FEEDBACK CONTROL OF CUBESAT FOR ATTITUDE AND TRAJECTORY CORRECTIONS

A Thesis

Presented to

the Faculty of the Department of Electrical and Computer Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in Electrical Engineering

by

Mandar Phadnis

December 2013

FEEDBACK CONTROL OF CUBESAT FOR ATTITUDE AND

TRAJECTORY CORRECTIONS

Mandar Phadnis

Approved:

Chair of the Committee
Dr. Earl J. Charlson, Professor
Electrical and Computer Engineering

Committee Members:

Dr. Robert Provence, Lecturer
Electrical and Computer Engineering

Dr. Matthew Franchek, Professor
Mechanical Engineering

Dr. Suresh K. Khator, Associate Dean
Cullen College of Engineering

Dr. Badri Roysam, Professor and Chair
Electrical and Computer Engineering

ACKNOWLEDGEMENTS

Foremost, I would like to express the deepest appreciation to my technical guide, Dr. Robert Provence, who was a source of continuous knowledge, encouragement and help. Without his guidance and inspiration this thesis would not have been possible.

I would like to thank my committee chair, Professor Earl Charlson for his continuous support and assistance throughout my thesis. I am grateful to Professor Matthew Franchek, for his ideas and assistance with mathematical modeling and simulation.

I would like to thank my family for keeping me sane and supporting me through the rough times. Finally, I would like to thank my friends for making this a wonderful and memorable experience.

FEEDBACK CONTROL OF CUBESAT FOR ATTITUDE AND TRAJECTORY CORRECTIONS

An Abstract

of a

A Thesis

Presented to

the Faculty of the Department of Electrical and Computer Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in Electrical Engineering

by

Mandar Phadnis

December 2013

Abstract

In order to analyze the performance of Cube Satellites with regards to orbit maintenance and attitude maneuvers, a three unit (3U) CubeSat with a mass of four kilograms in six degrees of freedom, is studied and modeled in this research. With a major constraint on their mass and volume, thrusting in small satellites is a limited operational resource. An error in the thrust vector can propagate to a large error in the attitude or trajectory over time. Six degree of freedom rigid body dynamics is implemented using unit quaternion based torque induced roll-pitch-yaw motion. A single thruster is included along the roll axis with a simulated error and the effects due to thrust on the satellites trajectory and attitude are observed over one orbit. Spherical harmonic gravity is implemented to account for the gravitational disturbances in the two body model. Attitude correction is achieved using quaternion feedback control while trajectory corrections are attained using nonlinear state feedback control with backstepping technique. Stability analysis is done using Lyapunovs second method to assure global asymptotic stability. Numerical simulations are conducted to demonstrate attitude and trajectory corrections of the CubeSat using above control strategies.

Table of Contents

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Background

Recent advancements in nanosatellite and picosatellite technologies have redefined the field of space exploration. These satellites have transformed the initially expensive and intricately complex satellite technology into a modular and cheaper endeavor with the extensive implementation of commercial off-the-shelf equipment. Increasing number of universities and other organizations are developing and launching such satellites capable of performing myriad scientific experiments, giving immense exposure to engineering students in satellite design and deployment lifecycles.

Cube Satellites (CubeSats) are types of picosatellites that originated around 1999 [1]. They were the brainchild of Dr. Jordi Puig-Suari of California Polytechnic State University (Cal. Poly.) and Dr. Bob Twiggs of Stanford University. The CubeSat standard was the result of an evolution of CubeSats and their deployment mechanisms developed mainly at these Universities. One of the primary aims of developing the CubeSat standard was to provide universities and private companies a quick and affordable entry into space by launching these picosatellites as secondary payloads with standardized deployment mechanisms. CubeSats have been launched in different configurations of one unit (1U), two units (2U) and three units (3U). Each unit of CubeSat is a 10 X 10 X 10 centimeters cube weighing roughly one kilogram.

1.2 Past and Future Missions

Since 1999, over seventy CubeSat have been successfully placed into orbit. Under the NASA sponsored LONESTAR program, University of Texas, Austin and Texas A&M have

joined field of CubeSats. The main goal of this program is to develop and test autonomous rendezvous and docking, in addition to communications, control systems and formation flying of CubeSats. AggieSat 2 and Bevo-1 CubeSats already successfully placed into orbit which were designed and developed completely by students of these universities.

In accordance with the LONESTAR, the next phases of CubeSats are being designed. AggieSat 4 and Bevo 2 CubeSats are planned for launch in 2014. UT Austin's ARMADILLO CubeSat developed with a purpose of performing atmospheric measurements and debris instrumentation is scheduled for launch by the end of 2014.

1.3 Motivation

With the advantages of flexibility, upgradability and modularity, these picosatellites, however, come with a major constraint on their propulsion systems due to the strict limitation on their mass, power and volume. This makes orbital maintenance and correction a challenging task. Micro-propulsion systems are proving to be a promising answer in such applications. But the restricted fuel supply in these systems makes orbital maneuvers a limited and costly operation. An error in the orbit maintenance thrusts or in the satellite attitude during such a thrust can propagate over time to a large error in the final position. Hence, accurate and precise thrusts are necessary to avoid unnecessary orbit corrections. Control algorithms are required to detect and correct any trajectory errors, with minimum control effort.

In this regard, we study a 3U CubeSat in six degrees of freedom and analyze the error propagation in trajectory and attitude resulting from errors in satellite attitude during orbital maneuvers or due to thruster misalignments. For example, the 3U CubeSat being developed at UT Austin, comes with a single fixed cold-gas thruster as shown in Fig. 1.

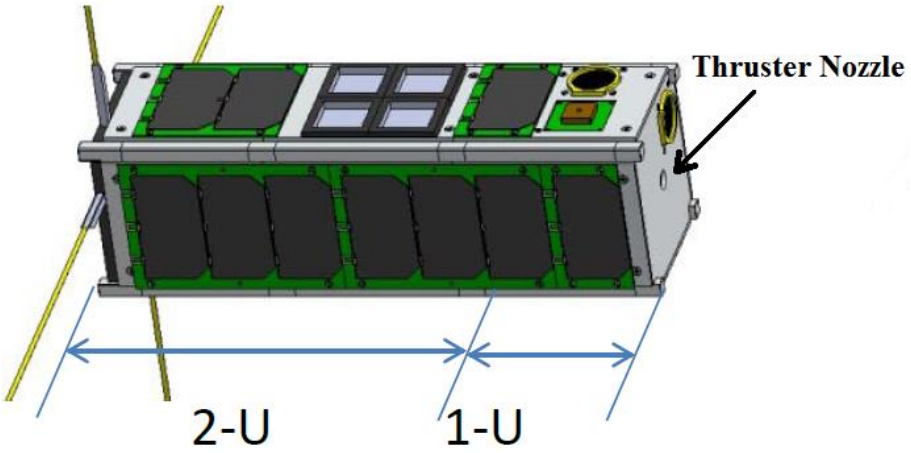

Figure 1. CubeSat Structure showing thruster nozzle; image taken from Austin Satellite Design datasheets online (http://austinsat.net).

During thrust maneuvers, the satellite may have an attitude error or there might be a mechanical error in the thruster alignment. Such errors due to pointing inaccuracies may result in a variance in the desired thrust vector. An analysis of these types of errors can be crucial in development of control algorithms capable of in-orbit corrections.

1.4 Contributions

The goal of this thesis is to design and develop an attitude and trajectory control system for the CubeSat class of small satellites. This system can be used to correct an error in pointing or in the orbital trajectory of the CubeSat. It is achieved by first developing a mathematical model of the satellite in six degrees of freedom in Spherical harmonic gravity field. Quaternion feedback control is used for attitude correctional maneuvers using three axis reaction wheels for torque actuation. Trajectory control is implemented by designing a nonlinear state feedback controller, that uses the error in the actual state of the satellite and the desired state simulated by a trajectory planner and by means of backstepping control technique, provides a globally asymptotically stable control law that uses thrust actuation for achieving desired orbit. This thesis assumes there are ideal sensors for providing the state feedback of the satellite.

CHAPTER 2

LITERATURE REVIEW

2.1 The CubeSat Standard

CubeSats owe their success primarily to their strict adherence to the initially developed standard bringing down their costs and deployment times. With a volume of 10cm X 10cm X 10cm, a one unit (1U) CubeSat can weigh at most 1.33kg, while a three unit (3U) CubeSat can have a volume of 10cm X 10cm X 34cm with a mass of around 4.0 kg [2]. Such standardizations make the deployment much easier using already available and extensively tested deployment mechanisms. A detailed description of CubeSat standard can be found in [2].

2.2 Micro-propulsion Systems

Initially CubeSat were not required to have any form of propulsion for their entire mission duration. However, with the advancement of CubeSats, various government and private organizations other than just universities started developing these picosatellites. The mission requirements began increasing to translational and attitude maneuvers, orbit maintenance and de-orbiting burns, necessitating the need for a capable propulsion system. Conventional propulsion systems were usually too bulky for the CubeSats and had to be heavily modified to meet the volume, weight and power constraints. Various micro-propulsion ideas are being developed with an application to CubeSats. Some of those are discussed below.

