공학석사학위논문

Attitude Determination and Control System for Low Earth Orbit CubeSat Considering Operation Scenario

운영 시나리오를 고려한 저궤도 큐브위성의 자세결정 및 제어에 관한 연구

2016 년 8 월

서울대학교 대학원

기계항공공학부

장 주 영

Attitude Determination and Control System for Low Earth Orbit CubeSat Considering Operation Scenario

운영 시나리오를 고려한 저궤도 큐브위성의 자세결정 및 제어에 관한 연구

지도교수 기 창 돈

이 논문을 공학석사 학위논문으로 제출함

2016 년 7 월

서울대학교 대학원

기계항공공학부

장 주 영

장주영의 공학석사 학위논문을 인준함

2016 년 6 월

위 원 장 ____________________

부위원장 ____________________

위 원 ____________________

Abstract

Attitude Determination and Control System for Low Earth Orbit CubeSat Considering Operation Scenario

JooYoung Jang
School of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

SNUGLITE (Seoul National University GNSS Laboratory satellITE), a cubesat scheduling for launch in 2017, must maintain nadir pointing attitude control in order to obtain mission data successfully and send them to ground station using S-band antenna. This thesis will examine the ADCS (Attitude Determination and Control System) algorithm and how the proposed algorithm satisfies the ADCS requirements for this particular operation scenario. The ADCS is composed of 3-axis MEMs gyroscope sensor, 2-axis coarse photodiode type sun sensors, 3-axis MEMs magnetometer, dual frequency GPS receivers, and 3-axis magnetic torquers. In the simulation environment, studies were done on the eclipse and various disturbance torques, including gravity gradient torque, solar radiation torque, and aerodynamic torque. The LQG (Linear Quadratic Gaussian) controller has been chosen for ADCS algorithm for SNUGLITE. Furthermore, this thesis will provide SILS (Simulation In the Loop System)—which takes low earth orbit environments into account—to verify the proposed ADCS algorithm. Due to the fact that the precise launch schedule of the cubesat has not yet been decided, the orbit elements have been chosen for the worst case scenario: it will provide a maximum eclipse time in altitude of 600[km] from the circular sun’s synchronous orbit. In order to monitor the low earth orbit environment which has a direct effect on the attitude dynamics, we developed a low earth orbit simulator that is comprised of attitude dynamics and orbit dynamics which can influence each other. We used two computers, one for the ADCS algorithm and one for the orbit environment, and transmitted data using serial communication. Thus, the team of scholars after us could use this SILS we developed to further the research on PILS (Processor In the Loop System).

The operation scenario consists of two parts. The first step starts from deployment of cubesat from P-POD (Poly Picosatellite Orbital Deployer), and detumbling using B-dot control. Second, the use of LQG controller for nadir pointing control. However, the second part of the operation scenario is also divided into two segments. In the first phase, only GPS is used as a payload but magnetic boom is not deployed, while in the second phase, magnetic boom is used for the earthquake mission.

After evaluating the simulation results, we have come to a conclusion that all the ADCS requirements were met. For instance, the attitude estimation errors were less than 5 [deg] in eclipse and less than 2 [deg] per day. In addition, the attitude control errors were less than 10 [deg] in eclipse and less than 5[deg] per day. Finally, the ADCS algorithm enabled the cubesat to turn over even in an up-side-down position.

In summary, this thesis developed and verified the ADCS algorithm which was based on the Matlab by using the LQG controller; moreover, it offers the space environment simulator which could be used for the PILS (Processor In the Loop System) study in the future. We expect these results will contribute to making SNUGLTIE’s mission a success.

Keywords: SILS, LQG controller, ADCS, Cube satellite, Low earth orbit environment

Student Number: 2014-22513

Table of Contents

List of Figures

List of Tables

1. Introduction

1. Motivation and Objectives

The purpose of this thesis is development and verification of ADCS algorithm for cubesat SNUGLITE which is scheduled to launch in 2017. This chapter deals with mission of SNUGLITE, necessity for ADCS, and necessity for SILS (Software In the Loop System).

Mission for SNUGLITE(Seoul National University GNSS Laboratory satelLITE)

In recent years, a natural disaster such as Nepal, China Yunnan earthquake brought about many loss of life. Meanwhile, another natural disaster such as ionospheric storms phenomenon can give huge damage in the communication and navigation field. In order to overcome these problems, SNUGLITE has a mission to measure the ionosphere and magnetic field of low earth orbit for more accurate ionospheric modeling. These data would be used for forecasting the earthquake. These mission will contribute to future research and development on those fields.

To be specific, SNUGLITE is two unit cubesat planning to launch at 2017 for 400-600 [km] height. SNUGLITE has two missions as follows. First mission measures GPS signals using dual frequency GPS receiver for calculating TEC (Total Electron Contents). In second mission, magnetic field would be measured by magnetometer to capture magnetic field fluctuation which represents pre-earthquake phenomenon. Mission of SNUGLITE is visualized in the figure below.

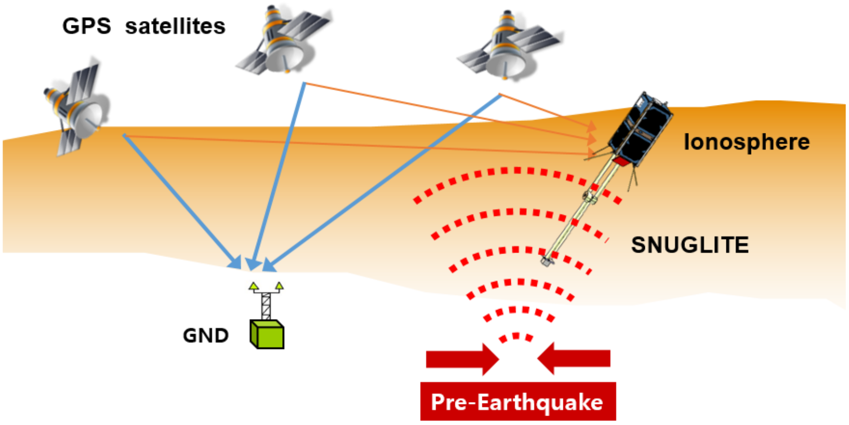


Figure 1-1. SNUGLITE Mission Concept

Necessity for ADCS(Attitude Determination and Control System)

ADCS is one of the essential subsystems of cubesat for successful mission. This subsystem plays major rule for survival from external disturbances which generates disturbance torques to the cubesat.

Necessity of ADCS could be summed up to three reasons. First reason is due to communication problem. Since, SNUGLITE use directional S-band antenna to send mission data to ground station, the bottom face which adheres S-band antenna should aim at the ground station while passing by. The second reason is due to GPS receiver. Since SNUGLITE is in 400-600[km] height, GPS receiver is adhered in upper face in order to get GPS signals from GPS. Therefore, SNGULITE should always maintain nadir pointing in order to get mission data and transmit it successfully to the ground station. Final reason is due to magnetic sensor attached in magnetic boom. Magnetic sensor should aim at the pre-earthquake regions for receiving magnetic field data. It is used to determine whether earthquake would occur in the future.

Necessity for SILS(Software In the Loop System)

It is almost impossible to realize space environment in ground to verify ADCS algorithm. Therefore, space environment simulator is used for verification. [15] The purpose of space environment simulator is to verify Matlab based ADCS algorithm, and extend it to OBC (On-Board Computer) based ADCS algorithm afterwards. The proposed algorithm should meet ADCS requirements considering operation scenario for low earth orbit. There has been several space environment simulator COS (Commercial On-the-Shelf) programs that are widely used for verification of ADCS algorithm. For example, STK (Satellite Tool Kit), and Spaecraft Control Toolbox written by PSS (Princeton Satellite System) are the most famous programs required to pay for usage, while GMAT(General Mission Analysis Tool), and other various Open Source simulators are existed for free of charge. [21] However, there are some limitations to use programs in advance. The following paragraph shows specific shortages for each programs.

First of all, STK not only offers orbit dynamics, but also attitude dynamics of cubesat. Also, there are three dimension visualization tool for cubesat attitude representation, so that many satellites were tested using STK. However, the tools for attitude visualization and attitude dynamics are expensive to use for student level.

Spacecraft Control Toolbox which is written by PSS is augmented in Matlab so that it is easy to interwork with Matlab based ADCS algorithm. In addition, this toolbox gives attitude dynamics propagator. However, there are limitation for use in orbit dynamics propagator.

GMAT is an open source program that has been provided by NASA to be available for free to the mission design of the satellite. However, GMAT couldn’t realize attitude dynamics considering the moment of inertia of the satellite. In addition, it does not visualize the attitude in three dimension.

Above all, there are open source simulators such as DTUSat -1 simulator, but all of those free programs has limitation to be extended for the OBC (On Board Computer) algorithm..

Of the above-mentioned simulators, there were no program that was available for free, extensible for the PILS (Processor In the Loop System) test of future SNUGLITE. As a result, in this paper we have developed a SILS that replicate the space environment. The following is a structural diagram of a SILS presented in this paper. [14]

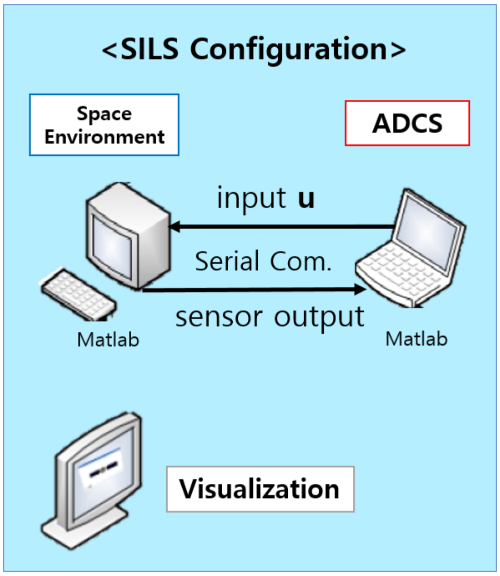


Figure 1-2. Software In the Loop System Configuration

SILS is composed of three parts; Space Environment, ADCS, and Visualization. In the case of Space Environment, attitude dynamics and kinematics of the satellite is taken into account. These nonlinear equations are used to update attitude information. This generated attitude information with the noise level of the sensor modeled as white noise, are used to generate measurement value of the sensor. In addition, external disturbance torques including the gravity gradient, magnetic field disturbance, solar radiation wind, and air friction toruqe are included in orbit propagtor to update the track information in orbit propagation.

ADCS generates control input for actuator based on the sensor readings that have been provided from the Space Environment, using an extended Kalman filter and

LQG controller. The attitude of the actual cubesat is expressed from three dimension visualization in real-time.

More detail information of theoretical content is treated in each of the following chapters.

2. Previous Work

This chapter deals with pre-study materials divided into two categories; attitude determination and attitude control.

Attitude Control

- Actuator

For cubesat, since the size is small compared to general satellite, there are limitation of loading space and power storage capacity. Therefore, low power consumption and lightweight actuator -magnetic torque- was been chosen for SNUGLITE actuator. Multiple studies implemented ADCS of cubesat using three axis magnetic torquer as primary actuators [1-3].

However, there are some drawbacks using magnetic torquer as actuator. In principle, 3-axis magnetic torquer is not possible to generate torque in the direction of the Earth's magnetic field. In addition, it generates relatively weak torque compared to the thruster or reaction wheel [1].

Nevertheless, there are more advantages than drawbacks to use magnetic torque as an actuator. If oblique inclination angle is given, full state controllability is ensured regularly within a period, making it possible for full state control. [2] Also, the three-axis magnetic torquer has better reliability since it has less dynamic movement compared to reaction wheel [1]. Moreover, it is said that three axis magnetic torquer is advantageous for LEO (Low Earth Orbit) satellite since it could provide relatively larger torque compared to the MEO (Middle Earth Orbit) or GEO (Geostationary Earth Orbit) satellites because the Earth's magnetic field is relatively large. The overall advantages guarantees that three axis magnetic torque is suited for the cubesat actuator.

- Control algorithm

In the case of the algorithm to be used for attitude control of the cubesat, PD (Proportional - Derivative) controller [3-4,8] or LQ (Linear Quadratic) controller were typically used [5,7]. The following table shows actuators and control algorithms that were used for attitude control of the other cubesats [6-8,13].

Table 1-1. Algorithm & Actuator for Cubesat Attitude Control

	Actuator	Algorithm	Purpose
Delfi-n3Xt	3-axis magnetic torquer 3-axis reaction wheel	PD	Sun pointing
SwissCube	3-axis magnetic torquer 1-axis reaction wheel	PD	Nadir pointing
AAUSAT-1	3-axis magnetic torquer	Constant LQR	Nadir pointing
AAUSAT-3	3-axis magnetic torquer Permanent Magnet	Model Predictive Control (MPC)	Nadir pointing
Compass-1	3-axis magnetic torquer	Constant LQR	Nadir pointing
SNUGLITE	3-axis magnetic torquer	LQG	Nadir pointing

In addition to the above the presented cubesats, there are a number of cube satellites, but were excluded since there were no big difference with the examples presented above.

SNUGLITE uses three -axis magnetic torquer as the actuator, and LQG controllers for control algorithm in order to satisfy ADCS requriements.

Attitude Determination

- Sensors

In a space environment, it is impossible to know the attitude of the actual cubesat. Therefore, in order to perform the attitude control of the cubesat, estimated states are used for attitude control using state feedback law. Therefore attitude determination accuracy is very important.

Sensors that are used to determine the attitude are divided into reference sensors, and rate sensors. Reference sensors include sun sensor, geomagnetic sensor, and star tracking sensor, e.t.c. Each sensors estimates satellite’s attitude using each models of sun, the bright star, and geomagnetic field and compare them with the measured value of the cube satellite. On the other hand, rate sensos, such as gyro sensors, measures angular velocity of the cubesat on the inertial coordinate system and get attitude information by integrating the measured data. [5]

Star tracking sensor is characterized with high performance, but expensive since it basically uses a large size optical system. On the other hand, gyro sensor and magnetic sensor have been miniaturized using MEMs technology. It is widely used for small satellites such as cubesats attitude determination. Moreover, for the case of the sun sensor, photodiode type coarse sun sensor is used as solar panel and attitude determination sensor. In the case of Earth sensor, it scans the Earth and gets satellite’s roll and pitch axis attitude, but yaw angle couldn’t be measured. [10]

- Attitude determination algorithms

Attitude determination algorithm could be largely divided into statistical estimation method and deterministic estimation method. For statistical estimation method, typically QUEST, REQUEST, and extended Kalman filter (EKF) algorithms are used while TRIAD is used for deterministic estimation method. [24] For more information about QUEST and REQUEST see the references [5, 9, 11]. TRIAD and EKF would be dealt in detail at Chapter 3

The following table indicates attitude determination algorithm and sensors used for other cubesats. [5, 9, 11-13]

Table 1-2. Algorithm & Sensor for Cubesat Attitude Estimation

	Sensors	Algorithms
Delfi-n3Xt	3 axis Magnetometerx 2 Coarse Sun sensor x 2 3 axis Gyroscope x 2	UKF q-method
SwissCube	3 axis Magnetometer 3 axis Gyroscope 6 Sun sensors	Optimal REQUEST method
AAUSAT-3	3 axis Magnetometer Sun sensor 3 axis Gyroscope	UKF single point SVD-method
Compass-1	3 axis Magnetometer 5 slit Sun sensor	QUEST

Since limiting loading space exists in SNUGLITE, it uses MEMs-based three axis gyro, magnetometer, and two axis (x,y) coarse sun sensor to determine the attitude. Meanwhile, since MEMs-based sensors are the low cost, the noise performance is not good as compared with the expensive sensors. Therefore, extended Kalman Filter is used to estimate the most-likelihood state using the extended Kalman filter the estimation algorithm. To summarize the attitude determination algorithm and attitude control algorithm used by SNUGLITE sum up as follows.

Table 1-3. SNUGLITE ADCS Hardware & Algorithm

	Actuator and Sensor	Algorithm
Attitude Control	3-axis magnetic torquer	LQG (Linear Quadratic Guassian)
Attitude Determination	3-axis MEMs magnetometer 2-axis coarse sun sensors 3-axis MEMs gyroscope	EKF (Extended Kalman Filter)

3. Contents and Method

In this paper, ADCS algorithm for SNUGLITE is presented. In order to verify whether the ADCS algorithm is reasonable, we verified the performance of the ADCS algorithm considering the actual operational scenario of the cubsat. To test performance, we used the SILS (Software In the Loop System) and checked the validity of the ADCS algorithm on the basis of the ADCS requirements.

