Design of a Propulsion Control System for the CaliPER Cubesat for NASA JPL's Europa Clipper Mission

A senior design project report submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical Engineering at Harvard University

Kian Abbott

S.B. Degree Candidate in Electrical Engineering

Advisor: Adrian Arteaga Garcia

Thesis Reader: Dr. Frank Keutsch

Harvard John A. Paulson School of Engineering and Applied Sciences and Engineering
Allston, MA
November 27, 2023

Table of Contents

Acknowledgements

This project has been a complex undertaking that would not have been possible without a lot of support and a lot of guidance.

Thank you to the 2022-23 CaliPER team: Grace, David, Jonathan, Jeff, Jack, Ja(x)son, Will, Kaylee, Adam, and Richard for welcoming me into an incredible section at the apex of your projects, and for leaving me prepared to follow in your footsteps. Thank you to the 2021-22 team- Jacob, Jordan, Jonathan, Matthew, Spencer, and Obinna- the efforts of whom made this project possible in the first place.

Thank you to my out of sequence peers Corrine and Ben alongside whom I've had the privilege of experiencing this slightly odd journey.

For an endless supply of free components and tireless support of capstones and beyond, I want to thank Leo, James, Champa, and the full Active Learning Lab Staff team.

I'd like to express my gratitude to anyone who offered the time to look over my work and provide feedback and mentorship- to Salma, Frank, Katia, Jason, and all the ES100 staff. It must be quite complex to keep track of such a wide array of projects. The care and personalized support that underlies each one certainly did not go unnoticed.

Huge thanks to my section leaders- Ben, Elaine, and Seymur not only for your invaluable insights, but for keeping up with my project, and with me throughout its development.

To Adrian, thank you so much for offering your time to continue the collaboration between JPL and ES100. In providing me with all the resources, wisdom, and support I could ask for you have been an invaluable mentor.

And finally thank you and best of luck to my peers currently steeped in their own work- Lachlain, Matt, Michael, Maycee, Bernard, the 2023-24 JPL team, and all the other 2024 engineers. I can't wait to see what incredible feats you all will accomplish.

Abstract

In 2024 NASA JPL plans to launch the Europa Clipper spacecraft equipped with the REASON radar with the aim of measuring features of Europa. A proposed solution to aid in the collection of this data is to launch an auxiliary cube satellite dubbed CaliPER, for calibrating reason en route to Europa. This project continues the work of previous Harvard engineering students in the design and simulation of a control system to guide the cubesat through calibration maneuvers utilizing an onboard guidance navigation and control (GNC) sensor suite, reaction wheel set, and warm gas propulsion system. Simulink simulation confirms the control system's ability to navigate the spacecraft through calibration maneuvers and dump momentum accumulated from environmental torques through the propulsion system.

List of Figures

List of Tables

1 Introduction

1.1 Background and Motivations

The presence of liquid water on a planet is a key requisite to the development of organic life. Accordingly, researchers exploring the development of life in the solar system often look to instances of liquid water on other planets not only as a potential source for extraterrestrial life but additionally as a means to study mechanisms that could have contributed to the development of life on Earth. Previously studied features of Europa including fissures on the moon's icy surface, magnetic field perturbations, and detection of water vapor in its atmosphere have led researchers to believe that there might be a considerably sized ocean beneath the frozen exterior [1], highlighting the moon as an important destination for further exploration.

In 2024, NASA plans to launch the Europa Clipper spacecraft to analyze the topology and makeup of the icy surface of Jupiter's moon as well as search for a liquid ocean beneath. To the end of taking measurements penetrating the moon's surface, the spacecraft will be equipped with the Radar for Europa Assessment and Sounding: Ocean to Near Surface (REASON). To mitigate interference and increase the accuracy of readings, the radar requires calibration with a ground station. In its deployed state Clipper spans over 30m. JPL lacks a test facility large enough to perform calibration on earth. Additionally, an antenna of REASON's size will sag in earth's gravitational field, skewing any calibration results. Performing calibration after launch poses issues as well. Orienting Clipper to perform this calibration would significantly complicate its maneuvering en route to Europa as well as pose a risk to thermally sensitive components being exposed to the sun. The calibration then will be aided by the Calibration Post Earth for Reason (CaliPER) cube satellite (cubesat) launched alongside Clipper relaying signals to the ground station which characterize REASON's beam pattern.

An effective guidance, navigation, and control (GNC) system is critical to CaliPER's role in this exploration. In order to carry out calibration, CaliPER will have to follow the trajectory of Clipper at a safe distance prior to approaching it to perform calibration maneuvers. These tasks require awareness of the position and orientation of the cubesat relative to Clipper and the Earth, as well as the ability to make precise alterations to these relative positions and orientations within a specified window of Clipper's travel time.

1.2 Project Goals

This project outlines the development and verification, through simulation and physical testing, of a stable and robust propulsion control system to perform detumbling and guide the cubesat during the various phases of its mission. The control system is additionally responsible for maintaining appropriate propellant temperature to properly engage the propulsion system during burns. In executing these tasks, the propulsion control system interfaces with multiple of CaliPER's other subsystems, namely the primary GNC system, propulsion system, and the electrical power system. The primary GNC system is responsible for fine orientation of the cubesat through engaging reaction wheels. This system contains an inertial measurement unit (IMU) and star tracker for measuring the orientation of the cubesat in space. Information from these sensors is sent to the propulsion control system to inform guidance of the cubesat alongside

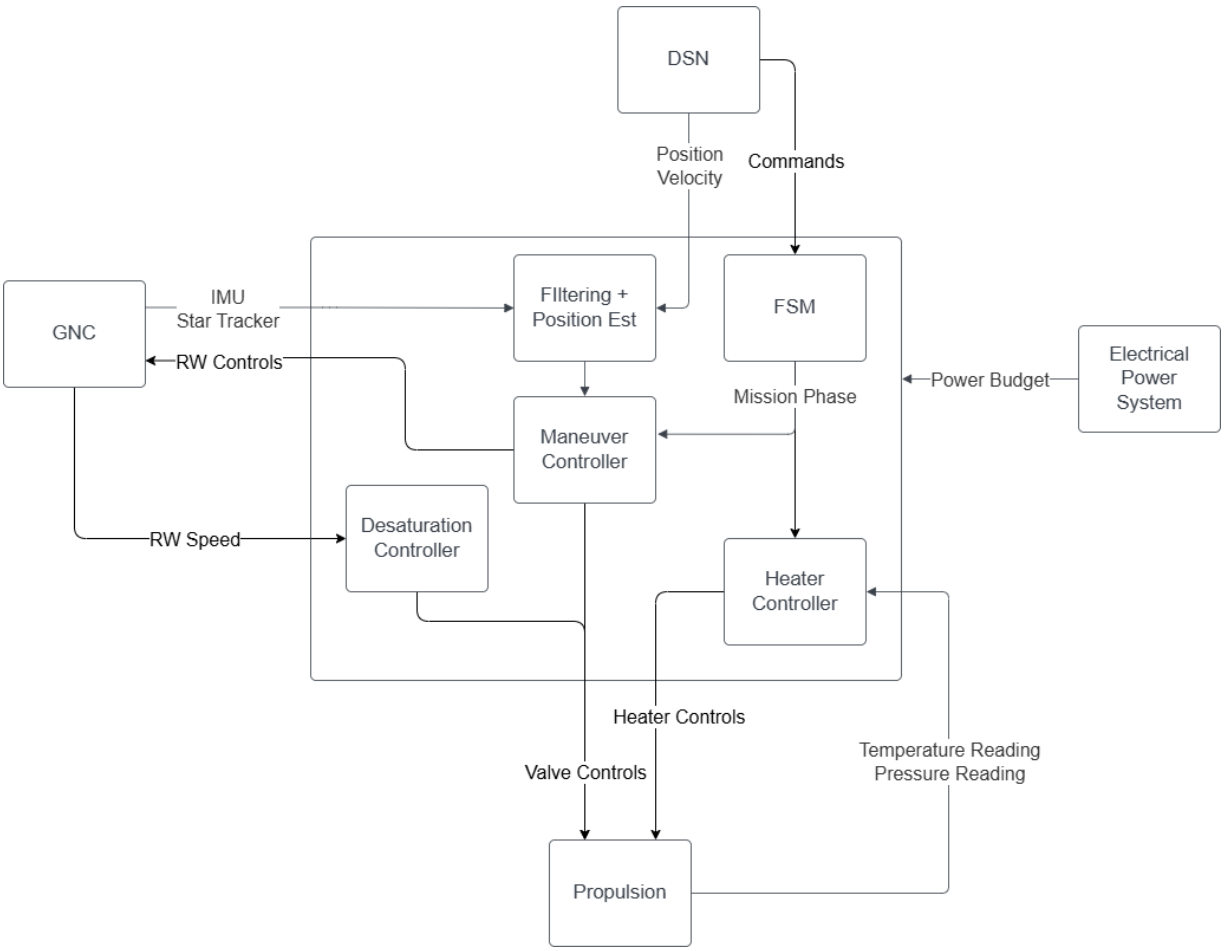

Figure 1: Propulsion Control System Block Diagram. Propulsion control system is boxed alongside related systems.

the speed of reaction wheels to inform desaturation. Guidance control is additionally informed by position and velocity information, and commands driving mission phase issued by the earth based Deep Space Network (DSN). The controller dictates thrust imparted to the cubesat by issuing commands to the propellant heater and solenoid valves contained in the propulsion system. The state of the propulsion system is in turn monitored with readings from temperature and pressure sensors housed in the propulsion system. Throughout the course of the mission, commands issued to the propulsion system must abide by the power budget set forth by the electrical power system.

1.3 Prior Work on CaliPER

Extensive research, calculation, and planning regarding requirements for a 12U version of CaliPER was undertaken by a prior group of Harvard engineering students. With regards to the GNC and propulsion control systems, this previous work generated and verified through calculations a four phase mission for CaliPER, a planned trajectory for the mission, upper and lower bounds on the distance between CaliPER and clipper over the duration of CaliPER's mission, a cross path for REASON calibration, a burn pattern for this path, a necessary pointing accuracy between CaliPER and clipper, maximum slew rate and maximum stabilization time for the GNC system, and minimum torque and ΔV required of the GNC and propulsion systems.

Further work by another group of Harvard engineering students has expanded and updated many of these requirements for a 6U CaliPER, as well as prototyped key subsystems under these new requirements. Relevant requirements to CaliPER's guidance are summarized in Table 1 [2], [3]. Additionally a connected cross with burns as indicated by the arrows in Figure 2 was finalized as the calibration trajectory for REASON.

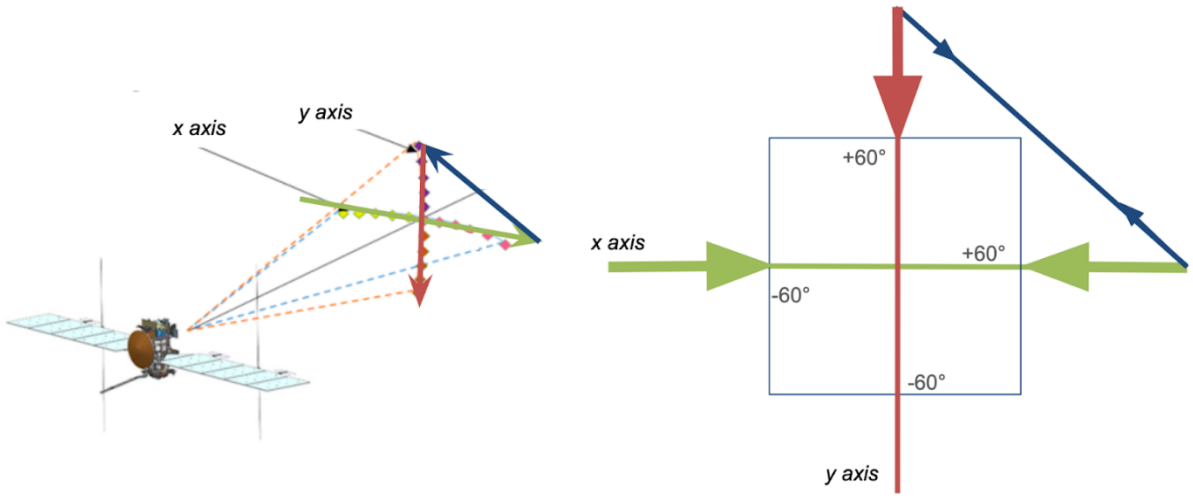

Figure 2: Connected Cross Calibration Pattern. (left) View of the calibration pattern including Clipper's orientation. (right) View of the calibration pattern seen by Clipper looking in the direction of the nadir. The box indicates the 60° calibration zone.

Designation	Definition
GS1	Spacecraft shall characterize the REASON beam pattern in the along and across track direction ± 60° from the nadir point.
GS2	Spacecraft shall characterize the REASON beam pattern to a resolution of ± 0.1° degrees in the along and across track directions.
GS3	Spacecraft shall never come within 75 km of Clipper.
GS4	Spacecraft shall stay within 20,000 km of Clipper during Mission Operations.
G1	The spacecraft shall point at Europa Clipper within ± 0.1° at all times during phase 3 of the mission.
G2	The GNC system shall have a slew rate no faster than 10°/sec.
G3	The GNC system shall produce more than 3.6 μNm of torque.
G4	The GNC subsystem shall desaturate its reaction wheels before they reach 75% of their momentum capacity during all phases of the mission.

G5	The spacecraft shall have 120 ± 10 seconds to re-stabilize after a destabilizing occurrence.
GNC1	The reaction wheel shall have a storage capacity of at least 0.1Nms.
P1	Each calibration session shall be completed within 12 hours.
P2	The propulsion system shall have the capacity to produce a ΔV of at least 60.72m/s.

Table 1: GNC and Propulsion Control System Relevant Requirements

With regard to the GNC system, further analysis on options by which to achieve appropriate attitude control, slew rate, and stabilization led to the selection of sun sensors, a star tracker, and an IMU as sensor inputs for the system, and reaction wheels as actuators. The team fabricated and tested a sun sensor and reaction wheel, tested a star tracker and IMU, and tested a control system for a 1-dimensional reaction wheel setup. With regard to the propulsion system, the team further specified CaliPER's trajectory by dividing the period during which it chases Clipper into phases, as well as updated the calibration trajectory. Through further analysis the team selected warm gas propulsion as the optimal method for the propulsion system, and generated a physical model of the valve, manifold, and nozzle placements.

1.4 Additional Project Considerations

JPL operates under NASA. While NASA's primary mission is the extension of human knowledge with a focus on space exploration, it is a United States Agency. While much of their work is published and freely posted there is a natural preferential access, particularly regarding cutting edge discoveries, given to other groups based in the United States. Although less pronounced since the time of the Cold War, the concept of space as a final frontier through which nations may exert dominance and showcase prowess still leaves its exploration as a politicized topic. Expanding collaborations between groups like JPL and institutions like Harvard and Caltech can in part serve to concentrate knowledge in the U.S. if careful consideration is not given to the distribution of research and resources globally.

In a similar vein, satellites and the expensive measurement tools they carry are multimillion dollar or even billion dollar investments for countries that in many circumstances do not return monetary benefit. While the end of expanding human knowledge is valuable in itself, the utility of a spacecraft must be weighed against the myriad of other uses for these large sums. With a $500 budget, this project represents a far more constrained development case. This holds the benefit of encouraging solutions that showcase how scientific research can be at least somewhat accessible to private individuals in a wider range of circumstances. Regardless of budget size however, the project must still aim to maximize the knowledge return on investment, bearing in mind resources diverted from other causes.

Since JPL was transferred to NASA in 1958 their work has been dedicated to missions that advance scientific knowledge of space. Among the responsibilities of CaliPER's design is to

continue in the tradition of advancing this understanding. Despite this, an array of aerospace technologies also has defense applications, as showcased in JPL's association with the US Army in the decades prior to the transfer. Successful designs are often retained and repurposed within industries. Trajectory determination and control is a problem relevant both to spacecrafts and long-range weaponry, so when designing solutions for it the ease with which the system could be abused or repurposed should be considered. The propulsion control system will be developed specifically for use in space and its design will be guided by the physical properties of CaliPER and its propulsion system. While any developed control algorithm could be repurposed for guidance of another device, the development of this control system would do little to further the design of systems that do not align with the constraints of a cubesat.

The collaboration between JPL and Harvard students on this project affords an opportunity to inject industry insight into the college team driven world of cubesat development. Prior to recent missions like MarCO, design of cubesats had largely been the domain of university teams like Students for the Exploration and Development of Space. The result of this was the creation of a largely open source but sometimes resource or expertise limited body of work outlining emerging best practices and design standards for cubesats. Increased interest in cubesats as secondary or even primary spacecrafts for missions has increased industry resources devoted to developing them. This project aims to retain the open source nature of cubesat development and utilize the unique collaboration to uplift the university development community.

2 Background Research

2.1 Cubesat Guidance

2.1.1 Modeling the Cubesat

The movement of a cubesat in space can be well modelled as a rigid body, a body where particles within the body maintain a fixed distance from one another including under the application of external forces and torques. CaliPER will experience several forces and torques throughout its mission including external solar radiation, aerodynamic, magnetic field, and gravitational gradient torques, and internal solar panel, reflect array, and antenna deployment torques. The rigid body model disregards any deformation of the cubesat. Forces experienced throughout the duration of the mission should remain below the 777N benchmark applied to CaliPER's bus during load testing. This test saw a maximum deformation of under 3.5mm in all axes, safely allowing deformation to be disregarded in modeling the cubesat's dynamics [4].

The translational movement of a rigid body with mass m is described by Newton's Second Law

$$\Sigma\vec{F} = m\dot{\vec{v}}$$

Equation 1: Newton's Second Law

where $\vec{F}$ is the vector describing the net force externally applied to the body and $\dot{\vec{v}}$ is the vector describing the time derivative of the velocity of the body's center of mass. The dependence of a body's inertia tensor on its orientation in the reference frame leads, in general, to a time varying inertia tensor as the body rotates in the inertial frame. As a result, calculations involving the inertia tensor are performed in the body frame where axes are fixed to the principal axes of the body. The resultant equation describing rotational motion in the body frame is

$$\Sigma\vec{\tau} - \vec{\omega} \times I\vec{\omega} = I\dot{\vec{\omega}}$$

Equation 2: Rotational Analog of Newton's Second Law in the Body Frame

where $\vec{\tau}$ is the net torque applied to the body, I is the inertia tensor describing the distribution of mass within the rigid body, and $\vec{\omega}$ is the angular velocity of the rigid body with all quantities reported in the body frame. It is useful in framing the control of the cubesat to divide the torque applied to the body into externally, and internally applied portions where the externally applied portion arises from the beforementioned environmental torques and the internally applied portion arises from the angular acceleration of the reaction wheels as directed by the controller. When an internal torque is applied, the body adjusts its motion to maintain its angular momentum, so this portion is negated. The equation of motion becomes

$$\Sigma\vec{\tau}_{ext} - \dot{\vec{\rho}} - \vec{\omega} \times (I\vec{\omega} + \vec{\rho}) = I\dot{\vec{\omega}}$$

Equation 3: Body Frame Rotational Equation of Motion with Reaction Wheels

where $\vec{\tau}_{ext}$ is the environmental torque, $\vec{\rho}$ is the reaction wheel angular momentum and $\dot{\vec{\rho}}$ is the reaction wheel generated torque.

The orientation of the body frame with respect to the inertial frame can be described with three Euler angles ψ, θ, ϕ where the transformation from the inertial frame to the body frame is achieved by rotating the inertial frame's axes by these angular quantities around the Z, Y, and X axes respectively. This project uses implicit rotations where θ represents rotation about the new Y axis, Y_1 after rotating ψ about Z, and ϕ represents rotation around the new X axis, X_2 after rotating about Z and Y_1. This sequence of rotations and use of intermediate frames 1 and 2 aligns with the standard avionics yaw, pitch, and roll. This coordinate transformation can be represented by three applications of a 3d rotation matrix. This composition is described by

$$T_{eb} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & C\Phi & S\Phi \\ 0 & -S\Phi & C\Phi \end{pmatrix} \begin{pmatrix} C\Theta & 0 & -S\Theta \\ 0 & 1 & 0 \\ S\Theta & 0 & C\Theta \end{pmatrix} \begin{pmatrix} C\Psi & S\Psi & 0 \\ -S\Psi & C\Psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Equation 4: Inertial Frame to Body Frame Transformation [5]

where Cx and Sx represent cos(x) and sin(x). Its inverse transformation, which transforms the body frame back to the inertial frame, is given by the following composition.

$$T_{be} = \begin{pmatrix} -C\Psi & -S\Psi & 0 \\ S\Psi & -C\Psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -C\Theta & 0 & -S\Theta \\ 0 & 1 & 0 \\ -S\Theta & 0 & -C\Theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -C\Phi & -S\Phi \\ 0 & S\Phi & -C\Phi \end{pmatrix}$$

Equation 5: Body Frame to Inertial Frame Transformation [5]

Attitude of a body can also be expressed as a quaternion in the form

$$q = \cos(\theta/2) + \sin(\theta/2)(ir_1 + jr_2 + kr_3)$$

Equation 6: Attitude Quaternion Form

where i, j, and k are the imaginary units, ($r_1 \quad r_2 \quad r_3$) is a normalized axis of rotation in the inertial frame, and θ represents the amount by which to rotate around ($r_1 \quad r_2 \quad r_3$) to reach the body frame attitude. This is a special case of a general quaternion $q = q_0 + q_1 i + q_2 j + q_3 k$. Attitude quaternions satisfy the property $\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$.

