Design and Analysis of a CubeSat

A Major Qualifying Project

Submitted By:

Robaire Galliath

Oliver Hasson

Andrew Montero

Chistopher Renfro

David Resmini

Submitted To:

Professor Michael A. Demetriou

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.

August 2019 - March 2020

Contents

List of Tables

List of Figures

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ac condimentum eros. Sed quis est eu ante ultrices rhoncus. Praesent a odio eget dolor hendrerit tincidunt. Proin ullamcorper lacus odio, nec dignissim enim hendrerit vitae. Quisque quis risus tellus. Sed sit amet lectus non massa accumsan malesuada. Aenean purus erat, fringilla id ante et, finibus vehicula ex. Sed a ex mi. Duis nec libero ex. Donec aliquam eu tortor quis sodales. Nulla vel urna vitae tellus accumsan bibendum ut quis purus. Sed at ipsum nibh. Fusce erat nulla, aliquet pretium tempor ut, facilisis non sem.

1 Introduction

1.1 Purpose

The goal of this project is to design and conduct analysis of a CubeSat on an extreme low Earth orbit (eLEO) mission. The satellite will be carrying a mass spectrometer to conduct atmospheric observation. Following deployment from the ISS the satellite will enter a 250 - 600 kilometer orbit and maintain this orbit as long as possible. The overall project is composed of three separate MQP teams, each responsible for a different aspect of the satellite design and analysis. This portion of the overall project focuses on the attitude determination and control, orbital determination and control, and command subsystems of the satellite.

1.2 Project Constraints

There are four major constraints on this project: the orbit profile, the primary propulsion system, the scientific payload, and the satellite form factor. As mentioned in the introduction, the satellite must enter and maintain a 250 - 600 kilometer orbit as long as possible. This is to allow the scientific payload, the miniature Ion Neutral Mass Spectrometer (mini-INMS), to conduct atmospheric analysis in low Earth orbit. A Busek electro-spray thruster has already been selected as the primary propulsion system for this mission profile and cannot be changed. It is expected that the satellite adhere to the CubeSat form factor, although the final size of the satellite is flexible.

1.3 CubeSat Background

Cube satellites (CubeSats) are miniature satellites used for space research and technology development. There are a particular class of nano-satellite that was developed by the California Polytechnic State University and Stanford University in 1999 in order to promote the design, manufacture, and testing of satellite technology in low Earth orbit [1]. CubeSats are comprised of multiple 10 cm by 10 cm by 10 cm units, referred to as ’U’s. Layouts can vary greatly, but the most common form factors are 1U and 3U [1]. In recent years, larger CubeSats have been developed to increase available space for mission payloads. Typically, CubeSats are deployed by a launch mechanism attached to the upper stage of a launch vehicle and offer an easy way to deploy CubeSats into Earth orbit.

1.4 Educational and Social Impacts of CubeSats

The continued expansion and development of CubeSat technologies has yielded a variety of positive social, economic, and educational effects. The space industry has been heavily impacted by the surge of CubeSat and related space technology start-ups in the last five years. From 2000 to 2014 space start-ups have received a combined sum of $1.1 billion in venture capital investments **???**. The number has only increased in recent years, with a total investment of $3.9 billion in 2017 alone **???**. These companies provide satellite components, development, or launch and integration services.

The wide availability of CubeSat components and hardware has significantly reduced the cost and complexity of creating a flight capable system. Many universities and several high schools have launched CubeSats thanks to these advantages **???**. Low development and launch cost also provides an opportunity for cost effective flight testing of new and experimental technologies. In a notable example, the *Mars InSight* mission carried two CubeSats as a secondary payload in order to test new miniaturized deep space communication equipment [2].

The expansion and promotion of CubeSats has also garnered interest from students at all levels in the space sector. Space agencies such as NASA and ESA are able to use CubeSats to inspire students to pursue an education and career in STEM fields.

2 Introduction

The primary goal of the MQP-2001 Cube Satellite Project is to create a conceptual design of a 6U CubeSat, carrying a mini Ion Neutral Mass Spectrometer (mINMS) made by NASA Goddard for the purpose of scientific experimentation and data collection in the ionosphere. The 6U CubeSat will be deployed from the ISS via a NanoRacks DoubleDeployer at 400 km with a 51.6 degree inclination. It will then enter a 250-600km orbit after detumbling, maintaining a perigee within extreme Low Earth Orbit (eLEO) for as long as possible with the given propulsion system.

The MQP-2001 CubeSat Project is composed of three MQP sub-teams, each with a different focus. Team Two, led by Professor Demetriou with team members Robaire Galliath, Oliver Hasson, Andrew Montero, Chris Renfro and David Resmini, focuses on CubeSat attitude determination and control, orbital control, and command and data handling subsystems. Team 2 is also taking lead on the design and development of a test bed. Team One is led by Professor's Taillefer and Gatsonis, with team members Tristan Andreani, Edward Beerbower, Roberto Clavijo, Samuel Joy, Benjamin Synder and Jeremiah Valero. These members are responsible for the orbital analysis and environmental effects analysis along with the power, propulsion and telecommunications subsystems. Lastly, team three is led by Professor Karanjgoakar with team members Christian Anderson, Rory Cuerdon, Brian Kelsey and Nicole Petilli, with a focus on the mechanical design and the structural and thermal analysis.

While the overall goal of the project is to collect data from the ionosphere with a 250km perigee, there are several subsystems with their own main objectives that work towards the success of the mission. The objectives of this MQP are to perform mechanical design, structural analysis, and orbital analysis for the 6U CubeSat in compliance with NanoRacks deployer requirements. Additionally, thermal cycling effects will be considered for a given loading scenario on the frame of the 6U.

2.1 Background and Literature Review

CubeSats are small, cost effective satellites that expand commercial access to space. Defined by the standardized scale,'U' (a cube with dimensions of 10cm x 10cm x 10cm, ~1.3kg), CubeSats are typically 1U, 2U, 3U, 6U, or 12U[1,11]. Each Unit is comprised of hardware components specifically selected to complete the satellite's mission. The concept began in 2000 as a plan to provide scientific and military laboratories another tool to grow their operations in space. Many colleges and high schools have programs that allow students to design and build their own CubeSats, illustrating that the concept lends itself to valuable educational experience [1 ,7].

In recent years, the CubeSat has become a unique tool in the scientific community. A cooperative culture has formed around the implementation of CubeSats into everyday space science. NASA's CubeSat Launch Initiative (CSLI) provides opportunities to launch small satellites aboard larger launch vehicles as secondary payloads[8]. The industry consists of companies (Clyde Space, ISIS, etc.) providing interested parties with components necessary to construct the satellite, who then work with integration services (NanoRacks, SpaceFlight Services, etc.) to facilitate the satellite's flight aboard a launch vehicle.

CubeSats are currently in an era of rapid growth in popularity and technological opportunity. It is estimated that the global CubeSat market was valued at $152 million in 2018, and is projected to rise to nearly $375 million. The coming years expect reduced mission costs, increased opportunity in government, military, and commercial applications, and a greater demand for data from earth observation in LEO. One promising project is SpaceX's Starlink, which aims to provide reliable and affordable broadband internet services around the globe. There are currently about 400 Starlink satellites in orbit, with plans to build the constellation to 12,000 by the completion of the project. At the minimum number of Starlinks necessary for operation, the program could bring its services to all U.S territories in time for 2021 hurricane season.

2.2 Social, Economic, and Educational Considerations

The continual expansion and development of CubeSat opportunities has yielded a variety of positive social, economic and educational effects. Since CubeSats were first theorized and developed by Cal Polytech, many universities and even high schools have started similar micro satellite programs as a result of their affordable cost, in addition to a flux of new commercial companies [1]. The CubeSat industry has already had a large impact on the space industry, which has seen a surge of space start-ups in the last five years. From 2000 to 2014, space startups received a sum total of $1.1 Billion in venture capital Investments. That number increased exponentially in the following 3 years, with more than 120 investors contributing $3.9 billion to space start-ups in 2017 ALONE! A majority of these start-ups reflect new technologies and abilities related to CubeSat Components, Development, or Launch and Integration. CubeSats have also created a shift from cost plus to fixed cost payments, which are less risky for the government or investors and maintains performance control of contractors[4, 7].

One of the major benefits of CubeSats is their versatility. CubeSats themselves are a significantly cheaper method to test new technologies so that they can be developed further and flight tested before integration on larger missions. This is because of their low mass and thus low launch cost. Their smaller, modular size also simplifies development and testing, as many subsystem components are available off-the-shelf from different suppliers. Often CubeSat projects can be flight ready within one or two years. Many Universities and Institutions develop CubeSats with a specific payload in mind, including but not limited to remote sensors to communications modules [7, 9].

In recent years, CubeSats have seen expanded use with the ISS as a result of the Japanese Experiment Module Small Satellite Orbital Deployer, and have even flown with missions to the moon and to Mars. The first commercial entity to utilize the ISS as a deployment option was NanoRacks LLC in 2013. With NASA and JAXA, NanoRacks developed and launched a deployer directly tied to the Japanese Experiment Module, allowing CubeSats to be checked by astronauts before deployment. This is a quick and efficient method of launching CubeSats into LEO. Between 2014 and 2017, NanoRacks deployed a total of 176 CubeSats from their deployer on the ISS, and plans to ramp up these numbers in the next few years. The many advantages and developments of CubeSats has attracted more start up companies in recent years. As of 2018, 51% of CubeSats are developed by the private sector, showing CubeSats are no longer just for research by institutes and universities. The larger commercial flux has led to many impressive satellite technologies, including the first commercial optical communication downlink system (Analytical Space, Inc.) and the first CubeSat to employ a new hybrid (dual-purpose) antenna and solar power system [6,11].

Technological developments can be found in a variety of topics, including solar and space physics, Earth Science and applications from space, planetary science, astronomy and astrophysics, and biological and physical sciences in space. From 2000 to 2015, the number of publications on CubeSats has risen to 536 publications total outlining the many advances made [6].

Expansion and promotion of CubeSat use has also helped NASA and the ESA collaborate with interested schools and students to further promote the benefits of the space industry. CubeSats inspire students from many levels of education to pursue STEM fields and take part in an innovative and exciting new part of the space sector. These programs teach students about the many components in a CubeSat as well as the many fields of knowledge needed, organizational skills and communication necessary to efficiently design and build a CubeSat [9].

Unfortunately not all CubeSat effects are positive. More traffic complicates flight paths and adds an element of danger to many missions as the risk of collisions increase. Unlike larger satellites, CubeSats are generally designed without collision avoidance capabilities and without a specified deorbiting plan. Being so small makes them difficult to track, adding to the uncertainty of any area containing CubeSats being safe for other spacecraft. Additionally, having such a large number of satellites also increases the risk of collisions and increased debris resulting from those collisions. The European Space Agency has already experienced a collision due to CubeSat debris. Their Sentinel-1A was struck and had its solar panel destroyed by CubeSat related debris and the debris from that collision put their Sentinel-1B spacecraft at risk. A study using NASA's LEGEND (LEO to-GEO Environment Debris) model, assuming a post mission disposal compliance rate of 90%, shows that continuing the current increase of CubeSats could result in a 75.3% increase of collisions in J1, a 342.2% increase in J2, and an 89.8% increase in J3. This increase of collisions and debris could make operations in LEO difficult and interrupt human spaceflight, or cause damage to the International Space Station. NASA estimates that of all launches into space, 94% are now space debris, where 64% of that are fragments (volume ~100cm2)[5].

There is also a risk of NanoSats being used to gather intelligence from other satellites, disrupting the operations of larger satellites or by spying directly. Their size makes them difficult to detect and prevent these activities. Additionally CubeSats being used for communications could become targets for hackers to cause disruptions. As CubeSats increase in accessibility, more effort and technology will be necessary to keep other satellites and communications secure[5, 10].

