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 Background

2.1 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.

2.1.1 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.

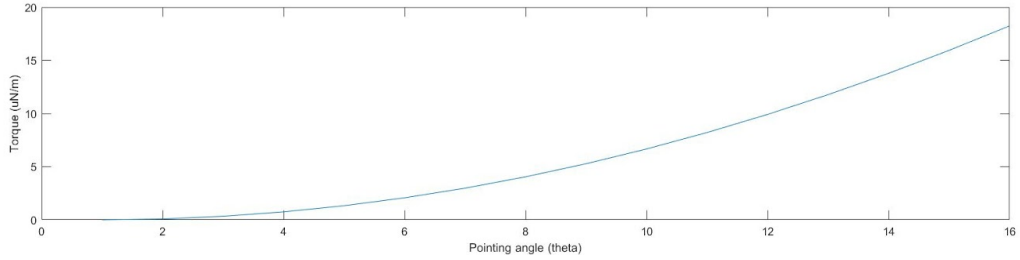

Figure 1: Pointing Angle vs. Torque

2.1.2 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 1: 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	$-20°C$ to $60°C$
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	$58 \times 58 \times 25$ 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 3: 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	$155 \times 76 \times 34$ 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 4: 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 5: 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 6: 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	$\pm 0.01\%$
Magnetic Sensitivity	$\pm 1\%$

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 7: 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	$< 0.1°$

2.1.3 Control Logic

2.1.3.1 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.

2.1.3.2 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:

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

$$Y = R cos(\epsilon) sin(\lambda)$$

$$Z = R sin(\epsilon)$$

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

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

2.1.3.3 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.

2.1.3.4 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.

2.2 Orbital Determination and Control

2.2.1 eLEO Orbit Decay

2.2.2 Component Selection

2.3 Command and Data Handling

2.3.1 Data Handling Needs

2.3.2 Component Selection

2.4 ADCS Testbed

3 Analysis

3.1 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.

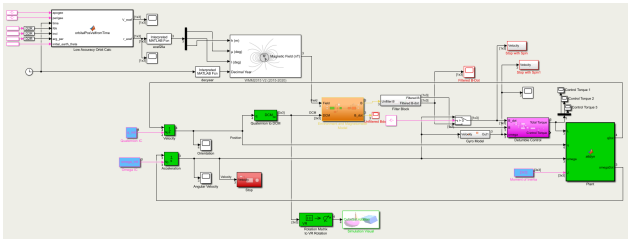

Figure 8: Total Detumbling Simulink Simulation

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.

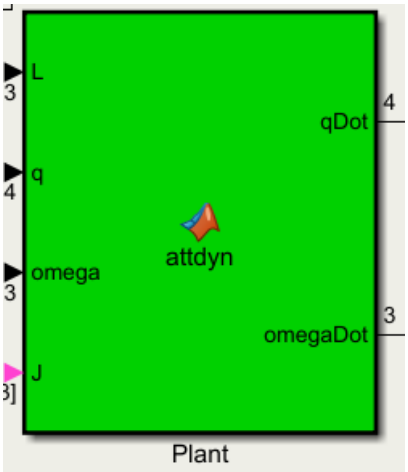

Figure 9: Attitude Dynamics Plant

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.

3.1.1 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.

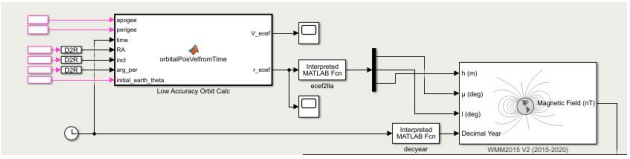

Figure 10: Simulation Path to Find Inertial Magnetic Field Vector

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, $\theta_{i_e earth}$ 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.

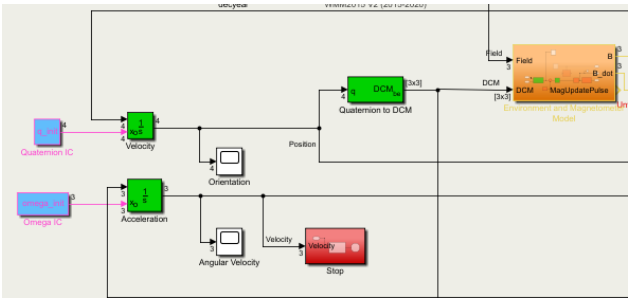

Figure 11: Integration of the q and omega vectors and Simulink q to DCM block

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.

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.

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

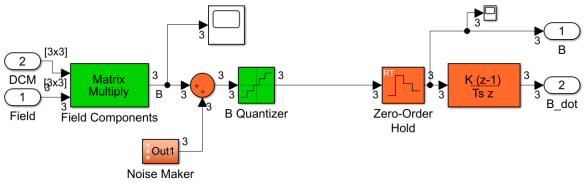

Figure 12: Contents of the Magnetometer Simulation Block

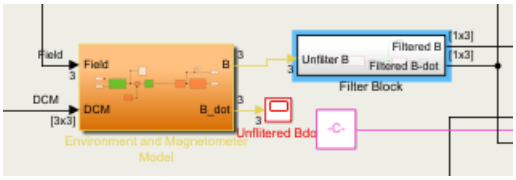

Figure 13: Magnetometer Simulation Block and Low Pass Filter

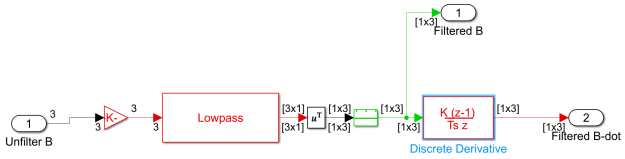

Figure 14: Contents of the Low Pass Filter Block

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 $\dot{B}$.

3.2 Reaction Wheel Sizing

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

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, $6.978x10^6m$.

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

Where C_{ps} is the center of solar pressure and C_g is the center of gravity.

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

Where l is the angle of incidence to the Sun, F_s, is the BLANK, $1.367\frac{W}{m^2}$ c is the speed of light, $3x10^8\frac{m}{s}$, A_s is the BLANK, 0.01 m and q is the coefficient of reflection, 0.6. We can then find the torque on the vehicle by using the equation below.

$$T_m = DB$$

Where D is equal to the dipole moment of the vehicle and B is the Earth's magnetic field.

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

Where V is the velocity of the vehicle and C_{pa} is the center of WHAT IS THIS LETTER pressure.

Torque Reaction Wheel Sizing, T_{rw}

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

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.

3.3 Simulink Control

3.3.1 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.

3.3.1.1 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.

3.3.1.2 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.

3.3.1.3 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.

3.4 Attitude Maintenance

3.5 System Power Requirements

4 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.

Appendix