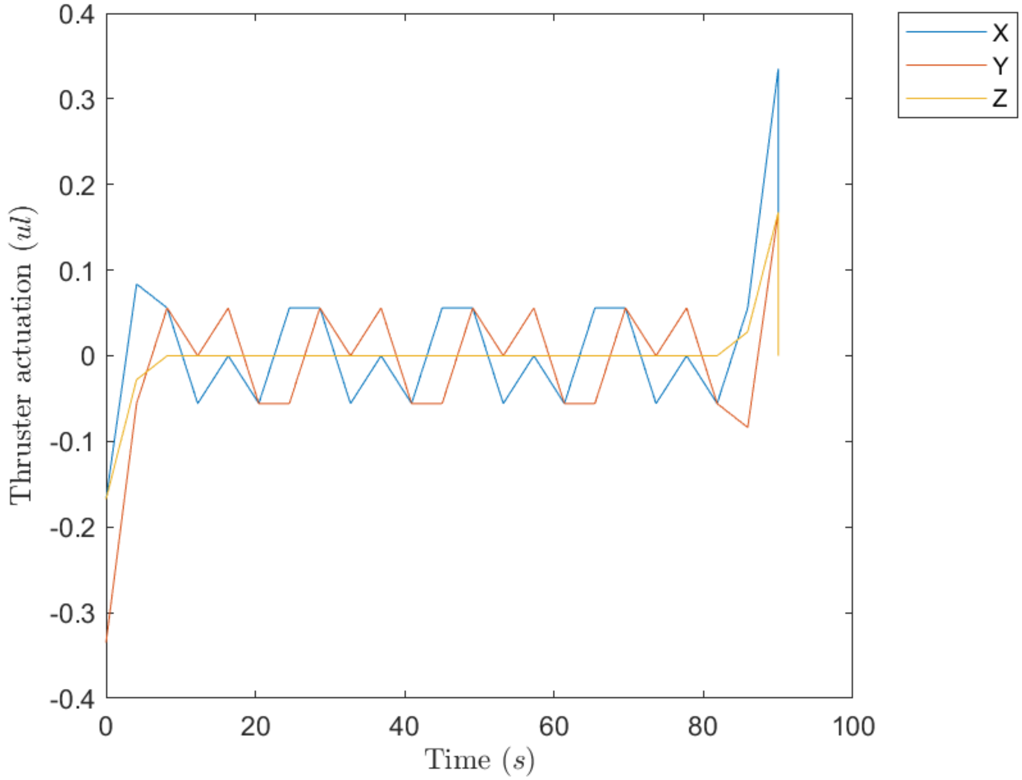


Figure 10: Vehicle Thruster Acceleration

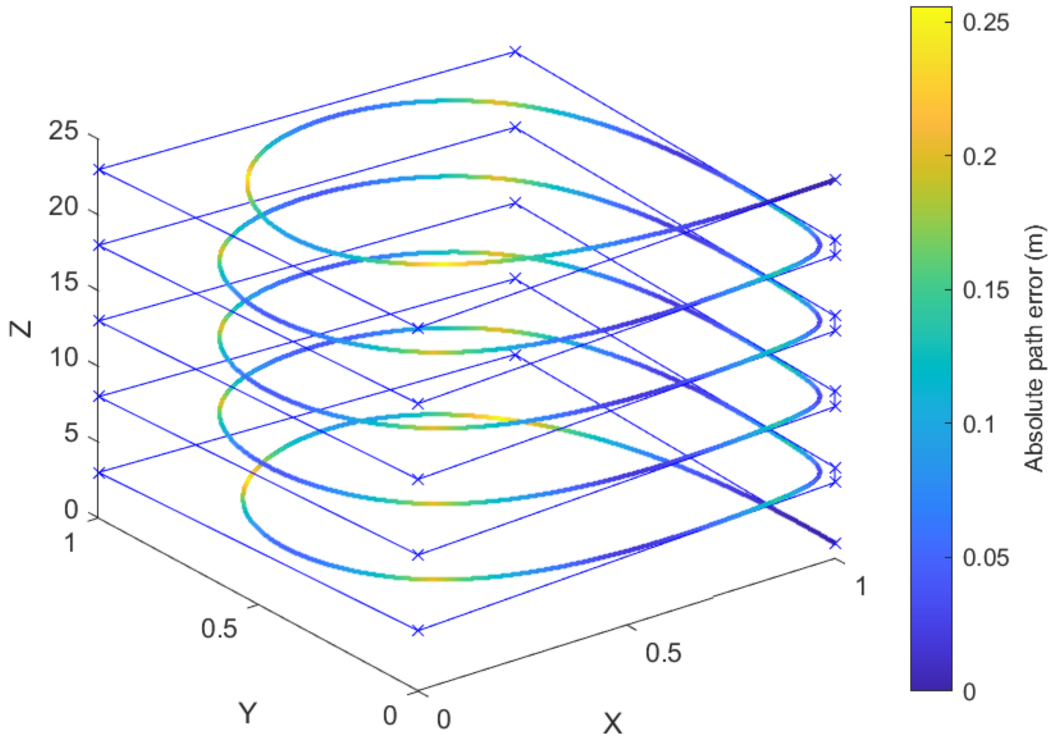


Figure 11: Vehicle Trajectory Error