The body's angular velocity can be expressed in two ways. $\vec{\omega}$, as used in Equation 2, is the body frame angular velocity vector. This vector points in the direction of the axis of rotation as given by the right hand rule with its magnitude corresponding to the speed of rotation. By Euler's rotation theorem, another valid representation of the angular velocity is in terms of the rates of change of the Euler angles. This is the format in which body angular velocity is most often reported by sensors, including the GNC selected BNO055 IMU. The rates of change are measured in a sequence of rotations, analogously to the Euler angles themselves. $\dot{\psi}$ is measured in a frame where the body Z axis aligns with the inertial Z axis (inertial frame or frame 1), $\dot{\theta}$ is measured in a frame 2, and $\dot{\phi}$ is measured in the body frame. The transformation between the

measured Euler angle derivates and body frame angular velocity vector sequentially undoes these rotations. This transformation is

$$\vec{\omega} = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & C\phi & S\phi \\ 0 & -S\phi & C\phi \end{pmatrix} \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & C\phi & S\phi \\ 0 & -S\phi & C\phi \end{pmatrix} \begin{pmatrix} C\theta & 0 & -S\theta \\ 0 & 1 & 0 \\ S\theta & 0 & C\theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & -S\theta \\ 0 & C\phi & S\phi C\theta \\ 0 & -S\phi & C\phi C\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

Equation 7: Euler Angle Derivative to Body Frame Angular Velocity Transformation [5]

and the corresponding inverse transformation is

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & S\phi T\theta & C\phi T\theta \\ 0 & C\phi & -S\phi \\ 0 & S\phi/C\theta & C\phi/C\theta \end{pmatrix} \vec{\omega}$$

Equation 8: Body Frame Angular Velocity to Euler Angle Derivative Transformation [5]

where Tx signifies $\tan(x)$.

The analogous transformation to Equation 8 that relates the derivative of a quaternion attitude to $\vec{\omega}$ is

$$\begin{pmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}$$

Equation 9: Body Frame Angular Velocity to Quaternion Derivative Transformation

where $q_0 = \cos(\theta/2)$, and $(q_1 \quad q_2 \quad q_3) = \sin(\theta/2)\,\vec{r}$ as in Equation 6.

2.1.2 Modeling Thrust

External torques will be applied to CaliPER in the form of environmental torques and deployment torque. CaliPER's angular momentum can also be altered through engagement of the reaction wheels in the GNC subsystem, which will be responsible for counteracting these torques. The final source or torque and primary quantity of interest in the propulsion control system is the thrust generated by the propulsion system. This thrust will additionally serve as the primary external linear force acting on the cubesat. The instantaneous thrust produced by one thruster is given by

$$F_t = \dot{m}_p v_{eff}$$

where $\dot{m}_p$ is the propellant mass flow rate and v_{eff} is the propellant's effective exit velocity, which is a function primarily of nozzle geometry and environmental conditions. The effective exit velocity is given by

$$v_{eff} = v_e + \frac{(P_E - P_O)A_e}{\dot{m}_p}$$

where v_e is the propellant exit velocity, A_e is the exit area of the nozzle, and P_E and P_O are the exit and ambient pressures. Thrust is varied by actuating solenoid valves in the propulsion system which restrict the mass flow rate. In full stroke solenoid valves controlled with pulse width modulation (PWM), thrust can be linearly related to the PWM signal's duty cycle [6]. This allows the thrust–duty cycle relationship to be approximated with as little as a reading of thrust with the valve fully open and a reading at an intermediate duty cycle. Torque generated by a given nozzle is given by the cross product between the thrust vector with direction exiting the nozzle and its position vector relative to CaliPER's center of mass.

2.2 Propulsion and Control Schemes

2.2.1 Orbit Navigation Autonomy

Cubesat navigation in deep space poses unique challenges from earth orbiting missions, with latency associated with ground communication being foremost among them. Navigation methods have been developed with varying degrees of reliance on onboard sensors and earth-based communication. Navigation methods can be broadly categorized into fully-autonomous, semi-autonomous, and non-autonomous schemes. The sensors used to perform navigation are a fundamental aspect for the categorization of a method.

The GNC sensors which underpin the propulsion control system—IMU, star tracker, and sun sensor—are typically utilized in fully-autonomous control systems where position and attitude determination, and guidance computation are performed on board the satellite. While these systems can achieve moderate accuracy and feature a more quickly operating control loop, the overall accuracy of the system is heavily linked to the quality of sensor placed on the satellite, which can lead to prohibitively high materials cost. For the sensors selected for CaliPER, a fully autonomous control system likely would not produce the necessary accuracy for REASON calibration. A particular point of difficulty is drift of the BNO055 IMU over time when reporting orientation, as well as inaccuracies at the extremes of the angular position range, especially in the 88°-90° rotation range. This inaccuracy can be buffered with repeated recalibration utilizing the orientation generated by the star tracker; however, this relies on the consistent ability to resolve star tracker images. Resolving images poses some difficulty in itself because of the image resolution constraint arising from CaliPER's constrained memory resources.

To compensate for issues in maintaining accurate attitude, the control system will also include a ground based radiometric element, the primary communication type used in non-autonomous systems. This communication will originate with the DSN and communicate

CaliPER's position and velocity. Despite the delay introduced by ground station communication, and the limited window of DSN communication availability, these transmissions will serve as valuable calibration points to combat the inaccuracies of onboard sensors.

2.3 Prior Cubesat Missions

2.3.1 MarCO

In the interest of better understanding common implementations of propulsion controllers as well as exploring the possibility of relying on parts with flight heritage, a study of cubesat missions with propulsion was undertaken. One such mission was Mars Cube One (MarCO). The mission involved a pair of cubesats which, similarly to CaliPER, acted as a communication relay between the main measurement vehicle, the Interior Exploration using Seismic Investigations, Geodesy and Heat Transport (InSIGHT) Lander, and an earth ground station [7]. The mission utilized a micro propulsion system (MiPS) developed by VACCO which was controlled by the XACT50 attitude determination and control system developed by Blue Canyon Technologies [8]. The system received inputs from a gyroscope, star tracker, and sun sensors [9]. Although usually controlled directly by the XACT50, the propulsion system could also communicate with the command data and handling (CD&H) system microcontroller, which interfaced with all spacecraft subsystems. The CD&H system was equipped with a Texas Instruments MSP430F2618 flight CPU driven by JPL written embedded flight software [10]. The system controlled the burn type and duration for 8 thrusters using closed loop, variable control. It also controlled three thermal control zones at the main propellant tank, prior to the main control valve, and at the plenum before the thruster control valves (Figure 3).

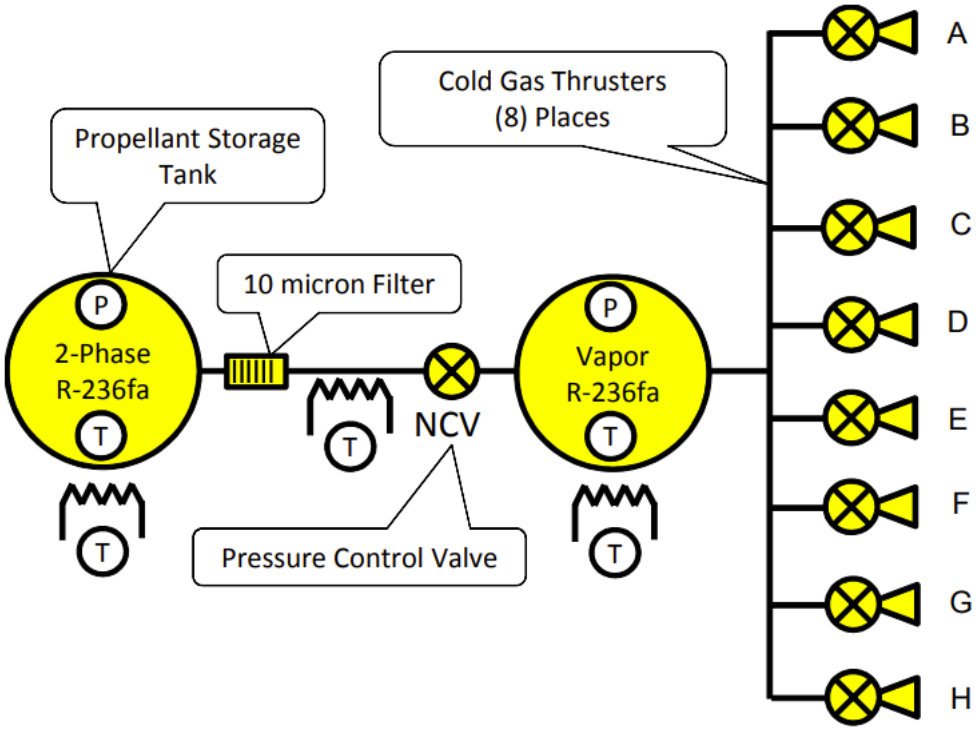

Figure 3:MarCO Propulsion Controller Preliminary Schematic

MarCO's propulsion system was responsible for desaturating reaction wheels and performing trajectory correction maneuvers (TCMs), with 4 thrusters tilted to provide angular control and 4 for axial thrust. Reaction wheel desaturation was performed automatically utilizing the onboard control system. Commands for performing trajectory control maneuvers were sent from the ground station to the CD&H subsystem to be translated into maneuvers utilized by the propulsion system. The basic structure of the control loop employed by XACT units for attitude control is outlined in Figure 4, although some modification beyond this skeleton was needed to accommodate propulsion system desaturation. One TCM was planned for each of the cubesats to correct velocity imparted during deployment from the launch vehicle with a ΔV allowing for four additional maneuvers over the course of the mission. Each TCM was divided into multiple thruster firings with each firing lasting for a duration on the order of 10s to 100s of seconds, and firings spaced out over multiple days [11]. This division of maneuvers allowed time for the cubesats to communicate with ground station in between incremental ΔV adjustments, reducing position and velocity error in between communications.

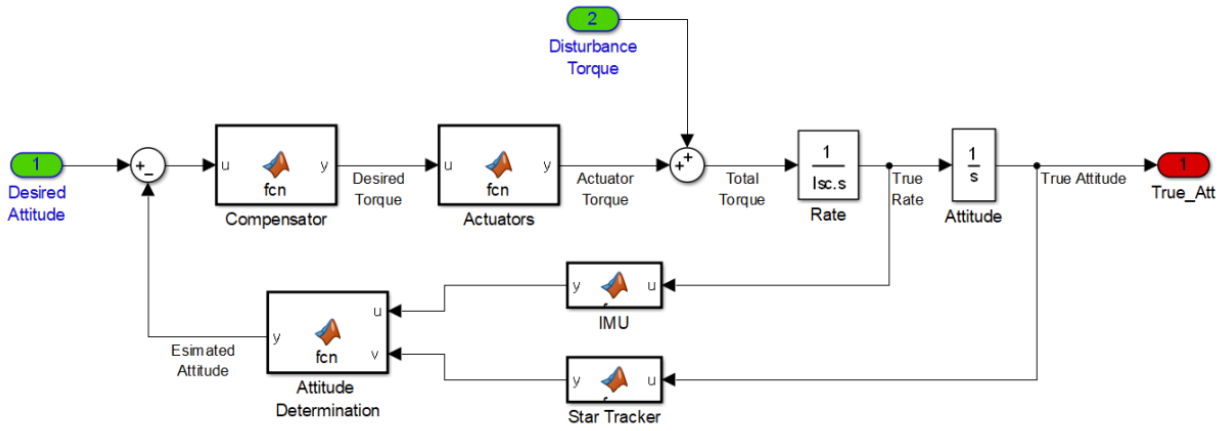

Figure 4: XACT Basic Control Loop

2.3.2 Lunar Flashlight

The Lunar Flashlight was another cubesat mission which utilized a propulsion system. Its mission was to map features on the permanently shadowed side of the moon and determine the presence of water ice in the explored area [12]. The propulsion control system was set up similarly to MarCO, again utilizing an XACT-50 for thruster control; however, the CD&H system could communicate with the thrusters only through the XACT unit in this mission. Integration testing of the XACT-50 and propulsion system offered further insight on the information passed between them. Among tested commands were commands to dump reaction wheel momentum, engage and disengage thruster attitude control, set the time over which burns would ramp up, and to send 10-bit commands directly to the thrusters [13].

Analysis of a similar design employing the XACT-100 GNC by current ES100 students revealed that the unit was too expensive to be a viable option. In a similar way, despite the existence of commercial solutions, development of a propulsion control system will be driven by the prohibitive cost of these solutions.

3 Design Goals and Independent Technical Specifications

The most critical metrics of success for the propulsion control system will be the ability to generate torques to appropriately desaturate reaction wheels, and to generate torques and forces necessary to guide the spacecraft along its planned trajectory. The functioning of the propulsion control system should not obstruct the ability of the main GNC controller to maintain the spacecraft's attitude. Critical design independent technical specification are outlined in Table 2.

Designator	Spec	Value	Measurement
PC1	Minimum Distance from Clipper	75km	Simulink Simulation
PC2	Maximum proportion of RW momentum capacity	75%	Simulink Simulation, 1D Desaturation Test
PC3	Peak power draw of valve driver circuitry	1.5W	Physical circuit build measuring operating voltage and current
PC4	Peak power draw of heater circuitry	30W	Physical circuit build measuring operating voltage and current
PC5	Sensing and valve driver circuitry operating temperature range	-20℃ to 60℃	Verified through component datasheet review

Table 2: Design Independent Technical Specifications

PC1 was inherited from the work of the propulsion system design team. It ensures a safe distance between CaliPER and Clipper while allowing for a 12 hour pass through the calibration zone with the designed ΔV budget. PC2 was inherited from the work of the 2022 JPL team. PC3 represents a slight increase from 1.25W allotted for burns in the estimated power budget produced by the propulsion team. This power budget is derived from the 250mW holding power consumption of the selected solenoid valves. Any given rotation or translation requires a maximum of 4 thrusters. 1.25W then accounts for the holding power of 4 valves with margin built in for usage of one additional valve. Outside of the valve's holding power, the primary powered component in the valve driver circuitry is the NE555 timer with a maximum quiescent current of 6mA [14]. If the part takes its supply voltage from the 12V rail used for the solenoid this represents a power consumption of 72mW. Leaving margin for the remaining components in the driver circuitry led to the overall increase to 1.5W. PC4 was also inherited from the estimated power budget. PC5 represents a very common operation range for components rated for space hardware.

4 Control System Design Approach

4.1 Control System Components

4.1.1 Controller Finite State Machine

The propulsion control system will be responsible for a variety of tasks throughout the course of CaliPER's mission. A broad division of the mission is given in Figure 5. During Phase 1 CaliPER detaches from its launch vehicle, presumed to be a SpaceX Falcon Heavy, and the GNC system activates to stabilize the cubesat's orientation. During this phase the propulsion control system can only be activated for desaturation. During Phase 2 the propulsion system is activated to produce a velocity such that CaliPER begins to catch up to Clipper, which will have been released before it. During Phase 3 CaliPER's position and velocity must be controlled to move through the cross calibration trajectory. During Phases 2 and 3 the propulsion system can also be activated for desaturation.

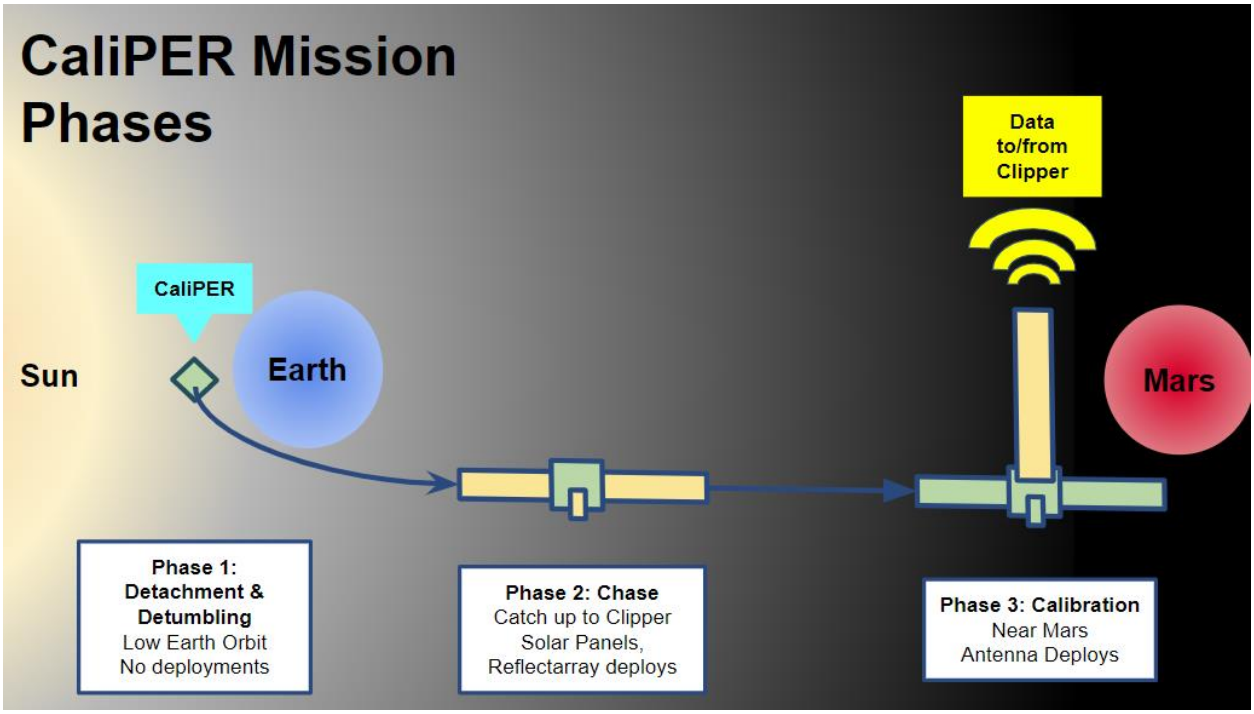

Figure 5: CaliPER Mission Phases

A finite state machine (FSM) will be responsible for tracking CaliPER's mission phase and activating the appropriate propulsion control system functionality for that phase. A generalized view of the FSM is outlined in Figure 6. The default state of the FSM on reset is the drift state, in which all 8 valves and the main control valve are closed, and the heater is off. In this state the propulsion system is inactive. Each of the remaining states represents one of the primary functionalities of the propulsion control system. Each of these states activates one of the controllers shown in Figure 1, with the desaturation state activating the desaturation controller, the preheat state the heater controller, and the position/velocity state the maneuvering controller.

Transitions out of the drift state will be activated either by reaching a predetermined point in the mission phase or, in the case of the desaturation state, by a reaction wheel velocity reported by the GNC system to be above a threshold $\overrightarrow{\omega_{th}}$. Return to the drift state will occur once the estimated value that a controller sets matches the target value. The set values are the body rotation rate $\vec{\omega}$ in the case of the desaturation controller, CaliPER's position or velocity vector in the Clipper frame $\vec{x}, \vec{v}$ in the case of the maneuvering controller, and the propellant chamber temperature T in the case of the heater controller. Target and controller estimated values are denoted by a subscript "t" and "e" respectively. The FSM will store target position or velocity values for the maneuvering controller that correspond with each mission phase. The desaturation and temperature controllers have preset target values of $\overrightarrow{\omega_t} = \vec{0}$rad/s and $T_t = 30°C$ respectively. A detailed breakdown including the timing of mission phases and specific position and velocity set points is given in section 10.1. The final possible transitions handle a propellant preheat failure. Should an unsafe propellant pressure threshold be reached, the system will stop preheat and begin the scheduled maneuvering burn.

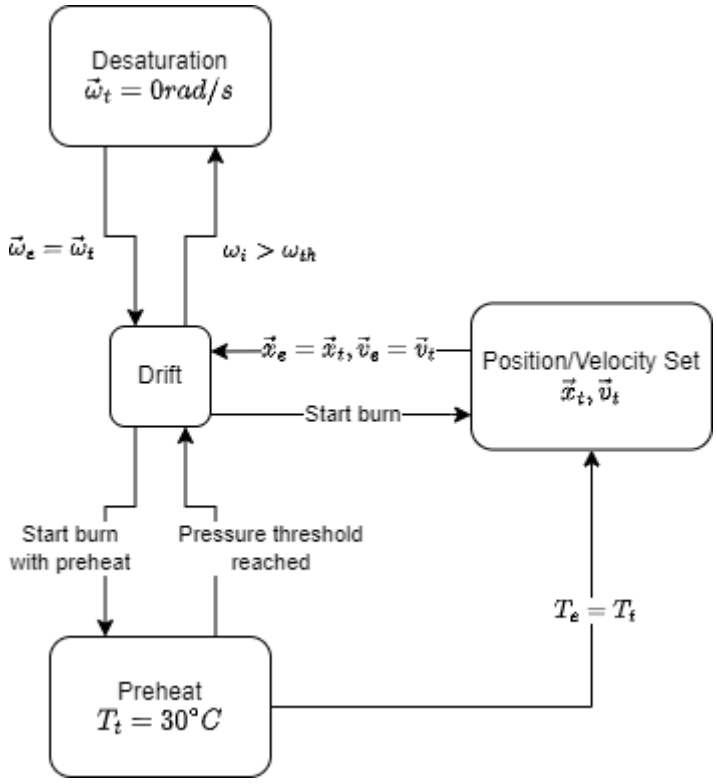

Figure 6: Propulsion Control System Finite State Machine

4.1.2 Maneuvering Controller

The maneuvering controller is responsible for controlling CaliPER's position and velocity relative to Clipper. Guidance of the cubesat to different positions and along different velocities relies on coordination between the thrusters for force generation and reaction wheels for

repointing. As shown in Figure 7, propulsion nozzles 2, 3, 6, and 7 are placed symmetrically about CaliPER's geometric center in the X-Y plane. Assuming the spacecraft's center of mass lies approximately at its geometric center as stipulated of its undeployed configuration by McMullin's requirement C1 [4], and assuming minimal thrust imbalance between nozzles, activation of these thrusters will produce Z axial motion without generating a net external torque. A common control scheme utilizes reaction wheels to repoint the spacecraft in the desired direction of motion and initiate the motion by activating this Z axial thruster combination. This controller can be further refined by adjusting the duty cycle of thrusters 2 and 3 relative to 6 and 7 to account for center of mass offset in the X direction and of thrusters 2 and 7 relative to 3 and 6 to account for center of mass offset in the Y direction.