2.2.1 Miniature Ion Electrospray Thrusters

A plasma discharge is bombarded by electrons to form ions. These ions are extracted using electrostatic extraction grids and accelerated to provide thrust. Typically Xenon gas is

used although mercury vapor, Argon and Krypton have been employed in the past. These thrusters have no moving parts, occupy significantly less space and provide very high specific impulse of around 2500 seconds [3]. They can be used to provide smoothly modulated thrust vectors as opposed to discrete pulsed thrust impulses.

2.2.2 Pulsed Plasma Thrusters

In micro Pulsed Plasma Thrusters (μPPTs), a solid Teflon fuel bar is etched in an electric arc discharged over one of its surfaces to generate plasma. This plasma is accelerated by two electrodes under electrodynamic forces using self-generated magnetic-fields and current to provide thrust [4]. They have the advantage of having no moving parts, solid long lasting fuel and a high specific impulse (≈ 500 seconds). They are usually pulsed and may cause jitter in flexible structures. They often come with integrated power system and charging-discharging synchronization microcontrollers.

2.2.3 Cold gas Micro-propulsion System

Cold gas micro-propulsion systems offer a great deal of simplicity. Pressurized gas propellant is vented through a converging-diverging nozzle to provide thrust. Several benign propellant options are available for environment critical missions. Although they are cost effective, they provide very low specific impulse (<100 seconds) and hence limited Δv .

One of such cold gas thrusters is being developed at University of Texas at Austin. The propulsion system provides thrust by expelling liquid propellant compressed at around 100 psi serially through three valves (to maximize Δv) and a converging-diverging nozzle. The measured impulsive thrust force is 151 mN per pulse at 85^{o} C [5]. This system, weighing at 400 grams including 90 grams of propellant, has been tested to provide at least 10 m/s of Δv capability. Encasing the main and secondary tanks, internal piping and the nozzle in one

block of plastic, the entire system would occupy a total volume of less than 0.5U. The in-orbit demonstration for this thruster is planned to occur on the Bevo-2 satellite due later in 2013 [5].

2.3 Spherical Harmonic Gravity Representation

Since the Earth is not a perfect sphere with an uneven distribution of mass throughout, its gravitational potential deviates from the Newton's point mass gravity model. This model describes gravitational force acting on a satellite as,

$$\vec{F} = -\frac{(GM_e m_s)}{R^2}\left(\frac{\vec{R}}{R}\right) \quad \text{(N)}, \tag{1}$$

where G is the gravitational constant, M_e is the mass of the Earth, m_s is the mass of the satellite, and R is the radius of the orbit.

The Earth's actual gravity field is determined by the uneven distribution of the material that it consists. Being an oblate sphere, it bulges at the equatorial regions and flattens at the poles. The gravitational potential changes over the surface of the Earth and so does the acceleration due gravity acting on a satellite orbiting it. These perturbations are significant especially in Low Earth Orbits (LEOs) (300km to 1000km) where the gravitational anomalies are dominant. If ignored, these perturbations can result in considerable errors in the satellites simulated position and velocity.

The Spherical Harmonic model has been widely used to approximate the Earth's gravity to account for its non-spherical nature. The gravitational potential for this model is as given in [6]

$$U(\mathrm{r},\phi,\lambda) = \frac{\mu}{r} + \sum_{n=2}^{\infty}\sum_{m=0}^{n}\frac{\mu}{r}\left(\frac{R_e}{r}\right)^n P_{n,m}(\varepsilon)(\mathrm{C}_{n,m}\cos(\mathrm{m}\,\lambda) + \mathrm{S}_{n,m}\sin(m\lambda)) \tag{2}$$

,

where, μ is Earth's gravitational constant, R_e is the equatorial radius, r is the magnitude of position vector X = (x_1 x_2 x_3), $P_{n,m}$ are the associated Legendre functions calculated as

$$P_{n,m} = (1-\varepsilon^2)^{m/2}\left(\frac{\partial^m P_n}{\partial \varepsilon^m}\right), \tag{3}$$

where P_n are the Legendre polynomials. Parameter ε is the sine of the latitude ϕ, given as ε = sin (ϕ) = $\frac{x_3}{r}$ and the longitude is computed as $\lambda = \tan^{-1}\left(\frac{x_2}{x_1}\right)$.

The spherical harmonic gravity potential has the form

$$U(\mathrm{r},\phi,\lambda) = \frac{\mu}{r} + U_{zonal}(\mathrm{r},\phi) + \mathrm{U}_{\mathrm{sec}torial}(\mathrm{r},\lambda) + \mathrm{U}_{tesseral}(\mathrm{r},\phi,\lambda) \tag{4}$$

.

The zonal harmonics account for gravitational variations along the latitude, sectorial harmonics consider the variations along longitudes and tesseral harmonics account for checkered sections of earth (dependent on both latitude and longitude). When these functions are weighted by the gravity coefficients obtained from Geodesy (EGM2008) [7], they represent the true variations in the gravity potential to a high level of accuracy.

CHAPTER 3

MATHEMATICAL PRELIMINARIES

3.1 Co-ordinate Systems

The following co-ordinate systems have been employed in this thesis.

a) Earth Centered Inertial (ECI)

The ECI frame shown in Fig. 2 has its origin at the center of the Earth, with x axis and y axis lying in the equatorial plane and z axis pointing North to complete a right handed co-ordinate frame. The co-ordinate axes and the origin are fixed in inertial space.

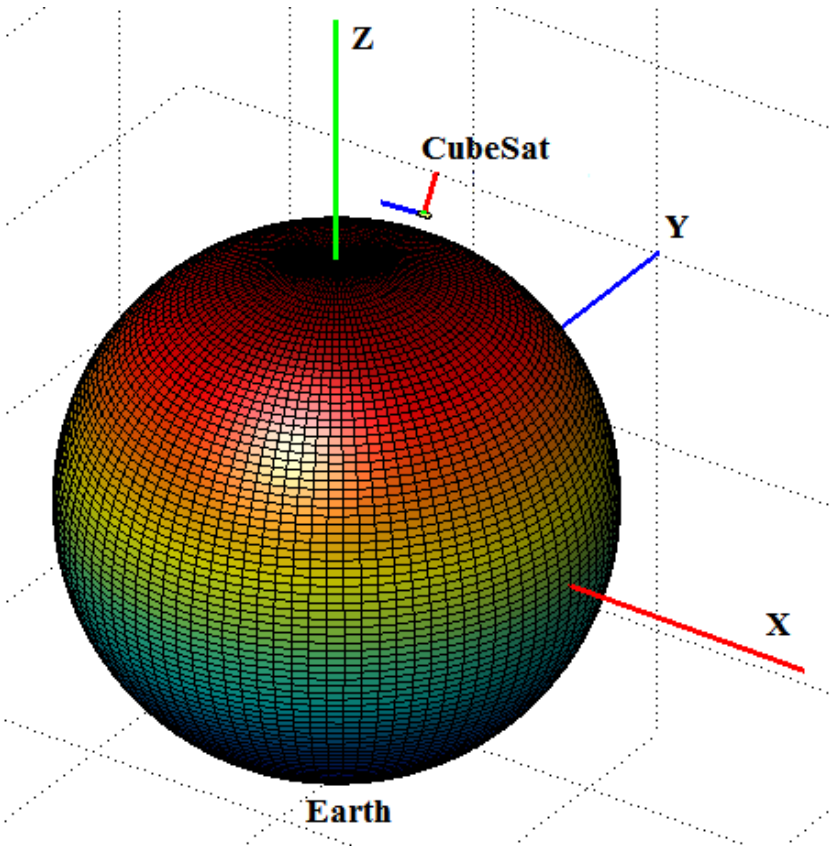

Figure 2. Earth Centered Inertial Frame.

b) Earth Centered Earth fixed (ECEF)

This frame also has its origin at the center of the Earth with the axes fixed to the rotating Earth. The rotation which is along the z axis is obtained by a rotation matrix that converts the co-ordinates of a given point in the inertial frame to co-ordinates in the rotating frame based on the constant angular velocity of the rotation of the Earth.

c) Body Fixed Frame

With the center of mass of the CubeSat as the origin, this co-ordinate axis has its x axis as the yaw, y axis as roll and z axis as pitch. It is fixed to the satellite body.

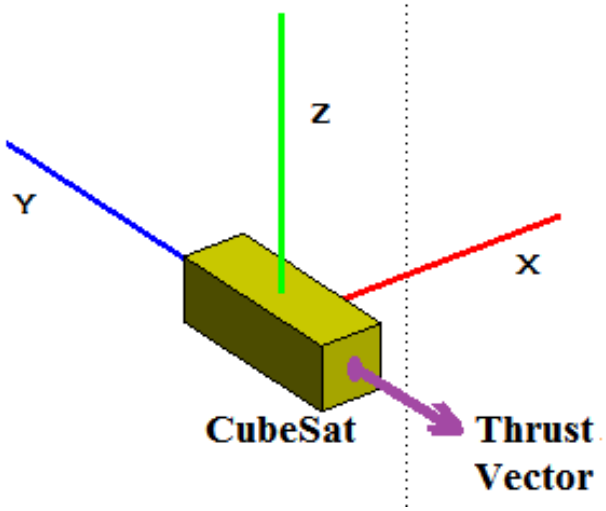

Figure 3. CubeSat Body fixed frame.