In terms of content, the configuration of the paper is composed of three stages; development of ADCS algorithm, the design process of the low earth orbit simulator, and verification of ADCS algorithm. Structurally, this paper lists SILS components and describes sequentially; space environment module, ADCS module, and visualization module. In the simulation part, superiority of the presented algorithm compared to the existing studies is presented.

In Section 2, a description will be given about low-orbit environment. This chapter explains about sun-synchronous orbit, and presents Keplerian orbit elements that would be used for the orbit modeling in simulation. In addition, it deals with the disturbances existing in low-orbit environment, considering gravity gradient, solar radiation torque, air friction, and residual dipole moment. Furthermore, specification of cubesat is being introduced, and the induction process of cubesat's attitude dynamics and kinematics is being handled.

Chapter 3 begins with introducing the ADCS requirements and operational scenario of t satellite. In addition, this chapter gives detail procedure of ADCS algorithm derivation process of the cube satellite. It is classified in two categories; attitude determination and attitude control.

In Chapter 4, simulation results would be analyzed. In the first part, visualization module will be described which is developed in Matlab GUI-based. It describes the data communication protocol, and introduces the UI (User Interface). Next part deals with the results of the algorithm that is presented in the existing ADCS algorithm and compares with previous ADCS algorithm using SILS. As a result, attitude estimation error is within 5 [deg] which satisfies ADCS requirements, as well as attitude control error from less than 10 [deg] which also satisfies ADCS requirements.

4. Contribution

Attitude determination and control algorithms developed in this paper have been developed based on the existing ADCS algorithms with EKF and LQR. [1] The contribution points has been based on studies covered in previous studies.

First of all, the attitude determination and control algorithm was presented considering operating scenario on a low orbit. A simulation was performed in accordance with the actual operational scenario (angular velocity decay mode or detumbling mode - mission mode (GPS) - mission mode (magnetic field)). Taking into account the actual operating conditions, the criteria for escaping the angular velocity decay mode has been changed. Initial conditions for mission mode are given by TRIAD algorithm. Also, it was changed to LQG algorithm from LQR and EKF algorithm by augmenting dynamics in the EKF.

Secondly, in order to verify the performance of the ADCS, we have developed a space environment simulator that elaborately replicates the actual space environment. To add a solar eclipse, we used the sun model JPL DE405 (Jet Propulsion Laboratory Development Ephemeris 405) inside ADCS. For the shadow model, we used the spherical model. Also, we used coupled dynamics between orbit and attitude propagator so that position and velocity of the cubesat are influenced by attitude and angular velocity of the cubesat in real time. In order to maintain the objectivity of verification, space environment module and ADCS module has been developed in separate manner. By doing this, we expect that space environment simulator could easily be extended for the OBC (OnBoard Computer) based ADCS in the future for PILS (Processor In the Loop System) test. We have also added a disturbance torques to the space environment module. To be specific, gravity gradient, solar radiation and the air drag were applied to orbital and attitude dynamics.

Fianlly, we added a GUI-based Visualization in SILS. The attitude and orbit visualization of the satellite was developed by GUI-based matlab for ground station software in the future. This enables to see errors between the actual attitude and the estimated attitude in 3 dimension in real time. UI has been designed user-friendly so that the user could easily enter necessary parameters such as Keplerian orbit parameters, and can select disturbances in their own need.

II. Space Environment

This section introduces methods of modeling orbit and cubesat in SILS. Orbit and cubesat modeling (or space environment) is done to replicate the low-orbit space environment such as gravity gradient, solar wind, and air friction force which impacts on the orbit and attitude propagation of the satellite. True position and attitude are used to generate the sensor measurements. Space environment corresponds to the left side of the figure below, while ADCS corresponds to the right-hand side. It is treated in the next chapter.

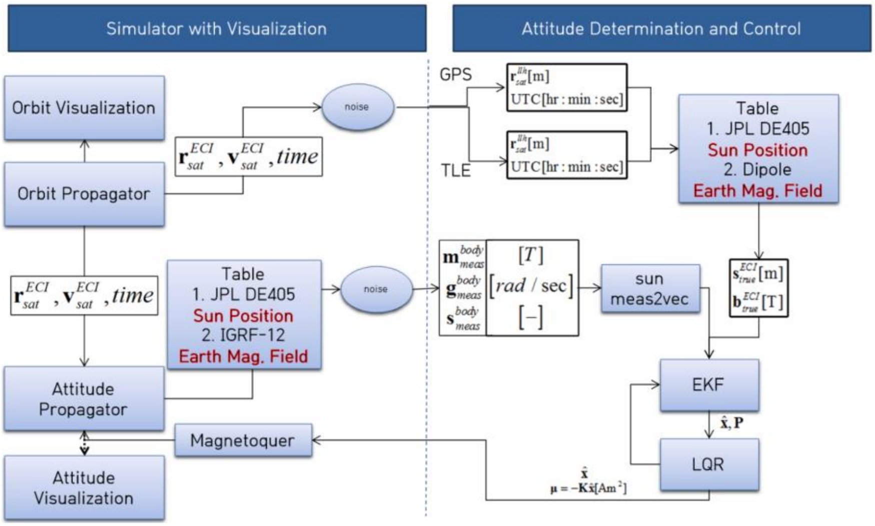

Simulator with Visualization
Attitude Determination and Control
Orbit Visualization
Orbit Propagator
noise
$\mathbf{r}_{sat}^{ECI}, \mathbf{v}_{sat}^{ECI}, time$
GPS
TLE
$\mathbf{r}_{sat}^{llh}[\mathrm{m}]$
UTC[hr : min : sec]
$\mathbf{r}_{sat}^{llh}[\mathrm{m}]$
UTC[hr : min : sec]
Table
1. JPL DE405
Sun Position
2. Dipole
Earth Mag. Field
$\mathbf{r}_{sat}^{ECI}, \mathbf{v}_{sat}^{ECI}, time$
Table
1. JPL DE405
Sun Position
2. IGRF-12
Earth Mag. Field
noise
$\mathbf{m}_{meas}^{body}$ $[T]$
$\mathbf{g}_{meas}^{body}$ $[rad/\sec]$
$\mathbf{s}_{meas}^{body}$ $[-]$
sun
meas2vec
$\mathbf{s}_{true}^{ECI}[\mathrm{m}]$
$\mathbf{b}_{true}^{ECI}[\mathrm{T}]$
Attitude
Propagator
EKF
$\hat{\mathbf{x}}, \mathbf{P}$
Magnetoquer
LQR
$\hat{\mathbf{x}}$
$\boldsymbol{\mu} = -\mathbf{K}\hat{\mathbf{x}}[\mathrm{Am}^2]$
Attitude
Visualization

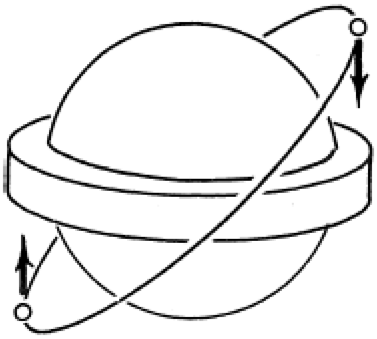
Figure 2-1. Software In the Loop System Block Diagram

As seen above, Space Environment simulator is composed of orbit propagator, and attitude propagator. Each would be dealt in the next section.

1. Modeling of Orbit

Sun synchronous orbit

In situations where the projectile of SNUGLITE is not selected, we considered the LEO cubesat assuming circular sun-synchronous orbit with altitude 600 [km]. Sun-synchronous orbit refers to the orbital plane maintaining a constant angle to the sun. [16] The orbital plane is moving due to the J2 effect caused by Earth's oblateness. [17] It could be expressed as formula as follows.



$$\Omega = \Omega_0 + \dot{\Omega}(t - t_0)$$

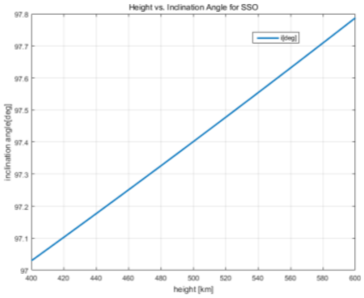
$$\dot{\Omega} = -\frac{3}{2} J_2 \left(\frac{R_\oplus}{p}\right)^2 n \cos i \qquad (2.1)$$

$$n : \text{mean motion} \left(= \sqrt{\frac{\mu_\oplus}{a^2}} \right)$$

$$p : \text{semi-latus rectum} \left(= a\left(1 - e^2\right) \right)$$

Figure 2-2. Perturbation of LEO due to Oblateness of Earth

If the orbit of the Earth is close to circle (e = 0), orbital inclination angle and apogee could be dependent in each other since the earth is rotating 360 [deg] in one year. [17] The following are Kepler elements used in this paper.



$$\begin{cases} a = 6978.137[km] \\ e = 0 \\ i = 97.7874^o \\ \Omega = 18.4053^o \\ \omega = 0^0 \\ \nu = 0^o \end{cases}$$

Figure 2-3. Sun Synchronous Orbit height vs. inclination angle

Definition of coordinate frame

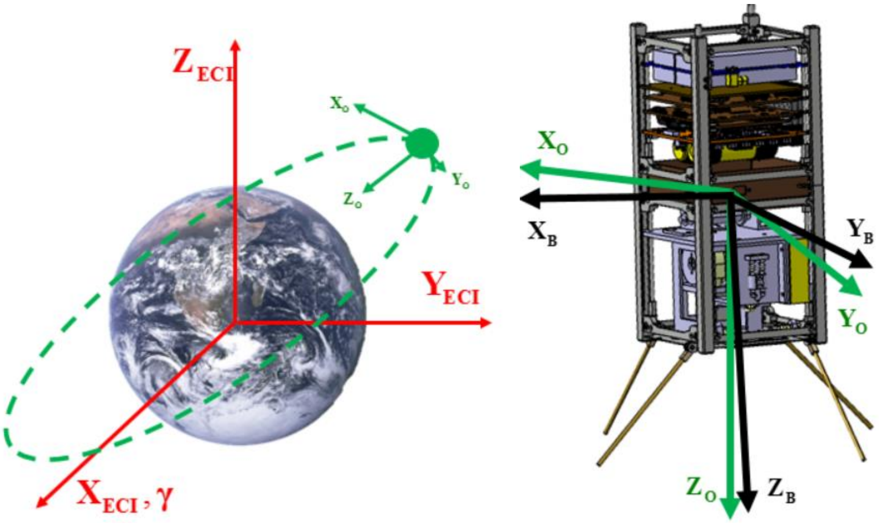


Figure 2-4. Definition of Coordinate frame

1) Earth Cenetered Inertial (ECI) frame

Earth fixed inertial coordinate system is defined about the earth. X-axis referes to Vernal equinox, and the z-axis referes to rotation of the earth axis, and y axis ax outer product between z and x axes. It is displayed in red color in [figure 5].

2) Earth Centered Earth Fixed (ECEF) frame

Earth centered earth fixed frame has its x-axis as the direction in which there is a British Greenwich Meridian, while z-axis is the Earth rotation axis. GPS receiver gives position and velocity of the in ECEF frame. In order to rotate from ECI frame, precession, nutation, rotation, and wobble of the earth's roation axis should be considered.

3) Orbital frame

Orbital coordinate system is defined based on the center of gravity of the. It’s along track is defined as x-axis, and the earth-oriented direction (cross-track) as z-axis. In the [figure 5] above, it is displayed in green.

4) Body frame

Body coordinate system is based on the center of gravity of the satellite. It’s axes are defined in the direction perpendicular to each faces of the satellite. In the figure above it is displayed in black.

5) NED (North, East, Down) frame

NED coordinate system is based on the center of gracity of the satellite. It’s x-axis is defined in the direction toward north pole, while z-axis is toward vertical down direction. NED frame is used for IGRF-12 model output.

6) LLH (Latitude, Longitude, and Height) frame

Geodetic longitude is defined as an angle between British Greenwich Meridian and position of the satellite parallel to the equator. Eastward direction is defined positive, while west as negative. Geodetic latitude is defined as an angle between the equatorial plane and the straight line that passes through the satellite and the center of the earth.

Disturbance torques

1) Gravity gradient torque

Gravity gradient torque is generated if the mass of the satellite is asymmetrically distributed. Then gravitational force induces a torque to the satellite. [5] In this chapter, gravity gradient torque is derived from the law of universal gravitation between the satellite and the earth. In other words, the third body such as the sun and the moon, shall be out of the question.

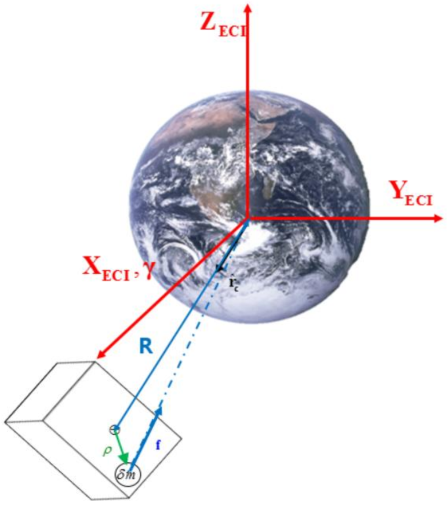


Figure 2-5. Concept of Gravity Gradient Torque

Assuming that the satellite has rigid body with equally distributed mass along the body, we could derive the gravitational force as follows:

$$\ddot{\mathbf{r}} = -\frac{GM_\oplus}{r^3}\mathbf{r} = -\frac{GM_\oplus}{(\rho+R)^3}(\boldsymbol{\rho}+\mathbf{R})$$

$$\mathbf{M} = \int_f \rho df \qquad (2.2)$$

This two body equation could be used to induce gravity gradient torque using 1st order Taylor series expansion as follows: [18]

$$\mathbf{M_G} = 3n^2\hat{r}_c^b \times \left(\mathbf{I}\hat{r}_c^b\right) \qquad (2.3)$$

$$\begin{cases} n : \text{mean motion}\left(\text{rad/sec}\right) \\ \hat{r}_c : \text{line of sight vector} \end{cases}$$

Gravity gradient torque generates a torque so that the axis with the minimum moment of inertia is aligned with the direction of gravity direction. Therefore, as defined in [figure 5], gravity gradient torque helps for attitude stabilization for nadir point control where along track $\mathbf{x_b}$ has maximum moment of inertia, and cross track $\mathbf{z_b}$ has minimum moment of inertia. [18]

Factors that affect the maximum torque that can be gravity gradient torque gives is a trajectory and moment of inertia of the satellite. The following sets forth the formula

In order to calucate maximum torque that can be given by gravity gradient torque, [25] gives the formula as follows. Moment of inertia, orbit radius, and maximum departure angle from nadir point vector is factors that affects maximum torque of gravity.

$$\mathbf{T_g} = \frac{3\mu_\oplus}{2R^3} \left| \mathrm{I_{max}} - \mathrm{I_{min}} \right| \sin\left(2\theta\right) \qquad (2.4)$$

$$\begin{cases} \mu_\oplus = 3.986 \times 10^{14} m^3 / s^2 \\ \theta : \text{maximum departure angle from nadir pointing} \\ R : \text{orbit radius} \end{cases}$$

2) Solar radiation pressure torque

Solar radiation pressure torque is cuased by materials emitted by sun such as X-rays, neutral particles of electromagnetic wavelengths. It's strength of disturbance torque toward satellite is in proportion the emitted electric density of radiation. [18] The solar energy density is proportional to the inverse square of the distance. If the distance is 1AU (Astronomical Unit), the average solar energy density is about $1367\left[\mathrm{W/m^2}\right]$. [25]

Factors affecting solar radiation wind torque are the arm vector between the center of gravity and center of pressure of the satellite, outer reflectance coefficient, and the area of the satellite. The following formula showssolar radiation torque. [25]

$$\mathbf{M}_{solar} = \mathbf{r}_{mp} \times \mathbf{F}_{solar} \qquad \mathbf{F}_{solar} = \frac{F_s}{c} A\left(1+q\right)\cos\beta \cdot \mathbf{s}^{body} \qquad (2.5)$$

$\mathbf{r}_{mp}$: CoM to CoP vector $[m]$

F_s : solar constant (energy flux coming from the Sun)

c : speed of light

A : cross section area of sat $\left[m^2\right]$

q : reflectance factor $(0 \sim 1)$

β : incidence angle b/w sun vector and sat surface normal vector

$\mathbf{s}^{body}$: Sun vector pointing from sun to sat in body frame

3) Aerodynamic torque

When the satellite has relative speed difference between the air, aerodynamic torque is generated by the air in the opposite direct on the area of the satellite. The torque due to air friction has critical impact in attitude dynamics for a low-orbit satellites within 600 [km] height. [25] The main elements of the air friction torque are geometric structure of the satellite, arm distance between the center of gravity and the center of pressure of the satellite. [25] The following sets forth the formula.

$$\mathbf{M}_{aero} = -\frac{1}{2}\rho V_{rel}^{2} C_d A \cos\alpha \left(\mathbf{r}_{mp} \times \mathbf{u}_{rel}^{body}\right) \qquad (2.6)$$

ρ: air density $\left[kg/m^3\right]$

V_{rel}: relative sat velocity w.r.t atmospheric wind $\left[m/s\right]$

C_d: air drag coefficient

A: cross section area of sat $\left[m^2\right]$

α: incidence angle b/w sat vel vector and sat surface normal vector

$\mathbf{u}_{rel}^{body}$: relative sat velocity unit vector w.r.t atmospheric wind

$\mathbf{r}_{mp}$: CoM(center of mass) to CoP(center of pressure) vector $\left[m\right]$

4) Residual magnetic dipole

The residual magnetic dipole refers to a torque due to the internal satelite residual magnetic dipole caused by closed loop magnetic materials. This magnetic dipole and the earth's magnetic field makes residual dipole.[19] It has more impact on small satellites than large one since the moment of inertia is small in small satellites. The major elements for the residual magnetic dipole include height of the orbit, the residual dipole, and angle of inclination,. The following sets forth the formula. [25]

$$\mathbf{T_m} = DB \qquad (2.7)$$

$$\begin{cases} D: \text{residual dipole } \left[Am^2\right] \\ B \approx \dfrac{2M}{R^3}\left[T\right]: \text{Earth's magnetic field in polar orbit} \\ M = 7.96\times 10^{15}\left[Tm^3\right]: \text{magnetic moment of the earth} \end{cases}$$

The following plot compares the normal size of each disturbance torques used for simulation.