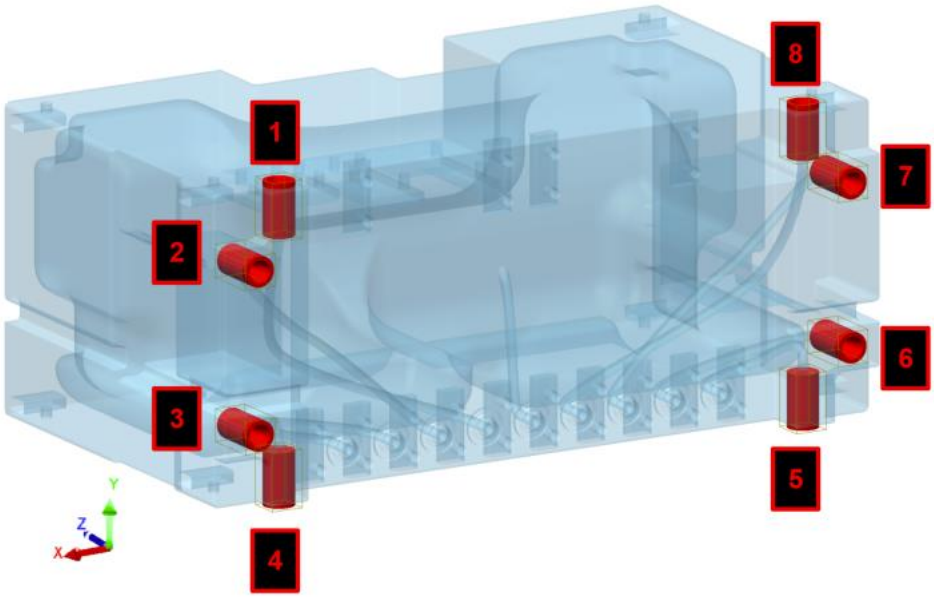

Figure 7: Thruster Nozzle Locations within the Propulsion System

4.1.2.1 Maneuvering Controller: Slewing

Maneuvering is broken into two segments. The first is a repointing segment during which only the reaction wheels are active, and the spacecraft is reoriented to the desired direction of travel. The second is a travel segment during which the thrusters are activated to produce the required linear thrust force. The control loop for the repointing segment is shown in Figure 8.

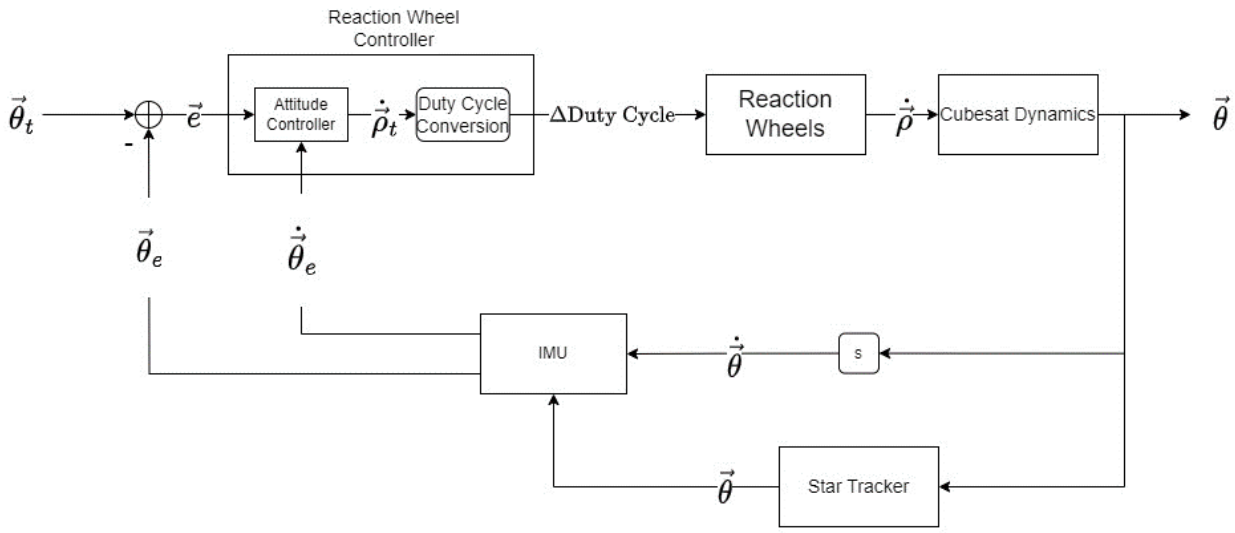

Figure 8: Repointing Control Loop

The control loop takes an input attitude $\vec{\theta}_t$ which is derived from the targeted velocity vector or the direction vector between CaliPER's current position and its targeted position in the Clipper frame. An error is calculated between this target and the current estimated attitude $\vec{\theta}_e$ which is used by the attitude controller to generate a target reaction wheel induced torque $\dot{\vec{\rho}}_t$. This is converted to an appropriate change in duty cycle based on the reaction wheel geometry. The reaction wheel output then influences CaliPER's attitude which is read by the IMU's gyroscope as a new set of Euler angle derivatives. The IMU's sensor fusion algorithm filters and integrates the gyroscope readings to produce a new estimated attitude. $\vec{\theta}$ is used to represent attitude, although the control loop can be enacted using Euler angles or quaternion attitude.

The torque induced by reaction wheels is determined in part by their orientation in the body of a spacecraft. While only three reaction wheels are needed to control attitude, spacecrafts often include a fourth for redundancy in the event of a failure, and to aid in compensation for imbalances between wheels. The most common configurations are the pyramidal configuration and the NASA standard configuration. CaliPER's design utilizes a pyramidal configuration canted up towards the Y-axis. The reaction wheels are assumed to be placed with their centers in the X-Z plane along either the $x = z$ line or $x = -z$. Their spin axes point either directly to or away from the Y axis. This configuration is shown in Figure 9. The ith reaction wheel is located in the ith quadrant of the X-Z plane.

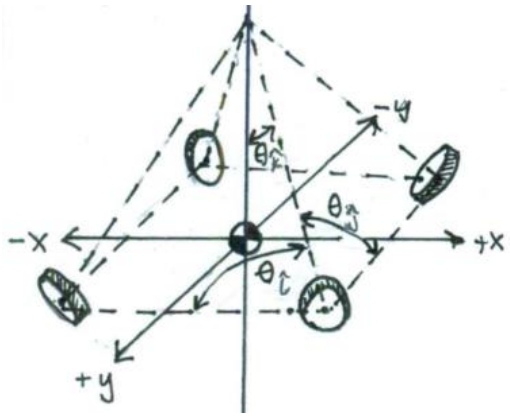

Figure 9: Pyramidal Reaction Wheel Configuration [2]

The torque generated by RWs in a given configuration can be determined by multiplying a vector containing each of the RW torques by a matrix A_{RW} whose columns are normalized direction vectors of the RWs' spin axes. In the case of the above pyramidal configuration, this relationship is

$$\dot{\vec{\rho}}_{net} = \begin{pmatrix} -A_{RWx} & -A_{RWx} & A_{RWx} & A_{RWx} \\ A_{RWy} & A_{RWy} & A_{RWy} & A_{RWy} \\ -A_{RWz} & A_{RWz} & A_{RWz} & -A_{RWz} \end{pmatrix} \begin{pmatrix} \dot{\rho}_1 \\ \dot{\rho}_2 \\ \dot{\rho}_3 \\ \dot{\rho}_4 \end{pmatrix}$$

Equation 10: RW Torque Vector to Body Frame Torque Transformation

where the ratio between components is given by $A_{RWx} : A_{RWy} : A_{RWz} = \frac{C_{\theta i}}{C_{\theta k}} : 1 : \frac{C_{\theta j}}{C_{\theta k}}$. Each column of the matrix is normalized and $\dot{\rho}_1$,…, $\dot{\rho}_4$ are the individual scalar torques applied by each RW. When evaluated for $\theta_i = 43°$, $\theta_j = 25°$, $\theta_k = 48°$ as designed by the GNC team, the transformation is as given below.

$$\dot{\vec{\rho}}_{net} = \begin{pmatrix} -0.5445 & -0.5445 & 0.5445 & 0.5445 \\ 0.4892 & 0.4892 & 0.4892 & 0.4892 \\ -0.6748 & 0.6748 & 0.6748 & -0.6748 \end{pmatrix} \begin{pmatrix} \dot{\rho}_1 \\ \dot{\rho}_2 \\ \dot{\rho}_3 \\ \dot{\rho}_4 \end{pmatrix}$$

Equation 11: RW Torque Vector to Body Frame Torque Transformation with CaliPER GNC Selected Cants

The attitude controller is a PID controller which utilizes the error to generate necessary reaction wheel induced torques in the X, Y and Z directions. Four variants of PID control were tested. The first utilized a control law of the form

$$\dot{\vec{\rho}}_t = -(K_p \vec{\theta}_e + K_i \int \vec{\theta}_e dt + K_d \dot{\vec{\theta}}_e)$$

Equation 12: Slewing Control Law 1

where K_p, K_i, and K_d are scalar proportional, integral, and derivative constants respectively. This controller is a three-dimensional implementation of standard PID control apart from the outer

sign negation, which is needed to account for the RWs producing an internal torque as in Equation 3. The second utilized a control law of the form

$$\dot{\vec{\rho}}_t = -(K_p\vec{\theta}_e + K_i\int \vec{\theta}_e dt - K_d\dot{\vec{\theta}})$$

Equation 13: Slewing Control Law 2

where the derivative term is now based on the rate of change of the Euler angles $\dot{\vec{\theta}}$. Note the double negation of the derivative term. This term of the controller no longer induces torque in the direction of error to reduce it, but instead applies a torque in a direction to oppose $\dot{\vec{\theta}}$ to drive it to 0rad/s in all components. The third controller mirrors the structure of the second but instead utilizes quaternion attitudes. It takes the control law

$$\dot{\vec{\rho}}_t = -\left(K_p\begin{pmatrix} q_{e,1} \\ q_{e,2} \\ q_{e,3} \end{pmatrix} + K_i\int \begin{pmatrix} q_{e,1} \\ q_{e,2} \\ q_{e,3} \end{pmatrix} dt - K_d\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}\right)$$

Equation 14: Slewing Control Law 3

which drives the $\sin(\theta/2)\,\vec{r}$ portion of the spacecraft's attitude quaternion to match that of the reference quaternion and drives the rotation rate of the body to 0rad/s in the three spatial directions.

The final controller mirrors the third but makes use of the body angular velocity directly as opposed to utilizing the quaternion derivative to bring the spacecraft to rest. It takes the control law

$$\dot{\vec{\rho}}_t = -\left(K_p\begin{pmatrix} q_{e,1} \\ q_{e,2} \\ q_{e,3} \end{pmatrix} + K_i\int \begin{pmatrix} q_{e,1} \\ q_{e,2} \\ q_{e,3} \end{pmatrix} dt - K_d\,\overrightarrow{\omega_b}\right)$$

Equation 15: Slewing Control Law 4

All controllers use the same duty cycle conversion method to generate a 4 element vector of duty cycle changes

$$\vec{u} = dA_{RW}^{+}\dot{\vec{\rho}}_t$$

Equation 16: Duty Cycle Conversion Output

where A_{RW}^{+} is the pseudoinverse of A_{RW} which reverses the transformation described by Equation 11, and d is a scale factor based on the geometry of the RWs and driving motor characteristics which relates a change in duty cycle over a given period of time to the torque induced by the driven RW.

The final slewing controller design uses slewing control law 4. Testing and comparison among the control laws is described in section 6.3.1.1.

4.1.2.2 Maneuvering Controller: Translation

Once the spacecraft has achieved sufficient pointing accuracy in the direction of travel, the thrusters can be activated to produce linear translation. In this controller, position and velocity vectors $\vec{x}, \vec{v}$ are in the Clipper frame, which during the time of REASON calibration is considered to be inertial. The axes of this frame are as defined in Figure 2 where the z direction is REASON's nadir.

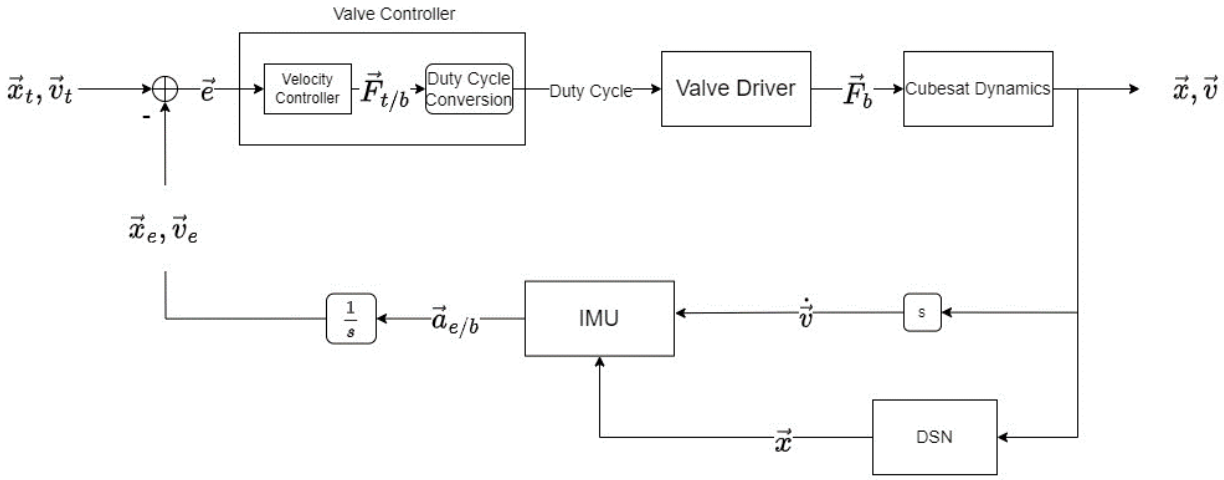

Figure 10: Maneuvering Thrust Control Loop

The input to the control loop is either a target position or velocity which is provided by the FSM. Error $\vec{e}$ between the target and current vectors is calculated and passed to the velocity controller which converts the error to a body frame vector and determines the necessary body frame thrust $\overrightarrow{F_t}$ to maneuver the cubesat to the target. The thrust is converted to a vector of duty cycles which drive the eight solenoid valves. The true force $\vec{F}$ influences CaliPER's position and velocity $\vec{x}, \vec{v}$. Changes in the cubesat's linear state are measured as a body frame acceleration vector by the IMU's accelerometer. The estimated acceleration output is then integrated and converted to the inertial frame externally from the IMU on the compute platform which hosts the controller. To compensate for IMU drift in tracking position and velocity, whenever these values are reported from the DSN the IMU will be recalibrated with the provided values.

Although the control loop inputs only linear velocity targets, the velocity controller also draws on attitude information $\vec{\theta}_e$ to inform conversion between body and inertial frame vectors. Using quaternion attitudes, conversion from a body frame vector to an inertial frame vector is given by the quaternion product

$$\begin{pmatrix} 0 \\ \vec{r} \end{pmatrix} = q \otimes \begin{pmatrix} 0 \\ \vec{r}_b \end{pmatrix} \otimes q^{-1}$$

where q is the current attitude quaternion, $q^{-1} = q_0 - q_1 i - q_2 j - q_3 k$, and the body frame vector is converted into a quaternion by prepending 0 as its real part.

A controller that assumes a balanced center of mass with regards to thruster placement can produce the control vector by observing only the z component of the error with the PID control law

$$z_{balanced} = K_p * error_z + K_i \int error_z dt + K_d error_z$$

Equation 17: Balanced Z-axis Translation Control Law

which can be converted to a thrust vector by multiplying by a vector which selects for the Z-axial thrusters 2, 3, 6, and 7. This controller can be improved to account for an unbalanced center of mass by producing different thrusts in each thruster that balance the torque around the center of mass. The appropriate duty cycles to balance torque around the center of mass can be found by solving the system described by

$$\sum_{i=2,3,6,7} D_i\left(\vec{r}_i \times \vec{F}_{thrust,i}\right) = \vec{0}$$

Equation 18: System Describing Balanced Duty Cycles

Where D_i is the duty cycle of the ith thruster, $\vec{r}_i$ is the location of the ith thruster with respect to the center of mass, and $\vec{F}_{thrust,i}$ is the vector describing the ith thruster's thrust direction. This a system of three equations; however, restricting to the z axial thrusters, 4 duty cycles are needed. Additionally, when restricted to the z-axial thrusters, they cannot produce torque in the z direction, so the equation described in the third component adds no additional constraints. Two additional conditions can be imposed to define the system. The constraints chosen were

$$D_2 = 1 \quad ; \quad D_2 = D_7$$

Equation 19: Additional Constraints Used to Solve for Torque Balanced Duty Cycles

The first constraints selects for positive duty cycles, which are physically necessary. The second constraint provides some symmetry between the upper and lower thrusters; however, it could be replaced if alternative needs arose.

The final improvement that can be made to the controller is the introduction of differential thrust to account for various imbalances like a center of mass offset, thruster misalignment, or thruster imbalance. Center of mass offset in this case refers to a center of mass that is not only not aligned with the geometric center of the spacecraft but is unknown by some margin because of manufacturing tolerance, or a dynamic condition like fuel sloshing during the mission. Thruster misalignment describes deviation of the thrust vectors from their designed direction, and thruster imbalance refers to the difference in thrust produced between two thrusters operating at the same duty cycle. These effects can be mitigated by superimposing a small differential thrust which causes the spacecraft to turn in a direction to counteract disturbances as it accelerates in the primary direction. This can be achieved with the institution of two additional PID controllers

$$x_d = K_p * error_x + K_i \int error_x dt + K_d error_x$$

$$y_d = K_p * error_y + K_i \int error_y dt + K_d error_y$$

Equation 20: Additional Differential Thrust Controllers

The full control law is then

$$\vec{u} = (0 \ ; \ D_2 - x_d - y_d \ ; \ D_3 - x_d + y_d \ ; \ 0 \ ; \ 0 \ ; \ D_6 + x_d + y_d \ ; \ D_7 + x_d - y_d \ ; \ 0)$$

Equation 21: Full Translation Control Law

It is important to note that this controller does not assume the specific location of the spacecraft's center of mass but does assume that its projection onto the XY plane falls between the Z-axial thrusters. This is necessary to be able to balance torques using only the Z-axial thrusters. This assumption was verified through SolidWorks analysis of CaliPER's model with all subsystems integrated. Since the assumption holds, the control law described in Equation 21 was implemented and tested. There is a similar method; however, of producing a balanced torque around the center of mass that holds even when the XY projection of the center of mass does not fall between the Z-axial thrusters. Finding the appropriate base duty cycles still involves solving Equation 18 but now considering two additional thrusters beyond 2, 3, 6, and 7. If the center of mass were at a higher Y-value than the upper Z-axial thrusters 2 and 7, firing the axial thruster would produce a torque in the -X direction. Thrusters 4 and 5 should additionally be included in Equation 18 to generate a torque in the +X direction. Activating these six thrusters would produce a force with both a Z and a Y-component in the body frame. Once the system is solved for the torque balanced duty cycles, the direction of this force can be found. The slewing step that precedes the translation should be planned now to map the direction of this force, instead of the Z-axis, onto the reference velocity vector.

4.1.3 Desaturation Controller

The desaturation control loop (Figure 11) is responsible for dumping the momentum accumulated by the reaction wheels over the course of the mission. This is accomplished by firing thrusters to produce a torque which decreases the angular momentum stored in the reaction wheels.