3.2 Classical Orbital Elements and r, v vectors

Any orbit can be completely specified with regards to its shape, size and orientation, using five orbital elements, namely, the semi-major axis (a), eccentricity (e), inclination (i), longitude of ascending node (Ω) and argument of periapsis (ω). A sixth orbital element, true anomaly (T), is required to determine the position of the satellite in the orbit [8]. An inertial frame of reference with origin at the center of the Earth is used to define the state vector in Cartesian coordinates x = [r, v]. The radius and velocity vectors are convenient while calculating the orbit trajectory. They can be suitably integrated with the gravity model to

propagate the orbit using numerical integration techniques. Fig. 4 shows the classical orbital elements.

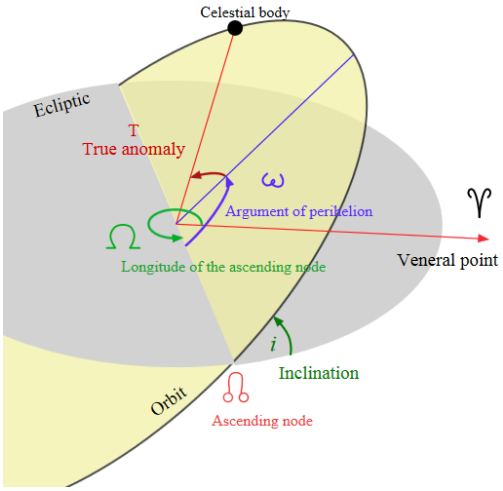

Figure 4. Classical Orbital Elements [18].

3.3 Euler Angles

Euler Angles use three elemental rotations about mutually orthogonal axes to represent the rotation of an object or a coordinate frame in a three dimensional Euclidean space [9]. These sequential intrinsic rotations usually denoted by $[\phi,\theta,\psi]$ are also commonly referred to as yaw, pitch and roll angles. There is some ambiguity related to the use of Euler angles as the same rotation can be represented by different sets of Euler Angles and it further depends on the sequence in which these rotations are applied. In order to reduce the ambiguity this thesis uses the $R_z(\phi)R_x(\theta)R_z(\psi)$ convention for Euler Angle Rotation among many other conventions.

3.4 Euler Rotation Matrix

Euler angles are typically implemented using rotation matrices. Given below are the rotation matrices for Euler angle rotations of an angle α about x, y and z axis respectively [9], are as follows:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix}, \quad \text{(5-a)}$$

$$R_y(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}, \text{ and} \quad \text{(5-b)}$$

$$R_z(\alpha) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad \text{(5-c)}$$

Every elemental rotation can be represented by a corresponding Rotation Matrix. The final orientation of a vector can be obtained by the multiplication of the three matrices corresponding to the three Euler Angles. Notice that the matrix multiplication is not commutative faithful to the rotations in space. According to the Euler Angle convention used in this thesis, the rotation matrix can be denoted by the following sequence of rotations about their respective angles. Using Eqns. (5) we get the rotation matrix as

$$R_z(\phi)\, R_x(\theta)\, R_z(\psi)$$

$$= \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\phi)\cos(\psi) - \sin(\phi)\cos(\theta)\sin(\psi) & -\cos(\phi)\sin(\psi) - \sin(\phi)\cos(\theta)\cos(\psi) & \sin(\theta)\sin(\phi) \\ \sin(\phi)\cos(\psi) + \cos(\phi)\cos(\theta)\sin(\psi) & -\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\theta)\cos(\psi) & -\sin(\theta)\cos(\phi) \\ \sin(\theta)\sin(\psi) & \sin(\theta)\cos(\psi) & \cos(\theta) \end{bmatrix}. \quad \text{(6)}$$

Euler Rotation matrix can be multiplied to a given vector to rotate it in a desired orientation while preserving its length. If a rotation matrix R is used to rotate a vector from world frame to body frame, an inverse rotation from body frame to world frame can be obtained by multiplication using the transpose of matrix (R^T) [9].

Rotation matrices use nine parameters to represent three degrees of freedom, thus having a high degree of redundancy. They are also prone to numerical integration errors which can quickly propagate to large anomalies in rotations. Rotation matrices have singularities at certain orientations and care needs to be taken to avoid them [9].

3.5 Quaternions

Euler's Theorem states any rotation of a rigid body in three dimensional space can be represented as a rotation by a single angle called the Euler Angle about a fixed axis called the Euler Axis, usually denoted by $\hat{e}$ [11]. Quaternions are used to implement the Euler's theorem. A quaternion uses four parameters to represent rotation and has the following form

$$q = q_0 + q_1 i + q_2 j + q_3 k \tag{7}$$

and are also denoted as $q = [q_0, q_1, q_2, q_3]$,

where q_0 is the scalar part while $q_v = [q_1, q_2, q_3]$ is the vector part.

The norm (||.||) of a quaternion is obtained by taking the square root of its dot product with itself, or

$$q \cdot q = \sqrt{{q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2} \tag{8}$$

A unit quaternion can be obtained by dividing a quaternion by its norm ($\frac{q}{\| q \|}$). This process is called Normalization of the quaternion. General rotations are represented using unit quaternions (|q| = 1) that lie on a unit hypersphere in four dimensions (R^4) [10]. Any general rotation of θ degrees about an axis u = [u_1 u_2 u_3], where |u| = 1 can be represented by a unit quaternion as

$$q_0 = \cos(\theta / 2) \text{ and} \tag{9-a}$$

$$q_v = \sin(\theta / 2)\hat{u} . \tag{9-b}$$

The inverse of a unit quaternion is simply its conjugate defined as

$$q^{-1} = q^* = q_0 - q_1 i - q_2 j - q_3 k . \tag{10}$$

It cancels out the effect of the rotation q. Simple geometric intuition can explain that q^{-1} is rotation by the same angle θ with the axis pointing in the opposite direction [10].

If a rotation q_a is followed by a rotation of q_b, the overall rotation is given by $q_b * q_a$ where the quaternion multiplication is defined as [12],

$$q_b * q_a = \left[(q_{b1})(q_{a1}) - (q_{bv} \cdot q_{av}),\quad q_{b1}(q_{av}) + q_{a1}(q_{bv}) + (q_{bv} \times q_{av}) \right] . \tag{11}$$

A quaternion –q represents the exact same rotation as quaternion q. This is evident from the fact that a rotation of theta about an axis u is the same as the rotation of –theta about the opposite axis of –u. A quaternion of q = [±1 0 0 0] represent no rotation. Quaternions have low redundancy with only four parameters to describe three degrees of freedom. They are

also immune to numerical integration errors as they can be easily normalized to represent pure rotation. They are free from singularities and are unique for every rotation [10].

3.6 Conversions

3.6.1 Rotation Matrix to Quaternion

Obtaining quaternion from a rotation matrix can be a bit complicated.

Given a rotation matrix $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$,

the quaternion $q = [q_0\ q_1\ q_2\ q_3]$ can be calculated as [11]

$$q_0 = \pm\frac{1}{2}\sqrt{1+trace(R)}\ , \tag{12}$$

where $trace(R) = r_{11} + r_{22} + r_{33}$ and

$$q_1 = \frac{r_{23}+r_{32}}{4q_0} \quad q_2 = \frac{r_{31}+r_{13}}{4q_0} \quad q_3 = \frac{r_{12}+r_{21}}{4q_0}\ . \tag{13}$$

In case $q_0 \approx 0$, an alternate formula is applied using the highest diagonal element of the matrix. For example, if r_{11} is the highest element then the following derivations can be employed:

$$q_1 = \frac{1}{2}\sqrt{1+r_{11}-r_{22}-r_{33}}\ , \tag{14}$$

$$q_0 = \frac{r_{23}-r_{32}}{4q_1}\ , \quad q_2 = \frac{r_{12}+r_{21}}{4q_1}\ , \text{ and } \quad q_3 = \frac{r_{31}+r_{13}}{4q_1}\ . \tag{15}$$

Analogous formulas are used for the other cases of r_{22} or r_{33} being the highest elements. An algorithm was implemented in Matlab to use these multiple cases for calculating the most accurate quaternion from the rotation matrix or Axis Angle.

3.6.2 Quaternion to Rotation Matrix

Rotation matrices are convenient when performing co-ordinate transformations. In order to convert a unit quaternion $q = [q_0\ q_1\ q_2\ q_3]$ to a rotation matrix R, the following relation can be expressed as [9],

$$R = \begin{bmatrix} 1-2(q_2^{\,2}-q_3^{\,2}) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^{\,2}-q_3^{\,2}) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2(q_1^{\,2}-q_2^{\,2}) \end{bmatrix}. \quad (16)$$

3.6.3 Axis Angle to Quaternion

Given an angle α in degrees and an axis of rotation $u = [u_1\ u_2\ u_3]$, the corresponding unit quaternion can be calculated by first normalizing the axis of rotation and then using Eqns. (9) and finally normalizing the obtained quaternion [9], is

$$\hat{u} = \frac{u}{|u|},$$

$$q = \begin{bmatrix} \cos(\theta/2) & \sin(\theta/2)\hat{u} \end{bmatrix}, \text{ and} \quad (17)$$

$$q = \frac{q}{\|q\|}.$$

3.6.4 Quaternion to Axis Angle

It is desirable to have a rotation described as an angle about an axis when displaying the numerical simulation results in a three dimensional animation. The quaternions obtained

from the numerical integration can be converted to Axis Angle format by normalizing the quaternions and applying the following equations [9],

$$\theta = 2\cos^{-1}(q_0) \text{ and} \tag{18}$$

$$u = q_v / \sqrt{1 - q_0^2} \,. \tag{19}$$

Care should be taken to avoid the singularity at θ = 0 where u becomes undefined. This can be easily handled as when the angle is zero any arbitrary axes can be assigned without affecting the outcome.

