SSC15-VIII-4

An Inverse Dynamics Satellite Attitude Determination and Control System with Autonomous Calibration

Sanny R. Omar[1]
Faculty Advisors: Dr. David Beale[2] and Dr. JM Wersinger[3]
Auburn University, Auburn, Alabama, 36849

Attitude determination and control systems (ADACS) are responsible for establishing desired satellite orientations. Proper satellite orientation is necessary for many science instruments and communication systems. Popular sensors include magnetometers, sun sensors, and rate gyros, and popular actuators include reaction wheels and magnetorquers. This paper investigates an ADACS design using these sensors and actuators that could feasibly be implemented on a CubeSat. The B-Dot law is used for satellite de-tumbling, and a linear inverse dynamics PD controller is utilized for steady state pointing, allowing for the analytical estimation of optimal controller gains. The inverse dynamics controller calculates desired satellite angular accelerations and then calculates the torques required to achieve these angular accelerations. This makes controller performance independent of initial conditions or system inertia properties. This system uses the magnetorquers to dump reaction wheel momentum and analyzes the satellite's kinematic response to applied torques in order to calibrate the rate gyros and estimate system moments of inertia. Simulation results corresponded well to the analytical predictions. Often, an oscillating equilibrium would occur when controller gains were low, but this oscillation could be mitigated by selecting large controller gains such that the system was heavily overdamped and scaling down large commanded angular acceleration values to within system capabilities.

Nomenclature

A	=	Area of magnetorquer
$\vec{B}$	=	Magnetic field vector
$\vec{H}$	=	Angular momentum vector
I	=	Moment of inertia, Electrical current
k	=	Constant value
k_d, k_p	=	Derivative and proportional controller gains
$\hat{n}$	=	Rotation vector of quaternion
N	=	Number of turns in magnetorquer coil
q_d, q_m, q_e	=	Desired, measured, and error quaternions
s	=	Laplace transform variable
$\vec{s}$	=	Sun vector
T	=	Torque
t	=	Time
w	=	Width of PSD detector area
α	=	Angular acceleration
θ	=	Angle
μ	=	Magnetic moment
ω	=	Angular velocity

[1] Undergraduate Student, Auburn University Department of Aerospace Engineering, 1513 Robertson Dr., Crestwood, KY 40014

[2] Professor, Auburn University Department of Mechanical Engineering, 3418C Wiggins Hall, Auburn University

[3] Professor Emeritus, Auburn University Physics Department, 206 Allison Lab, Auburn University

I. Introduction

Attitude Determination and Control Systems (ADACS) are important components of many modern satellites and enable these spacecraft to maintain desired orientations in space. This is important for the functioning of many science instruments and communication systems. The advent of the CubeSat form factor (10 x 10 x (10-30) cm cubic satellites) has demanded cheaper and smaller yet more effective technologies, including ADACS systems, than have been used in the past. This study details the design of such an ADACS, including explanations of the attitude determination and attitude control algorithms. Also discussed is a means of calibrating rate gyros and estimating satellite inertia properties in space. An ADACS system based on these design principles will be implemented on the AubieSat II CubeSat.

This ADACS will consist of reaction wheels and magnetorquers as actuators and magnetometers, rate gyros, and position sensitive detectors (PSDs) as sensors. In order to de-tumble the satellite, the ADACS will implement the "B-Dot" de-tumble control law using only the magnetometer and magnetorquers. For steady state pointing, the ADACS will first extract a magnetic field vector and a sun vector in the body-fixed reference frame from the magnetometer and PSDs respectively. The components of these vectors in the orbital frame (origin on satellite center of mass, *z-axis* toward Earth, and *x-axis* along the velocity vector) will then be determined by the satellite based on known satellite position and the IGRF magnetic field lookup tables and an Earth-Sun orbit propagator. Running the Triad algorithm with these vectors will provide a direction cosine matrix and a quaternion relating the body and orbital frames. Additionally, satellite angular velocities and moments of inertia will be estimated based on the satellite's kinematic response to torques applied by the magnetorquers or reactions wheels. These angular velocity estimates will be used, in addition to changes in the measured quaternion over time, to calibrate the rate gyros which can then be utilized to calibrate the other sensors.

From the measured quaternion and a desired quaternion (relating the orbital and body frames) supplied by the satellite operator or onboard computer, an error quaternion will be calculated relating the current orientation to the desired orientation. Components of this error quaternion along with satellite angular velocity values (about the body axes) will be plugged into a PD control algorithm to determine the satellite angular accelerations required to slew to a correct attitude. This PD control algorithm is easily linearized, allowing for a preliminary analytical stability analysis and a characterization of system behavior for various controller gains. Based on a dynamic model of the satellite, the reaction wheel torques required to achieve these angular accelerations will be determined and the proper currents will be sent to the motors. The magnetorquers will constantly act to remove residual angular momentum from the system to prevent wheel saturation. By calculating desired angular accelerations with the PD algorithm instead of calculating desired motor torques directly as has been done in the past, the accuracy and stability of the control system can be greatly improved.

II. Existing Literature and Expected Contributions

Attitude determination and control systems have been designed and built since the early days of the space age. Thus, many books[6,12,13,15] and research papers have been written on the topics of spacecraft attitude determination and control. Due to the advancement and miniaturization of technology, an increasing number of satellites, including small ones, contain ADACS systems[4,5,7,11]. The B-Dot de-tumble law is commonly used to reduce initial satellite angular velocity[3]. The TRIAD algorithm is also common for complete attitude determination when the components of two vectors are known in two reference frames[9]. Quaternions are commonly used in feedback loops for satellite attitude control systems[7,10,11,13] as they do not exhibit the gimbal lock phenomenon that can result with Euler angles. PD and PID controllers are among the most common for satellite attitude control[12,13]. Other control algorithms have been theorized, but many would be too difficult or risky to implement in reality.