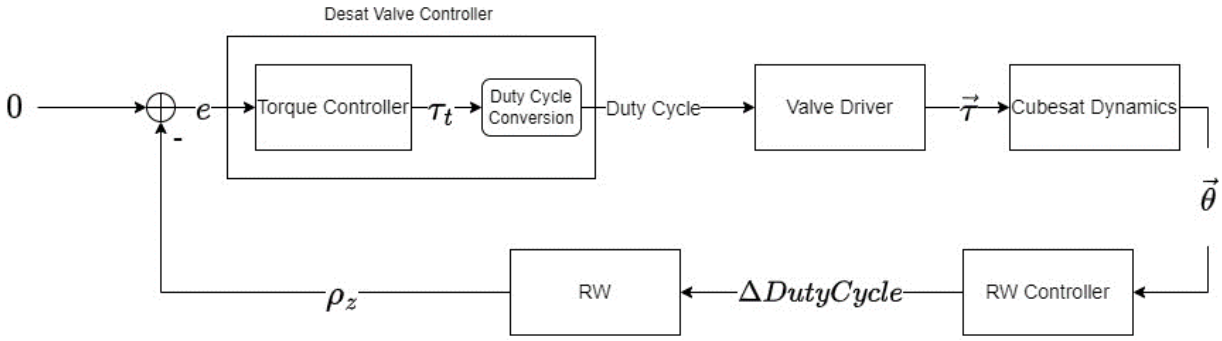

Figure 11: Desaturation Control Loop Block Diagram

Since the only thrusters capable of producing torque without net force are the pairs 1 and 5, and 4 and 8, and since they produce thrust around the z-axis, momentum will be dumped one axis at a time through the z-axis. Much like the maneuvering controller then, the desaturation controller will operate in stages. It will begin by dumping momentum in the current z-axis, then slew 90° around x so that the z-axis aligns with the previous y axis. It will then dump the y-axis momentum, slew so that the z-axis aligns with the original x-axis, and dump momentum in the final direction. The controller exploits the fact that angular momentum of a body remains fixed in an inertial frame if no external forces are present. This allows the body z-axis to take on the angular momenta of all three inertial axes at different points in time.

The controller takes in a reference angular momentum for the current z-axis. This reference is always 0 since the controller aims to minimize angular momentum. The torque controller uses the error to produce a target induced torque. An analogous duty cycle conversion process as in the translation controller takes place, and a command vector is sent to the valve driver. The torque imparted by the thrusters will begin to influence the orientation of the cubesat, but will be balanced by a change in momentum brought on by the reaction wheel controller, which is set to maintain CaliPER's attitude throughout the time when the desaturation controller is active. This change in angular momentum produces the desired effect of lowering ρ_z, the angular momentum stored among the reaction wheels in the z direction. While its operation is quite similar to the other controllers, there are some notable differences in the structure of this control loop. While the controller operates through the attitude vector, there are no relevant outputs here since it does not seek to alter the attitude. Instead, the length of the control loop from the output of the valve driver to the output of the RWs is analogous to the plant in a standard control loop. The only output value of interest is ρ_z.

The control law for this controller uses a standard PID law on the error term. The result of the PID is the coefficient used to activate either thrusters 1 and 5, in the event that the error is negative or thrusters 1 and 4 if it is positive.

4.1.4 Temperature Controller

The temperature control loop (Figure 12) is responsible for preheating propellant in preparation for burns. The target temperature, which is always 30°C when this controller is

activated, is compared to the estimated temperature as reported by a resistive temperature device (RTD). The error is fed to an on/off controller which controls the heater inside the propulsion system. The on/off controller can additionally be inhibited by a reading above the threshold 75psi to activate the heating failure sequence of the FSM.

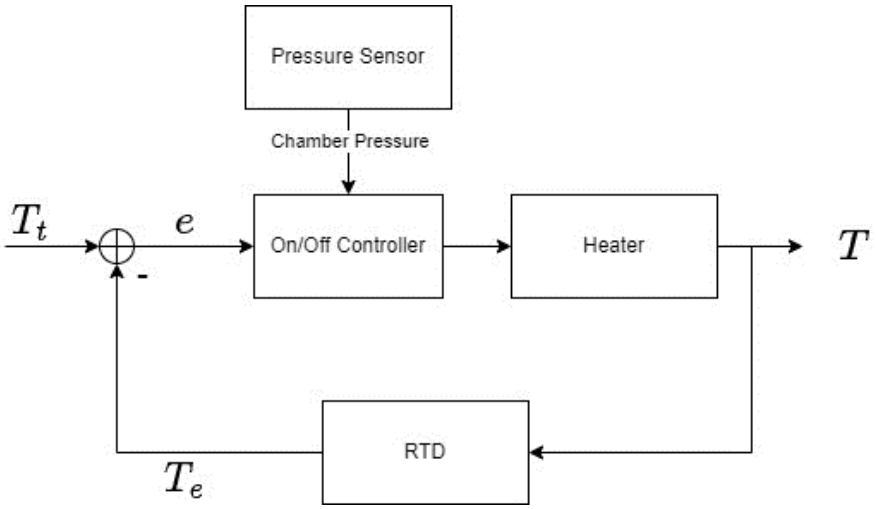

Figure 12: Temperature Control Loop Block Diagram

4.2 Supporting Circuitry

4.2.1 Solenoid Valve Driver

The propulsion system is currently designed to use the “High Speed In-Line Solenoid Valve” manufactured by the Lee Company [15]. The manufacturer has indicated that a spike and hold driver is required for this solenoid valve and has provided a schematic for this circuit (Figure 13). Duty cycles produced by the desaturation and maneuvering control loop will be input to the control signal input of the circuit. The NE555P generates pulses, the timing of which is defined by R1 and C1. The pulses are output to the base of Q2 to deliver the spike voltage V1 to the solenoid, after which hold voltage V2 is delivered. The hold voltage remains across the solenoid until the control input signal is lowered, cutting off Q4.

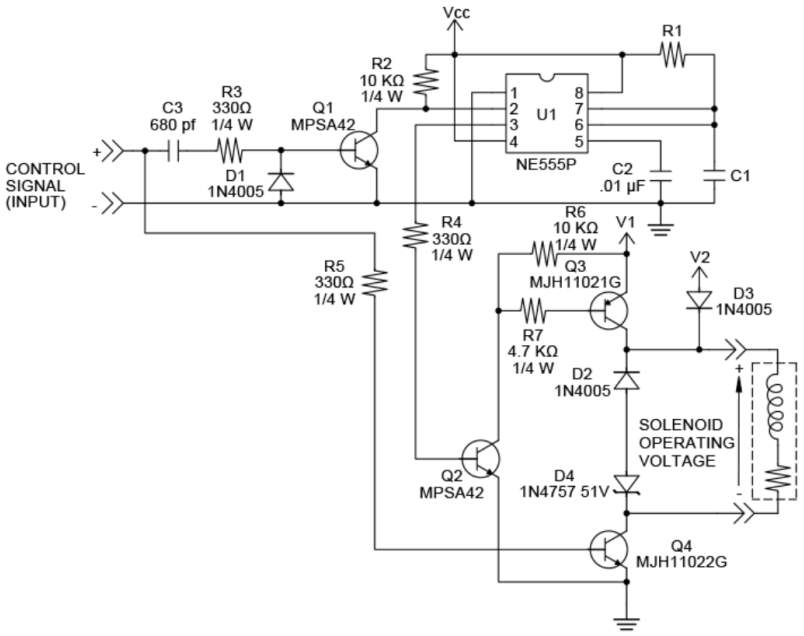

Figure 13: Spike and Hold Solenoid Valve Driver [16]

4.2.2 Temperature Sensing Circuit

An RTD will report the temperature of the propellant chamber. The sensor will need to be biased in order to take the reading. Current source biasing is a common technique that offers greater accuracy than voltage division. As such the current source shown in will be used for biasing the RTD with a DC voltage Vin and current determined by Rsense. The design assumes an interface with CaliPER's onboard computer that allows for analog reading of the current across the RTD.

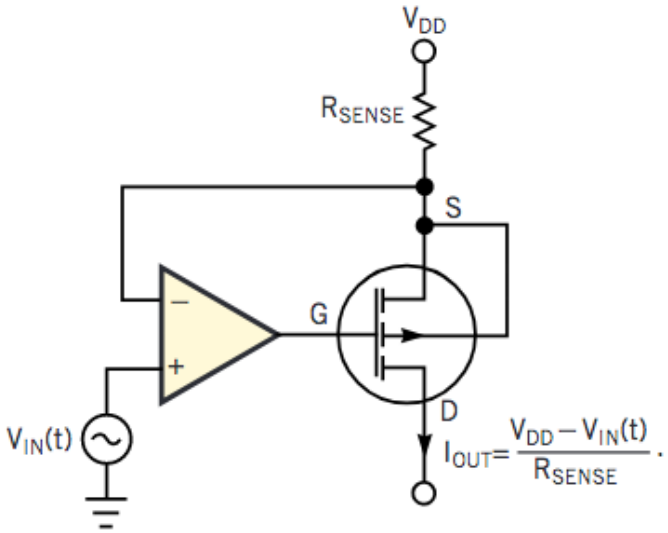

Figure 14: RTD Biasing Current Source

5 Design Dependent Technical Specifications

A number of design dependent requirements constrain each of the control systems such that they can meet the mission requirements as outlined in the design independent specifications and as described by the work of other sub teams. These design dependent requirements are summarized in Table 3, Table 4, and Table 6.

PCM1 comes from the maximum rated frequency for the selected solenoid valves. PCM2 is derived from analogous burn times during the MarCO mission. The longest optimal TCM burns calculated for the MarCO cubesats lasted over 160s and CaliPer's estimated power budget accounts for burns of up to 240s [17]. The spec considers a maneuver that involves an acceleration burn for 240s and a deceleration burn for 240s to stop at a position with added margin. The REASON beam pattern is to be calibrated within 0.1° degree accuracy. PCM3 is derived from the linear distance swept out by a 0.1° arc with CaliPER at closest approach, 75 km away from Clipper.

$$tan(0.1) * 75km = 131m$$

Again taking MarCO's 162s burn as representative, the CaliPER could drift at a speed of

$$131/162 = 0.8m/s$$

and just reach the edge of the 0.1° accuracy within a TCM period. Taking a 0.75 safety factor to minimize the chance of driving the GNC system to the edge of its stability yields a steady state velocity error of

$$tan(0.075) * 75km/162s = 0.6m/s$$

PCM4 comes from the PWM frequency expected by the electronic speed controller of the SURPASS 1100 KV BLDC motor.

Designator	Spec	Value	Measurement
PCM1	Maximum Valve PWM frequency	500Hz	Oscilloscope measurement of Raspberry Pi Output
PCM2	Maximum maneuvering stabilization time	500s	Simulink simulation
PCM3	Maximum Steady State Error	0.6m/s	Simulink simulation
PCM4	Maximum RW PWM frequency	50Hz	Oscilloscope measurement of Raspberry Pi Output

Table 3: Design Dependent Technical Specifications for the Maneuvering Control Loop

PCD1 mirrors the specification for time allotted for the GNC system to stabilize the spacecraft. PCD 2 is a reflection of the sensitivity of the RW reported speed. As per R1.4 of the RW design dependent specifications: “The reaction wheel design shall maintain bidirectional, closed-loop control within 200 rpm of true angular velocity of the flywheel.” [2] The angular momentum that is associated with this sensitivity for the chosen flywheels is

$$L = J_{RW} * 200 * 0.10472 \frac{rad}{s} RPM^{-1} = 1.8 * 10^{-4} * 200 * 0.10472 = 3.7692\, mNms$$

With a guaranteed capacity of at least 0.1Nms, the sensitivity allows for reliable desaturation to near 3% of capacity. With additional margin this was approximated to 5% capacity.

Designator	Spec	Value	Measurement
PCD1	Maximum desaturation time	120s	Simulink simulation, 1D desaturation test
PCD2	Maximum Steady State Error	5% of capacity	Simulink simulation, 1D desaturation test

Table 4: Design Dependent Technical Specifications for the Desaturation Control Loop

Chamber Temperature (deg C)	**Single Nozzle Thrust (mN)**
50	247
40	186
30	136
20	97
10	68
0	46
-10	30
-13	25
-20	18
-30	10

Table 5: Predicted Single Nozzle Thrust by Chamber Temperature [3]

Per National Space Transport System (NSTS) standard 1700.7B, 100 psi is the threshold beyond which a container is considered a pressure vessel and must adhere to additional regulation [18]. The heating system must then ensure that heating never causes propellant to surpass this threshold. 75 psi is additionally the pressure at which the designed nozzles began to

release some liquid propellant with gas propellant, which reduces thrust efficiency. PCT2 is again inherited from the estimated power budget which allots 2 hour segments for propellant preheating. PCT3 is derived from testing of thrust vs. temperature as described in Table 5. Thrusts vary most significantly with respect to temperature at higher temperatures. Approximating a 10% change in thrust in the highest temperature range measured yields a corresponding 4°C change.

$$\frac{10℃}{247mN - 186mN} * 0.1 * 247mN \approx 4℃$$

Designator	Spec	Value	Measurement
PCT1	Maximum Pressure	75 psi	Simulink simulation, temperature control physical test
PCT2	Maximum settling time	2 hours	Simulink simulation, temperature control physical test
PCT3	Maximum Steady State Error	4℃	Simulink simulation, temperature control physical test

Table 6: Design Dependent Technical Specifications for the Temperature Control Loop

6 Simulation Implementation

Simulation of CaliPER's dynamics was implemented in a MATLAB base workspace containing supporting functions which describe the dynamics and controllers. These functions interact with one another as well as sensor and actuator blocks in a connected Simulink workspace. An overview of the architecture of the Simulink workspace is shown in Figure 15. Source Simulink and MATLAB code can be found at https://github.com/kristian4488/es100_prop_control.git.

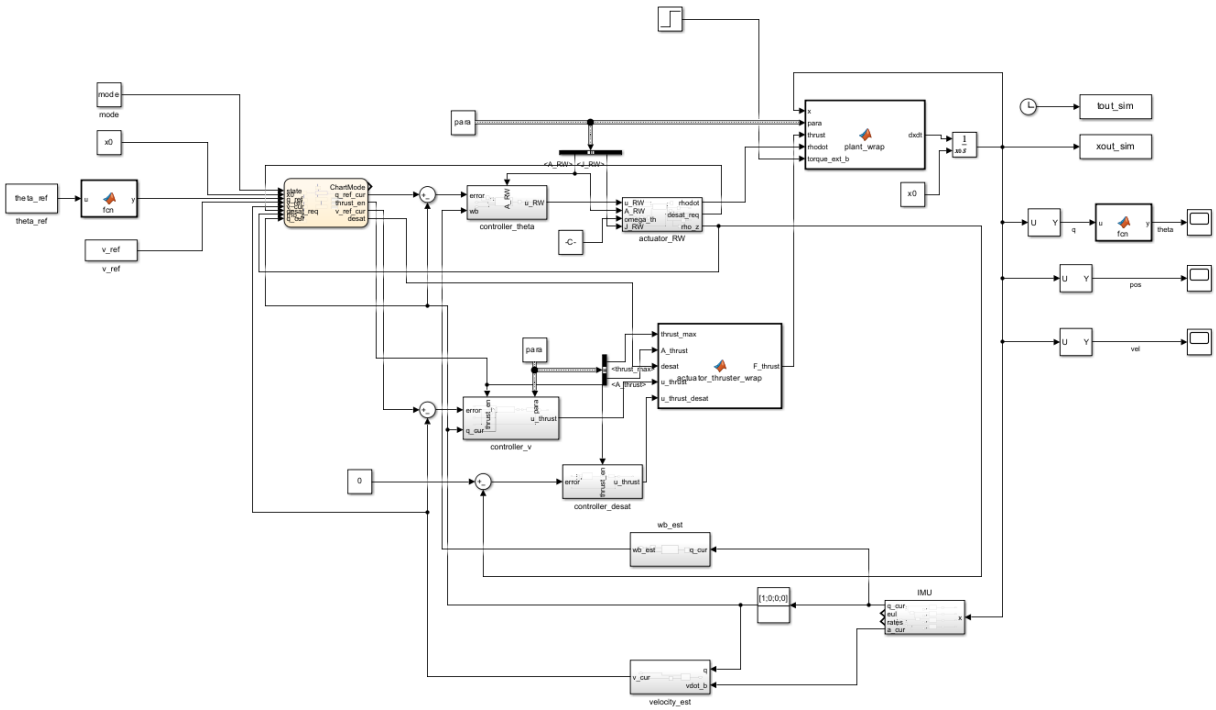

Figure 15: Full Simulink Workspace Architecture

6.1 CubeSat Dynamics

Physical parameters of the cubesat with its subsystems integrated were modelled by McMullin [4]. The spacecraft's mass and inertia tensor were extracted from this model. The inertia tensor of the deployed state was used for testing the propulsion control system since this will be the cubesat's state during calibration, when activation of the propulsion system will be most frequent and pointing requirements will be most stringent. Model parameters can easily be changed by updating the principal moments for simulation of additional mission phases.

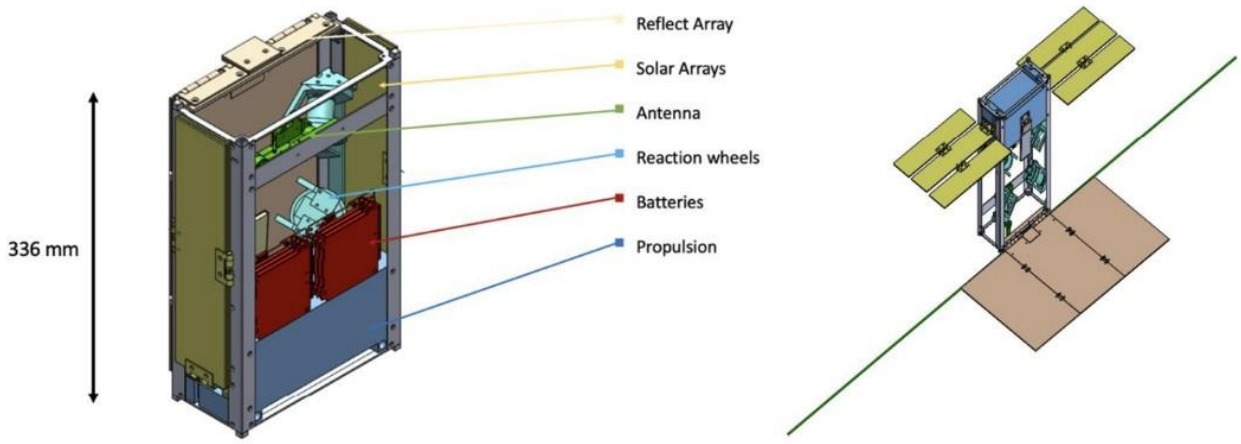

Figure 16: CaliPER Models. Left: undeployed state with subsystem locations shown. Right: Deployed state.

The differential equation of the state vector is built using Equation 1, Equation 3, Equation 9, combined with actuator outputs $\vec{F}_{thrust}$ and $\dot{\vec{\rho}}$, and supporting equations to convert between body and inertial frame quantities. The state vector equations are based on the general rigid body simulator implemented by Ayora [5], with updates to reflect reaction wheel dynamics. The first order differential equation system describing the state is

$$\frac{d}{dt}\begin{pmatrix} q \\ \vec{\omega}_b \\ \vec{\rho} \\ \vec{x}_b \\ \dot{\vec{x}}_b \\ \vec{x} \\ \dot{\vec{x}} \end{pmatrix} = \begin{pmatrix} 0.5 * T_{\frac{\omega b}{quat}} * q \\ J^{-1}\left(\vec{\tau}_{\frac{net}{b}} - \dot{\vec{\rho}} - \left(\vec{\omega}_b \times (J\,\vec{\omega} + \vec{\rho})\right)\right) \\ \dot{\vec{\rho}} \\ \dot{\vec{x}}_b \\ \frac{\vec{F}_{net}}{m} - \left(\vec{\omega}_b \times \dot{\vec{x}}_b\right) \\ q^{-1} \otimes \dot{\vec{x}}_b \otimes q \\ q^{-1} \otimes \frac{\vec{F}_{net}}{m} \otimes q \end{pmatrix}$$

Equation 22: Dynamics System of Differential Equations

where $T_{\frac{\omega b}{quat}}$ is the matrix which transforms $\vec{\omega}_b$ to a quaternion attitude derivative as described in Equation 9 and J is the inertia tensor. $\vec{F}_{net}$ is the sum of thrust forces and $\vec{\tau}_{net}$ is the sum of torques generated by the thrusters

$$\vec{\tau}_{thrust,i} = (\vec{x}_{nozzle} - \vec{x}_{cm}) \times \vec{F}_{thrust,i}$$

and externally input environmental torques. The physical quantities $J, m, \vec{x}_{nozzle,i}, \vec{x}_{cm}$ are defined in the main MATLAB function 'run_sim' from which the simulation is called in a parameter 'para'. An initial condition for the differential equation 'x0' is also defined here. The

parameter and initial condition are passed to the ‘plant_wrap’ Simulink block which wraps the ‘plant’ function containing the differential equation.