CHAPTER 4

MODEL DYNAMICS

4.1 Gravity Model Implementation

For accurate representation of the Earth's gravitational field, a spherical harmonic gravity model of order 40 was developed in Matlab/Simulink using a recursive, non-singular algorithm as presented in [6]. The gravitational potential was calculated as presented in Eqn. (2) reproduced below,

$$U(\mathrm{r},\phi,\lambda) = \frac{\mu}{r} + \sum_{n=2}^{\infty}\sum_{m=0}^{n}\frac{\mu}{r}\left(\frac{R_e}{r}\right)^n P_{n,m}(\varepsilon)(\mathrm{C}_{n,m}\cos(\mathrm{m}\,\lambda) + \mathrm{S}_{n,m}\sin(m\lambda)).$$

C_{nm} and S_{nm} are the un-normalized cosine and sine gravity coefficients resulting from the distribution of mass throughout the Earth. Normalized gravity coefficients from EGM2008 were used to calculate the un-normalized coefficients using the relation given as,

$$C_{n,m} = N(n,m)\overline{C}_{n,m} \quad \text{and} \quad S_{n,m} = N(n,m)\overline{S}_{n,m}, \tag{20}$$

where

$$N(n,m) = \left(\frac{(n-m)!(2n+1)(2-\delta_{om})}{(n+m)!}\right)^{1/2}, \tag{21}$$

and δ_{om} is 1 for m=0 and is zero otherwise.

As these coefficients are not a function of the state, they were pre-computed for faster simulations. The gravity model was used to propagate the orbit of a 4.2 kg CubeSat at an altitude of about 385 km with a 53^o inclination using numerical integration techniques with the initial condition state vector $x_0 = [r_0, v_0]$ given as x_0 = [-662.12, 4425.025, 5075.50, -6.79, -3.087, 1.81] with r_0 in kilometers and v_0 in kilometers per second. The resultant trajectory

over multiple orbits was obtained as shown in Fig. 5. The perturbations due to gravitational anomalies are evident.

An algorithm was implemented to convert the state vector array into orbital elements to better understand the effects of gravitational anomalies on the satellite and its orbit. Fig. 6 shows the perturbations in orbital elements due to gravitational anomalies.

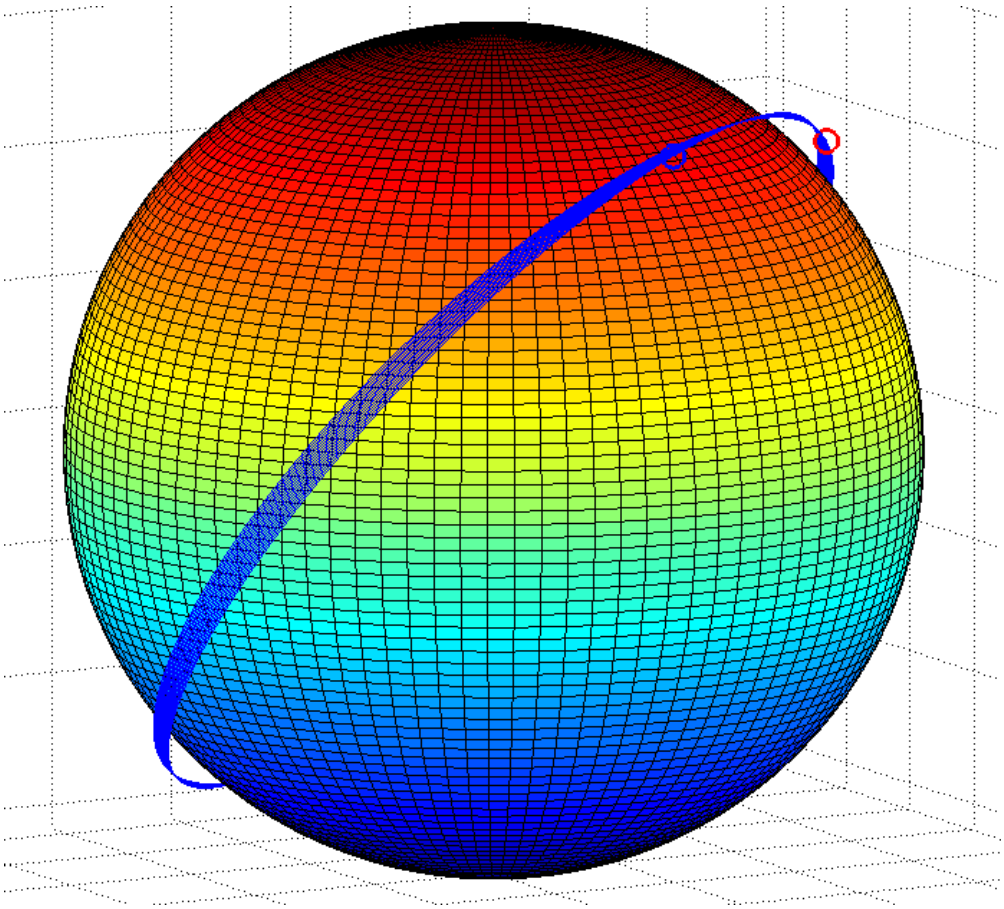

Figure 5. Orbit simulation under spherical harmonic gravity.

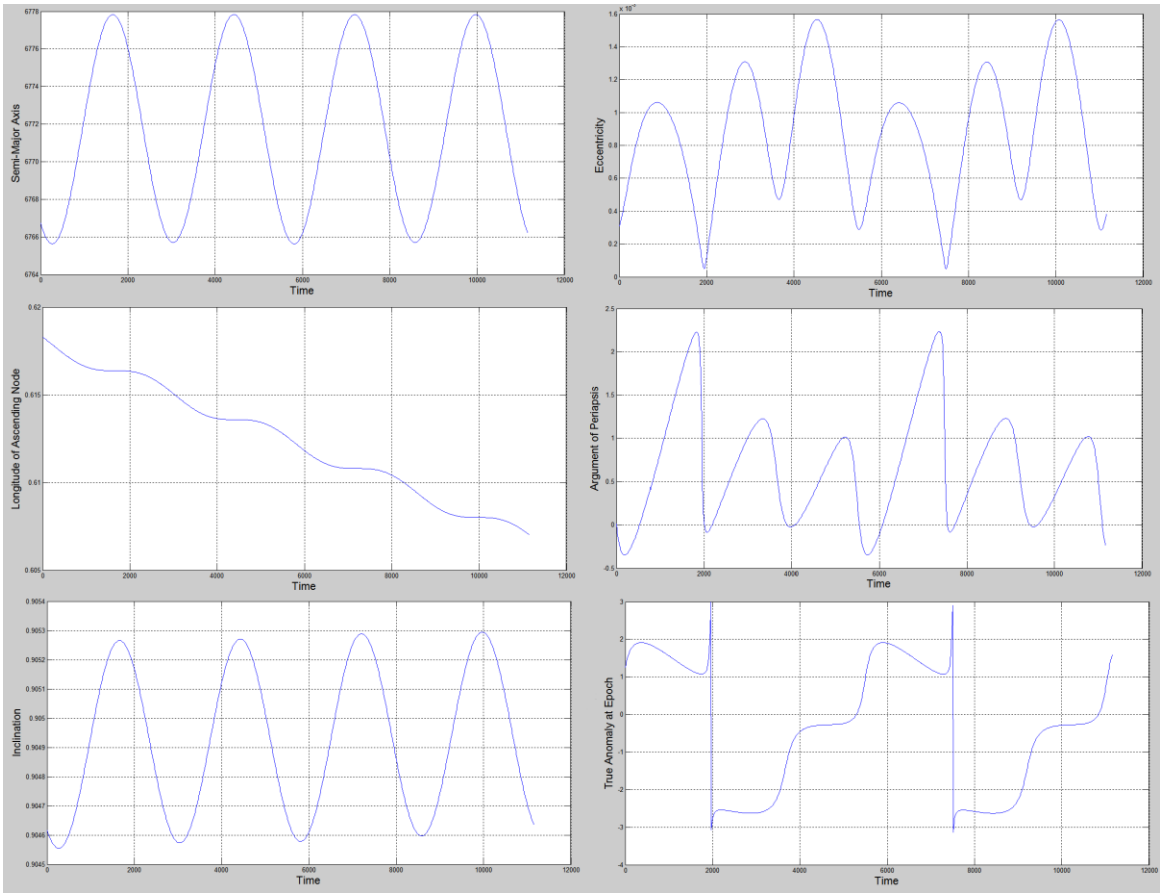

Figure 6. Perturbations in classical orbital elements due to gravitational anomalies.

4.2 CubeSat as a Rigid Body

We consider the 3U CubeSat as a rigid body with six degrees of freedom. Its mass is taken to be 4.2 kilograms and the moment of inertia matrix in the body fixed frame is J = diag (0.035 0.007 0.035). The state of a rigid body is defined by its position, orientation, linear velocity and angular velocity. The position and velocity are depended on the forces acting on the center of mass of the body while the orientation and the angular velocity are dependent on the torques acting on it.

