



Performance Analysis of Control Algorithms for FalconSat-3

Paul Tisa, Dr. Paul Vergez
USAF Academy Dept. of Astronautics
2354 Faculty Drive
USAF Academy CO 80841
719-333-2668
c06paul.tisa@usafa.edu, paul.vergez@usafa.edu

16th AAS/AIAA Space Flight
Mechanics Conference

Tampa, Florida January 22-26, 2006

AAS Publications Office, P.O. Box 28130, San Diego, CA 92198

ABSTRACT

FalconSat-3 is the United States Air Force Academy’s first attempt at three-axis attitude determination and control. This requirement is a product of the three US Department of Defense payloads the satellite will carry, which are the Micro Propulsion Attitude Control System (MPACS), the Flat Plasma Spectrometer (FLAPS), and the Plasma Local Anomalous Noise Environment (PLANE). Most restrictively, FLAPS requires ± one degree attitude determination within two standard deviations and ± five degrees attitude control within one standard deviation of ram direction. The following paper gives a cursory background into FalconSat-3; followed by a brief explanation into the development of a Simulink model in MATLAB. The main focus is the utilization of the Simulink model to test the implementation and performance of the following controllers: proportional-derivative (PD), proportional-integral-derivative (PDI), separate pitch from roll/yaw, linear quadratic regulator (LQR), B-dot, spin rate, cross product law, and “bang-off-bang”. The “bang-off-bang” controller is used in conjunction only with FalconSat-3’s pulse plasma thrusters not the magnetorquers like the other controllers. As such, it is discrete not continuous, and its implementation is quite different than the rest.

By the end of the analysis, several advantages and disadvantages of each controller were divulged. Ultimately, while many of the control algorithms could be used to meet FalconSat-3’s attitude control requirements, proper integration of the B-dot, spin rate, and cross product law controllers yielded the best balance between competing performance characteristics.

TABLE OF CONTENTS

Abstract _____ 2
Introduction _____ 2
FalconSat-3 Background _____ 2
Coordinate Frames _____ 3
Model Dynamics _____ 4
Simulation Basics _____ 5
Theory _____ 6
Results and Discussion _____ 8
Conclusions _____ 13

Recommendations for Future Research 14

Acknowledgements _____ 14
References _____ 14
Appendices _____ 16

INTRODUCTION

FalconSat-3 is the most recent satellite project undertaken by the United States Air Force Academy. Many subsystems are being advanced in their complexity from FalconSat-2, but of main concern for this research is the requirement change from simplified two- to three-axis attitude control. Specifically, the satellite’s controller must be able to maintain ± five degrees attitude control within one standard deviation of ram direction using the satellite’s magnetorquers. While previous research has shown that this is possible, comparing the performance of several controllers to determine the best choice has never been conducted. The first step in successfully carrying out this task is to understand the system that must be controlled and the available actuators that must do the controlling.

FALCONSAT-3 BACKGROUND

By industry standards with a mass of 46.1 kg, FalconSat-3 is considered a micro satellite⁶. It has five payloads, three with attitude requirements, with the most arduous requirement coming from FLAPS, Flat Plasma Spectrometer. The pertinent satellite properties and spatial dimensions are summarized in Table 1 and Figure 1, respectively.

TABLE 1: FALCONSAT-3 PROPERTIES

Body Mass:	35.5 kg
Boom and Tip Mass:	10.6 kg
Total Mass:	46.1 kg
Pre-boom Inertia	* [3.64 3.69 1.31] kg - m ²
Post-boom Inertia	* [67.40 67.45 1.31] kg - m ²
Coefficient of Drag	2.6
Spacecraft Dipole	0.05 A-m ²
Orbital Characteristics:	Altitude = 560 km Inclination = 0° Eccentricity = 0

* = Products of inertia are all zero and are left out, the displayed values are for the moments of inertia only