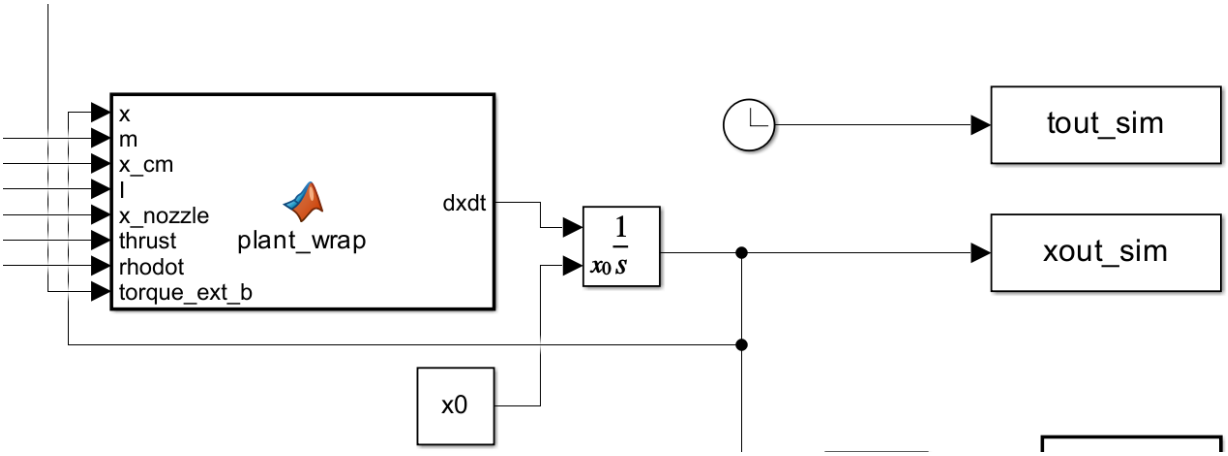

Figure 17: Simulink Plant Block. The state vector and time are passed back to the MATLAB workspace on simulation completion for data logging.

The differential equation solver used in Simulink is ODE4 (Runge-Kutta) with a time step 0.001s. These settings were chosen to capture the fastest events in the controller (up to 500 Hz = 0.002s period operating speed of the solenoid valves) while being able to complete simulations of several hundred seconds in under approximately 10 minutes.

6.2 Finite State Machine

The drift, maneuvering, and desaturation states of the simplified FSM shown in Figure 6 were implemented in Simulink alongside a reset state entered only once to initialize the state. The full FSM of section 10.1 was not implemented since all relevant states for testing except reset are captured in the simplified FSM. Inputs and output of the FSM are shown in the exterior view of the block (Figure 18) and state definitions are shown in the interior view (Figure 19).

The ‘v_ref_cur’ and ‘q_ref_cur’ are used to select between pure slewing and a velocity set maneuver in the overall Simulink architecture. When performing a pure slew, the quaternion ‘q_ref’ of reference value ‘theta_ref’ defined in run_sim.m is passed to ‘q_ref_cur’ while ‘v_ref_cur’ remains unchanged. When performing a velocity maneuver, ‘v_ref’ is passed to the output while ‘q_ref_cur’ remains unchanged. ‘thrust_en’ allows the FSM to enable or inhibit all thruster control. This output is passed to all thruster valves, and in a flight implementation would be the only signal passed to the ninth control valve of the propulsion system.

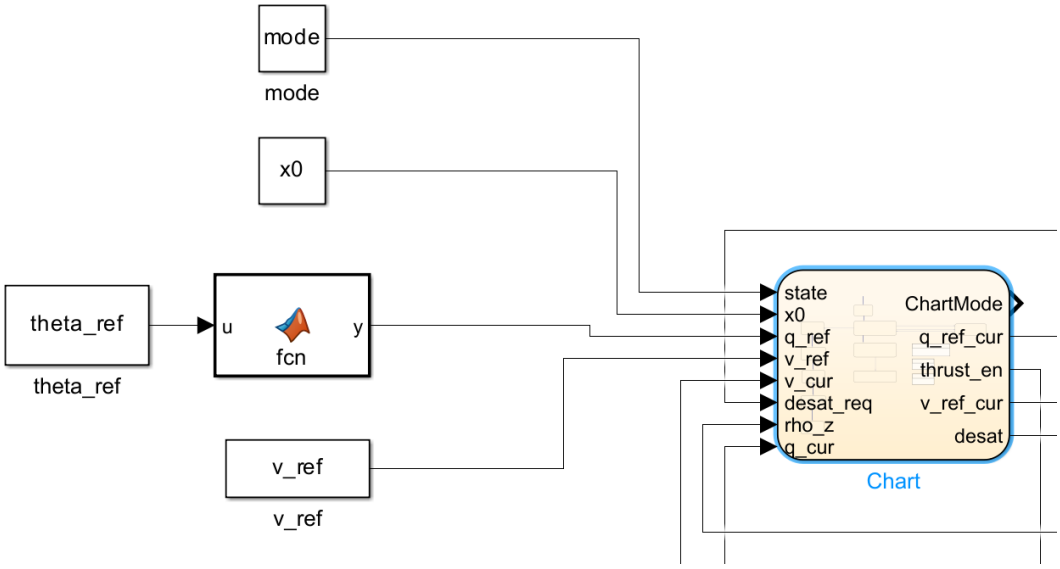

Figure 18: FSM Exterior View

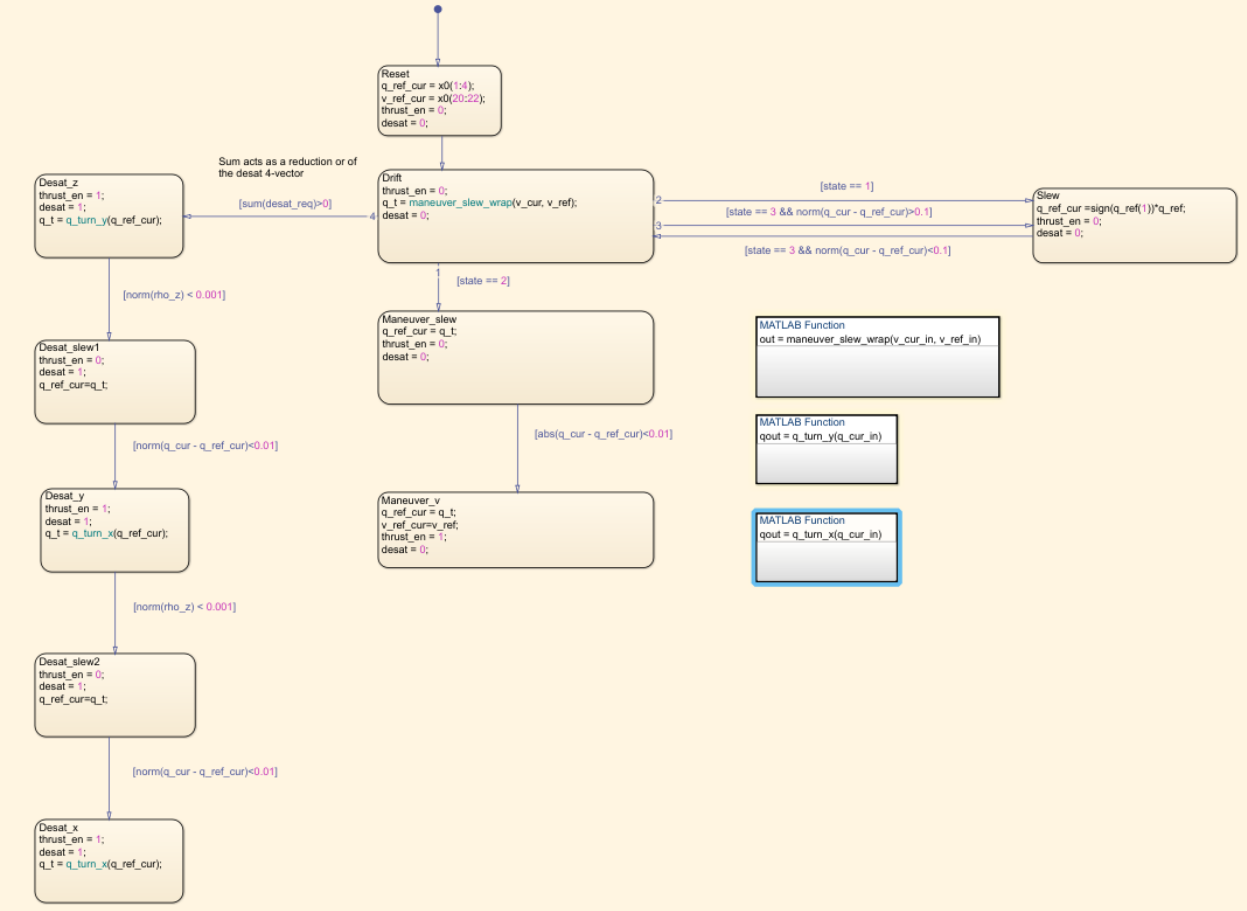

Figure 19: FSM Interior View. The FSM begins in the Reset State which descends directly from the start point. It proceeds to drift, from which it can Slew (right), Maneuver (central states) or Desaturate (left states). Additional helper functions are included in the lower right.

The Simulink FSM implementation explicitly splits the maneuvering state into its slewing and linear movement portions and introduces a ‘maneuver_slew’ function. This function calculated the attitude quaternion associated with velocity in a commanded direction as

$$q_t = (\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\hat{n})$$

Equation 23: 'maneuver_slew' output quaternion

where $\theta = \cos^{-1}(\hat{z} \cdot \hat{v})$ and $\hat{n} = \frac{\hat{z}\times\vec{v}}{||\hat{z}\times\vec{v}||}$. Transition from slewing to linear movement occurs when the spacecraft’s attitude is such that its Z-axial thrusters match the input velocity direction to within a margin where the magnitude of the difference of quaternions is below 0.01. This margin was chosen to keep the slewing portion of the maneuver within the 120s allotted slewing time while achieving sufficient accuracy in Z-axial thruster direction. The general desaturation state is similarly split into Z, Y, and X desaturation states, separated by repointing states.

6.3 Controllers

All controllers share the overall Simulink workspace and pass their outputs to the plant. The FSM is responsible for selecting an active controller. The signal path of each controller within the overall workspace is presented alongside the controller’s interior.

6.3.1 Maneuvering Controller

6.3.1.1 Slewing

The slewing controller is utilized during both pure slewing maneuvers and the first portion of a linear translation maneuver. The signal path of this control loop is highlighted in Figure 20. It takes its reference ‘theta_ref’ from the MATLAB base workspace as a set of Euler angles for readability; however, attitude calculations in the final version of this controller are performed using quaternions. The reference signal is compared to the current attitude to produce the error signal which is fed to ‘controller_theta’ which produced control output vector ‘u_RW’ to be passed to the actuator block ‘actuator_RW_wrap.’ The actuator produces a torque which influences the plant. The relevant portions of the state vector are read in the ‘IMU’ block to generate a new estimated attitude.

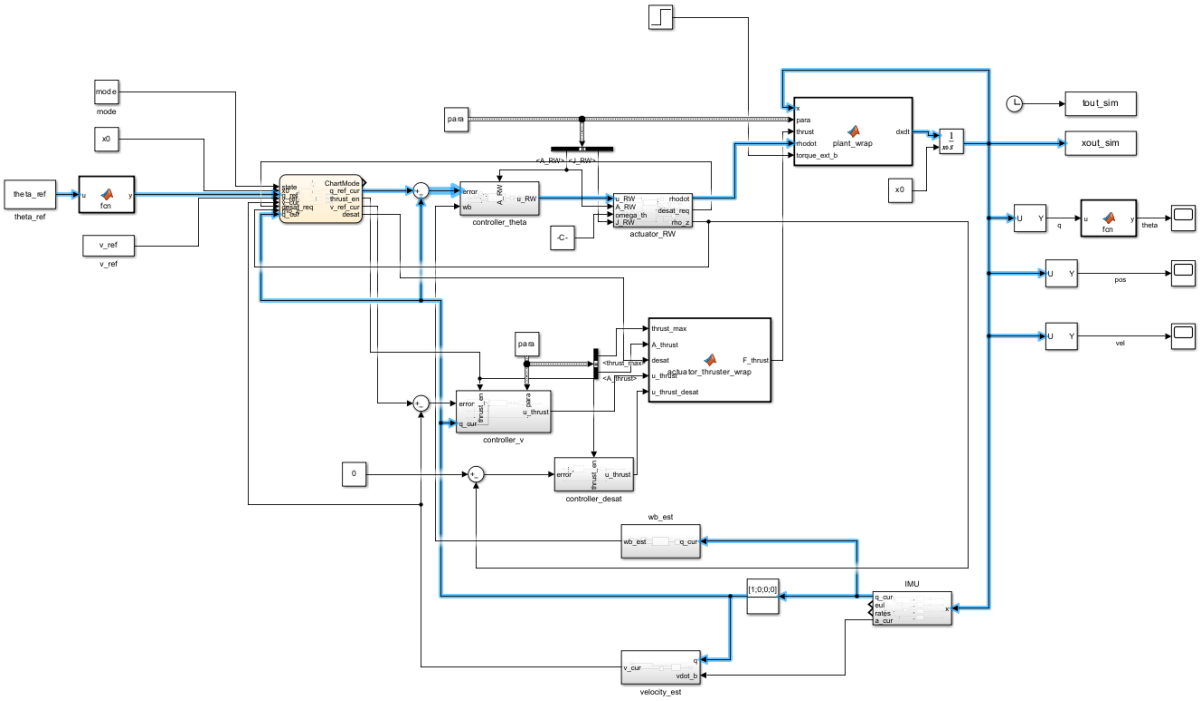

Figure 20: Slewing Controller Signal Path

The interior of ‘controller_theta’ (Figure 21) takes the necessary functions of the error (only proportional, and integral in the final controller), and attitude derivative, and passes them to a wrapper block ‘theta_PID_wrap’ containing the control law. The wrapper returns the control duty cycle vector.

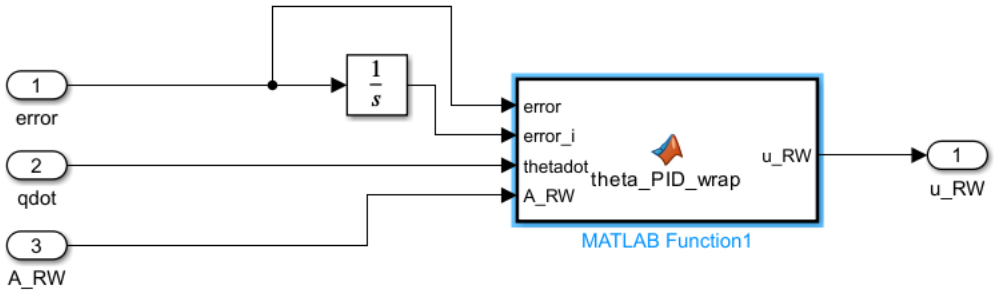

Figure 21: 'controller_theta' interior

Each of the control laws outlined in section 4.1.2.1 were implemented in the wrapper in different versions of the Simulink model. Controllers 1 and 2 are archived in branches error_d and euler_quat_plant of the codebase respectively. Controller 3 is archived as commit e6e90aa of the main branch and controller 4 is represented in the latest commit of main.

Initial testing of controller 1 confirmed that it is able to slew to a target orientation along one axis as shown in Figure 22; however, it exhibits an inefficient usage of the RWs where in its

settled state, both pairs of wheels are active and producing torques in opposite directions (Figure 23). This would cause unnecessary power drain and leave the spacecraft particularly vulnerable to any imbalance in the RWs. The behavior arises from using only the error in the control law in a pyramidal RW setup. In a 3 perpendicular RW setup, the angular velocity of each RW controls an independent axis, and so the body could not reach steady state with any RW rotating; however, in a 4 RW setup the redundancy allows the angular momenta to cancel in steady state.

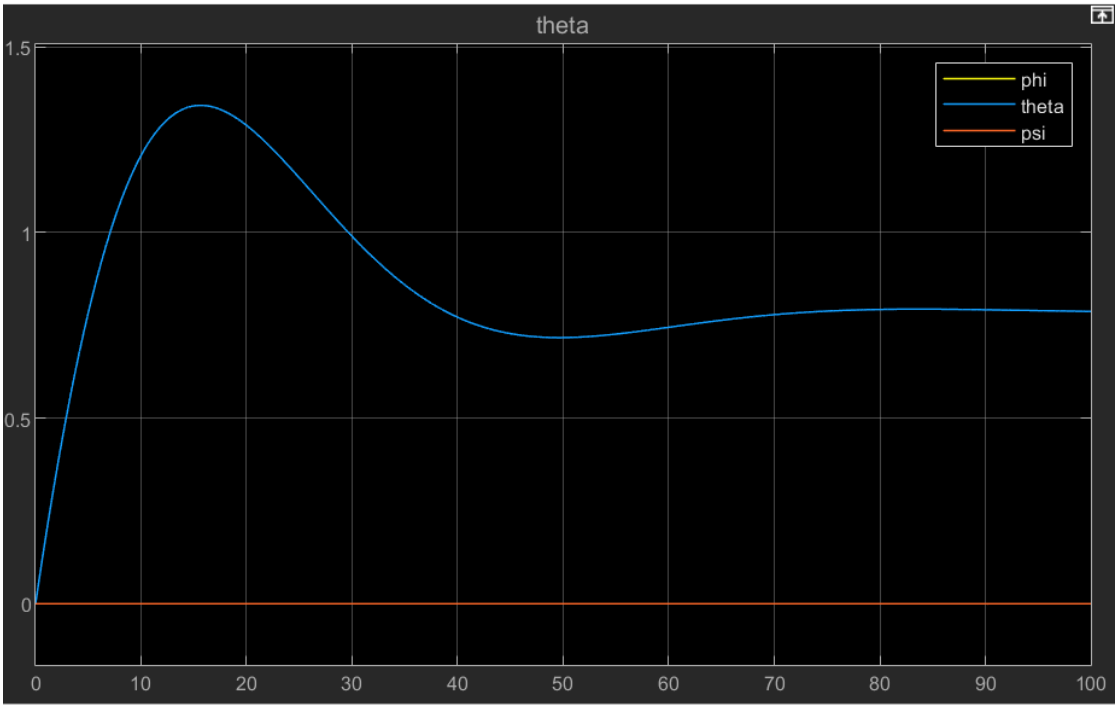

Figure 22: Controller 1. theta_ref = [0, pi/4, 0]

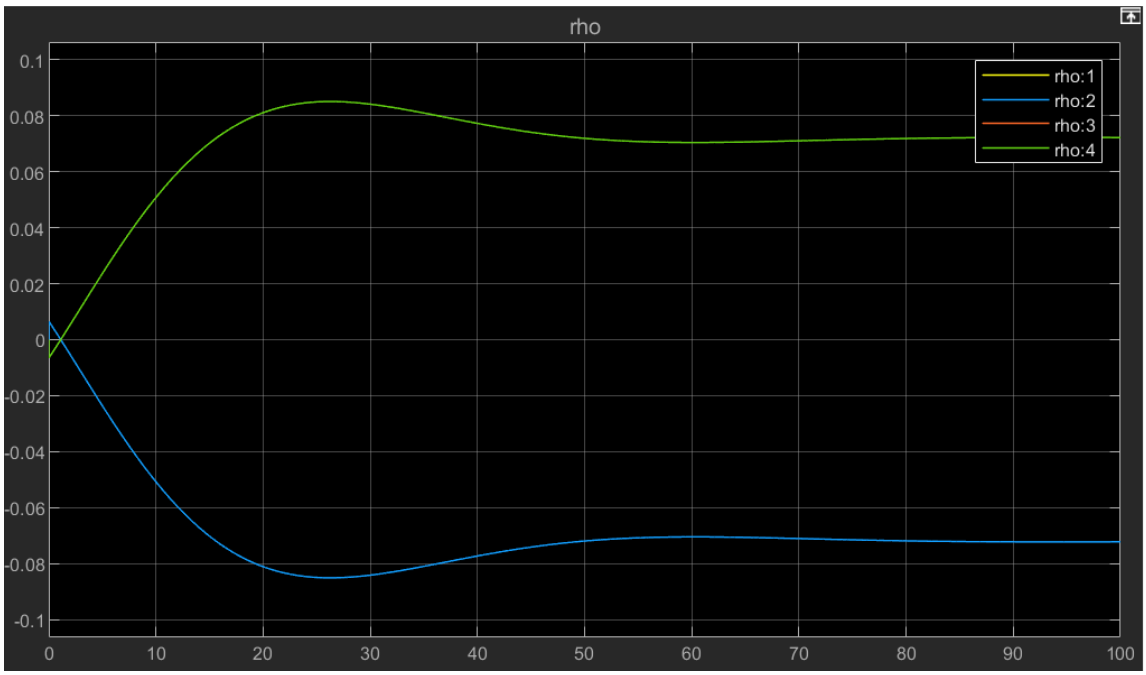

Figure 23: Controller 1 RW angular momenta. theta_ref = [0, pi/4, 0]

Nonideal usage of the RWs encouraged the implementation of Controller 2, utilizing $\dot{\vec{\theta}}$ for its derivative term. This update removed the overshoot in the orientation (Figure 24), and drove the controller to settle with the RWs not spinning on completion of the maneuver (Figure 25). This controller was also proven to operate as predicted in select multi axis slewing case (Figure 26). Additional information on multi axis slewing is provided in section 7.