It is advantageous to define the various relations between these mechanical quantities, in order to better understand the dynamics of the body and implement them in the mathematical model. Listed below are a few basic relations that form the basis of the simulation model

[12]. The Position represented by $r = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}$ describes the coordinates of the CubeSat's center of mass, in the ECI reference frame. Quaternion $q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}$ represents the orientation of the CubeSat body fixed frame with respect to the world coordinate axes. The linear momentum is the product of mass and linear velocity of the CubeSat given as $P = m_s v$. Whereas the angular momentum is the product of moment of inertia matrix J and the angular velocity vector $\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}$ and given as $L = J\omega$.

4.3 Thruster Module

The CubeSat considered in this thesis has a single micro-propulsion thruster along its roll axis as shown in Fig. 7. The misalignment error in the thruster module is modeled by assuming that the thruster has an angular tilt around the ideal or nominal thrust direction by small constant angles $\Delta\alpha$ and $\Delta\beta$ as shown in the fig.

In body fixed frame, the actual thrust vector can be resolved as

$$F_t = F_{ideal} + F_{error} \quad \text{and} \tag{22}$$

$$F_t = T_m \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + T_m \begin{bmatrix} \cos(\Delta\alpha)\sin(\Delta\beta) \\ 1 - \cos(\Delta\alpha)\cos(\Delta\beta) \\ \sin(\Delta\alpha) \end{bmatrix}.$$

Since the angles $\Delta\alpha$ and $\Delta\beta$ are small the following approximations can be applied:

$$\cos(\Delta\beta) = \cos(\Delta\alpha) = 1 \text{ and}$$

$$\sin(\Delta\alpha) = \Delta\alpha \text{ and } \sin(\Delta\beta) = \Delta\beta,$$

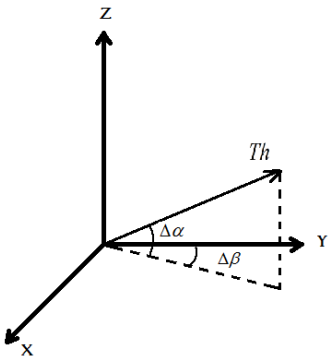

Figure 7. Thrust Vector with Error

which reduces the equation to

$$F_t = T_m \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + T_m \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta\alpha \\ \Delta\beta \end{bmatrix} \quad (23)$$

4.4 Reaction Wheels

Reaction wheels are actuators used to produce torque in order to cause a rotation of the satellite in orbit. They work on the principle of conservation of angular momentum. When the reaction wheels are rotated in one direction the satellite rotates in the opposite direction to conserve angular momentum [13]. The angular momentum of the wheels is related to its angular velocity by

$$L_w = I_w \omega_w, \quad (24)$$

where I_w is the moment of inertia of one reaction wheel and ω_w is its angular velocity. The derivative of the angular momentum is the torque generated on the satellite, or

$$\dot{L}_w = \tau_w. \quad (25)$$

Three degrees of rotational freedom can be achieved using different configurations of reaction wheels, some providing redundancy as a failsafe backup. We will however consider

the most basic configuration of three reaction wheels placed orthogonally to each other, each providing rotational capability around the roll, pitch and yaw axes.

4.5 The State Vector

We define the position vector as r(t), the velocity vector as v(t), the orientation by quaternion q(t) and the angular velocity by ω(t), in the inertial frame of reference. The state vector is defined as

$$y = \begin{bmatrix} r & q & P & L & L_w \end{bmatrix}, \tag{26}$$

where P is the linear momentum, L is the angular momentum and h_w is the angular momentum stored in the reaction wheels. The dynamics of the system can be described by the following relations of their states [12]:

$$\dot{r}(t) = v(t) = \frac{dr(t)}{dt} \quad \text{and} \tag{27}$$

$$\dot{q}(t) = \frac{1}{2}\omega(t) * q(t), \tag{28}$$

where ω(t)*q(t) is short for the quaternion multiplication between the quaternions [0 ω] and q. Then

$$\dot{P}(t) = m_s \dot{v}(t) = m_s a(t) = m_s (g + d) + F_t \ , \tag{29}$$

where P (t) is the linear momentum, m_s is the mass of the satellite, g is the acceleration due to gravity and d is the gravity perturbations. F_t is the external force applied by the thruster. Also

$$\dot{L}(t) = J\,\dot{\omega}(t) = [\omega\times](\mathrm{J}\,\omega + L_w) + \tau \ , \tag{30}$$

where L(t) is the angular momentum, J is the inertia tensor and τ is the torque applied by the reaction wheels

$$\dot{L}_w(t) = \tau,$$

where $h_w(t)$ is the angular momentum stored in the reaction wheels.

Thus the derivative of the state vector is as given below

$$\dot{y} = \begin{bmatrix} v & \dot{q} & \dot{P} & \dot{L} & \tau \end{bmatrix}. \tag{31}$$

CHAPTER 5

ATTITUDE AND TRAJECTORY CONTROL

5.1 Error Analysis

In order to observe the thrust induced error in the satellite trajectory and attitude, the dynamic satellite model was given an error thrust vector. A similar 'Trajectory Planner' system was given the exact same thrust, however without the error. The behavior of the CubeSat without any control corrections was observed over one complete orbit. The simulated errors in orbital elements are as shown below, in Fig. 8.

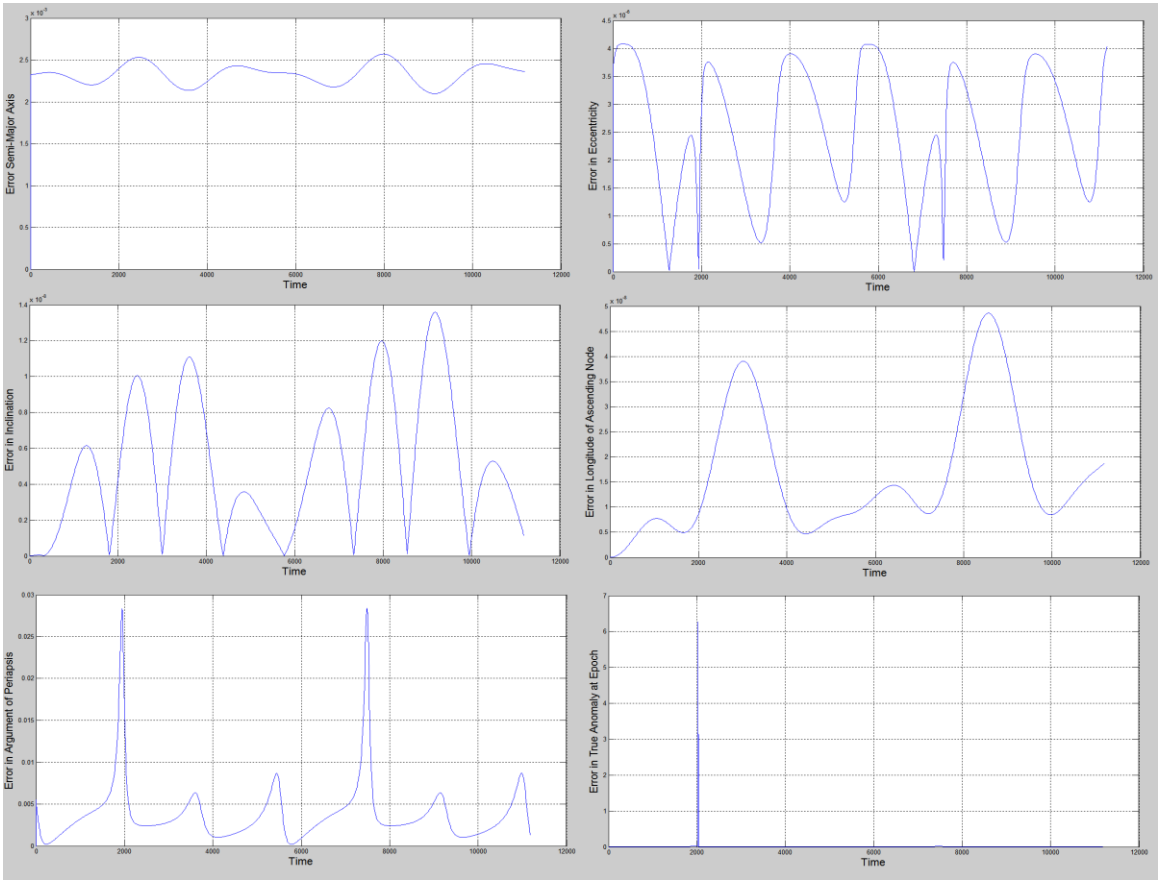

Figure 8. Errors in classical orbital elements.

The errors in trajectory and attitude at the end of one orbit were taken as the initial condition to test the control algorithm for convergence of the errors states to zero.

5.2 Quaternion Feedback Attitude Control

The CubeSat has a single thruster along the roll axis which needs precise pointing before orbital maneuvering thrusts can be applied. With the goal of acquiring the desired attitude and to correct any attitude errors a Quaternion Feedback Attitude Control is employed as presented in [14].

The rotational motion of the CubeSat can be described by Euler's equation for rigid body rotation about body fixed axes with the origin at its center of mass, which is given by,

$$J\dot{\omega} = [\omega\times](J\omega + L_w) + u \quad , \tag{32}$$

where, $[\omega\times] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$.