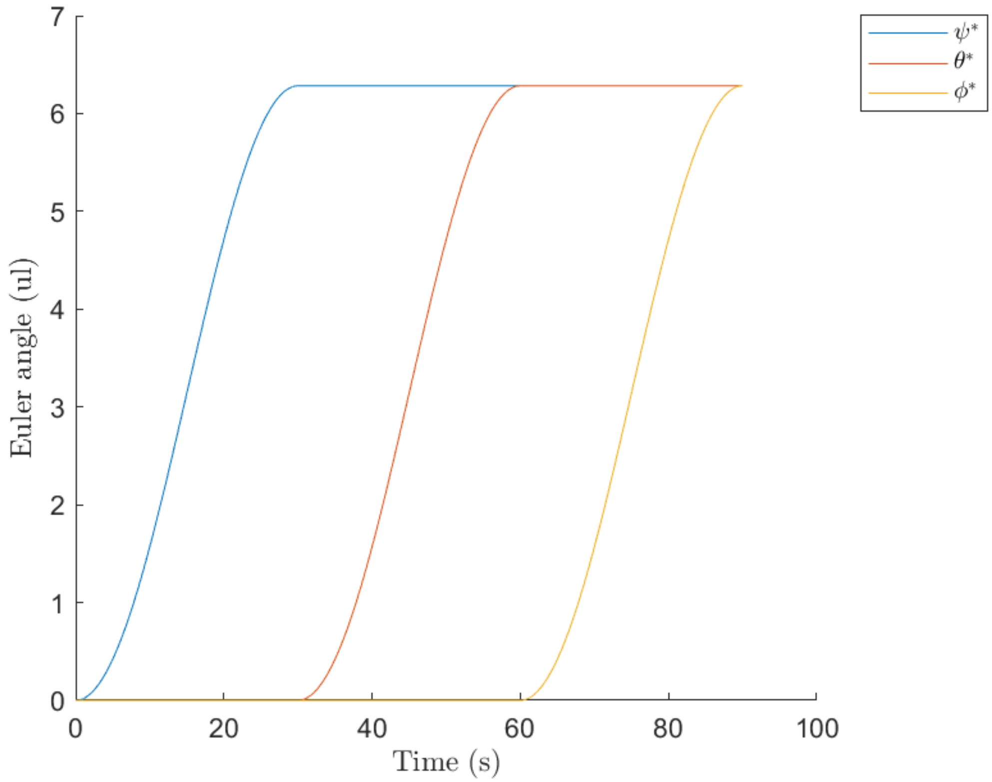


Figure 12: Vehicle Euler Angles

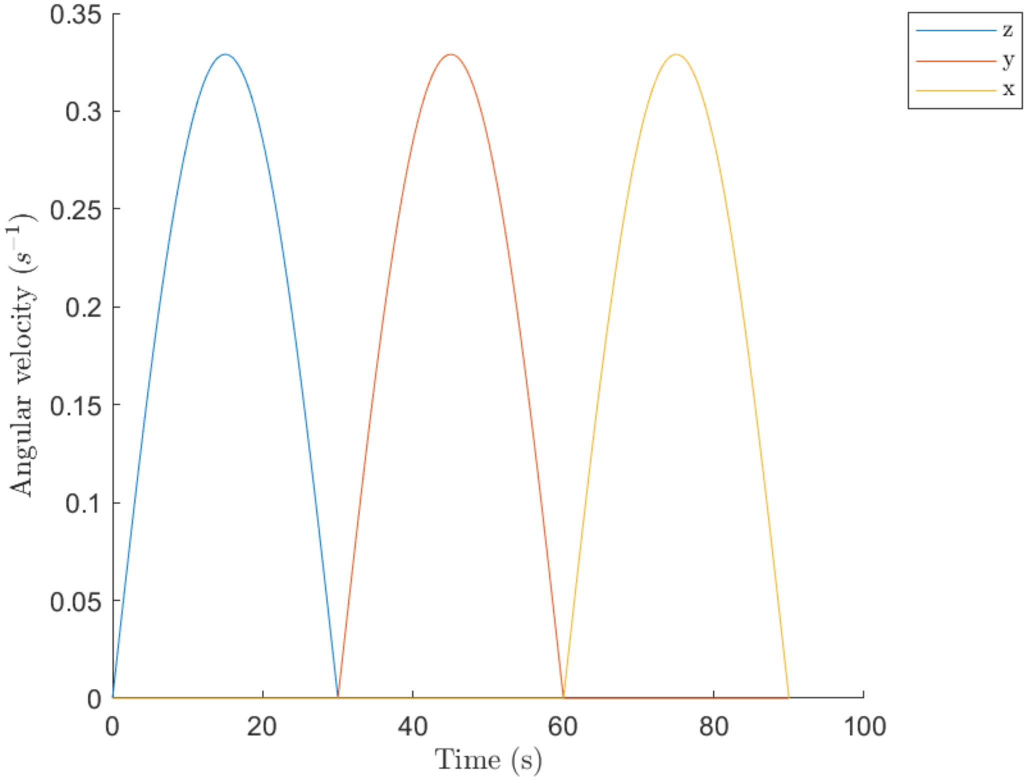


Figure 13: Vehicle Angular Velocity