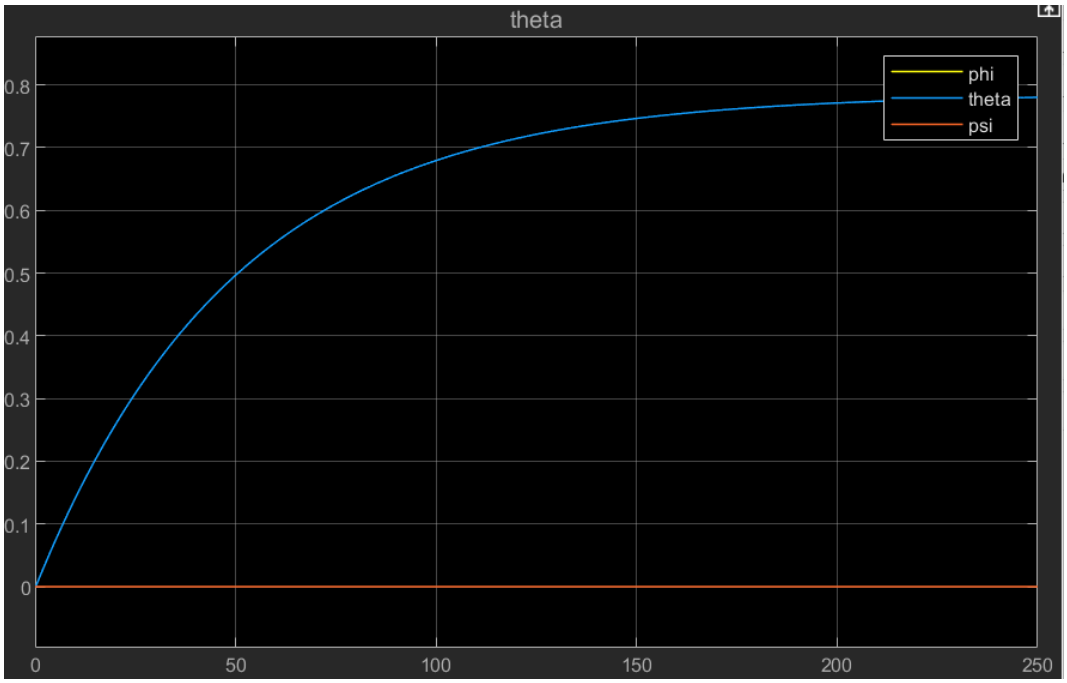

Figure 24: Controller 2. theta_ref = [0, pi/4, 0]

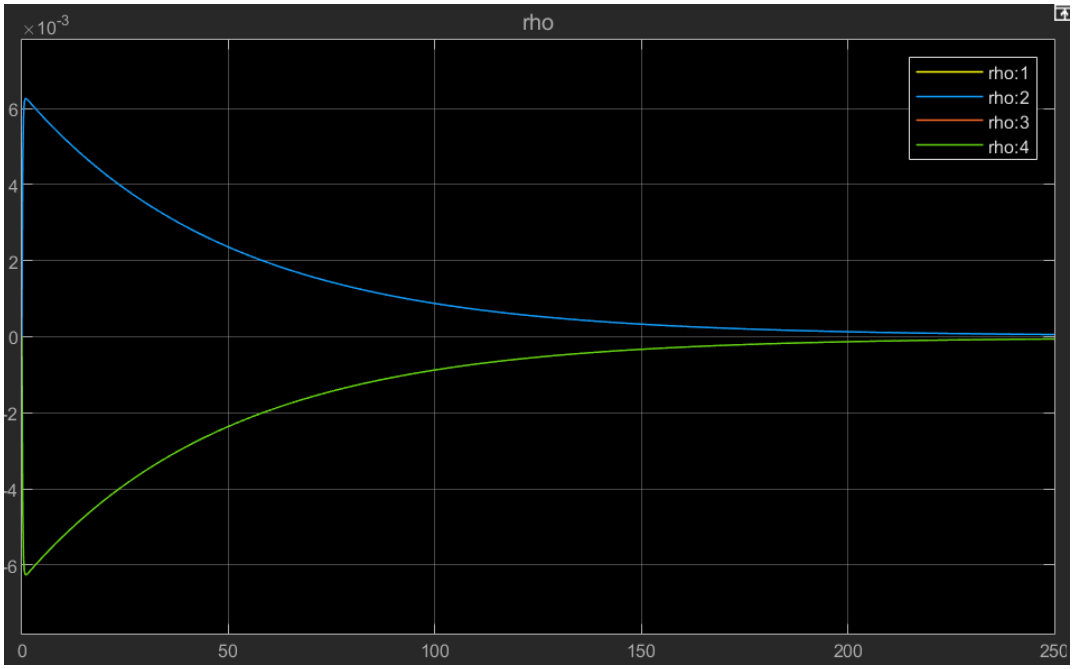

Figure 25: Controller 2 RW angular momenta. theta_ref = [0, pi/4, 0]

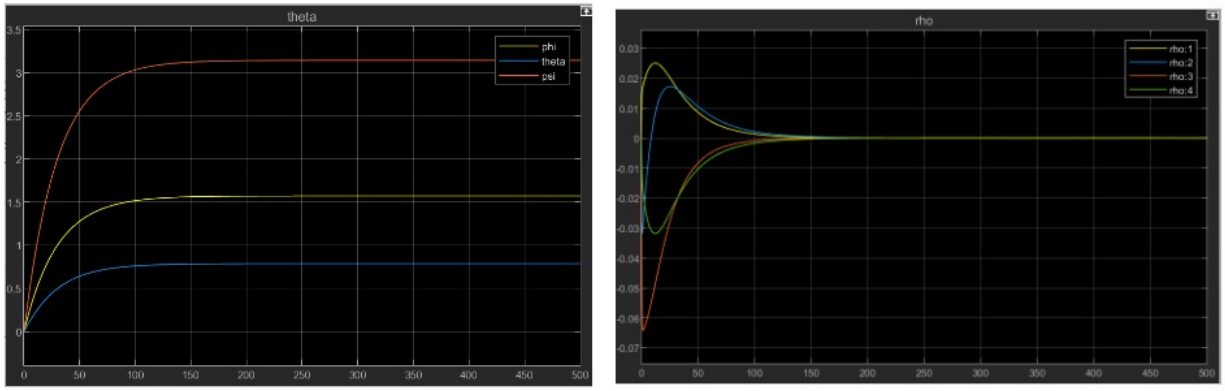

Figure 26: Controller 2 Attitude and RW Angular Momenta. theta_ref = [pi/2, pi/4, pi]

Controller 2 was however, unable to slew to or control its orientation from attitudes with a pitch of pi/2. This results from the singularity in the transformation matrix containing tan in Equation 8. This inherent weakness of utilizing Euler angles for attitude representation is associated with gimbal lock. While this phenomenon is not detrimental to aircrafts where pitch must be limited during maneuvers, it necessitates the addition of additional checks on maneuvering in spacecrafts where full range of motion could feasibly be commanded. This further encouraged the implementation of the quaternion based third controller. This controller accurately slewed in cases with a pitch of pi/2 and retained the RW usage properties of controller 2 as shown in Figure 27. Controller 3; however, suffered from slow settling times depending on the choice of input, particularly in multi axis slewing cases. Examining the quaternion error revealed that slow settling was a result of this control law not utilizing the real part of the

quaternion. Since this component was undriven, its error would slowly proceed to zero, and in certain cases could not stabilize around zero. This motivated the creation of the fourth controller, which utilizes the body angular velocity for its derivative component. Using the body angular velocity indirectly incorporates all components of the quaternion through the transformation $T_{\frac{\omega b}{quat}}$. As the final slewing controller, full analysis of its performance is presented in section 7.

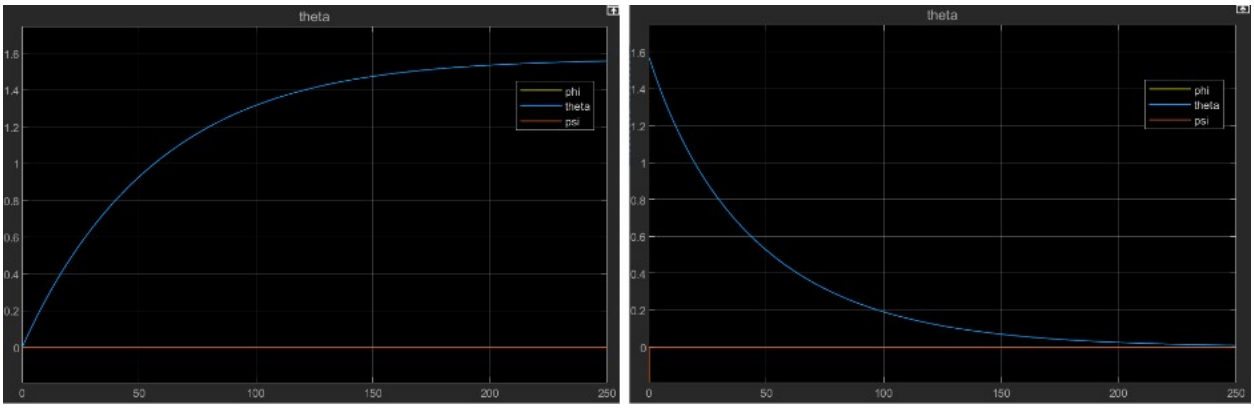

Figure 27: Controller 3. Slewing to and from a pitch of pi/2

6.3.1.2 Translation

The translation controller takes a velocity ‘v_ref’ which is fed by the FSM as a reference value for comparison to the current estimated velocity. The error is passed to the ‘controller_v’ which generates the control duty cycle vector for the solenoid valves. Since ‘thrust_en’ is implemented as a separate variable, the duty cycle vector contains only eight entries. The duty cycle vector is passed to the ‘actuator_thruster_wrap’ block which returns an 8x3 thrust force matrix representing the thrust vectors of each thruster. This matrix is combined into the net force acting on the spacecraft within the plant and in turn influences its velocity in the state vector. The state vector is read by the IMU to update the estimated velocity.

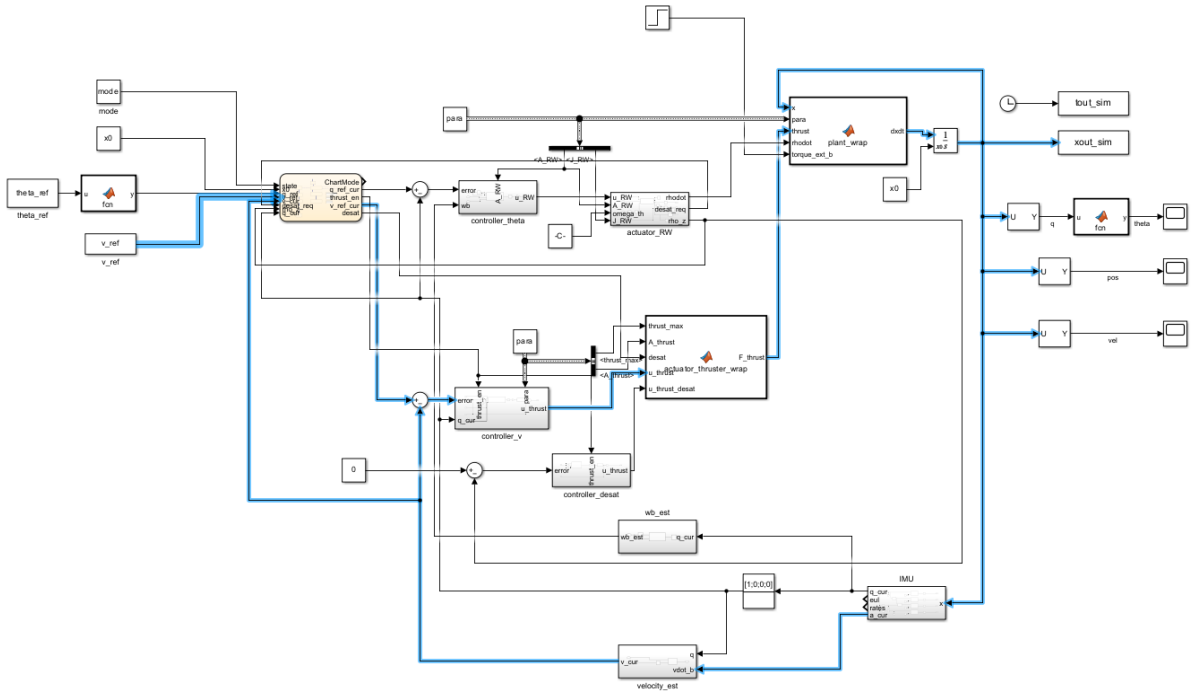

Figure 28: Translation controller signal path

Analogously to the slewing controller, the interior of the translation controller takes the necessary functions of the error (integral and derivative in this case) to pass to the 'control_v_PID_wrap' block which implements the control law outlined in Equation 17. Its output is passed to a saturator which restricts to the range [0,1] to represent a valid duty cycle. This is then passed to duty cycle conversion, which returns an 8 element vector of duty cycles. The 'thrust_en' signal is the final control for the duty cycle control vector before it is passed to the thruster block.

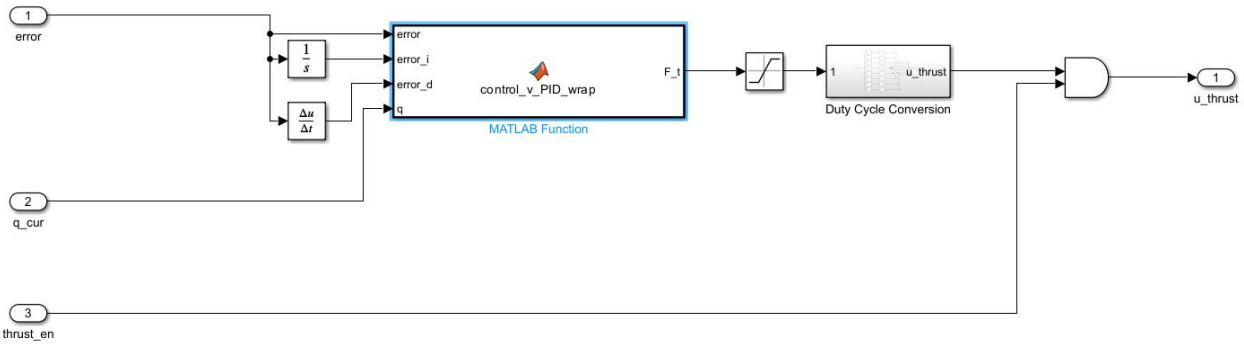

Figure 29: 'controller_v' interior view

6.3.2 Desaturation Controller

The desaturation controller takes in reference angular momentum 0Nms and compares this against the current ρ_z returned from the RWs. It uses the difference between these signals to

produce the duty cycle vector ‘u_thrust_desat’ which is passed to the thrusters. The thrusters impart a torque which is registered by the plant and the state vector, in turn influencing the current attitude ‘q_cur’ read by the IMU. The simultaneously active slewing controller produces ‘u_RW’ to change the reaction wheel angular momenta in response. This produces the desired change in ‘rho_dot’. As seen in Figure 30, while the main portion of the desaturation controller (error calculation and command thrust generation) occupies only the lower portion of the Simulink workspace, the maneuvering controller must be active to complete the signal path of the desaturation controller.

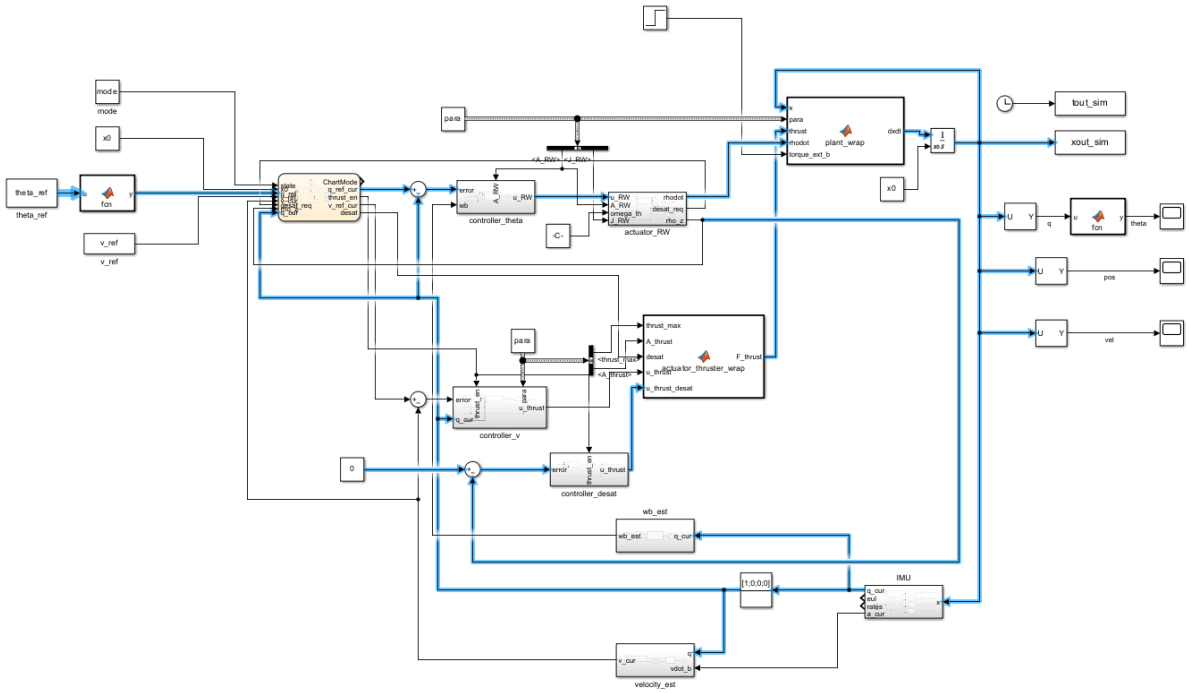

Figure 30: Desaturation controller signal path

6.4 Actuators

6.4.1 Reaction Wheels

The RW actuator block takes in physical information about the RWs: A_{RW} and the RW’s moment of inertia about the spin axis, J_{RW}. It additionally takes the commanded RW duty cycle changes u_{RW} and the RW speed associated with desaturation ω_{th}. The block generates the duty cycles of the RWs by integrating the command vector, and limiting it to the range [0.1, 0.2] which represents the valid range of input duty cycles to the motor selected by the GNC system (1120KV SURPASS BLDC Motor). The actuator_RW_wrap block executes a MATLAB function which linearly maps the duty cycle to range to $\pm\ 1571\,\frac{rad}{s} = \pm 15000\ RPM$, the maximum speed of the chosen RW motor. This block additionally returns the 3 component RW induced angular momentum $\vec{\rho} = A_{RW} * \vec{\omega}_{RW} * J_{RW}$. The block then outputs the RW induced torque by differentiating $\vec{\rho}$, as well as a vector of binary values ‘desat’. An asserted (1) bit in the

ith position of the ‘desat’ vector indicates that the ith RW has reached its threshold speed and is requesting desaturation from the FSM.

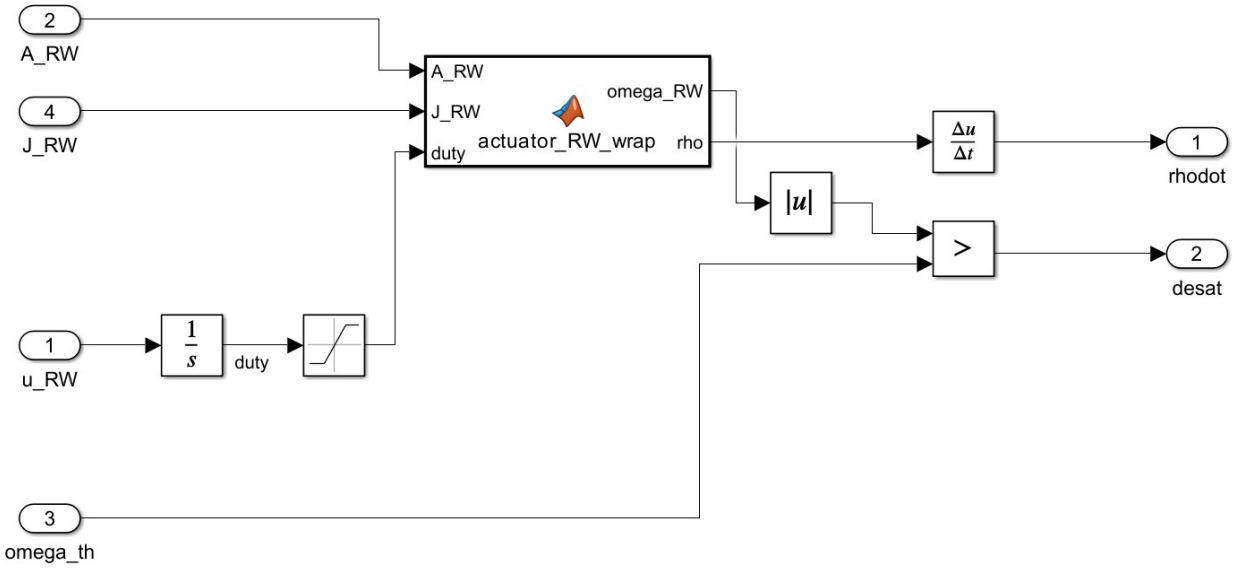

Figure 31: ‘actuator_RW’ Interior View

6.4.2 Thrusters

The thruster actuator block wraps a MATLAB function describing the relationship between the duty cycle and thrust force produced by the propulsion system. Similarly to the RWs, a matrix A_{thrust} is defined. The ith row of this matrix is a direction vector in the body frame that defines the direction of thrust produced by the ith nozzle. This matrix is then scaled by 0.136N, the thrust force predicted at 30C with a fully open valve. The simulation again matches the conditions expected for a calibration maneuver but can be updated for other temperature maneuvers in other phases of the mission. The output thrust matrix is calculated by multiplying a diagonalized version of the control duty cycle vector by the scaled A_{thrust}. The actuator_thruster_wrap block receives two control vectors, ‘u_thrust’ from the velocity controller and ‘u_thrust_desat’ from the desaturation controller. A signal ‘desat’ is passed in from the FSM as well to multiplex between the two.