The quaternion kinematic differential equation is as described by Eqn. (28) as presented above and reproduced as

$$\dot{q}(t) = \frac{1}{2}\omega * q \quad .$$

Given an initial quaternion q_a and a desired quaternion q_d, the attitude error quaternion q_e is calculated as

$$\begin{bmatrix} q_{1e} \\ q_{2e} \\ q_{3e} \\ q_{4e} \end{bmatrix} = \begin{bmatrix} q_{1d} & q_{2d} & q_{3d} & q_{4d} \\ -q_{2d} & q_{1d} & q_{4d} & -q_{4d} \\ -q_{3d} & -q_{4d} & q_{1d} & q_{2d} \\ -q_{4d} & q_{3d} & -q_{2d} & q_{1d} \end{bmatrix} \begin{bmatrix} q_{1a} \\ q_{2a} \\ q_{3a} \\ q_{4a} \end{bmatrix} . \tag{33}$$

The control torque u is represented as

$$u = -[\omega\times](J\omega + L_w) - D\omega - Kq_e \,, \tag{34}$$

where D and K are 3x3 constant gain design diagonal matrices [14].

The resultant closed loop equations are as follows:

$$J\dot{\omega} = -D\omega - Kq_e \text{ and} \tag{35}$$

$$\dot{q}(t) = \frac{1}{2}\omega * q \,. \tag{36}$$

Since the body-fixed axes coincide with the principal axes of the CubeSat, the following constraints on the gain matrices K = diag (K_1,K_2,K_3) and D = diag (d_1,d_2,d_3) are sufficient to assure global asymptotic stability [14] or

$$\frac{J_2 - J_3}{K_1} + \frac{J_3 - J_1}{K_2} + \frac{J_1 - J_2}{K_3} = 0 \text{ and} \tag{37}$$

$$D = dJ \,, \tag{38}$$

where d is a positive scalar.

The equilibrium point of the closed loop system described by the following equations:

$$q_e = \begin{bmatrix} \pm 1 & 0 & 0 & 0 \end{bmatrix} \text{ and} \tag{39-a}$$

$$\omega = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} . \tag{39-b}$$

Shortest angular path of rotation can be assured by governing the sign of quaternion feedback gain by the initial sign of q_1 [14] is

$$u = -[\omega\times](J\omega + L_w) - D\omega - sign(q_{e0})Kq_e \,. \tag{40}$$

The block diagram is shown in Fig. 9.

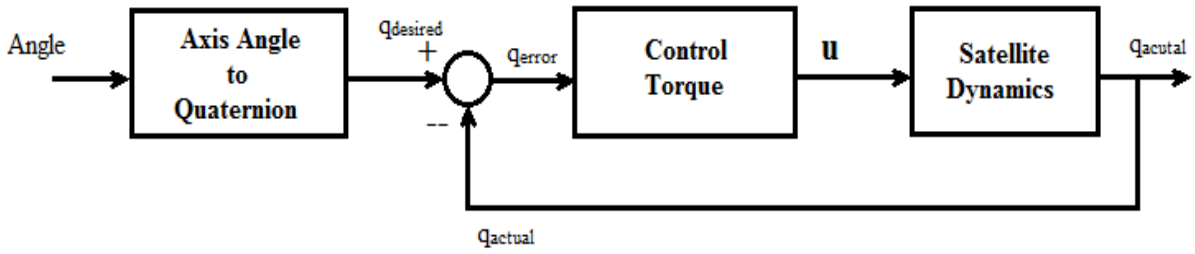

Figure 9. Quaternion Feedback Attitude Controller.

5.3 The Backstepping Control Technique

A special class of nonlinear dynamic systems can be treated as if built from subsystems, one inside the other. In such a scenario, we can stabilize the inner subsystem using some control strategy. This stable subsystem can then be used to develop controller to stabilize the next outer subsystem and so on. The process is repeated until the control law for the final external control input is derived. Hence this technique is called Backstepping Control [15].

5.4 Nonlinear State Feedback Controller Design

In this section, a control law is derived for trajectory tracking problem, using the above mentioned backstepping technique. We start with the nonlinear state equations for the satellite system,

$$\dot{x}_1 = x_2 \quad \text{and} \tag{41-a}$$

$$\dot{x}_2 = f(x_1, x_2) + \frac{u}{m_s}, \tag{41-b}$$

where x_1 is the position vector, x_2 is the velocity vector, $f(x_1, x_2)$ is the spherical harmonic gravity acceleration, m_s is the mass of the satellite while u is the control thrust input vector which is to be designed. The reference trajectory is taken from a trajectory planner, which in practical scenarios will come from an on board computer running a highly accurate mathematical model. We will call this the desired trajectory x_d with $\dot{x}_d$ and $\ddot{x}_d$ being the desired velocity and accelerations. In order to achieve trajectory tracking, the actual position and velocity should converge to the desired position and velocity. Hence, it is necessary to define error states in position and velocity which can be control to converge to the equilibrium point of zero.

We define the error states in position and velocity, respectively, as

$$z_1 = x_1 - x_d \text{ and} \tag{42-a}$$

$$z_2 = x_2 - \dot{x}_d \,. \tag{42-b}$$

We can obtain the new nonlinear system from the above equations by taking the derivative of the error states or

$$\dot{z}_1 = \dot{x}_1 - \dot{x}_d = x_2 - \dot{x}_d \text{ and}$$

$$\dot{z}_2 = \dot{x}_2 - \ddot{x}_d \,.$$

Substituting Eqn. (42-b) and Eqn. (41-b) we get the new nonlinear system as

$$\dot{z}_1 = z_2 \text{ and} \tag{43-a}$$

$$\dot{z}_2 = f(x_1, x_2) + \frac{u}{m_s} - \ddot{x}_d \tag{43-b}$$

Starting with the stabilization of the inner subsystem, we consider state z_2 as a virtual controller to the state z_1 and set is as $z_2 = \nu$.

Hence, $$\dot{z}_1 = \dot{\nu} \tag{44}$$

Setting $\nu = -\alpha_1 z_1$, where α_1 is a positive 3x3 design matrix is assures z_1 is asymptotically stable. In order to stabilize z_1, the state z_2 should track the virtual control input ν. We can consider this as using the control input u, to force z_2 to track ν by regulating the following output,

$$y = z_2 - \nu$$

Therefore, $$\therefore\ y = z_2 + \alpha_1 z_1 \tag{45}$$

By taking the derivative of the above equation

$$\dot{y} = \dot{z}_2 - \dot{\nu},$$

and substituting Eqns. (43-b) and (44) we get

$$\dot{y} = f(x_1, x_2) + \frac{u}{m_s} - \ddot{x}_d + \alpha_1 \dot{z}_1 \tag{46}$$

Now we can design the control law u as

$$u = m_s(-f + \ddot{x}_d - \alpha_1 z_2 - \alpha_2 y) \quad (47)$$

This control law makes Eqn. (46) to be

$$\dot{y} = -\alpha_2 y$$

Hence, y asymptotically approaches zero and thus z_2 approaches ν.

Substituting Eqn. (45) in Eqn. (47), we get,

$$u = m_s(-f + \ddot{x}_d - \alpha_1 z_1 - \alpha_2(z_2 + \alpha_1 z_1)) \quad (48)$$

Thus the closed loop equations of the system become

$$\dot{z}_1 = z_2 \text{ and} \quad (49\text{-a})$$

$$\dot{z}_2 = -k_1 z_1 - k_2 z_2, \quad (49\text{-b})$$

where $k_1 = \alpha_2 \alpha_1$ and $k_2 = \alpha_1 + \alpha_2$.

The block diagram is shown in Fig. 10.

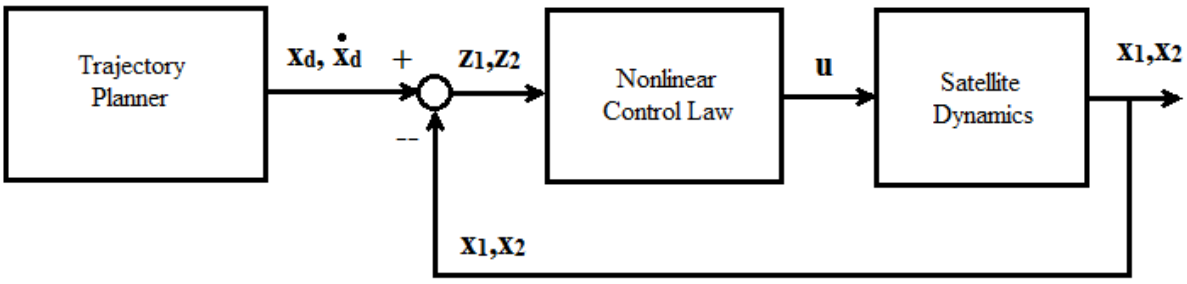

Figure 10. Nonlinear Trajectory Controller.