Perhaps the greatest contribution of this work is the introduction of what is called an inverse dynamics control algorithm. This control algorithm uses a linear PD controller to calculate the desired satellite angular acceleration vector based on an error quaternion and satellite angular velocity vector. The equations of rotational motion along with wheel and satellite angular velocities and moments of inertia are analyzed to calculate the reaction wheel motor torques required to achieve these angular accelerations. This is in contrast to legacy quaternion feedback PD control algorithms[13] where motor torques are calculated directly by the PD algorithm. In the legacy algorithms, neither wheel angular velocities nor the couplings between the rotational degrees of freedom are taken into account. This causes such controllers to behave differently, and sometimes undesirably, when wheel or satellite angular momentum is large. Because the inverse dynamics controller analyzes the equations of rotational motion, the commanded angular accelerations are achieved regardless of satellite and wheel moments of inertia, stored angular momentum, or initial conditions. This ensures predictable controller behavior in any situation and minimizes the need for extensive simulation to determine ranges of system stability. Furthermore, because the PD control algorithm calculates angular accelerations instead of torques, only one proportional gain and one derivative gain are needed and these gains will remain constant regardless of system inertia properties. Because the PD controller itself is linear, stable gain values can be estimated analytically. This greatly reduces simulation requirements and is a significant improvement over legacy PD control algorithms where six gain values are required that are all dependent on the moments of inertia of the specific system at hand.

Additionally, a means of actively reducing reaction wheel angular momentum using magnetorquers is discussed, along with a way of accurately estimating system moments of inertia based on kinematic responses to applied torques. This paper also provides a way to estimate satellite angular velocities based on these kinematic responses and the coupling between the equations of rotational motion about each axis. These estimates can be used to calibrate the rate gyros for bias error.

III. De-Tumble

When the satellite is initially deployed or if control is temporarily lost, the satellite will be in a state of tumble (random angular velocity about the body axes). In this case, the B-Dot controller can be utilized to de-tumble the satellite and reduce the total angular momentum. B-Dot is an algorithm that has been used on many satellite missions and whose stability has been demonstrated[3]. The B-Dot control law involves only the magnetorquers and magnetometer and does not require complete knowledge of satellite attitude. Running B-Dot will serve to reduce the angular velocity of the satellite (in the body frame), but will not cause the satellite to point in any specific direction. The B-Dot controller does this by applying a torque to reduce the magnitude of the satellite's angular velocity. The desired magnetic moment $\left(\mu_{id}\right)$ for the B-Dot algorithm is illustrated in Fig. 1 and is given by the equation

$$\vec{\mu}_{tot} = k*(\vec{B}\times\vec{\omega}) = k*\dot{\vec{B}} \approx k\frac{(\vec{B}_2-\vec{B}_1)}{\Delta t} \quad (1)$$

where k is a constant negative gain value, $\vec{\omega}$ is the spacecraft angular velocity about the body axes, $\vec{B}$ is measured magnetic field value (in the body axes), $\dot{\vec{B}}$ is the time rate of change of the magnetic field, and t is time. The total magnetic torque generated by the magnetorquers $(\vec{T})$ given this magnetic moment is calculated by

$$\vec{T} = \vec{\mu}_{tot}\times\vec{B} \quad (2)$$

where the $\vec{\mu}$ generated by each magnetorquer is given by

$$\vec{\mu} = IAN(\hat{n}) \quad (3)$$

where I is the applied current, A is the area of the magnetorquer, N is the number of turns of wire in the coil, and $\hat{n}$ is the unit vector normal to the magnetorquer face given by the right hand rule based on the direction of the current in the magnetorquer. This torque works to reduce the magnitude of the satellite's angular velocity. Based on the value of $\vec{\mu}_{tot}$, the currents that must be applied to each magnetorquers can be calculated.

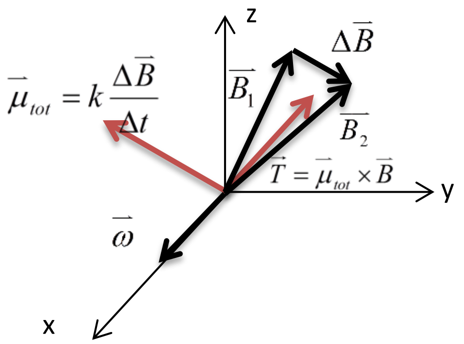
z
$\Delta\vec{B}$
$\vec{B}_1$
$\vec{\mu}_{tot} = k\frac{\Delta\vec{B}}{\Delta t}$
$\vec{B}_2$
$\vec{T} = \vec{\mu}_{tot}\times\vec{B}$
y
$\vec{\omega}$
x

Figure 1. Torque and Magnetic Moment Vectors Associated with the B-Dot De-Tumble Law

IV. Complete Attitude Determination

After de-tumble, the attitude of the satellite must be determined completely if three-axis attitude control is desired. Complete attitude determination is defined as the calculation of a direction cosine matrix relating the satellite body-fixed reference frame to the orbital reference frame (origin on satellite center of mass, *x-axis* parallel to velocity, and *z-axis* towards

Earth's center). Deriving this matrix requires knowledge of the components of two linearly independent vectors in each frame. The sun vector $(\vec{s})$ and magnetic field vector $(\vec{B})$ are two such vectors. $\vec{B}$ can be determined in the orbital frame from the International Geomagnetic Reference Field (IGRF) lookup tables based on the satellite's position relative to Earth. $\vec{s}$ can be determined in the orbital frame based on known satellite position relative to Earth and the position of the Earth relative to the sun. The $\vec{B}$ vector in the body frame can be measured directly from a standard 3-axis magnetometer onboard the satellite. $\vec{s}$ in the body frame can be determined from position sensitive detectors (PSDs). When the satellite is in eclipse and $\vec{s}$ cannot be calculated, the rate gyros can be utilized to provide a second vector for attitude determination by propagating satellite attitude.

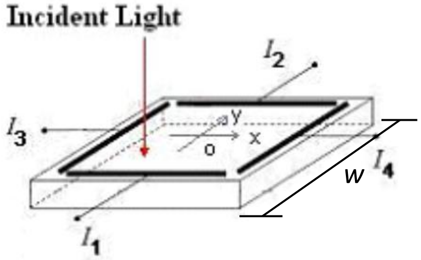

Figure 2. Position Sensitive Detector (PSD) Diagram[2]

A PSD (Fig. 2) provides the x and y coordinates of a point of light shining on the PSD. Four currents (I_1, I_2, I_3, I_4) corresponding to the location of the light spot on the detector face are produced by the PSD[1]. The Cartesian position of the light spot relative to the PSD coordinate system (x_p, y_p) can be calculated based on these current values by the equations

$$x_p = \left(\frac{w}{2}\right)\frac{I_4 - I_3}{I_4 + I_3} \tag{4}$$

$$y_p = \left(\frac{w}{2}\right)\frac{I_2 - I_1}{I_2 + I_1} \tag{5}$$

where w is the width of the square PSD. By directing light onto the PSD through a pinhole and measuring the resulting x and y coordinates of the light spot, the orientation of the sun vector relative to the plane of the PSD can be determined. If the location of the pinhole in the PSD frame is given by (x_h, y_h, z_h), the sun vector (relative to the PSD) can be calculated as

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}_{psd} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_h - \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}_s \tag{6}$$

The sun vector can then be determined in the satellite body frame by multiplying the sun vector in the PSD frame by the direction cosine matrix relating the PSD and body frames.