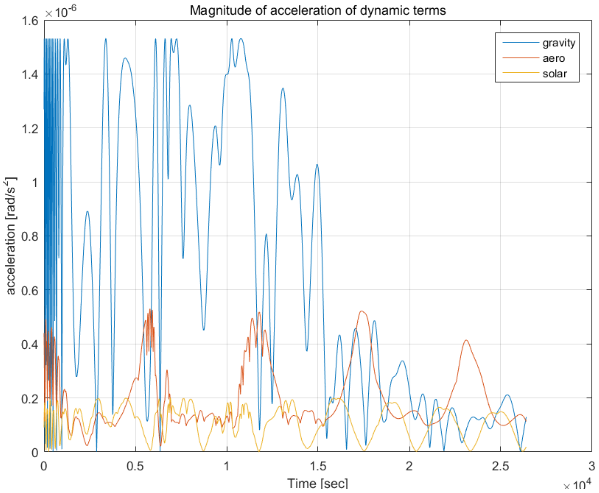


Figure 2-6. Comparing Values of Disturbance Torques

Earth Magnetic/Sun Model

1) Sun Model

Sun model provides a vector from the earth to the sun on the inertial coordinate system at any time. As the position of the satellite that orbits the earth changes, the distance vector of the sun from the satellite is changed. However, since the distance to the sun is far far away compared to the orbital radius of the low-orbit satellite, the vector from the satellite to the sun can be assumed to have a constant vector for orbital period. [5]

$$\hat{\mathbf{s}}_{ECI} \approx \mathbf{s}_{ECI}^{satellite \to sun} \qquad (2.8)$$

Sun model used in this paper is a JPL DE (Jet Propulsion Laboratory Development Ephemeris) 405 model. The sun is assumed to be fixed on the inertial coordinate system, and it assumes that the angle of the argument of perigee (ω) is constant as 282.94^o of virtual orbit where the center is defined as the Earth, and the angle of right ascension as 0^o. In addition, time starts from Gregorian date January 1, 2000, 12: 00: 00 p.m. or the 2451545.0 Julian date.

$$t_{JD,2000} = t_{JD} - 2451545 \qquad (2.9)$$

Mean anomaly of the sun is defined as follows:

$$M_{\odot} = 357.527723^o + 0.9856474^o t_{JD,2000} \qquad (2.10)$$

The first part of the equation (2.10) represents mean anomaly at the start time of Julian date, while second part refers to mean anomaly after time passes.($0.9856474^o \times 365.25 = 360^o$)

Mean longitude of the sun is as follows:

$$\lambda_{mean} = \omega + M_{\odot} \qquad (2.11)$$

Ecliptic longitude projected into the celestial sphere is as follows:

$$\lambda_{ecliptic} = \lambda_{\odot} + 1.914666471^o \sin\left(M_{\odot}\right) + 0.02^o \sin\left(2M_{\odot}\right) \qquad (2.12)$$

Linearlized obliquity of the ecliptic is as follows:

$$\varepsilon = 23.439291^o - 3.5603559 \times 10^{-7} t_{JD,2000} \qquad (2.14)$$

Therefore, sun position is finally given as follows:

$$\begin{aligned} X_{ECI} &= \cos\left(\lambda_{ecliptic}\right) \\ Y_{ECI} &= \sin\left(\lambda_{ecliptic}\right)\cos\left(\varepsilon\right) \\ Z_{ECI} &= \sin\left(\lambda_{ecliptic}\right)\sin\left(\varepsilon\right) \qquad (2.15) \end{aligned}$$

2) Geomagnetic Field Model

Earth's magnetic field is mainly generated by Earth’s outer core or magnetic field interference due to self-density variation of the ionosphere, solar radiation variation, etc. [5] In this paper, we use the IGRF (International Geomagnetic Reference Field) -12 model provided by IAGA (InternationalAssociation of Geomagnetism and Aeronomy).

Magnetic field vector of IGRF-12 model which is effective until 2020, has values based on the position of the satellite defined in spherical coordinate system. In accordance with the time, it is represented by a gradient of the negative of earth magnetic scalar potential. The expression is as follows.[20]

$$\mathbf{B} = -\nabla V \qquad (2.16)$$

Earth magnetic scalar potential V has its value as follows. [20]

$$V\left(r,\theta,\phi,t\right) = a\sum_{n=1}^{N}\sum_{m=0}^{n}\left(\frac{a}{r}\right)^{n+1} \times\left[g_n^m\left(t\right)\cos\left(m\phi\right) + h_n^m\left(t\right)\sin\left(m\phi\right)P_n^m\left(\cos\theta\right)\right] \qquad (2.17)$$

$$\begin{cases} a = 6371.2\left[km\right]: \text{geomagnetic conventional Earth's mean reference spherical radius} \\ r: \text{radial distance from the center of the earth} \\ \theta,\phi: \text{geocentric co-latitude, east longitude} \\ P_n^m\left(\cos\theta\right): \text{Schmidt quasi-normalized associated Legendre functions of degree n, order m} \\ g_n^m, h_n^m: \text{Gauss coefficients functions of time, degree n, and order m } \left[nT\right] \end{cases}$$

The coefficients used in equation (2.17) g_n^m, h_n^m is given from A.D. 1900 to 2015 in 5-year-period, while it’s rate of change is gived in assumption as 1st order for 2015-2020 as follows.

$$\begin{aligned} g_n^m\left(t\right) &= g_n^m\left(t_0\right) + \dot{g}_n^m\left(t-t_0\right) \\ h_n^m\left(t\right) &= h_n^m\left(t_0\right) + \dot{h}_n^m\left(t-t_0\right) \qquad (2.18) \end{aligned}$$

The mean value for SV (Secular Variation) is estimated and used for the coefficient of $\dot{g}_n^m\left(t\right), \dot{h}_n^m\left(t\right)$. Using equation (2.17) or (2.18), we could derive geocentric northward, eastward, and inward magnetic vectors. [20]

$$X = \frac{1}{r}\frac{\partial V}{\partial\theta}, Y = \frac{1}{r}\frac{\partial V}{\partial\phi}, Z = \frac{\partial V}{\partial r} \qquad (2.19)$$

Using equation (2.19), we could derive declination angle (D), inclination angle (I), intensity (H), and total intensity (F) as follows.

$$H = \sqrt{X^2+Y^2}, F = \sqrt{X^2+Y^2+Z^2}, D = \text{arctan}\left(Y/X\right), I = \text{arctan}\left(Z/H\right) \quad (2.20)$$

2. Modeling of Cubesat

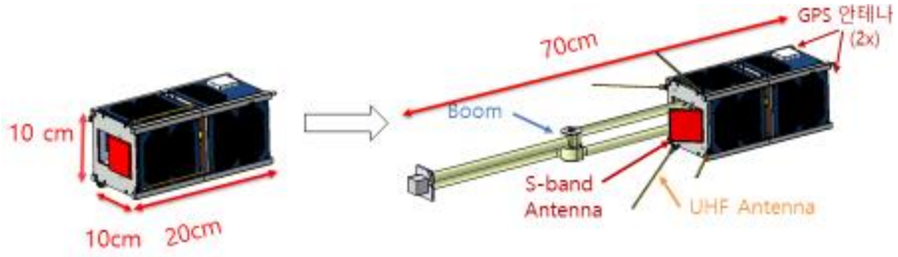


Figure 2-7. Configuration before/after Magnetic Boom Emission

SNUGLITE has two operating procedures. The first mode use GPS receiver for more precise space environment modeling. In second mode magnetometer is used after emission of magnetic boom, and collect earthquake precursor data. The following table gives specification of SNUGLITE before and after magnetic boom emission.

Table 2-1. Specification of SNUGLITE before/after Magnetic Boom Emission

	Before boom emission	After boom emission
Mass $[kg]$	2.1	2.1
Length $[mm]$ $(\text{width} \times \text{length} \times \text{height})$	$100 \times 100 \times 200$ (TBD)	$100 \times 100 \times 700$ (TBD)
Moment of Inertia $[kgm^2]\ (I_{xx}, I_{yy}, I_{zz}\ )$	$(0.0088, 0.0088, 0.0035)$	$(0.0269, 0.0269, 0.0035)$

Non-diagonal term of moment of inertia matrix is assumed to be relatively small and neglected.

SNUGLITE Specification

Sensors and actuator used for SNUGLITE is as follows.

Table 2-2. Three Axis MEMs Magnetometer

Model	HMC5843	**Interface**	I2C
Mass	50[mg]	**Power**	0.9[mA]@3.3[V]
Size	$4\times4\times1.3\left[mm\right]$	**Precision**	700[nT]
Range	$\pm400\left[\mu T\right]$	**Etc.**	OBC mounted

Table 2-3. Two Axis Photodiode Type Sun Sensor

Configuration	Photodiode	**Interface**	Analog
Mass	50[mg]	**Power**	930[uA]@3.3[V]
Precision	1.85[deg]	**Etc.**	Panel mounted

Table 2-4. Three Axis MEMs Gyroscope

Model	ADIS16251	**Interface**	SPI
Precision	$0.00458\left[{}^{o}/\sec\right]$	**Power**	15[mA]@5.0[V]
Angular Random Walk	$0.05\left[{}^{o}/\sqrt{\sec}\right]$	**Rate Random Walk**	$0.005\left[{}^{o}/\sec\sqrt{\mathrm{Hz}}\right]$
Range	$\pm20\left[{}^{o}/\sec\right]$	**Etc.**	Panel mounted

Table 2-5. Three Axis Magnetic Torquer

Company	GomSpace	**Area**	$1.55\left[m^2\right]$
Power	0.28[W]@3.3[V]	**Dipole Moment (Max)**	$0.043\left[Am^2\right]$

Attitude dynamics

5) Nonlinear Dynamics

In order to express the attitude of the cubsat, this chapter induces a nonlinear equation of attitude dynamics. When the satellite is assumed as rigid body having a uniform mass distribution, dynamics of satellite could be derived by Euler's2nd equation as the following.

$$\begin{aligned} &I_{xx}\dot{p} - I_{xy}\dot{q} - I_{xz}\dot{r} + I_{xy}pr + (I_{zz} - I_{yy})qr + (r^2 - q^2)I_{yz} - I_{xz}pq = M_x \\ &I_{yy}\dot{q} - I_{yx}\dot{p} - I_{yz}\dot{r} + pr(I_{xx} - I_{zz}) + (p^2 - r^2)I_{xz} + pqI_{yz} - qrI_{xy} = M_y \\ &I_{zz}\dot{r} - I_{xz}\dot{p} - I_{zy}\dot{q} + pq(I_{yy} - I_{xx}) + (q^2 - p^2)I_{xy} - prI_{zy} + qrI_{xz} = M_z \qquad (2.21) \end{aligned}$$

M_x, M_y, M_z in equation (2.21) represents the moments including control input $\mathbf{M}_c$, disturbance torques due to space environment such as gravity gradient torque $\mathbf{M}_G$, solar radiation pressure $\mathbf{M}_{SRP}$, and aerodynamic torque $\mathbf{M}_{AD}$, while *p*, *q*, and *r* represents the angular velocity of the inertial coordinate system of the body coordinate system. Expressing equation (2.21) in matrix form is as follows.

$$\begin{aligned} \bar{\bar{\mathbf{I}}}\dot{\boldsymbol{\omega}} &= -\boldsymbol{\omega} \times \left(\bar{\bar{\mathbf{I}}}\boldsymbol{\omega}\right) + \mathbf{M} \\ \mathbf{M} &= \mathbf{M}_G + \mathbf{M}_{SRP} + \mathbf{M}_{AD} + \mathbf{M}_c \quad (2.22) \end{aligned}$$

$\bar{\bar{\mathbf{I}}}$ represents the moment of inertia matrix, assuming the center of gravity coincides with the center of gravity before the boom is emitted. After the emission, the center of gravity moved about 2.2 [cm] downward in $\mathbf{z_b}$ axis. [Table 2-1] gives specific information of moment of inertia.

$$\bar{\bar{\mathbf{I}}} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$$

6) Nonlinear Kinematics

The way to express the attitude of the cube satellite, Euler angle is commonly used. However it is impossible to calculate the time rate of change of the

Euler angles when pitch angle is 90 ° due to singularity problem. [22] Thus, we used quaternions in order to solve this problem. At this time, since quaternions are three degrees of freedom, its square sum is always 1. ($\mathbf{q}^T\mathbf{q} = 1$) Each quaternion elements is defined as the principle angle (ϕ) and principle axes $\mathbf{l} = (l_1, l_2, l_3)$ as follows.

$$\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$$

$$q_0 = \cos\left(\frac{\phi}{2}\right),\ q_1 = l_1 \sin\left(\frac{\phi}{2}\right),\ q_2 = l_2 \sin\left(\frac{\phi}{2}\right),\ q_3 = l_3 \sin\left(\frac{\phi}{2}\right)$$

Quaternion of the satellite is defined by the attitude between the orbital coordinate system and the body coordinate system. Therefore, the angular velocity that is used for attitude propagation is as follows.

$$\boldsymbol{\omega}_{OB}^{B} = \boldsymbol{\omega}_{EB}^{B} - \mathbf{R}_{O}^{B}\boldsymbol{\omega}_{EO}^{O} = \boldsymbol{\omega} - \boldsymbol{\omega}_{EO}^{B}$$

$\boldsymbol{\omega}_{EO}^{O}$ implies the mean angular speed of the orbital coordinate system of the satellite. It has the average revolution speed (*n*) in the -y-axis direction defined in orbital coordinate system. For the altitude 600 [km], its size is approximately 0.0621[deg/s]. ($\boldsymbol{\omega}_{EO}^{O} = \left[0, -0.0621, 0\right]^T$) $\mathbf{R}_{O}^{B}$ refers to a Direct Cosine Matrix that will rotate from the body coordinate system to the orbital coordinate system. Generally, it is defined as a transformation of 3-2-1 $\mathbf{R}_{O}^{B} = \mathbf{R}(1,\phi)\mathbf{R}(2,\theta)\mathbf{R}(3,\psi)$, and expressed in quaternion as follows. [1]

$$\mathbf{R}_{O}^{B} = \begin{bmatrix} {q_1}^2 + {q_2}^2 - {q_3}^2 - {q_4}^2 & 2(q_2q_3 + q_1q_4) & 2(q_2q_4 - q_1q_3) \\ 2(q_2q_3 - q_1q_4) & {q_1}^2 - {q_2}^2 + {q_3}^2 - {q_4}^2 & 2(q_3q_4 + q_1q_2) \\ 2(q_2q_4 + q_1q_3) & 2(q_3q_4 - q_1q_2) & {q_1}^2 - {q_2}^2 - {q_3}^2 + {q_4}^2 \end{bmatrix}$$

The following equations show quaternian propagation. [27]

$$\dot{\mathbf{q}} = \frac{1}{2}\tilde{\boldsymbol{\omega}}(t) \otimes \mathbf{q}(t) = \frac{1}{2}\boldsymbol{\Omega}\left(\boldsymbol{\omega}_{EO}^{B}\right)\mathbf{q} \qquad (2.23)$$

$$\tilde{\boldsymbol{\omega}} = \begin{bmatrix} \boldsymbol{\omega}_{EO}^{B} \\ 0 \end{bmatrix}, \boldsymbol{\Omega}\left(\boldsymbol{\omega}_{OB}^{B}\right) = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix},$$

$\boldsymbol{\Omega}$ is skew-symmetric matrix, and $\otimes$ means quaternian multiplication operation. [1]

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Xi}(\mathbf{q})\boldsymbol{\omega}_{OB}^{B}, \ \boldsymbol{\Xi}(\mathbf{q}) = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \qquad (2.24)$$

III. ADCS Algorithm

1. ADCS (Attitude Determination and Control System) requirement

SNUGLITE must meet the following four ADCS requirements in order to get GPS signal which is the main task, and transmit to the ground station using S-band antenna which has directivity.