5.5 Stability Analysis

Lyapunov's second method is extensively used to determine global asymptotic stability of nonlinear dynamical systems. This method uses an energy like function which is globally positive definite i.e.,

$$V(z_1, z_2) > 0 \qquad \forall\, [z_1, z_2] \neq 0 \text{ and}$$

$$V(z_1, z_2) = 0 \qquad \forall\, [z_1, z_2] = 0 .$$

We then observe the time derivative, $\dot{V}(x) = \frac{dV(x)}{dt}$ of this energy function, to see if the energy of the function is continuously decreasing to zero which implies asymptotic stability in the sense of Lyapunov [16]. Global asymptotic stability demands one more condition of radial unboundedness to be satisfied by the energy function. This condition is given as

$$V(z_1, z_2) \to \infty \text{ as } z_1 \to \infty \text{ or } z_2 \to \infty .$$

In order to assure asymptotic stability, the function $\dot{V}(x)$ should either be globally negative definite as

$$\dot{V}(z_1, z_2) < 0 \qquad \forall\, [z_1, z_2] \neq 0 \text{ and}$$

$$\dot{V}(z_1, z_2) = 0 \qquad \forall\, [z_1, z_2] = 0 .$$

Or, it should be globally negative semi-definite such that

$$\dot{V}(z_1, z_2) \le 0 \qquad \forall\, [z_1, z_2] ,$$

in addition to satisfying the LaSalle's condition stated as follows.

LaSalle's Theorem: If a system is asymptotically stable and the set of all points (z_1, z_2) such that $\dot{V}(x) = 0$ contains no other trajectory other than the trivial trajectory of (z_1, z_2) = (0, 0), then that system is globally asymptotically stable. [16]

For the considered closed loop system Eqns. (49)

$$\dot{z}_1 = z_2 \text{ and}$$

$$\dot{z}_2 = -k_1 z_1 - k_2 z_2 ,$$

we take the energy function candidate as

$$V(z_1, z_2) = \frac{1}{2} z_1^{\,T} k_1 z_1 + \frac{1}{2} z_2^{\,T} z_2 . \qquad (50)$$

Taking the derivative we get

$$\dot{V}(z_1, z_2) = z_1^{\,T} k_1 \dot{z}_1 + z_2^{\,T} \dot{z}_2 .$$

Using Eqns. (49-a) and (49-b) the above equation becomes

$$\dot{V}(z_1, z_2) = z_1^T k_1 z_2 + z_2^T (-k_1 z_1 - k_2 z_2) ,$$

$$\dot{V}(z_1, z_2) = z_1^T k_1 z_2 - z_2^T k_1 z_1 - z_2^T k_2 z_2 , \text{ thus}$$

$$\dot{V}(z_1, z_2) = -z_2^T k_2 z_2 . \quad (51)$$

As k_2 is a constant positive 3x3 diagonal matrix, the above function is negative semi-definite matrix. This assures asymptotic stability. Applying LaSalle's theorem by setting $\dot{V}(z_1, z_2) = 0$, implies $z_2 = 0$ hence $\dot{z}_2 = 0$. Using Eqn. (49-b), we get $z_1 = 0$. Thus, $\dot{V}(z_1, z_2) = 0$ contains no other trajectory except the trivial trajectory (0, 0). Hence, the designed control law provides global asymptotic stability.

5.6 The Pointing Problem

The above presented nonlinear control strategy gives a control thrust vector in the world frame. Since the CubeSat model under consideration has a single fixed thruster, it is necessary to align the thruster to the control vector. This issue is commonly known as the pointing problem. Given two vectors b and d in three dimensional Euclidean space, although there are multiple orthogonal rotational matrices that transform the vector b into vector d, there is only one rotation matrix with minimal required transformation, derived using a single rotation about a fixed axis (Euler axis) by the smallest possible angle [17]. The basic idea is to find the angle ϕ between the two vectors b and d, then find the normal to the plane defined by the two vectors. We can then rotate vector b about this normal, by an angle $-\phi$ to transform it to vector d.

The Euler axis ($\hat{\Phi}$) can be found using the formula given as

$$\hat{\Phi} = \frac{b \times d}{|b \times d|} . \tag{52}$$

Whereas the angle is found using the equation

$$\phi = \sin^{-1}\left[\frac{|\mathrm{b} \times \mathrm{d}|}{|b| \cdot |d|}\right]. \tag{53}$$

Using this Euler angle and Euler axis found above, we can generate a torque and apply it to the satellite using the attitude controller to align its thruster with the control thrust vector. Here, it is assumed that the time delay in rotation is negligible.

CHAPTER 6

NUMERICAL SIMULATIONS

In order to test the attitude control, various attitude changes were simulated for the satellite. For this thesis, we simulate a 180 degree rotation about an arbitrary axis u = [1 1 1]. The attitude controller normalizes the rotation axis and obtains a desired quaternion from the axis angle. It then calculates the change in quaternion based on the current attitude. The control torque shown in Fig. 11 is applied through reaction wheels in order to achieve the resultant attitude.

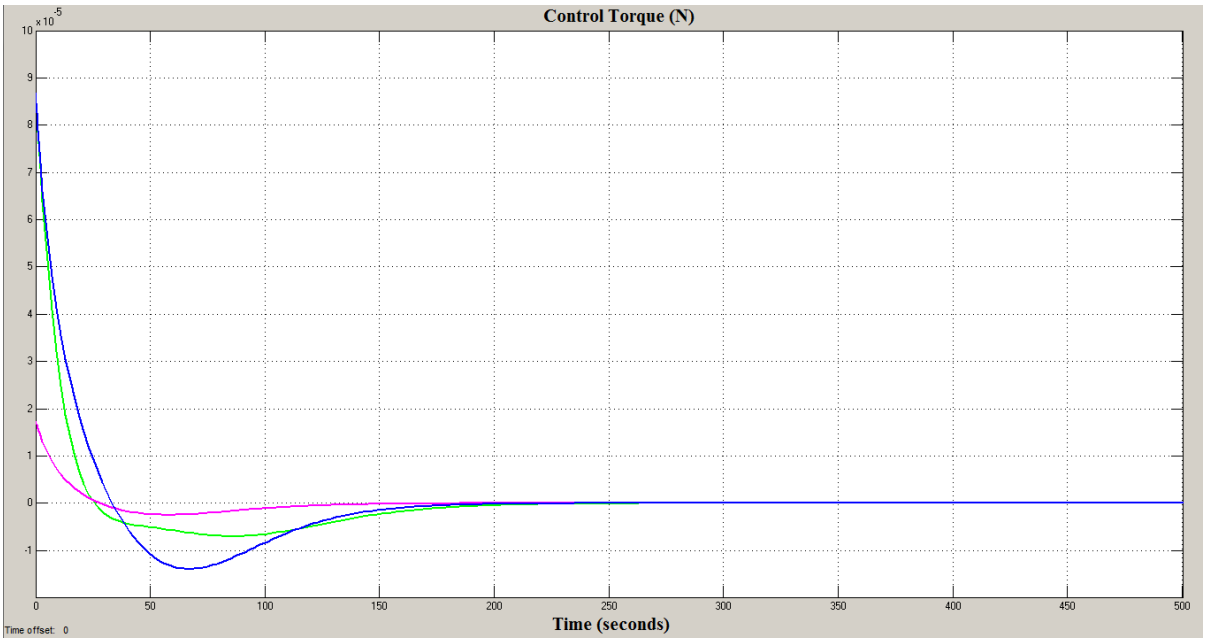

Figure 11. Control Torque for Attitude Correction.

For trajectory control test, we simulate a six degrees of freedom CubeSat model, in spherical harmonic gravity of the order 40. At t = 0, we apply an erroneous thrust vector and observe the trajectory error over one complete orbit. We then take this error as the initial condition and with a trajectory planner as reference for the desired trajectory, perform orbit corrections using nonlinear state feedback controller. The mass of the CubeSat is taken as 4.2 kilograms, with its center of mass at the geometrical center. The inertia matrix in body fixed

frame is taken as $J = \begin{bmatrix} 0.035 & 0 & 0 \\ 0 & 0.007 & 0 \\ 0 & 0 & 0.035 \end{bmatrix}$. The thrust given was a 151 mN thrust for duration of 0.1 seconds, with an angular error of $[\alpha, \beta] = [1.5^{\circ}, 1.5^{\circ}]$.

The error state vector after one complete orbit was found and used as the initial condition. The inputs to the trajectory controller were the error states between the actual position and velocity and the desired position and velocity, the latter given by the trajectory planner. The trajectory planner is similar error free two body simulator which provides the desired position (x_d) and velocity ($\dot{x}_d$). As can be seen from the graphs below in Figs. 11, 12 the position and velocity error states (z_1 and z_2) converge to zero.