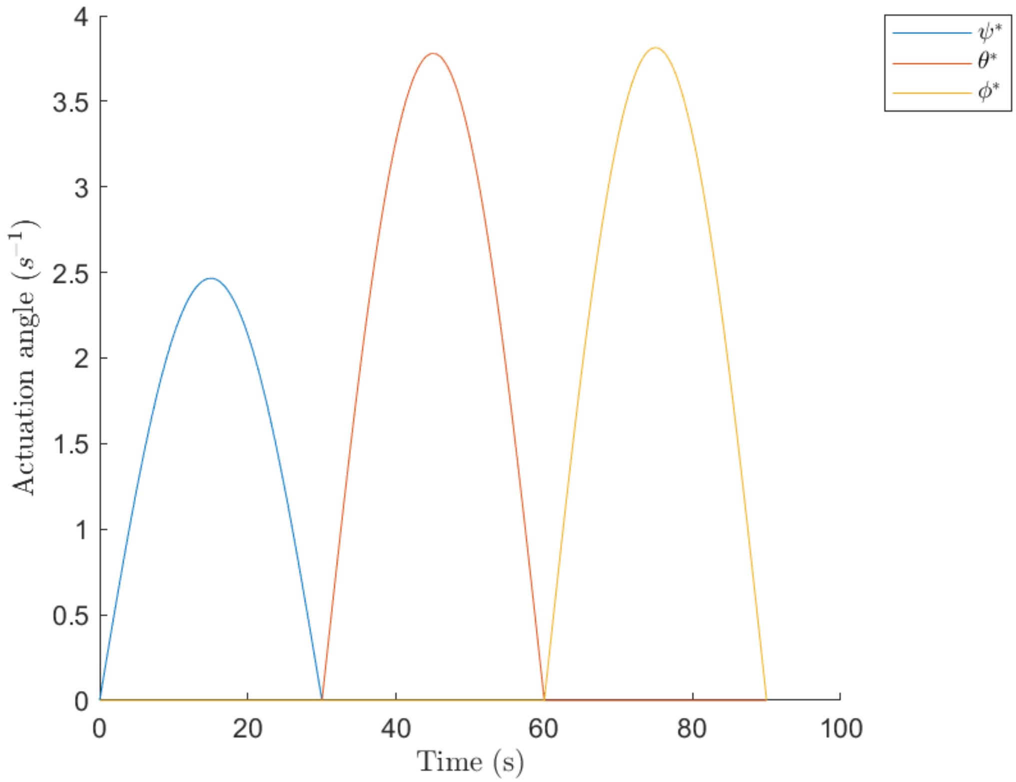


Figure 14: Vehicle RCG Actuation

References

[1] Kanis, S., “What is Astrobee?,” NASA Available: https://www.nasa.gov/astrobee.