Table 3-1. ADCS Requirements

	ADCS Requirements
ADC.1	The satellite should stabilize its angular velocity after emission of P-POD less than mean motion within two days.
ADC.2	ADCS should have attitude determination error less than **5 [deg].**
ADC.3	ADCS should have attitude control error less than **5 [deg].**
ADC.4	ADCS should keep its downward face in nadir direction, while upper face for zenith direction.

The angular velocity is relatively fast just after discharged from P-POD. Therefore it is difficult to track GPS signals from visible satellites, which makes it impossible to apply the attitude determination algorithm. In other words, when an angular velocity is not attenuated quickly, the time for mission performance period would decrease, reducing the probability for successful mission. Therefore, it is necessary to perform angular speed damping control, which is ADCS first requirement.

It is impossible to know the actual attitude of cubesat in the space. Therefore, it is necessary to perform the attitude control based on the estimated attitude. This implies that attitude control performance becomes better as the attitude determination accuracy gets better. Therefore, ADCS must determine the attitude of less than 5 degrees with respect to the true attitude.

It is necessary to control the attitude of cubesat within 10 degrees relative to the earth direction vector for each three axis. This would be the third requirement.

When exiting the angular velocity decay mode (or detumbling mode), cubesat will have arbitrary orientation since detumbling mode only decreases angular velocity, not angle. Therefore, although it may be the case that reversal occurs. For up-side-down case, gravity gradient torque generates a torque that prevents the restoration. Therefore, even when turned over, the bottom surface of the satellite should able to aim direction toward the earth, while the upper surface of the satellite must be capable of pointing zenith direction. This is the last ACDC requirement.

2. Operation Scenario

There are generally two methods to enter the orbit for cubesat; injecting into the space in using the arm of the robot at the International Space Station, and projecting the cubesat through the P-POD. Because where there are people, the International Space Station has more constraints than using P-POD considering safety such as battery discharing requirement. On the other hand, P-POD method has less constraints compared to International Space Station.

In the case of SNUGLITE, it plans to use the P-POD injection system, and the operational scenario is as follows.

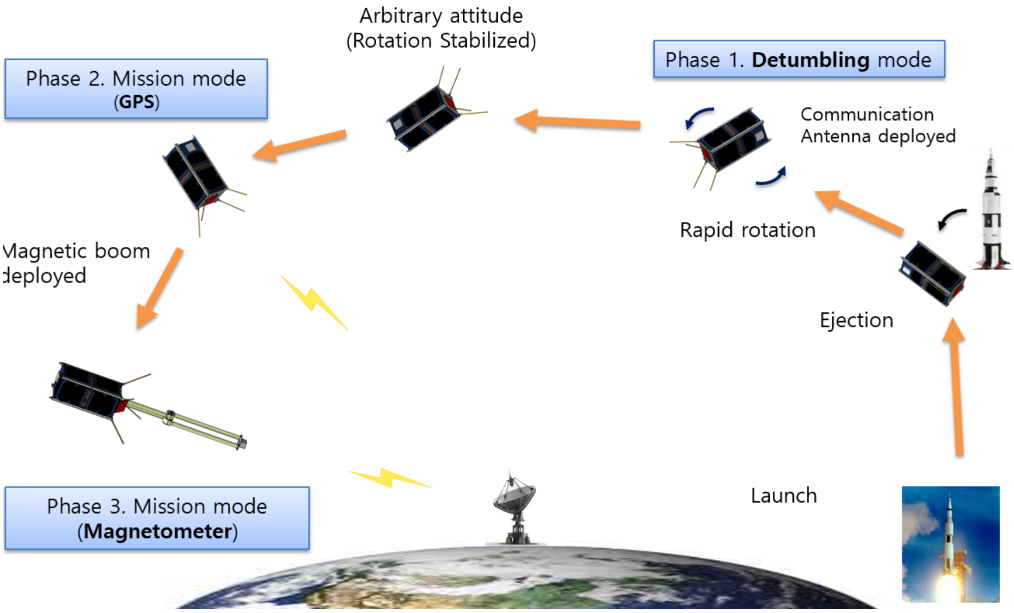


Figure 3-1. SNUGLITE Operation Scenario

Table 3-2. Description of Operation Mode

	Explanation	Characteristics
Detumbling mode	Decreasing angular velocity of the satellite	Determination : undo Control : B-dot
Mission mode (GPS)	Verification of self-developed GPS receiver & collecting measurement data	Determination: EKF Control: LQG
Mission mode (Mag)	Collect GPS & Magnetic field data for observing precursor of an earthquake	Determination: EKF Control: LQG

In the case of SNUGLITE, it plans to approach orbit in the process of the P-POD. Because it is able to charge the battery using P-POD, cubesat has full charged battery after the injection from P-POD. Just after injection mode, UHF antenna would be deployed automatically to broadcast the state information of the satellite in the omni-directional UHF transceiver. Subsequently starts angular velocity decay mode - Mission mode (GPS collection) - magnetic field boom injection mode – mission mode (magnetic field observation).

3. Attitude Estimation

TRIAD (TRI-axis Attitude Determination)

TRIAD is the oldest and simplest deterministic attitude determination method, using a vector of two measurements. [11] More specifically, it is composed of the vectors $\mathbf{r}_1, \mathbf{r}_2$ defined at the orbital coordinate system, and the vectors $\mathbf{b}_1, \mathbf{b}_2$ defined at the body coordinate system. In order to inhence the performance of attitude determination for TRIAD, the first vectors $\left(\mathbf{r}_1, \mathbf{b}_1\right)$ is defined as vectors which has more accurate measurements. The following shows the procedure of the attitude determination using TRIAD algorithm.

Table 3-3. TRIAD Algorithm

	Explanation	Formula
Step. 1	Get orbit axis vectors using GPS output and Sun&IGRF model. Get body axis measurement vectors using magnetometer & sun sensor	$\begin{cases} \mathbf{r}_1 = \mathbf{s}_{\text{model}}^{orbit} \\ \mathbf{r}_2 = \dfrac{\mathbf{s}_{\text{model}}^{orbit} \times \mathbf{b}_{\text{model}}^{orbit}}{\lvert \mathbf{s}_{\text{model}}^{orbit} \times \mathbf{b}_{\text{model}}^{orbit} \rvert} \\ \mathbf{r}_3 = \dfrac{\mathbf{r}_1 \times \mathbf{r}_2}{\lvert \mathbf{r}_1 \times \mathbf{r}_2 \rvert} \end{cases}$ $\begin{cases} \mathbf{b}_1 = \mathbf{s}_{meas}^{body} \\ \mathbf{b}_2 = \dfrac{\mathbf{s}_{meas}^{body} \times \mathbf{b}_{meas}^{body}}{\lvert \mathbf{s}_{meas}^{body} \times \mathbf{b}_{meas}^{body} \rvert} \\ \mathbf{b}_3 = \dfrac{\mathbf{r}_x \times \mathbf{r}_y}{\lvert \mathbf{r}_x \times \mathbf{r}_y \rvert} \end{cases}$
Step. 2	Augment vectors and get DCM[1]	$\mathbf{R}_x^b = \left[\mathbf{b}_1\, \mathbf{b}_2\, \mathbf{b}_3\right]$ and $\mathbf{R}_x^O = \left[\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{r}_3\right]$ $\Rightarrow \mathbf{R}_O^b = \mathbf{R}_x^b \left(\mathbf{R}_x^O\right)^T$
Step. 3	Get attitude from DCM	$\mathbf{R}_O^b =$

[1] DCM : Direct Cosine Matrix

		$\begin{bmatrix} q_1^{\ 2}+q_2^{\ 2}-q_3^{\ 2}-q_4^{\ 2} & 2(q_2q_3+q_1q_4) & 2(q_2q_4-q_1q_3) \\ 2(q_2q_3-q_1q_4) & q_1^{\ 2}-q_2^{\ 2}+q_3^{\ 2}-q_4^{\ 2} & 2(q_3q_4+q_1q_2) \\ 2(q_2q_4+q_1q_3) & 2(q_3q_4-q_1q_2) & q_1^{\ 2}-q_2^{\ 2}-q_3^{\ 2}+q_4^{\ 2} \end{bmatrix}$ $\Rightarrow \hat{\mathbf{q}}_0 = [\hat{q}_0 \quad \hat{q}_1 \quad \hat{q}_2 \quad \hat{q}_3]^T$

EKF (Extended Kalman Filter)

In order to solve the limitations of the Kalman filter that has been applied only to a linear system, Extended Kalman Filter (EKF) is designed to be used for nonlinear systems by linearizing at the nominal state. [11] State variables for EKF that were applied in this paper are composed of 10 values, adding the angular velocity vector to the previous states used in [1] and [26]; quaternion and gyro bias. [27]

$$\mathbf{x}(t_k) = \begin{bmatrix} \mathbf{q}(t_k) \\ \boldsymbol{\omega}(t_k) \\ \mathbf{b}(t_k) \end{bmatrix}, \ \mathbf{x}(t_k): \text{state vector}$$

$$\mathbf{q} = [q_0 \quad q_1 \quad q_2 \quad q_3]^T, \ \boldsymbol{\omega} = [\omega_1 \quad \omega_2 \quad \omega_3]^T, \mathbf{b} = [b_1 \quad b_2 \quad b_3]^T$$

Quaternions ($\mathbf{q}$) are defined in the body coordinate system, and it is defined by the relative attitude between the body coordinate system and orbital coordinate system. Angular velocity ($\boldsymbol{\omega}$) and the gyro bias ($\mathbf{b}$) are defined in the body coordinate system, and it means relative angular velocity and the bias of body coordinate system respect to the ECI frame, respectively.

The state equation of the nonlinear continuous system for used in Extended Kalman filter are as follows. Control input $\mathbf{u}(t_k)$, state $\mathbf{x}(t_k)$, and white noise $\mathbf{w}(t_k)$ are used.

$$\hat{\dot{\mathbf{x}}}(t_k) = \mathbf{f}\left(\hat{\mathbf{x}}(t_k), \mathbf{u}(t_k)\right) + \mathbf{w}(t_k) \qquad (3.1)$$

We modeled quarternion state equation, euler’s 2nd equation, and gyro bias equation as follows:

$$\mathbf{f}\left(\hat{\mathbf{x}}(t_k), \mathbf{u}(t_k)\right) = \begin{bmatrix} \frac{1}{2}\mathbf{\Omega}\left(\hat{\boldsymbol{\omega}}_{OB}^{B}(t_k)\right)\hat{\mathbf{q}}(t_k) \\ \bar{\bar{\mathbf{I}}}^{-1}\left(\mathbf{M}_c - \hat{\boldsymbol{\omega}} \times \left(\bar{\bar{\mathbf{I}}}\hat{\boldsymbol{\omega}}\right)\right) \\ -\frac{1}{\tau}\hat{\mathbf{b}}(t_k) \end{bmatrix}$$
$$\mathbf{M}_c = \mathbf{u} \times \mathbf{B}_{\oplus} \qquad (3.2)$$

Compared to equation (2.1) which is used for cubesat in space environment module, control input torque and inertia term was only used, while disturbance torques from space environment were neglected in equation (3.1). This assumption is relatively valid because as shown in [figure 2-6], the disturbance torques generate relatively small torque compared to inertial and control input torque. We neglected the disturbance torques in consideration of simplification of linearization process. Even though these disturbance torques (gravity gradient torque, solar wind, air friction) were excluded from the state equations, we modeled these as Gaussian white noise and augmented to process noise, which means uncertainty of EKF.

$$\mathbf{w}(t_k) = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\eta}_D(t_k) \\ \boldsymbol{\eta}_{bias}(t_k) \end{bmatrix}$$

$$\mathbf{w}(t) \sim N(0, \mathbf{Q})$$

$$\mathbf{Q}=\begin{bmatrix}\mathbf{0}_{4\times4} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3}\\ \mathbf{0}_{3\times4} & \left(\sigma_G^2+\sigma_{SRP}^2+\sigma_{AD}^2\right)\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \left(\sigma_{RRW}\right)^2\cdot\mathbf{I}_{3\times3}\end{bmatrix} \qquad (3.3)$$

σ_{RRW} : Gyroscope rate random walk noise

σ_G : Worst case gravity gradient torque process noise

σ_{SRP} : Worst case solar radiation pressure torque process noise

σ_{AD} : Worst case aerodynamic torque process noise

In equation (3.3), we modeled quarternion process noise as zero, since it doesn’t have uncertainty in the kinematics of the cubesat. By contrast, angular velocity noise was modeled as the maximum size of each disturbance torques as it was ignored in the EKF. Finally, gyro bias process noise has its value equal to the rate random walk of the gyro sensor. The process noise $\mathbf{w}(t)$ in equation (3.2) is used for time propagation of erro state and state covariance update. The state variables, error state, and error state covariance are defined as follows. [27]

$$\Delta\mathbf{x}(t)=\mathbf{x}(t)-\hat{\mathbf{x}}(t)$$
$$\mathbf{P}(t)=E\left[\Delta\mathbf{x}(t)\Delta\mathbf{x}(t)^T\right]$$

E referes to expectation, $\mathbf{x}(t)$ as true state, and $\hat{\mathbf{x}}(t)$ as estimated state.

For linearlization, we divded equation (3.1) by state variables and derived state transition matrix.

$$\mathbf{F}=\frac{\partial\mathbf{f}(\mathbf{x},\mathbf{w})}{\partial\mathbf{x}}=\begin{bmatrix}\frac{\partial\dot{\hat{\mathbf{q}}}}{\partial\hat{\mathbf{q}}} & \frac{\partial\dot{\hat{\mathbf{q}}}}{\partial\hat{\boldsymbol{\omega}}} & \frac{\partial\dot{\hat{\mathbf{q}}}}{\partial\hat{\mathbf{b}}}\\ \frac{\partial\dot{\hat{\boldsymbol{\omega}}}}{\partial\hat{\mathbf{q}}} & \frac{\partial\dot{\boldsymbol{\omega}}}{\partial\hat{\boldsymbol{\omega}}} & \frac{\partial\dot{\boldsymbol{\omega}}}{\partial\hat{\mathbf{b}}}\\ \frac{\partial\dot{\hat{\mathbf{b}}}}{\partial\hat{\mathbf{q}}} & \frac{\partial\dot{\hat{\mathbf{b}}}}{\partial\hat{\mathbf{b}}} & \frac{\partial\dot{\hat{\mathbf{b}}}}{\partial\hat{\boldsymbol{\omega}}}\end{bmatrix} \qquad (3.4)$$

$$\frac{\partial\dot{\hat{\mathbf{q}}}}{\partial\hat{\mathbf{q}}}=\frac{1}{2}\mathbf{\Omega}\left(\hat{\boldsymbol{\omega}}_{OB}^{B}\right),\ \frac{\partial\dot{\hat{\mathbf{q}}}}{\partial\hat{\boldsymbol{\omega}}}=\frac{1}{2}\Xi\left(\hat{\mathbf{q}}\right),\ \frac{\partial\dot{\hat{\mathbf{q}}}}{\partial\hat{\mathbf{b}}}=\mathbf{0}_{4\times3}$$

$$\frac{\partial\dot{\hat{\boldsymbol{\omega}}}}{\partial\hat{\mathbf{q}}}=\mathbf{0}_{3\times4},\ \frac{\partial\dot{\boldsymbol{\omega}}}{\partial\hat{\boldsymbol{\omega}}}=\begin{bmatrix}0 & a\hat{\omega}_3 & a\hat{\omega}_2\\ b\hat{\omega}_3 & 0 & b\hat{\omega}_1\\ c\hat{\omega}_2 & c\hat{\omega}_1 & 0\end{bmatrix},\frac{\partial\dot{\boldsymbol{\omega}}}{\partial\hat{\mathbf{b}}}=\mathbf{0}_{3\times3}$$

$$\frac{\partial\dot{\hat{\mathbf{b}}}}{\partial\hat{\mathbf{q}}}=\mathbf{0}_{3\times4},\ \frac{\partial\dot{\hat{\mathbf{b}}}}{\partial\hat{\boldsymbol{\omega}}}=\mathbf{0}_{3\times3},\ \frac{\partial\dot{\hat{\mathbf{b}}}}{\partial\hat{\mathbf{b}}}=-\frac{1}{\tau}\mathbf{I}_{3\times3}$$