After both vectors are determined in the orbital and body frames, the TRIAD algorithm can be utilized to determine a direction cosine matrix relating the frames. The triad algorithm hinges on the relation

$$\left[\vec{R}_1 \quad \vec{R}_2 \quad (\vec{R}_1 \times \vec{R}_2)\right] = [A]\left[\vec{r}_1 \quad \vec{r}_2 \quad (\vec{r}_1 \times \vec{r}_2)\right] \tag{7}$$

where the $\vec{R}$ and $\vec{r}$ are column vectors in each of the two frames and [A] is the direction cosine matrix relating the two frames. If we substitute $\hat{s}$ in the orbital and body frames for $\vec{R}_1$ and $\vec{r}_1$ respectively and $\hat{B}$ for $\vec{R}_2$ and $\vec{r}_2$, we can solve for [A] by multiplying both sides of Eq. (7) by the inverse of the matrix on the right hand side of the equation. The calculation of [A] constitutes full attitude determination. A quaternion relating the body and orbital frames (q_m) can extracted from [A] for use in the attitude control algorithm. Quaternions are based on Euler's theorem which states that a coordinate frame can be transformed into any desired orientation by a rotation of angle θ about a vector $\hat{n}$ as shown in Figure 3.

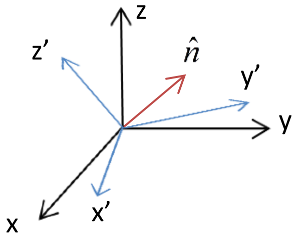

Figure 3. Coordinate Frame Rotation about Vector $\hat{n}$

A quaternion is defined based on $\hat{n}$ and θ as

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ n_x \sin(\theta/2) \\ n_y \sin(\theta/2) \\ n_z \sin(\theta/2) \end{bmatrix} \tag{8}$$

If a direction cosine matrix [A] is defined as follows

$$\begin{bmatrix} r_{x2} \\ r_{y2} \\ r_{z2} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} r_{x1} \\ r_{y1} \\ r_{z1} \end{bmatrix} \tag{9}$$

$$\left[\vec{r}_2\right] = [A]\left[\vec{r}_1\right]$$

The quaternion associated with [*A*] can be calculated using Eq. (10).

$$\begin{aligned} q_0{}^2 &= \frac{tr(A)+1}{4} = \frac{a_{11}+a_{22}+a_{33}+1}{4} \\ q_1 &= \frac{a_{32}-a_{23}}{4q_0} \\ q_2 &= \frac{a_{13}-a_{31}}{4q_0} \\ q_3 &= \frac{a_{21}-a_{12}}{4q_0} \end{aligned} \tag{10}$$

V. 3-Axis Attitude Control

A. PD Inverse Dynamics Control Algorithm

The attitude control algorithm is dependent on three parameters: A measured quaternion (q_m) relating the body frame in the current orientation to the orbital frame, a desired quaternion (q_d) relating the body frame in the desired orientation to the orbital frame, and the spacecraft angular velocity vector about the body axes $(\overrightarrow{\omega_{sc}})$. Quaternions are preferable to a direction cosine matrix for the attitude control algorithm for several reasons. A quaternion not only contains four terms instead of nine but also specifies the line of rotation necessary to get from one frame to another, a valuable parameter for attitude control. Quaternions are also preferable to Euler angles because they do not exhibit the mathematical singularities or discontinuities such as gimbal lock that plague Euler angles.

The first step in the attitude control algorithm is to calculate the error quaternion (q_e) relating the satellite orientations specified by q_m and q_d. q_e quantifies the rotation the satellite must make in order for it to get from its current orientation to the desired orientation. q_e is calculated by the following equation

$$q_e = L_d q_m \tag{11}$$

where

$$L_d = \begin{bmatrix} q_{0_d} & q_{1_d} & q_{2_d} & q_{3_d} \\ -q_{1_d} & q_{0_d} & q_{3_d} & -q_{2_d} \\ -q_{2_d} & -q_{3_d} & q_{0_d} & q_{1_d} \\ -q_{3_d} & q_{2_d} & -q_{1_d} & q_{0_d} \end{bmatrix} \tag{12}$$

The angular velocity vector can be extracted from the change in the measured quaternion over time using a similar procedure. If q_1 is the measured quaternion at time t_1 and q_2 is the measured quaternion at time t_2, then the quaternion q_{12} that gives the rotation from q_1 to q_2 can be calculated by the equation

$$q_{12} = L_2 q_1$$

$$L_2 = \begin{bmatrix} q_{0_2} & q_{1_2} & q_{2_2} & q_{3_2} \\ -q_{1_2} & q_{0_2} & q_{3_2} & -q_{2_2} \\ -q_{2_2} & -q_{3_2} & q_{0_2} & q_{1_2} \\ -q_{3_2} & q_{2_2} & -q_{1_2} & q_{0_2} \end{bmatrix} \tag{13}$$

The angle of rotation associated with q_{12} can be calculated based on the first quaternion.

$$\theta = 2\cos^{-1}\left(q_{0_{12}}\right) \tag{14}$$

Dividing this angle by the time required for the satellite to go from q_1 to q_2 gives the magnitude of the angular velocity of the satellite, provided that θ is small.

$$\omega_{sc} = \frac{\theta}{t_2 - t_1} \tag{15}$$

The orientation of the angular velocity vector will be about the rotation vector $(\vec{n})$ associated with q_{12}. Thus, $\overrightarrow{\omega_{sc}}$ can be calculated by the equations

$$\hat{n} = \frac{1}{\sin\left(\theta/2\right)} \begin{bmatrix} q_{1_{12}} \\ q_{2_{12}} \\ q_{3_{12}} \end{bmatrix} \tag{16}$$

$$\overrightarrow{\omega_{sc}} = \omega_{sc}\hat{n} \tag{17}$$

$\overrightarrow{\omega_{sc}}$ can also be provided by the rate gyros. Rate gyros must be calibrated before use to correct for bias errors. Rate gyros can be calibrated based on other sensors or based on the equations of rotational motion and the response of the satellite to applied torques. Once calculated, q_e and $\overrightarrow{\omega_{sc}}$ can be used with a simple, linear PD control algorithm (Eq. (18)) to calculate desired satellite angular accelerations.