Sources: 1: https://www.hindawi.com/journals/ijae/2019/5063145/ 2: https://www.nanosats.eu/ 3: https://sst-soa.arc.nasa.gov/12-passive-deorbit-systems 4: https://www.marketsandmarkets.com/Marke Reports/cubesat-market-58068326.html 5: https://www.space.com/36506-cubesats-space-junk-apocalypse.html 6 : $https : //www.nasa.gov/mission_pages/cubesats/overview$ 7: Commercial Marketplace 8: ESA2 9: ESA1 10: https://www.cbinsights.com/research/industries-disrupted-satellites/ 11: ResearchPaper1

2.3 Previous CubeSat MQPs

Yuh this part sucks

2.4 Payload: mINMS

The payload of this mission is the mini Ion Neutral Mass Spectrometer or mINMS for short. It was developed at NASA's Goddard Space Flight Center, with two apertures for detecting ions of densities between $10^3 - 10^8/cm^3$ and Neutrals of densities between $10^4 - 10^9/cm^3$, with very low energies between 0.1eV and 20eV. These apertures must be oriented in the RAM facing direction (the direction of movement), and are capable of making high resolution, in-situ measurements of [H], [He], [O], [N2], [O2] & [H+], [He+], [O+], [N2+], and [O2+]. The instrument occupies nearly 1.5U of volume and has a mass of 560 grams[12]. A picture of the mINMS is shown below (human hand for size comparison):

IMAGE ONE YUH

Since its creation, the mINMS has been used on a few missions, with more planned in the next few years. This includes the ExoCube 3U CubeSat launched in January 2015 on a rideshare mission and the Dellingr 6U launched in August 2017 from the ISS. The petitSat planned to be launched in August 2021 also from the ISS. The ExoCube was designed by Cal Polytech in collaboration with NASA Goddard Institute with the mINMS as its primary payload. It was a 3U CubeSat with mass of 4kg deployed in Low Earth Oribt (LEO). The mission had issues with transmitting power, however it was able to validate that the mINMS was operating. ExoCube, shown below, was in operation for 7 months[13].

IMAGE 2 YUH

Following the ExoCube mission was the Dellingr 6U mission, launched from the ISS NanoRacks Deployer. Despite many issues that arose, the team successfully achieved a resilient mission and provided many lessons and areas of growth for CubeSat missions, eventually inspiring the petitSAT, GTOSat and BurstCube missions. Dellinger was also able to provide clear detection of ionized hydrogen (H+), helium (He+) and oxygen (O+) in the atmosphere in May of 2018, proving Ion detection capabilities. As of October 2018, the team turned on the neutral mode, which is still the primary focus. These measurements are necessary for studies of the dynamic ionosphere-thermosphere-mesosphere system, or simply put to define the steady state background atmospheric conditions[14].

IMAGE 3 YUH

The final mission with the mINMS payload, the Plasma Enhancements in The Ionosphere-Thermosphere Satellite, or petitSat, is planned to be launched in 2021 with a very similar design to that of Dellingr. It is the first to utilize a Dellingr-X frame, designed based on lessons learned from the Dellingr mission, which is more reliable, cheaper, and protects electronics. Additionally, deployable solar arrays and a more advanced star tracker were included to avoid previous issues. The goal of the petitSat mission, run by NASA scientists at the Goddard Space Flight Center, is to determine how perturbations in the density of plasma within the ionosphere, also called "blobs," distort the transmission of radio raves. These blobs commonly interfere with GPS and radar signals from Earth (get reflected back), and it is theorized that fast-traveling waves coming from the thermosphere may have an effect, as they lead to a phenomenon called Medium Scale Travelling Ionospheric Disturbances. Scientists are trying to determine the relationship between these two phenomena (see how closely related), therefore the INMS will continue to observe density changes in response to daily and seasonal cycles, while a second instrument measures distribution, motion and velocity of ions[15].

Sources: 12: Mass Spectrometers for CubeSats (provided by Gatsonis as well) 13: Exocube 14: Dellingr and Dellinger 2: Specifics 15: Intro to PetitSat

2.5 Project Goals

The cubesat analysis MQP team, comprised of three separate groups, was tasked with creating a 6U cubesat to gather atmospheric information in an extreme low Earth orbit (eLEO). Not many cube satellite missions orbit this low to the earth, so being able to take atmospheric readings from eLEO utilizing the NASA Goddard mINMS payload will provide researchers with valuable information from a scientifically rich area not frequently explored.

The principal goal of the subsystem design project team, Team 1 (GT), is to maximize the lifespan of the cubesat in LEO. By modifying orbital parameters and maximizing the efficiency of burns, the mission and lifespan of the satellite will be optimized, providing an increased duration of data collection, adding to the value of the mission.

The primary goal of the CubeSat analysis MAD team, Team 2, is to first successfully detumble the 6U CubeSat once it is seperated from its launch vehicle. After detumbling, Team 2 then handles the attitude determination and orbital control of the 6U. Team 2 is also in charge of handling the command and data handling that deals with storing and downloading payload data, along with creating a functioning testbed.

The first goal of the NAD project team, Team 3, is to perform mechanical design of the

6U eLEO cubesat, in accordance with components chosen by the previous SEG Teams, to meet NanoRacks deployer design requirements. The second goal of team 3 is to conduct thermal and structural analysis on the 6U eLEO cubesat to meet respective NanoRacks requirements and ensure all chosen components are suitable for the eLEO mission.

2.6 Project Requirements and Constraints

2.6.1 Deployer and Launch Options

With the exponential growth of CubeSat missions, one can find multiple deployer options for CubeSats based on size and target orbit. The first deployer ever designed was the P-POD(Poly-Picosatellite Orbital Deployer) by Stanford and CalPoly Tech SLO in the early 2000's. The P-POD is a standard deployment system developed around the 10 cm x 10 cm x 10 cm initial CubeSat design, with a length of 34 cm to hold up to three 1U CubeSats. The design minimizes interaction with primary payload by enclosing cubesats in a dormant state, and uses a spring and pusher plate to guide CubeSats along interior rails and out the deployer door, as seen below. P-POD's have a good flight heritage and have been used extensively for over a decade. An updated design can hold a 6U CubeSat in a 2x3U orientation. P-POD's allow up to 1 kg per Unit, and can eject multiple Cubesats at once per deployer[16].

Illustration of P-POD Double Deployer (2x3U). Credit: CalPoly

ENTER IMAGE 1 YUHH

Since the invention of the P-POD, multiple similar designs have been used to decrease weight or increase the size or mass of payload. This includes Tyvak Launch Systems deployer, which promotes custom manufactured deployment mechanisms with flight tested COTS 3U, 6U, and 12U deployment mechanisms used on rideshare missions, as well as the ISIPOD or ISIS Payload Orbital Dispenser, later upgraded and renamed as Quadpacks. The Quadpacks deployer, developed by Planetary Sciences Corp, provides a variety of opportunities for various CubeSat sizes, ranging from 1U to 16U. The mechanism is very closely related to that used in a P-POD, however the main designs include a 12U dispenser broken into 4 sections of 3U areas (4x3U), or a 16U (4x6U). It offers a very flexible configuration or deployment sequence, with the ability to release many satellites at a time, for example four 3U's, a combination of 3U's, 2U's and 1U's, or 12 to 16 1U's in total. Each section of the QuadPack can its door independently of the other 3 doors.

ENTER IMAGE 2 YUH Quadpacks 12U dispenser for dnepr launch june 2014; Credit: ISIS [1]

Quad Packs are also extensively used on SHERPA Kick Stage Vehicles (also known as the "space tug") to move payloads into desired orbits. SHERPA's can be carried on any EEVL (Evolved Expendable Launch Vehicle) including the Atlas V, Delta IV, and Falcon 9 (all certified), and can hold up to 1500 kg of payload, with 40+ QuadPacks[17].

The final and most recommended deployment mechanism is the NRCSD, or the NanoRacks CubeSat Deployer, located on the Japanses Experiment Module on the ISS. When CubeSat deployment operations begin, the NRCSDs are unpacked, mounted on the JAXA MPEP (Multi-Purpose Experiment Platform) and placed on the JEM airlock slide table for transfer outside the ISS. A crew member operates the JRMS (JEM-Remote Manipulating System) – to grapple and position for deployment. The CubeSats/nanosatellites are deployed when the JAXA ground controllers command a specific NRCSD. The NRCSD Configuration can be seen below:

ENTER IMAGE 3 YUHUHHHER NRCSD Configuration; Credit: NanoRacks

ENTER IMAGE 4 YER IFRIT FRUIT

Since its first use in July 2014, 200+ payloads have been sent to the ISS and deployed from the NRCSD. The NRCSD is a self-contained system that is still electrically isolated from the ISS to protect the crew. Onboard the ISS, NanoRacks Platforms are installed in EXPRESS Rack inserts to supply power and USB data transfer capability for NanoRacks Modules, allowing CubeSats to conduct experiments on the ISS and be checked by astronauts. The NRCSD is able to launch CubeSats with a maximum length of 50 cm[18].

As noted above, a 6U CubeSat is a major limiting factor, such that only a few of deployers are designed for such a mission, including the NanoRacks and Quadpacks deployers. Based on these two options, there are certain launch opportunities able to carry a 6U cubesat.

To be deployed by NanoRacks, CubeSats must first reach the ISS by a Cygnus or Dragon Spacecraft. This of course limits CubeSats to a 51.64 degrees inclination and requires additional compliance with ISS safety regulations. Additionally, using any of the EELV's, a SHERPA Spacecraft carrying our 6U CubeSat in a 16U Quadpack could deploy the satellite at a variety of orbits. The major disadvantage is waiting for the next SHERPA launch opportunity however.

Lastly, in certain cases, Antares and Falcon 9 rockets (not going to the ISS) will have additional space and can carry pico or micro satellites. This is known as a rideshare opportunity. Antares launches are currently limited to P-POD configurations, while the SpaceX rideshare website provides little information on the deployment options or capabilities, which is why the teams recommends to launch with a Falcon 9 carrying CubeSats in a Dragon Module to the ISS, and deploy from a NanoRacks deployer on the Japanese Experiment Module (JEM). There are many more opportunities as Dragon Spacecraft frequently visit the ISS[19].

16:https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/cubesat-concept 17: https://ieeexplore.ieee.org/document/7118918 18: https://ieeexplore.ieee.org/document/71189
19:https://www.spacex.com/smallsat

2.6.2 Launch Vehicle

The SEG team decided that the best deployer option would the NanoRacks deployer, requiring a Falcon 9 to carry the cubesat to the ISS in a Dragon Module. This was the best option as NanoRacks is one of the most flight tested deployers compatible with a 6U CubeSat, has a strong mission success flight record, and allows multiple launch options for our CubeSat to ride share to the ISS[20].

The NanoRacks requirements are broken up into various sections, some requiring further collaboration with the other two SEG teams. The topics are listed and further explained below: - 10 Rail or 10 Tab Requirements -The first ten requirements all relate to the dimensions, placement, and material properties of the rails or tabs along which the CubeSat will be deployed from the NRCSD. It is the full SEG's choice to decide between the rail or tab configuration. Both have similar requirements that are the responsibility of the design team, who create a final CAD model of the whole structure including the rails or tabs. - 2 Major Design Requirements (Mass and COM) -Two major design requirements that are affected by all teams decisions are the total mass and center of geometry distance from center of mass along each axis. It is the design teams responsibility to ensure these requirements are met by adding components carefully, considering specifications provided by the other teams. -12 Deployment Requirements (Switches, locations, deployables) -The deployment requirements denote the locations and directions for a variety of possible deployment switches, in accordance with the minimum of 3 deployment switches corresponding to independent electrical inhibits on the main power system. The 6U system must also consider deployment velocity and tip off rate. -3 RBF/ABF and Electrical Switch Requirements -Remove Before Flight and Apply Before Flight features are necessary design considerations when utilizing NanoRacks Deployer to ensure the safety of the ISS and its habitants. The design team in collaboration with the command team must ensure that these features are included and that an access panel is placed on the +Y face for physical accessibility. -10 Structural and Environment Requirements (7 Considered) -The structural analysis team must ensure the CubeSat is capable of withstanding the random vibration environment during launch though a vibration test report, as well as integrated loads of 1200N across all load points in the Z-direction and depressurization/ vacuum conditions. This will require a Safety Data Template. -Should the CubeSat contain any detachable parts, additional coordination with NanoRacks is required. -14 Electronics Requirements (battery, capacitors and wiring) -An electric schematic and battery test report ensures that the battery and its charging methods are safe and that all wiring and circuitry is protected. This will require coordination between the design and power team. -3 Material Based Requirements (outgassing, hazardous materials) -In addition to a materials list for the rails, the design team must provide a bill of materials for the entire CubeSat to ensure all materials are resistant to stress corrosion, comply with NASA guidelines from hazardous materials as well as outgassing regulations. Total Mass Loss (TML) must be less than 1%, and Collected Volatile Condensable Material (CVCM) less than 0.1%. -3 Orbital Requirements (debris, re-entry) -Lastly, the thermal and orbital analysis teams are responsible for creating an Orbital Debris Assessment Requirements (ODAR) report should the CubeSat exceed 5 kg, or if it is determined that the cubesat will survive re-entry.

