

Design of CubeSats for Formation Flying & for Extreme Low Earth Orbit

A Major Qualifying Project Report

Submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science

in Aerospace Engineering

by



Adam Koubeck



Colin Maki

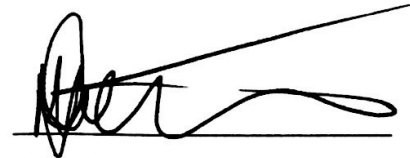


Matthew Sanchy



Tristram Winship

Approved by:



Professor Michael A. Demetriou, Advisor
Aerospace Engineering Program
Mechanical Engineering Department, WPI

Abstract

The purpose of this project is to design and analyze the attitude determination and control, command and data handling, and structural subsystems of CubeSats designed for two distinct missions. One mission was to fly a CubeSat in extreme low earth orbit, while the other was to fly two CubeSats in formation. Attitude sensors and an onboard computer selected in previous years' designs were reviewed and updated. Structural analysis simulations were conducted in ANSYS to ensure the CubeSats were capable of withstanding the launch environment. Additionally, the test bed for an attitude determination and control system experiment designed and constructed by the students in MAD-1701 was improved upon, and an attitude control system, including hardware and software coded in MATLAB, was designed. Finally, recommendations were provided for future teams looking to further develop this experimental setup.

Acknowledgements

We would like to thank the following individuals for their help and support throughout our project:

Project Advisor: Professor Demetriou

Mechanical Design, Thermal Analysis, and Environmental Effects Analysis Team: Professor Gatsonis and students Nicholas Bograd, Gregory Jacobson, Patrick Kroyak, Caitlin Lopez, Jackson Peters.

Power, Telecommunications, and Propulsion Team: Professor Blandino and students Matias Campos Abad, Robbie Crockett, Matthew Escalante-Hurtado, Alexander Kant, Marcus Knodler, Lucas Mancinelli, Jacob Nichols.

For help with test bed design and prototyping: Jack Agolli, Mark Landergan, James Loiselle, and Austin Scott.

Table of Authorship

Adam Koubeck - AK
Colin Maki - CM
Matthew Sanchy - MS
Tristram Winship - TW

Abstract	MS
1. Introduction	
1.1 CubeSat Background	AK
1.2 Our CubeSat Project	AK
1.3 Previous CubeSat Projects	AK
1.4 CubeSat Formation Flying Mission	CM, MS
1.5 CubeSat eLEO Mission	AK
2. Attitude Determination and Control	
2.1 Subsystem Overview	TW
2.2 Sensor and Actuator Selection	MS, TW
2.2.1 Fine Sun Sensor	MS, TW
2.2.2 Coarse Sun Sensor	MS, TW
2.2.3 Magnetometer	MS, TW
2.2.4 Gyroscope	MS, TW
2.2.5 Global Positioning System	MS
3. Command and Data Handling	
3.1 Onboard Computer Selection	AK
3.2 System Requirements and Sensor Usage	AK
3.3 Operational States	AK, CM
3.3.1 Deployment	AK, CM
3.3.2 Detumbling	AK
3.3.3 Initial Attitude Determination	AK
3.3.4 Routine	AK
3.3.4.1 Formation Flying Routine	CM
3.3.4.2 eLEO Routine	AK
3.3.5 Deactivation	AK
3.4 Modeling of States	AK
3.4.1 Block Diagrams	AK, CM

3.4.2 On/Off States	AK, CM
3.5 Modeling of Available Memory	AK, CM
3.5.1 Volatile Memory	AK, CM
3.5.2 Non-Volatile Memory	AK, CM
4. Structural Analysis	
4.1 Testing Requirements	MS, TW
4.2 Analysis in ANSYS	MS
4.2.1 Formation Flying CubeSat Results	MS
4.2.2 eLEO CubeSat Results	TW
5. Attitude Determination and Control Experiment Design	
5.1 Previous Design Flaws and Recommendations	MS, TW
5.2 Improvements to Test Bed Design	MS
5.2.1 Test Bed Plate Improvements	MS, TW
5.2.2 Attitude Determination and Control Components	MS, TW
5.2.3 CubeSat Printing Model	TW
5.2.4 Reaction Wheel Design	MS, TW
5.2.5 Test Bed Power Budget	AK, TW
5.2.6 PID Control for Reaction Wheels	MS
5.3 Future Recommendations	MS, TW
6. Conclusions	
6.1 Attitude Determination and Control Conclusions	TW
6.2 Command and Data Handling Conclusions	AK
6.3 Structural Analysis Conclusions	MS, TW
6.4 Attitude Determination and Control Experiment Design Conclusions	TW

Table of Contents

Abstract	0
Acknowledgements	1
Table of Authorship	2
Table of Contents	4
1. Introduction	6
1.1 CubeSat Background	6
1.2 Our CubeSat Project	6
1.3 Previous CubeSat Projects	6
1.4 CubeSat Formation Flying Mission	7
1.5 CubeSat eLEO Mission	7
2. Attitude Determination and Control	8
2.1 Subsystem Overview	8
2.2 Sensor and Actuator Selection	8
2.2.1 Fine Sun Sensor	9
2.2.2 Coarse Sun Sensor	10
2.2.3 Magnetometer	12
2.2.4 Gyroscope	13
2.2.5 Global Positioning System	15
3. Command and Data Handling	19
3.1 Onboard Computer Selection	20
3.2 System Requirements and Sensor Usage	21
3.3 Operational States	23
3.3.1 Deployment	24
3.3.2 Detumbling	24
3.3.3 Initial Attitude Determination	25
3.3.4 Routine	25
3.3.4.1 Formation Flying Routine	26
3.3.4.2 eLEO Routine	27
3.3.5 Deactivation	27
3.4 Modeling of States	27
3.4.1 Block Diagrams	27
	4

3.4.2 On/Off States	31
3.5 Modeling of Available Memory	35
3.5.1 Volatile Memory	35
3.5.2 Non-Volatile Memory	38
4. Structural Analysis	42
4.1 Testing Requirements	42
4.2 Analysis in ANSYS	43
4.2.1 Formation Flying CubeSat Results	44
4.2.2 eLEO CubeSat Results	54
5. Attitude Determination and Control Experiment Design	60
5.1 Previous Design Flaws and Recommendations	60
5.2 Improvements to Test Bed Design	62
5.2.1 Test Bed Plate Improvements	62
5.2.2 Attitude Determination and Control Components	66
5.2.3 CubeSat Printing Model	69
5.2.4 Reaction Wheel Design	71
5.2.5 Test Bed Power Budget	73
5.2.6 PID Control for Reaction Wheels	74
5.3 Future Recommendations	78
6. Conclusions	81
6.1 Attitude Determination and Control Conclusions	81
6.2 Command and Data Handling Conclusions	81
6.3 Structural Analysis Conclusions	82
6.4 Attitude Determination and Control Experiment Design Conclusions	83
Works Cited	85
Appendix A: Non-Volatile Memory Plots	89
Appendix B: Reaction Wheel Sizing	91
Appendix C: MATLAB Script for 3-Axis PID Controller	94

1. Introduction

1.1 CubeSat Background

A CubeSat is a particular class of nanosatellite developed in 1999 by Cal Poly San Luis Obispo to make a platform for education and space exploration. They have become a comparatively cheap way to perform scientific investigations and technological demonstrations in Earth's orbit [16]. The reason these nanosatellites are so efficient and cheap is because they use a standard size and form. Each CubeSat is made up of units, or "U's" that measure 10x10x10 centimeters. Traditionally CubeSats have been designed using 1U, 2U, and 3U configurations. This is so the CubeSat can be deployed using a Poly Picosatellite Orbital Deployer (P-POD) which has enough space for a standard 3U stack including an antenna. Recently, companies have been designing larger CubeSats of 6U and 12U to allow customers to design and deploy more powerful CubeSats.

1.2 Our CubeSat Project

The purpose of this Major Qualifying Project was to design two separate CubeSat projects. The first was a CubeSat capable of flying in formation with at least one other CubeSat, and the second was a CubeSat that can operate for 90 days at extreme low Earth orbit (eLEO).

1.3 Previous CubeSat Projects

At the beginning of this year's CubeSat project, we were recommended to look at previous CubeSat MQP's. Some of the primary sources used in this project were those the reports from the WPI CubeSat project in 2017, which was tasked to design a CubeSat to support a mission involving the Sphinx-NG instrument [3], [8], [14]. This instrument was designed to perform solar and X-ray spectroscopy while in Earth's orbit. Several of the design choices for our current missions were based on this previous MQP, because of similar CubeSat parameters and structure.

1.4 CubeSat Formation Flying Mission

The CubeSats are either a 16U configuration of 4x2x2 or a 20U configuration of 5x2x2. Each of the CubeSats has a “target” that keep it flying in formation. These “targets” are imaginary objects the CubeSats follow so they stay on course. The goal of having CubeSats fly in formation with each other is to see if multiple smaller satellites can satisfy the role of a larger satellite with increased replaceability and decreased cost.

1.5 CubeSat eLEO Mission

The goal of the eLEO Mission was to design a CubeSat that can be operated at extreme low Earth orbit (eLEO) altitudes, with the goal of having it operate at about 210 km above the Earth. This is new territory for CubeSats, as most operate much higher in orbit around the Earth. Orbiting this low to the Earth presents several challenges, the foremost being atmospheric drag, which is still present at the altitudes the CubeSat is planned to operate in, as well as atmospheric winds. The CubeSat was originally thought to be a 3U design, but this was changed to a 4U design during the preliminary design process to accommodate more fuel for the thrusters.

2. Attitude Determination and Control

2.1 Subsystem Overview

The attitude determination and control subsystem is needed to gather data regarding the positioning and orientation of the CubeSat and, when necessary, alter the position of the spacecraft. The three main occasions when attitude determination will be used is during detumbling, initial attitude determination, and attitude maintenance. Detumbling is bringing the spacecraft under control after it has been released from its P-POD or other launching vehicle, as when released the spacecraft has a tendency to begin rotating in a way that will make it nearly impossible to operate normally. Therefore, it is important to bring the spacecraft under control. Once that has been done, the spacecraft must determine its current attitude, and rotate itself into the position that is required by the mission. The last part is to maintain this position throughout the course of CubeSat operation, adjusting the attitude as needed using the chosen actuators. For this project we were not tasked with creating the code and the necessary data filters to provide the attitude controls for our respective CubeSat missions. The task for this fell to the graduate students for our respective missions, and we were only tasked with determining the sensors that were necessary through feedback from our graduate students.

To these ends, the orientation and position of the CubeSat will be determined by a variety of sensors outlined below and the actuators used to reorient the spacecraft will be an array of micro pulsed plasma thrusters (μ PPTs). Both CubeSat missions used the same sensors unless otherwise noted.

2.2 Sensor and Actuator Selection

The sensors used in this design were based off of the sensors used by the 2017 CubeSat MQP team for the design a 3U CubeSat [3], [8], [14]. These sensors were in turn based off of the design of the 2013 WPI CubeSat project [5]. Although these CubeSat missions featured different size CubeSats, both used similar sensors and actuators in their designs. The following sections

detail the selection of sensors and actuators used in these designs and technical specifications of components along with reasons for their selection.

2.2.1 Fine Sun Sensor

A fine sun sensor provides high precision pointing accuracy by detecting sunlight with a small margin of error. The fine sun sensor selected by the 2017 CubeSat mission was the New Space Systems CubeSat Sun Sensor, pictured in Figure 2.1 [3], [10]. Since the purpose of that mission was to take X-ray measurements of the sun, this sensor with a precision of 0.5 degree pointing accuracy was selected. Ultimately, it was decided that without a science payload in need of such high precision, neither mission required a fine sun sensor. Technical specifications for this device are shown in Table 2.1.

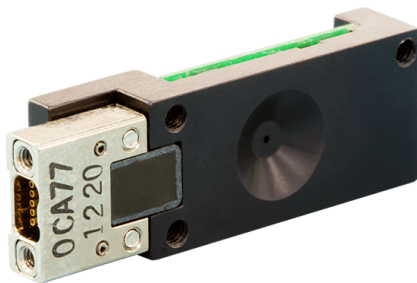


Figure 2.1. New Space Systems CubeSat Sun Sensor [10].

Table 2.1. Technical specifications of the New Space Systems CubeSat Sun Sensor [10].

Mass	<5g
Power	<10mA
Size	33x11x6mm
Field of View	114°
Accuracy	0.5°
Supply	5V
Connector	9 way female Nano-D
Price	3,300.00USD

2.2.2 Coarse Sun Sensor

Coarse sun sensors also allow for attitude determination based on the relative position of the sun. These come at the cost of lower precision, but with the benefit of lower power consumption. The coarse sun sensor of choice was determined by the 2013 CubeSat team through extensive research of technical specifications and prior CubeSat designs [5]. The CSS-01,02, pictured in Figure 2.2, was chosen for its simplicity and heritage [7]. The formation flying mission opted to use six coarse sun sensors, as the larger CubeSat was able to accommodate this greater number. For the eLEO mission, in which one end of the CubeSat is completely filled by the thruster, five coarse sun sensors were used.



Figure 2.2. CSS01,02 coarse sun sensor [7].

This sensor, developed by Space Micro, has been in use for over 20 years by a variety of spacecraft. A passive sensor, it requires no input power and produces a small current when in view of the sun. At a resolution of +/-5 degrees, this sensor is suited to aid in general reorientation of the CubeSat. With the implementation of a Kalman filter to process measurements and reduce sensor noise, attitude determination based on coarse sun sensor measurements become more reliable and sufficient for any pointing required by either mission. Technical specifications are provided in Table 2.2.

Table 2.2. CSS-01,02 technical specifications [7].

Mass	10g
Power	None (Passive)
Sizing	
- Housing Diameter	1.270cm
- Flange Diameter	2.286cm
- Sensor Height	0.899cm
Field of View	120° (One Axis)
Accuracy	5°
Connector	Flying Leads
Price	-

2.2.3 Magnetometer

Use of magnetometers allows the CubeSat to take readings of the Earth's magnetic field relative to its body-fixed axes. These readings are used to reorient the spacecraft. This information can be used in the programmed control laws as it provides insight into the orientation of the spacecraft.

Magnetometers used previously include the HMC5883L used in 2017 and earlier and the NAG-1102 prior to that [3], [5]. However, as the HMC5883L was discontinued, the LSM303, seen in Figure 2.3, was chosen as a replacement because it has specifications that are extremely similar to the HMC5883L, as shown in detail in Table 2.3 [28].

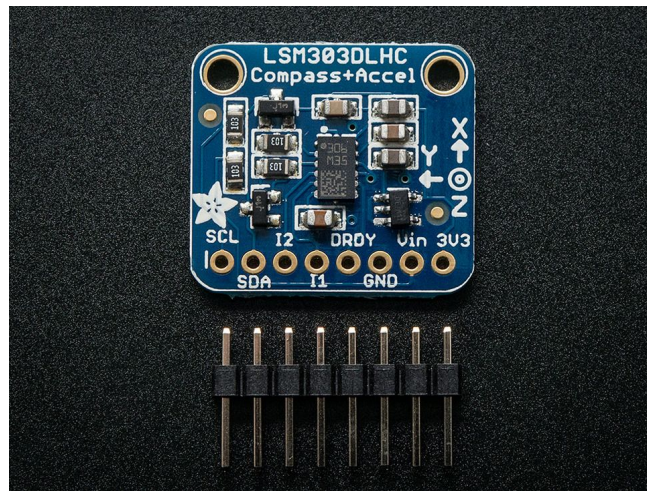


Figure 2.3. LSM303 Triple-axis Accelerometer and Magnetometer [28].

Table 2.3. Comparison of HMC5883L and LSM303 [3], [28].

Parameter	HMC5883L	LSM303
Mass	18mg	-
Power	2.16-3.6V	2.16-3.6V
Size	17.8x17.8x0.9mm	3x5x1mm
Linearity	+/-0.1%	-
Heading Accuracy	1-2°	-
Connector	Direct Interface	Direct Interface
Price	10.00USD	14.99

2.2.4 Gyroscope

Gyroscopes are necessary to obtain readings on the angular velocity and angular acceleration of the CubeSat. These readings are useful in maintaining, and adjusting the orientation of the spacecraft, as well as determining when the CubeSat is stable enough to begin operation.

The CubeSat teams in 2013 and 2017 recommended the ADXRS450 and the updated ADXRS453, respectively [3], [5], [12]. These are both high performance devices designed for platform stabilization, but the updated model was designed for use in high vibration environments, making it well suited to this design. The ADXRS453 came in two packages: one designed for measuring rotation about the x- or y- axis and one about the z-axis in the body-fixed frame. As measurements about all three axes were required for this mission, two of the former and one of the latter configuration was used in this design. Both packages are shown in Figure 2.4. These design considerations were carried out by the 2017 CubeSat team and remain applicable to this project.



Figure 2.4. Analog Devices ADXRS453 gyroscope [12].

The EVAL-ADXRS453Z breakout board was also required in order to condition the data gathered by these gyroscopes and allow for easy connection with the onboard computer. Specifications for the ADXRS453 are detailed in Table 2.4.

Table 2.4. Technical specifications of the ADXRS453 [12].

Max Measurement	+/-300°/sec - +/-400°/sec
Sensitivity	80 LSB
Bandwidth	77.5 Hz
Supply	3.15V to 5.25V
Connector	Direct Interface
Price	
- Sensor	48.24USD
- EVAL Board	70.00USD

2.2.5 Global Positioning System

A global positioning system (GPS) will also be implemented in this design. The GPS will be used to pull data from existing sun and magnetic field reference models to complement data from other sensors while also providing information on position and velocity. This is especially vital during the initial attitude determination, as the data from the sun sensors and magnetometers is compared to the data calculated from the GPS coordinates. This allows the CubeSat to adjust its attitude until the sensor data is nearly identical to the data calculated from the GPS coordinates, meaning the CubeSat is in line with its planned attitude.

The Surrey SGR-05U was the GPS receiver unit of choice for the 2013 CubeSat team and has seen continued use in the 2017 design [3], [5], [24]. This unit, pictured in Figure 2.5, was small, did not draw great amounts of power, and was designed with small satellite use in mind. The eLEO CubeSat team chose this unit as the GPS for their mission.

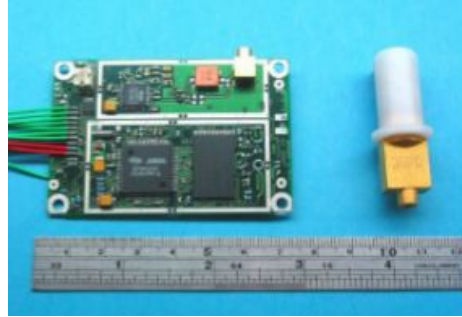


Figure 2.5. Surrey SGR-05U Space GPS Receiver [24].

For the formation flying mission, conducting the precise maneuvers requires a high degree of precision, so other sensors were considered. Due to the inherent constraints in designing a CubeSat, only low-mass, low-power units were considered. Many receivers were found with high position and velocity accuracy and short fix times. However, upon further research it was found that these receivers all had maximum altitudes of about 18,000 m and maximum velocities of about 500 m/s, both of which are far too low for this mission. Two GPS receivers were found that met the mission requirements: SkyFox Labs piNAV-NG, shown in Figure 2.6, and SkyFox Labs pi-NAV-L1, shown in Figure 2.7, [25] [26]. While no record of flight heritage could be found for these models, their specifications better fit the mission when compared to the Surrey SGR-05U.



Figure 2.6. SkyFox Labs piNAV-NG [25].



Figure 2.7. SkyFox Labs piNAV-L1 [26].

The technical specifications for these devices are given in Table 2.5. All three sensors were comparable in dimensions and mass. The two SkyFox Labs receivers drew much less power and were slightly more precise in velocity measurements than the SGR-05U. The piNAV-L1 was eliminated as an option because of its high relative mass (about twice that of each of the other receivers) and its long fix time. The piNAV-NG was finally chosen to replace the SGR-05U, as it was decided that the decrease in power consumption, increase in accuracy, drastic decrease in price, and faster fix time more than make up for its slightly larger size and lack of flight heritage.

Table 2.5. Comparison of GPS technical specifications [24], [25], [26].

Parameter	SGR-05U	piNAV-NG	piNAV-L1
Mass	20g	24g	47g
Power	0.8W	.125W	.120W
Size	70x45x10mm	71.1x45.7x11mm	75x35x12.5mm
Position Accuracy	10ms	10m	10m
Velocity Accuracy	0.15m/s	0.10m/s	0.10m/s
First Fix	180sec (Cold), 90sec (Warm)	90sec (Cold)	5min (typical)
Connector	Serial	2 mm, 2×10 pin header	2.54 mm, 2×10 pin header
Price	26,300.00USD	6,900.00EUR	5,900.00EUR

3. Command and Data Handling

The command and data handling (C&DH) subsystem has control over all the other subsystems. The onboard computer (OBC) tells the other subsystems what to do and when to do it based on the mission requirements. In this way, the OBC is the brain of the CubeSat, while the C&DH system is the nervous system that transmits the signals from the OBC to the sensors of the satellite. The C&DH subsystem also collects data from all of the CubeSat's sensors and subsystems. This means the C&DH subsystem must be able to handle the high computational and data storage needs of the CubeSat. The first step for this project was to determine the OBC that we would use for the mission. Once this was decided, we then selected the sensors that would be used by the CubeSat to complete its mission.

Once we had the sensors and the OBC finalized, we set out to estimate the amount of memory that would be used by each of the sensors, as well as the memory that would be used by the OBC and the electronic power subsystem (EPS) would not exceed the capacity of the OBC. To do this we clarified the data being stored in the OBC as either volatile or non-volatile memory. Volatile memory was data that would be used by the OBC, and then immediately overwritten by more current data. This, in effect, was data and information coming from the sensors of the CubeSat, because there was no need to store the past data of the sensors when only the most current data would be used to complete a task such as aligning the satellite's x-axis with its velocity vector. The non-volatile memory was the exact opposite, in that all of the data sent to the OBC's memory would be stored there until being transferred back to the ground. This type of memory would be used by the OBC, EPS, and whatever cargo sensor was put inside the CubeSat for a given mission. For both of these types of memories, we needed to model the accumulation of memory and make sure that there was enough storage in the OBC to account for both the volatile and non-volatile memory.

Finally, we identified the key operational states that the CubeSat would go through during its mission and stated which sensors and subsystems would be active during each state. Using this information, we created a block diagram of the actions and queries for each phase to show what the OBC would be controlling throughout the mission during each operational state.

This was then combined with the memory estimates of the sensors to create a model of the memory used for each operational stage of the CubeSat. All of these steps were done for both the formation flying and the eLEO CubeSat.

3.1 Onboard Computer Selection

The OBC selection was heavily based off of the previous 2017 CubeSat MQP, and because of this we ended up choosing the Clyde Space 01-02928 as the OBC for both CubeSat missions [14] [18]. It contained 8 MB of MRAM storage, which was used for the volatile memory, and 4 GB of flash memory storage, which was used for the non-volatile memory. Since this OBC had already been vetted by a previous MQP, we were confident in its reliability, and we were confident in its ability to store all the required memory because it had easily done so for the previous MQP.

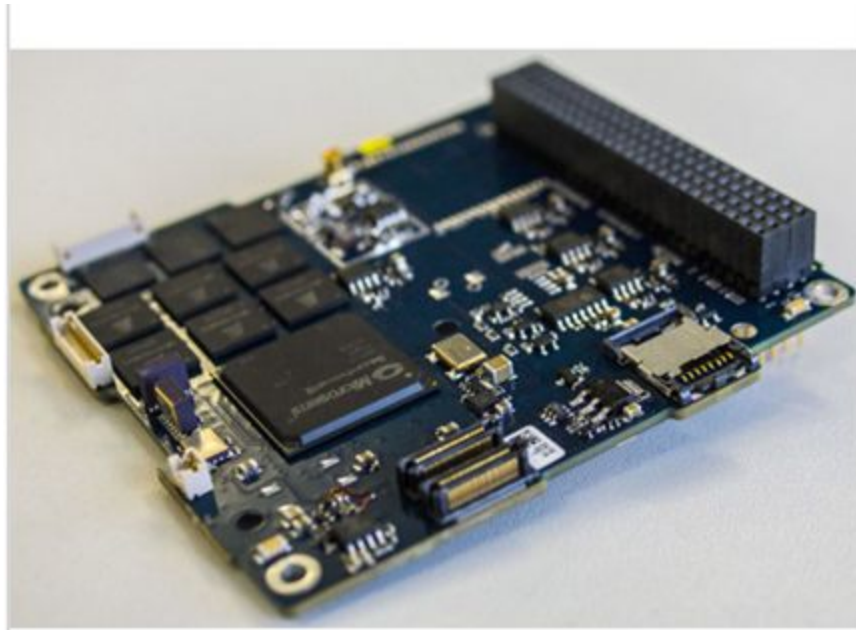


Figure 3.1. Clyde Space 01-02928 [18].

Table 3.1. Clyde Space 01-02928 technical specifications [18].

Processing Speed	50 MHz delivering 62.5 DMIPS
Memory Storage	8 MB MRAM, 4 GB Flash
Operating Temperature	-40 to 80 deg C
Power Used	400 mW typical, 1 W max

3.2 System Requirements and Sensor Usage

Once we had selected the OBC that we would be using, we then went about estimating how much memory would be used by each of the sensors, as well as the OBC and EPS. Once we found this information, we could model the memory usage to confirm if there was enough flash and MRAM memory in the Clyde Space OBC. However, the data usage of each sensor was not readily available, so we were required to use estimation. Thankfully, the previous Sphinx-NG MQP had already estimated the memory usage of the sensors they had used, and since we were using those same sensors and OBC and EPS, we were able to use their estimations for our CubeSat C&DH.

As previously mentioned, the majority of the sensors on the CubeSat would use the MRAM memory of the OBC, with the memory being constantly overwritten as new data is transmitted from the sensors. The only sensor that did not give back data to the OBC was the μ PPT's, which only operated from commands sent by the OBC.

Table 3.2. OBC Memory Storage [14].

Storage Type	Total Storage
MRAM (volatile)	8 MB
Flash (non-volatile)	4 GB

Table 3.3. Volatile Memory for eLEO CubeSat [14].

Sensor	Number on Satellite	Memory Used per Sensor	Total Memory Used
Gyroscope	1	100 KB	100 KB
Magnetometer	1	100 KB	100 KB
GPS	1	200 KB	200 KB
Coarse Sun Sensor	5	20 KB	100 KB

Table 3.4. Volatile Memory for Formation CubeSat [14].

Sensor	Number on Satellite	Memory Used per Sensor	Total Memory Used
Gyroscope	1	100 KB	100 KB
Magnetometer	1	100 KB	100 KB
GPS	1	200 KB	200 KB
Coarse Sun Sensor	6	20 KB	120 KB

Table 3.5. Non-Volatile Memory for CubeSat [14].

Sensor	Memory Used / sec
OBC	0.20 KB/s
EPS	0.05 KB/s

3.3 Operational States

During every satellite mission there are a sequence of events that the satellite must adhere to for it to accomplish its mission. For both missions, we identified the main operational states for our CubeSat and created block diagrams for each of them to show how the sensors and the OBC would interact during the operational states. The states take the satellites from deployment to deactivation, as shown in Figure 3.2. Each of the states is described in more detail later in the chapter.



Figure 3.2. Operational States of CubeSat.

The longest operational state of the mission is routine, which is where the majority of the mission will be spent, and when the CubeSat is orbiting around the Earth. For the eLEO mission this will last for the 90 days the mission is planned to last, at which point the satellite will receive a deactivation signal and enter the deactivation state. The formation flying mission has no set duration and will last as long as it can.