$$\begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix}_{sc} = k_p \begin{bmatrix} q_{e1} \\ q_{e2} \\ q_{e3} \end{bmatrix} + k_d \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_{sc} \tag{18}$$

The *k* values in Eq. (18) are constant controller gains and can be selected based on simulations and through an analytical stability analysis. k_d will be negative (to reduce the angular velocity) and k_p will be positive to bring the satellite closer to the desired orientation. It is suboptimal to calculate the desired reaction wheel torques directly from the PD control algorithm. This is because the effects of a given set of applied torques on a satellite's rotational motion will vary depending on the angular momentum of the satellite and wheels. Thus, more precise control can be achieved by calculating the desired angular accelerations and then determining the torques required to realize these angular accelerations based on Euler's equation of rotational motion

$$\sum M = \vec{T} = \frac{d\vec{H}}{dt} + \vec{\omega} \times \vec{H} \qquad (19)$$

where $\vec{T}$ represents the applied external torques, $\vec{H}$ represents the system angular momentum, and $\vec{\omega}$ represents the angular velocity of the frame in which $\vec{H}$ is measured (the satellite body frame in this case). This type of control system is called an inverse dynamics controller. Eq. (19) essentially states that the summation of moments acting on the satellite is equal to the total change in the angular momentum of the satellite. Because no external torques are applied with a reaction wheel control system, the summation of external moments will be zero and the total angular momentum of the satellite (relative to inertia) will remain constant. The total angular momentum of a spacecraft with wheels with symmetric tops is given by the equation

$$\vec{H} = I_{sc}\overrightarrow{\omega_{sc}} + \sum_{i=1}^{3} I_w \overrightarrow{\omega_w} \qquad (20)$$

where I_{sc} and $\overrightarrow{\omega_{sc}}$ are the moments of inertia and angular velocities of the spacecraft and I_w and $\overrightarrow{\omega_w}$ are the moments of inertia and angular velocities of the wheels. Substituting Eq. (20) into Eq. (19) and setting $\vec{T} = 0$ yields

$$\begin{gathered} 0 = \frac{d}{dt}\left[I_{sc}\overrightarrow{\omega_{sc}} + \sum_{i=1}^{3} I_w \overrightarrow{\omega_w} \right] - \\ \left[I_{sc}\overrightarrow{\omega_{sc}} + \sum_{i=1}^{3} I_w \overrightarrow{\omega_w} \right] \times \overrightarrow{\omega_{sc}} \end{gathered} \qquad (21)$$

Re-arranging Eq. (21) gives the equation for the required wheel angular accelerations and motor torques

$$\begin{aligned} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}_m &= I_w \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}_w = \\ &\left(\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}_{sc} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_{sc} + I_w \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_w \right) \\ &\times \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_{sc} - \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}_{sc} \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix}_{sc} \end{aligned} \qquad (22)$$

In Eq. (22), *I* values represent moment of inertia components, *ω* values represent angular velocity components, and *α* values represent angular acceleration components. The subscript "*sc*" denotes a spacecraft property while the subscript "*w*" denotes a wheel property. An index of 1 indicates the wheel aligned with the *x* body-axis while indices of 2 and 3 refer to the wheels aligned with the *y* and *z* body axes respectively. Wheel angular velocity and angular acceleration values are given about the wheels' axes of rotation. All wheels are assumed to have the same moment of inertia (I_w) about the rotation axis. The *T* values represent the reaction wheel motor torques required to achieve the calculated wheel angular accelerations. Note that these are not externally applied torques and do not change the total angular momentum of the satellite.

B. Active Angular Momentum Dumping using Magnetorquers

The magnetorquers can be utilized during the steady state pointing operation to reduce the angular momentum stored in the reaction wheels. This will reduce the power consumption of the ADACS and will make the system last longer. By exerting a magnetic torque that is as close as possible to anti-parallel with the total wheel angular momentum vector ($\overrightarrow{H_w}$), the total wheel angular momentum can be reduced without perturbing the satellite's attitude.

$\overrightarrow{H_w}$ can be calculated by

$$\overrightarrow{H_w} = I_w \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \qquad (23)$$

Magnetorquers can only generate torques perpendicular to the magnetic field vector as evidenced by Eq. (2). Thus, to calculate the ideal magnetic moment of the magnetorquers we perform the operation

$$\vec{\mu}_{id} = k * \left(\overrightarrow{H_w} \times \vec{B} \right) \qquad (24)$$

where *k* is a scaling factor chosen to ensure that $\vec{\mu}_{id}$ does not exceed the capabilities of the magnetorquers. Producing this magnetic moment yields a magnetic torque of

$$\vec{T} = \overrightarrow{\mu_{id}} \times \vec{B} = k * \left(\overrightarrow{H_w} \times \vec{B} \right) \times \vec{B} \qquad (25)$$

This torque vector will be perpendicular to $\vec{B}$ and will be at an angle greater than 90° from $\overrightarrow{H_w}$ as shown in Figure 4.

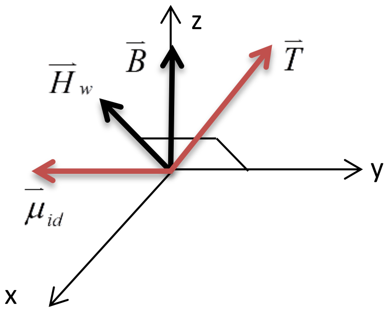

Figure 4. Magnetic Torque Vector for Active Momentum Dumping

Thus $\vec{T}$ will work to decrease the total angular momentum of the system. The satellite must, however, maintain a constant angular momentum if it is to maintain a desired attitude. The only way to maintain a constant total satellite angular momentum ($\vec{H}_s$) in the presence of $\vec{T}$ is for the wheels to decelerate. This wheel deceleration will increase $\vec{H}_s$ and decrease $\vec{H}_w$. If the change in wheel angular momentum is planned such that

$$\Delta\vec{H}_w = \vec{T}\Delta t \tag{26}$$

The total angular momentum of the satellite will remain unchanged while the total wheel angular momentum will be reduced.