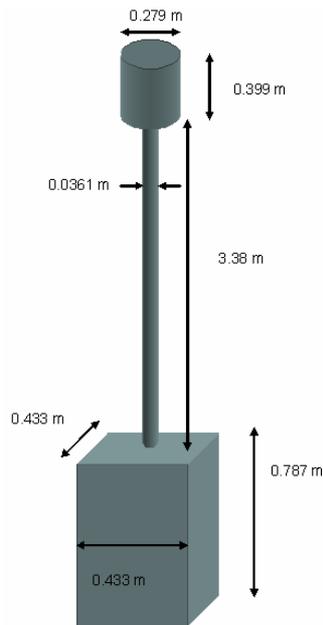


Figure 1: FalconSat-3 Configuration⁴

As seen in Figure 1, FalconSat-3 has a gravity gradient boom, a passive actuator. The satellite's main active actuators are three magnetorquers with a configurable dipole

moment of $3.0 \text{ A}\cdot\text{m}^2$. Their orientation with respect to the spacecraft's body frame is not important for this discussion.

The other active actuators on FalconSat-3 are the pulse plasma thrusters (PPTs), which are located on the satellite's earth facing facet during nominal operation. Previous research has shown they are effective enough to be used for attitude control during the satellite's operational lifetime. However, their main purpose is to demonstrate an ability to affect FalconSat-3's attitude and will not be relied on as a primary actuator for the spacecraft. As such, most research has focused on the magnetorquers.

COORDINATE FRAMES

An area of primary importance to define is the utilized coordinate systems. Throughout the simulation and algorithms, the three primary coordinate frames are inertial, local orbital, and body. Their characteristics are summarized in Table 2 and the relationships between the frames are visually displayed in Figures 2 and 3.

TABLE 2: COORDINATE FRAMES DEFINED

Name	Origin	Fundamental Plane	1 st Axis	Definition	2 nd Axis	Definition	3 rd Axis	Definition
Earth-Centered, inertial	Center of Earth	Earth's mean equator	I	Direction of mean vernal equinox	K	Direction of mean rotational axis (N is +)	J	Completes right-hand system
Local Orbital	Satellite center of mass	Satellite's orbital plane	P	Ram direction	Q	Direction of orbit normal (\mathbf{N}_{\oplus} is +)	R	Completes right-hand system
Body	Satellite center of mass	Parallel to satellite's base plate	X	Parallel to base plate (PLANE is +)	Z	Parallel to boom (Deployment is +)	Y	Completes right-hand system

Local Orbital and Earth Inertial

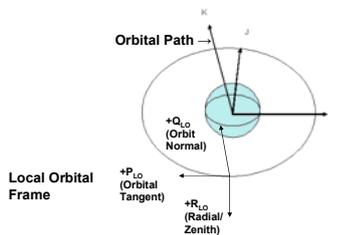
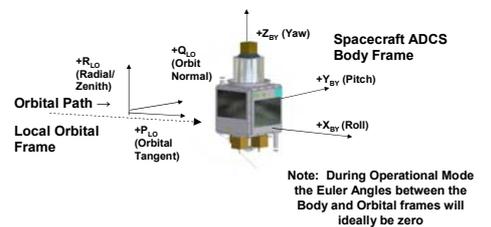


Figure 2: Illustration of FalconSat-3's Local Orbital and Inertial Frames⁴

Body and Local Orbital Frame



Note: During Operational Mode the Euler Angles between the Body and Orbital frames will ideally be zero



Figure 3: Illustration of FalconSat-3's Body and Local Orbital Frames¹

It is important to note that many of the most important sources for this work define the body coordinate system differently. In their works listed in the reference section, Dr. Vaios, Dr. Hashida, and Dr. Steyn define the z-body axis as parallel to the boom but positive towards the earth. The x-body axis is just as defined in this paper, which means to maintain a right handed coordinate system, the y-axis is opposite FalconSat-3's y-body axis^{3,6,7}.

There are several more frames of debatable value that are left out. As mentioned previously, the magnetorquers and magnetometer are not truly aligned with the spacecraft's body frame. Especially because of the focus of this paper, there is a great concern over information coming from the magnetometer and going to the magnetorquers. Technically, coordinate frames should be defined for each magnetorquer and for the magnetometer, and any data passed to or from these instruments should be transformed to the appropriate coordinate frame. However, as every controller tested suffers from this same error, its effect on each controller's performance is assumed to be the same.

MODEL DYNAMICS

Delving into explicit particulars of all the model dynamics would disrupt the purpose of this paper, but there are a some critical details that are necessary for clarity.

