College of Engineering
Daniel Guggenheim School of Aerospace Engineering

AE 2220: Dynamics

Fall 2022

Final Project - Phase 2 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 2, we are discussing the dynamics theory of our robot, developing code to simulate various missions 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 Two-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:

$$\vec{\mathbf{F}}_{thrust} = m\frac{d\vec{\mathbf{v}}}{dt} \tag{1}$$

Since all six faces have thrusters, it will not be necessary to rotate the robot in order to translate in any direction. Thus, equation (1) applies in all three axes. For 2D motion, this means that the equations of motion for the translation of the vehicle are:

$$\ddot{\vec{\mathbf{x}}} = \frac{\vec{\mathbf{F}}_{thrust}}{m} \tag{2}$$

$$\dot{\vec{\mathbf{x}}} = \ddot{\vec{\mathbf{x}}}t \tag{3}$$

$$\vec{\mathbf{x}} = \dot{\vec{\mathbf{x}}}t \tag{4}$$

2.2 Two-Dimensional Equations of Motion: Rotation about the z-axis

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 the xy plane, the resultant z-component of the angular momentum can be found through trigonometry:

$$H_z = \mathbf{H}\cos(\phi) \tag{5}$$

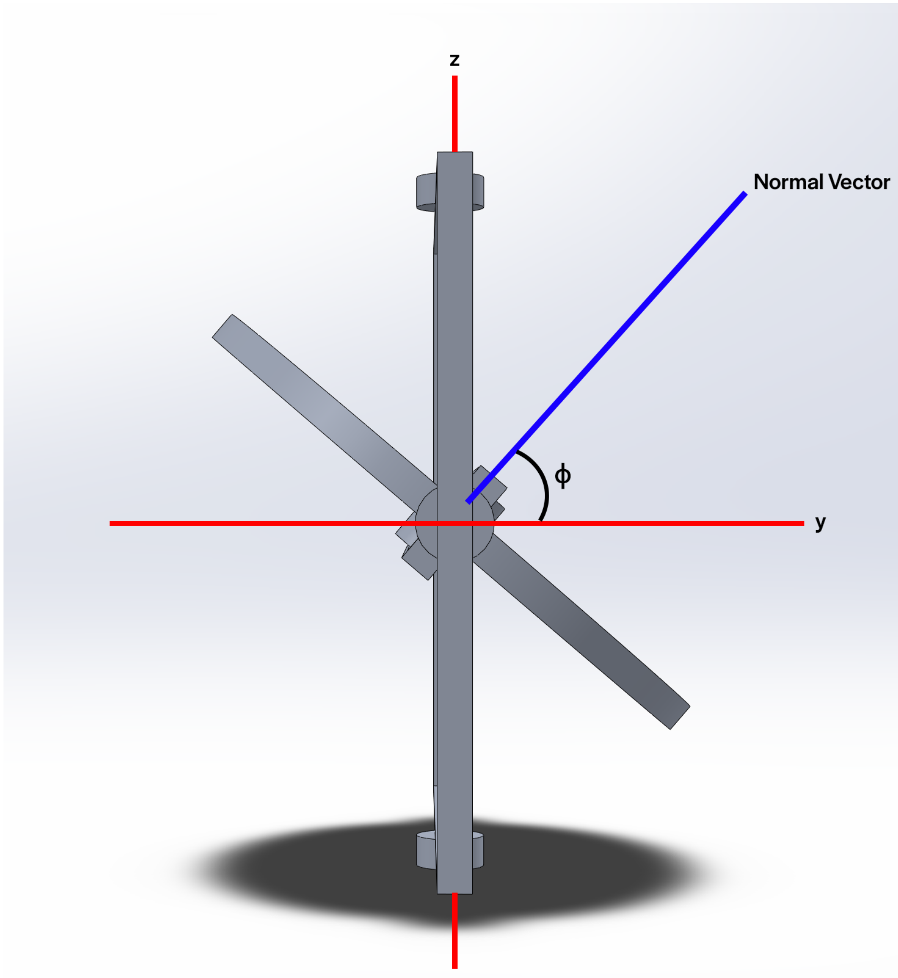

Figure 1: Depiction of angle ϕ

where ϕ is the angle between the y-axis and the vector $\mathbf{H}$ normal to the flywheel surface, as shown below.