C. Moment of Inertia Estimation and Rate Gyro Calibration

System moments of inertia must be known in order to properly implement the inverse dynamics control algorithm. Reaction wheel moments of inertia are generally provided by the manufacturer, but satellite moments of inertia are often difficult to calculate. Once in space, the moment of inertia of the satellite about each axis can be easily calculated after the satellite has been de-tumbled using B-Dot. If the satellite is assumed to initially be irrotational, accelerating one wheel will result in an equal and opposite change in satellite angular momentum about the wheel axis such that

$$\begin{gathered} I_s\alpha_s = -I_w\alpha_w \\ I_s = \frac{-I_w\alpha_w}{\alpha_s} \end{gathered} \tag{27}$$

Angular accelerations can be accurately measured by the rate gyros even before calibration as their measurement is unaffected by bias error. After calculating I_s about one axis, the satellite can be de-tumbled again and the moment of inertia about the other two principal axes can be calculated in a similar manner.

Once wheel and satellite moments of inertia are known, the satellite angular velocities can be determined based on the satellite's kinematic response to applied torques. These angular velocities are necessary to calibrate the rate gyros which will generally have some bias error due to thermal fluctuations, launch vibrations, and manufacturing errors. This bias error will cause the gyros' angular velocity estimates to always be off by a constant amount but will not affect angular acceleration measurements. Carrying out the matrix multiplication in Eq. (22) yields the scalar equations

$$\begin{gathered} I_{xx}\dot{\omega}_x + I_w\dot{\omega}_1 + (\mathrm{I}_{zz} - I_{yy})\omega_y\omega_z \\ +I_w(\omega_y\omega_3 - \omega_z\omega_2) = 0 \end{gathered} \tag{28}$$

$$\begin{gathered} I_y\dot{\omega}_y + I_w\dot{\omega}_2 + (\mathrm{I}_{xx} - I_{zz})\omega_z\omega_x \\ +I_w(\omega_z\omega_1 - \omega_x\omega_3) = 0 \end{gathered} \tag{29}$$

$$\begin{gathered} I_{zz}\dot{\omega}_z + I_w\dot{\omega}_3 + (\mathrm{I}_{yy} - I_{xx})\omega_x\omega_y \\ +I_w(\omega_x\omega_3 - \omega_y\omega_1) = 0 \end{gathered} \tag{30}$$

solving these equations for the *x*, *y*, and *z* components of $\overrightarrow{\omega_{sc}}$ yields

$$\omega_x = \frac{I_w\omega_y\omega_1 - I_w\dot{\omega}_3 - I_{zz}\dot{\omega}_z}{\left(I_{yy} - I_{xx}\right)\omega_y + I_w\omega_3} \tag{31}$$

$$\omega_z = \frac{I_w\omega_x\omega_3 - I_w\dot{\omega}_2 - I_{yy}\dot{\omega}_y}{\left(I_{xx} - I_{zz}\right)\omega_x + I_w\omega_1} \tag{32}$$

$$\omega_y = \frac{I_w\omega_z\omega_2 - I_w\dot{\omega}_1 - I_{xx}\dot{\omega}_x}{\left(I_{zz} - I_{yy}\right)\omega_z + I_w\omega_3} \tag{33}$$

From these equations, it is possible to solve for ω by substituting Eq. (31) into Eq. (32) and Eq. (32) into Eq. (33) and solving numerically. Note that all terms in these equations are known except for satellite angular velocities. Recognizing that

$$d\omega = \alpha dt \tag{34}$$

$\overrightarrow{\omega_{sc}}$ can be solved for at various points in time for more accurate gyro calibration.

D. Analytical Stability Analysis

Using a linear controller is advantageous because it allows for an analytical stability analysis and characterization of controller behavior for various gains. The simplest analysis is for the case of zero initial satellite angular velocity and a desired rotation angle θ about the *x* body-axis. For this simple case, the integral of the *x* angular acceleration will give the *x* angular velocity and the integral of the *x* angular velocity will yield the change in rotation angle θ. In reality, angular velocity values in the *y* and *z* axes will have an effect on the required rotation about the *x* axis. However, for the scenario of a satellite rotating about only one axis, it is easy to draw a block diagram and calculate a corresponding transfer function. A stability analysis based on this transfer function gives a reasonable first order approximation of how the system will behave based on selected gains and certain inputs and outputs. After estimating ideal gains analytically, a numerical simulation can be utilized to optimize the gains and ensure system stability.

For small values of θ, the $\sin(\theta/2)$ terms in q_e can be replaced by $\theta/2$. For the purpose of creating an input-output model, we can assume that the input is $\theta_{des}/2$. Based on the PD control equation, the input-output model becomes

$$\left(\frac{\theta_{des}}{2}-\frac{\theta}{2}\right)k_p+\dot{\theta}k_d=\ddot{\theta} \tag{35}$$

where θ_{des} denotes the desired angle. The difference between the measured and desired rotation angles about the *x*-axis represents the *x*-component of the error quaternion in this scenario. From Eq. (35) the block diagram in Figure 5 can be constructed.

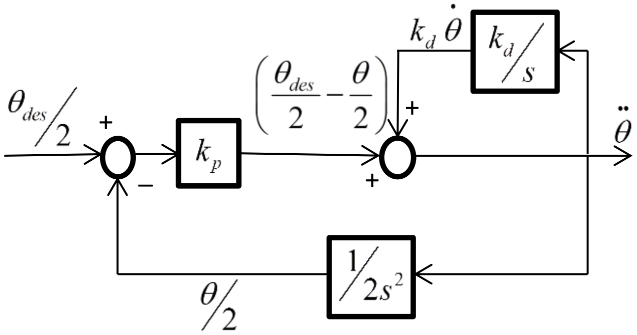

Figure 5. Block Diagram for PD Control Algorithm

Based on this block diagram and Eq. (35), the transfer function in the Laplace Domain relating the output (θ) to the input ($\theta_{des}/2$) of the controller is

$$T(\mathrm{s})=\frac{k_p/2}{s^2-sk_d+\frac{k_p}{2}} \tag{36}$$

From basic control theory, it is known that a system will be stable when the poles of its transfer function (roots of the denominator) have negative real components. If the poles have unreal components, the system will oscillate about an equilibrium. The poles of this transfer function can be written as