As can be seen above, meeting all chosen NanoRacks requirements will require the coordination of members from each team, in addition to continual updates to our important documents ensuring all notable components are safe and the best pick for this mission.

20: Nano-Racks Double Deployer

2.6.3 Power Subsystem Requirements

The selected solar panels are the Photon 3U body mounted panels, manufactured by Clyde Space. These panels utilize Spectrolab XTJ Prime solar cells with a BOL efficiency of 30.7% and EOL efficiency of 26.7%. The Photon 3U panels have a standard operating temperature range of -40C to 80C, though they are advertised to have available testing for different ranges. Our 6U CubeSat will require 4 of these body mounted panels, one on each long side.

The chosen battery is the Optimus-40, manufactured by Clyde Space. The Optimus-40 has dimensions of 95.89 mm by 90.17 mm by 27.35 mm and a full discharge voltage of 6.2 V. The battery's operating temperature is -10 C to 50 C. Our 6U CubeSat will require a single battery placed within the CubeSat.

The selected EPS is the Starbuck Nano-Plus, manufactured by Clyde Space. The EPS has dimensions of 95.89 mm by 90.17 mm by 20.82 mm and has PDMs with 10 latching current limiters. The Starbuck Nano-Plus has an operating temperature is -40 C to 85 C. The EPS can be placed anywhere inside the CubeSat.

2.6.4 ADC Subsystem Requirements

The selected magnetorquer is the NCTR-M002, manufactured by New Space. The NCTR-M002 requires less than 200 mW of power, each, to operate and its dimensions are 70 mm by a diameter of 10 mm. The NCTR-M002 can operate within a range of -20 C to 60 C. Our 6U CubeSat will require three of the NCTR-M002 magnetorquers, one for each axis. There is no specific location the magnetorquers must be located in.

The selected reaction wheel is the RWP050, manufactured by Blue Canyon Technologies. The RWP050 requires less than 1 Watt of power, each, to operate and its dimensions are 58 mm by 58 mm by 25mm. The RWP050's operating temperature was not specified within Blue Canyon's provided spec sheets. Like the magnetorquers, our 6U CubeSat will require three reaction wheels, one for each axis, and can be placed wherever they fit within the CubeSat.

The selected GPS is the NGPS-01-422, manufactured by New Space. The NGPS-01-422 can operate within a range or -10 ° C to 50 ° C while consuming less than 1.0 W of power. The GPS can be placed anywhere on the CubeSat and has the dimensions of 155 mm by 76 mm by 34 mm with an antenna that has the dimensions of 54 mm x 54 mm x 14.1 mm.

The ADC decided on a component that combines the accelerometer and magnetometer into one piece of hardware. The combined accelerometer and magnetometer is the LSM303AGR that has the dimensions of 2 mm by 2 mm by 1 mm and consumes 1130 μA while operating and 2 μA while idle. The LSM303AGR can operate within a temperature range of -40 ° C to 85 ° C. This component must be located on the center of mass of the 6U CubeSat to function effectively.

The selected gyroscope is the EVAL-ADXRS453, manufactured by Analog Devices. The single gyroscope will be aligned in the center of all three primary axes. The EVAL-ADXRS453 consumes 0.0189 W, can operate at a temperature between -40 ° C and 105 ° C and its dimensions are 33 mm by 33mm by 3 mm.

The final component of the ADC subsystem are the fine sun sensors. The sun sensor chosen is the Nano-SSOC-A60, manufactured by New Space. Our 6U CubeSat will incorporate five fine sun sensors, located in all four corners with one attached to the front of the payload. To work effectively, the fine sun sensors should be located 90 degrees apart for best coverage. The sun sensors require < 2 mA of power to operate and its dimensions are 27.4 mm by 14 mm by 5.9 mm. The Nano-SSOC-A60 can operate within a temperature range of -30 ° C to 85 ° C.

2.6.5 Structural Design Requirements

The following section outlines the various requirements and constraints relevant to each subsystem. Often these parameters require the teams to coordinate with one another to ensure all components and aspects of the mission will operate as planned. The requirements and constraints of the design team's subsystems will be discussed first.

The design of the 6U itself has specific Center of Mass and rail requirements according to NanoRacks Double Deployer specifications, which can be found in Appendix. All components provided by the other teams (excluding solar panels) must fit within the walls of the 6U sized structure, which is in a 30 cm x 20 cm x 10 cm configuration (height, length, width). On top of that, the components must be situated in such a way that the center of mass is as close as possible to the center of geometry to minimize the necessary attitude control computation time/ complexity. Additional Center of Mass requirements are provided by NanoRacks, as seen in the table below:

Table 1: Center of Gravity Requirements

Axis	Distance to Geometric Center
X-Axis	5 cm
Y-Axis	3 cm
Z-Axis	8 cm

Location requirements are provided by the other teams; examples include the accelerometer, which must be at the Center of Mass, the reaction wheels along each axis intersecting with the COM, and the payload which has its aperture in the RAM facing direction (opposite side of the engines). The design team must also create rails or tabs for proper ejection from the deployer. A visual of the rail design and load regions can be found below:

IMAGE OF RAILS BUT IT ISN'T INCLUDED BECAUSE ITS A SCREENSHOT OF GOOGLE DRIVE INSTEAD

Design of the 6U CubeSat must additionally comply with the following constraints. The most important constraint provided is limiting the mass of the total 6U cubesat to a maximum of 12 kg. The design team must ensure that no hazardous materials are used according to section BLANK of the Deployer requirements, and must ensure Total Mass Loss (TML) is less than 1%, and Collected Volatile Condensable Material (CVCM) less than 0.1% due to outgassing.

The structural analysis team must ensure that the final design of the CubeSat is strong enough to withstand all conditions of the mission. NASA requires that all vehicles entering space pass the ground structural tests outlined in the general environmental verification standard (GEVS). For our project we followed the requirements for structural tests from nanoracks (sec. 1.3.2) because they are more relevant to CubeSats specifically and still comply with the GEVS requirements. Nanoracks requires the vehicle to survive a random vibration test, a structural load test for 1200N in the z-axis direction, an airlock depressurization test, and must ensure no detachable parts come loose.

The thermal analysis team is responsible for ensuring that all selected components discussed above will operate within their allowable temperature ranges throughout the duration of the mission. It must also be ensured that any heat that these components produce within the internal CubeSat structure is negligible in order to maintain allowable internal temperature profiles.

2.7 Overall Project Management

As previously mentioned, the Systems Engineering Team (SEG) is divided up into three teams by closely related subsystems and the relevant analysis required for each. Students and Professors are split among these teams based on the amount of responsibility given to each team. The figure below simplifies how the teams are divided and the major roles of each team:

ENTER IMAGE OF CRAPPY SEG GROUP BREAKDOWN YUR BUT YUH FIGURE N: SEG GROUP STRUCTURE

Sub-Groups communicate via Slack and have a shared google drive for component lists and other necessary information that needs to be shared with the whole team. Proprietary documents, and CAD drawings were not share on the google drive. Sensitive documents deemed able to share with the whole group were stored on a secure OneDrive folder, while those deemed only sharable with a specific sub-group, were stored on one individual flash drive, to ensure that no confidential knowledge got leaked.

In addition to the above sub-team designated responsibilities, team members presented twice a week, first all together to their advisors on technical advances and weekly progress, then on a rotating schedule for the full SEG meetings. As team lead, Brian focused on managing team tasks and coordinating with the other teams more often then designing CAD models and assemblies. This included taking notes during meetings, tracked tasks based on a year long timeline for the three subsections, and coordinated with the other SEG teams to complete the Component List. During this coordination he contributed to research and writing on launch options, deployment options and requirements, and payload specifications. As the Design lead, Nicole was responsible for the flash drive with all sensitive documents, and ensuring these documents were safely shared when needed by other members for simulations or the final design assembly.

2.8 MQP Objectives, Methos, and Standards

2.9 Team 1

Power Objectives: The power subsystem for this project is responsible for supplying the power that is generated, stored, and distributed throughout the CubeSat. Many of the CubeSat's components will require continuous power draw in order for the CubeSat to remain functional during its lifespan. To account for this, a power budget was created considering all of the power consuming hardware that will be implemented into the design. To ensure proper power delivery, hardware power requirements and their operational priority and duration were taken into consideration. A power budget timeline of hardware was created to help illustrate and analyze the overall power consumption of the CubeSat. The timeline showed what hardware should be turned on and off throughout the mission for each orbit.

Propulsion Objectives: The propulsion subsystem has two objectives in order to complete the goal of our CubeSat. The first objective is to determine the number of thrusters required to keep the CubeSat in orbit. It has been decided that the Busek electrospray thruster BET-300P will be the thruster used in the CubeSat. The second objective of the propulsion subsystem is to determine the burn time to optimize the lifespan of the CubeSat in orbit.

TeleCommunications Objectives: The Telecommunications sub-system has three primary objectives. The first is to select hardware that meets requirements posed by other subsystems, such as power usage and structural placement. The second is to identify a viable Ground Station Network (GSN) that will allow the CubeSat to transmit data at an acceptable daily rate. The ground station sites should be inside the satellite's coverage, given the orbital inclination, and also able to transmit and receive in X-band frequencies. The third is to establish the uplink and downlink budgets, for use by the payload and data handling instruments.

Environmental Effects Objectives Understanding how the CubeSat will behave in the space environment is key to the success of the mission. In our team's desired orbit there is atmospheric drag, solar radiation pressure and free electrons. This makes up the thermosphere and ionosphere. The environment will affect the structure of the CubeSat, along with the telecommunications, propulsion, and detumbling and control systems. In order to accurately predict the functional lifetime and success of the CubeSat, it must be designed with consideration to every environmental hazard. In addition, the temperature fluctuations must be taken into consideration as the thermosphere can range from 500 °C to 2000 °C. This is due to the ionosphere harboring extremely charged electrons to make a plasma environment. However, the atmospheric density is quite low, thus the ambient temperature would feel cold to the human skin.

Payload Objectives The payload subsystem is the driving parameter behind the entire mission. At this current time, the team does not have the exact payload specifications beyond knowing it is approximately 1U, does not require specific pointing parameters, and collects data within the ionosphere. Once the technical parameters such as power draw, mass, and data transmission speed are determined, the team will properly interface the payload with the CubeSat and determine the majority of the design parameters and constraints for the mission.

Orbital Objectives This MQP team is responsible for determining the orbital parameters for the mission. Utilizing Systems Toolkit (STK), the team will determine the propellant needed to transfer from initial orbit to the desired elliptical orbit, then maintain that orbit for as long as possible with the provided propulsion system. Other orbital parameters to analyze include the solar flux upon the spacecraft to determine power draw, the drag profile on the spacecraft, and environmental effects of eLEO such as electron flux.

For the ADC portion of the project, the first objective is to define the control modes, such as detumbling, based on system-level requirements. The next objective is to quantify the disturbance torques based on the mission profile. Once that is completed, the objective is to select spacecraft control methods (e.g. b-dot) for each control mode (e.g. detumble) based on mission requirements and constraints. Then, the objective is to select and size sensors and actuators in order to determine attitude and control spacecraft for each control method. The next objective is to define attitude determination algorithms and control algorithms based on capabilities, requirements, and constraints. For the ODC portion of the project, the first objective is to define control modes based on system-level and mission requirements. The next objective is to quantify drag based on the mission profile. Once that is completed, the objective is to select orbit control methods (e.g. low impulse maneuver) for each control mode (e.g. apogee raising) based on mission requirements and constraints. Then, the objective is to select and size sensors (GPS) and actuators (thrusters) in order to determine orbital parameters and to control the spacecraft for each control method. The next objective is to define orbital determination algorithms and control algorithms based on capabilities, requirements, and constraints. For the CDH portion of the project, the first objective is to define mission phases(e.g. pre-deployment, deployment, data acquisition) based on system-level requirements and mission profile. The next objective is to quantify data requirements. Once that is completed, the objective is to select data handling and spacecraft command methods for each mission phase (e.g. apogee raising) based on mission requirements and constraints. Then, the objective is to select computer components in order to handle data and command spacecraft. The next objective is to define data handling and transmission algorithms based on capabilities, requirements, and constraints. During the entire project and for all portions, iterating steps as necessary to achieve goals and document steps are objectives as well. For the test bed portion of the project, the objective is to construct a 3 DOF lab testbed for the test and validation of ADCS control systems. Once the physical testbed is completed the next objective is to create an electronic control system for the remote control and analysis of the system.