6.5 Sensors: IMU

The IMU sensor subsystem simulates sampling the true state vector with the sample rate and bandwidth limitations of the BNO055. The true state vector is sliced into its relevant portions to produce the current attitude, acceleration, and body rates. Note that the body rates are not included directly in the state vector but rather are found from the body angular velocity. These values are passed to zero order holds which sample the signal at 100Hz as specified by the BNO055 data sheet to be the input and output data rate in sensor fusion mode [19]. Since the bandwidth limitations of the device in sensor fusion mode are not presented, the default

bandwidth limits for the accelerometer and gyroscope (62.5Hz, and 32Hz respectively) are converted to rad/s and passed as cutoffs to first order LPFs.

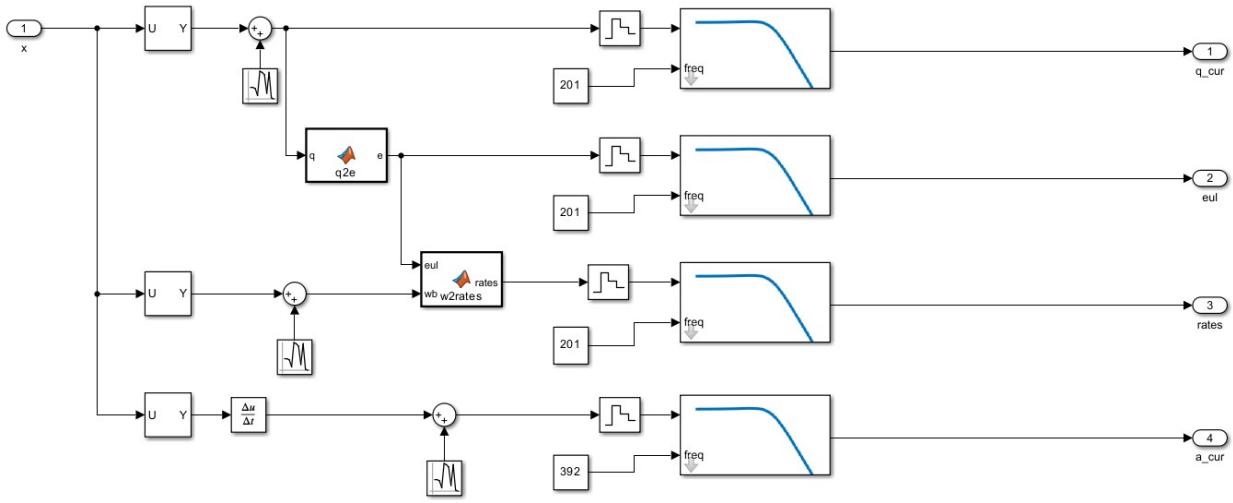

Figure 32: 'sensor_IMU' Interior view

A full simulation of IMU drift, and the star tracker and DSN update procedure is outside of the scope of this project, but error is approximately simulated by adding a noise signal directly after the inputs of the subsystem. Basic recalibration can be simulated by passing the true state vector values to the IMU at set intervals.

7 Evaluation of Results

7.1 Maneuvering Controller: Slewing

The focus of this project is on the development of a controller for the propulsion system. In keeping with this end, technical specifications of the maneuvering controller reflect the desired overall behavior of the spacecraft under maneuvers, without specific regard to the individual slewing and translation components. Although design of the slewing controller is secondary to the primary goal, understanding and tuning the performance of this controller was integral to creating a functioning overall maneuvering controller. Preliminary testing that revealed key differences between the slewing controllers was highlighted in section 6.3.1.1. The remaining treatment of this controller describes the random testing of the final controller, controller 4.

The controller was evaluated under slewing to 10 random orientations chosen from a uniform distribution of Euler angles. The proportional, integral, and derivative coefficients used were $K_p = 0.2; K_i = 0; K_d = 0.4$. The reported trials were conducted with no IMU error. Checking trials 2, 6, and 9 with angular sensor error with variance 0.1° did not appear to qualitatively change the behavior of the controller, nor did it impact the settling time significantly enough to be seen through the additional noise (the effect of noise on trial 9 as a representative can be seen in Figure 33 and Figure 34). It should additionally be noted that while the noise causes commands to vary more rapidly, the inertia of the spacecraft filters out high frequency movement. The resultant orientation is somewhat smoothed compared to the commands and again not significantly different from the case without noise, with the exception of the steady state error being largely dictated by the sensor accuracy as opposed to the controller (Figure 35). Settling time is defined as the time for the largest component of the quaternion error to come within 0.017 of its target value. 0.017 was chosen as a proxy for GS2's 0.1° accuracy requirement since sin(0.1°) = 0.017, and sin (which produce the rotation axis coefficient) varies most rapidly around 0°. The settling times of the 10 trials are summarized below.

Trial	Settling Time (s)
1	45
2	99
3	33
4	45
5	29
6	-
7	103
8	-
9	20
10	57

Table 7: Maneuvering Controller 4 Settling Times

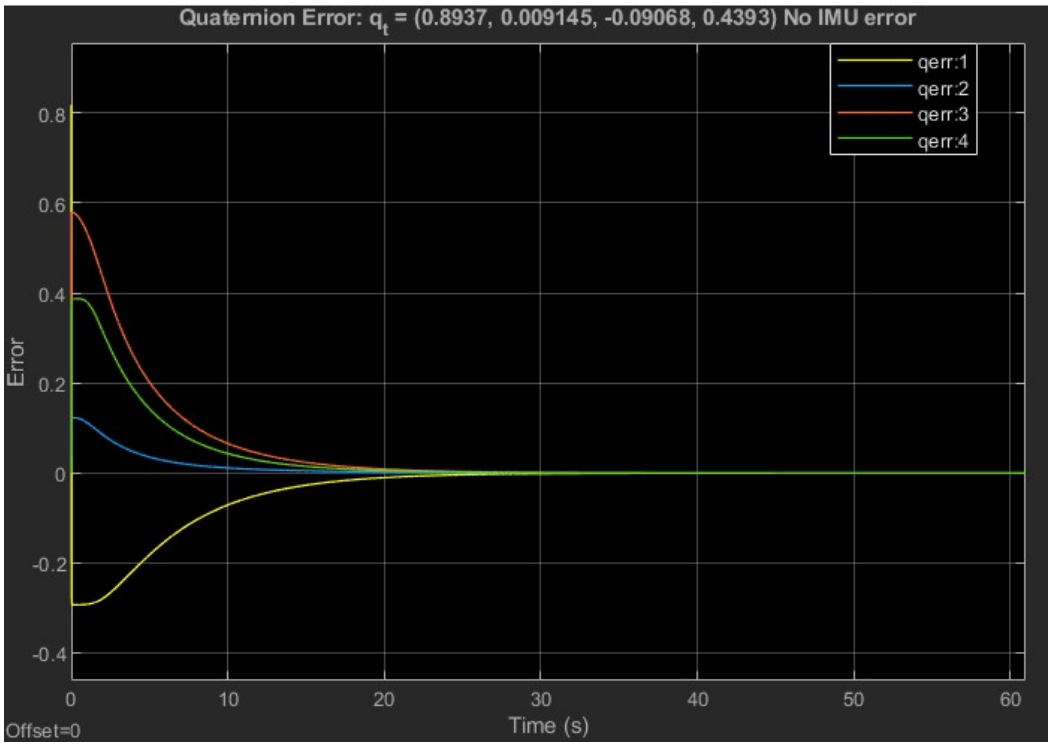

Figure 33: Trial 9 Quaternion Error, No IMU noise

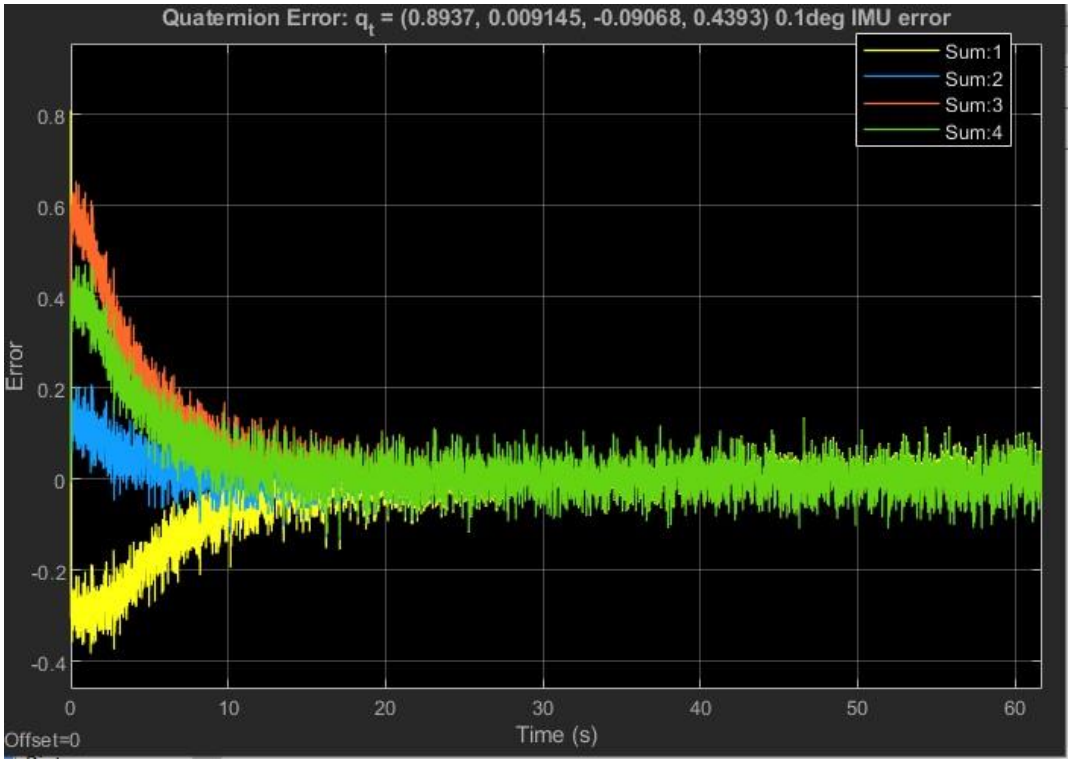

Figure 34: Trial 9 Quaternion Error with 0.1 Degree IMU noise variance

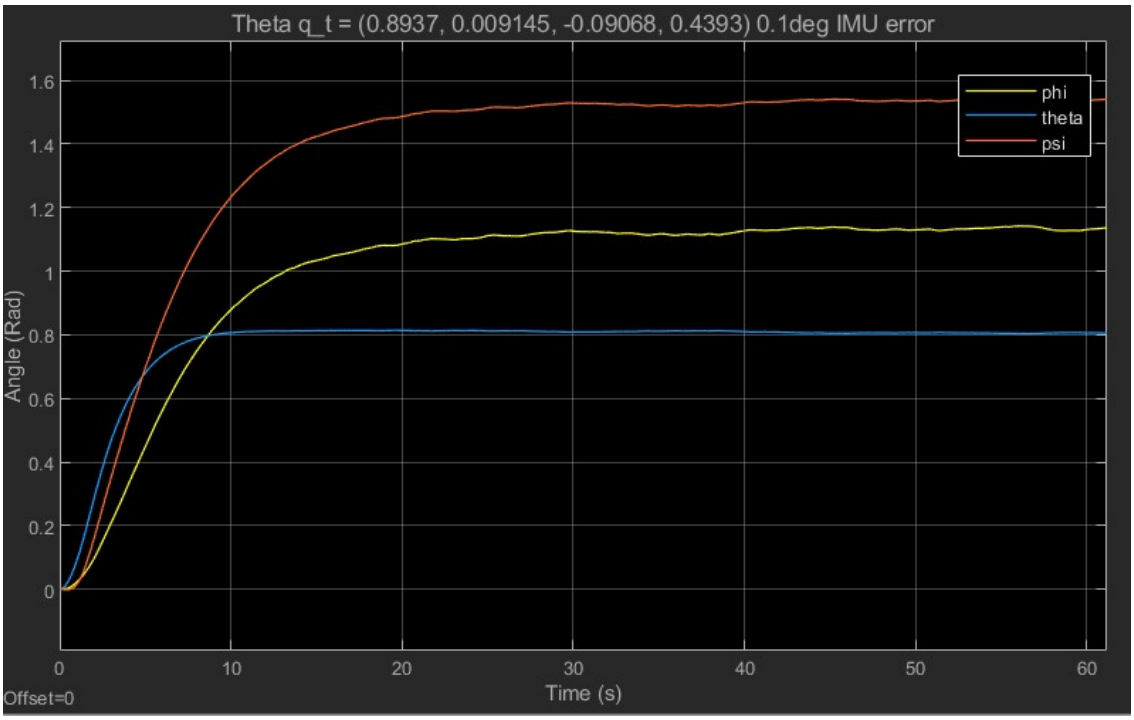

Figure 35: Trial 9 Euler Angles with 0.1 Degree IMU noise variance

Trials 6 and 8 led to growing orientation instability. During these trials the controller began to converge to the target orientation, but instability grew after approximately 150s. In these trials the controller did not meet the definition of settling at any point. The average settling time among the convergent trials was 53.875 seconds.

7.1.1 Maneuvering Controller: Multi Axis Slewing Stability

In every trial the last component to settle was the real part of the quaternion. This stems from the fact that this component is included only indirectly in the body angular velocity. Plotting the settling times against this component reveals that the stability of controller 4 appears to be correlated with the magnitude of its real part. The settling time in arbitrary multi axis slewing cases generally decreases with increasing magnitude of the real part, irrespective of the vector part. Notably, the times appear to increase asymptotically as the magnitude of the real component approaches 0. The target real components during the unstable trials 6 and 8 were 0.03858 and 0.04844 respectively. During these trials, the real component proceeded slowly to its target value, during which time, the vector components began oscillation, which was amplified to noticeable instability within 200s. The most likely cause of stability issues in this region is a coupling of the undriven real part with increased sensitivity of the driven vector part, since a real part near 0 leads to a high coefficient of the vector part (near 1). A real part with a magnitude around 0.15 or less, which corresponds to a rotation of greater than $\pm 162°$ about the quaternion axis, tends towards instability.

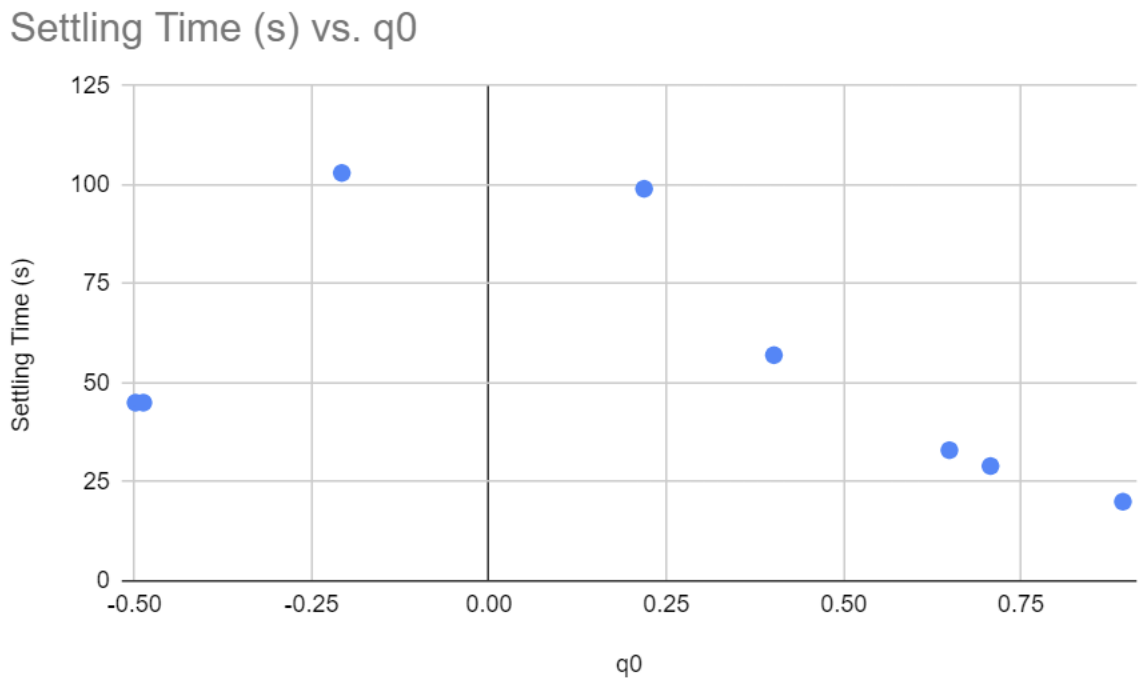

Figure 36: Settling time of the 8 stable trials against the real part of the target quaternion

The simplest way to mitigate this instability is likely to track two copies of the quaternion orientation. CaliPER's initial quaternion is arbitrarily defined as the unit quaternion, as is the convention in many aerospace applications, however, there is no need to define CaliPER's local frame, and the inertial Clipper frame as being initially aligned. It would be valid to track a second copy of CaliPER's attitude which transforms with the first but uses a different initial quaternion. With a carefully chosen second local frame, commands that would guide CaliPER to a quaternion with a small real part in the first local frame can be issued using the second frame. A reasonable choice of second frame may be one with initial quaternion $(0.5, 0.5, 0.5, 0.5)$ where the order of the x, y, and z axes is rotated as compared to the unit quaternion.

7.2 Maneuvering Controller: Overall

The verification of the maneuvering controller involved five maneuvering trials with random reference velocities in the range [0,0,0] to [6,6,6]. An upper bound of six in each component was chosen as this is the maximum velocity planned during the connected cross calibration trajectory. Target velocities were restricted to positive components only to decouple possible instability of the translation controller from the slewing controller by preventing cases where CaliPER would be guided to a low real part in its attitude. During each of the five trials, a random center of mass offset in the range of $\pm\ 1.5cm$ was applied in each direction to CaliPER's modelled center of mass. This range was chosen based on the typical center of mass uncertainty of 1-3cm as listed by Laron and Wertz [20]. IMU error error with variance 0.1° in attitude and $0.0098m/s^2$ in linear acceleration were used. Error values from [21] were used as a guide. Settling time and steady state error of trials were recorded and are summarized in the table

below. Settling time is defined by all components of the velocity reaching 95% of their final value. The controller often produced step responses with a knee, a point past which thruster remained active for some time, but at a much lower duty cycle than during the main rise. A 95% threshold often reasonably approximated this knee as shown in Figure 37.

	Trial	Settling Time (s)	Steady State Error (m/s)
	1	144	0.2364
	2	106	0.2133
	3	162	0.2395
	4	150	0.2212
	5	160	0.2682
Average		144.4	0.2357

Table 8: Maneuvering Controller Settling Times and Steady State Error Magnitudes

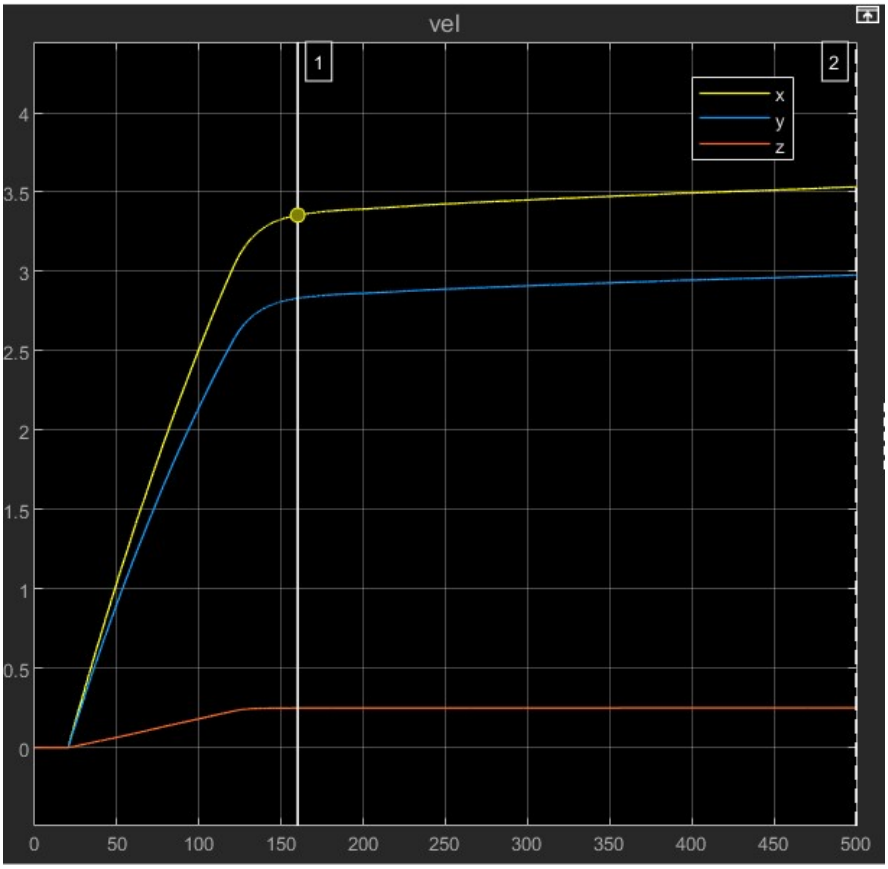

Figure 37: Trial 5 of Maneuvering with cursor placed at the settling time boundary. The boundary is just after the knee of the step responses.