Three positive right hand rotations can be defined by Euler angles to go from the orbital to body frame. For FalconSat-3, the 2-1-3 rotation used by previous ADCS researchers was maintained and formed the following direction cosine matrix, C:

$$C^{B/O} = \begin{bmatrix} C\psi C\theta + S\psi S\phi S\theta & S\psi C\phi & -C\psi S\theta + S\psi S\phi C\theta \\ -S\psi C\theta + C\psi S\phi S\theta & C\psi C\phi & S\psi S\theta + C\psi S\phi C\theta \\ C\phi S\theta & -S\phi & C\phi C\theta \end{bmatrix} \quad (1)^7$$

, where C is the cosine function and S is the sine function. ϕ, θ, ψ are roll, pitch, and yaw, respectively. There are a number of rotation combinations that will end with the same results. However after this is converted to its appropriate quaternion counterpart, to eliminate singularities present in the Euler angles, intermediate values will change based upon the rotation used. Quaternion attitude for a 2-1-3 rotation is defined as:

$$q_1 = \frac{S(\psi)S(\theta) + C(\psi)S(\phi)C(\theta) + S(\phi)}{q_4} \quad (2),$$

$$q_2 = \frac{C(\phi)S(\theta) + C(\psi)S(\theta) - S(\psi)S(\phi)C(\theta)}{q_4} \quad (3),$$

$$q_3 = \frac{S(\psi)C(\phi) + S(\psi)C(\theta) - S(\phi)S(\theta)C(\psi)}{q_4} \quad (4),$$

$$q_4 = -\frac{1}{2} * \sqrt{1 + C(\psi)C(\theta) + S(\psi)S(\phi)S(\theta) + C(\psi)C(\phi) + C(\theta)C(\phi)} \quad (5)$$

, where the used nomenclature has q_4 as the scalar component.

The rotation matrix in quaternion form is:

$$C^{B/O} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (6)^7$$

To return to Euler angles from quaternion attitude, the simulation used the following relationships:

$$\phi = a \sin(-2q_2q_3 - q_1q_4) \quad (7)$$

$$\theta = a \tan 2 \left(\frac{2(q_1q_3 + q_2q_4)}{-q_1^2 - q_2^2 + q_3^2 + q_4^2} \right) \quad (8)$$

$$\psi = a \tan 2 \left(\frac{2(q_1q_2 + q_3q_4)}{-q_1^2 + q_2^2 - q_3^2 + q_4^2} \right) \quad (9)$$

, where atan2 is a tangent function with bounds between -180° and 180° instead of -90° and 90° .

The plant equation used to model the motion of FalconSat-3 is:

$$I\dot{\vec{\omega}}_B^I = N_{MT} + N_{PPT} - \vec{\omega}_B^I \times I\vec{\omega}_B^I - \dot{I}\vec{\omega}_B^I \quad (10)^7$$

, where $\vec{\omega}_B^I$ equals an angular body rate vector with respect to the inertial frame, I is the 3x3 inertia matrix, N_{MT} is the magnetorquer vector, and N_{PPT} is the PPT vector.

The quaternion attitude was updated through the following equation:

$$\dot{q} = \frac{1}{2} \Omega q \quad (11)^7$$

$$, \text{ where } \Omega = \begin{bmatrix} 0 & \omega_z^o & -\omega_y^o & \omega_x^o \\ -\omega_z^o & 0 & \omega_x^o & \omega_y^o \\ \omega_y^o & -\omega_x^o & 0 & \omega_z^o \\ -\omega_x^o & -\omega_y^o & -\omega_z^o & 0 \end{bmatrix} \quad (12)^7,$$

$$, \text{ where } \vec{\omega}^o = \begin{bmatrix} \omega_x^o \\ \omega_y^o \\ \omega_z^o \end{bmatrix} = \vec{\omega}_B^I - C^{B/O} \vec{\omega}_0^o \quad (13) \text{ and}$$

$$\vec{\omega}_0^o = \begin{bmatrix} 0 \\ \omega_0^o \\ 0 \end{bmatrix} \quad (14) \text{ and } \omega_0^o = \sqrt{\frac{\mu_{\oplus}}{a^3}}. \quad (15)$$