$$s_1,s_2=\frac{k_d\pm\sqrt{{k_d}^2-2k_p}}{2} \tag{37}$$

As we see from Eq. (37), k_d must be negative and k_p must be positive for the system to be stable. If the term in the square root in Eq. (37) is positive, the system is overdamped and the satellite will converge to the desired orientation more slowly than optimal. If the term in the square root is negative, the system is underdamped and the satellite will oscillate about the equilibrium position before stabilizing. If the term within the square root is zero, the system is critically damped and will converge to the desired orientation as quickly as possible without any oscillations. A critically damped system is ideal. According to Eq. (37), controller gains for such a system must be chosen such that

$$k_p={k_d}^2\big/2,\ k_d<0 \tag{38}$$

As will be shown in the simulation however, the analytically calculated gains for critical damping will not always result in ideal system behavior. This is because the analytical analysis assumes that the derivative term is simply the derivative of the proportional term, but this is not the case. The derivative term, a three-dimensional angular velocity vector, is not the derivative of the proportional term, a four-dimensional quaternion. However, these analytical solutions serve as good first order approximations of system behavior given a set of controller gains.

E. Simulation Results

A numerical simulation makes few, if any, simplifying assumptions and most accurately depicts the way a system will behave in real life. MSC Software's Adams multi-body dynamics simulator was utilized to model the case of a 1.5U (10x10x15 cm) CubeSat with three orthogonally mounted reaction wheels. The inverse dynamics PD control algorithm given in Eq. (18) was implemented in Adams based on specified k_p and k_d values and a given desired quaternion relative to an inertial "ground" reference frame representing the orbital frame. When integrator error was accounted for, actual satellite angular accelerations matched the desired angular accelerations calculated using Eq. (22) almost perfectly. However, simulations with analytically calculated gains did not always behave exactly the way the analytical analysis predicted. Often, the satellite would oscillate about an equilibrium position in the steady state but the magnitude of the oscillations would neither increase nor decrease. This oscillation was technically a stable solution (just as the analytical solutions indicated), but not the kind of steady state solution one would want for a satellite. The frequency and amplitude of this oscillation could be minimized by making the system over-damped by simply increasing the magnitude of k_d. Regardless of the initial conditions, the system would always stabilize to the same constant amplitude oscillation for given gain values. However, for small initial angular momentum values, the system took longer to reach the oscillatory mode.

In simulations with zero initial angular momentum, the satellite would first reach the desired attitude and would start oscillating only after the simulation was allowed to keep running for a long time. For example, a system that converged to the desired attitude in 10 seconds would start oscillating if the simulation was allowed to run for another 100 seconds. A system with an initial angular momentum would reach the oscillatory mode more quickly. Increasing the time step would also cause the system to reach the oscillatory mode more quickly. This led to the conclusion that the oscillatory mode was a stable equilibrium while the precise pointing mode was an unstable equilibrium. When the satellite was pointing precisely where it should be, a small disturbance, such as integrator error due to a large time step or large angular momentum, would trip it to the oscillatory mode where it would remain indefinitely.

Simulations were run with a CubeSat model where the system moments of inertia were 0.06 kg*m^2 about each axis and each wheel had a moment of inertia of 0.0523 kg*m^3 about its axis of rotation. Wheel moment of inertia was large to reduce the required wheel angular velocities and minimize integrator error. Due to the nature of the inverse dynamics controller however, the system would behave the same regardless of wheel moment of inertia. Figures (6-8) contain simulation results for critically damped, underdamped, and overdamped systems respectively with the aforementioned characteristics and zero initial angular velocity. This would correspond to a case where the steady state pointing control began after de-tumble was complete. Simulations were run with gain values corresponding to an underdamped, overdamped, and critically damped system. As expected, the underdamped system overshot the equilibrium before converging, and the overdamped system took longer to stabilize than did the critically damped system. When these same simulations were run with finite initial satellite angular velocity values however, the satellite did not completely stabilize at an equilibrium position but rather oscillated about the equilibrium. The same thing happened for systems with zero initial angular velocity when simulations were allowed to run for a long period of time. Figures (9-11) are graphs of simulation runs with an initial satellite angular velocity of $\overrightarrow{\omega_{sc}} = (.5,1,.7)$ rad/s. As seen in Figures (9-11), making the system over-damped by increasing the magnitude of the derivative gain (making it more negative) decreased the amplitude and frequency of the oscillations.