3.3.1 Deployment

The deployment state is the initial phase of the CubeSats operation. For the eLEO mission, deployment of the satellite will be done using a P-POD, which is the standard delivery method for CubeSat's of 4U and less. The formation flying CubeSat will require a larger launch vehicle due to its size. The purpose of the deployment state is to make sure that the CubeSat has actually been launched and is a safe distance away from the launch vehicle before the sensors of the satellite are activated and detumbling begins. To do this, the state begins with the OBC and EPS being turned on, either remotely or manually before launch. There is also a switch on the outside of the CubeSat that is pressed when the CubeSat is placed within the launch vehicle. Once this switch is depressed, a timer counts down from 30 minutes, and once the timer reaches 0, the other sensors are activated, and the mission continues to the next phase. The reason for the 30 minute timer is to make sure that if the switch is temporarily depressed while inside the launch vehicle, the satellite does not automatically activate. Furthermore, the 30 minute time limit was approved by our advisors as being an adequate amount of time for the CubeSat to travel a safe distance from the launch vehicle after being released. The detailed block diagram for this state can be found below.

3.3.2 Detumbling

When a CubeSat is released from a launch vehicle, the slightest force will send it tumbling endlessly because of the absence of friction in outer space. To stop this from happening, the CubeSat must detumble, or reduce its rotation to the point where it will be able to continue with its mission. This is the most critical part of the mission, because if the detumbling state is not successful, then the CubeSat will continue tumbling indefinitely, and not be able to complete its mission.

When the CubeSat first enters the detumbling phase, it gathers angular velocity data through its gyroscope and magnetometer. This includes the direction and magnitude of the angular velocity, which is then fed through the OBC to the ADC system, which calculates which μ PPT's need to fire and at what force. The OBC then sends this information to the μ PPT's,

which fire as instructed. This cycle then repeats until the CubeSat's angular velocity has been reduced to a stable rate. Following consultation with our advisors and past research, we concluded that this stable rate should be considered 0.1 rad/s for our CubeSats. Once the angular velocity has been reduced to 0.1 rad/s, the antenna is deployed, and the initial attitude determination begins.

3.3.3 Initial Attitude Determination

Now that the CubeSats are in a stable position in orbit around the Earth, the OBC must direct the μ PPT's to adjust the attitude until it is in the optimal position to carry out its mission. The optimal position for both CubeSat missions is to have the x-axis facing the velocity vector. This was chosen because by facing the velocity vector, the atmospheric drag that is still present will be cut down, thus reducing power consumption by the main thruster to keep the CubeSat in orbit. Furthermore, the reduction in drag will make it less likely the CubeSat will be subjected to forces that may cause it to begin tumbling again, which would need to be corrected before the mission continues.

The GPS within the CubeSat has the latitude, longitude, and altitude of the satellite, and from there it can use the Magnetic Field Model and Astronomical Almanac to determine the expected magnetic field vector and the expected sun vector the CubeSat sensors should be registering for the x-axis to be aligned with the velocity vector. These expected vectors are then compared by the OBC to the actual magnetic field and sun vectors from the coarse sun sensors and magnetometer. If the vectors are different, then the OBC uses the ADC system to calculate which μ PPT's should fire at what strength to align the expected and actual vectors. This cycle continues until the vectors are the same, meaning the x-axis is aligned with the velocity vector. Once this occurs the mission progresses to the routine state.

3.3.4 Routine

The majority of both missions will be spent in the routine state, which is where the CubeSat will perform whatever it has been sent up to orbit to do. When defining how this operational state would work, we realized that the most critical part of the state would be keeping

the CubeSat in the correct attitude to stay in its orbit. Because of this, the routine state is designed to operate similar to a continually-running initial attitude determination state, with the sun sensors, magnetometer, gyroscope and GPS sending data to the OBC and the OBC using this data to determine if the CubeSat is in its correct attitude. The correct attitude is the same as for the initial attitude determination; when the x-axis of the CubeSat is aligned with its velocity vector.

When we originally designed the routine state, we divided it into two separate parts; one for when the CubeSat was in sunlight, and the other for when the CubeSat was in darkness. The thinking was that by using the sun sensors to determine when the CubeSat was in sunlight or not, certain sensors could be shut off when the satellite was in darkness, thus conserving power until more could be created once the CubeSat entered the sunlight again. We created a sun sensor operational state to go with this that would determine when the OBC would switch from the sun routine state to the dark routine state and vice versa. Attitude adjustment would happen during the sun routine, but not during the dark routine as the GPS, gyroscope, and magnetometer would be shut off. However, during our review of the routine states, we realized that we needed to have all the sensors on throughout the CubeSat mission to be able to instantly adjust attitude if needed. Turning off sensors while in the dark would create not unnecessary risk of the CubeSat going off course because of drag or some other external forces. Furthermore, the Power Group reported to us that turning off the GPS, gyroscope, and magnetometer would not create substantial power savings, thus making the risk of turning them off during the dark routine not worth taking.

After this feedback, we adjusted the routine state so that all sensors are on at all times, and constantly checking the attitude and altitude of the CubeSat and adjusting them if need be to keep the satellite in orbit.

3.3.4.1 Formation Flying Routine

For the formation flying mission the CubeSats had to be in the correct position relative to each other at all times. This required constant checking of sensors to gather positional data similar to how attitude was determined. The CubeSats would then use this positional data to fire thrusters to correct any variances in position.

3.3.4.2 eLEO Routine

For the eLEO mission, the low orbit presented challenges that were normally not encountered for satellite missions because of the presence of drag and slight winds that could cause the CubeSat to go off course. Because the satellite is operating at such a low altitude, it was absolutely critical that it not deviate from the optimal path, unless it would reenter the atmosphere and burn up.

3.3.5 Deactivation

The final stage of the CubeSat operation will be deactivation, which is when the OBC shuts off all sensors except the antennae, OBC and EPS and allows the satellite's orbit to decay until it reenters and burns up in the atmosphere. This will continue unless the CubeSat receives a reactivation command, in which it will turn on all of its sensors again and resume the routine state to continue operation. Usually CubeSats do not survive long enough to be deactivated, but we believed that it was important to visualize all aspects of the mission and be prepared for every eventually that may occur.

3.4 Modeling of States

3.4.1 Block Diagrams

Once we had determined the operational states of the CubeSat, we set out to create block diagrams to visualize how the OBC would interact with all of the sensors. The figure below shows the key of what each shape and color represent in the block diagrams. The blue rectangles represent a sensor being used, like a sun sensor creating a sun vector. The yellow ovals indicate a change in state where the OBC ends one operational state and begins another. Input Vectors, such as the sun vectors created by the sun sensors, are indicated by orange arrows, while gray trapezoids indicate an action taking place as a result of inputs. An example of an action would be turning on the sensors of the CubeSat after deployment. Lastly, queries are indicated by green hexagons, and are yes/no questions asked by the OBC to determine its next action. The most

common query throughout the operational states was “Is the x-axis aligned with the velocity vector?” which would determine if the attitude would need to be adjusted.

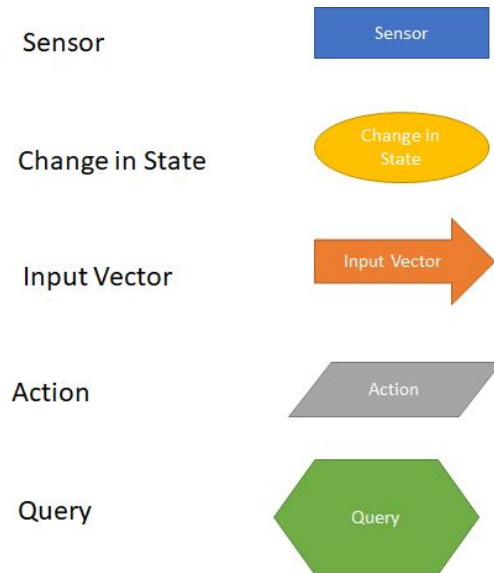


Figure 3.3. Key for Block Diagrams.

The block diagrams for the operational states are meant to be overviews of what the OBC will do during the CubeSat mission. Because this project did not involve any actual coding of the CubeSat and its programs, we tried to focus on the big picture of the general actions and interactions of the OBC to have the CubeSat complete its mission. The following figures show the block diagrams of each of the operational states.

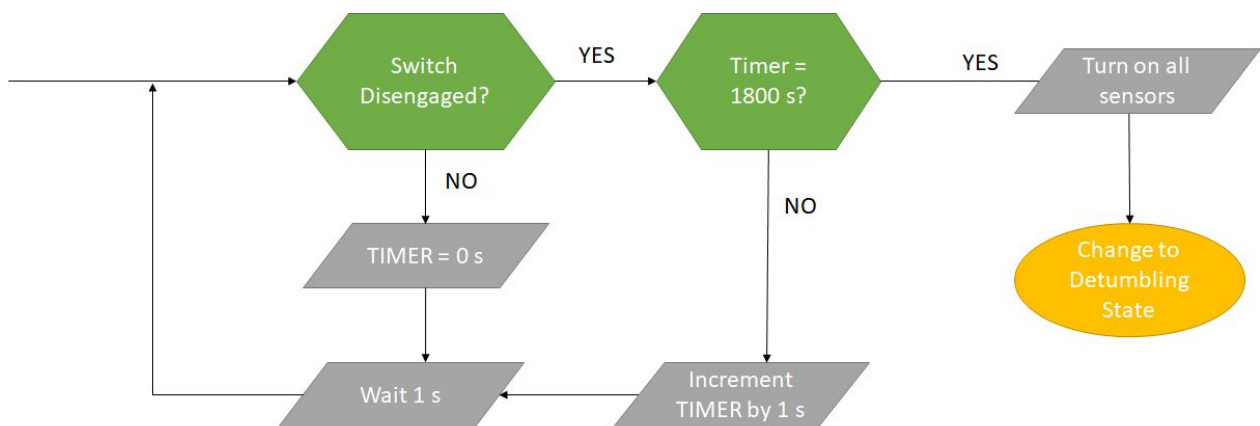


Figure 3.4. Deployment Block Diagram (both missions).

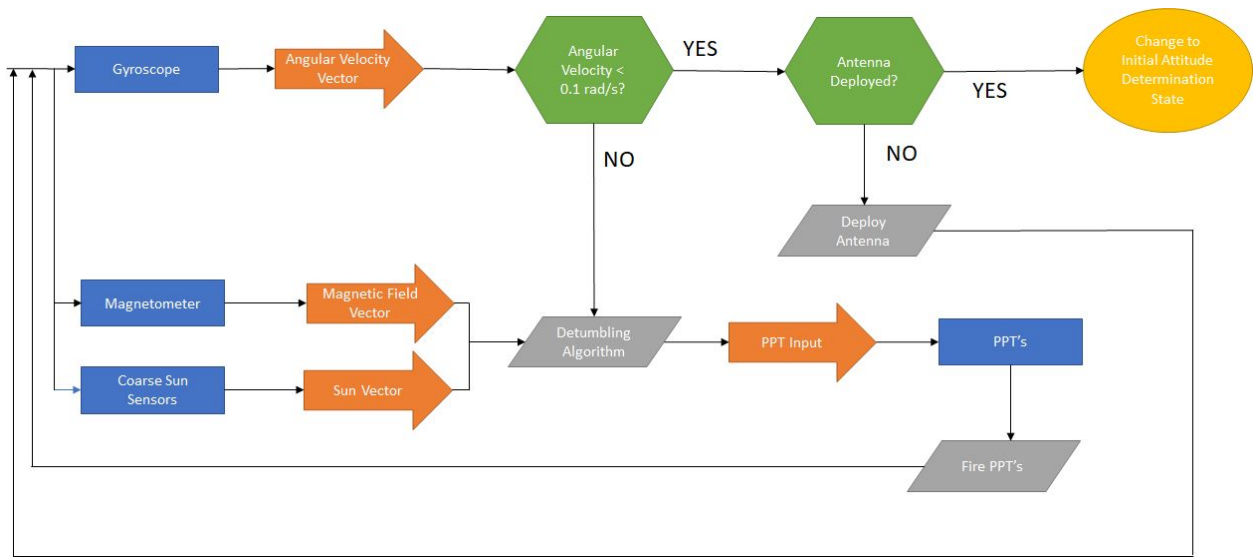


Figure 3.5. Detumbling Block Diagram (both missions).

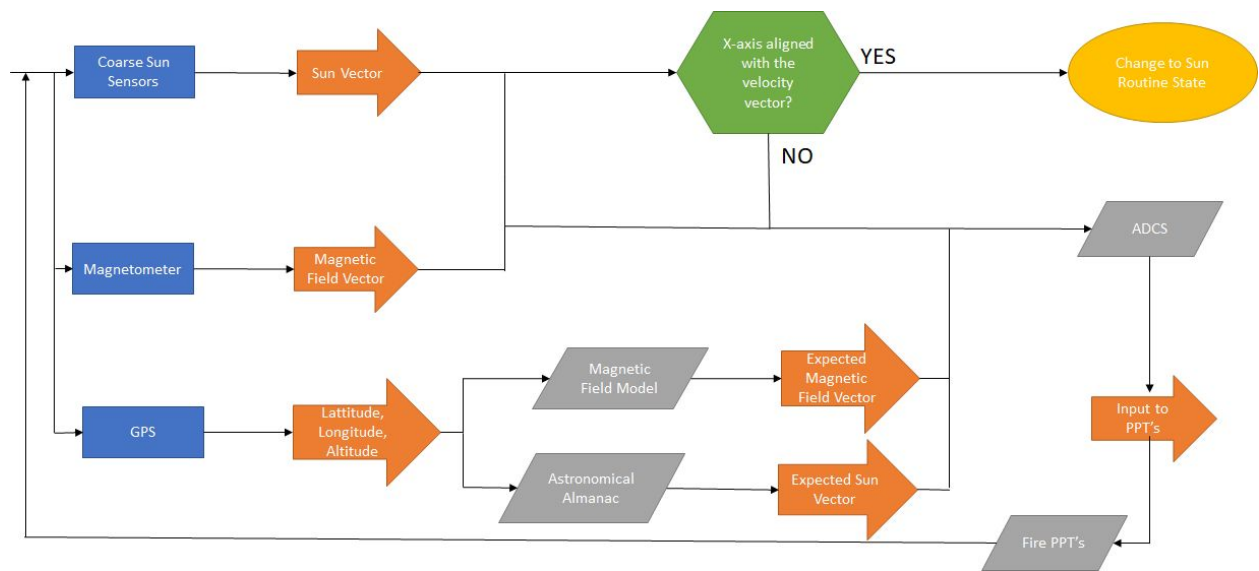


Figure 3.6. Initial Attitude Determination Block Diagram (both missions).

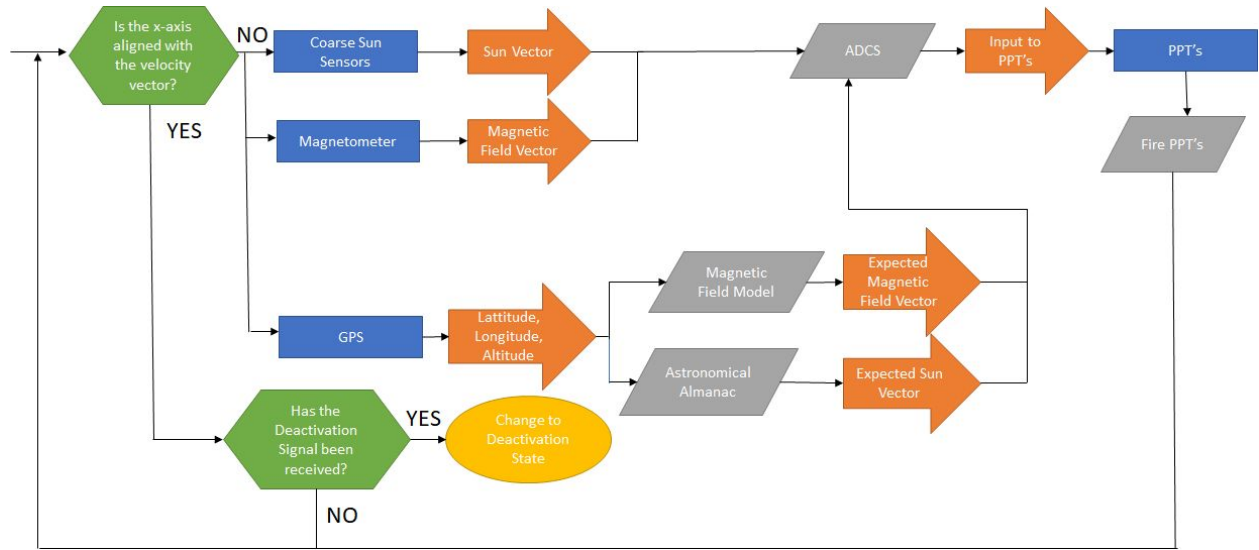


Figure 3.7. eLEO Routine Block Diagram.

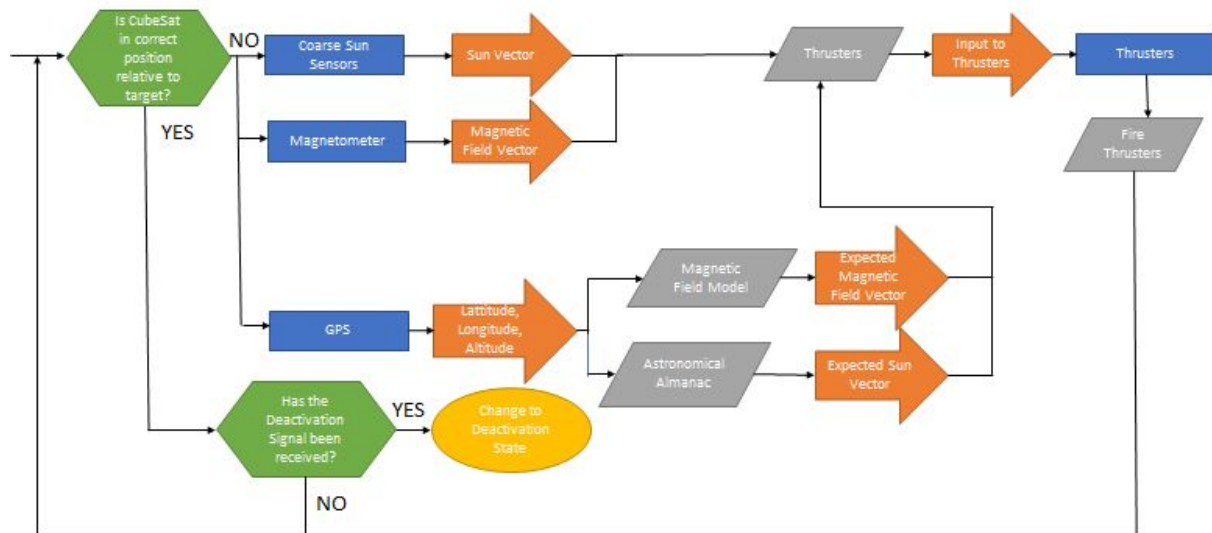


Figure 3.8. Formation Flying Thruster Routine Block Diagram.

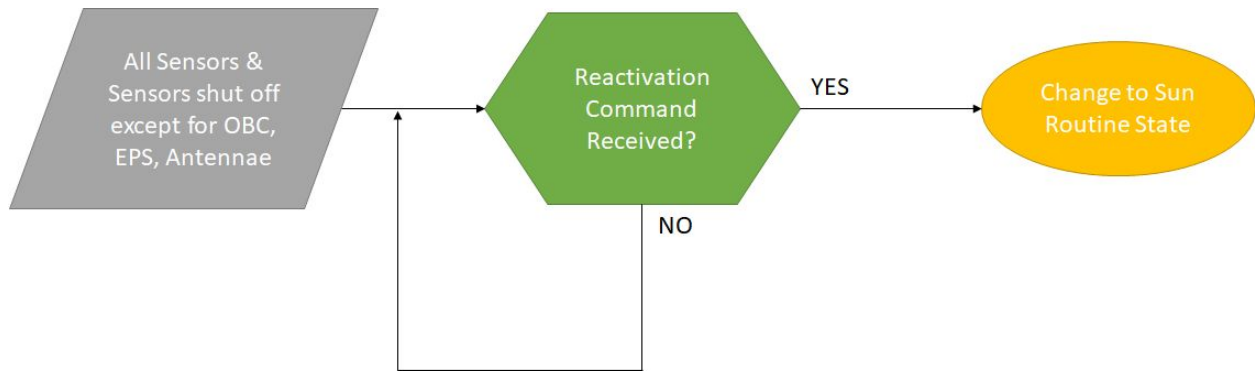


Figure 3.9. Deactivation Block Diagram (both missions).

3.4.2 On/Off States

From the block diagrams that we created, we were then able to determine when each sensor would be turned on throughout the mission. We first created a table showing if the sensors were on or off during each operational state and used this information to create plots showing the on/off states of each sensor. The OBC and EPS are on throughout the entire mission, as they are mission-critical to keep the CubeSat in orbit. The gyroscope, GPS, magnetometer, sun sensors and μ PPTs on the other hand, are turned off during the deployment state, after which they are turned on and stay on throughout the mission until the deactivation State. The “off” state refers to the sensor being completely shut off, while the “on” sensor refers to the sensor being active and ready to operate, not necessarily that it is collecting or analyzing data every second of a stage. For example, while the μ PPTs are on throughout the routine state, they are most likely not going to be firing all the time, only when the attitude needs to be adjusted. And even then, not all μ PPTs will fire so as to conserve fuel and to simplify the attitude adjustment process.

Table 3.6. On/Off States of CubeSat Sensors.

Sensors	Deployment	Detumbling	Initial Attitude Determination	Sun Routine	Dark Routine	Deactivation
OBC	ON	ON	ON	ON	ON	ON
EPS	ON	ON	ON	ON	ON	ON
Gyroscope	OFF	ON	ON	ON	ON	OFF
Magnetometer	OFF	ON	ON	ON	ON	OFF
GPS	OFF	ON	ON	ON	ON	OFF
Sun Sensors	OFF	ON	ON	ON	ON	OFF
μ PPTs	OFF	ON	ON	ON	ON	OFF

The plots and table have the routine state split into two - the sun routine and dark routine. This is a byproduct of when we planned to have two separate states for routine operation, but we kept them in after we had changed the routine state to show that all sensors are turned on when the CubeSat is in the sunlight and in the darkness. Each plot is divided into each of the operational states to easily show during which states the sensors are turned on.

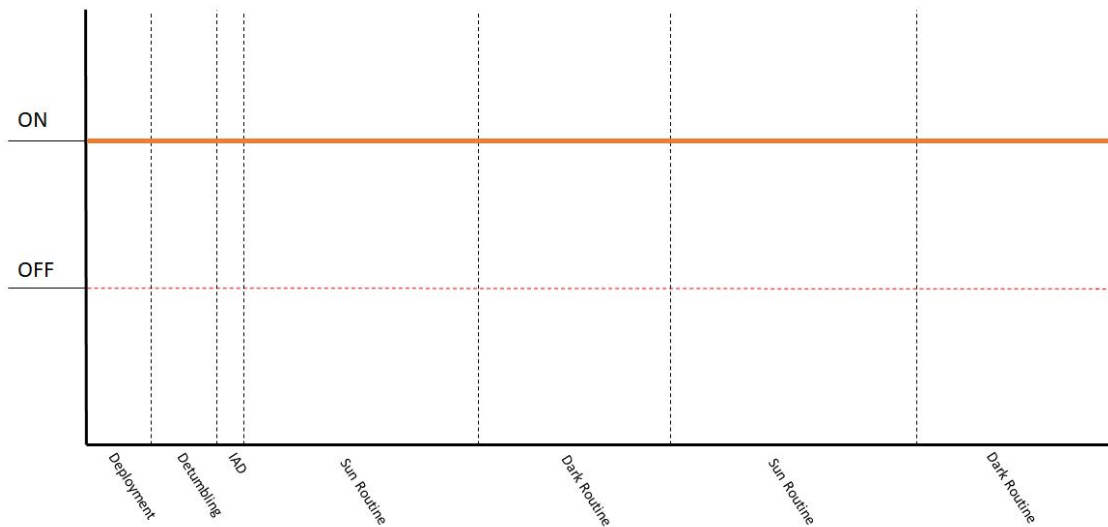


Figure 3.10. OBC State Plot.

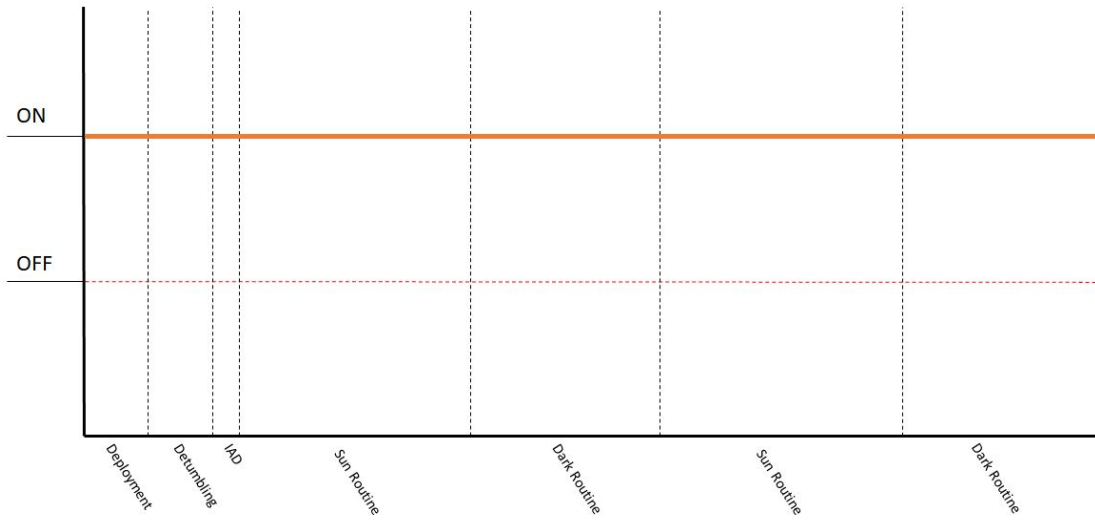


Figure 3.11. EPS State Plot.

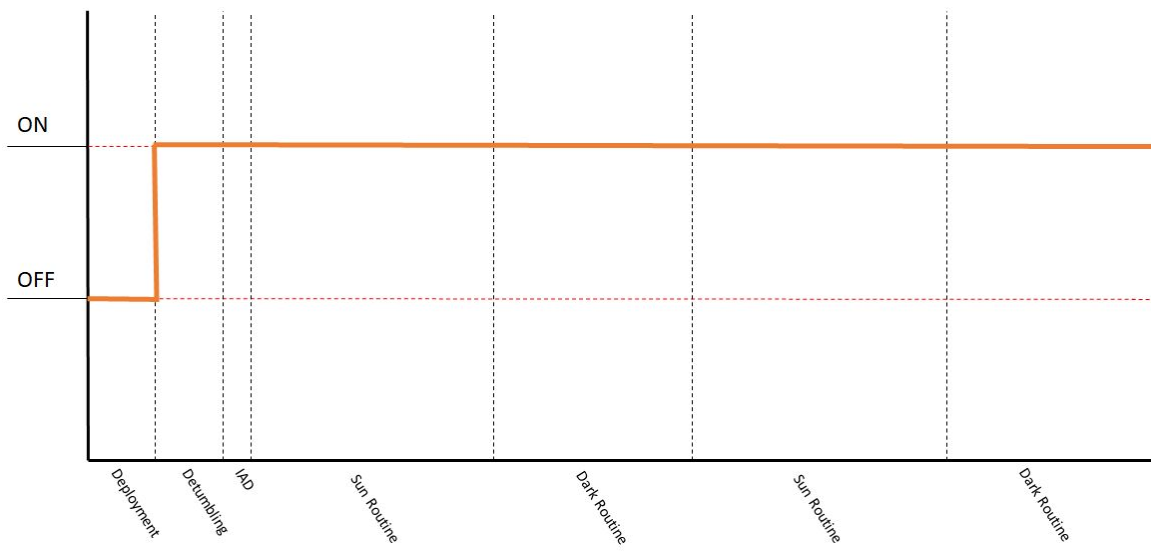


Figure 3.12. Coarse Sun Sensor State Plot.