To be specific, $\frac{\partial\dot{\hat{\mathbf{q}}}}{\partial\hat{\mathbf{q}}}$ has its value as follows. [1]

$$\frac{\partial\dot{\hat{\mathbf{q}}}}{\partial\hat{\mathbf{q}}}=\frac{1}{2}\mathbf{\Omega}\left(\hat{\boldsymbol{\omega}}-\mathbf{R}_{Orbit}^{Body}\boldsymbol{\omega}_{EO}^{O}\right)-\frac{1}{2}\mathbf{\Omega}\left(\frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial\mathbf{q}}\boldsymbol{\omega}_{EO}^{O}\right)\mathbf{q}\qquad(3.5)$$

$\frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial\mathbf{q}}$ represents the derivative of Direct Cosine Matrix (DCM) $\mathbf{R}_{Orbit}^{Body}$ relative to quarternions.

$$\frac{1}{2}\mathbf{\Omega}\left(\frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial\mathbf{q}}\boldsymbol{\omega}_{EO}^{O}\right)\mathbf{q}=\begin{bmatrix}\frac{1}{2}\mathbf{\Omega}\left(\frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial q_0}\boldsymbol{\omega}_{EO}^{O}\right)\mathbf{q} & \cdots & \frac{1}{2}\mathbf{\Omega}\left(\frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial q_3}\boldsymbol{\omega}_{EO}^{O}\right)\mathbf{q}\end{bmatrix}$$

$$\frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial q_0}=2\begin{bmatrix}q_0 & q_3 & -q_2\\ -q_3 & q_0 & q_1\\ q_2 & -q_1 & q_0\end{bmatrix}\qquad\frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial q_1}=2\begin{bmatrix}q_1 & q_2 & q_3\\ q_2 & -q_1 & q_0\\ q_3 & -q_0 & -q_1\end{bmatrix}$$

$$\frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial q_2}=2\begin{bmatrix}-q_2 & q_1 & -q_0\\ q_1 & q_2 & q_3\\ q_0 & -q_3 & -q_2\end{bmatrix}\qquad\frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial q_3}=2\begin{bmatrix}-q_3 & q_0 & q_1\\ -q_0 & -q_3 & q_2\\ q_1 & q_2 & q_3\end{bmatrix}$$

$$\boldsymbol{\omega}_{EO}^{O}=\left[0,-0.0621,0\right]^{T}\left[\text{deg/s}\right]$$

Further, in order to reduce the computation load for linearlization, we avoided using inverse transform of the moment of inertia matrix when driving the

$\frac{\partial \dot{\boldsymbol{\omega}}}{\partial \hat{\boldsymbol{\omega}}}$ equation, and neglected the non-diagonal inertia terms. *a*,*b*,and *c* refer to difference between each inertia of principal axes as follows.

$$a = \frac{I_y - I_z}{I_x}, b = \frac{I_z - I_x}{I_y}, c = \frac{I_x - I_y}{I_z}$$

Deriving equation (3.1) by control input $\mathbf{u}$ we get the following input matrix.

$$\mathbf{G} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{w})}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial \dot{\hat{\mathbf{q}}}}{\partial \mathbf{u}} & \frac{\partial \dot{\hat{\boldsymbol{\omega}}}}{\partial \mathbf{u}} & \frac{\partial \dot{\hat{\mathbf{b}}}}{\partial \mathbf{u}} \end{bmatrix}^T \qquad (3.6)$$

$$\frac{\partial \dot{\hat{\mathbf{q}}}}{\partial \mathbf{u}} = \mathbf{0}_{4\times3}, \frac{\partial \dot{\hat{\boldsymbol{\omega}}}}{\partial \mathbf{u}} = \begin{bmatrix} 0 & \frac{B_z}{I_x} & -\frac{B_y}{I_x} \\ -\frac{B_z}{I_y} & 0 & \frac{B_x}{I_y} \\ \frac{B_y}{I_z} & -\frac{B_x}{I_z} & 0 \end{bmatrix}, \frac{\partial \dot{\hat{\mathbf{b}}}}{\partial \mathbf{u}} = \mathbf{0}_{3\times3}$$

As similar to derivation of state transition matrix in equation (3.4), we ignored 유도할 non-diagonal inertia terms and derived input matrix (3.6).

Next, disturbance matrix could be derived from equation (3.1) derived by process noise $\boldsymbol{\eta}$ as follows.

$$\boldsymbol{\Gamma} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{w})}{\partial \boldsymbol{\eta}} = \begin{bmatrix} \frac{\partial \dot{\hat{\mathbf{q}}}}{\partial \boldsymbol{\eta}} & \frac{\partial \dot{\hat{\boldsymbol{\omega}}}}{\partial \boldsymbol{\eta}} & \frac{\partial \dot{\hat{\mathbf{b}}}}{\partial \boldsymbol{\eta}} \end{bmatrix}^T = \begin{bmatrix} \mathbf{0}_{4\times4} & \mathbf{0}_{4\times6} \\ \mathbf{0}_{6\times4} & \mathbf{I}_{6\times6} \end{bmatrix} \qquad (3.7)$$

Measurement vector is composed of 3-axis magnetic field vector, 2-axis sun vector, and 3-axis gyro measurement.

$$\mathbf{z}(t_k) = \mathbf{h}(\mathbf{x}(t_k)) + \mathbf{v}(t_k) \qquad (3.8)$$

For eclipse case, we could not use sun sensors. Therefore, we divided measurement vectors for eclipse case, day case. In addition, we could not use magnetic sensor with magnetic torque simultaneously, so divided into the estimation case and estimating and control case which doesn't use magnetic sensor.

Table 3-4. Measurement Vector for Eclipse & Day

	Day	Eclipse
Estimation	$\mathbf{z}(t_k)=\begin{bmatrix}\mathbf{b}_{meas}(t_k)\\ \mathbf{s}_{meas}(t_k)\\ \boldsymbol{\omega}_{meas}(t_k)\end{bmatrix}$	$\mathbf{z}(t_k)=\begin{bmatrix}\mathbf{b}_{meas}(t_k)\\ \boldsymbol{\omega}_{meas}(t_k)\end{bmatrix}$
Estimation and Control	$\mathbf{z}(t_k)=\begin{bmatrix}\mathbf{s}_{meas}(t_k)\\ \boldsymbol{\omega}_{meas}(t_k)\end{bmatrix}$	$\mathbf{z}(t_k)=\boldsymbol{\omega}_{meas}(t_k)$

The observation vector $\mathbf{h}(\mathbf{x}(t_k))$ which estimates measurement vector $\mathbf{z}(t_k)$ using estimated states is as follows.

$$\mathbf{h}(\mathbf{x}(t_k))=\begin{bmatrix}\mathbf{b}_{mag}^{Body}(t_k)\\ \mathbf{s}_{sun}^{Body}(t_k)\\ \boldsymbol{\omega}_{gyro}^{Body}(t_k)\end{bmatrix}=\begin{bmatrix}\mathbf{R}_{Orbit}^{Body}\mathbf{R}_{ECI}^{Orbit}\mathbf{R}_{NED}^{ECI}\dfrac{\mathbf{B}_{IGRF12}^{NED}}{|\mathbf{B}_{IGRF12}^{NED}|}\\ \mathbf{R}_{O}^{B}(\hat{\mathbf{q}})\mathbf{R}_{ECI}^{O}\mathbf{s}_{sun}^{ECI}\\ \hat{\boldsymbol{\omega}}+\hat{\mathbf{b}}\end{bmatrix} \qquad (3.9)$$

In equation (3.9), we used the IGRF (International Geomagnetic Reference Field) version 12 magnetic field model provided by the NGDC (National Geophysical Data Center). It gives geographical magnetic field vector in each position. In addition, we adopted the model for line of sight vector of the sun from the satellite as JPL DE405 (Jet Propulsion Laboratory Development Ephemeris405) model. For IGRF-. 12, since the magnetic field vector outputs magnetic vector in the NED (North, East, Down) frame,, we should convert it into the body coordinate system. The procedure has four steps as follows.

$$\mathbf{R}_{NED}^{Body} = \mathbf{R}_{Orbit}^{Body}\mathbf{R}_{ECI}^{Orbit}\mathbf{R}_{ENU}^{ECI}\mathbf{R}_{NED}^{ENU}$$

$$\mathbf{R}_{Orbit}^{Body} = \begin{bmatrix} q_1^2+q_2^2-q_3^2-q_4^2 & 2(q_2q_3+q_1q_4) & 2(q_2q_4-q_1q_3) \\ 2(q_2q_3-q_1q_4) & q_1^2-q_2^2+q_3^2-q_4^2 & 2(q_3q_4+q_1q_2) \\ 2(q_2q_4+q_1q_3) & 2(q_3q_4-q_1q_2) & q_1^2-q_2^2-q_3^2+q_4^2 \end{bmatrix}$$

$$\mathbf{R}_{ECI}^{Orbit} = \begin{bmatrix} \dfrac{\mathbf{v}_{ECI}}{|\mathbf{v}_{ECI}|} & -\dfrac{\mathbf{r}_{ECI}}{|\mathbf{r}_{ECI}|} & \left(\dfrac{\mathbf{r}_{ECI}}{|\mathbf{r}_{ECI}|}\right)\times\left(\dfrac{\mathbf{v}_{ECI}}{|\mathbf{v}_{ECI}|}\right) \end{bmatrix}^T$$

$$\mathbf{R}_{NED}^{ENU} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \mathbf{R}_{ENU}^{ECI} = \mathbf{R}\left(90^o-\theta_g,1\right)\mathbf{R}\left(90^o+\lambda_T,3\right) \quad (3.10)$$

$\mathbf{v}_{ECI}$: S/C velocity in ECIframe $[\mathrm{m/sec}]$

$\mathbf{r}_{ECI}$: S/C position in ECIframe $[\mathrm{m}]$

ϕ_T : Greenwich Mean Sideral Time $[\mathrm{deg}]$

λ_T : geodetic latitude $[\mathrm{deg}]$

In equation (3.8), $\mathbf{v}$ represents sensor noise, and we modeled it as Gaussian white noise. [1]

$$\mathbf{v} \sim N\left(0,\mathbf{R}\right)$$

By taking expectation for square of $\mathbf{v}$, we could get measurement noise covariance matrix as follows.

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{mag,3\times3}(t_k) & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{2\times3} & \mathbf{R}_{sun,2\times2} & \mathbf{0}_{2\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times2} & \mathbf{R}_{ARW,3\times3} \end{bmatrix} \quad (3.11)$$

$$\mathbf{R}_{mag,3\times3}\left(t_k\right)=\frac{\sigma_{mag}^2}{|\,\mathbf{B}_{IGRF12}\,|^2}\mathbf{I}_{3\times3}$$
$$\mathbf{R}_{sun,2\times2}\left(t_k\right)=\sigma_{sun}^2\mathbf{I}_{2\times2}$$
$$\mathbf{R}_{ARW,3\times3}\left(t_k\right)=\sigma_{ARW}^2\mathbf{I}_{3\times3}$$

σ_{mag} : magnetometer measurement noise$\left[T\right]$

σ_{sun} : sun sensor measurement noise$\left[\text{deg}\right]$

σ_{ARW} : gyro angular random walk $\left[\text{deg}/\sqrt{\text{sec}}\right]$

Next, observation matrix is derived using observation vector (3.9) and state variables as follows.

$$\mathbf{H}=\frac{\partial\mathbf{h}\left(\mathbf{x}\left(t_k\right)\right)}{\partial\mathbf{x}}=\left[\frac{\partial\mathbf{h}\left(\mathbf{x}\left(t_k\right)\right)}{\partial\mathbf{q}}\quad\frac{\partial\mathbf{h}\left(\mathbf{x}\left(t_k\right)\right)}{\partial\boldsymbol{\omega}}\quad\frac{\partial\mathbf{h}\left(\mathbf{x}\left(t_k\right)\right)}{\partial\mathbf{b}}\right]\qquad(3.12)$$

$$\frac{\partial\mathbf{h}\left(\mathbf{x}\left(t_k\right)\right)}{\partial\mathbf{q}}=\begin{bmatrix}\frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial q_0}\mathbf{R}_{ECI}^{Orbit}\mathbf{R}_{NED}^{ECI}\frac{\mathbf{B}_{IGRF12}}{|\,\mathbf{B}_{IGRF12}\,|} & \cdots & \frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial q_3}\mathbf{R}_{ECI}^{Orbit}\mathbf{R}_{NED}^{ECI}\frac{\mathbf{B}_{IGRF12}}{|\,\mathbf{B}_{IGRF12}\,|}\\ \frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial q_0}\mathbf{R}_{ECI}^{Orbit}\mathbf{Rs}_{sun,2\times1}^{ECI} & \cdots & \frac{\partial\mathbf{R}_{Orbit}^{Body}}{\partial q_3}\mathbf{R}_{ECI}^{Orbit}\mathbf{Rs}_{sun,2\times1}^{ECI}\\ \mathbf{0}_{3\times1} & \cdots & \mathbf{0}_{3\times1}\end{bmatrix}_{8\times4}$$

$$\frac{\partial\mathbf{h}\left(\mathbf{x}\left(t_k\right)\right)}{\partial\boldsymbol{\omega}}=\begin{bmatrix}\mathbf{0}_{3\times3}\\ \mathbf{0}_{2\times3}\\ \mathbf{I}_{3\times3}\end{bmatrix}_{8\times3},\quad\frac{\partial\mathbf{h}\left(\mathbf{x}\left(t_k\right)\right)}{\partial\mathbf{b}}=\begin{bmatrix}\mathbf{0}_{3\times3}\\ \mathbf{0}_{2\times3}\\ \mathbf{I}_{3\times3}\end{bmatrix}_{8\times3}$$

In conclusion, using matrix $\mathbf{F},\mathbf{H},\boldsymbol{\Gamma}$ which we derived previously, we used it for recursive Extended Kalman Filter algorithm for state and covariance propagation as follows. [1]

time update

$$\dot{\hat{\mathbf{x}}}^{-}\left(t_k\right)=\mathbf{f}\left(\hat{\mathbf{x}}^{+}\left(t_{k-1}\right),\mathbf{u}\left(t_{k-1}\right)\right) \tag{3.13}$$

$$\dot{\mathbf{P}}^{-}\left(t_k\right)=\mathbf{F}\left(t_{k-1}\right)\mathbf{P}^{+}\left(t_{k-1}\right)+\mathbf{P}^{+}\left(t_{k-1}\right)\mathbf{F}\left(t_{k-1}\right)^{T}+\mathbf{\Gamma}\left(t_{k-1}\right)\mathbf{Q}\left(t_{k-1}\right)\mathbf{\Gamma}\left(t_{k-1}\right)^{T}$$

meas. update

$$\hat{\mathbf{x}}^{+}\left(t_k\right)=\hat{\mathbf{x}}^{-}\left(t_k\right)+\mathbf{K}\left(t_k\right)\left[\mathbf{z}\left(t_k\right)-\mathbf{h}\left(\hat{\mathbf{x}}^{-}\left(t_k\right)\right)\right]^{-1}$$

$$\mathbf{P}^{+}\left(t_k\right)=\left(\mathbf{I}-\mathbf{K}\left(t_k\right)\mathbf{H}\left(t_k\right)\right)\mathbf{P}^{-}\left(t_k\right)\left(\mathbf{I}-\mathbf{K}\left(t_k\right)\mathbf{H}\left(t_k\right)\right)^{T}+\mathbf{K}\left(t_k\right)\mathbf{R}\left(t_k\right)\mathbf{K}\left(t_k\right)^{T}$$

4. Attitude Control

Attitude control algorithms of cubesat can be divided into three stages as seen in [Table 3-2]. The first step is the angular velocity decay mode (detumbling mode), and the second and third stage are a mission modes. Although briefly covered in the operation scenario section, it is important to stabilize the angular velocity in the initial stage since the cubesat has a fairly fast angular velocity. In order to stably carry out the mission, ADCS of cubesat has to attenuate the angular velocity.[1] Therefore, we use b-dot control for detumbling mode. When the angular velocity is converged below a certain criteria, ADCS switches its mode into the mission mode from the angular velocity decay mode, and performs the earth-oriented attitude control required for the mission. The total diagram for ADCS process is explained in [figure 3-2].