2.10 Tasks and Timetable

3 Background

3.1 Attitude Determination and Control

The purpose of the attitude determination and control subsystem (ADCS) is to properly position and orient the spacecraft to meet the needs of the mission. The ADCS is responsible for three distinct operations throughout the mission: detumbling, initial attitude determination, and attitude maintenance. Successful operation in all three phases is vital to the overall success of the mission. This is accomplished by combining a variety of sensors and actuators in a closed-loop control system.

3.1.1 Requirements

Each phase of the mission has different requirements. In order to successfully detumble the satellite must correct for the angular spin imparted during deployment. It is expected that such an angular velocity would not have a magnitude greater than ten degrees per second about any axis. Once the satellite has reduced its angular velocity to less than ??? degrees per second, it must determine its orientation with respect to the Earth inertial frame. From this point onwards the satellite must maintain its attitude within plus or minus five degrees. As the orbit dips lower into the atmosphere the effects of drag become significant. Should the angular orientation deviate further than this limit the torques induced by atmospheric drag risk overcoming the strength of the on board actuators, causing the spacecraft to enter an uncontrollable spin. The torque exerted on the spacecraft can be described as a function of atmospheric density ρ, cross sectional area A, velocity V_{rel}, drag coefficient C_D, center of pressure c_p, and center of gravity c_g, as shown in equation 1.

$$\tau_d = \frac{1}{2}\rho A C_D V_{rel}^2 (c_p - c_g) \tag{1}$$

Velocity is a function of the orbital velocity and the pointing angle. As the velocity increases, it is evident that the torque experienced by the spacecraft increases as well. This relationship is shown in figure 1.

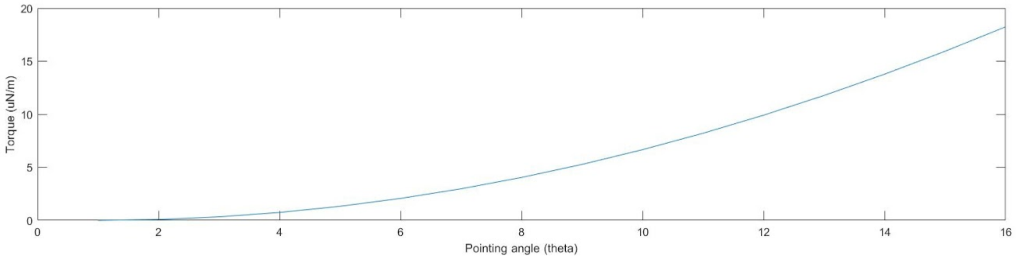


Figure 1: Pointing Angle vs. Torque

3.1.2 Sensor & Actuator Selection

A variety of sensors are included on board the satellite to allow it to determine its angular position and rotation rates. This includes a gyroscope, magnetometer, and sun sensor.

Magnetorquers create a torque using an external magnetic field and a magnetic dipole moment. A unique property of the magnetorquer is that is has no moving parts, unlike a reaction wheel that relies on moving parts. Due to this unique characteristic, magnetorquers are less prone to malfunction, and along with being cheap, lightweight and simple, are great for CubeSats. Magnetorquers are commonly made from a metal rod wrapped in a copper wire and connected to a power source, usually a 5V battery. Magnetorquers induce a magnetic moment by having a current run through them. Through the magnetic moment, ν, multiplied by the external magnetic field felt by the magnetorquer, B, the magnetic torque, τ, can be found.

$$\tau = \nu \times B \tag{2}$$

As mentioned earlier, magnetorquers are a method of controlling the attitude of a spacecraft, in this case a 4U CubeSat. This will be achieved by having magnetorquers, in combination with reaction wheels, interact with Earth's magnetic field which allows for a method of dumping excess momentum into the wheels.

The magnetorquer chosen for this task is the NCTR-M002 Magnetorquer Rod, which is manufactured by New Space. The NCTR-M002 uses a magnetic alloy rod that produces an amplification effect over an air cored magnetorquer, which in turn produces less power. The NCTR-M002 consumes around 200mW of power from a 5V power supply while producing a magnetic moment greater than 0.2 Am^2. The residual moment left over from the magnetorquer rod is basically negligible at less than 0.001 Am^2. A picture of the NCTR-M002 Magnetorquer Rod can be seen below.

The NCTR-M002 Magnetorquer Rod's performance characteristics are listed below:

Figure 2: NCTR-M002 Magnetorquer Rod

Table 2: Magnetorquer Parameters

Parameter	Value
Magnetic Moment	0.2 Am^2
Linearity	$< \pm 5\%$
Residual Moment	< 0.005 Am^2
Dimensions	$70mm \times \varnothing 10mm$
Mass	< 30 g
Power	200 mW
Operating Temperature	$-20^{\circ}C$ to $60^{\circ}C$
Vibration	14 g(rms)

Reaction wheels, also referred to as momentum wheels, are internal components that store rotational energy, proving satellites with three-axis control without the need for external sources or torque, like a propulsion system, to reorient the spacecraft. Reaction wheels are used in CubeSats because of their ability to control a satellites attitude with very high precision, while also being lightweight, compact and cheap. Reaction wheels adjust the attitude of a spacecraft by using conservation of momentum. By adjusting the momentum of a weighted wheel in the body of a spacecraft, reaction wheels cause the spacecraft body to spin in the opposite direction.

The reaction wheel chosen for this particular mission was the RWP050, manufactured by Blue Canyon Technologies. The RWP050 Reaction Wheel can create a maximum torque of 0.007 Nm and a momentum of 0.050 Nms while operating at less than 1 W at full momentum.

Parameter	Value
Mass	0.24 kg
Dimensions	58 × 58 × 25 mm
Voltage	10 - 14 V
Power	<1 W
Operating Temperature	$-20°C$ to $60°C$

The attitude and determination subsystem will also included a Global Positioning System (GPS) that will receive information from the on-board GPS receiver that will pull data from the magnetic field and the sun reference models.

The NGPS-01-422 is a great choice for the on-board GPS for our 4U CubeSat. The NGPS-01-422, manufactured by New Space, will be able to pull information from sensors as well as provide information on the CubeSat's position and velocity at any point along the CubeSat's orbit.

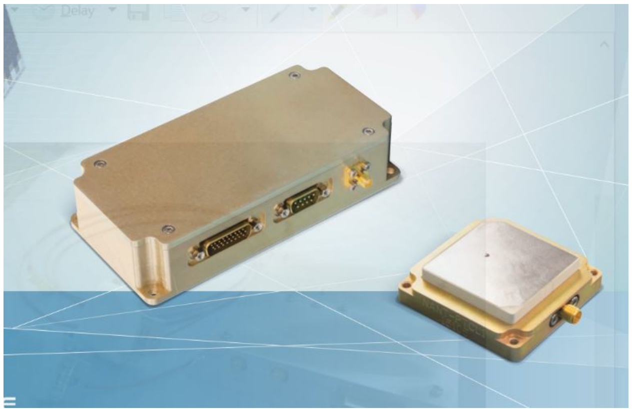


Figure 4: NGPS-01-422

Table 4: GPS Parameters

Parameter	Value
Mass	<500 g
Power Consumption	1.5 W
Position	< 10 m
Velocity	< 25 cm/s
Operating Temperature	$-10°C$ to $50°C$
Dimensions	155 × 76 × 34 mm

The NGPS-01-422 New Space GPS Receiver includes an antenna, NANT- PTCL1,which is also included in the photo above, with the specifications below:

Table 5: GPS Receiver Parameters

Parameter	Value
Mass	<80 g
Power Consumption	80 mW
Frequency	1575.42 MHz
Bandwidth	20 MHz
Operating Temperature	$-25°C$ to $55°C$
Dimensions	$54 \times 54 \times 14.1$ mm
Active Gain (RHC)	$>$ 16dBi
Noise Figure	$<$ 2 dB

The attitude determination and control subsystem will also include a gyroscope that will gather readings on the angular velocity and angular acceleration of the 4U CubeSat. Using these readings, the gyroscope will communicate with other sensors to adjust the orientation of the CubeSat as well as maintain the 5 degree pointing requirement.

The gyroscope our team has decided to move forward with is the gyroscope chosen by previous year's MQPs, the EVAL-ADXRS453, manufactured by Analog Devices. The EVAL-ADXRS453 consumes very little power, 0.0189W and is very lightweight, only weight 56.7 g. A picture of the EVAL-ADXRS453 along with its performance characteristics can be seen below.

Figure 5: EVAL-ADXRS453

Table 6: Gyroscope Parameters

Parameter	Value
Mass	<56.7 g
Power Consumption	18.9 mW
Operating Temperature	$-40°C$ to $105°C$
Dimensions	$33 \times 33 \times 3$ mm

Our group decided on a piece of hardware that would combine both the accelerometer and the magnetometer. The LSM303AGR is a triple-axis accelerometer and magnetometer that is incredibly small, lightweight, cheap and power efficient when being compared to other accelerometer and magnetometer components. The LSM303AGR will be able to take readings of Earth's magnetic field relative to the CubeSats body-fixed axes and use them to communicate with other sensors that will reorient the spacecraft both in the detumbling phase and post-detumbling.

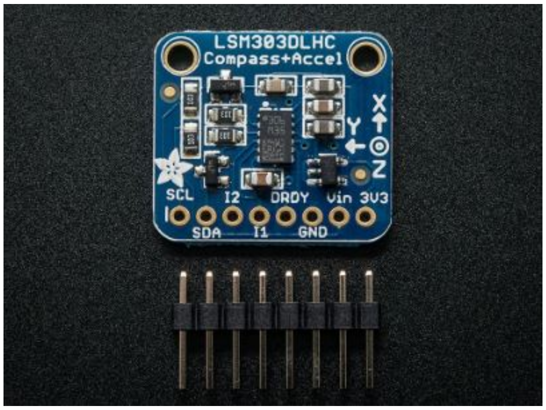


Figure 6: LSM303AGR Triple Axis Accelerometer and Magnetometer

Table 7: Accelerometer and Magnetometer Parameters

Parameter	Value
Mass	10 mg
Power Consumption	5 mW
Operating Temperature	$-40°C$ to $85°C$
Dimensions	$2 \times 2 \times 1$ mm
Linear Acceleration	±0.01%
Magnetic Sensitivity	±1%

Sun sensors, in general, are used for accurate sun-tracking, pointing and attitude determination. Two axis fine sun sensors, when compared to coarse sun sensors, offer a higher accuracy when measuring the incident angle of a sun ray in two orthogonal axes. Fine sun sensors also require a small amount of power in order to operate, unlike coarse sun sensors, but offer more information.

Our team chose to use five Nano-SSOC-A60 analog sun sensors, manufactured by NewSpace. The Nano-SSOC-A60 is a two-axis, low cost sun sensor that will be used for the sun-tracking and attitude determination of our CubeSat. Using the Nano-SSOC-A60, we will be able to determine the location of the sun with respect to our spacecraft, which will then be used in the QUEST quaternion and attitude of our CubeSat. The specifications of the Nano-SSOC-A60 are listed below.

Figure 7: Nano-SSOC-A60 Sun Sensor

Table 8: Sun Sensor Parameters

Parameter	Value
Mass	4 g
Power Consumption	10 mW
Operating Temperature	$-30°C$ to $85°C$
Dimensions	$27.4 \times 14 \times 5.9$ mm
Field of View	$\pm 60°$
Accuracy	$< 0.5°$
Precision	$< 0.1°$

3.1.3 Control Logic

3.1.3.1 Sun Pointing

Sun Pointing Vector in the Body Fixed Frame

$$\begin{bmatrix} V_1 \\ \vdots \\ V_N \end{bmatrix} = C \left(\begin{bmatrix} C_{K_1 \hat{n}_1} \\ \vdots \\ C_{K_N \hat{n}_N} \end{bmatrix} s + \begin{bmatrix} \hat{C}_{K_1 \nu_{V1}} \\ \vdots \\ \hat{C}_{K_N \nu_{VN}} \end{bmatrix} \right) \tag{3}$$

$$V = C \frac{n^T s}{||v||||s||} \tag{4}$$

$$C = \frac{V_{max} F_o}{F_{cal}} \tag{5}$$

F_o is equal to the Solar Flux of the Sun which is $3.9x10^{26}$ W

F_{cal} is the Calibration Flux Constant. F_{cal} is equal to the flux due to direct sunlight F_o in an ideal situation which is $3.9x10^26$ W

V_{max} is the maximum output voltage of the CSS which is directly proportional to F_d

$$F_d = F_o(\frac{n^T s}{||n||||s||}) \tag{6}$$

n is equal to the unit normal vector of the CSS.

s is equal to the direction vector from the spacecraft to the sun.

C is the calibration constant.