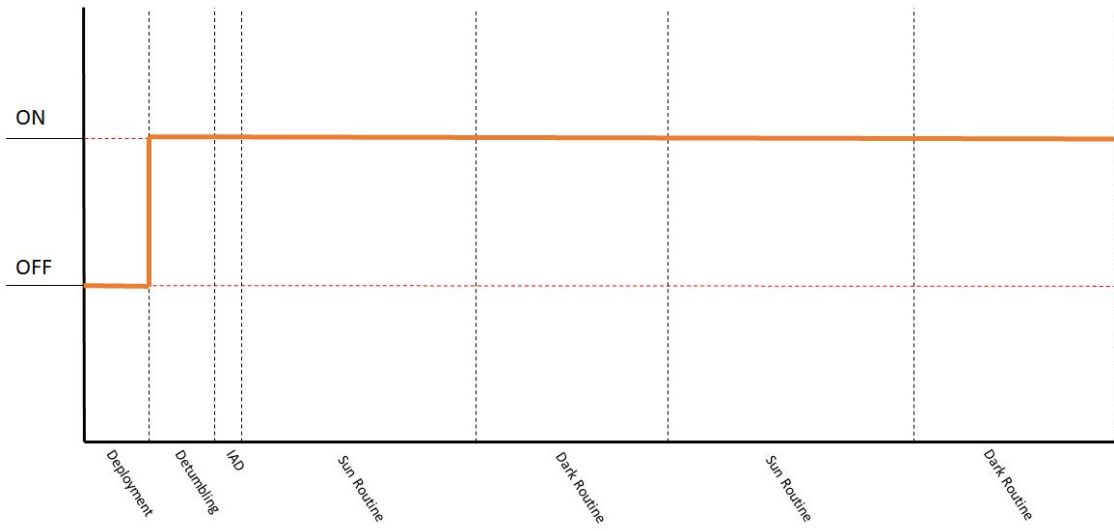


Figure 3.13. Gyroscope State Plot.

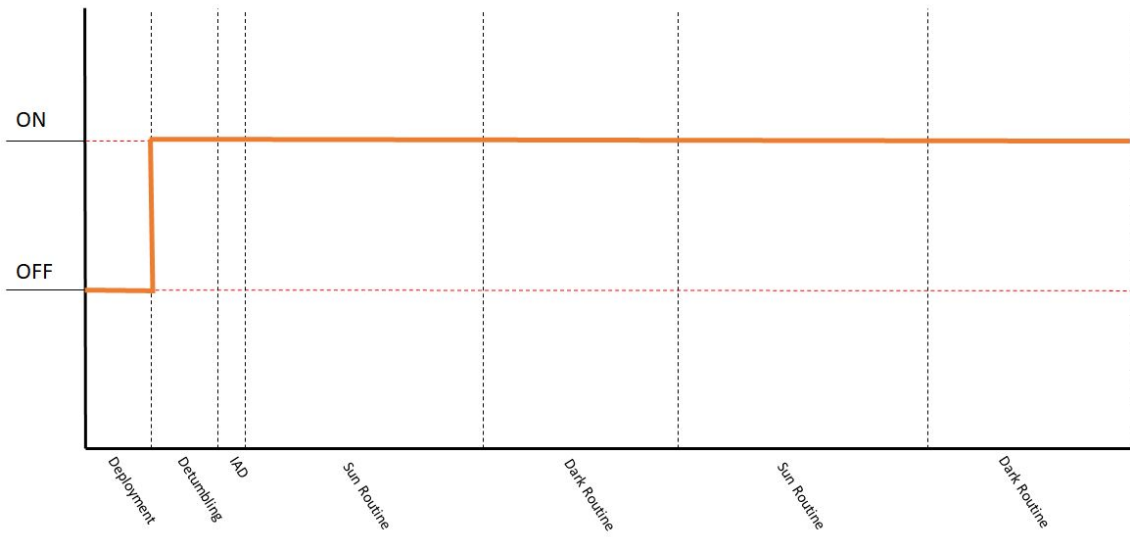


Figure 3.14. GPS State Plot.

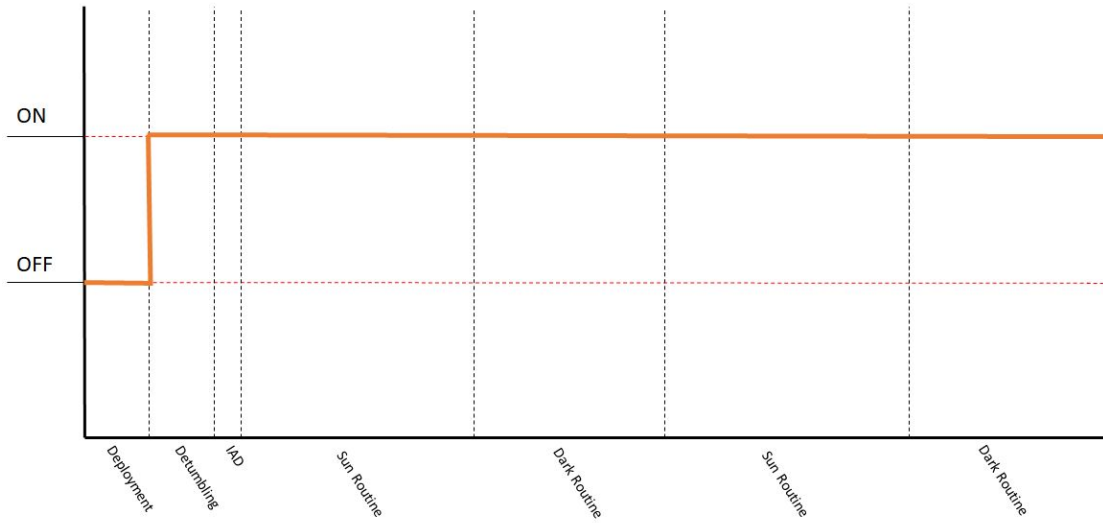


Figure 3.15. Magnetometer State Plot.

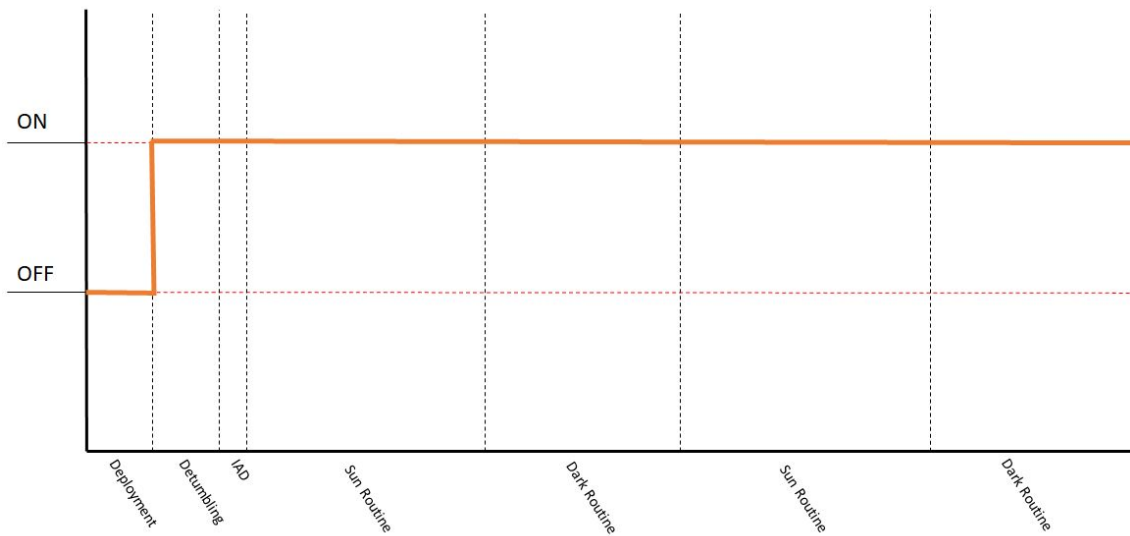


Figure 3.16. μ PPT State Plot.

3.5 Modeling of Available Memory

3.5.1 Volatile Memory

Once we had estimated the amount of memory that each of the sensors would use during operation, we set about to model that storage to determine if there was enough space on the OBC for all the sensors we had. To do this we used the on/off states to determine when the sensors

were operating, and calculated the total memory used for each operational state. We calculated this under the assumption that the sensors were constantly in use. While this will not be the case in reality, we thought it important to estimate the storage space using the “worst-case scenario” when if a sensor was turned on it was in constant operation. Once we calculated this, we created a table of the volatile memory used per stage and used it to create a plot. As with the on/off state plots, we showed sun routine and dark routine as separate stages, even though they are the both the same routine stage. This was to show that the memory used while in sunlight and in darkness was the same for the OBC.

Table 3.7. eLEO Volatile Memory Usage.

Operational State	Deployment	Detumbling	Initial Attitude Determination	Sun Routine	Dark Routine
Total Memory (KB)	0 KB	500 KB	500 KB	500 KB	500 KB
Total Memory (MB)	0 KB	0.5 MB	0.5 MB	0.5 MB	0.5 MB

Table 3.8. Formation Flying Volatile Memory Usage.

Operational State	Deployment	Detumbling	Initial Attitude Determination	Sun Routine	Dark Routine
Total Memory (KB)	0 KB	520 KB	520 KB	520 KB	520 KB
Total Memory (MB)	0 KB	0.52 MB	0.52 MB	0.52 MB	0.52 MB

As seen in the above table, only 0.5 MB of the 8 MB available was used at any given time. This means that there is plenty of MRAM storage in the OBC to contain the data sent to it by the sensors on the CubeSat. This means that there is plenty of space for other sensors to be added to this CubeSat for a particular mission, provided they use volatile memory. The following figures show plots of the memory usage, one with a y-axis maximum of 8 MB, the other a maximum of 1 MB. This was to provide a zoomed-in look at the memory usage.

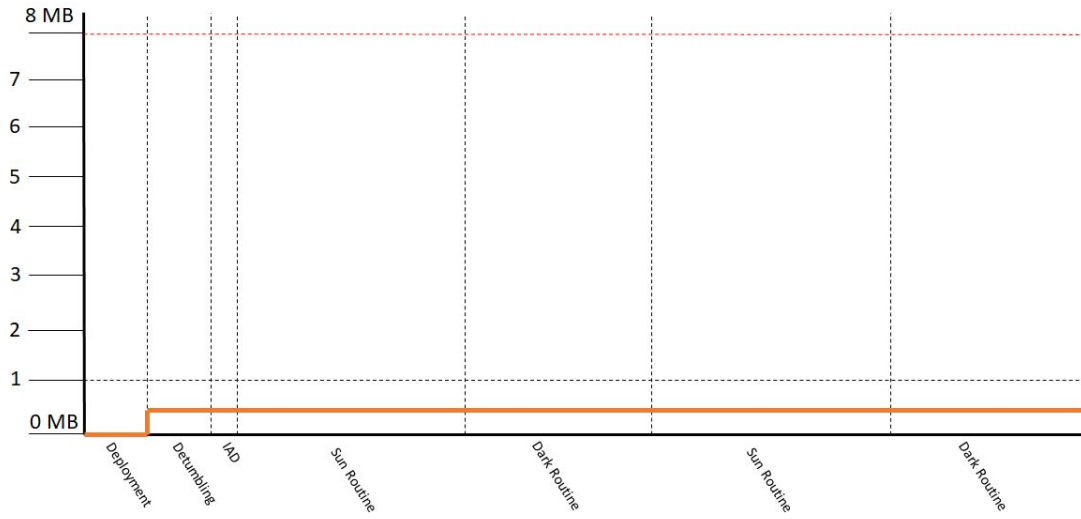


Figure 3.17. eLEO Volatile Memory Usage.

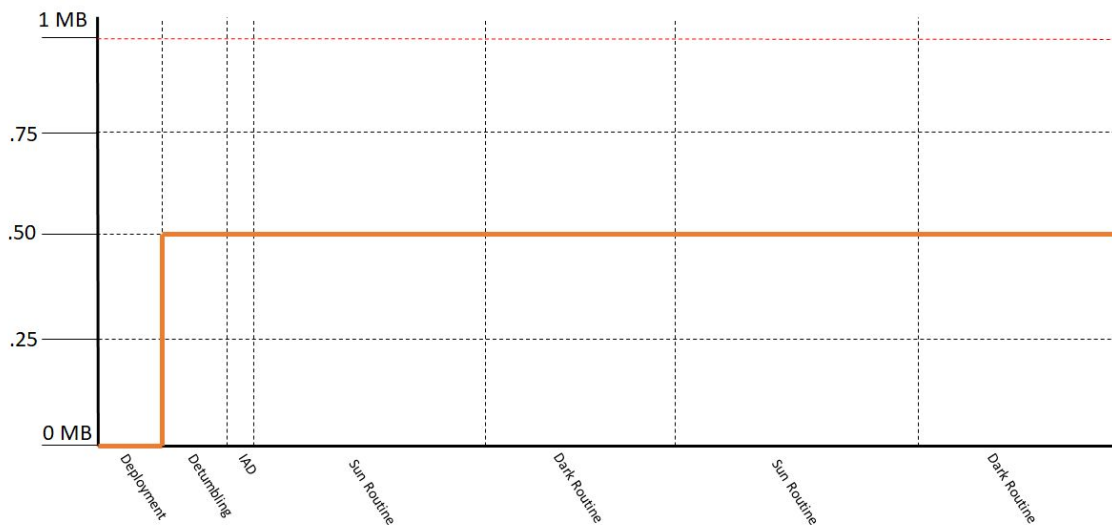


Figure 3.18. eLEO Volatile Memory Usage (Zoomed-In).

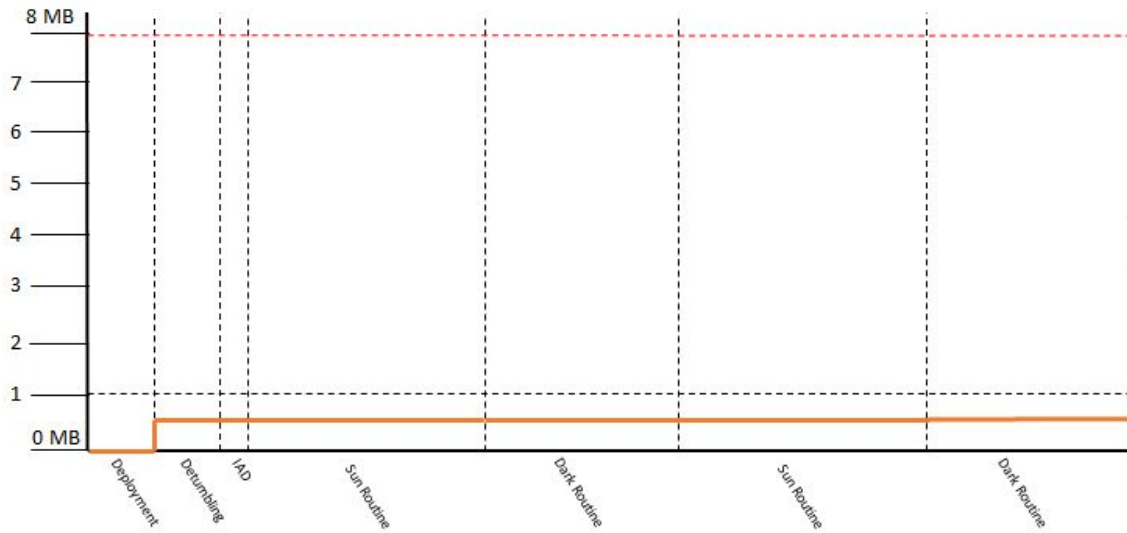


Figure 3.19. Formation Flying Volatile Memory Usage.

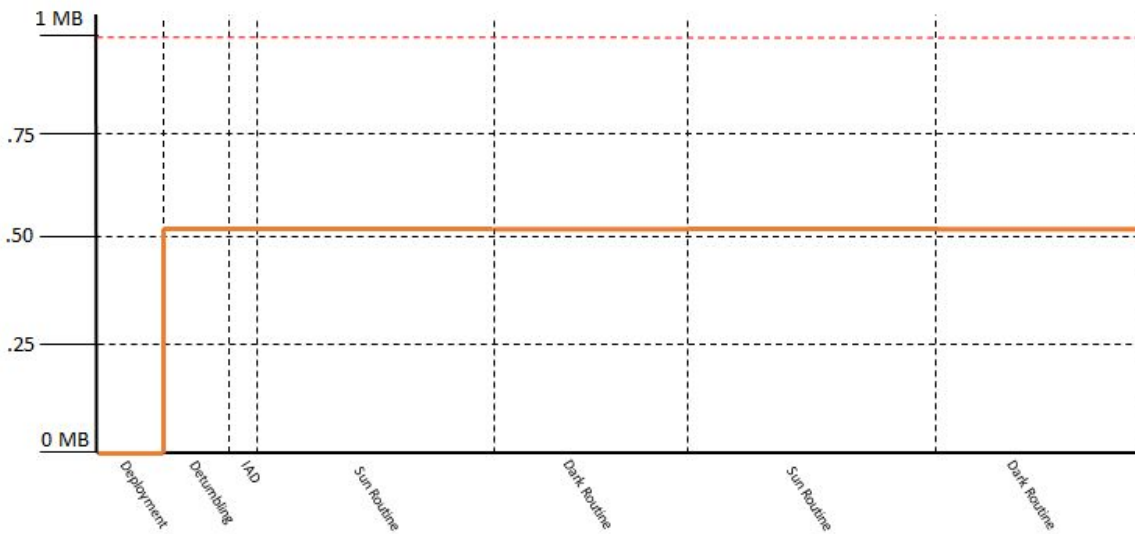


Figure 3.20. Formation Flying Volatile Memory Usage (Zoomed-In).

3.5.2 Non-Volatile Memory

For the non-volatile memory, we set out to model the usage in a different way. Since the data for the flash memory from Table 3.5 for the OBC and EPS was in KB/s, we created a simple MATLAB script to calculate and plot the storage used over the time of the mission. The script took the memory used per second by the OBC and EPS and plotted it with respect to time. These MATLAB plots can be seen in Appendix A.

Similar to plotting the volatile memory, we assumed a worst-case scenario for the accumulation of memory in the OBC. We assumed that there would be no transference of memory from the OBC to the ground station throughout the mission, to make sure that there was enough non-volatile memory storage in the OBC for the entire mission. Figure 3.21 shows the total amount of non-volatile memory gathered from the OBC and EPS until all 4 GB are used. It can be seen that the CubeSats can go for 186 days without transmitting data before the non-volatile memory is full.

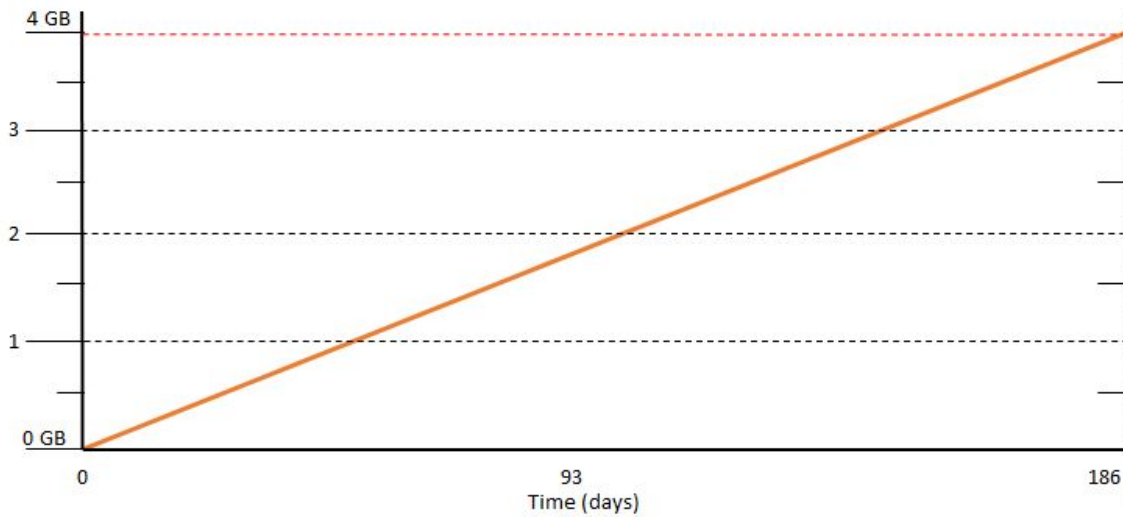


Figure 3.21. Time until Non-Volatile Memory Saturation.

Figure 3.22 shows the total estimated memory used for the eLEO mission, with the x-axis maximum being approximately 8,000,000 seconds to allow for the entire 90 days that were planned. Thankfully, without any transference of memory, the OBC and EPS only accumulated about 1.944 GB of memory, which is half of the 4 GB maximum of the OBC. The OBC used 1.5558 GB of memory, while the EPS used 0.38896 GB. This leaves approximately 2 GB of memory for another sensor, potentially to collect data, or something else for whatever mission is needed. It is also important to note that data would most likely be transferred during an actual mission, so there would be even more space available for additional sensors and data gathering devices.

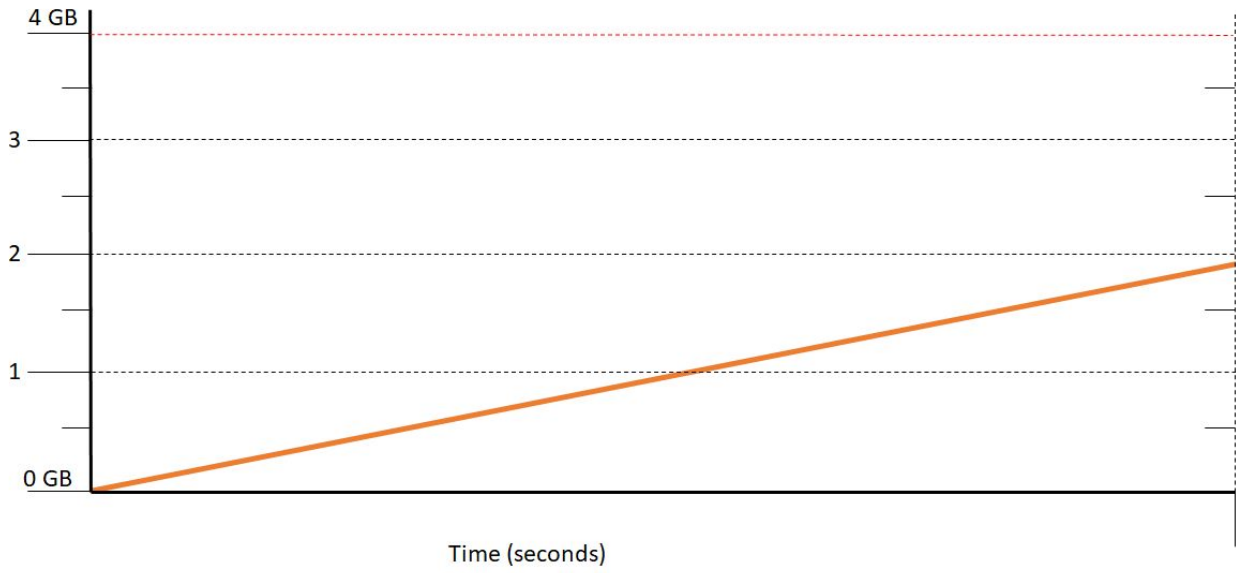


Figure 3.22. eLEO Total Non-Volatile Memory Usage.

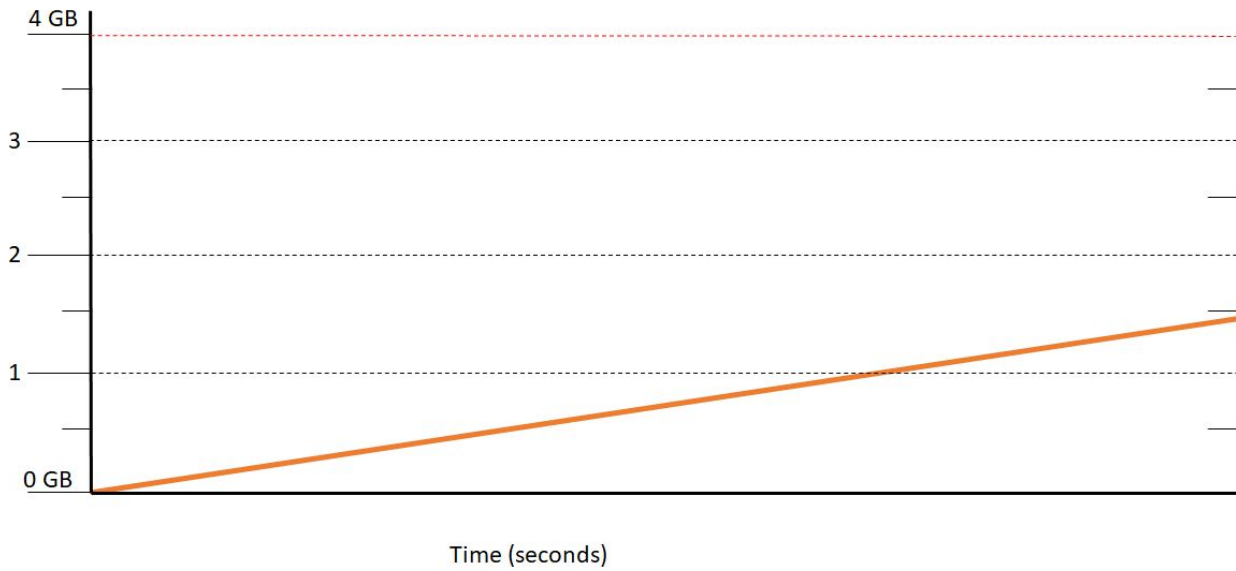


Figure 3.23. eLEO OBC Non-Volatile Memory Usage.

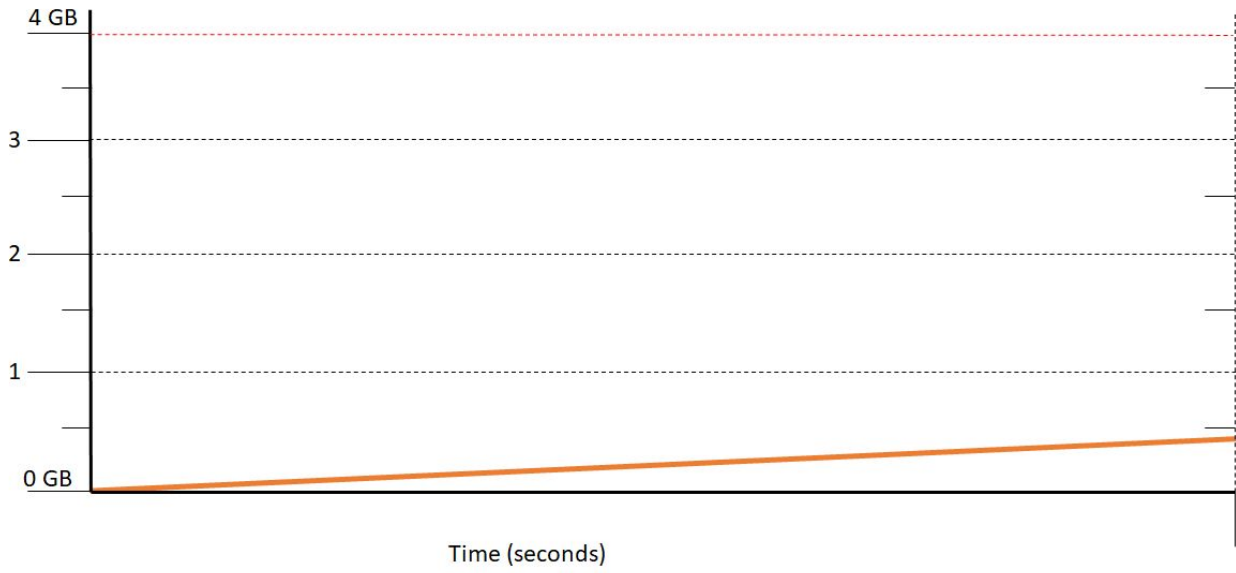


Figure 3.24. eLEO EPS Non-Volatile Memory Usage.