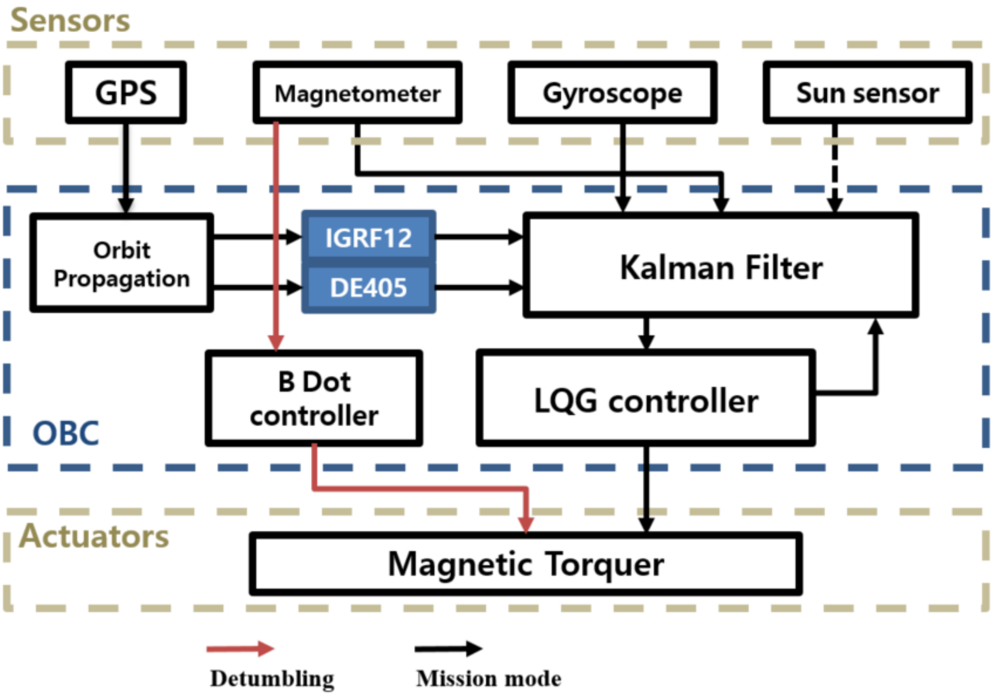


Figure 3-2. Attitude Control Algorithm

B-dot Control

In the initial angular velocity decay mode, the major factor of the vector change in sensor measurements is the angular velocity of the satellite. When using B-dot controller as equation (3.14) below, the control input to the three-axis magnetic torquer has its vector in the opposite direction of magnetic field change. With outer product of the Earth's magnetic field, torque is generated in the direction to angular velocity of the satellite. [3]

$$\mathbf{u} = -K \cdot \frac{\mathbf{B}_k - \mathbf{B}_{k-1}}{\Delta t}, K > 0 \qquad (3.14)$$

$$\mathbf{T} = \mathbf{u} \times \mathbf{B}$$

and K are a control input and gain, $\mathbf{B}$ is magnetic field measurement, and Δt is the measurement time interval. In this simulation, we used $K = 10^5$. In the angular velocity decay mode, magnetometer is used for sensor, and magnetic torquer as an actuator. Since a magnetic field is generated by a current generated in the magnetic torquer, magnetometer is not only measuring the earth's magnetic field, but also measures the magnetic field generated by the magnetic torquer. Thus, it is not possible to use a magnetometer and a magnetic torquer at the same time. In this reasons, B-dot algorithm alternately performs control and measurement (sensing) as shown in [figure. 3-3].

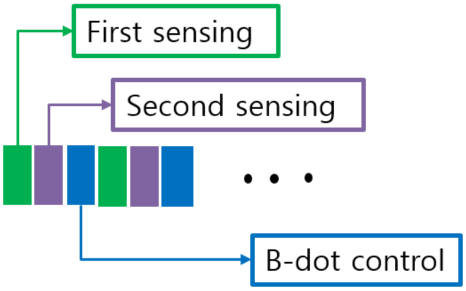


Figure 3-3. B-dot Control Algorithm

In the mission mode, attitude control is based on estimated states. Estimated states are driven using magnetic field sensor, sun sensor, and gyroscope.

LQG Control

Briefly described from the beginning, Mission mode must meet all of the requirements of the ADCS [Table 3-3] for the earth-oriented attitude control. State variables that is used for LQG control is the Euler angles and the angular velocity.

$$\mathbf{x}_c = \begin{bmatrix}\phi & p & \theta & q & \psi & r\end{bmatrix}^T \quad (3.15)$$

LQG controller generates input $\mathbf{u}$ that minimizes cost function.

$$J = \frac{1}{2}\mathbf{x}(t_k)^T \mathbf{Sx}(t_k) + \frac{1}{2}\int_{t_0}^{t_k}\left(\mathbf{x}^T\mathbf{Ax} + \mathbf{u}^T\mathbf{Bu}\right)dt \quad (3.16)$$

In the equation (3.16), $\mathbf{A,S}$ refer to the semi-positive definite matrix, and $\mathbf{B}$ is defined as the positive definite matrix. Since LQG controller can only be designed in a linear system, the kinematic equation (2.23) and dynamics equation (2.22) should be linearlized. In this case, kinematic eqaution is driven again using the expression of the euler angle and angular velocity. [1]

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}(\phi,1)\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}(\phi,1)\mathbf{R}(\theta,2)\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \mathbf{R}(\phi,1)\mathbf{R}(\theta,2)\mathbf{R}(\psi,3)\begin{bmatrix} 0 \\ -n \\ 0 \end{bmatrix} \quad (3.17)$$

ϕ,θ,ψ respectively are the Euler angles which represent the attitude of the satellite body coordinate system in respect to the orbit coordinate system; roll angle, pitch angle, and yaw angle. In equation (3.17), $\mathbf{R}(\phi,1),\mathbf{R}(\theta,2),\mathbf{R}(\psi,3)$ are each conversion matrix (transformation matrix) that converts orbital coordinate system to the body coordinate system.

$$\mathbf{R}(\phi,1)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\phi & \sin\phi\\ 0 & -\sin\phi & \cos\phi\end{bmatrix}$$

$$\mathbf{R}(\theta,2)=\begin{bmatrix}\cos\theta & 0 & -\sin\theta\\ 0 & 1 & 0\\ \sin\theta & 0 & \cos\theta\end{bmatrix}$$

$$\mathbf{R}(\psi,3)=\begin{bmatrix}\cos\psi & \sin\psi & 0\\ -\sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}$$

We derived new kinematic equations by arranging the equation (3.17) in respect to $\dot{\phi},\dot{\theta},\dot{\psi}$.

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix}1 & \tan\theta\sin\phi & \tan\theta\cos\phi\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sec\theta\sin\phi & \sec\theta\cos\phi\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} \qquad (3.18)$$
$$+\begin{bmatrix}1 & \tan\theta\sin\phi & \tan\theta\cos\phi\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sec\theta\sin\phi & \sec\theta\cos\phi\end{bmatrix}\begin{bmatrix}\sin\psi\cos\theta\\ \sin\psi\sin\theta\sin\phi+\cos\psi\cos\phi\\ \sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi\end{bmatrix}n$$

For the LQG controller, we must linearize the system. In order to linearlize the system, kinematics equation of satellite (3.18) and the dynamics equation (2.22), small angle $(\phi,\theta,\psi \ll 1)$ & small angular rate $(\dot{\phi},\dot{\theta},\dot{\psi} \ll 1)$ aprroximation, and the moment of inertia non-diagonal term were neglected. Then the equation (3.18) became as follows. [1]

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}\approx\begin{bmatrix}p+n\psi\\ q+n\\ r-n\phi\end{bmatrix} \qquad (3.19)$$

Since the size of the solar radiation pressure and aerodynamic torque are relatively small than the size of gravity gradient and control input as seen in [figure 2-6], these disturbance torques are ignored when linearlizing the dynamics of the satellite (2.22). First of all, the gravity gradient torque is as follows. [5]

$$3n^2\hat{r}_c \times (\mathbf{I}\hat{r}_c) = 3n^2 \begin{pmatrix} I_z \sin\phi\cos\phi(\cos\theta)^2 - I_y \sin\phi\cos\phi(\cos\theta)^2 \\ -I_x \sin\theta\cos\theta\cos\phi + I_z \sin\theta\cos\theta\cos\phi \\ -I_y \sin\phi\sin\theta\cos\theta + I_x \sin\phi\sin\theta\cos\theta \end{pmatrix} \qquad (3.20)$$

If small angle approximation $(\phi,\theta,\psi \ll 1)$ and diagonal moment of inertia $(I_{xy} = I_{yz} = I_{zx} = 0)$ is only considered, linearlize gravity gradient is drived as follows. [1]

$$3n^2\hat{r}_c \times (\mathbf{I}\hat{r}_c) \approx 3n^2 \begin{bmatrix} (I_z - I_y)\phi \\ (I_z - I_x)\theta \\ 0 \end{bmatrix} \qquad (3.21)$$

$\hat{r}_c$ is a unit vector from the center of the Earth to the center of the body. It is represented in the equation (3.22). [1]

$$\hat{r}_c = \mathbf{R}(\phi,1)\mathbf{R}(\theta,2)\mathbf{R}(\psi,3)\begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \qquad (3.22)$$

Magnetic torque is a cross product of control input and Earth mangetic field vector. It is defined as follows. [1]

$$\begin{bmatrix} T_{cx} \\ T_{cy} \\ T_{cz} \end{bmatrix} = \boldsymbol{\mu} \times \mathbf{B}_{\oplus} = \begin{bmatrix} \mu_y B_z - \mu_z B_y \\ \mu_z B_x - \mu_x B_z \\ \mu_x B_y - \mu_y B_x \end{bmatrix} \qquad (3.23)$$

$\boldsymbol{\mu}$ represents a value given by control input, and $\mathbf{B}$ is the Earth's magnetic field that is defined in the body coordinate system.

Substituting gravity gradient torque (3.20) and the magnetic moment torque (3.23) into the left hand side of equation (2.21), and neglecting non-diagonal moment of inertia terms, it is possible to derive non-linear equations of motion as follows. [1]

$$\begin{aligned} \dot{p} &= -\frac{I_z - I_y}{I_x} qr + \frac{3n^2(I_z - I_y)}{I_x} \sin\phi \cos\phi (\cos\theta)^2 + \frac{1}{I_x}\left(\mu_y B_z - \mu_z B_y\right) \\ \dot{q} &= -\frac{I_x - I_z}{I_y} rp + \frac{3n^2(I_z - I_x)}{I_y} \sin\theta \cos\theta \cos\phi + \frac{1}{I_y}\left(\mu_z B_x - \mu_x B_z\right) \qquad (3.24) \\ \dot{r} &= -\frac{I_y - I_x}{I_z} pq + \frac{3n^2(I_x - I_y)}{I_z} \sin\phi \sin\theta \cos\theta + \frac{1}{I_z}\left(\mu_x B_y - \mu_y B_x\right) \end{aligned}$$

Approximating small angle theory, we could get linearlized dynamics.

$$\begin{aligned} \dot{p} &\approx \frac{3n^2(I_z - I_y)}{I_x}\phi + \frac{n(I_z - I_y)}{I_x} r + \frac{1}{I_x}\left(\mu_y B_z - \mu_z B_y\right) \\ \dot{q} &\approx \frac{3n^2(I_z - I_x)}{I_y}\theta + \frac{1}{I_y}\left(\mu_z B_x - \mu_x B_z\right) \qquad (3.25) \\ \dot{r} &\approx \frac{n(I_y - I_x)}{I_z} p + \frac{1}{I_z}\left(\mu_x B_y - \mu_y B_x\right) \end{aligned}$$

Arranging equation (3.25) and (3.19), linear system for LQG is derived. [1]

$$
\begin{bmatrix} \dot{\phi} \\ \dot{p} \\ \dot{\theta} \\ \delta\dot{q} \\ \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & n & 0 \\ \frac{3n^2(I_z - I_y)}{I_x} & 0 & 0 & 0 & 0 & \frac{n(I_z - I_y)}{I_x} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{3n^2(I_z - I_x)}{I_y} & 0 & 0 & 0 \\ -n & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{n(I_y - I_x)}{I_z} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ p \\ \theta \\ \delta q \\ \psi \\ r \end{bmatrix}
$$

$$
+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{B_z}{I_x} & -\frac{B_y}{I_x} \\ 0 & 0 & 0 \\ -\frac{B_z}{I_y} & 0 & \frac{B_x}{I_y} \\ 0 & 0 & 0 \\ \frac{B_y}{I_z} & -\frac{B_x}{I_z} & 0 \end{bmatrix} \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} \qquad (3.19)
$$

Instead of using y-axis angluar velocity *q*, we defined new state variable $\delta q = q + n$ for Earth-oriented attitude control. This is because LQG controller makes every states to converge into zero, we offset the effect of the orbital mean motion of angular velocity *n* of the orbital coordinate system.

IV. Simulation

1. Simulation Configuration

SILS(Software In the Loop System)

The purpose of SILS is to objectively verify the performance of the ADCS algorithm in PC-based (matlab) on the ground before applying the ADCS algorithm to OBC (On-Board Computer). Therefore, as shown in the [figure 4-1], SILS was implemented using the serial communication between the two separate computers each represents the space environment simulator and ADCS algorithm.

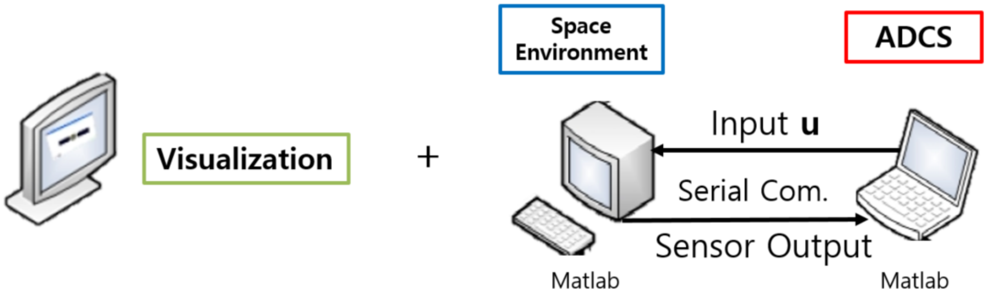


Figure 4-1. SILS Configuration

In Space Environment module, we used a sophisticated disturbance model that effects on the performance of the ADCS in low earth orbit, while ADCS used simplified model. We purposely used different models for each module in order to consider model uncertainty. In addition, simple model reduces the computation load of the OBC (On Board Computer) in ADCS. The [Table. 4-1] shows disturbance models used for each modules.

Table 4-1. Space Environment Module vs. ADCS Module

	Space Environment	ADCS
Moment of Inertia	Considered non-diagonal term	Neglected non-diagonal term
Gravity Model	Two-body + J2 Effect	Two-body
Air drag density	Harris-Priester model	EKF Fitler Noise
Solar Radiation Pressure	Spherical shadow model	EKF Fitler Noise
Magnetic Field Model	IGRF-12(Full order)	IGRF-12(11th order, T.B.D.)

The data transmitted between space environment module and ADCS module are as follows.

Table 4-2. Input Data for Environment Module

	Unit	Form
Gregorian Time	[year month day hour min sec]	[6 x 1]
Orbit Position	$\left[x_{ECI} \quad y_{ECI} \quad z_{ECI}\right](m)$	[3 x 1]
Orbit Velocity	$\left[v_x \quad v_y \quad v_z\right](m/\sec)$	[3 x 1]
Gyro output	$\left[p_{meas} \quad q_{meas} \quad r_{meas}\right](\deg/\sec)$	[3 x 1]
Magnetometer output	$\left[b_x^{meas} \quad b_y^{meas} \quad b_z^{meas}\right](nT)$	[3 x 1]
Sun sensor output		[3 x 1]

Table 4-3. Output Data for Space Environment Module

	Unit	Form
ADCS Input	$\left[u_x^{meas} \quad u_y^{meas} \quad u_z^{meas}\right](Am^2)$	[3 x 1]

2. Simulation Environment

The simulation data starts from April 10, 2010 0: 0: 0, lasts for 24 hours at 1-second intervals. Orbit was selected as a sun-synchronous orbit of height 600 [km].

Table 4-4. Keplar Element

	Value
a (semi-major axis)	6978[km]
e (eccentricity)	0
i (inclination angle)	97.78^o
Ω (right ascension of the ascending node)	18.41^o
ω (right ascension of the ascending node)	0^o
ν (true anomaly)	0^o

Sun-synchronous orbit calculated through the orbital elements in [Table. 4-4], which has a period about 5,800 [sec]. This orbit a mean motion approximately 0.0621 [deg /s] with eclipse included as shown in the [figure. 4-2].