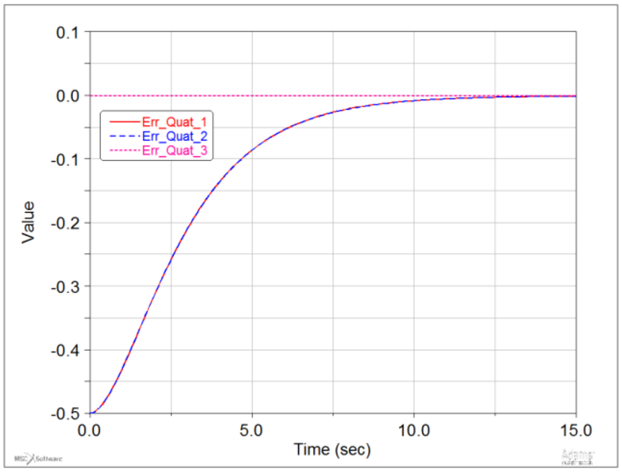

Figure 6. System with k_p = 1, k_d = -1.4, and $\omega_0 = (0,0,0)$

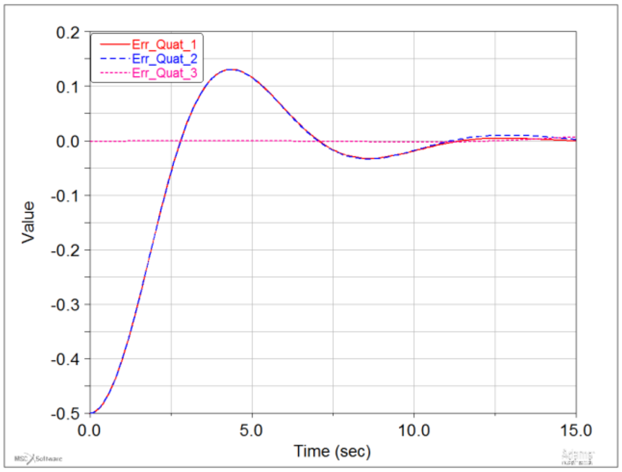

Figure 7. System with k_p = 1, k_d = -.5, and $\omega_0 = (0,0,0)$

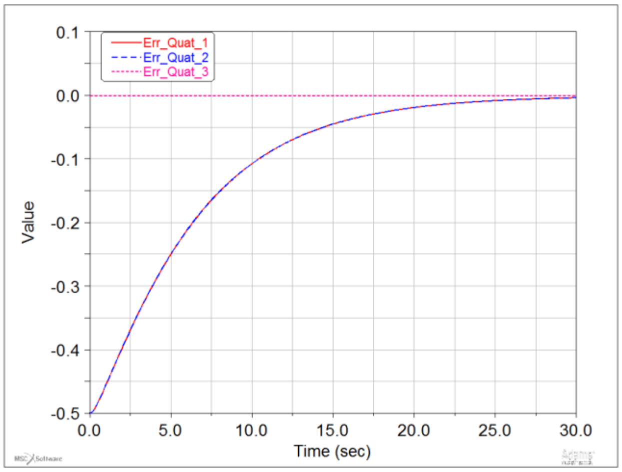

Figure 8. System with k_p = 1, k_d = -4, and $\omega_0 = (0,0,0)$

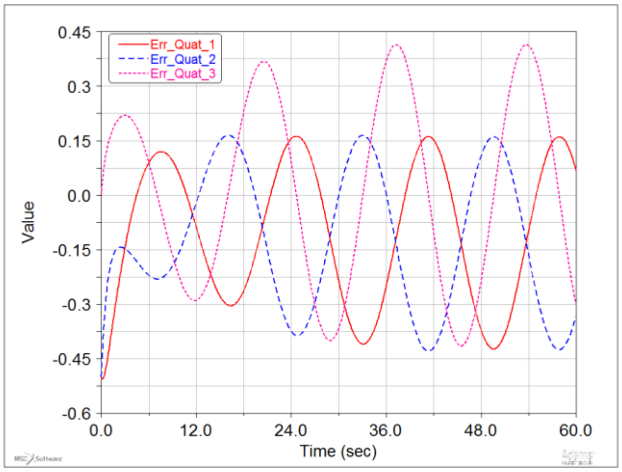

Figure 9. System with k_p = 1, k_d = -1.4, and $\omega_0 = (.5,1,.7)$

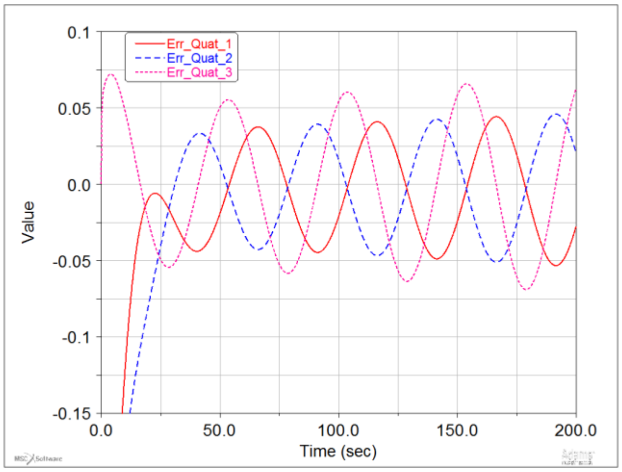

Figure 10. System with k_p = 1, k_d = -4, and $\omega_0 = (.5,1,.7)$

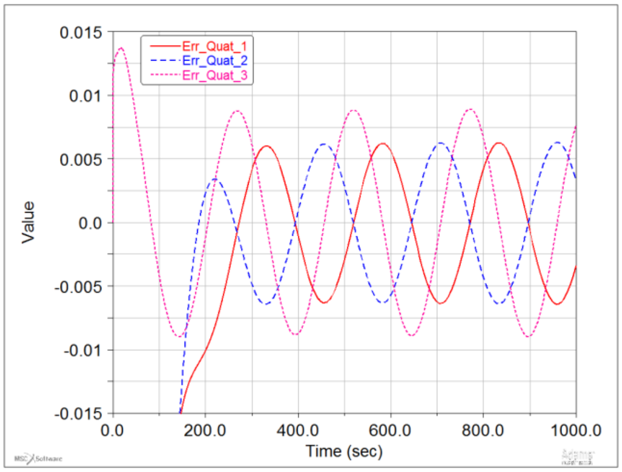

Figure 11. System with k_p = 1, k_d = -20, and $\omega_0 = (.5,1,.7)$

As can be seen in Fig. 11, a heavily overdamped system, while more stable, requires longer to reach equilibrium. For a real satellite, it may be impractical to increase the derivative gain to a point where the satellite takes several minutes to change orientation. Increasing both the proportional and derivative gains, however, makes the satellite reach equilibrium more quickly for a given damping configuration. For example, a critically damped system with k_p = 200 and k_d = -20 (Fig. 12) will reach equilibrium far more quickly than a system with k_p = 1, k_d = -1.4. If k_p is maintained at 200 and the magnitude of k_d is increased to, say, -500, the system will be very heavily overdamped and will experience very small steady state oscillations while still reaching equilibrium relatively rapidly (Fig. 13).