4. Structural Analysis

Structural Analysis is an important aspect of this design, and simulations were used to determine whether or not the CubeSat would be physically capable of withstanding the mission environment. This section details the methods used to simulate the stresses the CubeSats would be exposed to during the high-vibration launch phase.

4.1 Testing Requirements

Launch vehicle selection was beyond the scope of this project. Fortunately, in the case of a lack of specified launch vehicle vibration magnitude or frequency, the General Environmental Verification Standard (GEVS), formulated by NASA's Goddard Space Flight Center, may be used [11]. This standard specifies the qualification and acceptance frequencies and associated acceleration spectral densities associated with them that a spacecraft may have to withstand during launch. These values are shown below in Table 4.1.

Table 4.1. GEVS random vibration loads [11].

Frequency (Hz)	ASD Level (g^2/Hz)	
	Qualification	Acceptance
20	0.026	0.013
20-50	+6dB/oct	+6dB/oct
50-800	0.16	0.08
800-2000	-6dB/oct	-6dB/oct
2000	0.026	0.013
Overall	14.1 G_{rms}	10.0 G_{rms}

The above values were used only to test the structural components of the CubeSats. All other components, such as the thrusters, sensors, OBC, and solar panels, are expected to be tested by manufacturers. This assumption was in the project completed in 2017, which improved upon the simulations conducted earlier in 2013, and will continue to be used for this project [5], [8]. This simplification has the combined effect of both reducing computation time and improving results accuracy, as simulation software is inherently more accurate with less complex models.

4.2 Analysis in ANSYS

ANSYS was the chosen software for these simulations. In the past, SolidWorks simulation tools proved too inaccurate. While SolidWorks is a modeling software with analysis tools, ANSYS is a simulation software with some modeling capability. Therefore, models were first built in SolidWorks by the Mechanical Design Team, then exported as parasolid (.x_t) files for the more robust analysis ANSYS could provide.

Modal and random vibration analyses were conducted in ANSYS. The procedure very closely resembled that used in the 2017 project [8]. Since the GEVS standard ranges from 20 Hz to 2000 Hz and ANSYS searches for modes from low frequencies up, the modal analysis settings in were adjusted to search for up to 99 modes between 20 Hz and 3000 Hz. These settings ensure that, as long as fewer than 99 modes are found, no modes were missed in the range of 20 Hz to 2000 Hz. Random vibration studies then used these results along with PSD G Accelerations matching those in the “Acceptance” column of the GEVS in the x-, y-, and z-directions. One end of the satellite was chosen to be fixed to simulate containment within the launch vehicle. The random vibrations were applied to the satellites through these fixed supports, as external forces would be acting on the satellites through its contact with the launch vehicle.

To attain the more accurate results of stresses and deformation of the CubeSat structure, all nonstructural components were designated as rigid bodies. This designation allows ANSYS to take the mass properties of these components into account during its calculations without having to calculate effects on those components. All bodies within the design, both structural and nonstructural, were set as “bonded” to each other. This means that no separation or sliding

between the bodies is allowed, as would be the case when components are fixed to each by screws at multiple points.

4.2.1 Formation Flying CubeSat Results

Structural analysis for the formation flying mission was conducted on the 20U design. A graph depicting modes and their frequencies is shown in Figure 4.1. A total of 29 modes were found for this design with frequencies ranging from 33 Hz to 2,940 Hz. In previous years, modal frequencies below 100 Hz would be a cause for concern, as this leaves the possibility of resonance of the P-POD deployer, which has its natural frequency at about 100 Hz. However, the 16U and 20U CubeSats are too large to be deployed by the P-POD and would require a much larger deployer. Therefore, these low frequency modes were not considered to be prohibitive in the context of this project. All the same, these modes would be an area of concern in deployer selection.

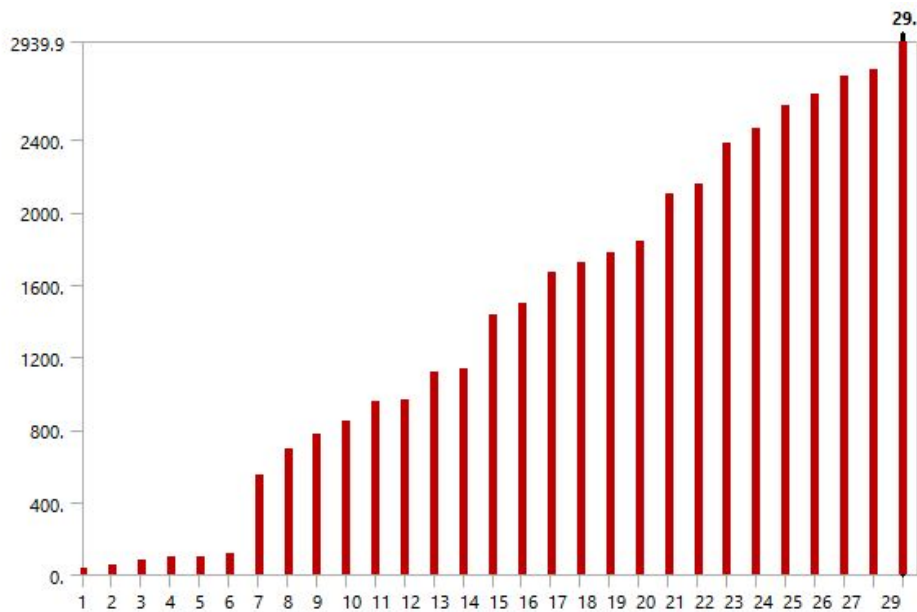


Figure 4.1. Modes of the 20U CubeSat design.

Results were collected for both the deformations and equivalent (von Mises) stresses caused by vibration. It was found that maximum deformation caused by vibration along a given axis occurred in the direction of that axis. Thus, images are only included for directional deformation along the same axis as the random vibration from which it resulted.

It was found that maximum stress and deformation from random vibration along the x-axis occurred at the coarse sun sensor mounting at the non-fixed end of the satellite. The deformation and stress are shown in Figures 4.2 and 4.3, respectively.

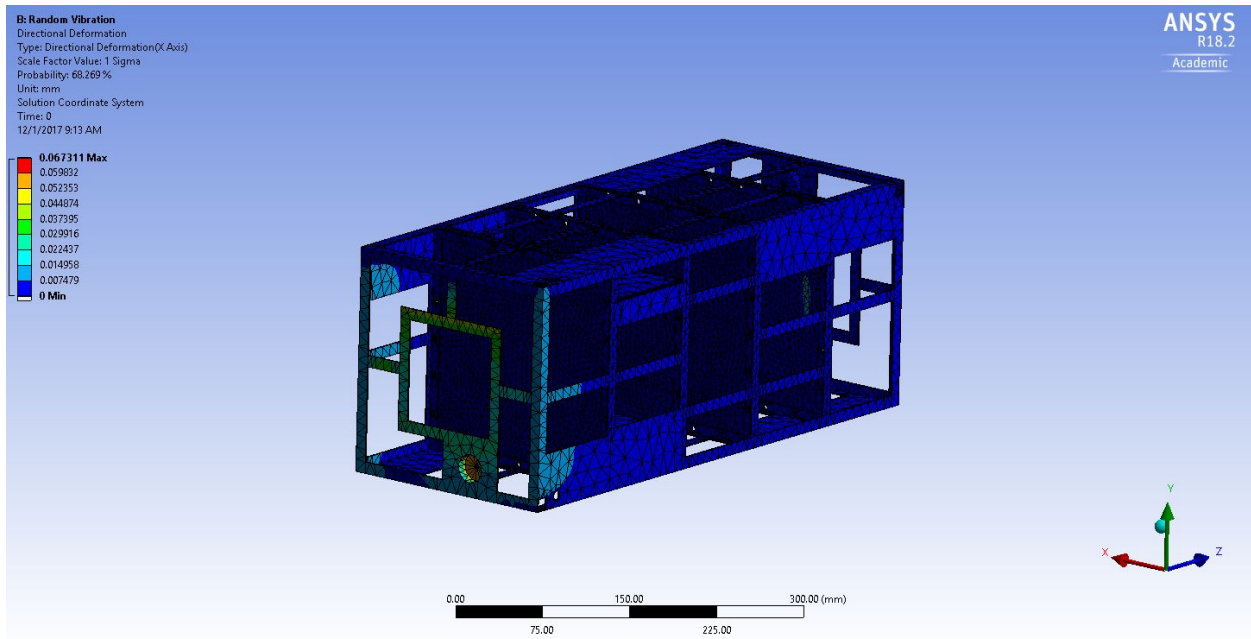


Figure 4.2. Deformation along the x-axis in response to vibration along the x-axis for the 20U.

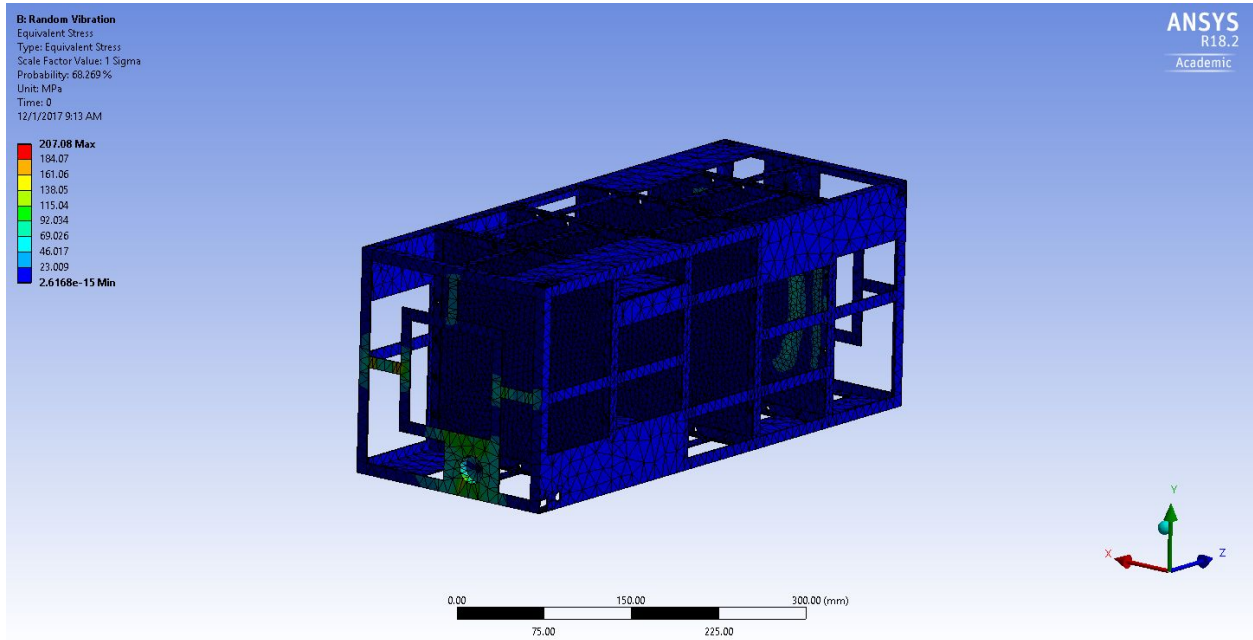


Figure 4.3 Equivalent stress due to vibration along the x-axis for the 20U.

Similar results were found for vibration along the y-axis. This makes sense, as the cross-section of this design is a square and the x- and y-axes both lie within the plane of this square. In other words, the x- and y-axes are orthogonal to the long axis of the satellite, which is fairly symmetrical, so results from vibrations along these two axes should be similar. Results are shown in Figures 4.4 and 4.5.

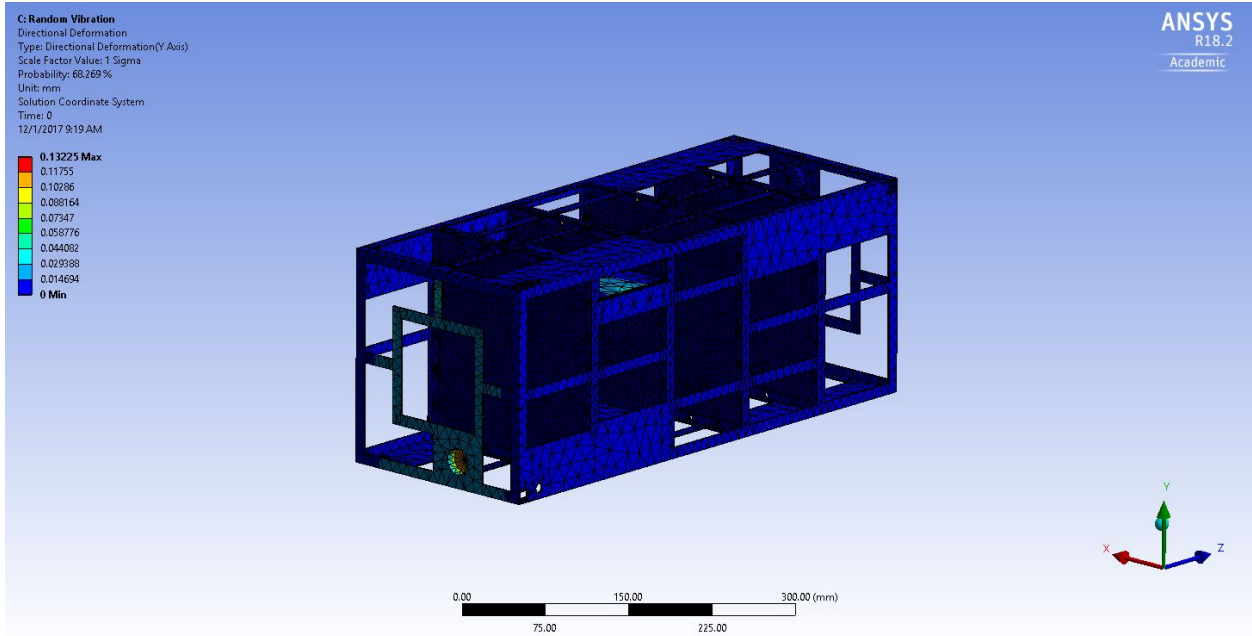


Figure 4.4. Deformation along the y-axis in response to vibration along the y-axis for the 20U.

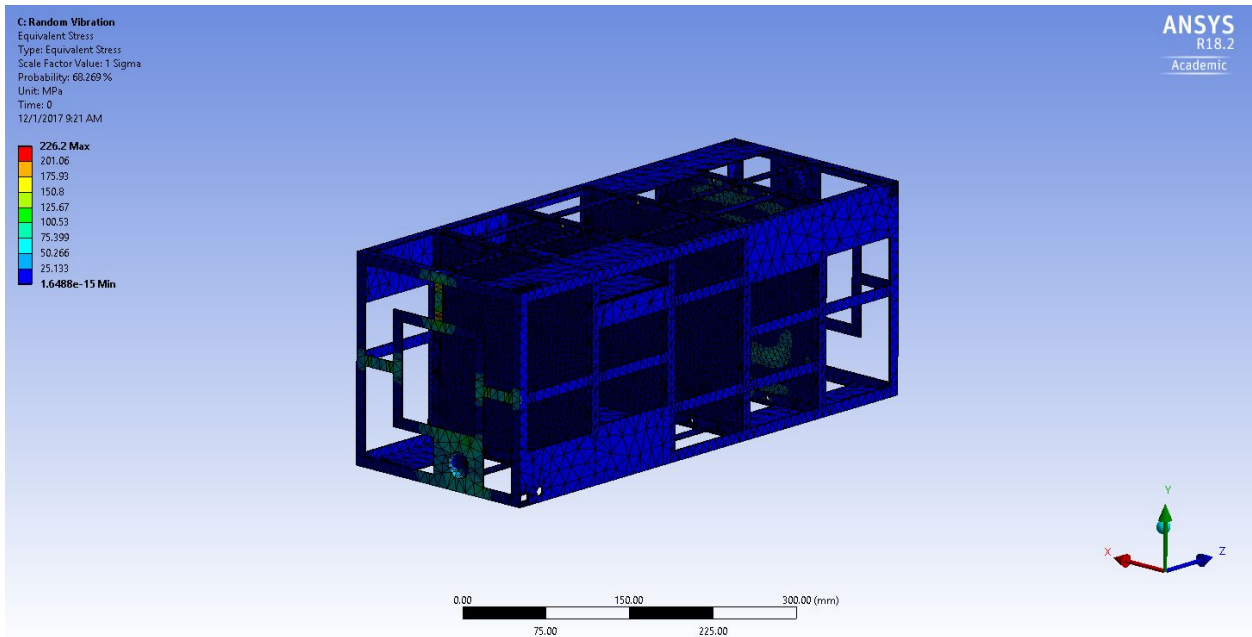


Figure 4.5. Equivalent stress due to vibration along the y-axis for the 20U.

Vibration along the z-axis of the 20U CubeSat resulted in the greatest deformation. This time some deformation occurred near the coarse sun sensor mounting, but maximum deformation and stress was along the frame connection. This can be seen in Figures 4.6 and 4.7.

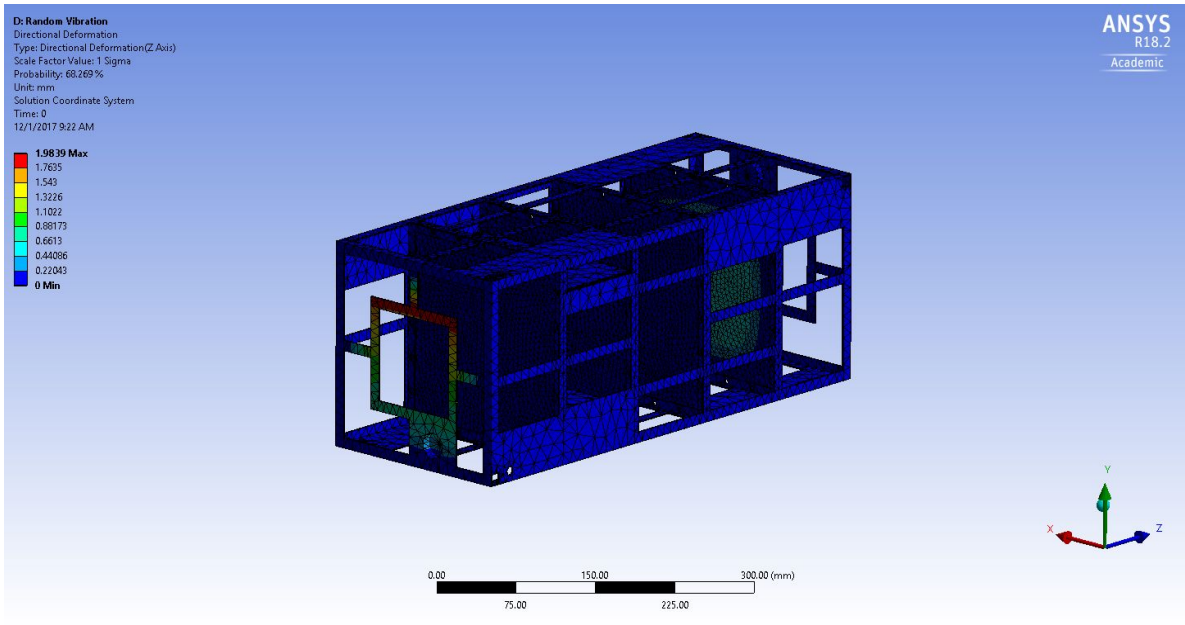


Figure 4.6. Deformation along the z-axis in response to vibration along the z-axis for the 20U.

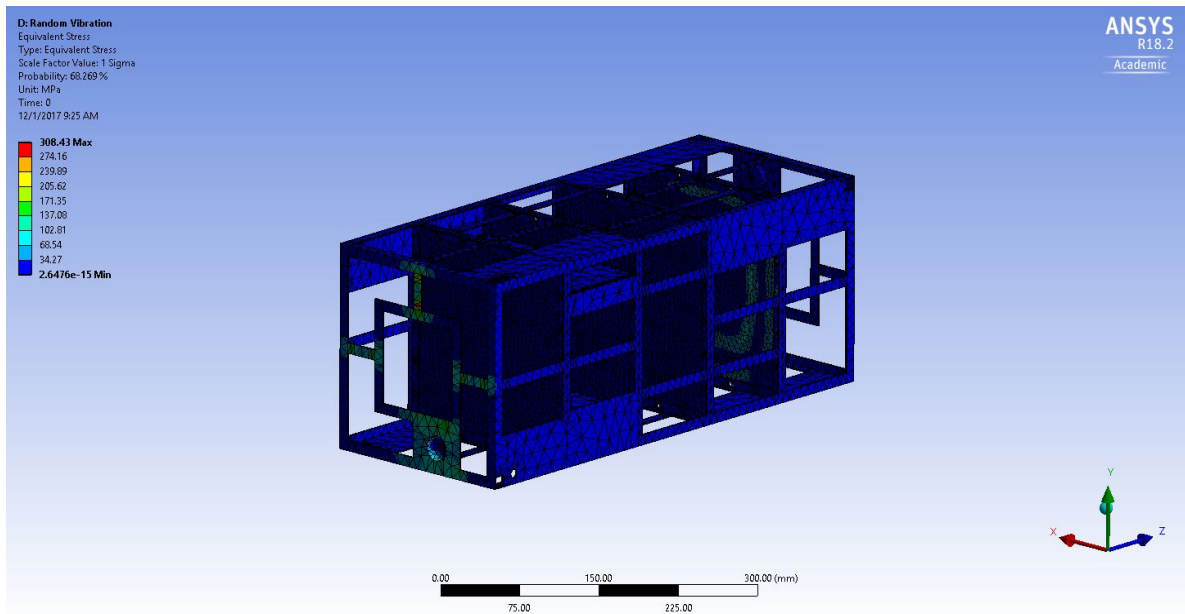


Figure 4.7. Equivalent stress due to vibration along the z-axis for the 20U.

A summary of the maximum deformations and stresses to vibration in each direction is available in Table 4.2. The maximum stress, at 308.43 MPa, falls well below the yield strength of aluminum 7075-T6, 503 MPa. With a maximum deformation of about 2 mm, there might be some cause for concern. However, this deformation should remain within the elastic region and no serious or lasting effects are likely.

Table 4.2. Maximum stresses and deformations of 20U CubeSat Design.

Vibration Axis	Maximum Deformation (mm)	Maximum Stress (MPa)
X	0.067311	207.08
Y	0.13225	226.2
Z	1.9839	308.43
Overall	1.9839	308.43

Analysis was conducted using the same parameters for the 16U design. Figures 4.8 through 4.14 show the results. Note that the axes of the 16U CubeSat are oriented differently than those of the 20U so that in the 16U case the y-axis is along the long axis of the satellite and the x- and z-axes are in the plane of a cross-section of the satellite.

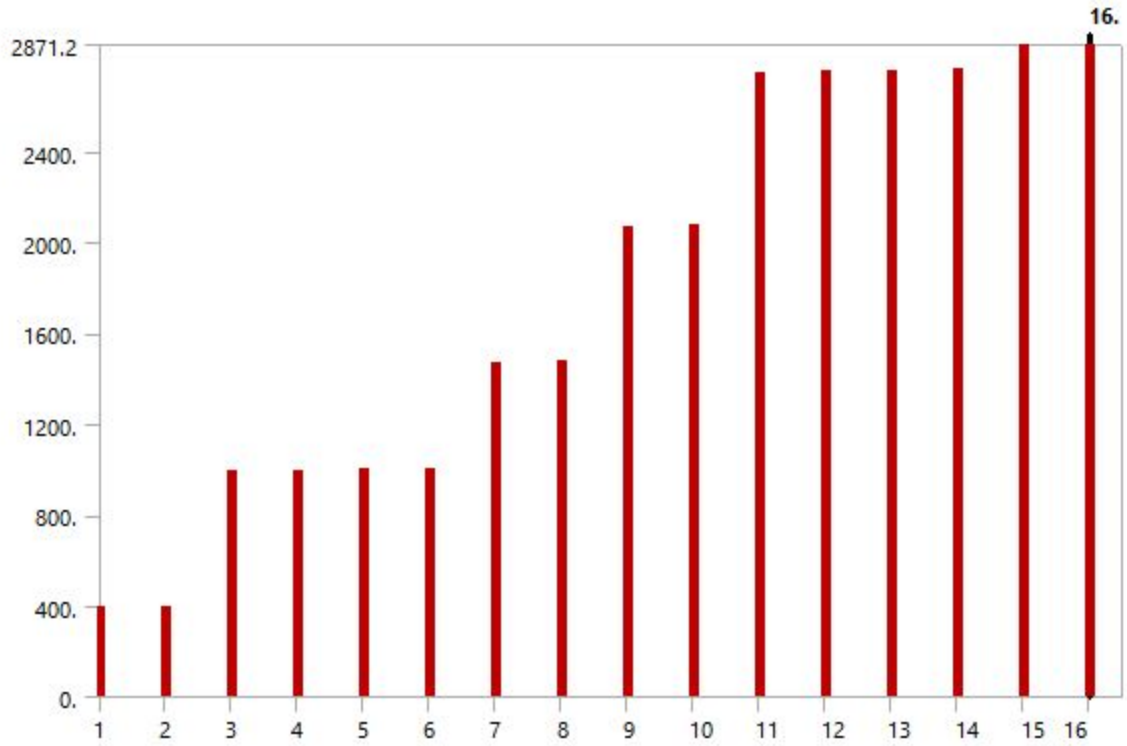


Figure 4.8. Modes of the 16U CubeSat design.