Thus, 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:

$$\tau_z = \frac{dH_z}{dt} = -\mathbf{H}\sin(\phi(t)) \cdot \omega_{max} \tag{6}$$

The vehicle includes four reaction control gyroscopes that counteract each other's x and y angular momentum components. To calculate the torque exerted on the body by the rotation of all four gyroscopes when $\dot{\phi}$ is positive with respect to the z-axis, we simply sum up the torques.

$$\Sigma\tau_z = \sum_{i=1}^{4} \tau_{z_i} = 4\tau_z = -4\mathbf{H}\sin(\phi(t)) \cdot \omega_{max} \tag{7}$$

Thus, in summary, the vehicle dynamics 2D rotation equations are as follows:

$$\tau_{z_{net}} = -4\mathbf{H}\sin(\phi(t)) \cdot \omega_{max} \tag{8}$$

$$H_z = \mathbf{H}\cos(\phi) \tag{9}$$

$$\omega_z = \frac{H_z}{I_z} \tag{10}$$

$$\phi = \omega_z t \tag{11}$$

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

A simulation of the vehicle was built using Matlab. The values for the different state variables involved in the simulation (vehicle mass, moment of inertia, etc.) were taken from our Phase 1 report.

A numerical approach was taken when simulating the dynamics of the vehicle. This means that changes in the system were computed over very small amounts of time and their impact on the state variables was found by computing the derivatives presented in their differential form. All code needed to run the simulator is included in the submission.

3.1 Two-Dimensional Translation

The vehicle was first simulated with a constant velocity in the x and y directions to check that translation was being simulated correctly. A constant speed of $0.5ms^{-1}$ in both axes was used.

For a translation-only pathing, a simple control algorithm can be implemented. Since rotation is not needed, only the thrusters will be used to complete a translation-only path. There are multiple ways this can be done, but most notably in two distinct ways:

1. Firing a thruster until the maximum allowable speed is reached and then firing the opposite-side thruster to bring the speed to zero
2. Firing a thruster until the halfway point of the linear trajectory, and then instantly firing the opposite-side thruster to bring the speed to zero

The first method will achieve the shortest path time. However, the second method will use less fuel, as the thrusters are firing for a shorter amount of time. Additionally, since the speed is capped at a safe allowable speed, it will be safer for the astronauts and the space station in general if a malfunction occurs. For all future case studies, we will utilize the second method, as it is the safest. For a maximum allowable speed inside the ISS, we will use Astrobee's real maximum allowable speed of 0.5 ms^{-1}.

To traverse a section of a path safely, the vehicle will first fire its thrusters in the direction that it needs to travel in. Before reaching its desired final position, the vehicle will fire its opposing thrusters to come to a complete stop.

The vehicle first accelerates from rest until it reaches the desired velocity in both axes. The vehicle is then allowed to coast for 5 seconds, after which the vehicle decelerates completely until it is at rest. The results of the simulation are shown below:

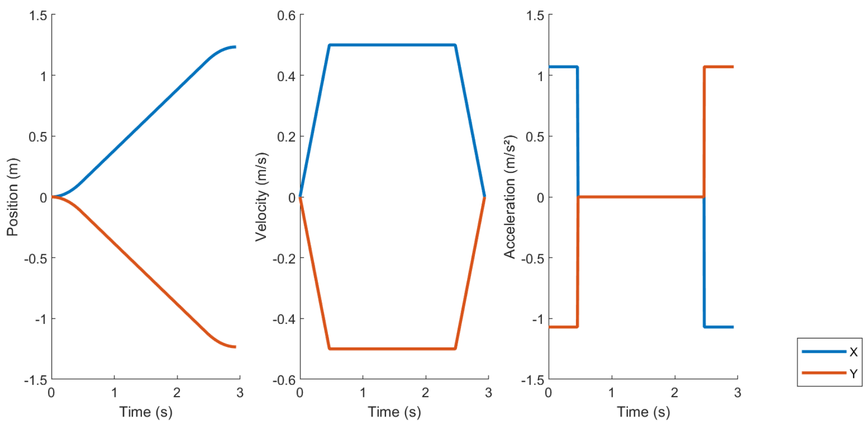

Figure 2: Translation at $0.5ms^{-1}$ in the x and y axes

3.2 Rotation about the z-axis

For this case study, only rotational motion is allowed, which means no thrusters will be fired. This is as simple as tilting the gyroscopes following the equations of motion derived above in order to get the desired rotation. This will be used when the robot needs to face a certain way, whether it is to pick up an item, push a button, or perform any other tasks that require the robot to face in a specific direction.