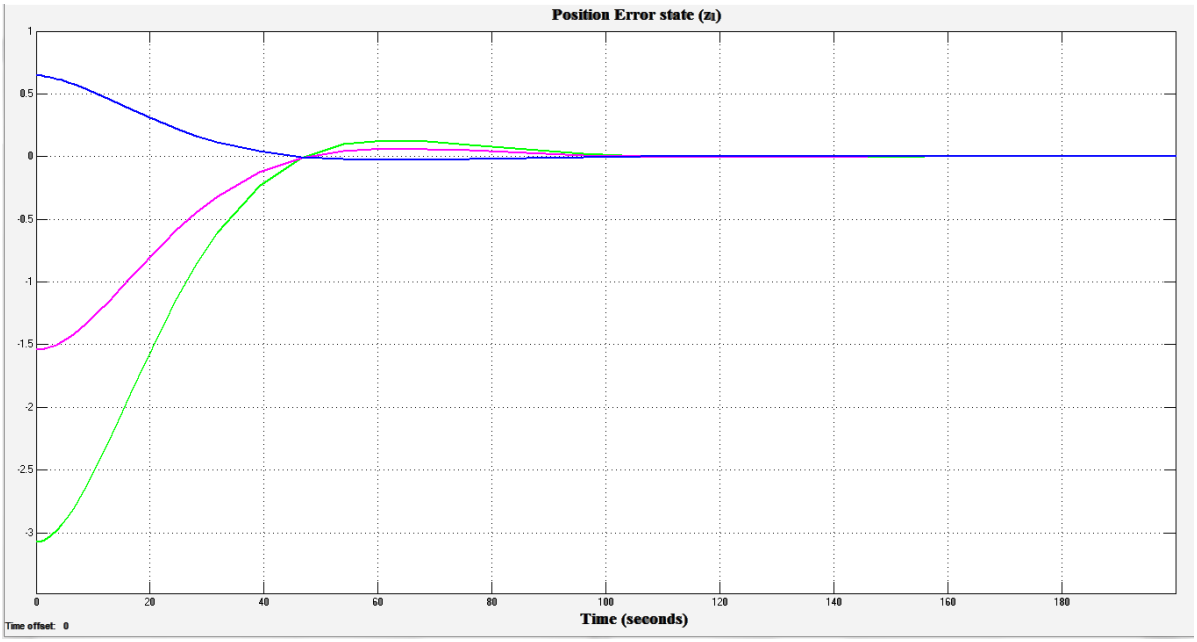

Figure 12. Position Error state (z_1).

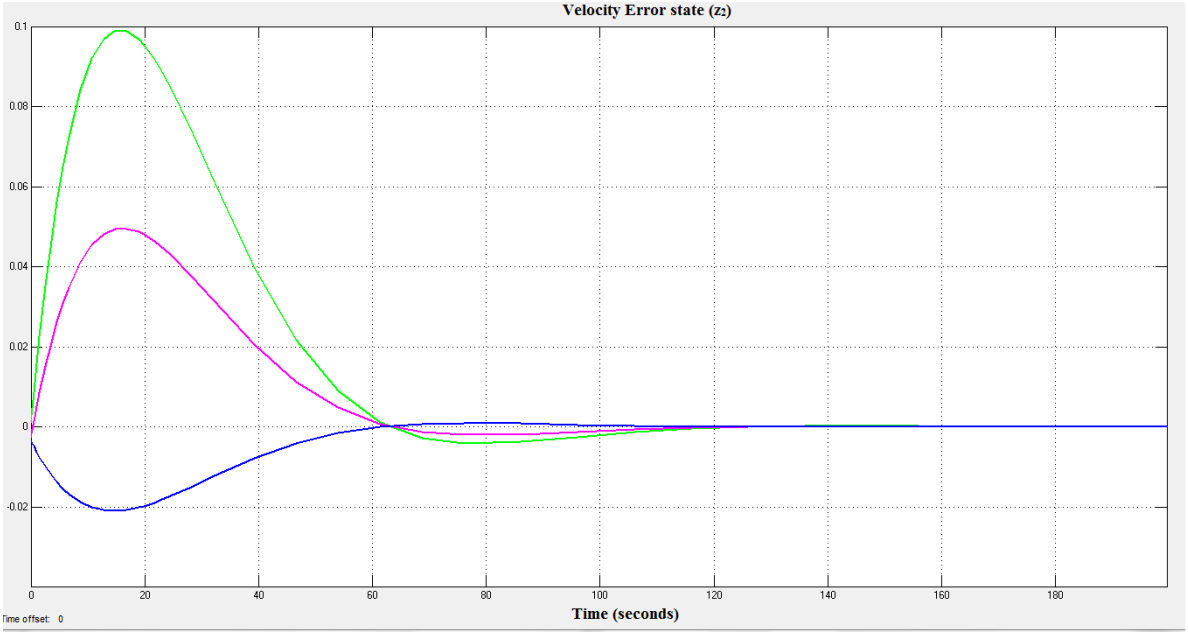

Figure 13. Velocity Error state (z_2).

The control thrust vector is as shown in graph below.

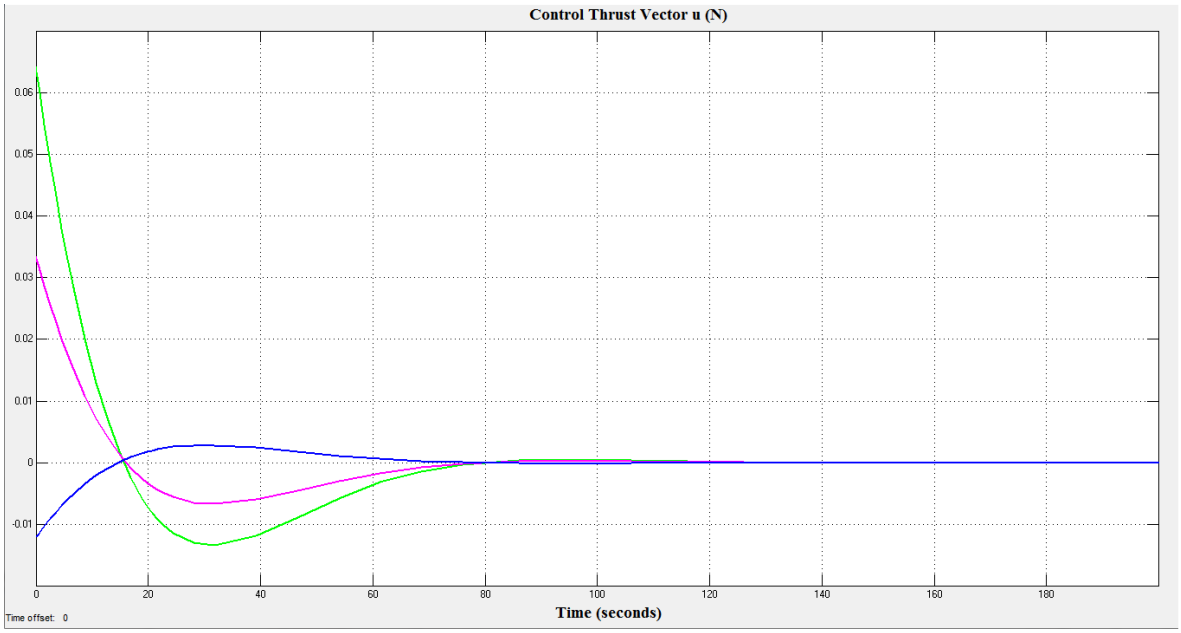

Figure 14. Control Thrust Vector (u) in Newtons.

CHAPTER 7

CONCLUSION

A six degree of freedom CubeSat model was realized using MATLAB/Simulink. Spherical harmonic gravity was used to implement a two body problem. Erroneous thrust vector was applied to the CubeSat and the deviation of the actual trajectory from the desired trajectory was observed over one orbit. The propagated error was used as input to a nonlinear state feedback control system to apply correctional thrusts. Stability analysis was done using Lyapunov techniques and LaSalle's theorem to prove global asymptotic convergence. This controller can be used in various CubeSat applications.

In the future work, the model can be made robust for disturbance rejection, for disturbance sources like atmospheric drag, gravity gradient, solar radiation winds, etc. The control thrust can be modified to accommodate impulsive thrusters as opposed to continuous thrust.

REFERENCES

[1] Jordi Puig-Suari, Ryan Nugent, Roland Coelho, *The CubeSat: The Picosatellite Standard for Research and Education* (2008).

[2] Simon Lee, Armen Toorian, Wenschel Lan, Riki Munakata, 'CubeSat Design Specifications' (2009).

[3] Francois Martel, Louis Perna, Paulo Lozano, *Miniature Ion Electrospray Thrusters and Performance Tests on CubeSats* (2012).

[4] Peter Vallis Shaw, *Pulsed Plasma Thrusters for Small Satellites* (2011).

[5] Steven Arestie, E. Glenn Lightsey, Brian Hudson, *Development of a Modular, Cold gas Propulsion system for Small Satellite Applications* (2012).

[6] G Gottlieb, Fast Gravity, Gravity Partials, *Normalized Gravity, Gravity Gradient Torque and Magnetic Field: Derivation, Code and Data* (1993).

[7] U.S. National Geospatial-Intelligence Agency (NGA), http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm2008/

[8] Roger Bate, Donald Mueller, Jerry White, Fundamentals of Astrodynamics (1971).

[9] James Diebel, *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors* (2006).

[10] Erik Dam, Martin Koch, Martin Lillholm, "Quaternions, Interpolation and Animation" (1998).

[11] Ashish Tewari, Atmospheric and Space Flight Dynamics (2007).

[12] David Baraff, 'An Introduction to Physically based Modeling' (2001).

[13] Henri Christian Kjellberg, *Design of a CubeSat Guidance, Navigation, and Control Module* (2011).

[14] B. Wie, H. Weiss, A. Arapostathis, *Quaternion Feedback Regulator for Spacecraft Eigenaxis Rotations* (1989).

[15] Wiktor Bolek, Jerzy Sasiadek, *Singularity of Backstepping Control for Nonlinear Systems* (2002).

[16] Seyed Kamaleddin, Yadavar Nikravesh, Nonlinear Systems Stability Analysis Lyapunov-Based Approach (2013).

[17] Itzhack Y. Bar-Itzhack, Daniel Hershkowitz, Leiba Rodman, *Pointing in Real Euclidean Space* (1997).

[18] Free source image Wikimedia Commons, "Classical Orbital Elements", (2007) http://commons.wikimedia.org/wiki/File:Orbital_elements.svg