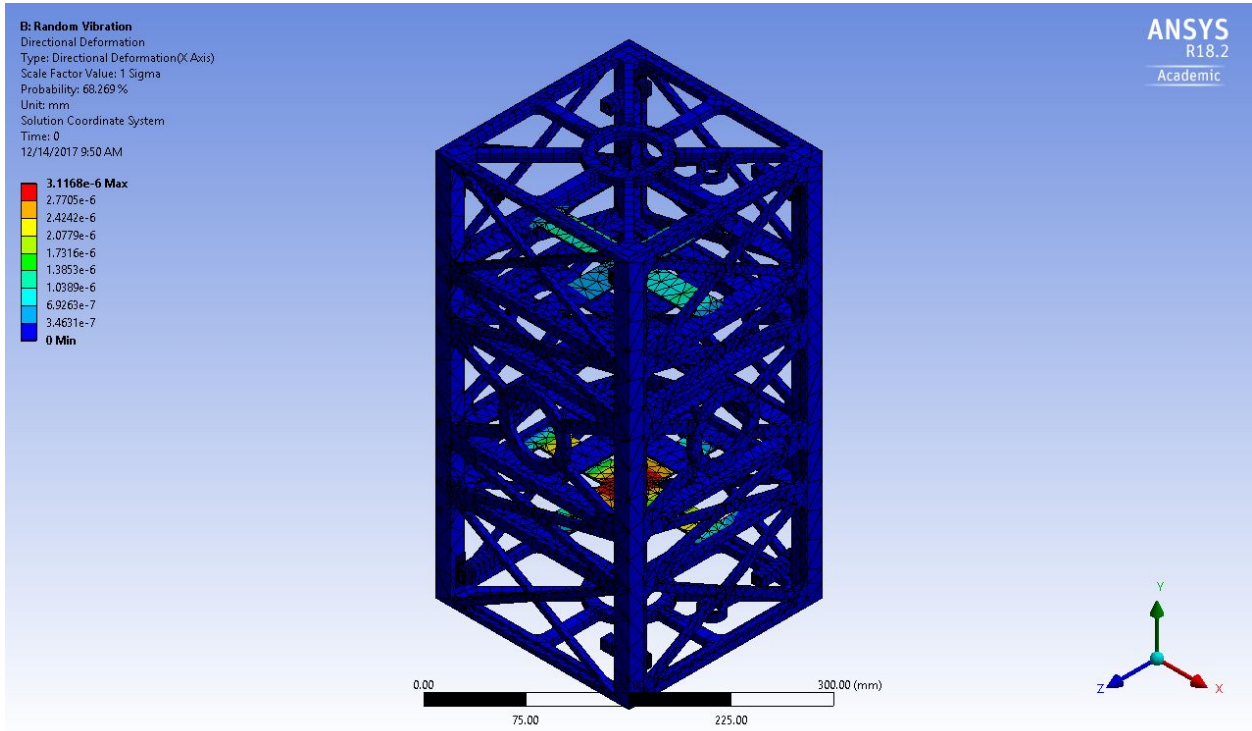


Figure 4.9. Deformation along the x-axis in response to vibration along the x-axis for the 16U.

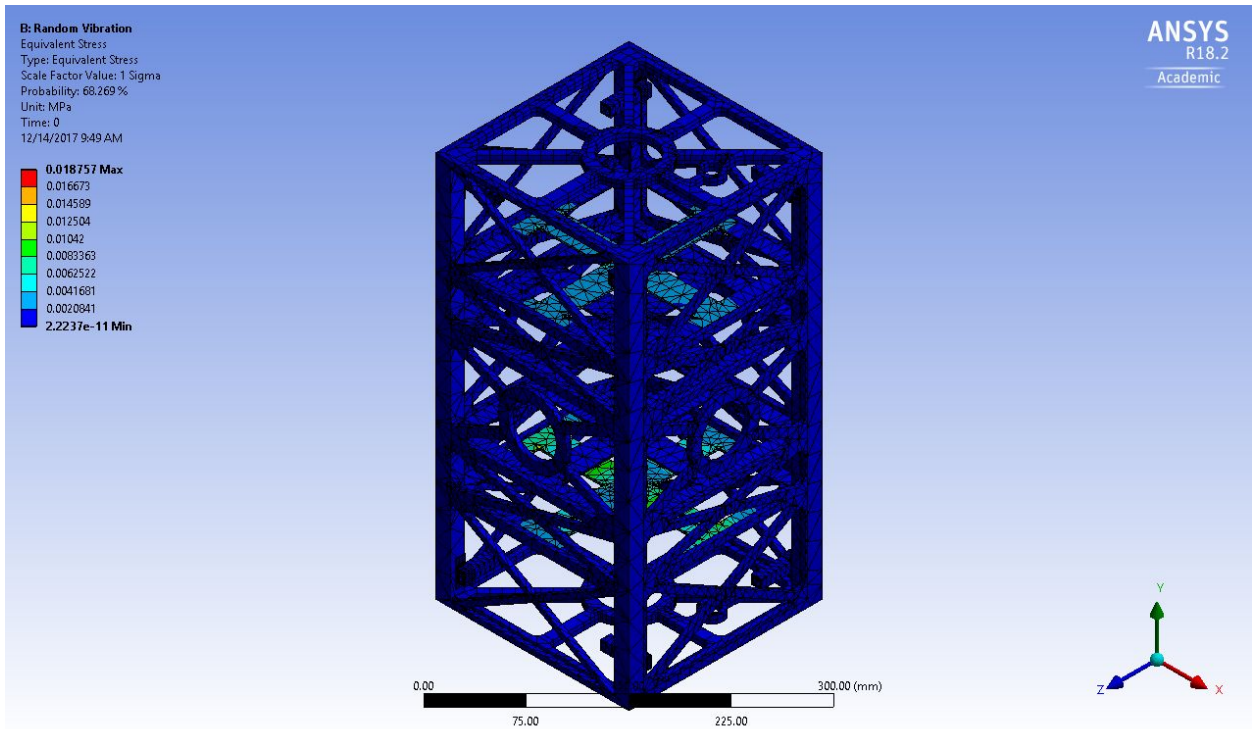


Figure 4.10. Equivalent stress due to vibration along the x-axis for the 16U.

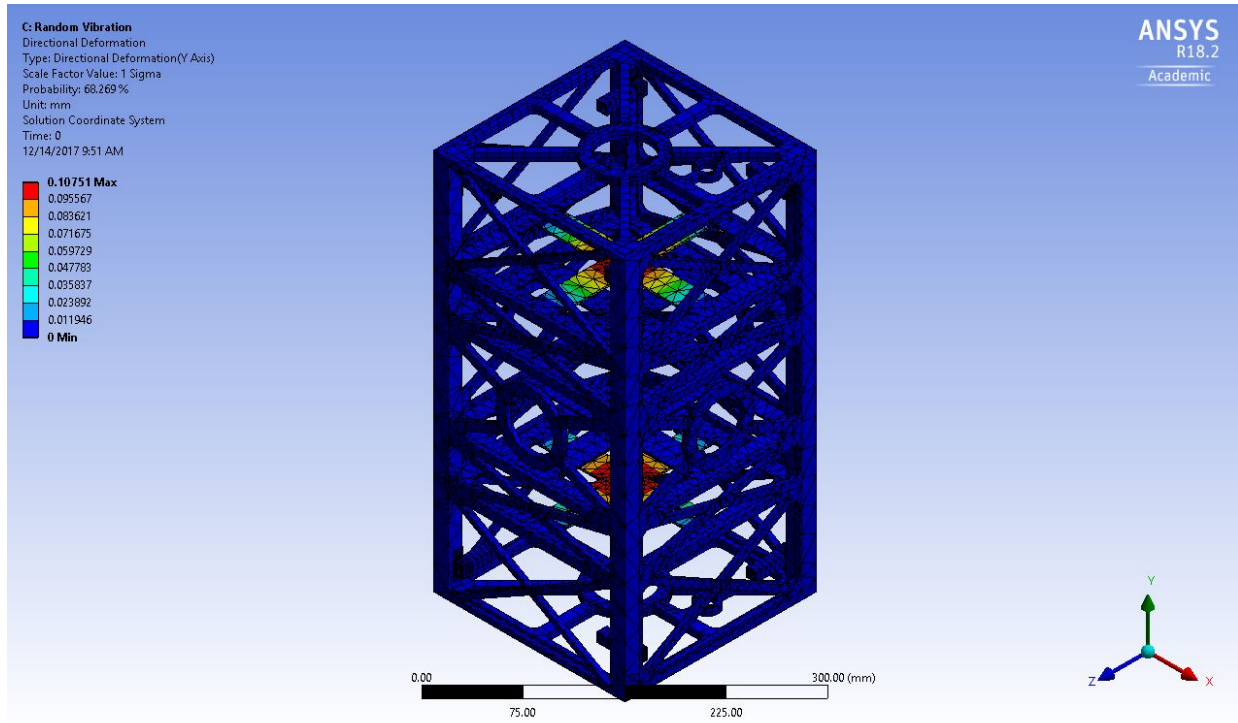


Figure 4.11. Deformation along the y-axis in response to vibration along the y-axis for the 16U.

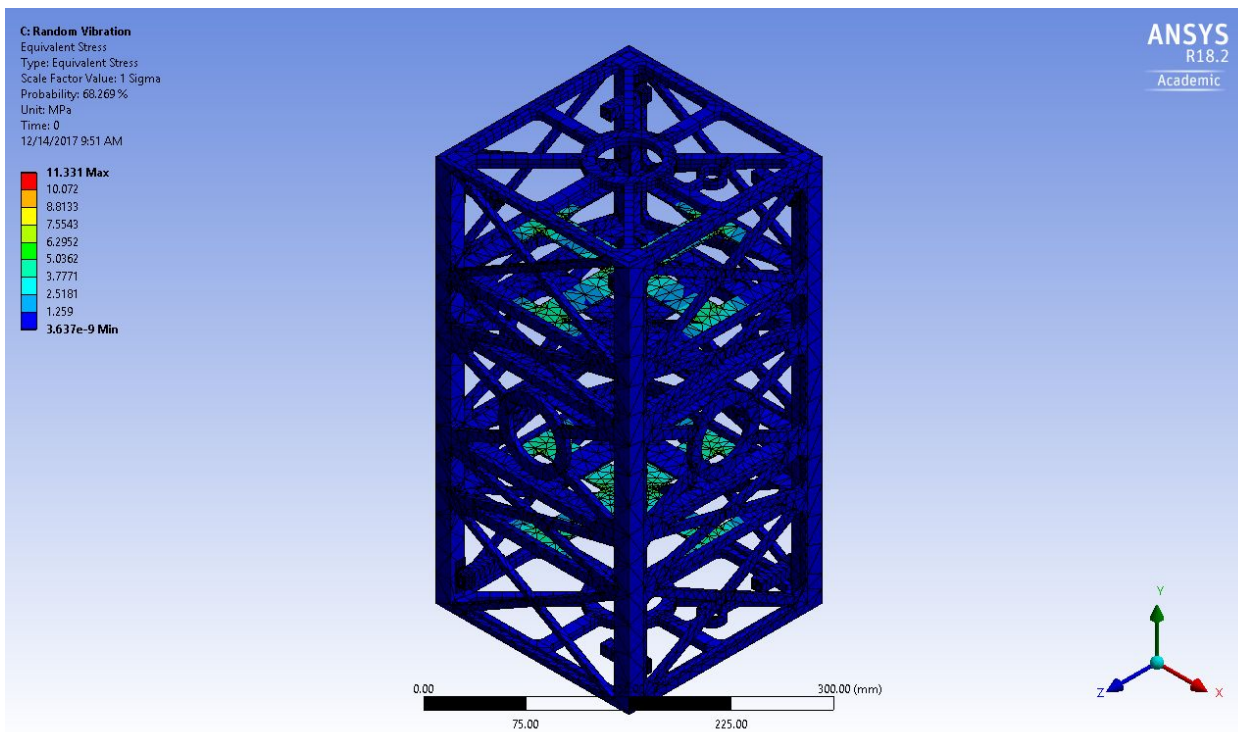


Figure 4.12. Equivalent stress due to vibration along the y-axis for the 16U.

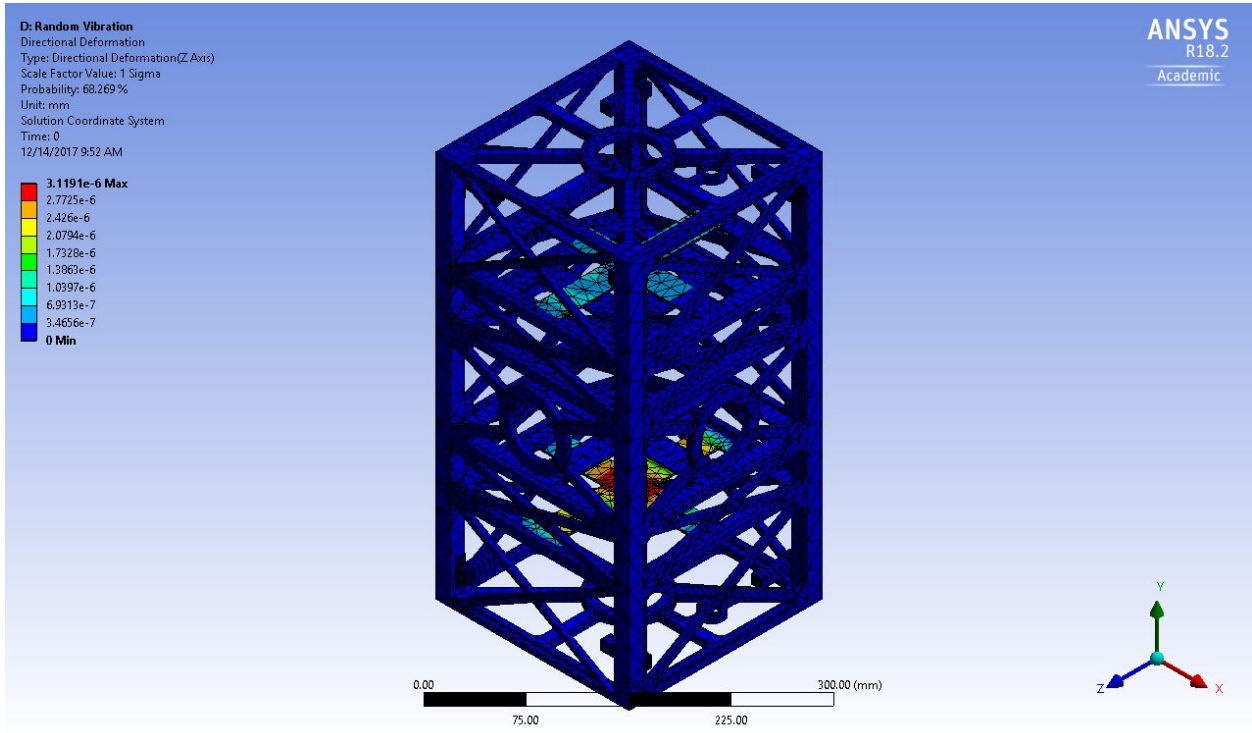


Figure 4.13. Deformation along the y-axis in response to vibration along the y-axis for the 16U.

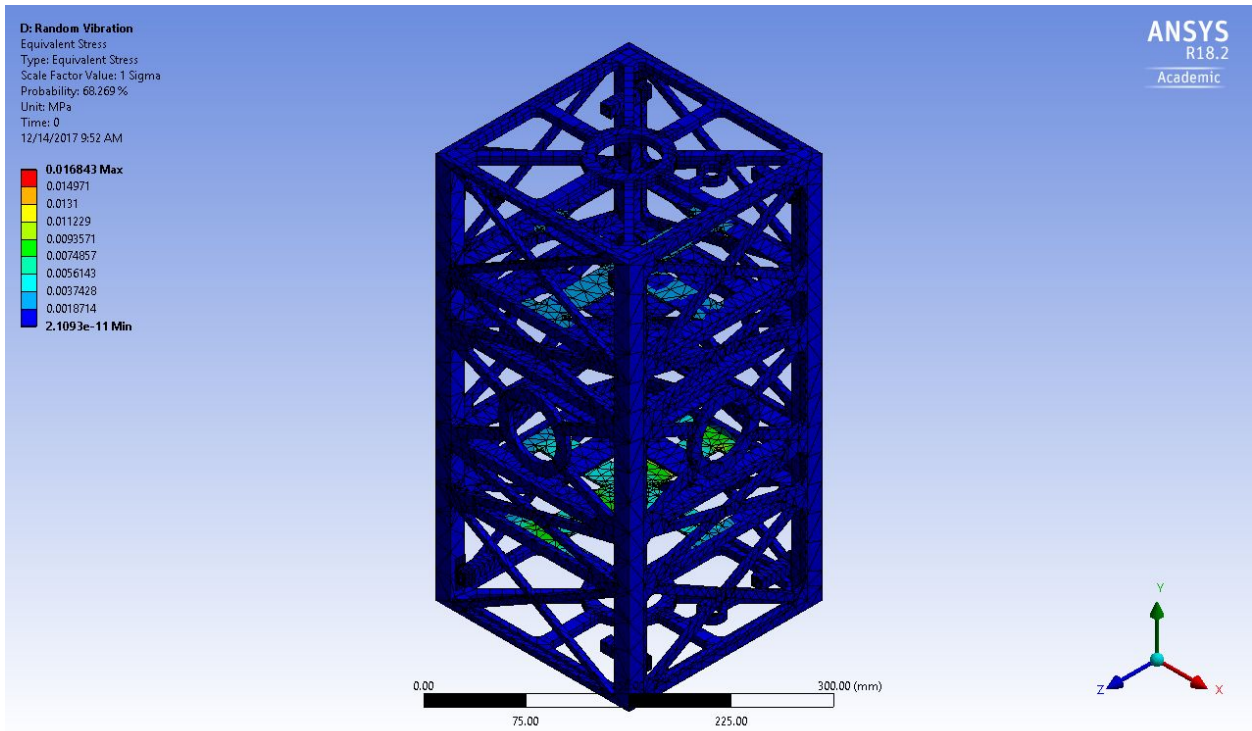


Figure 4.14. Equivalent stress due to vibration along the z-axis for the 16U.

Table 4.3 shows the maximum deformations and stresses resulting from random vibration for the 16U CubeSat. As can be observed from this data, the 16U CubeSat is even more structurally sound than the 20U, with stresses not even nearing the yield strength of the aluminum used. Additionally, the max stress and deformation occur along the y-axis (the long axis) of the 16U CubeSat, as expected. The maximum stress of 11.331 MPa leaves a safety factor of approximately 45.

Table 4.3. Maximum stresses and deformations of 20U CubeSat Design.

Vibration Axis	Maximum Deformation (mm)	Maximum Stress (MPa)
X	3.1168E-6	.018757
Y	0.10752	11.331
Z	3.1991E-6	0.016843
Overall	0.10752	11.331

4.2.2 eLEO CubeSat Results

The 4U eLEO CubeSat underwent structural analysis in ANSYS. For the eLEO CubeSat 51 modes were found. This was obtainable because the structure was not as complicated and it was smaller than the 20U design. Similarly to the formation flying mission the frequencies ranged from 33 Hz to 2,940 Hz.

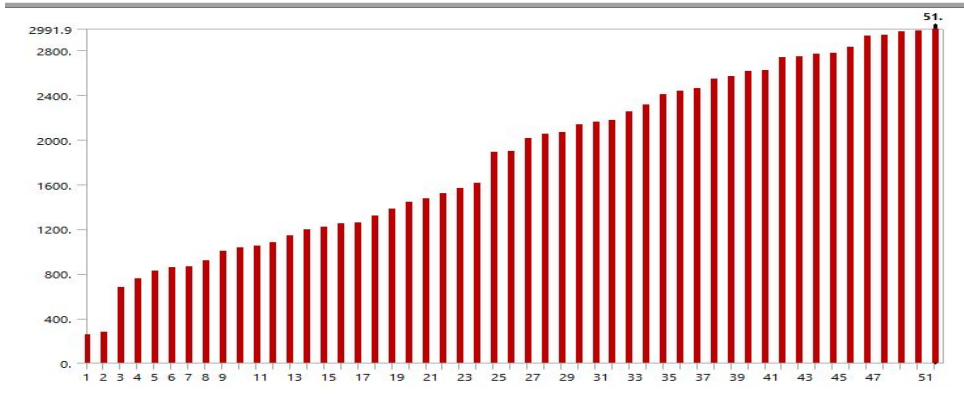


Figure 4.15. Modes of the eLEO CubeSat design.

Similarly to the 20U results were collected for both the deformations and equivalent (von Mises) stresses caused by vibration. It was found that maximum deformation caused by vibration along a given axis occurred in the x and y directions. This expected due to how the CubeSat will be mounted during the launch phase. Equivalently, the max stresses occur in the x and y directions. The maximum calculated stress is 108.22 MPa, making it about one fifth of the yield strength of aluminum 7075-T6, which is 503 MPa. From this the design of this CubeSat can withstand the vibrations during the launch phase of the mission.

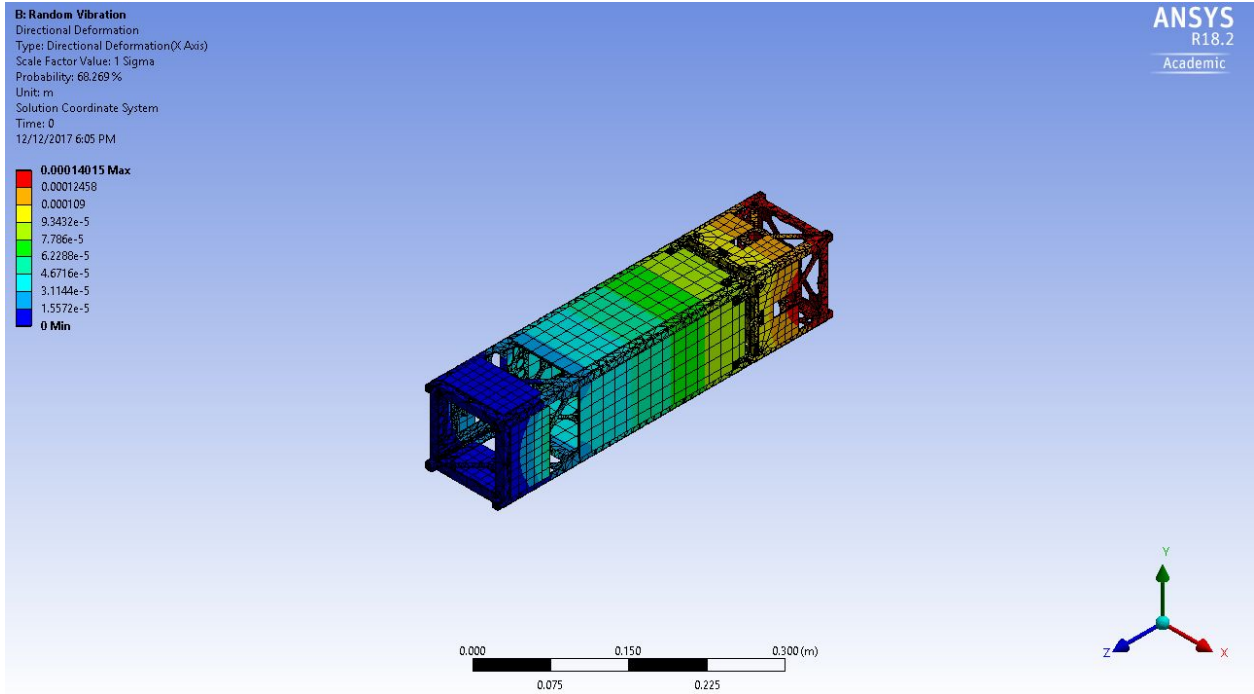


Figure 4.16. Deformation along the x-axis in response to vibration along the x-axis.

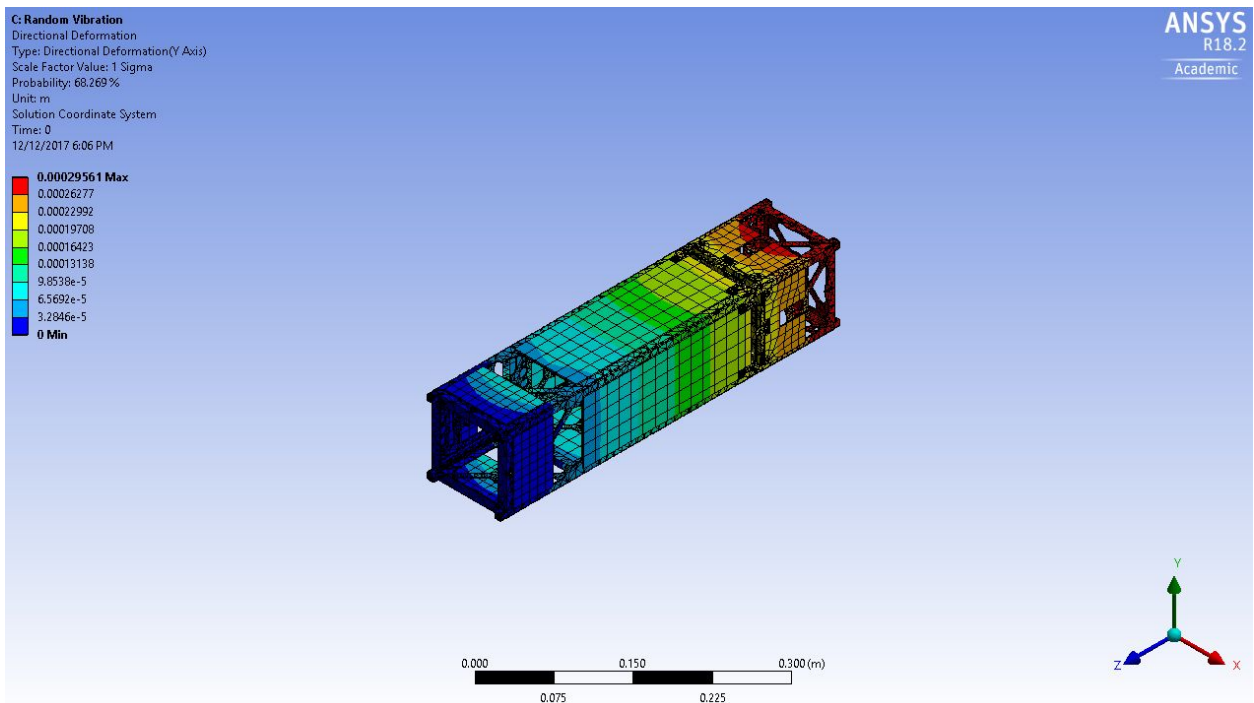


Figure 4.17. Deformation along the x-axis in response to vibration along the y-axis.

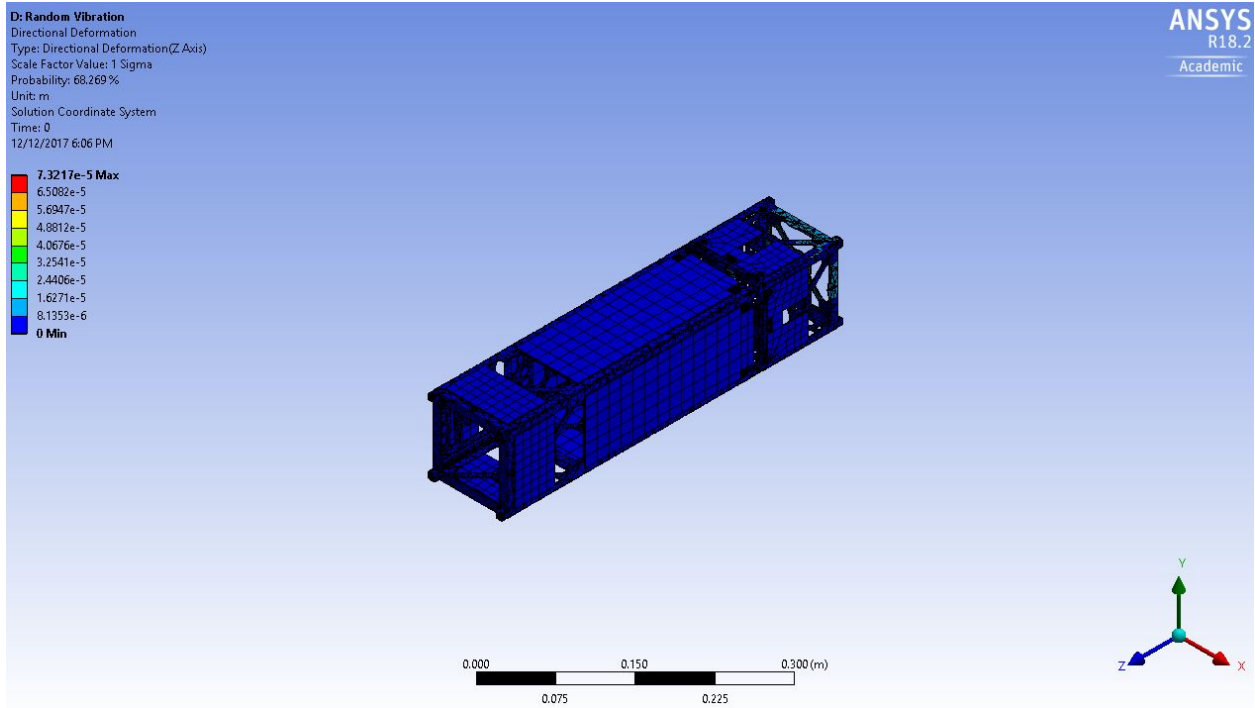


Figure 4.18. Deformation along the x-axis in response to vibration along the z-axis.