To test the simulation of rotation, the vehicle was rotated at a constant angular velocity of 1 revolution per second. The vehicle was then allowed to rotate at a constant angular velocity for 2 seconds. Finally, the reaction control gyroscopes were actuated to slow down the vehicle rotation until its angular velocity was zero. The results of the simulation are shown below:

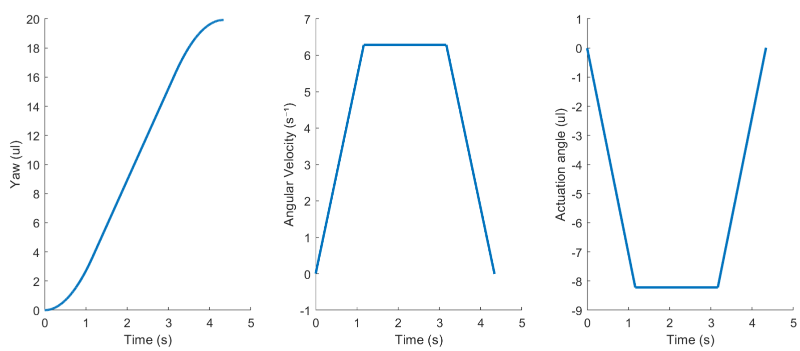

Figure 3: Rotation at a constant angular rotation speed

4 Case Studies

4.1 Translation and Rotation

Since this case study now involves both translation and rotation, it is not as simple as the previous two. The path to be followed was broken down into only translation or only rotation sections and the control algorithms described above were applied to each corresponding section. To showcase this, the robot was commanded to fire its thrusters and reach its maximum allowed speed before completing a 180-degree turn. The robot then used the same thruster to bring itself to a complete stop. The results of the simulation are shown below.