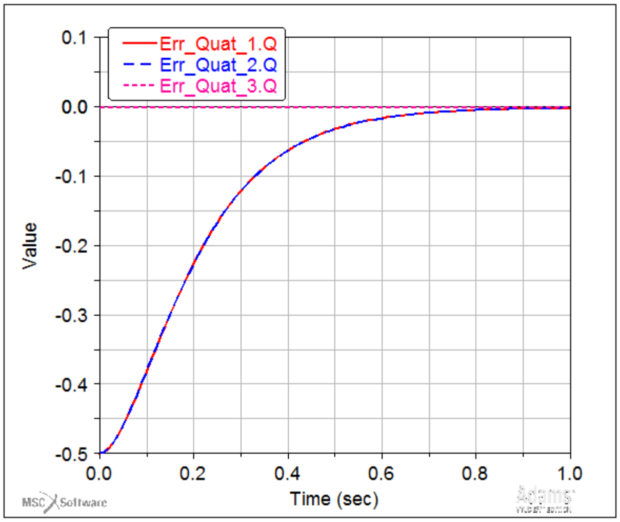

Figure 12. System with k_p = 200, k_d = -20, and $\omega_0 = (0,0,0)$

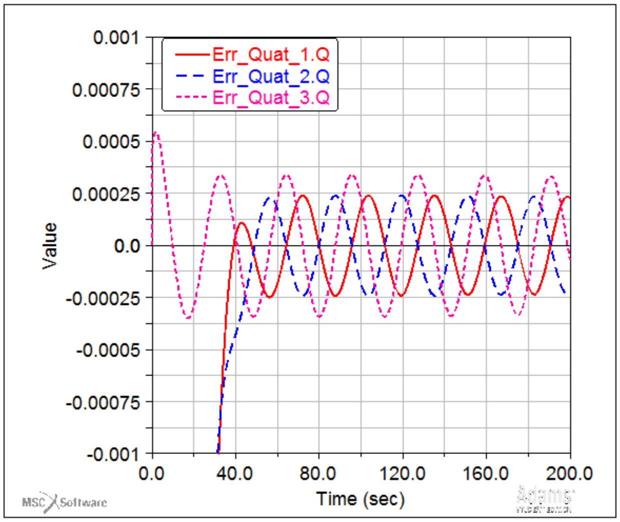

Figure 13. System with k_p = 200, k_d = -500, and $\omega_0 = (.5,1,.7)$

With such high gains, if the system experiences any perturbation or has some initial undesirable angular velocity, the commanded angular acceleration may exceed the capabilities of the reaction wheels. In this case, the commanded angular acceleration can be scaled down (Eq. (39)) such that it is less than a predetermined maximum achievable satellite angular acceleration $\alpha_{sc_{\max}}$.

$$\overrightarrow{\alpha_{sc}}' = \overrightarrow{\alpha_{sc}} \left(\frac{\left|\alpha_{sc_{\max}}\right|}{\left|\overrightarrow{\alpha_{sc}}\right|} \right) \tag{39}$$

VI. Conclusions

It is indeed possible to build a straightforward yet capable attitude determination and control system for a CubeSat or other small satellite. A magnetometer, rate gyros, and position sensitive detectors (PSDs) are adequate to provide 3-axis attitude determination. A magnetometer will give the local magnetic field vector in the satellite body frame and the PSDs will give the sun vector in the body frame. By comparing these values to the magnetic field vector and sun vector in the orbital frame given by IGRF lookup tables and a Keplarian Earth-Sun orbit propagator, a direction cosine matrix can be derived relating the orbital and body frames. An attitude quaternion can then be derived from this direction cosine matrix that specifies a line of rotation and a rotation angle to align the body frame with the orbital frame.

Based on a desired quaternion between the orbital and body frames and the measured quaternion, an error quaternion can be calculated that relates the current orientation to the desired orientation. This quaternion along with satellite angular velocity measurements can serve as inputs to a PD control algorithm that calculates the satellite angular accelerations required to slew the satellite to the correct attitude. Based on Euler's equation of rotational motion and system inertia properties, the reaction wheel angular accelerations and corresponding motor torques required to achieve these satellite angular accelerations can be determined. The PD control algorithm can be linearized and optimal proportional and derivative gain values can be estimated analytically. These gain estimates can then be refined using a numerical simulation.

According to simulation results using MSC Software's Adams multi-body dynamics simulator, the satellite will oscillate about an equilibrium position for gain combinations that are stable based on the analytical solutions. These oscillations eventually stabilize to a constant amplitude and frequency. The amplitude and frequency of these oscillations can be decreased by overdamping the system by increasing the magnitude of the derivative gain, but the system will take longer to reach a desired orientation. Increasing both the proportional and derivative gains will results in a heavily overdamped system with minimal steady-state oscillations that stabilizes to the desired orientation relatively quickly. Large gain values will result in larger commanded accelerations in the presence of perturbations. If a commanded angular acceleration exceeds the maximum capability of the spacecraft, it can be scaled down to a more reasonable value.

Acknowledgements

This work was supported by the Auburn University Student Space Program with funds from the Alabama Space Grant Consortium. This author was advised by Dr. JM Wersinger and Dr. David Beale over the course of this research. This author also benefitted from editing by Ms. Abigail Tillman and from conversations with Dr. Riccardo Bevilacqua, Dr. Andrew Sinclair, Dr. Doug Sinclair, and Mr. John Rakoczy.

References

[1] Andersson, Henrik. "Position Sensitive Detectors - Device Technology and Spectroscopy." PhD Thesis. 2008.

[2] AndyCui1. "Position Sensitive Device." 27 April 2009. Wikimedia. <http://commons.wikimedia.org/wiki/File:2D_Position_Sensitive_Device.png>.

[3] Capo-Lugo, Pedro, John Rakoczy and Devon Sanders. "The B-Dot Earth Average Magnetic Field." Acta Astronautica (2014): 92-100.

[4] de Weck, Olivier. Attitude Determination and Control. Presentation. Boston: Massechusets Institute of Technology, 2001.

[5] Gregory, Brian. Attitude Control System Design for ION, the Illinois Observing NanoSatellite. Masters Thesis. Urbana-Champaign: Univeristy of Illinois, 2001.

[6] Isidori, Alberto. Nonlinear Control Systems, Third Edition. Springer-Verlag, 1995.

[7] Junquan, L., et al. Design of Attitude Control Systems for CubeSat-Class Nanosatellite. Toronto: York University, 2013.

[8] Kulakowski, B., J. Gardner and J. Shearer. Dynamic Modeling and Control of Engineering Systems. New York: Cambridge University Press, 2007.

[9] Markley, F. Attitude Determination using Two Vector Measurements. Greenbelt, MD: NASA Goddard Space Flight Center, 1998.

[10] Mayhew, C., R. Sanfelice and A. Teel. On Quaternion-Based Attitude Control and the Unwinding Phenomenon. San Francisco: American Control Conference, 2011.

[11] Pietruszewski, A. and A. Spencer. Prox-1 Attitude Determination and Control. Atlanta, GA: Georgia Institute of Technology, n.d.

[12] Schaub, H. and J. Junkins. Analytical Mechanics of Aerospace Systems. 2002.

[13] Wei, Bong. Space Vehicle Dynamics and Control. Reston, VA: American Institute of Aeronautics and Astronautics, 1998.

[14] Weiss, A., et al. "Inertia-Free Spacecraft Attitude Control with Reaction-Wheel Actuation." AIAA Guidance, Navigation, and Control Conference. Ontario, Canada, 2010.

[15] Wertz, James R., David F. Everett and Jeffery J. Puschell. Space Mission Engineering: The New SMAD. Hawthorne: Microcosm Press, 2011.