C_k is a bias parameter which, in most cases, is considered equal to 1.

$$\widetilde{y} = Hx + \nu \tag{8}$$

As mentioned before, if there are at least three measurements the least squares method is used. To solve for the sun vector, the following estimate is used:

$$\hat{x} = (H^T H)^{-1} H^T \widetilde{y} \tag{9}$$

However, if there are less than three observations, the minimum norm method is used. Therefore, the following estimate is used:

$$\hat{x} = H^T (H H^T)^{-1} \widetilde{y} \tag{10}$$

It is obvious that this method has a lot of assumptions including negligible calibration errors and biases, but, although they are not true in flight, numerical simulation results demonstrate that this method is capable of achieving CSS pointing despite these biases.

3.1.3.2 Sun Pointing in the Inertial Frame

Precision up to 0.01° (36”) for years between 1950 to 2050.

$$n = JD - 2451545.0$$

Where n is the number of days since Greenwich Noon, Terrestrial Time, on the 1st of January, 2000.

JD is the Julian date for a desired time.

You then find L, the mean longitude of the Sun:

$$L = 280.46^\circ + 0.9856474^\circ n$$

Next, find the mean anomaly of the sun, g:

$$g = 357.528^\circ + 0.9856003^\circ n$$

Then, find λ which is the ecliptic longitude of the Sun.

$$\lambda = L + 1.915^\circ sin(g) + 0.020^\circ sin(2g)$$

Note: L and g both need to be between 0° and 360°. In order to do this, simply subtract or add 360° until this happens.

Next, you find the distance from the Sun to the Earth in AU:

$$R = 1.00014 - 0.01671 cos(g) - 0.00014 cos(2g)$$

Finally, find the rectangular equatorial coordinates:

$$X = Rcos(\epsilon)cos(\lambda)$$

$$Y = Rcos(\epsilon)sin(\lambda)$$

$$Z = Rsin(\epsilon)$$

Where ϵ is the obliquity of the ecliptic and can be found using the following equation:

$$\epsilon = 23.439^\circ - 0.0000004^\circ n$$

3.1.3.3 Spacecraft Detumbling

Because of the universal nature of CubeSat deployment systems, no system can truly guarantee to deploy a cubesat with zero angular velocity or even with a known angular velocity. As such, any CubeSat deployed will immediately begin tumbling, even before the satellite can activate any sort of attitude control system. Because the angular velocity during this tumbling state is unknown, generally makes it difficult to complete mission requirements, and can be much higher than angular velocities that occur throughout the rest of the mission, it is important to detumble the CubeSat after deployment.

3.1.3.4 B-dot Control

One of the detumbling methods common in CubeSat missions is B-dot control. The B-dot controller detumbles a spacecraft by commanding a magnetic dipole moment with magnetorquers while measuring the time derivative of the local geomagnetic field in the spacecraft body frame. This produces a resultant torque counter to the spin of the spacecraft, which will reduce the magnitude of angular velocity until a point at which the mission can begin. The magnetic dipole moment commanded by this control law can be expressed in vector notation as μ, where

$$missingequation \tag{11}$$

where k is the controller gain, $\vec{B}$ is the Earth's magnetic field, and

$$vecb = \frac{\vec{B}}{\|\vec{B}\|} \tag{12}$$

If the spacecraft's angular velocity is known, such as through rate gyroscope measurements, the commanded torque can be written as

$$vecm = \frac{k}{\|\vec{B}\|}\vec{\omega} \times \vec{b} \tag{13}$$

where $\vec{\omega}$ is the angular velocity of the spacecraft. This can be done by making the assumption that the change in the Earth's magnetic field due to change in orbital position occurs much more slowly than the change in the magnetic field in the spacecraft body frame due to the tumbling motion. The B-dot control law can then be written as

$$\vec{L} = \vec{m} \times \vec{B} \tag{14}$$

In order to make it easier to prove the stability of the control law, 14 can be rewritten as

$$\vec{L} = -k(I_3 - \vec{b}\vec{b}^T)\vec{\omega} \tag{15}$$

To prove the stability of a control law, it has to be shown to reduce a positive Lyapunov function asymptotically to zero by making the derivative of the function less than or equal to zero for all cases. For detumbling, we can use the Lyapunov function

$$V = \frac{1}{2}\vec{\omega}^T J \vec{\omega} \tag{16}$$

where J is the spacecraft moment of inertia matrix. This function is analogous to rotational kinetic energy, making it useful for this case, as the goal of the detumbling control law is to reduce the angular velocity. With this Lyapunov function and applying the B-dot control law written in 16, the derivative of the Lyapunov function can be written as

$$\dot{V} = -k\vec{\omega}^T(I_3 - \vec{b}\vec{b}^T)\vec{\omega} \tag{17}$$

Because the eigenvalues of $(I_3 - \vec{b}\vec{b}^T)$ are always 0, 1, and 1, $(I_3 - \vec{b}\vec{b}^T)$ is positive semidefinite. Therefore, $\dot{V}$ is always less than or equal to zero. The only case where $\dot{V}$ is equal to zero and global asymptotic stability cannot be obtained is when $\vec{\omega}$ is parallel to $\vec{b}$, in which case the spacecraft is already not tumbling. In this case, it would be in spin, and the control algorithm can move to initial attitude determination.

In order to use 13 in a bang-bang implementation, it can be rewritten as

$$\vec{L} = \sum_{i=1}^{N} -m_i^{max} sign(\vec{u_i} \cdot \dot{\vec{B}}) \tag{18}$$

where N is the number of magnetorquers on the spacecraft, m_i^{max} is the maximum moment the $i-th$ magnetorquer can delivery, and u_i is the direction of the magnetic moment for the $i-th$ magnetorquer. This implementation is less computationally expensive than the other implementation, but less efficient in terms of power consumption. Ultimately, this is the implementation that we have chosen because of the computational savings.

3.1.4 ADC Algorithms and Methods

The following subsections outline the ADC methods analyzed and chosen for our CubeSat. For each of the three attitude control phases, a different method of ADC was needed. With mission parameters specified, a generic block diagram was created to show a high level overview of the ADCS. Structure is shown in figure n below. From the graphic, we were able to gage how many of each sensor and actuator we would need, and what parameters we needed to model.

3.1.4.1 Initial Attitude Determination

In order to begin to understand what attitude determination and control algorithms were needed for this project, the team first needed to understand the two main categories of ADC, and how previous project teams applied them to their cubesat. The first subcategory is recursive ADC. Recursive methods work by comparing the current attitude to the most recent attitude. A recursive function is a function that calls itself, so this form of ADC provides an updated state estimate using its previous state estimate. For example, if the current attitude quaternion is denoted by qn then the recursive method would estimate

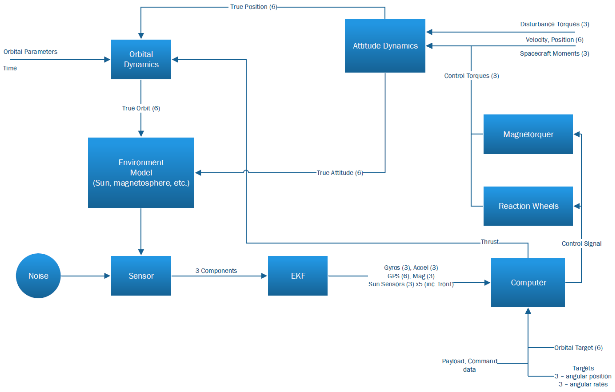


Figure 8: ADCS Flow Chart

that value by using information from its previous estimate, qn-1. The second type of ADC method is deterministic ADC. Deterministic methods use current sensor readings as well as reference readings to calculate an attitude estimate based on the difference between the two values.

Previous MQP teams used the deterministic TRIAD method for their attitude determination and control. This was used in unison with noise filtering methods in an attempt to minimize TRIAD error while also producing an optimal quaternion for initial attitude determination [2017 citation here]. The TRIAD method was used as a baseline to get a directional cosine matrix. With that information, newer methods were then used to produce an accurate quaternion from the TRIAD output (Farhat 2013). Due to the widespread application of this approach, as long with its efficiency, our team decided to maintain the same type of analysis for our project.

3.1.4.2 TRIAD Algorithm

Harold Black's 1964 TRIAD algorithm was the first published ADC method. It combines information from the unit vector to the Sun along with Earth's magnetic field vector. This method requires two sets of each vector. Two observation vectors are obtained by the magnetometer and the sun sensors, while two reference vectors are obtained by using a geomagnetic model of the atmosphere as well as a sun vector model. Values for the reference vectors are determined by the position of the satellite at a given time in the geodetic (Earth-fixed) reference frame (Black). These two sets of unit vectors over determine the attitude of the cubesat, which is useful given that many of the on board components are commercial over the shelf parts with high enough error to be insufficient as the only reference sensor. The TRIAD method allows for coordinates in the body fixed reference frame to be rotated into the Earth-fixed inertial reference frame. This, of course, is very useful for understanding where in the orbit the Cubesat is. However, because vectors must be normalized for this method, it is possible that there could be a decent amount of sensor noise. Later research, specifically by Grace Wahba, poses a potential way of minimizing input sensor error, thus making the TRIAD method more accurate. Solutions to Wahba's problem include Davenport's q-method and QUEST (Markley). These methods provide and optimum quaternion for expressing the spacecrafts attitude. The TRIAD method begins by defining two linearly independent body fixed vectors, b1and b2as well as their corresponding reference frame vectors r1and r2determined from Earth's magnetic field model and the Sun's direction to Earth. The attitude matrix, A, serves as the matrix that rotates the body fixed vectors into the earth fixed frame. Two rotation matrices can be defined in a way such that A1=A2,where:

$$Ar_i = b_i \text{for} i = 1, 2$$

However due to sensor error, we can not confidently say that A1=A2. The TRIAD method also assumes that one of the unit vectors, b1, is much more accurately determined than its counterpart. Therefore equation 2.2 is exactly satisfied for i=1 but is only an approximate for the case where i=2. The TRIAD algorithm is based off of two sets of orthonormal right-handed triads of vectors. One for the reference frame as well as one for the body frame (Markley). These triads are denoted as:

$$M_{obs} = v_1 \; v_2 \; v_3$$

$$v_1 = r_1$$
$$v_2 = r_x = \frac{r_1 \times r_2}{|r_1 \times r_2|}$$
$$v_3 = r_1 \times r_x$$
$$M_{ref} = w_1 \; w_2 \; w_3$$

$$w_1 = b_1$$
$$w_2 = b_x = \frac{b_1 \times b_2}{|b_1 \times b_2|}$$
$$w_3 = b_1 \times b_x$$

Substituting the triads into equation 2.2 the direct cosine matrix can be obtained. Equation 2.2 can then be rewritten as:

$$AM_{obs} = M_{ref}$$

Substituting in the relations for each individual vector in each triad and simplifying, the equation for the direct cosine matrix becomes:

$$A_{TRIAD} = b_1 r_1^T + (b_1 \times b_x)(r_1 \times r_x)^T + b_x r_x^T$$

This allows for coordinates to be rotated to the body fixed frame and vice versa, using a little linear algebra. It is important to note however, that for cases where either the reference vectors or observed vectors are parallel or antiparallel equation 2.12 is undefined, meaning the attitude matrix cannot be found. It is also easy to see how the assumption made earlier (that b1is entirely accurate) can lead to errors in the cosine matrix. The formula for ATRIAD heavily relies on vectors with subscript 1 which in practice have some degree of error associated with them (Markley). This means solutions to Wahba's problem have to be used to minimize this error.

3.1.4.3 Wahba's Problem

Mathematician Grace Wahba attempted to describe issues associated with using direction cosine matrices in attitude estimation and provide a way to build upon the TRIAD method. Improvements include, adding a way to weigh sensor measurements and allowing for more than 2 sets of measurements to be used. To understand how to improve upon the TRIAD method, one must understand the problem Wahba proposed. Wahba's problem serves to find an orthogonal matrix with a positive 1 determinant that minimizes the loss function:

$$L(A) = \frac{1}{2}\sum_{i=1}^{N} a_i||b_i - Ar_i||^2$$

This essential finds the rotation matrix A that brings the first set of unit vectors (b1,b2,. . . ,bn)into the best least squares coincidence with the second set of vectors (r1,r2,. . . ,rn). Where bi is a set of N unit vectors in the spacecraft body frame, ri is the corresponding set of vectors in the reference frame, and ai represents the non-negative weights of each sensor. These weights must be applied according to the accuracy of the sensors. This is necessary to relate Wahba's problem to Maximum Likelihood Estimation, a technique that uses sensor accuracy to help to more accurately model a system. Using the orthogonality of A, the unit norm of the unit vectors, and the cyclic invariance of the trace we can rewrite L(A) as:

$$L(A) = \lambda_0 - tr(AB^T)$$

$$\lambda_0 = \sum_{i=1}^{N} a_i \text{and} B = \sum_{i=1}^{N} a_i b_i r_i^T$$

Solutions to Wahba's problem include Davenport's q method, Singular Value Decomposition, Quaternion Estimator (QUEST) and Estimators of the Optimal Quaternion (ESOQ), and start with the rewritten error estimation above. Through various real world applications, QUEST and ESOQ are deemed significantly faster than their robust counterparts.