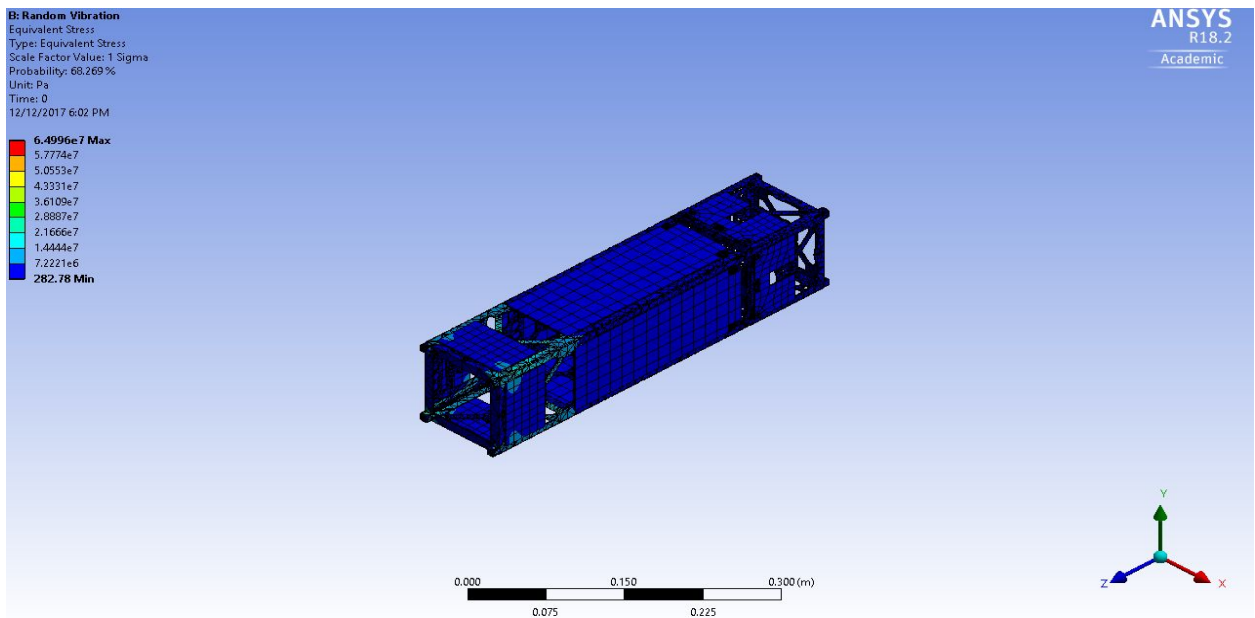


Figure 4.19. Equivalent stress due to vibration along the x-axis.

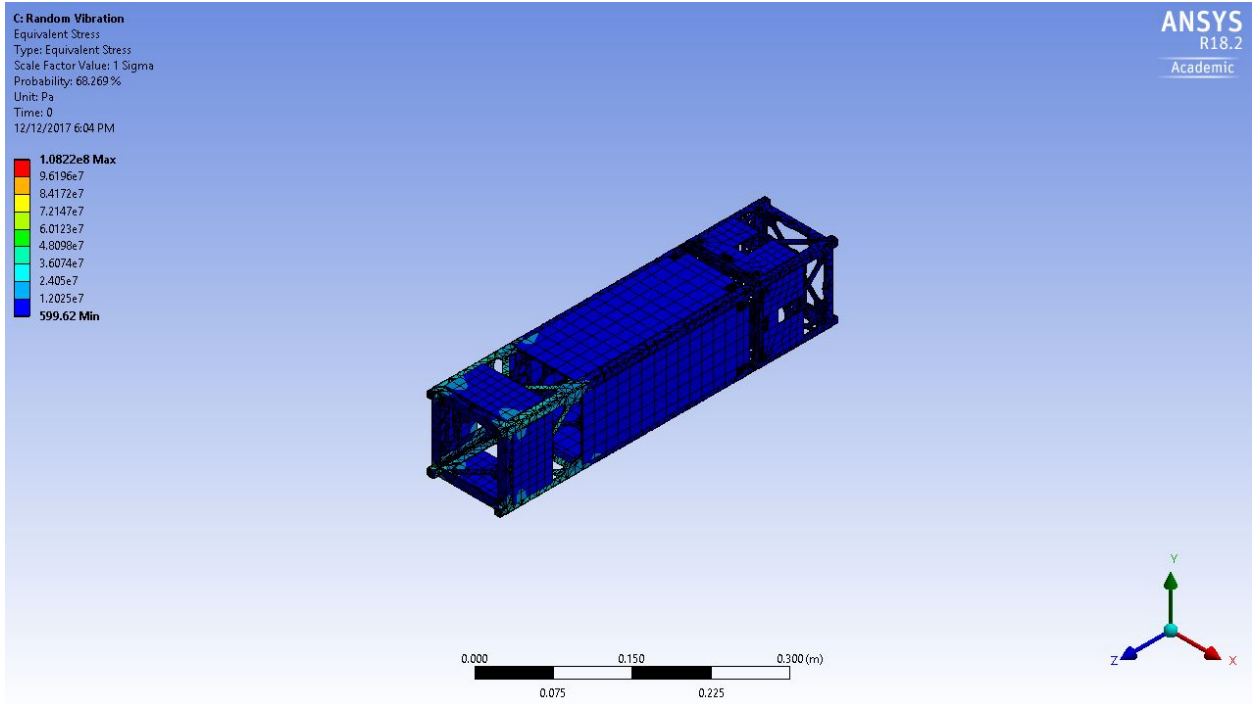


Figure 4.20. Equivalent stress due to vibration along the y-axis.

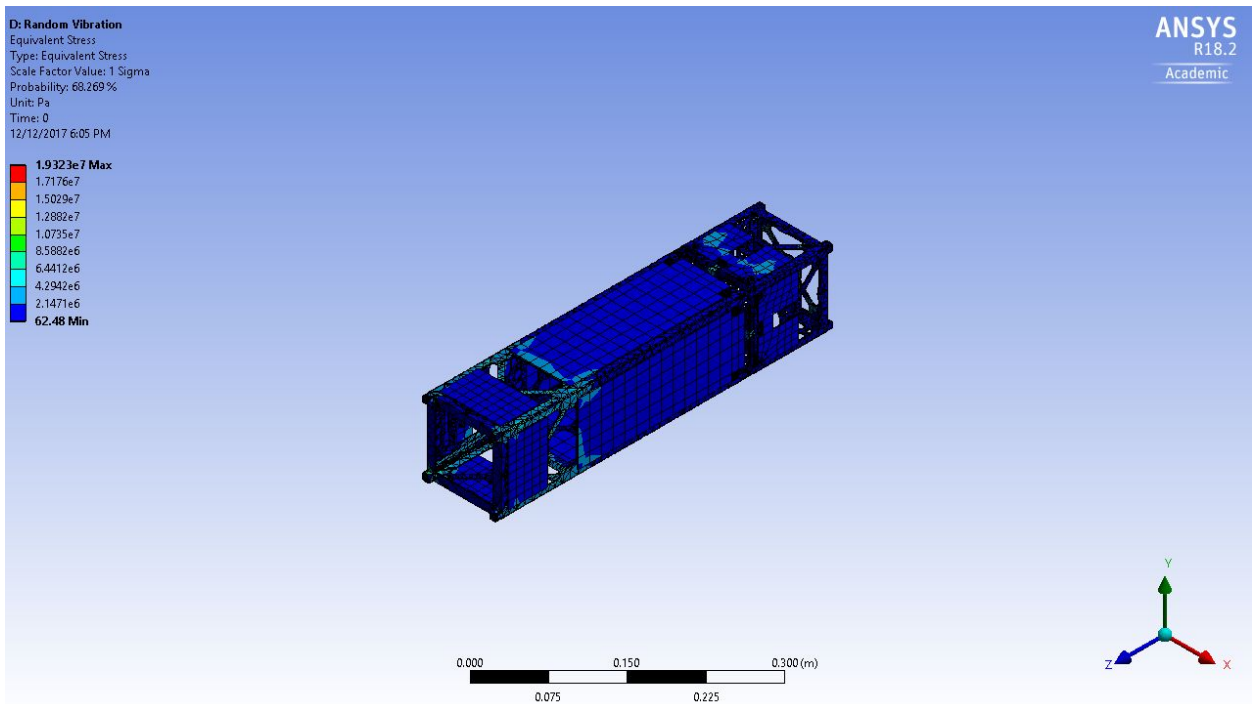


Figure 4.21. Equivalent stress due to vibration along the z-axis.

Below, in Table 4.4, the maximum deformation and stresses are shown in each axis.

Table 4.4. Maximum stresses and deformations of eLEO CubeSat Design.

Vibration Axis	Maximum Deformation (mm)	Maximum Stress (MPa)
X	0.14015	64.996
Y	0.26277	108.22
Z	0.065082	19.323
Overall	0.26277	108.22

5. Attitude Determination and Control Experiment Design

A test bed was constructed in order to test attitude determination equipment included in the CubeSat design. This test bed improved upon the design made last year [3]. The design, which allows for full rotation about the z-axis and some rotation about the x- and y-axes, is pictured in Figure 5.1. The 2017 CubeSat team conducted a thorough review of various test bed designs and selected and developed a design that was best suited to this project, so the same general design will be adhered to for this project.

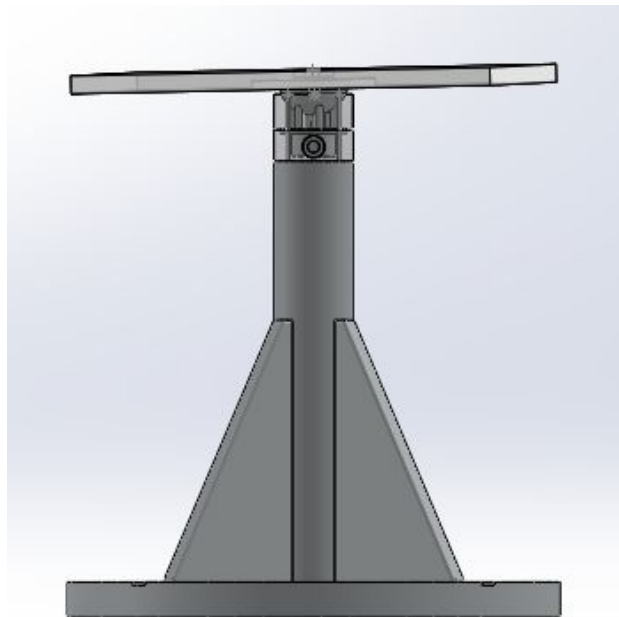


Figure 5.1. 2017 CubeSat MQP test bed design.

5.1 Previous Design Flaws and Recommendations

Last year's design was constructed, but it was found that there were some flaws. For this reason, a major task this year was to improve upon the design made last year. To this end, the chapter of the previous report on the test bed was reviewed thoroughly, and Jack Agolli, who led the design of the test bed last year, was contacted.

One issue was related to the supply of air being distributed through the test-tube and out four squared holes to lift the test bed. The air supply consisted of a tube line connected to an air tank. With the amount of air and time needed to lift the plate, the tank would run out of air quickly. Secondly, to lift the plate and keep it from oscillating and spinning due to air forces, a certain air pressure had to be attained. James expressed that they had no precise measuring device to measure the pressure and that in order to keep the plate steady, a pressure gage should be incorporated.

The major issue in last year's design involved balancing the platform. The team last year found that they could not adequately match the center of mass of the platform with its center of rotation, causing the platform to immediately tip once the air pressure was turned up. In addition, two designs of the air bearing were made last year, one which was a hemisphere and another that was more spherical.

A polycarbonate platform was used last year due to time constraints. However, the initial plan was to machine an aluminum platform. The issue with this is that aluminum is more dense than polycarbonate and as a result the center of mass would be significantly above the center of rotation so any rotation not purely about the vertical axis would cause instability.

The test bed design developed by the Hawaii Space Flight Laboratory, pictured in Figure 5.2, was recommended as a model that might be used to make improvements on the WPI test bed [32]. This design involves a complex counterbalance system that improves stability

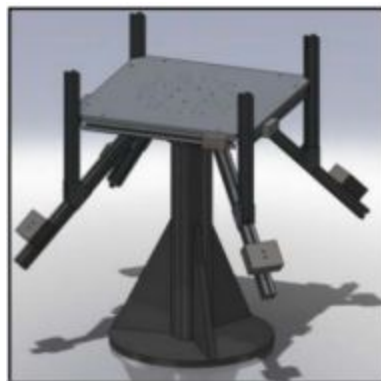


Figure 5.2. Hawaii Space Flight Laboratory test bed design [32].

5.2 Improvements to Test Bed Design

After speaking with James Loiselle, and Jack Agolli it was concluded that the most important improvement needed for the test bed was a counterbalance system. It was also decided that an attitude control system would be added to the test bed. This required the design and construction of reaction wheels as well as the addition of hardware and software to implement control laws.

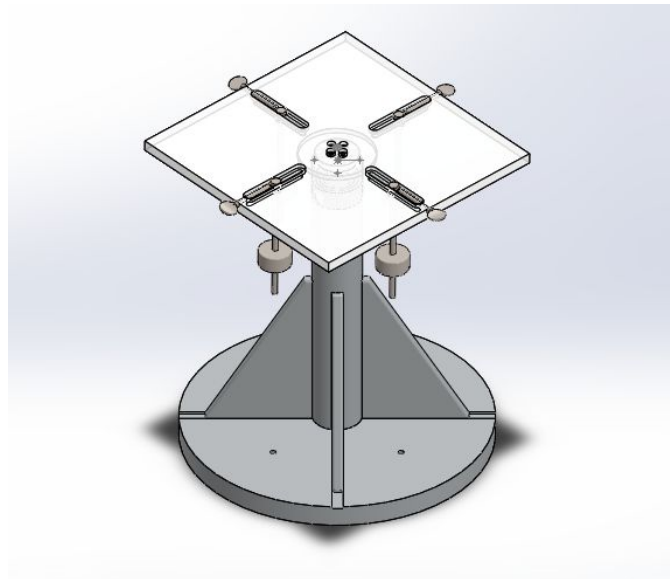


Figure 5.3. Test bed with counterbalance system.

5.2.1 Test Bed Plate Improvements

First, to solve the shortage of air, two lines were connected to the buildings air supply by the wall. There were two lines instead of one because we needed a line that would lift the test bed plate and then a separate line that lifted the isolated vibration control system table which run at different air supplies. Hence, this allowed us to have an unlimited supply of air feeding each component. Attached to each line was a pressure gauge. This would allow the two different plates to function in their optimal state.

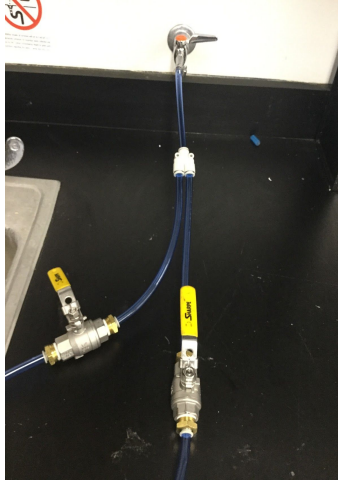


Figure 5.4. Two Airlines Connected to Building Supply.

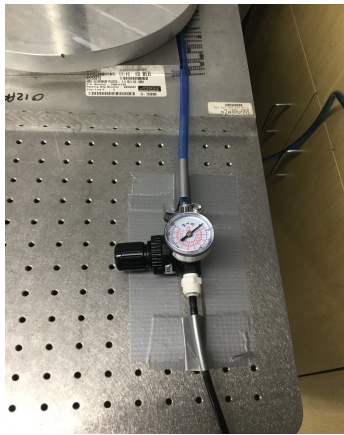


Figure 5.5. One of Two Pressure Gauge Setups.

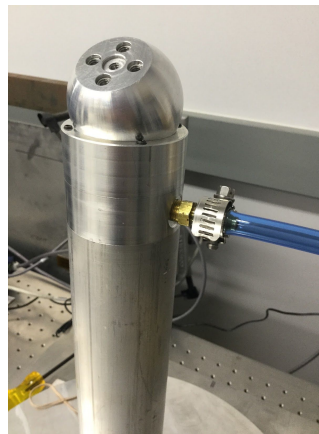


Figure 5.6. Airline Connected to Test Bed System.

The plate was squared off to provide an even weight distribution across the board. A counterbalance system, pictured in Figure 5.3, was designed for the test bed to improve stability. This design consists of four counterbored slots drilled into the platform along axes stemming from its center. Bolts are inserted into these slots and fitted with nuts to prevent sliding and cylindrical weights with threaded holes through their long axes. Finally, threaded holes are drilled from the sides of the platform into the counterbored portion of the slots. The slots allow for adjustment of the center of mass of the system in the plane of the platform by repositioning the bolts while the threaded weights allow for adjustment of the center of mass along the axis normal to the surface of the platform. Thumb screws inserted in the holes on the sides allow for fine adjustment of the center of mass. This adjustable counterbalance system is necessary because as components are added to the test bed, the center of mass will shift. It is essential that the center of mass remain aligned with or directly below the center of rotation to keep the system stable.

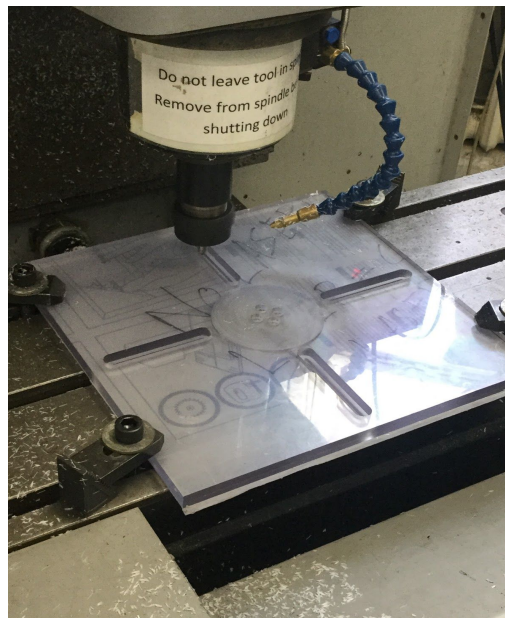


Figure 5.7. Remodeling of the Test Bed.

In the actual test bed, nuts were used as weights instead of machined blocks of aluminum as initially planned. The combined weight of the bolts and nuts proved adequate to balance the

platform with an empty 3D-printed CubeSat structure mounted atop. It is possible that more weight would be needed to lower the center of mass of the system were this CubeSat outfitted with the necessary equipment, describe below, to run the attitude control experiments.

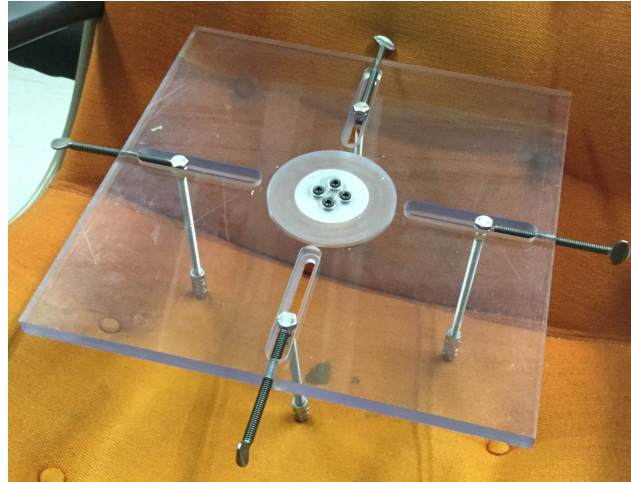


Figure 5.8. Final Test Bed.

When testing the new prototype design, much of the same instrumentation as last year should be used. The devices used were carefully selected for test bed prototype testing wirelessly. A list of devices used is included in Table 5.1. In addition to these devices, a 3D printed 4U CubeSat and reaction wheels will be added to the test bed. These devices allow for taking measurements, processing raw data, powering the system, and implementing an extended Kalman filter.

Table 5.1. Prototype testing instrumentation.

Instrument Type	Selected Device
Inertial Measurement Unit	Adafruit 9-DOF Fusion IMU Breakout
Microprocessor	Adafruit Feather HUZZAH with ESP8266 WiFi
Battery	EFOSHM Ultra Slim Battery
Computer	Raspberry Pi 3

5.2.2 Attitude Determination and Control Components

Attitude determination is the process of computing the orientation of the spacecraft relative to either an inertial reference or some object of interest such as the Earth [13]. This typically involves several types of sensors on each spacecraft and sophisticated data processing. The accuracy limit is usually determined by a combination of processing procedures and spacecraft hardware which will be discussed momentarily.

Attitude control is the combination of the prediction of and reaction to a CubeSat rotational dynamics. It consists of two areas, attitude stabilization which is the process of maintaining an existing orientation, and attitude maneuver control which is the process of controlling the reorientation of the spacecraft from one attitude to another [4]. In order to stabilize the CubeSat, it must have a system that keeps it moving evenly to provide stable rotation and flight. Spinning or gyroscopic motions are used to keep the CubeSats motion stable. Sensors are used to measure the state of the system, while thrusters and reaction wheels are very common components that are used to adjust the state of the system.

For this project, measuring components included an Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055 and a Adafruit Feather HUZZAH with ESP8266 WiFi microcontroller.

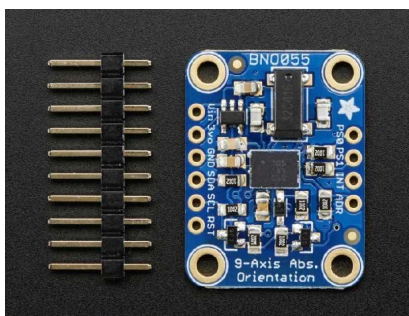


Figure 5.9. Adafruit IMU [34]

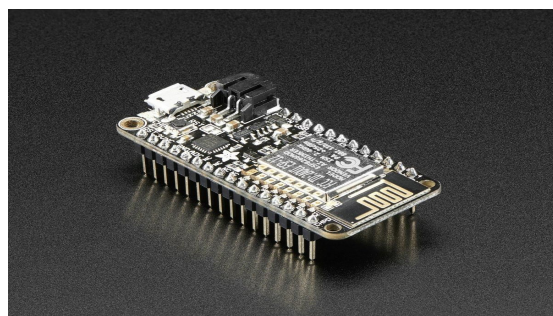


Figure 5.10. Microcontroller [34]

The inertial measurement unit (IMU) is a device that measures and reports a body's specific force, angular rate and magnetic field. In a navigation system, the data reported by the IMU is fed into a processor which calculates attitude, velocity and position. The IMU integrates angular rate from the test bed to calculate angular position while a Kalman filter estimates the attitude to transform acceleration measurements into an inertial reference frame where they are integrated once to get linear velocity, and twice to get linear position [8]. This will be connected to the heart of the CubeSat, the microcontroller. The microcontroller manages the interfaces with sensors, actuators and the onboard computer and performs the control tasks.

Furthermore, to provide stable motion, actuatorial devices such as the EMAX RS2205 2300KV brushless motor shown in Figure 5.11, and designed reaction wheel will be used.



Figure 5.11. EMAX RS2205 2300KV Brushless Motor [33]

For this test bed, there will be a 45 degree maximum rotation in the x and y axis, due to the counterbalance weights interacting with the test tube, and a 360 degree rotation about the Z-axis. Three brushless motors will be used to provide rotation about all three axes. The EMAX RS2205 motors were chosen due to precise and fast operation to accelerate in both loaded and unloaded conditions and its small size. The decision to use reaction wheels over thrusters came down to which provided the simplest solution. Reaction wheels are commonly used for attitude and controls test purposes as seen by the Air Force Institute of Technology, and MIT Lincoln Laboratory. If thrusters were to be used, some form of gas would be the stabilizing force, and

more complicated mechanism would have to be devised to open and close the release valve to have the air exit the tanks. Overall, we thought the mechanism would be too complicated to devise and it would cost more money to resupply the tanks when empty. Reaction wheels are easy to manufacture, they last longer and are easy to handle.

Other components used to assist the actuarial and measurement devices were a 1.8 Turnigy LiPo Battery, power bank, power distribution board (PDB), and three DJI Snail 430-R Racing ESC.



Figure 5.12. 1.8 Turnigy LiPo Battery [30]



Figure 5.13. Power Bank [33]



Figure 5.14. Power Distribution Board[33]



Figure 5.15. DJI Snail Racing ESC [27]

The Li-Po battery will be used to power the motors and the power bank is to power the microcontroller. The PDB will be used in directing the prescribed motor voltage to prevent the motors from burning out. Lastly, the DJI ESCs will be used to control the brushless motors

speed. These ESCs allow for rotation to be given in the clockwise and counterclockwise direction, which will allow the test bed in the experimental phase to be able to rotate in the positive and negative directions.

5.2.3 CubeSat Printing Model

To test the attitude determination and control we needed to have something that could contain and hold all the actuator and measurement devices. Originally the components were going to be laid at the center of the test bed while the counter weights would do the bulk of the balancing. Quickly we realized this would cause a great hassle in trying to keep everything from sliding of the plate as well as keeping the counter balances consistent in their position relative to the weight distribution of the materials. This led to the idea of creating some form of a container to place all the objects. To keep the test as close to the actual model, a designed 3D printed CubeSat was created.



Figure 5.16. 3D Printed CubeSat.

Initially, a prototype CubeSat was printed. This was created so that a physical model was present and a general idea of the size and free space could be seen. The final model was created

in Higgins Labs. The material used was acrylonitrile butadiene styrene, because it is a strong impact resistant material, has a high tensile strength, and is shock absorbent.

There were three components that went to the overall assembly, which included the skeletal CubeSat frame, a battery pack holder for the Li-Po battery, and a shelf that could hold the battery pack, ESCs, PDB, microcontroller, and IMU.

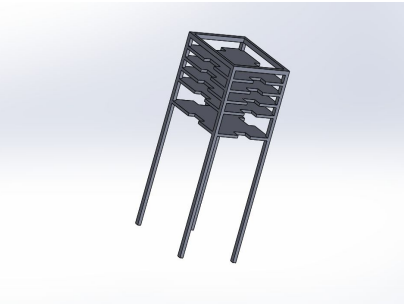
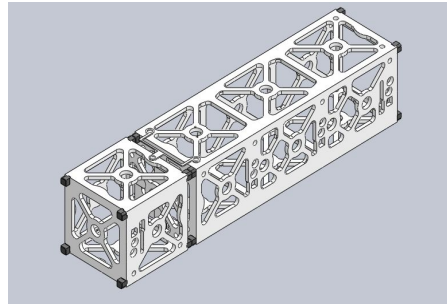
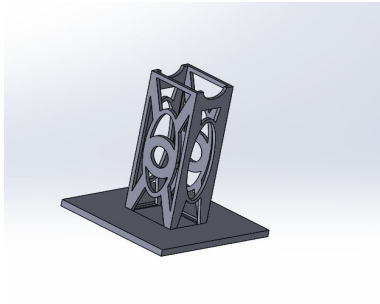


Figure 5.17. Battery Pack.

Figure 5.18. 3D CubeSat Model.

Figure 5.19. Shelf.