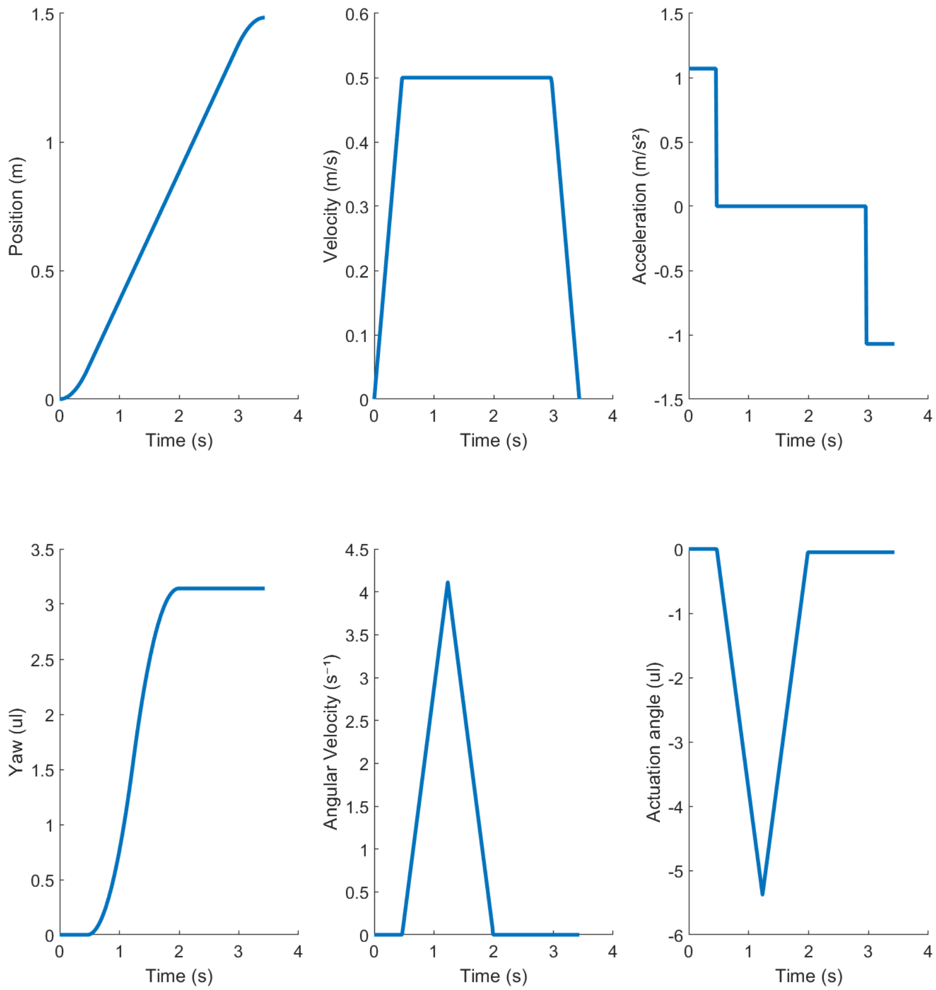

Figure 4: Translation and rotation throughout the same motion

4.2 Avoiding a Static Obstacle

This case study shows how obstacle avoidance might be performed using this vehicle. There are two approaches that might be used for this purpose: a purely translational approach, in which the vehicle uses only its thrusters to deviate around the obstacle, or a translational-rotational approach, in which the vehicle turns in combination with firing its thrusters to reach the desired positions. We chose to use the purely translational approach, as the use of opposing thrusters makes controlling the translation of the vehicle much simpler than using its reaction control gyroscopes.

For the translational approach, each of the path sections was treated as a single translational movement. The obstacle was placed 3 meters from the starting point of the vehicle and was assumed to have a size of $1 \times 1 \times 1$ meters. The vehicle was instructed to maintain a minimum distance of 0.25 meters from the obstacle. When going around the obstacle, the thrusters in the y direction of the vehicle were fired as the thrusters on the x component were braking the vehicle. The result is a curved motion around the corners of the path which reduces travel time and avoids having to bring the entire vehicle to a complete stop before each direction change. The results of the simulation are shown below.

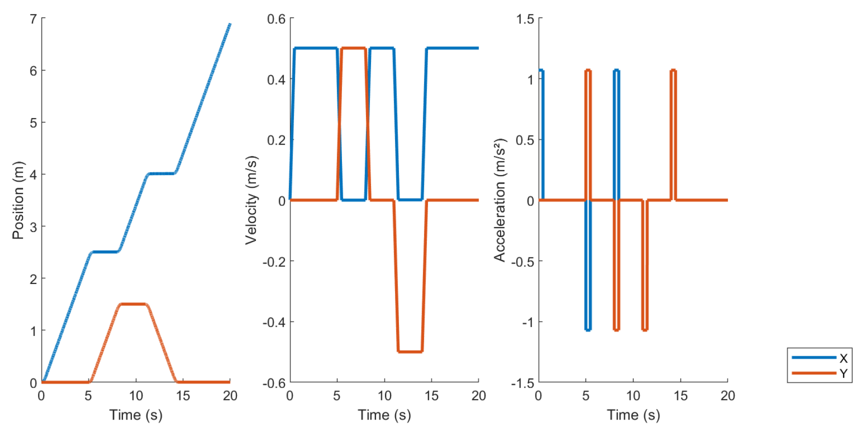

Figure 5: Avoiding an obstacle by going over it with purely translational motion

4.3 Completing a Mission

Finally, the full capabilities of the vehicle will be displayed through a complex mission. The mission will start at the origin and the vehicle will have to complete the following tasks:

- Move at the maximum allowed rate toward the right
- Avoid two obstacles as it moves forward
- Rotate 90 degrees while still moving forward
- Come to a complete stop after rotating

The results of this simulation are shown below:

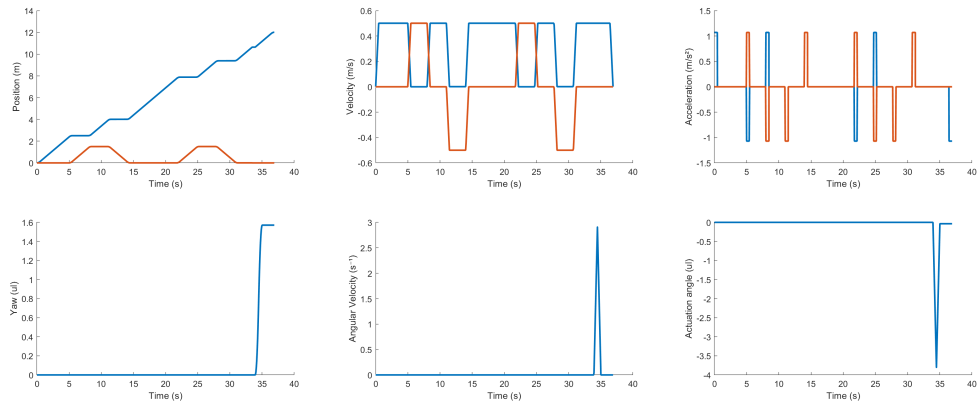

Figure 6: Simulating a complete mission with two obstacles and a 90-degree rotation

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 at which they are rotated to. 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 regards 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.

References

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