3.1.4.4 Quaternions

For the orientation of the spacecraft to be determined, controlled, or otherwise used, there must be a way of expressing it. A common way of doing this in other applications is with a sequence of Euler angles. This information can be determined and operated on as a three element vector and is generally enough to uniquely determine the orientation. For some applications, especially spacecraft, three Euler angles cannot determine the orientation because of singularities in the determination when an angle is near 90 degrees. For this reason, spacecraft often use quaternions, which are four element vectors with definitions for additional operations. Quaternions are composed of a three element "vector part" and a single "scalar part," and because of the additional element, there are no singularities when expressing orientation with quaternions (Crassidis). Quaternions are also computationally easy to work with because most calculations can be done with quaternion operations instead of trigonometry. Just like most other attitude representations, quaternion attitudes can be converted to other systems, such as Euler angle representations, which are more intuitive to visualize. Because of these advantages, quaternion attitude representations are great for a cubesat mission, as the computations are less taxing for the OBC and the lack of singularities allows for diverse mission profiles.

3.1.4.5 Davenports Q Method

Mathematician Harold Davenport proposed a solution to Wahba's problem that maximizes the following gain function, which is a quantity from eq. (wahba's problem):

$$g = tr(AB^T)$$

He specified a rotation matrix A that can be expressed in terms of euler parameters:

$$A = (q_0^2 - \epsilon^T\epsilon) * [I_{3\times3}] + 2\epsilon\epsilon^T - 2q_0|\epsilon|$$

Where ϵ =(q1,q2,q3)and q0 is the scalar term of the quaternion

The Gain function g() can then be written in terms of the 4x4 K matrix:

$$g(\bar{q}) = q^T[K]q$$

$$K = \begin{bmatrix} \sigma & Z^T \\ Z & S - \sigma[I_{3\times3}] \end{bmatrix}$$

$$B = \sum_{i=1}^{N} a_i b_i r_i^T$$

$$|S| = [B] + [B]^T$$

$$\sigma = tr([B])$$

$$[Z] = [B_{23} - B_{32}\ B_{31} - B_{13}\ B_{12} - B_{21}]^T$$

In order to maximize the gain function, we must abide by the unit length constraint, which means we cannot set the values to infinity. Due to this, a lagrange multiplier is needed to yield a new gain factor g'.

$$g(q) = q^T[K]q - \lambda(q^Tq - 1)$$

Differentiating and setting the equation equal to zero, will find the extrema of the function

.

$$0 = \frac{d}{dq}(g(q)) = 2[K]q - 2\lambda q$$

which can, alternatively, be expressed as:

$$[K]q = \lambda_{max}q$$

Therefore, the desired euler parameter vector is the eigenvector of the K matrix. In order to maximize the gain, we have to choose the largest eigenvalue. Through substitution of equation 3.9 into equation 3.7, and applying some linear algebra, it is easy to show that:

$$g(p) = \lambda$$

3.1.4.6 QUEST

The QUEST method (QUaternion ESTimator) builds off of davenport's Q method, and allows for more frequent attitude computations. Currently, it is the most widely used algorithm for solving Wahba's problem, and was first used in 1979 by the MAGSAT spacecraft (crassidis). The algorithm begins by rewriting equation (whatever 3.9 is in the new doc lmao), into two separate equations:

$$[(\lambda_{max} + trB)I_3 - S]\hat{q}_{1:3} = \hat{q}_4 z$$

$$(\lambda_{max} - trB)\hat{q}_4 - \hat{q}_{1:3}z = 0$$

From there, we can rewrite q using the classical adjoint representation, knowing that the adjoint divided by the determinant gives the inverse of a matrix.:

$$q_{1:3} = q_4((\lambda_{max} + trB)I_3 - S)^{-1}z = \frac{q_4(adj((\lambda_{max} + trB)I_3 - S)z)}{\det((\lambda_{max} + trB)I_3 - S)}$$

From there, we can use the Cayley-Hamilton theorem, which states that every square matrix over a real or complex field satisfies its own characteristic equation [cite], the previous equation can be rewritten in terms of the classical adjoint. This theorem holds for general quaternionic matrices, which is the case for this part of the QUEST algorithm and gives the optimal quaternion estimate as follows:

$$q = \alpha \begin{bmatrix} adj(\rho I_3 - S)z \\ \det(\rho I_3 - S) \end{bmatrix} \rho = \lambda_{max} + trB$$

The term is determined by normalizing the resultant q. This equation assumes we know the value of max, which is the solution to the characteristic equation of the K matrix. Finding the maximum eigenvalue of the characteristic equation is complicated for states using more than two observations. However, in the case of our CubeSat, we are only using two observations, which means the characteristic equation for K has a simple closed form solution. Several simplifications result from only having two observations, which ends up yielding the following equation for λ_{max}:

$$\lambda_{max} = a_1^2 + a_2^2 + 2a_1a_2[(b_1 \odot b_2)(r_1 \odot r_2) + ||b_1 \otimes b_2||||r_1 \otimes r_2||]^{\frac{1}{2}}$$

3.2 Command and Data Handling

The Command and Data Handling (C&DH) subsystem is composed of many different components and systems whose purpose is to manage the entire spacecraft throughout its mission. This includes during the startup, deployment, detumble, and general mission phases. The key components of the C&DH system are the onboard computer, flight software, radio, and data storage systems. The C&DH system is responsible for all aspects of the spacecraft's operation. It implements the control system, taking sensor data and issuing commands to the thrusters and ADC systems. It also collects, stores, and transmits sensor and payload data to ground stations. The individual component selection and overall architecture of the C&DH system is dependent on the overall needs of the mission and scientific payload.

3.2.1 Data Handling Requirements

The needs of the scientific payload drives the majority of the data handling requirements. Given the processing power of modern processors, little consideration need be given to the needs of the ADC system. The mission profile of the satellite is relatively simple, requiring only basic slew operations, well within the capabilities of the onboard computer. In comparison the payload will be producing large amounts of data that needs to be stored, possibly operated on, and eventually transmitted to ground stations. Without information about the payload it is difficult to assess the exact needs of the system. Assume that the payload only produces data during the eLEO portion of the orbit, about five minutes worth of data. It would be impractical to attempt to transmit this data as it is collected. At a minimum the data must be stored for transmission later. Due to inconsistencies in ground station coverage it may not be possible to down-link all of the collected data at once depending on orbital position. Thus there must be sufficient storage space to collect data over the course of several orbits. At this time more information about the payload and ground station coverage is needed to make a comprehensive assessment of the data storage needs.

3.2.2 Onboard Computer

We have selected the Clyde Space Kryten-M3 flight computer as our onboard computer (OBC). The Kryten-M3 is a flight proven OBC built around a Cortex-M3 processor core running at 50 MHz. Similar OBC's from Clyde Space were also chosen by previous MQPs (cite previous MQP). The OBC includes 8 MB of MRAM and 4 GB of bulk flash memory. Both the MRAM and bulk storage include automatic error detection and correction (EDAC). This is especially important for correcting potential errors introduced by the high radiation environment of orbit. The system also includes space for an external SD card, increasing the bulk storage capacity (citation from clydespace).

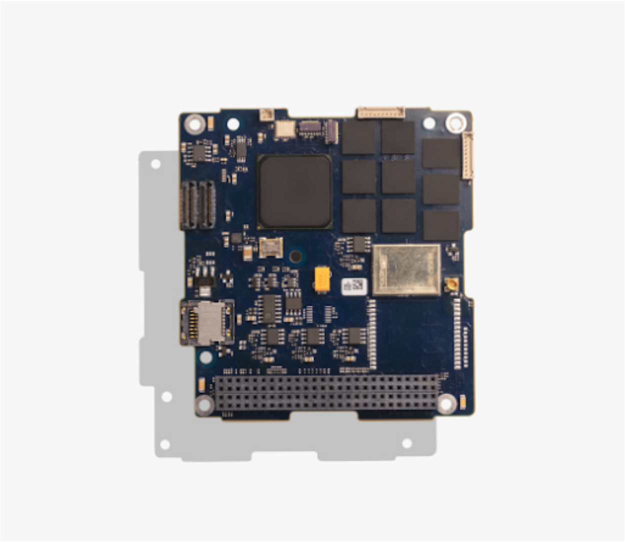


Figure 9: Kryten-M3 Onboard Computer

The Kryten-M3 is designed for use with FreeRTOS, a real-time operating system commonly used in high reliability embedded applications. FreeRTOS is highly configurable and capable of managing the entire spacecraft and scientific payload (citation from clyde). Once more information is available about the payload there will be a discussion of the processing and data storage needs of the OBC.

3.2.3 Radio Transceiver

The radio transceiver enables communication between the satellite and ground stations. Different radios operate in different bands and are capable of different data rates. Certain frequency bands are regulated by government agency and require specific approval to use. Depending on the data handling needs of the mission it may be necessary to use a radio capable of a higher data rate. As a result it may be necessary to consider the regulatory approval process needed to operate such radio systems. Once we have more information from the telecoms team about radio communications and ground coverage it will be discussed here.

3.3 ADCS Testbed

4 Analysis

4.1 Detumbling

Below is the analysis for the detumbling simulation, created in Simulink, the first stage of attitude control. Without successfully detumbling the spacecraft after deployment, the mission would not be able to continue to the next phase. The team simulated the detumbling of the CubeSat using Simulink. This was done by simulating the on board magnetometers and applying B-dot control theory to actuate the magnetometers. An option to use the CubeSat's gyroscope was also simulated.

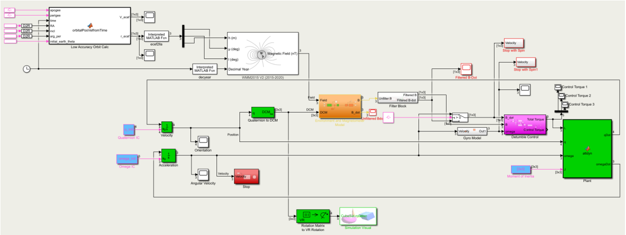


Figure 10: Total Detumbling Simulink Simulation

Overall, the detumbling of the satellite was accomplished by providing inputs into our custom Simulink function 'attdyn' or attitude dynamics. The attitude dynamics block required an input control torque, quaternion, angular velocity, and moment of inertia, in order to output the rate of change of the quaternion ($\dot{q}$) and the rate of change of the angular velocity ($\dot{\omega}$). Moment of inertia was a constant input into the system and was given by the geometry of our CubeSat.

The attitude dynamics block uses the inputted quaternion to create a quaternion product matrix, denoted $\Xi(q)$, that speeds up the quaternion multiplication process.This is because $\Xi(q)$ is the same as $[q\odot]$, which is a common quaternion operator.

From there, expressions for $\dot{q}$ and $\dot{\omega}$ can be derived and outputted.

$$[L] = \begin{bmatrix} L(1) \\ L(2) \\ L(3) \end{bmatrix} \tag{19}$$

$$[q] = \begin{bmatrix} q(1) \\ q(2) \\ q(3) \\ q(4) \end{bmatrix} \tag{20}$$

$$[\omega] = \begin{bmatrix} \omega(1) \\ \omega(2) \\ \omega(3) \end{bmatrix} \tag{21}$$

$$[\Xi(q)] = \begin{bmatrix} q(4) & -q(3) & q(2) \\ q(3) & q(4) & -q(1) \\ -q(2) & q(1) & q(4) \\ -q(1) & -q(2) & -q(3) \end{bmatrix}$$

$$\dot{q} = \frac{1}{2}\Xi_q\omega$$

$$\dot{\omega} = \frac{J}{\omega \cdot J\omega}$$

The quaternion and angular velocity inputs were first given by an initialized value from the simulations initialization file. Then, the outputted $\dot{q}$ and $\dot{\omega}$ got fed back into the Simulink loop, and integrated using a built in continuous integration block in Simulink. The resulting quaternion and angular velocity (ω), then got passed back into the 'attdyn' block.