As seen from above the design of the CubeSat frame was equivalent to the eLEO model. The eLEO CubeSat is only a 4U CubeSat compared to the formation flying 16U and 20U models, making it much smaller and easier to fit. The only difference between the 3D printed model and the eLEO model was there had to be holes made on the side where the motors would be attached and the thickness of the frame had to be greater to provide the necessary strength to hold everything together.

To print these components, they had to be designed in SolidWorks as separate pieces that would have to be assembled together. There were twenty-one separate parts made for the CubeSat, six parts for the battery pack holder, and ten parts for the shelf making a total of thirty-seven parts. By having the assembly of the CubeSat and test bed we could gather a general idea of our moments of inertia. This would help in the balancing of the CubeSat as well as creating the necessary programming for the Kalman filter with feedback and PID controller. The completed SolidWorks model, shown in Figure 5.20, included the CubeSat frame, internal

electronics, and the test bed. This was used to obtain the moments of inertia of the plate and CubeSat which can be seen in Figure 5.21.

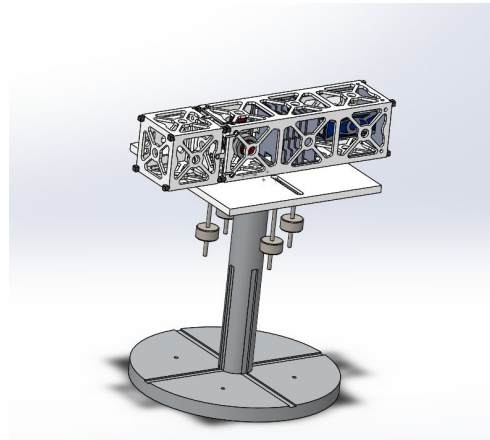


Figure 5.20. Complete Assembly.

Center of mass: (inches)		
X = 9.25		
Y = 9.94		
Z = 9.14		
Principal axes of inertia and principal moments of inertia: (pounds * square		
Taken at the center of mass.		
lx = (0.01, 1.00, 0.01)	Px = 1198.29	
ly = (-0.72, 0.01, -0.70)	Py = 2305.96	
lz = (-0.70, 0.00, 0.72)	Pz = 2353.61	
Moments of inertia: (pounds * square inches)		
Taken at the center of mass and aligned with the output coordinate system.		
Lxx = 2328.96	Lxy = 9.17	Lxz = 23.90
Lyx = 9.17	Lyy = 1198.47	Lyx = 10.68
Lzx = 23.90	Lzy = 10.68	Lzz = 2330.44

Figure 5.21. Moment of Inertia and Center of Mass.

5.2.4 Reaction Wheel Design

Reaction wheels are widely used for as actuation system in spacecraft. They operate on the principle of conservation of angular momentum; applying a torque to a wheel about one axis causes the satellite to rotate in the opposite direction about the same axis. One drawback of reaction wheels is that they can become saturated, meaning they have reached the peak angular rate and torque that the motor can supply, if they are constantly correcting a disturbance torque.

For this project, three reaction wheels were constructed using the same design. These reaction wheel were to be placed orthogonally, allowing for full control about all three axes of rotation. In practice, a fourth spare wheel is often added at a skew to continue to allow for control if one wheel fails. However, due to the ease with which a reaction wheel can be replaced in the test bed setup, this redundant wheel was not implemented.

In order to appropriately size the reaction wheels, Larson and Wertz was consulted [15]. This textbook offers various methods of sizing reaction wheels. These methods include sizing the reaction wheel based on disturbance torques, required momentum storage, or slew maneuvers. Due to the current lack of disturbance torques in the design and the ability to reset the experiment, the method of sizing based on slew maneuvers was used, based upon the equation:

$$T = 4\theta I/t_{dur}^2,$$

where T is the required torque to maneuver a spacecraft of moment of inertia I through an angle θ in time t_{dur} .

In order to use this method, a preliminary motor selection and flywheel design had to be made. These were to be iterated upon based on the torque the motor could supply and the speed at which the slew maneuver could be executed with the moment of inertia of the flywheel. This process had to be iterative because any adjustments to the flywheel also altered the moment of inertia of the entire spacecraft and test bed system. As a baseline, it was desired that the attitude control system be able to turn the system by 90° in 30 seconds. It was found that with the initial reaction wheel design, a torque of 0.349 mN-m would be required to complete the maneuver and that the maximum rotational speed of the reaction wheel would be 5905 rpm. These numbers were well within the capability of the selected motor and included a small safety margin as the spacecraft moment of inertia was rounded up from about 0.0449 kg-m² to 0.05 kg-m² to account for components yet to be added to the design. A summary of these calculations can be found in Appendix B. Unfortunately, due to time constraints, these numbers were not recalculated for the final design. However, with electrical input to the motor of 1 A at 12 V, there was a safety factor of above 44. Therefore, the motor could almost certainly supply the required torque; the only issue might be if the wheels become saturated and a coast period is required to complete the maneuver.

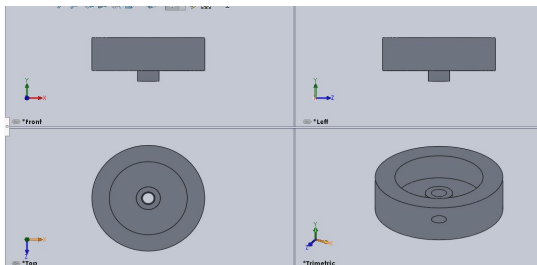


Figure 5.22. SolidWorks View of Wheel.



Figure 5.23. Machined Model.

Unfortunately, a mistake was made in the final steps of machining the flywheels and the inside right hand threading of the wheel turned out to be incompatible with the motor shafts left hand threading, rendering the reaction wheels unusable.

5.2.5 Test Bed Power Budget

To make sure that our testbed attitude system would operate correctly, we created a power budget for the different components in the system. However, it wasn't just one battery powering everything, instead there were two batteries used. One was a Turnigy 1800mAh 4S 40C LiPo battery, which was used to power the motors, and the other was a EFOSHM Ultra Slim 2600 mAh power bank battery that powered the microcontroller and the raspberry pi. The Turnigy battery has a life of forty hours per charge with continuous use, and a recharge time of five hours, and could output up to 14.8 V, which as seen in Figure 5.24, was more than the maximum of 14 V used by the motor [30]. It is also important to note that the 14 V max of the motors is not where the motors would operate when the system was in use.

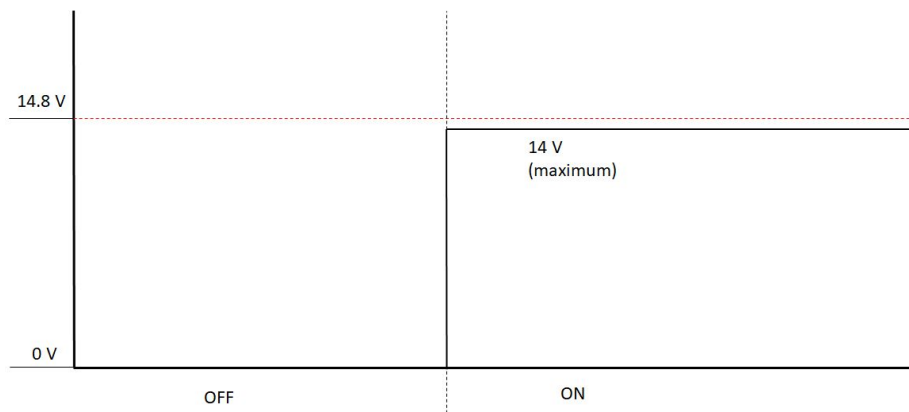


Figure 5.24. Voltage Usage for Turnigy LiPo Battery [30].

The most important factor of the power bank battery on the other hand, was the output of amps. As shown in Table 5.2 and Figure 5.25, the 1 A output was more than enough to power the microcontroller and raspberry Pi. The PDB and the DHI Snail 430-R Racing ESC also did not draw any power from the batteries, so they weren't included in the power budget [27].

Table 5.2. Power Budget for Power Bank Battery [23].

COMPONENT	POWER USED
Microcontroller	500 mA
Raspberry Pi Model 3 B	250 mA max

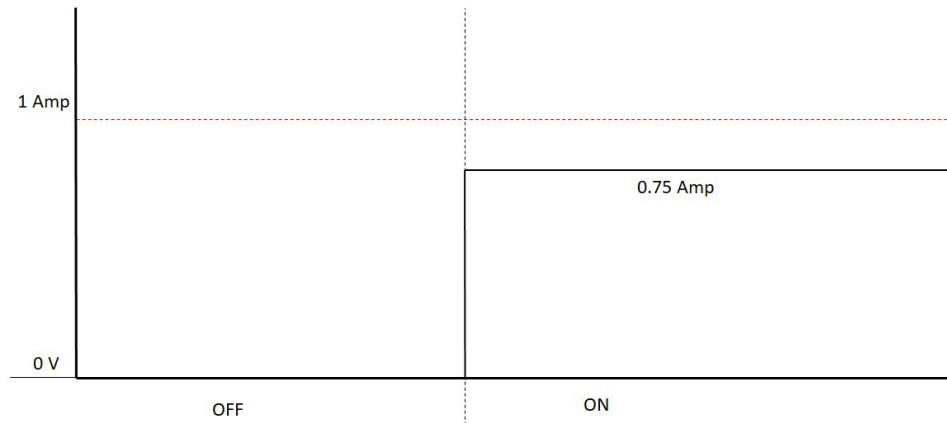


Figure 5.25. Power Budget for Power Bank Battery [23].

5.2.6 PID Control for Reaction Wheels

An actuation system cannot operate properly without a control scheme. A control scheme was therefore designed and operation was simulated in MATLAB. Initially, this scheme was only able to execute the slew maneuver governed by the equation in section 5.2.4. Here, this scheme will be called the “simple slew command scheme.” This scheme was considered somewhat unreliable, however, as it does not rely on any sensor inputs and as such errors in the maneuver could not be detected. Additionally, this scheme did not allow for an attitude hold mode. As a result, a PID controller for the attitude control system was designed.

The PID controller has several benefits over the simple slew command scheme discussed above. A PID controller is designed to respond to error; this lends itself well to an attitude hold mode. Disturbances to the desired state are considered errors in the PID controller and are automatically corrected. To perform a slew maneuver, a desired angle can be set for each axis of rotation. The difference between the current and desired attitude is then considered error by the PID controller and the attitude control system responds. Correctly tuning the PID controller also allows for a dynamic output torque, rather than the static torque outputted by the simple slew command scheme. In order to increase fidelity, the output torque of the PID controller was limited to 0.5 mN-m, slightly above the torque required to maneuver the system 90° in 30 seconds. The PID controller thus allows for the most rapid maneuver within the torque

constraints, whereas the simple slew command scheme also requires an input of maneuver duration, potentially resulting in suboptimal slew speed or unrealistic output torques. For comparison, results of simulations of the PID control scheme and the simple slew command scheme executing the same maneuver are shown in Figures 5.26 and 5.27.

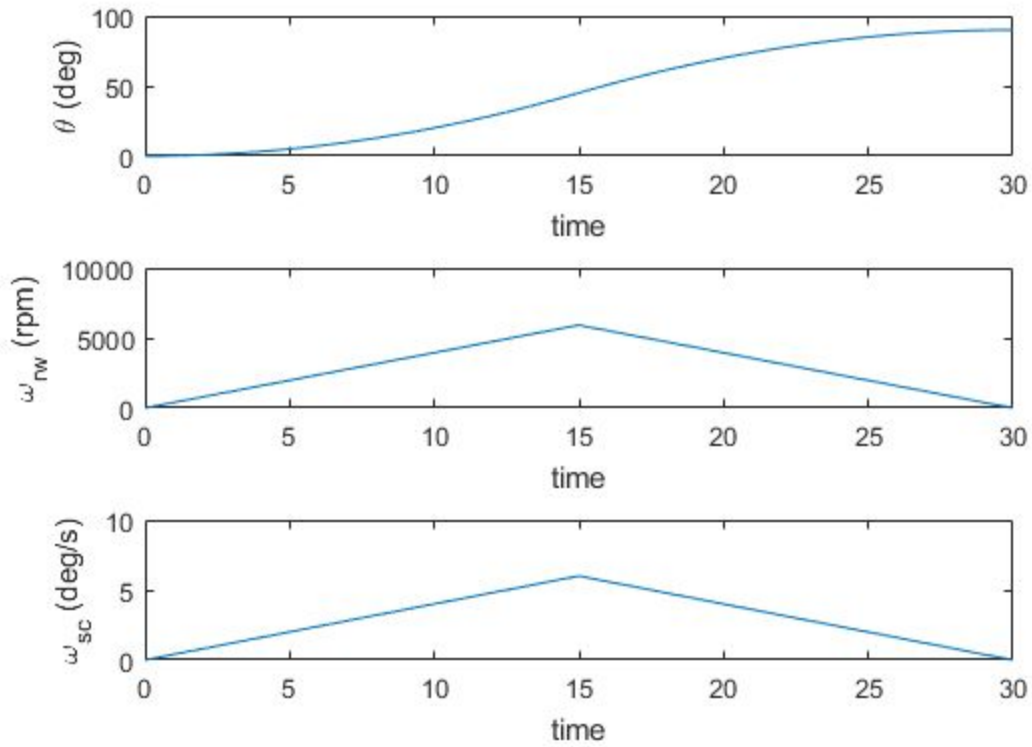


Figure 5.26. The simple slew command scheme completes a maneuver of 90° in 30 seconds.

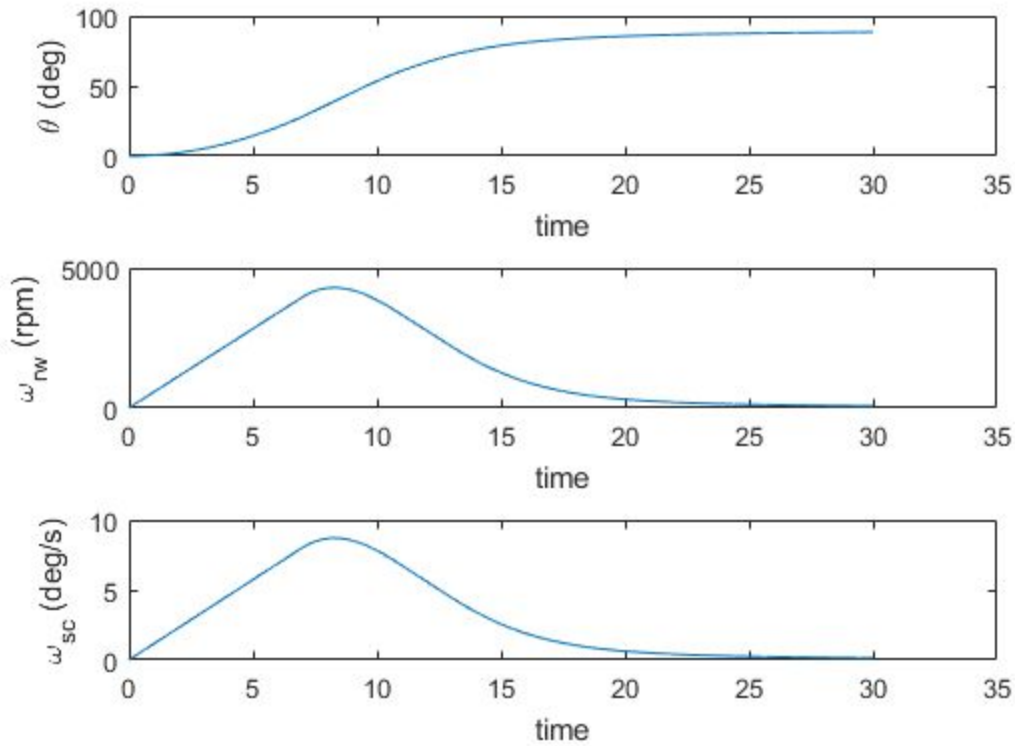


Figure 5.27. The PID controller executes a 90° maneuver without duration specified.

The PID controller was further developed for control about all three axes of rotation. In the context of the experiment, the x and y axes of the body fixed coordinates lie in the plane of the platform, while the z-axis extends normal to the plane of the platform. The IMU in the setup will measure deviations from initial position about these axes. Therefore, the desired angle for rotation about the x and y axes would be set to zero to keep the platform level, and the desired angle for rotation about the z-axis could be altered to perform slew maneuvers. Since the IMU reports not only angular rates but also the angles that describe the attitude of the system, measurements collected could be taken directly as inputs without the need for coordinate transformations using Euler equations.

A sample simulation of the three axes PID control is provided in Figures 5.28 and 5.29. The initial conditions for this simulation were set to the values shown given in Table 5.3. Initial spin rates and disturbance torques were set to zero because of the current lack of capacity to introduce these in the physical test bed setup.

Table 5.3. Initial conditions for sample 3 axes PID control simulation.

Axis of Rotation	Initial Angle (degrees)	Desired Angle (degrees)	Initial Spin Rate (degrees/s)	Disturbance Torque (N-m)
x-axis	-6	0	0	0
y-axis	18	0	0	0
z-axis	60	-30	0	0

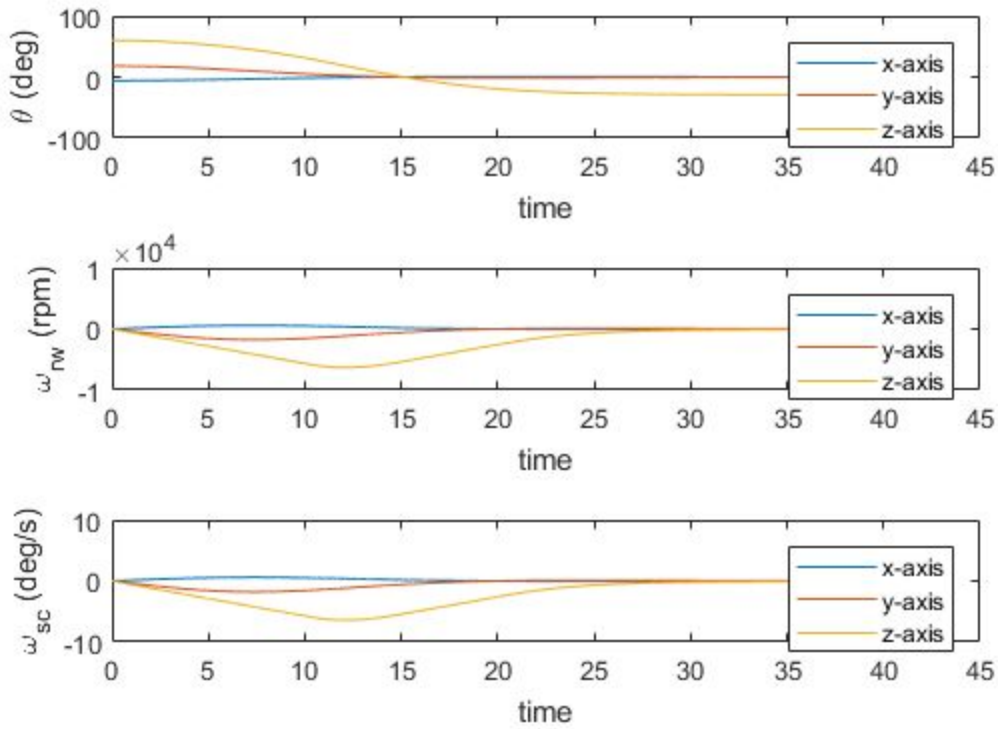


Figure 5.28. States of simulation over time. θ represents the spacecraft angle, ω_{rw} represents the spin rate of the reaction wheels, and ω_{sc} represents the spin rate of the entire system.

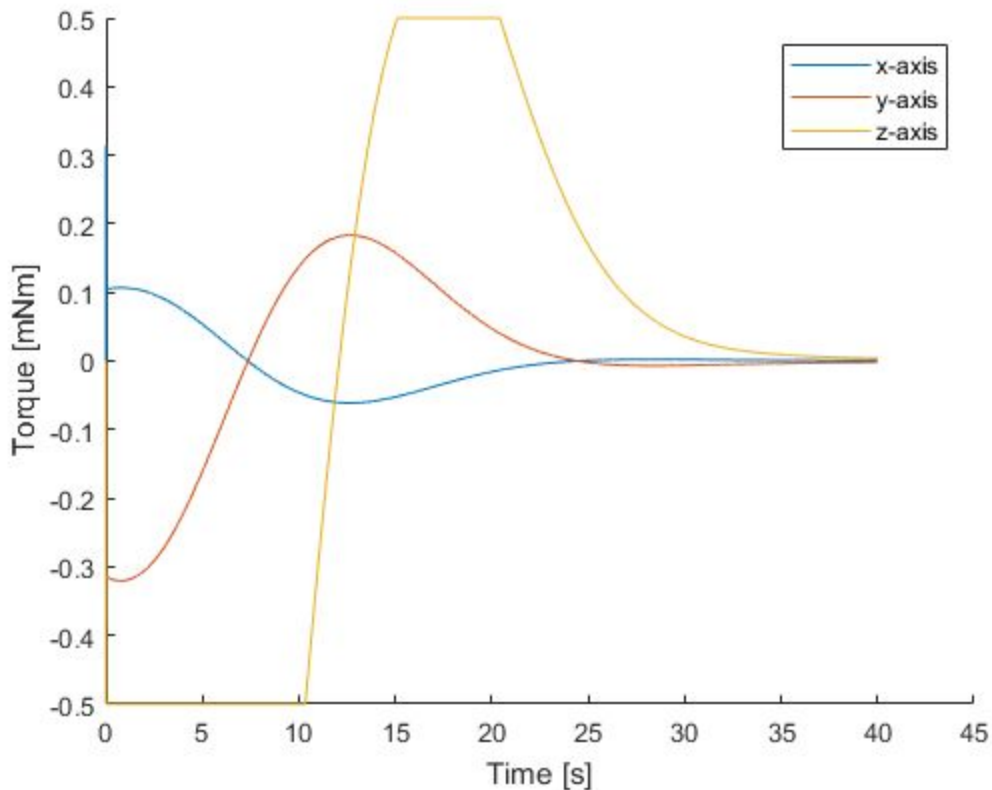


Figure 5.29. Output torques of each reaction wheel. The z-axis reaction wheel torque shows the system’s capability to limit output torque.

5.3 Future Recommendations

In the future, several improvements may be made to the current design of the test bed. As of now the test bed works and it is sufficient for early stages of experimentation. Although, to allow for further rotation in the x and y axis the counter weights can be shortened with the addition of heavier weights. Furthermore, a way of securely holding the CubeSat on the test plate would be beneficial. Currently duct tape is keeping the CubeSat from sliding as the plate rotates. One suggestion would be to drill holes in the to insert a screw or peg that is then inserted in the CubeSat for a secure hold. The tube supplying the air to lift the test plate also causes rotation restraints. Lastly, devising a mechanism to keep the weighted bolts at a fixed location would also be beneficial. When the plate had dramatic changes in its angle, the bolts would slip and slide in

their slots. A possible solution would be to insert spacers or even a spring to keep the bolts at a fixed point.

As for the setup, most of the parts required for full attitude control are ready for future teams, but require some assembly and programming not completed in this project. Future teams should also consider designing a system for introducing disturbance torques to better simulate real mission attitude control. One possible way to do this is to purposefully leave the counterbalance system incorrectly tuned to force the reaction wheels to compensate. However, this would be a time-invariant torque and would quickly lead to saturation of the reaction wheels.

A method of introducing transient disturbance torques would add a high degree of fidelity to experimentation with the test bed, possibly without needing to reset the test bed to allow for desaturation of the reaction wheels. One option is to include the fourth skew reaction wheel that exists on many actual spacecraft. However, instead of providing redundancy, this reaction wheel could provide a disturbance torque about all three axes simultaneously, forcing the other three reaction wheels to compensate. This fourth wheel could be programmed to vary its output torque, thus providing a transient disturbance torque.

The current control scheme, MATLAB script available in Appendix C, is not capable of being implemented in an actual experimental setup. This scheme is purely for simulation purposes. In the future, in order to allow for actual attitude control, the script should be modified so that the script takes inputs of angle, angular rate, and angular acceleration from the IMU, rather than calculating what the states should be. The output should be changed from a torque to pulse width modulation for control of the electronic speed controllers. Additionally, the current code has no capability to handle instrument noise. A low pass filter, Kalman filter, or both should be designed and added to the code as raw data from the sensors would cause the system to be constantly trying to correct itself when the observed error might only be a result of noise in the measurements. A method of detecting when the reaction wheels are saturated or close to saturation should also be implemented, this may involve a variable coast fraction for slew maneuvers.

Future teams may also wish to consider changing from the classical PID controller to the modern state space control scheme. Currently, three PID controllers are running together to

allow for control about three axes. A single state space scheme would be capable of handling all three axes of rotation as well as additional states, allowing for error calculation not only in orientation, but in angular rate. While the simulations seem to show a single input single output system performing correctly, in practice results may not be as expected.

6. Conclusions

6.1 Attitude Determination and Control Conclusions

For the attitude control, we found all the sensors necessary to create a functioning attitude determination and control system. We extensively researched the specs of each of the sensors to make sure they would work for each of our CubeSat mission simulations, as well as magnetorquers and μ PPT's for use as propulsion. However, due to the parameters of our project, we did not construct the physical control system.

Through the process of creating a physical model to test the attitude determination and controls we were able to create and gather the necessary components to make a controls system. A 3D printed CubeSat with necessary components were made to neatly place and organize on the test bed. Wireless communication with a the Xbee Wire Antenna and a Xbee Bluetooth USB to Serial Port Arduino Bee Adapter. Measurement sensors such as the IMU and microcontroller were studied and implemented to the CubeSat so the orientation and position of the CubeSat could be determined. Analysis was made to create and spec the appropriate actuating devices in order to provide smooth rotation about all three axes. A PID controller was successfully created to see how the CubeSat would rotate towards a stable state. Unfortunately, a multitude of issues came about during the final weeks of the project. As explained before, the reaction wheels had a thread issue causing a setback, the DJI Snail ESCs were lost in the shipping and delivery stage, and a Kalman filter with feedback was not created. As explained to us at the beginning of the project, the graduate assistants assigned to each of our mission groups were responsible for the creation of the attitude determination and control system. Since they have the knowledge of creating an extended Kalman filter for the attitude system, the task was given to them.

6.2 Command and Data Handling Conclusions

We were able to identify the five major stages of both CubeSat missions, and then create block diagrams detailing the actions that would need to happen within the OBC and its subsidiary systems to keep the CubeSat on mission. The operational block diagrams were largely

the same except for the routine portion of the mission, which is to be expected given that the two missions were different in nature.

The on/off states of all the sensors was successful in determining when the sensors would be using power, but for the most part once the CubeSats had entered the detumbling stage, every sensor was turned on for every other stage throughout. For the eLEO CubeSat, we looked into having a light and dark side phase, in which some sensors were turned off when the CubeSat was in eclipse, but we found that this had a negligible amount of power savings, as the main usage of the power generated from the solar panels was going to the motor and to maintaining attitude throughout the mission. Because of this, we stuck with the simple yet effective on/off stages and block diagrams we originally came up with.

For both missions there was more than enough space within the OBC for both volatile and non-volatile memory, and that the non-volatile memory storage of the OBC and EPS was more than enough for the targeted 90 day mission time. However, while both forms of memory had quite a large amount of free space left, it is important to note that this is something we wanted, as it allows the CubeSats to be used for various missions. The large amount of available memory allows a customer to input one of their sensors onto the CubeSat design and not have to worry about the maximum memory storage being surpassed during the mission.