The performance of the maneuvering controller against its design dependent specs is summarized below.

Designator	Spec	Value
PCM1	Maximum Valve PWM frequency	500Hz
PCM2	Maximum maneuvering stabilization time	500s
PCM3	Maximum Steady State Error	0.6m/s
PCM4	Maximum RW PWM frequency	50Hz

Table 9: Maneuvering Controller Design Dependent Specification Performance

PCM1 is enforced in the duty cycle conversion sub block of the velocity controller. The steady state errors of the trials fall well below the threshold set out in PCM3. Since there is no explicit duty cycle conversion in the modeling of the RWs, PCM4 is not verifiable in the simulation. The settling times to 95% of their final value were well below the 500s set out in PCM2; however, there are two important notes that hinder the consideration of this specification as verified. The first important consideration is that while the controller consistently reached the 95% threshold quickly, the thruster remained on at a low duty cycle for much longer, often throughout 500s simulation runs. It would be preferable, particularly during calibration, for thrusters to be off by the time of taking measurements, so even this low duty cycle state is somewhat undesirable for long periods.

The second note relates to modelling the IMU. Settling time and overall stability is informed largely by the accuracy of the IMU which had an attitude error variance of 0.1°, which is somewhat below what may be expected of the BNO055. Errors from this sensor are often in the range of 1-3°. More information about the sensor fusion algorithm is needed to determine how well the BNO055 could mitigate this discrepancy in a deployed setup. Take for example the zero-order hold used in modelling the sensor. Under a constant thrust this hold produces an acceleration that lags the true acceleration, and so when integrated the estimated velocity falls further behind the true velocity as time passes. Utilizing a first order hold, as well as an array of other feature that might be present in the BNO055 sensor fusion algorithm could help mitigate problems like these, but without manufacturer information reasoning deeply about its performance remains a challenge.

7.3 Desaturation Controller

The verification of the desaturation controller involved five trials with initial RW angular momenta of different signed-ness. The initial RW momenta were induced by applying a substantial external torque with components with magnitude 0.05Nm during the first 10s of simulation. The cubesat re-settles in its initial position after the 10s has passed. The quickly accumulated momentum brings the RWs to the point of requesting desaturation at which point the FSM directs the controller to activate. The controller used only proportional control with a coefficient $K_p = 0.5$. Slewing to repoint the Z-axis was initiated when the angular momentum of

the current Z-axis fell below 1mNms. Repointing was considered to be complete once the magnitude of the quaternion error fell below 0.01. The results of the trials are summarized below.

	Trial	Settling Time (s)	Steady State Error (Nms)
	1	191	0.006164
	2	198	0.005221
	3	147	0.002419
	4	215	0.002594
	5	211	0.002737
Average		192.4	0.003827

Table 10: Desaturation Controller Settling Times and Steady State Error Magnitudes

The performance of the desaturation controller against its design dependent specs is summarized below

Designator	Spec	Value
PCD1	Maximum desaturation time	120s
PCD2	Maximum Steady State Error	5% of capacity

Table 11: Desaturation Control Loop Design Dependent Specification Performance

The controller did not achieve the targeted desaturation time of 120s. Although the average steady state error fell below 5% of total capacity, the first two trials did show a steady state error above 5%. For this reason further verification is needed to consider this specification met.

Although the controller did not definitively meet either specification, it could still operate adequately for CaliPER's mission. The specification set out by PCD1 is somewhat strict when considering that desaturation requests would be made mostly during Phase 1 to dump momentum from detumbling (Table 12). During this phase the spacecraft remains drifting and available for desaturation for many hours before beginning to chase Clipper (see SECTION 10.1). During the most critical mission segment, Phase 3, when timing will be more heavily dictated by Clipper, desaturation might not be required assuming momentum is dumped on the order of 10s of hours before beginning this phase.

Item	Phase 1 (1 day)	Phase 2 (45 days)	Phase 3 (90 days)
Torque (mNm)	0.017	7.3E-6	4.4E-6
Momentum Storage (Nms)	1.48	0.028	0.037

Table 12: CaliPER Torque and Momentum Storage Requirements Per Phase [2]

If PCD1 were relaxed to approximately 500s, more time could be allotted for the desaturation of each axis, allowing the controller to more reliably meet PCD2. The trade between desaturation time and steady state error is illustrated in Figure 38. Momentum remaining after Z desaturation in ρ_z becomes $-\rho_y$ after repointing and continues to contribute to the steady state error throughout the test. Had each desaturation segment been elongated to allow for more thorough momentum dumping, the controller could more reliably meet the steady state error requirement.

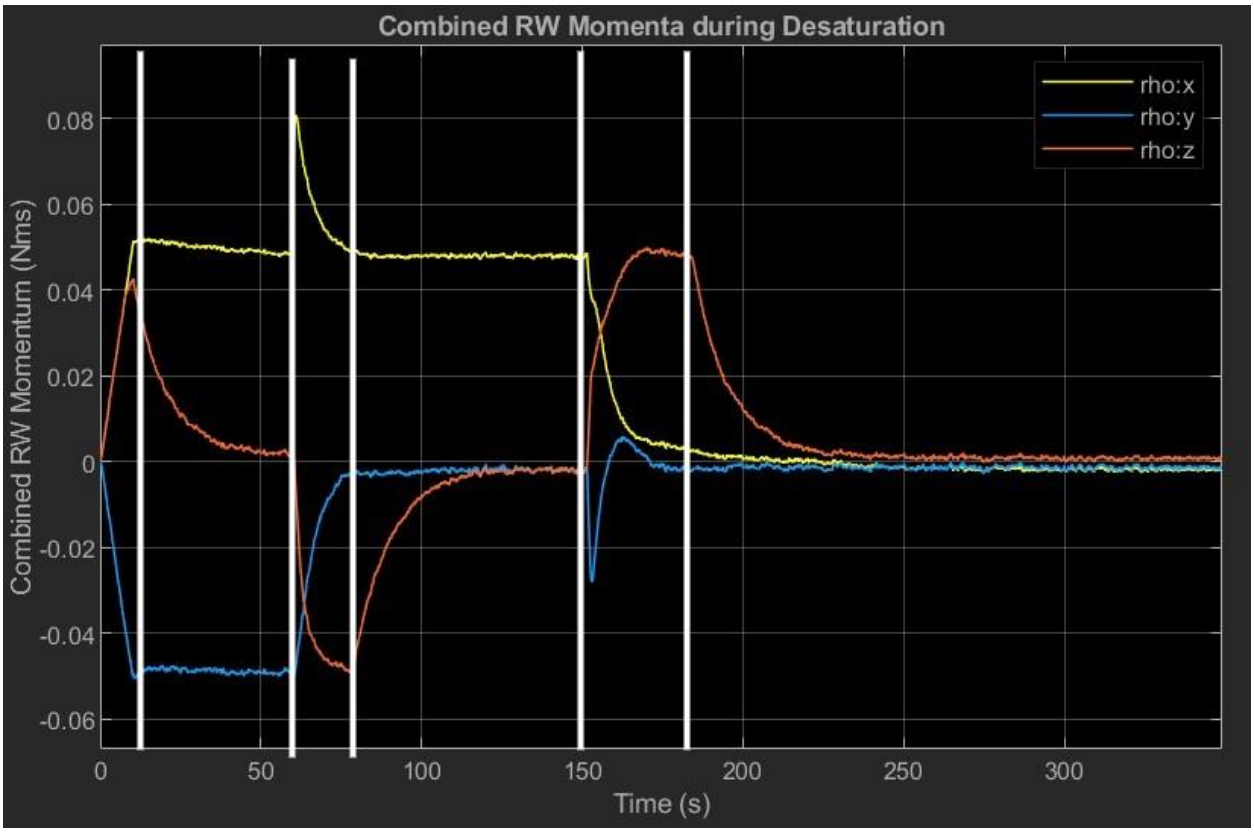

Figure 38: Desaturation Test with transitions between FSM states marked by vertical lines. The resulting sections of the graph represent 1) Application of forcing torque. 2) Z desaturation. 3) Repointing. 4) Y desaturation. 5) Repointing. 6) X Desaturation

7.4 Design Independent Specifications

The performance of the implemented controllers with regards to the design independent technical specification is outlined below.

Designator	Spec	Value
PC1	Minimum Distance from Clipper	75km
PC2	Maximum proportion of RW momentum capacity	75%

Table 13: Design Independent Specification Performance

PC1 was verified through the planned calibration trajectory's closest approach, which is 100km. Additionally, should CaliPER deviate from its planned trajectory and begin to approach Clipper with 0.6m/s per PCM3, the amount of distance travelled in a maneuvering period of 500s as per PCM2 is still just over 1% of the 25km buffer.

PC2 was verified in testing of the desaturation controller. The slewing component of desaturation generates only small, temporary excess momentum. 75% of the total RW capacity will not be exceeded so long as a call is made to the FSM to desaturate before this threshold. Driver circuitry was not fabricated, which leaves PC3-PC5 unverified.

8 Budget

A BOM is provided for the fabrication of the designed valve driver and RTD measurement circuits. Only the parts for the driver circuitry are represented here, as the costs of the valve and RTD themselves are represented in [3]. No additional materials beyond the compute platform and sensors tabulated in [2] would be required for prototyping the controllers on CaliPER.

Part	Unit Cost ($)	# of Units	Total ($)	Link	Estimated or Exact
NE555P	$0.46	1	$0.46	Digikey	Exact
1N4005	$0.20	3	$0.60	Digikey	Exact
1N4757	$0.28	1	$0.28	DigiKey	Exact
MPSA42	$0.41	2	$0.82	Digikey	Exact
MJH11022G	$8.19	1	$8.19	Digikey	Exact
MJH11021G	$4.85	1	$4.85	Mouser	Exact
Assorted 1/4W Resistors	$0.10	8	$0.80	Digikey - stackpole	Estimated
Assorted Capacitors	$0.23	3	$0.69	Digikey - Vishay	Estimated
TL081CP	$0.61	1	$0.61	Digikey	Exact
IRF9Z24PBF	$1.86	1	$1.86	Digikey	Exact
Total			$19.16		

9 Discussion and Conclusions

This project succeeded in developing controllers that interface CaliPER's propulsion and GNC architectures to the end of maneuvering the spacecraft and desaturating its reaction wheels. With regard to the performance of the controllers, through simulation the maneuvering controller met its steady state error requirement and shows promising results towards meeting its settling time requirement, although physical verification is needed to draw a conclusive verdict on the controller's interaction with the selected IMU. The desaturation controller did not definitively meet its settling time or steady state error requirements; however, the settling time requirement could be relaxed based on the mission timeline and demands. Doing so would allow the controller to meet both specifications. Designed valve driver and RTD sensing circuits were not fabricated, leaving their specifications unverified.

There are several possible avenues building upon the design and verification work outlined in this paper that could be taken to the end of refining these controllers. With regard to actuator simulation, simulation of thrusters could be refined with the introduction of an isentropic flow model. The current thruster simulation considers the thrust output at maximum duty cycle to remain constant throughout the duration of a burn, and between burns. During a single burn however, as propellant exits the chamber its pressure decreases, as well as its temperature by a small amount. This effect serves to decrease the thrust at a given duty cycle during a burn, as well as between burns assuming a constant preheat temperature. The isentropic flow model utilized by the propulsion team gives greater insight into the burn characteristics of the designed nozzle geometry as well as the time dependence of thrust force produced. This model is described by Hill and Martel [3] in Equations 2.10 to 2.20.

Introducing these dynamics to the model would add significant computation time, so expanding the simulation to include them would likely encourage some overall optimization of the code to maintain runtime on the order of minutes.

Simulation of the reaction wheels could potentially be improved with a more robust motor transfer function. The current actuator block assumes a linear mapping from duty cycle to speed. More advanced motor transfer functions convert the duty cycle to an applied voltage and utilize the inductive and resistive properties of the modelled motor to calculate output torques or speed. This refinement was impeded by limited information on the SURPASS motor, and the electronic speed controller that guides it. The dynamic imbalance data provided by Andrade, Collins, and Kim could additionally refine the model [2].

In a similar way the IMU block could be improved with a more detailed handling of errors. Although the BNO055 data sheet does not provide specific error tolerances, other information, like cross axis sensor error, could be used to refine the model. The BNO055 uses a sensor fusion algorithm. Although the equations for this algorithm are not provided, methodologies similar to those used by Hemerly could provide a starting point for a more sophisticated simulation [22].

In implementing a physical build, the conversion of controller code to embedded software would have to be addressed. The most complex calculations run in the simulation are likely solving the cubesat dynamics system of differential equations. Code for calculating control outputs is minimal to aid in the transition to embedded software; however, a careful design of system architecture would be necessary to implement controllers with sufficient response time on constrained hardware. The current controllers rely extensively on matrix multiplication for transformation and producing output vectors, as well as numerical integration and differentiation for recovering linear and angular velocities from IMU reported data. Software optimization on the chosen compute platform may be necessary for these functions to operate within the control loop at adequate speed to maintain requirements.

10 Appendices

10.1 Appendix A: Full Mission FSM

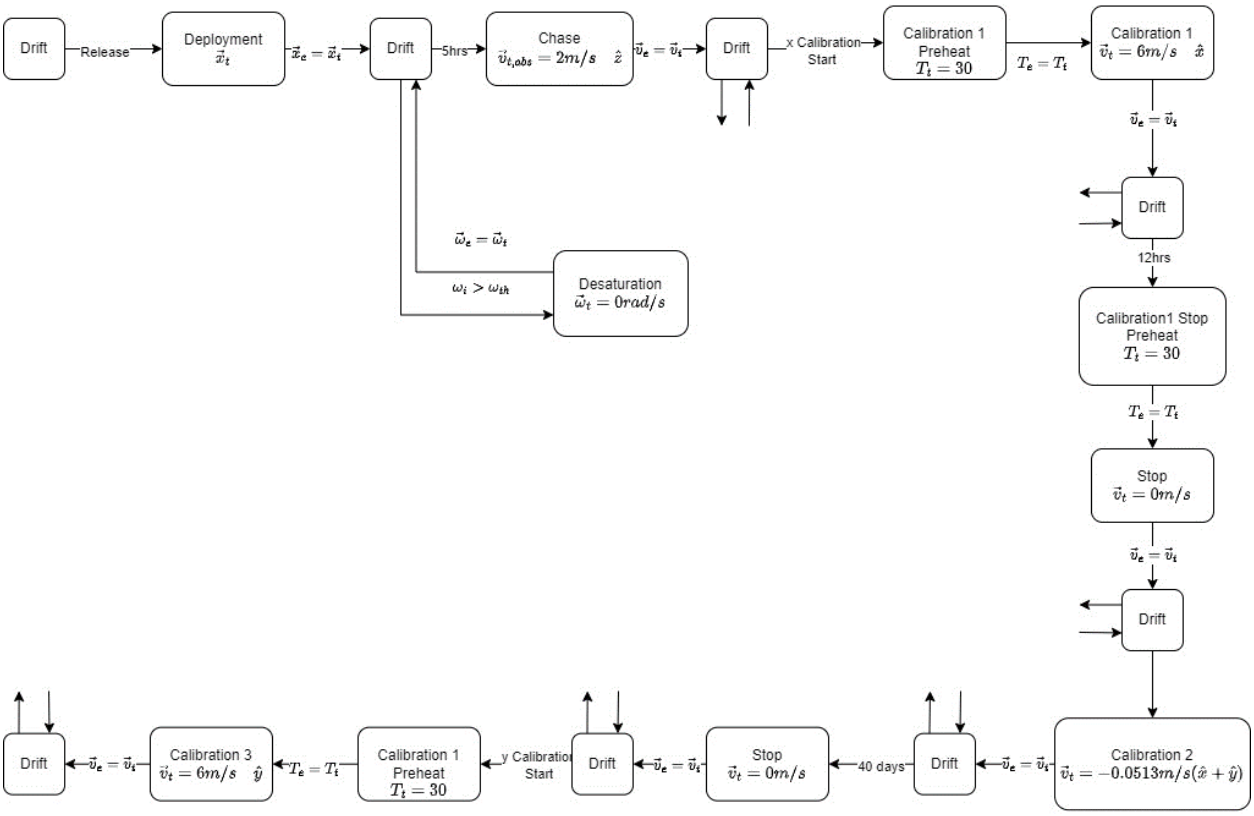

Figure 39: Detailed FSM State Diagram

Arrows into and out of the drift states represent possible transitions to the desaturation state if reaction wheel velocity surpasses the threshold. This is possible in any drift state apart from the state prior to deployment.

References

[1] Shekhtman, S. (2019, November 14). NASA Scientists Confirm Water Vapor on Europa [Text]. NASA. http://www.nasa.gov/feature/goddard/2019/nasa-scientists-confirm-water-vapor-on-europa

[2] Andrade, D., Collins, J., & Kim, G. (n.d.). Machining and Optimizing a GNC Subsystem for the CaliPER Calibration Satellite of Europa Clipper. Harvard University.

[3] Hill, J., & Martel, J. (n.d.). Design of the Calibration Trajectory and 3D-Printed Warm Gas Propulsion System for an Auxiliary CubeSat for the NASA JPL Europa Clipper Mission. Harvard University.

[4] McMullin, A. (2023). CubeSat Bus for CaliPER Mission in Support of Europa Clipper. Harvard University.

[5] Ayora, R. (2018). Implementation of a Modular 3D Rigid Body Motion Simulator [Masters of Science in Mechatronics, Aachen University of Applied Sciences, Centro de Ingeniería y Desarrollo Industrial]. https://cidesi.repositorioinstitucional.mx/jspui/bitstream/1024/339/1/M-RVA-2018.pdf

[6] PWM Solenoid Theory. (n.d.). TLX Technologies. Retrieved October 29, 2023, from https://www.tlxtech.com/solenoid-theory/pwm-solenoid-theory

[7] https://www.jpl.nasa.gov. (n.d.). Mars Cube One (MarCO). NASA Jet Propulsion Laboratory (JPL). Retrieved April 24, 2023, from https://www.jpl.nasa.gov/missions/mars-cube-one-marco

[8] MarCO (Mars Cube One). (2018, March 19). MarCO (Mars Cube One). https://www.eoportal.org/satellite-missions/marco#spacecraft

[9] Mars Cube One (MarCO). (2018). NASA. https://mars.nasa.gov/internal_resources/344/

[10] Schoolcraft, J., Klesh, A. T., & Werne, T. (2016, May 16). MarCO: Interplanetary Mission Development On a CubeSat Scale. SpaceOps 2016 Conference. SpaceOps 2016 Conference, Daejeon, Korea. https://doi.org/10.2514/6.2016-2491

[11] Klesh, A., Clement, B., Colley, C., Essmiller, J., Forgette, D., Krajewski, J., Marinan, A., & Martin-Mur, T. (n.d.). MarCO: Early Operations of the First CubeSats to Mars.

[12] https://www.jpl.nasa.gov. (n.d.). Lunar Flashlight. NASA Jet Propulsion Laboratory (JPL). Retrieved April 24, 2023, from https://www.jpl.nasa.gov/missions/lunar-flashlight

[13] Sternberg, D. C., Lai, P. C., Rizvi, A., Ortega, K. F., Lo, K. D., Adell, P. C., & Baker, J. D. (n.d.). Pre-Launch Testing of the Lunar Flashlight (LF) CubeSat GNC System.

[14] Petrov, P. (2015, March 6). Comparing the NE555 Timer and LM386 Amplifier as Inductorless DC-DC Converters. Electronic Design.

https://www.electronicdesign.com/technologies/analog/article/21800620/comparing-the-ne555-timer-and-lm386-amplifier-as-inductorless-dcdc-converters

[15] Impulse, D. (n.d.). High Speed In-Line Solenoid Valve. The Lee Co. Retrieved April 24, 2023, from https://www.theleeco.com/product/high-speed-in-line-solenoid-valves/

[16] The Lee Company—Lee Solenoid Valve Drive Circuit Schematics. (n.d.). Retrieved April 24, 2023, from http://67.199.46.28/engineering/electrical-engineering/lee-solenoid-valve-drive-circuit-schematics.cfm

[17] Martin-Mur, T. J., & Young, B. (2019). Navigating MarCO, the First Interplanetary CubeSats. PEER REVIEW.

[18] Safety Policy and Requirements For Payloads Using the Space Transportation System. (1989). National Aeronautics and Space Administration. https://snebulos.mit.edu/projects/reference/NASA-Generic/NSTS_1700-7B.pdf

[19] Larson, W., & Wertz, J. (n.d.). Attitude Determination and Control. In Space Mission Analysis and Design (Third). Kluwer Academic Publishers.

[20] Belabbas, B., Grosch, A., Heirich, O., Lehner, A., & Strang, T. (2013, January 1). Curvature Classification for Trains using Along-Track and Cross-Track Accelerometer and a Heading Rate Gyroscope.

[21] Hemerly, E. (n.d.). MEMS IMU Stochastic Error Modelling. Technological Institute of Aeronautics.