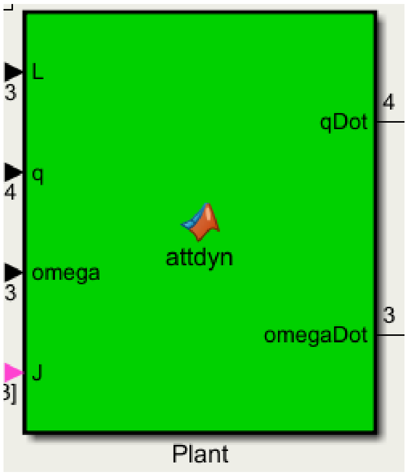


Figure 11: Attitude Dynamics Plant

Inputting the control torques into the 'attdyn' block was a little more complicated and required the simulation of the magnetometers and the application of B dot control.

4.1.1 Simulating the Input Control Torques

The first step in finding the necessary control torques for our attitude dynamics block, was to simulate the magnetometers. This was done with our custom Simulink function ‘Environment and Magnetometer Model.’ In order to use this function, we needed to take the inertial magnetic field reading and convert it to the body fixed frame by multiplying it by the spacecraft’s body fixed attitude.

To find the inertial magnetic field vector, we simulated the following path shown below.

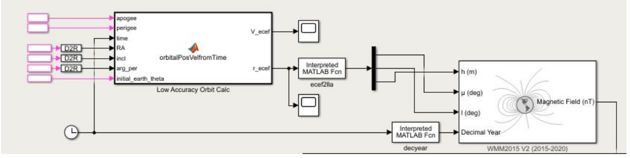


Figure 12: Simulation Path to Find Inertial Magnetic Field Vector

On the left, in the pink outlined boxes, are the constants that feed into the simple two-body time based orbit propagator. The constants fed into the orbit propagator block are the apogee, r_a, perigee, r_p, time, RA, inclination, i, argument of perigee, w, and the initial earth theta. The apogee, perigee and initial earth theta, θ_{i_earth} are unaffected constants. The RA, inclination and argument of perigee are entered as constants, in degrees, and translated into radians using a “Degrees to Radians” converter within Simulink. The “Degrees to Radians” converter follows the equation below.

$$Radians = \theta * \frac{\pi}{180}$$

The final input, time, t, was specified by the clock input block which outputs the current simulation time as the simulation was running.

These seven inputs were then sent through the low accuracy orbit propagation block to output V_{ecef} and r_{ecef}.

In order to find V_{ecef} and r_{ecef}, the following process was used. To start, the μ of Earth, the semi-major axis, a, period, p_t, the mean motion, in radians, n, and time since perigee, TSP.

$$\mu = 3.986e10^{14}$$

$$a = \frac{r_a + r_p}{2}$$

$$p_t = 2\pi * \sqrt{\frac{a^3}{\mu}}$$

$$n = \frac{2\pi}{p_t}$$

Time Since Perigee is equal to the remainder of $\frac{t}{p_t}$.

After this, the mean anomaly, M, the specific energy, and the eccentricity are found.

$$M = nTSP$$

$$\epsilon = \frac{-\mu}{2a}$$

$$e = \frac{r_a - r_p}{2a}$$

After this, we then need to compute the numerical approximation of the Inverse Kepler Equation. First, you need to find the eccentric anomaly, E. To do this, you initialize the values E_n and E_{nplus}, being 10 and 0 respectively. Then, we ran E_n and E_{nplus} through a while-loop that states as long as the absolute value of E_n and E_{nplus} is greater than 0.001, E_n and E_{nplus} equal each other, therefore, $E_{nplus} = E_n - (\frac{(E_n - \epsilon \sin(E_n) - M}{1 - \epsilon \cos(E_n)})$. Then, the output of the while-loop, E_{nplus} is equal to the eccentric anomaly, E. Using E, we then found the true anomaly, ν, the distance, d, the angular momentum, h, and the orbital parameter, p.

$$\nu = 2atan(\sqrt{\frac{1+e}{1-e}})tan(\frac{E}{2})$$

$$d = a(1 - e\cos(E))$$

$$h = \sqrt{\mu a(1 - e^2))}$$

$$p = \frac{h^2}{\mu}$$

From here, we are then able to compute the position components of the spacecraft.

$$x = d(\cos(RA)\cos(w+\nu) - \sin(RA)\sin(w + \nu\cos(i))$$

$$y = d(\sin(RA)\cos(w+\nu) - \cos(RA)\sin(w + \nu\cos(i))$$

$$z = d(\sin(i)\sin(w+\nu))$$

Using these, we find our position vector:

$$Position = [xyz]$$

From there, we are then able to find the velocity components.

$$\dot{x} = \frac{(x)(h)(e)}{(d)(p)}\sin(\nu) - \frac{h}{d}(\cos(RA)\sin(w+\nu) + \sin(RA)\cos(w+\nu)\cos(i))$$

$$\dot{y} = \frac{(y)(h)(e)}{(d)(p)}\sin(\nu) - \frac{h}{d}(\sin(RA)\sin(w+\nu) - \cos(RA)\cos(w+\nu)\cos(i))$$

$$\dot{z} = \frac{(z)(h)(e)}{(d)(p)}\sin(\nu) + \frac{h}{d}\sin(i)\cos(w+\nu)$$

From there, we can calculate the velocity vector:

$$Velocity = [\dot{x}\ \dot{y}\ \dot{z}]$$

Using ω_{earth}, which is equal to $7.29e^{-5}$ rad/s, we can find the true anomaly at time t, θ_t.

$$\theta_t = \omega_{earth}t + \theta_{i_{earth}}$$

Next, we defined a normal rotation matrix that includes θ_t.

$$[RM] = \left(\begin{bmatrix} cos(\theta_t) & sin(\theta_t) & 0 \\ -sin(\theta_t) & cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

Finally, to find the position of the spacecraft in ECEF, we transposed the product of the rotation matrix and the initial position vector defined above. Finding the velocity of the spacecraft in ECEF was a bit more complicated, but defined below.

$$[V_{ecef}] = RM\left(V\begin{bmatrix} \omega_{earthY} \\ \omega_{earthX} \\ 0 \end{bmatrix}\right)$$

Where V is the velocity vector defined above.

After we found V_{ecef} and r_{ecef}, we then converted r_{ecef} from ECEF to LLA using a built in Simulink block in the Aerospace toolbox. From there, the r_LLA coordinates filter into a Demux block in Simulink. The Demux block split the vector signals into scalar values. This specific Demux block splits the input into three scalar outputs, height, h, in meters, latitude, μ, in degrees, and longitude, l, in degrees. These three scalar outputs then flow into the World Magnetic Model, WMM2015, that is also built into Simulink, along with the decimal year, which is created by a clock and a built in MatLab function that converts the input clock time, that's equal to the time input in the orbital propagator, to decimal years. The World Magnetic Model implements the mathematical representation of the National Geospatial Intelligence Agency (NGA) World Magnetic Model, which is then used to find the Magnetic field vector, in nT, which is then fed directly into the Environmental and Magnetometer Model as one of the inputs.

To find the body fixed attitude, the quaternion within the detumbling loop was fed into a built in Simulink function that converted quaternions to a direct cosine matrix. Once we had both the inertial magnetic field reading, and the direct cosine matrix representation of the spacecraft's attitude, a simple matrix multiplication yielded our true magnetic field in body fixed coordinates.

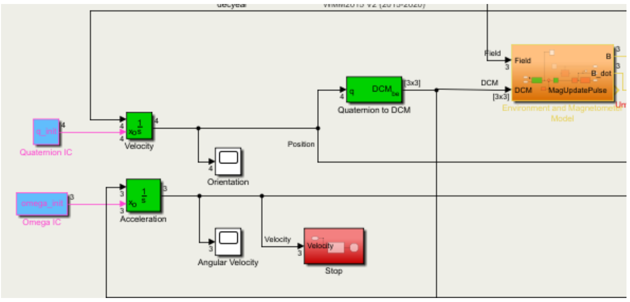


Figure 13: Integration of the q and omega vectors and Simulink q to DCM block

This output was then seeded with noise via a noise maker we created, to simulate how a real sensor would operate. The noise characteristics we inputted into the simulation are on par with those listed in the spec sheet for our magnetometer, and provided a random Gaussian distribution that was added to the true magnetic field vector. Next, the magnetic field with the addition of noise was passed through a Simulink Quantizer that discretizes our input to a specific interval. For our case, the quantization value was set to 200, which was an arbitrary value that we found worked well after some trial in error. The quantization block works by using a quantization algorithm with a round-to-nearest method to map signal values to quantized values at the output that are defined by the quantization interval. After the vector is quantized, it is passed through a built in Simulink function called 'Zero-Order Hold.' This block essentially sets the sample rate of our signal to any specific value. We set our value to $\frac{1}{220}$, or 220 Hz to be consistent with our sensors refresh rate. For comparison purposes, a 'Discrete Derivative' block was added in order to see the unfiltered rate of change of the magnetic field, or $\dot{B}$. This was to ensure that once we passed the magnetic field through our low pass filter, the filter was properly working.

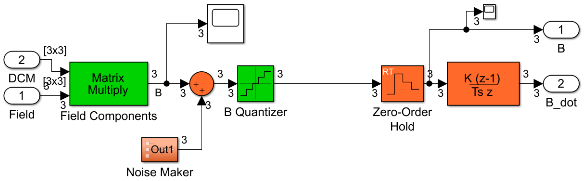


Figure 14: Contents of the Magnetometer Simulation Block

The next step was to pass our outputted B vector from our 'Environment and Magnetometer Model' through a low pass filter to filter out the noise we previously added.

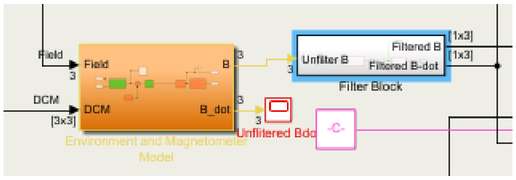


Figure 15: Magnetometer Simulation Block and Low Pass Filter

Simulink allows for users to specify passband edge frequency and stopband edge frequency in Hz. Those values were set to 90 and 120 Hz respectively. This was due to the sample rate of our magnetometer being 220 Hz. Any fast, small changes in magnetic field vector are most likely errors, so we want to remove any changes that aren't happening slowly.

In theory, this also means that if we were spinning fast enough, it would filter out that too, and make the satellite spin faster. However, because we are using gyroscope measurements as a second reference, this situation would not occur. The derivative of the signal was then found using another 'discrete derivative' block, to give us the filtered rate of change of the magnetic field, or $\dot{B}$.

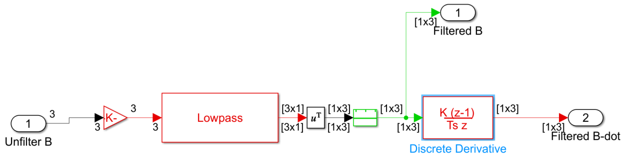


Figure 16: Contents of the Low Pass Filter Block

4.2 Reaction Wheel Sizing

The reaction wheels selected for the CubeSat had to be capable of producing enough torque to counteract the disturbance torques on the satellite. First, the gravitational torque was defined by:

$$T_g = \frac{3M}{2R^3}|I_x - I_y|sin(2\theta)$$

Where θ is equal to 5°, or 0.0872005 radians, M is the mass of the Earth, $3.9816x10^{14}\frac{m^3}{s^2}$ and R is the radius of the earth, $6.978x10^6 m$. The next disturbance torque needed to be accounted for was the torque due to solar pressure. This was defined by:

$$T_{sp} = F(C_{ps} - C_g)$$

Where C_{ps} is the center of solar pressure and C_g is the center of gravity. The flux due to solar pressure, F is defined as:

$$F = \frac{F_s}{c}A_s(1+q)cos(l)$$

Where l is the angle of incidence to the Sun, F_s, is the solar flux constant , $1.367\frac{W}{m^2}$, c is the speed of light, $3x10^8\frac{m}{s}$, A_s is the surface area of the RAM facing surface, 0.02 m, and q is the coefficient of reflection, 0.6. Next We can then calculate the torque due to the magnetic field by the following equation:

$$T_m = DB$$

Where D is equal to the dipole moment of the vehicle and B is the Earth's magnetic field. The final disturbance torque to consider was the torque due to aerodynamic pressure. This was defined as:

$$T_a = 0.5[\rho C_d A_s V^2](C_{pa} - C_g)$$

Where V is the velocity of the vehicle and C_{pa} is the center of aerodynamic pressure. With all of the disturbance torques calculated and summed, we now had a known value for the disturbance torques we needed to overcome. To find the torque for the reaction wheel sizing, T_{rw}, we evaluated the following expression:

$$T_{rw} = T_D(M_f)$$

Where M_f is the margin factor to help calculate the torque of the reaction wheel for the disturbance rejection and T_D is the reaction wheel torque for the worst case anticipated torque. The Reaction wheel torque, T_{rw} must be equal to the worst case anticipated disturbance torque plus some margin. Finally, the momentum storage can be calculated by:

$$h = T_D \frac{t}{4}(0.707)$$

Where t is the orbital period, in seconds and 0.707 is the rms average of a sinusoidal function.