6.3 Structural Analysis Conclusions

In all cases, the CubeSat structures appeared to hold up to the mission requirements. For the 4U CubeSat design fifty-one modes were found ranging between 33 Hz to 2,940 Hz. Due to the simplicity of the model, there were no issues with trying to mesh and run the simulation. For the 20U CubeSat design, a number of modes were found below the 100 Hz threshold set by the P-POD deployer. However, as discussed briefly in Chapter 4, this design is too large to be deployed by the P-POD deployer. It is likely that a more massive deployer would have a lower natural frequency, but resonance should certainly be an area of concern when selecting a deployer. Because of the large difference between the four modes below 100 Hz and the next

mode, it is possible that these low frequency modes are simply a result of an error in the simulation.

In terms of deformation and stress, the CubeSat designs all performed excellently. The greatest deformation of the 4U design occurred along the y-axis at 0.26 mm, being less than a half a millimeter deformation. For the 20U design, the greatest deformation occurred in the z-axis being less than a 2 mm deformation. Finally, in the 16U design, the greatest deformation occurred along the y-axis and had a magnitude of just over 0.1 mm. Across all simulations, the maximum equivalent (von Mises) stress remained well below the yield strength of the aluminum 7075-T6 used for the structure of the eLEO and formation flying CubeSat missions. The greatest stresses also occurred in the y axis for the 4U and along the z-axis of the 20U design. In the 20U design this only reached 61% of the yield strength of the material. This number is somewhat high, but still leaves a comfortable safety margin.

6.4 Attitude Determination and Control Experiment Design Conclusions

The reformation of the test bed was a success. The test bed was squared off and milled to the correct dimensions in the machine shops of Higgins Labs. Counter weights were purchased and fitted on the test bed to create a level surface when suspended in the air. An image of the completed setup is visible in Figure 6.1.

Initially, a simple test was performed to check the stability of the test bed. A force was induced on the plate to cause a disturbance, slowly over time the plate rotated and balanced itself to a steady equilibrium position. Then a test was conducted with the prototype 3D printed model placed on the test bed in a horizontal position. The CubeSat's moments of inertia and center of mass, calculated in SolidWorks as shown before, were used to get a general idea of where to position the CubeSat on the plate for balance.



Figure 6.1. Prototype CubeSat and Test bed at Equilibrium.

A horizontal position was chosen for the CubeSat to keep the center of mass low and thus make balancing the platform easier. Once the plate and CubeSat were balanced, the plate was disturbed, and it did return to a level steady state, showing that the counterbalance system was successful.

The incorporation of the two separate pressurized air lines was also a success. Both air lines supplied sufficient air to the vibration free table as well as the test bed. The pressure reading used was equivalent to the total weight of all the components and test bed. In the end the only components used were the prototype CubeSat and the test bed weight since the complete assembly could not be put together in time. Hence, the weight calculated came to be 6.35 lbs, so 6.35 lbs of pressure through the tube.

Works Cited

1. *3rd Generation 3U EPS*. Retrieved from <https://www.clyde.space/products/5-3rd-generation-3u-eps>
2. *40 whr CubeSat Battery*. Retrieved from <https://www.clyde.space/products/49-40whr-cubesat-battery>
3. Agolli, J., Gadoury, J., & Rathbun, A. (2017). *Design and Analysis of the Sphinx-NG CubeSat*. (). WPI: *Aluminum 7075-T6; 7075-T651*.
<http://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MA7075T6>
4. “Attitude Determination and Control Systems.” NASA, NASA, ntrs.nasa.gov/search.jsp?R=20110007876.
5. Billings, D., Graedel, I., Hoey, F., Martinez, N., & Torres, J. (2013). *Design and analysis for a CubeSat mission*
6. Boynton, R. (1996). *Using A spherical air bearing to simulate weightlessness*
7. *CSS-01,02 Coarse Sun Sensors*.(2015). Space Micro.
8. Curci, E., Jacobson, J., Schlack, W., & Slabinski, K. (2017). *Design and analysis of the sphinx-NG CubeSat*
9. *Emax Rs2205-2300 2205 2300kv Racing Edition Cw/ccw Motor For Rc Drone Fpv*.
Retrieved from https://www.banggood.com/Emax-RS2205-2300-Racing-Edition-CWCCW-Motor-For-FPV-Multicopter-p-1028793.html?ID=49005&cur_warehouse=USA
10. *Fine (digital) Sun Sensor*. New Space Systems.
11. GENERAL ENVIRONMENTAL VERIFICATION STANDARD (GEVS) For GSFC Flight Programs, & Projects. *GSFC-STD-7000A 4/22/2013 supersedes GSFC-STD-7000*
12. *High Performance, Digital Output gyroscope*. (2011). Analog Devices.
13. Hall, C. (2003). *Attitude Determination*. [ebook] Virginia. Available at:

<http://www.dept.aoe.vt.edu/~cdhall/courses/aoe4140/attde.pdf> [Accessed 24 Feb. 2018].

14. Hyon Ko, D., Laudage, S., Murphy, M., Pelgrift, D., & Young, S. (2017). *Design and Analysis of Sphinx-NG CubeSat*. (). WPI:
15. Larson, W. J., & Wertz, J. R. (1999). *Spacecraft mission analysis and design* (Third Edition ed.) Microcosm Press and Kluwer Academic Publishers.
16. Mabrouk, E. (2015). *What are SmallSats and CubeSats?* Retrieved from <http://www.nasa.gov/content/what-are-smallsats-and-cubesats>
17. *Magnetorquer* (2017). Retrieved from <https://en.wikipedia.org/w/index.php?title=Magnetorquer&oldid=778676722>
18. *Nanosatellite On Board Computer*. Retrieved from <https://www.clyde.space/products/3-nanosatellite-on-board-computer>
19. *NSS Magnetometer*. Retrieved from <https://www.cubesatshop.com/product/nss-magnetometer/>
20. *Pulsed Plasma Thruster* (2017). Retrieved from https://en.wikipedia.org/w/index.php?title=Pulsed_plasma_thruster&oldid=77772464
21. *Pumpkin CubeSat Kits*. Retrieved from <https://www.cubesatshop.com/product/pumpkin-cubesat-kits/>
22. Quadrino, M., Miller, D. and Cahoy, K. (2014). *Testing the Attitude Determination and Control of a CubeSat with Hardware-in-the-Loop*. [ebook] Massachusetts Institute of Technology. Available at: <http://ssl.mit.edu/files/website/theses/SM-2014-PrinkeyMeghan.pdf> [Accessed 24 Feb. 2018].
23. *Raspberry Pi 3 Model B*. Retrieved from <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
24. *SGR-05U Space GPS Receiver*. Retrieved from <https://www.sst-us.com/shop/satellite-subsystems/global-positioning-systems-gps-receivers/sgr-05u-space-gps-receiver>

25. SkyFox Labs. (2017). *piNAV-NG Datasheet*
26. SkyFox Labs (2018). *piNAV-LI Product Datasheet*.
27. *Snail 430-R Racing ESC*. Retrieved from
<http://store.dji.com/product/snail-430-4-racing-esc>
28. STMicroelectronics. (2011). *LSM303DLHC datasheet*
29. Tibbs, M. (2015). *DESIGN AND TEST OF AN ATTITUDE DETERMINATION AND CONTROL SYSTEM FOR A 6U CUBESAT USING AFIT'S CUBESAT TESTBED*. [ebook] Wright-Patterson Air Force Base. Available at:
<http://www.dtic.mil/dtic/tr/fulltext/u2/a618506.pdf> [Accessed 24 Feb. 2018].
30. *Turnigy 1800mAh 4S 40C Lipo Pack*. Retrieved from
https://hobbyking.com/en_us/turnigy-1800mah-4s-40c-lipo-pack.html
31. Walters, J. (2016). EFOSHM ultra slim 2600 mAh power bank review. Retrieved from
<https://macsources.com/efoshm-ultra-slim-2600-mah-power-bank-review/>
32. Wukelic DE. Air Bearing ADCS Test Bed. Hawaii Space Flight Laboratory; 2009
33. “Radio Control Planes, Drones, Cars, FPV, Quadcopters and More.” *Hobbyking*,
hobbyking.com/en_us/?gclid=CjwKCAiA_c7UBRAjEiwApCZi8RuliKCKhzeo66ob9FsW2FAVld9t12SkQHGkQRj6xz52DH4hokuJvBoCedQQA_vD_BwE&gclidsrc=aw.ds.
34. Industries, Adafruit. “{{{Name}}} {#Model}} - {{{Model}}} {{/Model}}.”
Adafruit Industries Blog RSS,
www.adafruit.com/?gclid=CjwKCAiA_c7UBRAjEiwApCZi8e6Y3sK1e_Tbf7fD_WsIm-lzVK_lvtDhkSECHX5QVQ7p2nDGEWT95qBoCDQMQA_vD_BwE.

Appendix A: Non-Volatile Memory Plots

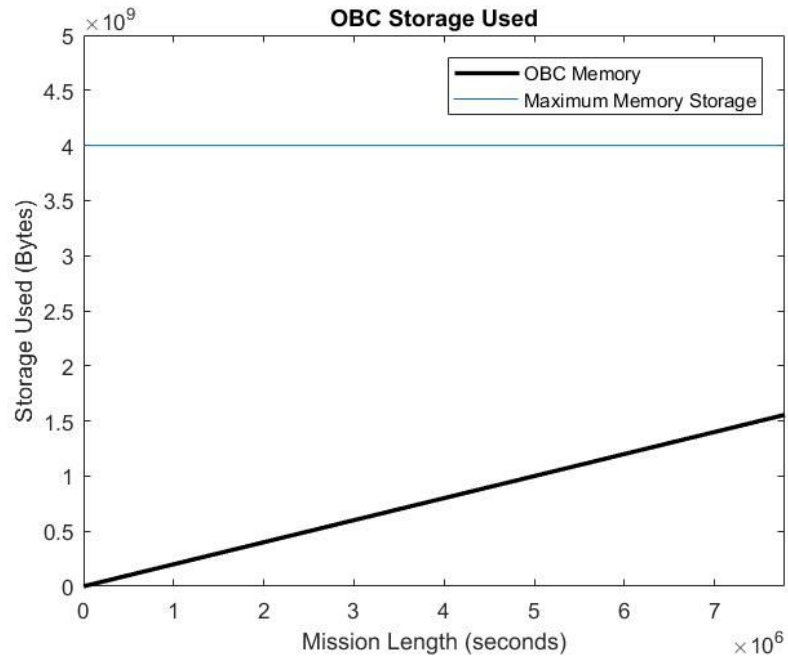


Figure A.1. MATLAB OBC Non-Volatile Memory Plot.

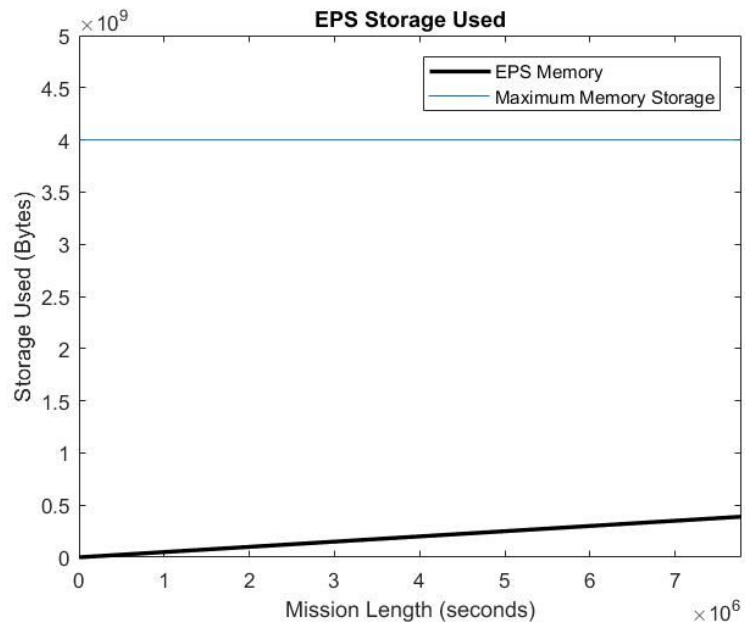


Figure A.2. MATLAB EPS Non-Volatile Memory Plot.

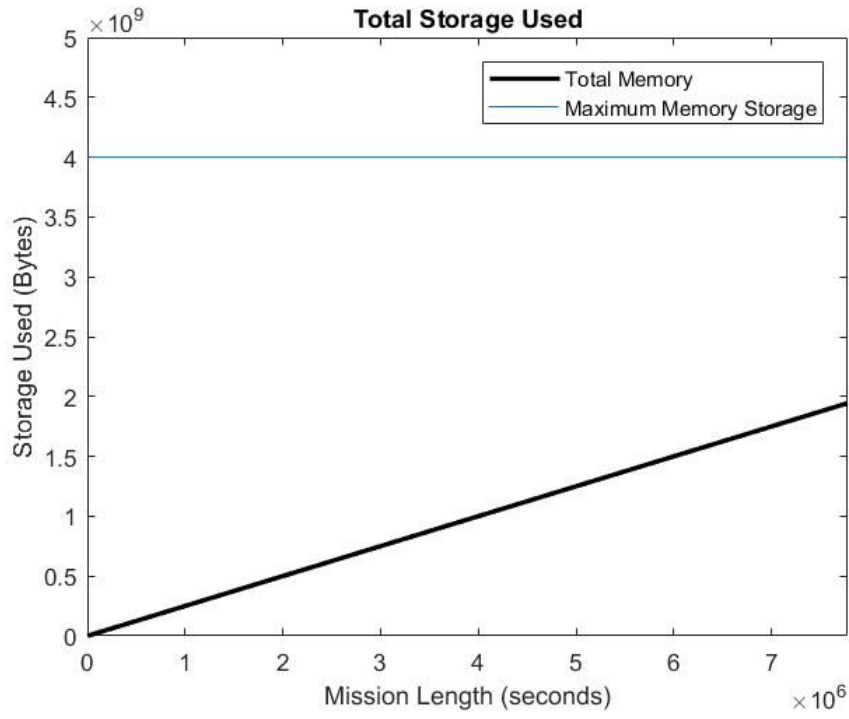


Figure A.3. MATLAB Total Non-Volatile Memory Plot.

Appendix B: Reaction Wheel Sizing

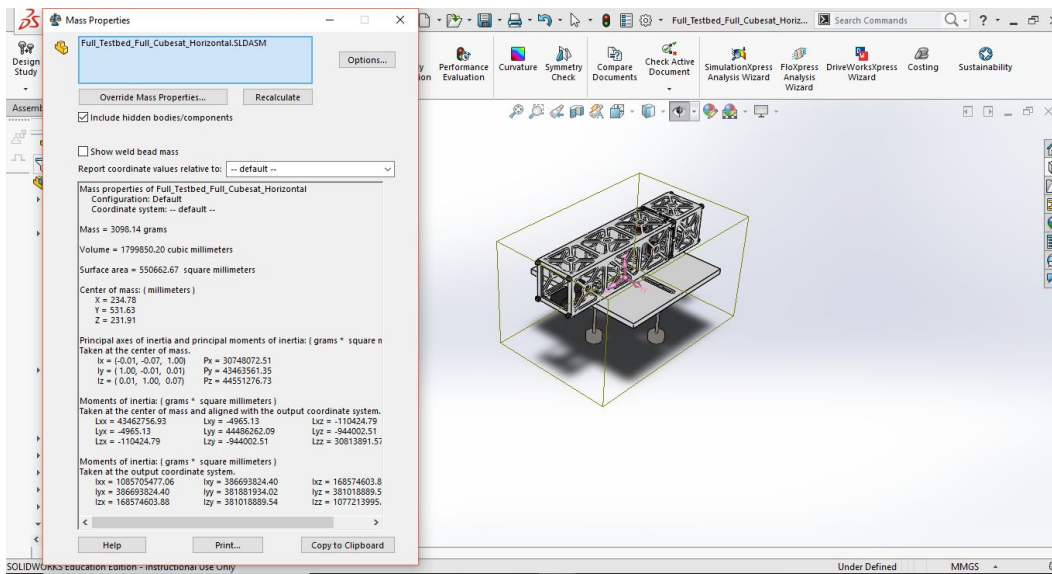


Figure B.1. Moment of Inertia for the rotating portion of the test bed system. The moment of inertia of chief interest, and conveniently the greatest, here is $I_{SC} = L_{yy} = 0.0477 \text{ kg-m}^2$.

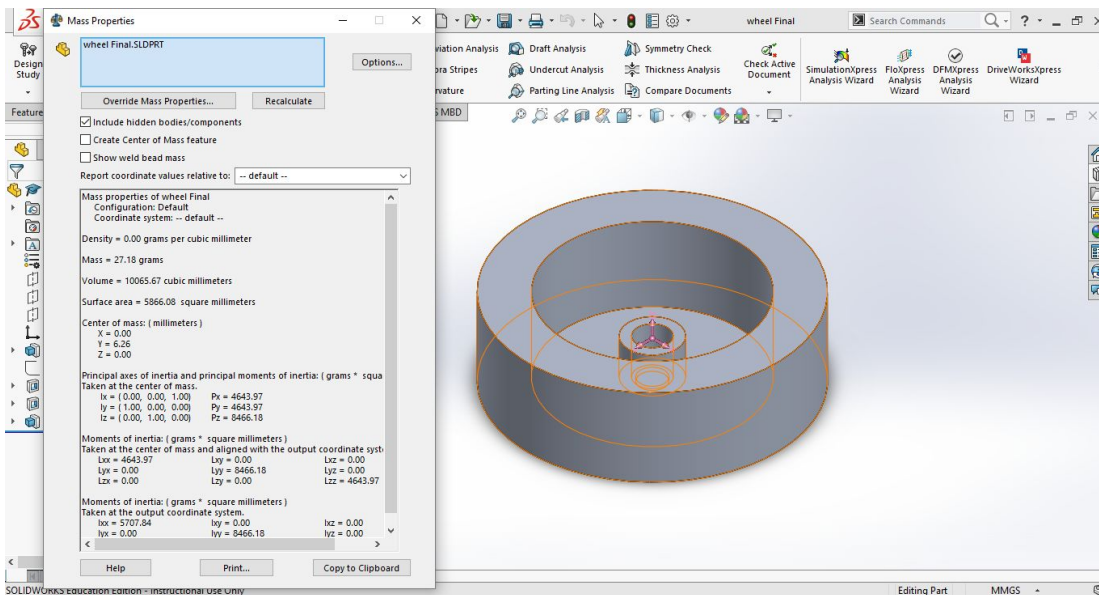


Figure B.2. Moment of Inertia for the flywheels. The moment of inertia of interest for here is $I_{RW} = L_{yy} = 8466.18 \times 10^{-9} \text{ kg-m}^2$.

For reaction wheel sizing, take maneuver angle $\theta = 90^\circ = \pi/2$ rad in time $t_{dur} = 30$ s. To complete this maneuver with no coast period, average angular $\omega_{ave} = 3^\circ/s$ and peak angular rate $\omega_{peak} = 6^\circ/s$. Using equation for torque requirement from Larson and Wertz [15] and substituting in these values with spacecraft moment of inertia increased to 0.05 kg to reflect the need to add more parts to the model:

$$T = 4\theta I/t_{dur}^2 = 4 * \pi/2 * 0.05/30^2$$

$$T = 0.349 \text{ mN} - \text{m}$$

Thus, the required torque to complete this maneuver is 0.349 mN-m. The acceleration of the reaction wheel is:

$$\alpha_{RW} = T/I_{RW} = 3.49e-4/8466.18e-9$$

$$\alpha_{RW} = 41.23 \text{ rad/s}^2 .$$

With a constant angular acceleration of 41.23 rad/s² for half of the duration of the maneuver, followed by and equal deceleration for the remaining duration, kinematics show the peak angular velocity of the reaction wheel is:

$$(\omega_{RW})_{peak} = \alpha_{RW}(t_{RW}/2) = 41.23 * (30/2)$$

$$(\omega_{RW})_{peak} = 618 \text{ rad/s} = 5906 \text{ rpm} .$$

The selected motor should certainly be capable of spinning the 27 g reaction wheel at 5906 rpm, given that it can spin a propellor to produce a kilogram of thrust at above 20,000 rpm given enough power [9]. Using conservation of angular momentum, the peak angular velocity of the test bed system can be found from the moments of inertia of the wheel and system along with the wheel's peak angular velocity:

$$(\omega_{SC}) = (\omega_{RW})_{peak} * I_{RW} / I_{RW} = 618 * 8466.18e-9/0.05$$

$$(\omega_{RW})_{peak} = 0.1047 \text{ rad/s} = 6^\circ/s .$$

This checks out with the peak angular rate that was logically deduced from the maneuver specifications. All that is left is to check that the motor can supply the necessary power. Since angular acceleration and moments of inertia are constant, the peak output power of the motor can be calculated as

$$(P_{motor})_{peak} = T(\omega_{RW})_{peak} = 3.49e-4 * 618$$

$$(P_{motor})_{peak} = 0.215 \text{ W} .$$

Brushless motors generally have an efficiency of about 85% to 90%. In the calculations used here, a conservative estimate of $\eta = 80\%$ was used. Using an input of 12 V at 1 A for an input power of 12 W, as seen in [9], the output power is estimated as

$$\begin{aligned} P_{motor} &= P_{out} = \eta P_{in} = 0.8 * 12 \\ P_{motor} &= 9.6 W . \end{aligned}$$

At 12 W, input, this motor supplies nearly 45 times the power needed.

Appendix C: MATLAB Script for 3-Axis PID Controller

pid_rw_3axis.m:

```
close all; clear all; clc

Isc = 0.05;           % s/c MoI estimate
Irw = 8446.18e-9;    % reaction wheel MoI
kp = 10e-4;          % proportional gain
ki = 5e-7;           % integral gain
kd = 2e-3;           % derivative gain
time = 0;            % total elapsed time
Dt = .01;            % time step
t(1) = time;         % begin recording times

%% x-axis Reaction Wheel
theta0x = pi/180*input('Input initial s/c x-angle: '); % initial s/c angle
theta_dx = pi/180*input('Input desired x-angle: '); % desired s/c
angle
w0x = pi/180*input('Input initial x-spin rate: '); % initial s/c spin
rate
Tdx = input('Input x-disturbance torque: '); % disturbance torque
disp(' ')
thetax = theta0x; % initialize s/c x-angle
errorx = theta_dx-thetax; % initial error

wscx = w0x; % initialize s/c x-spin rate
wrwx = 0; % initialize reaction wheel spin rate

omega_rwx(1) = wrwx; % record reaction wheel rate
omega_scx(1) = wscx; % record spacecraft rate
anglex(1) = theta0x; % record record spacecraft angle
errorTx = errorx; % total error for integral control term
error_priorx = 0; % initialize prior error for derivative control
termm

% This section is repeated twice below for the y- and z-axes.
% All variables mean the same as in this section, but have an 'x' in
% their names replaced with a 'y' or 'z' respectively.

%% y-axis Reaction Wheel
theta0y = pi/180*input('Input initial s/c y-angle: '); % initial s/c angle
theta_dy = pi/180*input('Input desired y-angle: '); % desired s/c
angle
w0y = pi/180*input('Input initial y-spin rate: '); % initial s/c spin
rate
Tdy = input('Input y-disturbance torque: '); % disturbance torque
disp(' ')
thetay = theta0y;
errory = theta_dy-thetay;
```



```

wscy = w0y;
wrwy = 0;

omega_rwy(1) = wrwy;
omega_scy(1) = wscy;
angley(1) = theta0y;
errorTy = errory;
error_priory = 0;

%% z-axis Reaction Wheel
theta0z = pi/180*input('Input initial s/c z-angle: '); % initial s/c angle
theta_dz = pi/180*input('Input desired z-angle: '); % desired s/c
angle
w0z = pi/180*input('Input initial z-spin rate: '); % initial s/c spin
rate
Td = input('Input z-disturbance torque: '); % disturbance torque
disp(' ')
thetaz = theta0z;
errorz = theta_dz-thetaz;

wscz = w0z;
wrwz = 0;

omega_rwz(1) = wrwz;
omega_scz(1) = wscz;
anglez(1) = theta0z;
errorTz = errorz;
error_priorz = 0;

%% Simulate
while t < 40
    i = i + 1;

    %% x-wheel
    proportionalx = kp*errorx; % proportional control term
    integralx = ki*errorTx; % integral control term
    derivativex = kd*(errorx-error_priorx); % derivative control term
    Tx = (proportionalx + integralx + derivativex); % output torque
    % Limit output torque
    if Tx > .5e-3
        Tx = .5e-3;
    elseif Tx < -.5e-3
        Tx = -.5e-3;
    end
    Tsx(i) = Tx;

    % Use equations of motion to calculate new states
    wrwx = Tx/Irw*Dt + wrwx;
    wscx = Tx/Isc*Dt + wscx + Tdx/Isc*Dt;
    thetax = thetax + wscx*Dt;
    % Update error terms
    error_priorx = errorx;
    errorx = theta_dx - wscx*time - thetax;
    errorTx = errorTx + errorx;

```

```

% Record for plotting later
omega_rwx(i) = wrwx;
omega_scx(i) = wscx;
anglex(i) = thetax;

% As before, the following two sections are for simulation in the
% y- and z- axes, again with the variables names changed to denote
% which axis they refer to.
%% y-wheel
proportionaly = kp*errorry;
integrally = ki*errorTy;
derivatively = kd*(errorry-error_priory);
Ty = (proportionaly + integrally + derivatively);
if Ty > .5e-3
    Ty = .5e-3;
elseif Ty < -.5e-3
    Ty = -.5e-3;
end
Tsy(i) = Ty;

wrwy = Ty/Irw*Dt + wrwy;
wscy = Ty/Isc*Dt + wscy + Tdy/Isc*Dt;
thetay = thetay + wscy*Dt;
error_priory = errorry;
errorry = theta_dy - wscy*time - thetay;
errorTy = errorTy + errorry;

omega_rwy(i) = wrwy;
omega_scy(i) = wscy;
angley(i) = thetay;

%% z-wheel
proportionalz = kp*errorrz;
integralz = ki*errorTz;
derivativetz = kd*(errorrz-error_priorz);
Tz = (proportionalz + integralz + derivativetz);
if Tz > .5e-3
    Tz = .5e-3;
elseif Tz < -.5e-3
    Tz = -.5e-3;
end
Tsz(i) = Tz;

wrwz = Tz/Irw*Dt + wrwz;
wscz = Tz/Isc*Dt + wscz + Tdz/Isc*Dt;
thetaz = thetaz + wscz*Dt;
error_priorz = errorrz;
errorrz = theta_dz - wscz*time - thetaz;
errorTz = errorTz + errorrz;

omega_rwz(i) = wrwz;
omega_scz(i) = wscz;
anglez(i) = thetaz;

```

```

%% time
t(i) = time;
time = time + Dt;

end

%% Plots
figure
subplot(311)
plot(t,anglex*180/pi)
hold on
plot(t,angley*180/pi)
plot(t,anglez*180/pi)
legend('x-axis','y-axis','z-axis')
xlabel('time'); ylabel('\theta (deg)');
subplot(312)
plot(t,omega_rwx/2/pi*60)
hold on
plot(t,omega_rwy/2/pi*60)
plot(t,omega_rwz/2/pi*60)
legend('x-axis','y-axis','z-axis')
xlabel('time'); ylabel('\omega_r_w (rpm)');
subplot(313)
plot(t,omega_scx*180/pi)
hold on
plot(t,omega_scy*180/pi)
plot(t,omega_scz*180/pi)
legend('x-axis','y-axis','z-axis')
xlabel('time'); ylabel('\omega_s_c (deg/s)');

figure
hold on
plot(t,Tsx*10^3)
plot(t,Tsy*10^3)
plot(t,Tsz*10^3)
legend('x-axis','y-axis','z-axis')
title('Control Torque')
xlabel('Time [s]')
ylabel('Torque [mNm]')

```