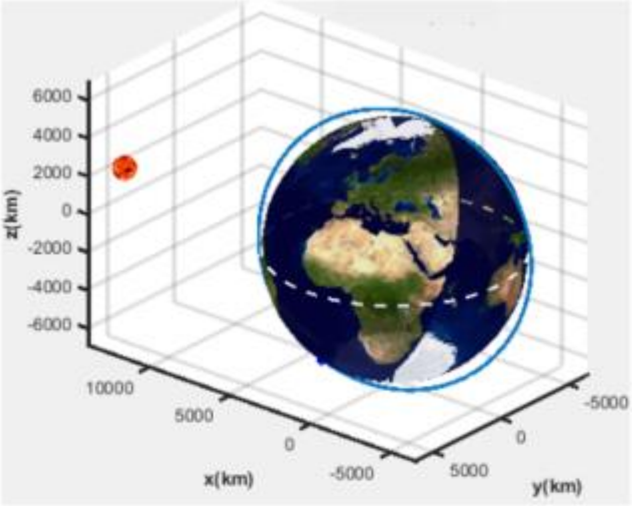


Figure 4-2. Three Dimension of Orbit Visualization

3. Simulation Result

Detumbling mode

Angular velocity attenuation mode is implemented with 2 sensing, 1 control process. The initial conditions were set as $[\phi, p, \theta, q, \psi, r]$ = $[20,10,20,10,20,10]\ \mathrm{deg}, \mathrm{deg}/s$. The convergence condition to change from the angular velocity decay mode to mission mode is derived as a normal value of the amount of change of the magnetic field difference between 2 measurements. This implies that when angular velocity has been decreased, magnetic field difference decreases in proportion to angular velocity. We selected the criteria when angular velocity converges nearly to mean motion.

$$|\Delta\mathbf{B}| = |\mathbf{B}_{k+1} - \mathbf{B}_k| < 10^{-8}\,[\mathrm{T}] \quad (\mathrm{T.B.D.})$$

The value of the above formula has been taken when the angular velocity of the satellite has converged within mean motion. We didn’t use gyroscope measurements for sensing angular velocity since gyro measurements had gyro bias which couldn’t be deleted without estimation. In effect, cubesat could reduce the power consumption using only magnetometer for B-dot control.

The goal of detumbling mode is to attenuate angular velocity less than mean motion of the orbit, which makes small angular velocity assumption of LQG controller effective. The error when linearlization of nonlinear system should be small in order to estimate and control in mission mode successfully. The angular velocity history in detumbling mode is shown in [figure. 4-3].

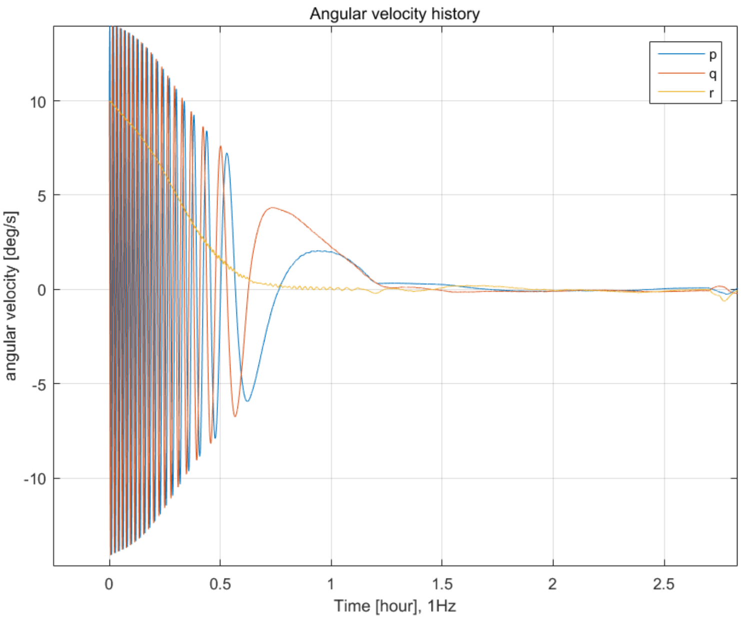


Figure 4-3. Detumbling Result

It can be seen that the three angular velocity converges to 0 within $2.7\,hr\ (\approx 1.7\,orbit\ period)$. Therefore, it is possible to satisfy the first ADCS requirement; the angular velocity decay mode should break out within two days.

Mission mode

Mission mode estimates its attitude using EKF, while uses control gain obtained in the LQG controller. To ensure the ADCS performance, we confirmed whether it meets the ADCS requirements for estimation and control, each.

In order to use EKF, it requires initial state variables and state covariance matrix. Initial state attitude information was estimated using TRIAD algorithm, while the angular velocity and gyro bias was assumed to be mean motion, and 0 each..

$$\mathbf{x}_0 = \left[\mathbf{q}_0 \quad \boldsymbol{\omega}_0 \quad \mathbf{b}_0\right]^T \qquad (4.1)$$

State covariance, which means uncertainty has been selected conservatively (fairly large uncertainty) since it converges to smaller values when system observable.

$$\mathbf{P}_0 = \begin{bmatrix} 0.995^2 \mathbf{I}_{4\times4} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} \\ \mathbf{0}_{3\times4} & n^2\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \left(0.3\left[\deg/s\right]\right)^2 \mathbf{I}_{3\times3} \end{bmatrix} \qquad (4.2)$$

In order to select the EKF of Process noise covariance matrix, we calculated the magnitude of the torque due to the disturbance. Elements required to calculate the torque due to the disturbance is as follows. [29]

Table 4-5. External Torque Parameter

	Explanation	Value	Unit
$\boldsymbol{\theta}$	Maximum departure angle from nadir	45	[degree]
$\boldsymbol{\mu}_{\mathbf{e}}$	Earth gravitational coefficient	3.986×10^5	$\left[\mathrm{km}^3 / s^2\right]$
a	Apogee of orbit	6.978×10^3	[km]
$\mathbf{r}_{\mathbf{mp}}$	Distance between S/C center of mass to center of pressure	0.024	$\left[\mathrm{m}\right]$
I_x	Undeployed Moment of Inertia (x-axis)	0.0088	$\left[\mathrm{kgm}^2\right]$
I_z	Undeployed Moment of Inertia (z-axis)	0.0035	$\left[\mathrm{kgm}^2\right]$
I_x	Deployed Moment of Inertia (x-axis)	0.0269	$\left[\mathrm{kgm}^2\right]$
I_z	Deployed Moment of Inertia (z-axis)	0.0035	$\left[\mathrm{kgm}^2\right]$
A	Surface area	0.02	$\left[\mathrm{m}^2\right]$
q	Reflectance factor	0.6	-
C_d	Drag coefficient	2.2	-
V	Relative satellite velocity w.r.t. wind	7.6e3	$\left[\mathrm{m/s}\right]$

Process noise covariance matrix **Q** has its values derived from using equations (2.4) through (2.6).

$$\sigma_{RRW} = 0.005\left[{}^{o}/\sec\sqrt{\mathrm{Hz}}\right]$$

$$\sigma_{G} = \frac{3\mu_e \cdot \sin\left(2\theta\right)}{2a^3}\frac{|\,\mathrm{I}_{\max} - \mathrm{I}_{\min}\,|}{\mathrm{I}_{\max}} = \begin{cases} \text{Undeployed: } 1.056\times10^{-6}\left[\mathrm{rad/s}^2\right] \\ \text{Deployed: } 1.531\times10^{-6}\left[\mathrm{rad/s}^2\right] \end{cases}$$

$$\sigma_{SRP} = r_{mp}\frac{F_s}{c}A\left(1+q\right)\frac{\cos\beta}{\mathrm{I}_{\max}} = \begin{cases} \text{Undeployed: } 0\left[\mathrm{rad/s}^2\right] \\ \text{Deployed: } 1.202\times10^{-7}\left[\mathrm{rad/s}^2\right] \end{cases}$$

$$\sigma_{AD} = r_{mp}\frac{\rho}{2}V_{rel}^2 C_d\frac{A}{\mathrm{I}_{\max}} = \begin{cases} \text{Undeployed: } 0\left[\mathrm{rad/s}^2\right] \\ \text{Deployed: } 2.961\times10^{-7}\left[\mathrm{rad/s}^2\right] \end{cases}$$

In the case of GPS Mission mode, it has a symmetrical moment of inertia since boom has not been injected. Therefore, disturbance caused by aerodynamics and solar radiation are 0, and there only exists disturbance torque due to gravity gradient torque. On the other hand, in the case of the Magnetometer mission mode, asymmetric moment of inertia is generated by the boom injection, thus the moment due to the friction of the solar wind and the air should be considered in process noise covariance. In the case of eclipse, we excepted the disturbance caused by the solar wind.

Estimation Result

Attitude estimation is performed by the algorithm of the Extended Kalman filter derived earlier, after exiting angular velocity decay mode, and using the initial state and initial covariance matrix obtained by (4.1) and (4.2). Attitude estimation is carried out 2 times while attitude estimation and control for 1 time on a mission mode. The following figure depicts the estimated error of the Euler angles.

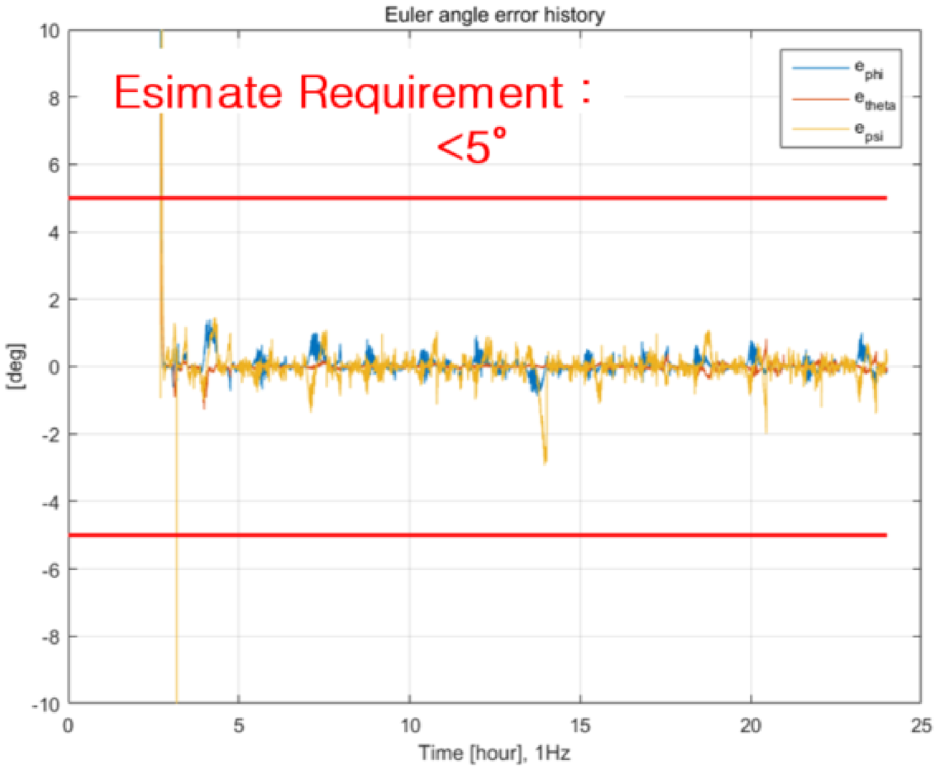


Figure 4-4. Error of Euler Angle Estimation

As seen in [Figure 4-4], the overall error of Euler angles are always estimated less than 5 [deg], which satisfies the third ADCS requirements. The reason there is no error in the first 2.7 [hr] is because it is in angular velocity decay mode, which doesn't estimates attitude. RMS value of the Euler angle errors are presented in the following table. Eclipse and Day are separated.

Table 4-6. RMS Day & Eclipse Attitude Estimation Error

	Day	Eclipse	Total
RMS [deg]	$e_\phi: 0.09, e_\theta: 0.05, e_\psi: 0.17$	$e_\phi: 0.27, e_\theta: 0.1, e_\psi: 0.54$	$e_\phi: 0.18,\ e_\theta: 0.07,\ e_\psi: 0.35$

The above figure shows that ψ angle error bounce of up to 3 [deg]. It is due to observability problem which occurs in eclipse situation since sun sensor is not used.

Next, [Figure 4-5] is a graph showing the angular velocity estimation error.

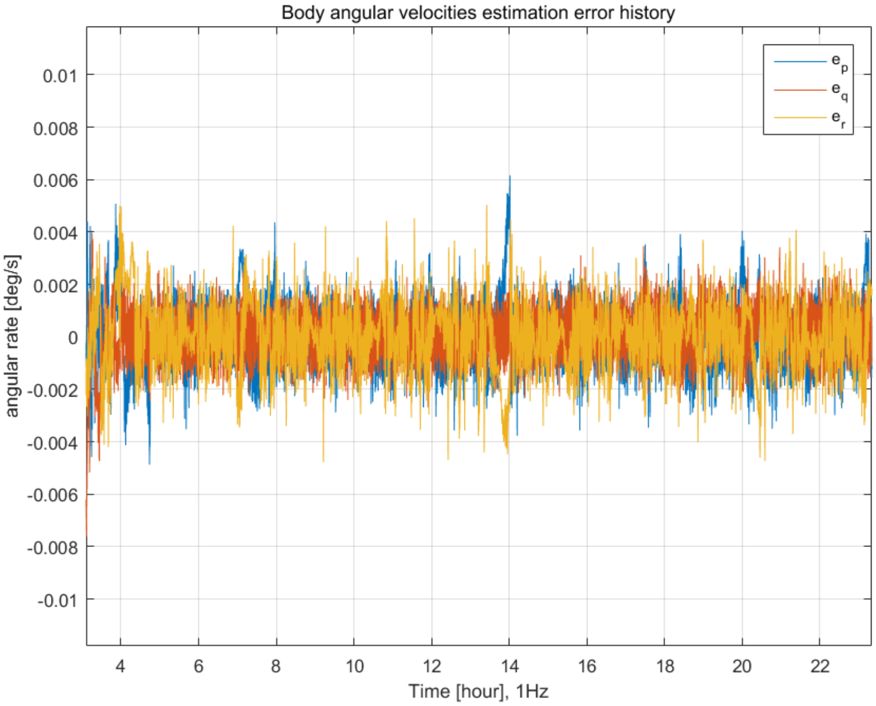


Figure 4-5. Error of Angular Velocity Estimation

The figure above shows the angular velocity estimation error after the angular velocity decay mode. Angular velocity estimation error can be confirmed that the three axis angular velocity has less than 0.004 deg/s erros for entire time period. RMS value of the angular velocity error after 2 trajectory cycle is as follows.

Table 4-7. RMS Day & Eclipse Angular Velocity Estimation Error

	Day	Eclipse	Total
RMS [deg/s]	$e_p = 8.5\times10^{-4}$ $e_q = 7.64\times10^{-4}$ $e_r = 1.1\times10^{-4}$	$e_p = 1.3\times10^{-3}$ $e_q = 7.4\times10^{-4}$ $e_r = 1.2\times10^{-3}$	$e_p = 1.1\times10^{-3}$ $e_q = 7.55\times10^{-4}$ $e_r = 1.3\times10^{-3}$

We observed that Euler angle estimation error and the angular velocity estimation error had less than 1 [deg], 0.1 [deg/s] each. This error level is determined to be capable of a LQG control.

Attitude Control Result

The initial conditions of the mission mode simulation are estimated after the exit of the angular velocity decay mode. Therefore the satellite has its angular velocity for three axes under mean motion of orbit. Thus, we guess the Euler angles using the attitude estimation result of TRIAD, while angular velocity as mean motion *n* of the orbit.

$$\left[\phi, p, \theta, q, \psi, r\right] = \left[\hat{\phi}_0, n, \hat{\theta}_0, n, \hat{\psi}_0, n\right] \ \mathrm{deg}, \mathrm{deg}/\,s$$

An important factor that greatly affects the control performance in the LQG control are weighting matrices. Weight matrix **A** is associated with the state variable, while the weighting matrix **B** for the 3 axis input. Both matrices must be semi-definite matrix, and considered as follows.

$$\mathbf{A} = \frac{1}{35^2} \times diag\left(\left[\frac{1}{1^2}, \frac{1}{5^2}, \frac{1}{1^2}, \frac{1}{5^2}, \frac{1}{10^2}, \frac{1}{10^2}\right]\right) \quad (4.3)$$

$$\mathbf{B} = \mathbf{I}_{3\times3}$$

In equation (4.3), it can be confirmed that both matrices A and B are given positive definite matrices. Weighting matrices used in LQG controller has a physical meaning: the larger the value increases, the more control constraint applies. For SNUGLITE, since Euler angles are important variables than the angular velocity, it has been given greater weight relatively than angular velocities. Further, the values ψ, r that are defined in a $\mathbf{z}_\mathbf{b}$ direction (earth oriented direction) has been given less weight than other axes. This is because, as was introduced in the mission section, of mission data acquisition and communication reasons. Finally, it can be confirmed that the coefficient $1/35^2$ multiplied in front of the **A** prevents the saturation of the magnetic torquer input by weighting more on input. The following figure represents the attitude error of true satellite.

