Space System Design Studio(SSDS) Alpha CubeSat: Attitude Control System(ACS) Research Summary

Alpha CubeSat is a 1-unit cube-shaped satellite projected to be launched in 2024 and will carry the world's first retroreflective, solo-flying light sail—paving the way for future missions to our nearest stellar neighbor, Alpha Centauri. Under Prof. Mason Peck's guidance and Joshua Umansky-Castro's mentorship, my role involves researching, developing, and verifying the onboard Attitude Control System (ACS). With the intention to build a working ACS that doesn't require expensive hardware, thus given the observable inputs (the data from the Inertia Measurement Unit (IMU)) and the controllable outputs (the current sent to the magnetorquers for the three axes), we aim to design a system that assists the satellite to de-tumble/spin-stabilize when it is released from the International Space Station (ISS) and then points north or south afterward for the stable release of the light sail.

The ACS controller and plant (dynamics simulator) prototype was built in SIMULINK based on a paper by a previous ACS team member, and they were auto-coded as C++ libraries. However, its performance and feasibility were unknown for the actual hardware. Our test setup consists of an air-bearing that holds our CubeSat to simulate a zero-gravity environment and a Helmholtz coil with adjustable current to simulate the magnetic field of low orbit. We realized the importance of maintaining a near-perfect balance of the CubeSat on the air-bearing, and one of the challenges we faced was solving the constant air-torque problem caused by imperfect airflow. For this, I designed/3D-printed a set of "airflow regulators," which blow air onto the cup holding the CubeSat to counter the air torque. We saw an unusually high magnetorquers current output with the cup relatively stabilized. Comparing outputs from a noiseless input and a noisy input, I was able to confirm that our system is extremely sensitive to noise. Referencing the procedures and equations from a paper that describes a similar application, I collected data from the IMU and calculated/plotted various noise characteristics, including Allan Variance (which measures the stability across different frequencies), Power Spectral Density (which measures the strength of the noise across frequencies), and the Probability Density Function (which concludes the Gaussian nature of the noise). I have developed filters to denoise the IMU data, including a moving kernel filter (which defines kernel functions to process a given array of past data as filtered output), and a digital low pass filter to eliminate high-frequency noises. To develop effective filters, I used simulated or collected data to design the filters offline, determined the optimized parameters, and then tested them on the real-time setup, verifying to smooth the data. After that, I partnered with another graduate student to develop the Extended Kalman Filter (EKF). In addition to the noise analysis I conducted, a covariance matrix was computed for the observation part of the filter.

I dug into the original SIMULINK file to solve some of the fundamental problems with our controller. With a full review of the file, I was able to debug the issues related to discrete-time differentiation and parameter scaling. And I found a way to adjust the model parameters after it is auto-coded into C++. With the corrected controller library, I developed an ACS simulation to run on computers, speeding up the simulation time and enabling us to perform batch long-duration simulations. For example, our pointing controller is a PD controller with two main parameters to tune: proportional and derivative gain. To find a range of parameters that converge, I designed a Monte Carlo simulation to generate a uniform pair of Kp and Kd in the log scale. With the previously mentioned codebase, I generated a thousand points and defined a range of convergence, which provided a set of potential candidates of parameters we should uplink when the CubeSat is in space.

Given a limited power budget, we aim to activate the magnetorquers as infrequently as necessary. Therefore, I have also simulated the power consumption at different duty cycles and characterized the acceptable range of these cycles.

With all the work described above, I could perform actual hardware experiments on the air-bearing and verify the ACS algorithm could converge to the desired direction.

In addition to the ACS, I am also involved in other parts of the mission, such as writing a custom GPS message processing script for our ChipSats (stamp-sized satellites installed on our light sail). Overall, I find it rewarding to work on a mission that will actually go into space and solve real problems independently, with solid verification using numerical methods and hardware tests.