4.3 Simulink Control

4.3.1 Detumble Control Subsystem

This is the subsystem that takes the filtered measurement readings and uses them to command control torques to detumble the spacecraft. There are two different control laws that can be chosen in this block by setting a variable in the initialization file depending on the desired control method.

4.3.1.1 Bang-Bang B-dot Control

The first control law block made for this model implements the control law described in 18 using an estimate of $\dot{\vec{B}}$ and a preset maximum for magnetorquer dipole moment. The block sums up the individual commanded dipole moments from each of the magnetorquers and the finds the resultant torque using 14. This resultant torque, $\vec{L}$, is then passed to the rest of the subsystem.

4.3.1.2 Proven-Stability B-dot Control

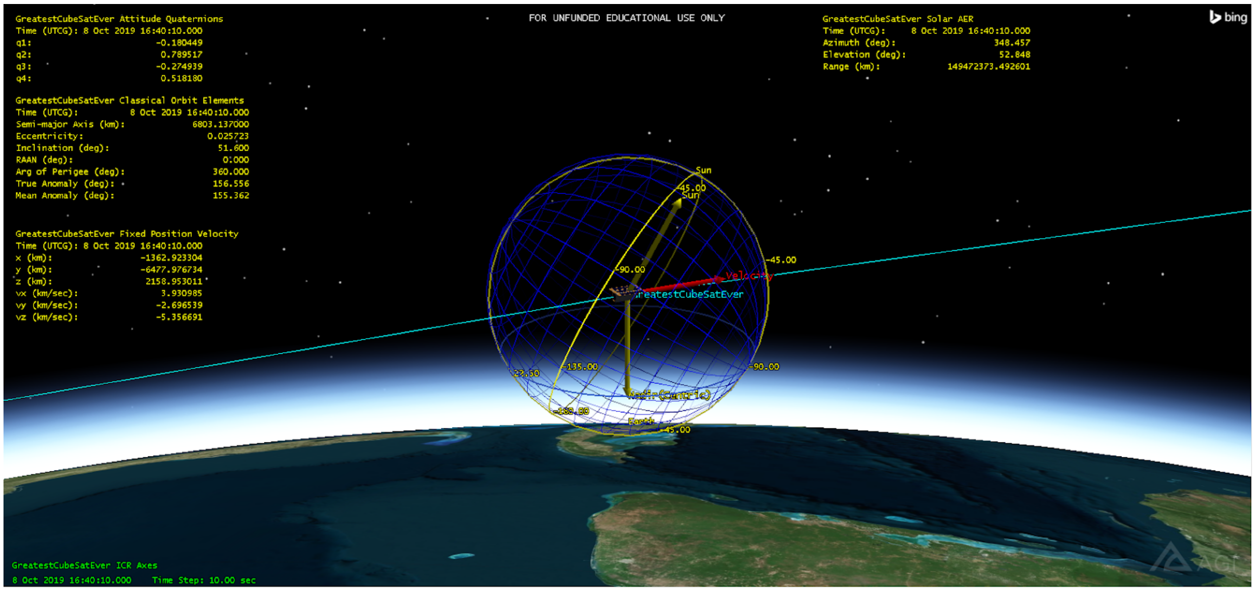
The second control law block, it is a direct improvement on the first control law because of the use of a direct measurement of the angular velocity. This block implements 15 to find a resultant torque given measurements from the gyroscopes and the magnetometers. Because this control law does not use $\dot{\vec{B}}$, which is found by taking discrete differences of a corrupted signal, it is more effective than the other controller. In addition, as was shown in an earlier section, this control law is has proven Lyapunov stability. The advantages of this control law far outweigh the downside of having to also use the gyroscopes for this law, so this is the control law that is used for our analysis. The option of using the other law is still available, although discouraged.

4.3.1.3 Control Activation Delay

In order to meet any potential guidelines for launching from the ISS, this block gives us the option of delaying the activation of the active controls until a set amount of time has passed. This also has the benefit of allowing our controls to ignore the initial transient signals that result from differences between initial conditions and initial estimates upon simulation start up. This block is very simple and does not apply a signal delay to the commanded torques when they are allowed to pass. In addition, it does nothing to any disturbance torques that may be added to the simulation regardless of simulation time or set delay.

4.4 STK

Systems Tool Kit, or STK, is a physics-based software packaged created by Analytical Graphics (AGI). STK allows engineers to perform analyses of ground, sea air and space objects in their respective environments. STK is used in government, commercial and defense applications around the world to determine the attitude, spatial relationships and time-dynamic positions of objects under many simultaneous constraints. STK's ability to simulate subsystems like ADC is what makes the software so valuable to the aerospace industry. Our team considered using STK to simulate the detumbling and attitude control of our 6U CubeSat. An example of the simulation can be seen below.



```
GreatestCubeSatEver Attitude Quaternions
Time (UTCG): 8 Oct 2019 16:40:10.000
q1:                         -0.180449
q2:                          0.789517
q3:                         -0.274939
q4:                          0.518180

GreatestCubeSatEver Classical Orbit Elements
Time (UTCG):           8 Oct 2019 16:40:10.000
Semi-major Axis (km):              6803.137000
Eccentricity:                         0.025723
Inclination (deg):                      51.600
RAAN (deg):                              0.000
Arg of Perigee (deg):                  360.000
True Anomaly (deg):                    156.556
Mean Anomaly (deg):                    155.362

GreatestCubeSatEver Fixed Position Velocity
Time (UTCG): 8 Oct 2019 16:40:10.000
x (km):                    -1362.923304
y (km):                    -6477.976734
z (km):                     2158.953011
vx (km/sec):                   3.930985
vy (km/sec):                  -2.696539
vz (km/sec):                  -5.356691
```

```
FOR UNFUNDED EDUCATIONAL USE ONLY
```

```
GreatestCubeSatEver Solar AER
Time (UTCG):      8 Oct 2019 16:40:10.000
Azimuth (deg):                    348.457
Elevation (deg):                   52.848
Range (km):            149472373.492601
```

```
GreatestCubeSatEver ICR Axes
8 Oct 2019 16:40:10.000    Time Step: 10.00 sec
```

Figure 17: STK Satellite View

The figure above contains a snapshot of our CubeSat in its simulated elliptical orbit. STK has the ability to overlay particular orbital elements as vectors, such as sun vector and the nadir vector. Attitude determination in STK is done primarily with the use of quaternions. We originally intended on integrating directly between STK and MATLAB. MATLAB was to be used as our attitude determination and controller, whereas STK was simply to be used as a visual simulation, with the potential of utilizing it for sensor readings such as magnetic field and the sun vector. Utilizing the work of Xiang Fang and Yunhai Geng of the Research Center of Satellite Technology of Harbin Institute of Technology (CITATION) Fang and Gent laid out detailed steps into the integration of MATLAB into a real-time running STK environment. MATLAB was to be used to introduce sensor noise, whether that be from MATLAB simulated sensors, or STK sensors with their data passed into MATLAB. We would then pass the noisy sensor data through an extended Kalman Filter (EKF) prior to being sent to the controller. This EKF-processed data could be compared to the full state vector to check on the functionality of our EKF. Attitude control commands would then be sent to the controller flow, and then pass the data back into the STK environment for visual simulation, completing the loop.

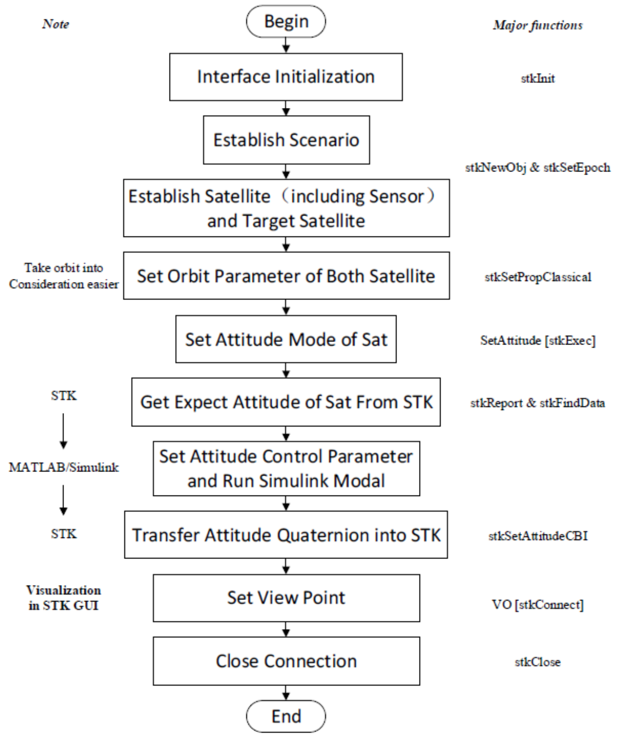


Figure 18: STK-Matlab Integration Workflow

Although without any code snippets or a codebase to be referenced, it is possible to integrate MATLAB into STK using the above figure and aforementioned paper as a guide. However, our focus changed from purely using MATLAB for our attitude and orbital simulation and control, to primarily utilizing Simulink into our control scheme.

Since changing our focus to Simulink, the MATLAB to STK integration has become less realistic of a goal, especially with the introduction of the Aerospace Blockset CubeSat Simulation Library, which included an orbital propagator, which simulates in 3-D the orbital and attitude elements of a CubeSat. We will utilize this CubeSat Simulation Library along with using a Level-2 S-Function block, which will connect our Simulink controller to a STK environment, similar to what we had planned with MATLAB. Similar to our approach outlined above, Simulink will do all of the spacecraft control, and STK will merely model and return whatever sensor data we request.

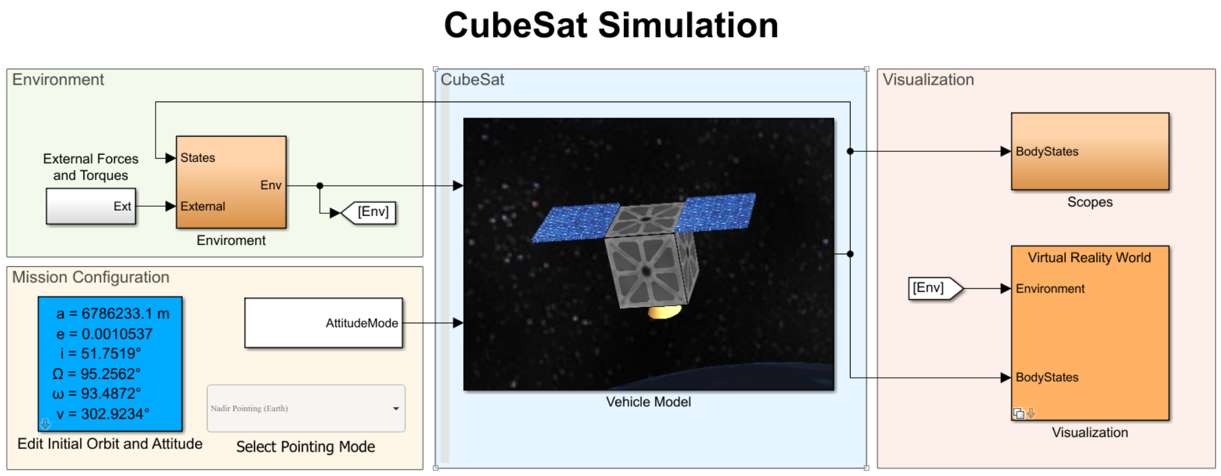
4.5 System Power Requirements

5 Results

References

[1] J. Foley, “The cubesat program.” California Polytechnic Institute, San Luis Obispo, 1999.

[2] “NASA’s insight mars lander,” *NASA*. NASA, Jun-2019.



a = 6786233.1 m
e = 0.0010537
i = 51.7519°
Ω = 95.2562°
ω = 93.4872°
v = 302.9234°

Nadir Pointing (Earth)

Copyright 2018-2019 The MathWorks, Inc.

Figure 19: Matlab CubeSat Simulation Toolbox

Appendix