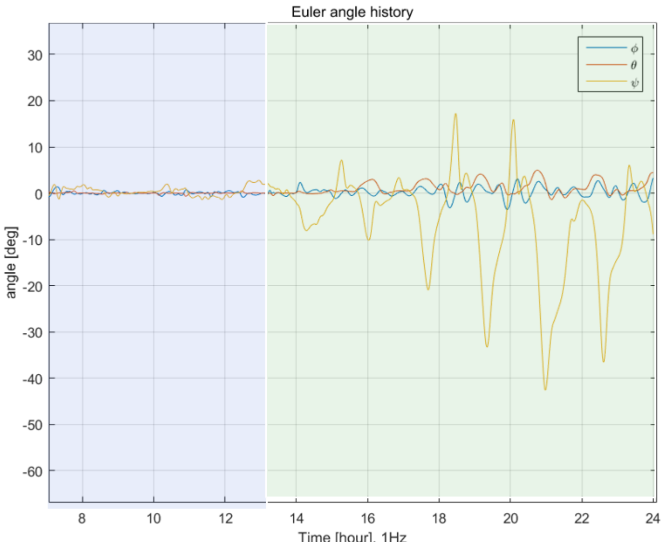


Figure 4-6. Error of Euler Angle Control

The figure above shows the angle information after 2 cycle of orbit. Blue color is a case of performing the GPS mission mode, while green color represents Magnetometer mission mode.

When performing the Magnetoemeter mission, the disturbance caused by the solar radiation pressure and the aerodynamics has been considered since the asymmetrical moment of inertia is added. Therefore, the attitude error of Magnetometer has larger angle errors than that of the mission mode of GPS. The ψ angle error is larger than the ϕ, θ, because we gave greater weight on them. On the other hand, since ϕ, θ has a control error within 10 degrees, it meets 2nd ADCS requirements. This could be verified by using point angle error defined as follows.

$$\theta_{perr} = \cos^{-1}\left(\frac{\mathbf{r}_c \cdot \mathbf{r}_b}{|\mathbf{r}_c||\mathbf{r}_b|}\right) \tag{4.3}$$

$\mathbf{r}_c$: earth pointing vector defined in body frame

$\mathbf{r}_b$: body axis vector defined in body frame

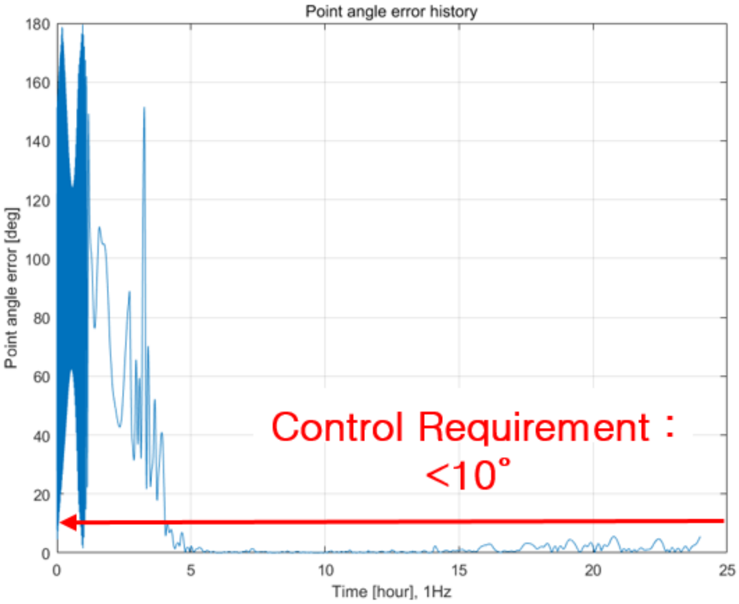


Figure 4-7. Error of Nadir Pointing Attitude Control

The RMS (Root Mean Square) values of Euler angle errors after 2 cycle of orbit are as follows.

Table 4-8. RMS Attitude Control Error

	Day	Eclipse	Total
RMS [deg]	1.857	1.162	1.639

The reason attitude control error in Day case has larger than in the case of eclipse is because disturbance due to solar radiation does not act in eclipse. [Figure. 4-7] shows a point angle error history obtained by the above equation (4.3). The errors of the Earth-oriented angle has maintained its Euler angle error in the range of about 1.639 [deg]. This makes it possible to confirm that our ADCS algorithm has satisfied 3rd ADCS requirement; control error should be within 10 [deg].

Up-side-down Control

In order to verify the last requirement of the ADCS, we considered the worst situation of cubesat when it has reversal attitude. This simulation had a goal to make sure that it is possible to reverse control. Initial conditions for Euler angles are given as $\phi_0 = 180^o$, $\theta_0 = 0^o$, $\psi_0 = 0^o$ while angular velocity as $p = 0\left[{}^0/\sec\right]$, $q = n\left[{}^0/\sec\right]$, and $r = 0\left[{}^0/\sec\right]$. The following figure shows the result.

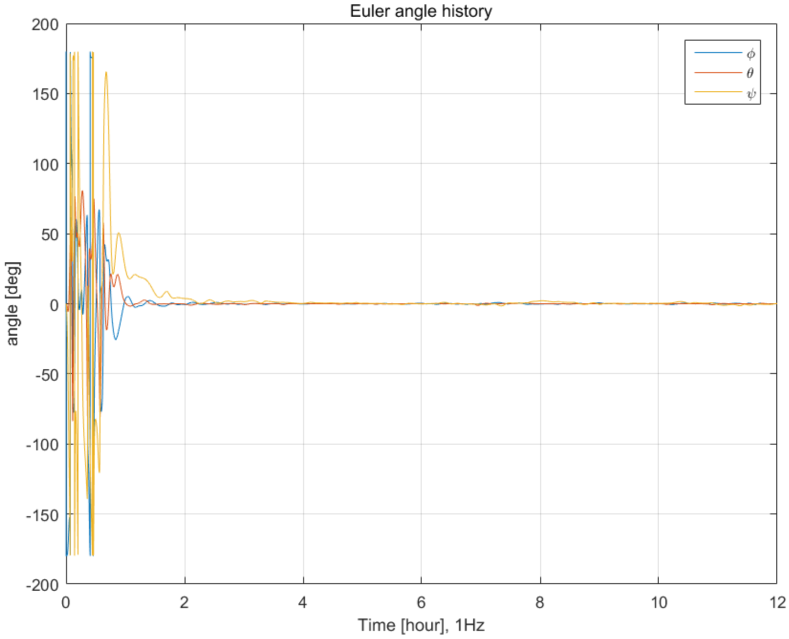


Figure 4-8. Error of Euler Angle after Upside-Down Control

As shown in the figure above, after one cycle of the orbital period, Euler angle converged. The RMS values for Euler angles after 1 cycle had $\phi : 0.164[\deg]$, $\theta : 0.042[\deg], \psi : 0.25[\deg]$. The simulation of inversion control met the 4th ADCS requirement of the attitude control.

V. Conclusion

In order to carry out the mission of the satellite, the bottom surface of the satellite should always point less than $10°$ of point error with respect to the earth-oriented attitude control. Furthermore, since it requires a lot of energy, the attitude control should be performed in ways to conserve energy. For instance, one can save energy as well as meet all the accuracy requirements by using the EKF and LQG controllers. In our research, we used the three-axis magnetic torque as an actuator, while three-axis gyro, magnetic field sensor, and 2 axis sun sensor were used as the attitude determination sensors. The attitude control was divided into two stages: detumbling mode—which attenuates the angular velocity—and mission mode—which maintains the nadir-pointing attitude control. In the angular velocity decay mode, the B-dot control was used to attenuate angular velocity, while mission mode performed the attitude control using the LQG and EKF.

In this paper, we have demonstrated the sequence of events to explain how ADCS met all four of the ADCS requirements. In the angular velocity attenuation simulation the results showed that the initial rapid angular velocity of the satellite had been attenuated less than mean motion of the orbital coordinate system within two days. In the mission mode simulation, not only has the attitude control requirement proven to met the ADCS requirement, but also the attitude determination in the low earth orbit environment, such as the eclipse and various disturbance cases.

We believe that verifying the ADCS algorithm before launching the cubesat to be critical for the successful mission as well as for the cubesat’s survival. We hope that our study will contribute to furthering the research on the development and verification of the PILS (Processor In the Loop System) based ADCS.

VI. References

[1] 김영두, “Attitude determination and control of cubesat using magnetic torquer and LQR controller”, 서울대학교, 2015.

[2] Psiaki, M., “Magnetic Torquer Attitude Control via Asymptotic Periodic Linear Quadratic Regulation” Journal of Guidance, Control, and Dynamics, Vol. 23, No. 3, May-June 2000, pp.532-538.

[3] Daniel M. Torczynski, “Magnetorquer Based Attitude Control for a Nanosatellite Testplatform_Torczynski”, Delft University of Technology, 2010

[4]Napoleon E. Cornejo, et.al, “Model-Based Fault Detection for the DELFI-N3XT Attitude Determination System”, Delft Univerisity of Technology

[5] Jens Gießelmann, “Development of an Active Magnetic Attitude Determination and Control System for Picosatellites on highly inclined circular Low Earth Orbits”,School of Aerospace, Mechanical and Manufacturing Engineering, Science, Engineering and Technology Portfolio RMIT University, June 2006

[6] J. Reijneveld and D. Chourkroun, “Attitude Control System of the DelfI-N3xT Satellite”, Flight Dynamics, GNC, and Avionics 6 (2013) 189-208

[7] Torben Graversen, et. al, “Attitude Control system for AAU CubeSat”, Aalborg University, 2002

[8] Basile Graf, “SwissCube Control Algorithm Design and Validation”, Laboratoire d’Automatique EPFL, February 23, 2007

[9] Gyeonghum Kim, “CNUSAIL-1 큐브위성의 자세결정 알고리듬 설계 및 성능분석”, Chungnam National University, 2014

[10] 윤형주, “위성 자세제어계 설계”, 항공우주연구원 2015년도 큐브위성 경연대회 1차 교육 워크숍, 2015

[11] Jordi MARTIN BENET, “SwissCube Attitude Determination Algorithm Design and Validation”, Master Project, Laboratoire d’Automatique EPFL, July 27, 2007

[12] J. Reijneveld and D. Choukroun, “Attitude Control System of the DELFI-N3T Satellite”, Progress in Flight Dynamics, GNC, and Avionics 6 (2013) 189-208

[13] Kasper Fuglsang Jensen, et. al, “Attitude Determination and Control System for AAUSAT3”, Master Thesis, Aalborg University, 2009

[14] N. Kemal Ure, “The Development of a Software and Hardware-in-The-Loop Test System for ITU-PSAT II Nano Satellite ADCS”, Massachusetts Institute of Technology, Aerospace Controls Lab

[15] Princeton Satellite Systems, Inc., “Attitude And Orbit Control Using The Spacecraft Control Toolbox v4.6”, Princeton Satellite Systems

[16]임이랑, “초소형 인공위성의 자세결정 및 제어 연구”, Master’s Thesis, KAIST Space Exploration Engineering Program, 2011.

[17] Roger R. Bate, “Fundamentals of Astrodynamics”

[18] James R. Wertz, “Spacecraft Attitude Determination and Control”, Volume73

[19] Takaya Inamori, et. Al., “Compensation of time-variable magnetic moments for a precise attitude control in nano- and micro-satellite missions”, Advances in Space Research, vol.48, no.3, 2011, pp.432-440.

[20] Erwan Thebault, et. Al., “International Geomagnetic Reference Field: the 12th generation”, Earth, Planets and Space (2015)

[21] Andrew J. Turner, “An OpenSource Extensible Spacecraft Simulation And Modeling Environment Framework”, Virginia Polytechnic Institute and State University, 2003.

[22] 김유단, et. Al., “비행동역학 및 제어”, 경문사

[23] Yaguang Yang, “Quaternion based model for momentum biased nadir pointing spacecraft”, Aerospace Science and Technology 14 (2010) 199-202

[24] Chris Hall, “Chapter 4. Attitude Determination”, March 3, 2003

[25] J. R. W. a. W. J. Larson, Space Mission Analysis and Design, Third ed.:
Microcosm Press and Springer, 1999.

[26] E. Babcock, "CubeSat Attitude Determination via Kalman Filtering of
Magnetometer and Solar Cell Data”, 25th Annual AIAA/USU Conference on Small Satellites

[27] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman Filtering for
Spacecraft Attitude Estimation," Journal of Guidance Control and Dynamics, vol. 5, pp. 417-

429, 1982.

[28] Enrico Silani, et. Al. “Magnetic spacecraft attitude control: a survy and some new results”, Dipartimento di Elettronica e Informazione, Control Engineering Practice 13 (2005) 357-371.

[29]Sung Hyuk Choi, et. Al., “Attitude Determination of cubesat during eclipse considedring the satellite dynamics and torque disturbance”, Department of Mechanical and Aerospace Engineering, Seoul National University, JKSAS 2016.

요약

SNUGLITE는 성공적으로 임무를 수행하기 위해 운용 기간 중에 확득한 임무 데이터를 지향성이 있는 S대역 송신기로 지상국에 전송해야 하므로 지구지향 자세 제어를 실시해야 한다. 본 논문에서는 SNUGLITE의 성공적 임무수행을 위한 자세결정 및 제어 서브시스템 ADCS (Attitude Deter-mination and Control System)요구사항을 작성했으며, 자세결정 센서로 자이로스코프, 태양 센서, 자기장 센서, GPS 수신기를 사용했고, 구동기는 3축 자기장 토커를 사용하였다. 또한 저궤도 우주 환경에서 외란에 의한 토크, 일식 등에도 지구를 지향하도록 LQG 제어기를 채택하였다.

제시한 ADCS를 SNUGLITE의 운영 시나리오에 맞추어 검증하기 위해 저궤도 환경을 고려한 SILS (Software In the Loop System)를 구축하였다. 발사체가 선정되지 않았으므로, 최악의 상황에서도 ADCS 요구사항을 만족하기 위해 일식 비중이 높은 고도 600 [Km]의 태양동기궤도를 큐브위성의 궤도로 가정하였다. 저궤도 우주환경을 모사하기 위해 큐브위성의 궤도, 자세역학에 가장 큰 영향을 주는 중력구배, 태양풍, 공기마찰력을 고려하여 우주환경 시뮬레이터를 개발하였다. 이때, 큐브위성의 궤도 역학과 자세 역학은 서로 연결되어 상호 영향을 주도록 구현하였다. 실제 큐브위성의 운용 상황을 모사하기 위해 ADCS 알고리즘과 우주환경 시뮬레이터를 서로 다른 컴퓨터에서 작동하도록 분리하고, 시리얼 통신으로 정보를 주고받도록 SILS를 개발하였다. 이때 운영 시나리오를 고려하여 P-POD (Poly Picosatellite Orbital Deployer)에서 사출된 이후 B-dot 제어를 수행하는 각속도 감쇠모드와 LQG 제어기로 지구지향 제어를 수행하는 자세유지 모드로 나누었다. 또한 자세유지모드는 붐 사출 전인 Phase 1 모드와 붐 사출 후인 Phase 2 모드를 나누어 운영 시나리오를 구성하였다.

시뮬레이션은 ADCS 요구사항을 만족하는지 여부에 따라 성능의 합/불 여부를 결정하였다. 시뮬레이션 결과, 자세결정 오차는 일식에서 5도 이내, 비일식에서 2도 이내로 추정했으며, 자세제어 오차는 일식에

서 10 도， 비일식에서 5 도 이내로 수렴하였다． 또한， 각속도가 이틀 이내에 mean motion 이내로 수렴함을 보였다． 마지막으로， 큐브위성이 뒤집어진 상황에서도 지구지향 하도록 자세 제어함을 보여 모든 요구사항을 만족함을 보였다．

본 논문은 Matlab 기반으로 개발한 SNUGLITE 의 LQG 알고리즘을 개발 및 검증 하였으며， 향후 OBC（On-Board Computer）기반의 PILS（Processor In the Loop System）개발에 있어 필요한 우주환경을 제공한다． 이는 큐브위성 SNUGLITE 의 임무 수행을 성공적으로 수행하는데 기여할 것으로 예상된다．

주요어： SILS， LQG 제어기， 자세제어 서브시스템， 큐브위성, 궤도환경

학 번： 2014 – 22513