The only other general equation worth mentioning is how to calculate quaternion error. The controllers' responses are all related to quaternion and/or rate error, so it is a crucial concept. For both, the idea is to subtract the measured from commanded in the same coordinate frame to determine the error vector. Equations 16 and 17 illustrate how this is done for both the rates and quaternion attitude:

$$\vec{\omega}_e = \begin{bmatrix} \omega_x - \omega_x^{cmd} \\ \omega_y - \omega_y^{cmd} \\ \omega_z - \omega_z^{cmd} \end{bmatrix} \quad (16)$$

$$\vec{q}_{e,1-4} = \begin{bmatrix} q_{1e} \\ q_{2e} \\ q_{3e} \\ q_{4e} \end{bmatrix} = \begin{bmatrix} q_{4cmd} & q_{3cmd} & -q_{2cmd} & -q_{1cmd} \\ -q_{3cmd} & q_{4cmd} & q_{1cmd} & -q_{2cmd} \\ q_{2cmd} & -q_{1cmd} & q_{4cmd} & -q_{3cmd} \\ q_{1cmd} & q_{2cmd} & q_{3cmd} & q_{4cmd} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (17)^6$$

SIMULATION BASICS

Without getting into unnecessary detail, it is important to include the limitations and nomenclature of the developed Simulink model to draw proper conclusions from the results. A figure of the current model is posted in Appendix A. Figure 4 gives a simplified overview of the simulation parts.

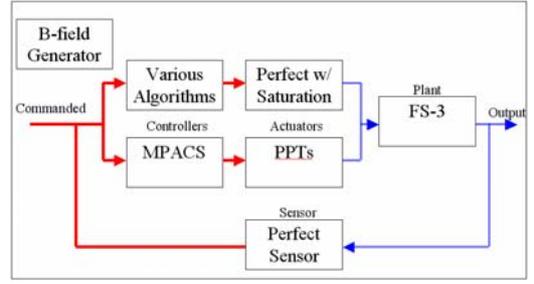


Figure 4: Block Diagram of Simulation

There are several important limitations in the model that are prevalent from the above diagram. First, the “B-field Generator” block is really an orbit propagator that calculates the magnetic field vector in the inertial frame that is later transformed into the necessary coordinate frame. However, at the time of the controller testing, the orbit propagator does not account for any perturbations or allow the inputs to change over time in any other manner. For example, during the actual operation of FalconSat-3, new two-line elements, TLEs, will be sent to the satellite every two weeks. As such, error in the generated magnetic field vectors will increase as the simulation continues. While untested, this was not perceived as a significant source of error while testing different controllers because it would slowly affect each in the same way.

Second, there are torques acting besides those put on the system by the currently tested controller and actuator. Gravity gradient and disturbance torques were left out of this research as they were not correctly modeled when testing started. After they were modeled more accurately, they were left out as it was assumed there effect on the system would be similar for every controller.

Next, the magnetorquers were modeled as perfect actuators at this point in design. An expectable amount of torque from the torque rods was calculated to be $9.93 \cdot 10^{-5}$ N-m² through the equation:

$$T = DB \quad (18)^8$$

, where “T” equals torque in N-m, “B” is Earth’s worst case magnetic field at an inclination of 35° in tesla, and “D” is the magnetorquer’s dipole moment in A-m². The magnetic field was calculated by interpolating the magnetic field over the equator, $M/R^3 = 2.38 \cdot 10^{-5} T$, and over the poles, $2M/R^3 = 4.77 \cdot 10^{-5} T$ for a satellite with the same altitude but with FalconSat-3’s inclination.

Another important point of consideration is the sensor block, which during testing was simplified to a perfect sensor. During real operation, the sensors would generate noise and imperfections in the information. However, during actual operation, FalconSat-3 would have a Kalman filter implemented that would assist in the detrimental effects of imperfect sensor and propagator data. It would calculate best guess data to pass back to the controller. As the focus of this research was on the controllers, both the filter and imperfect sensors were excluded. The comparative difference in the controllers' performance was assumed negligible.

THEORY

Proportional-Derivative (PD)

The PD control implemented in the model was the simplest controller possible. It is a variation of the cross product law controller as outlined by Dr. Hashida. Instead of using the calculated error vector to determine the most favorable magnetorquing direction, it is directly utilized as the commanded torque vector. The error/commanded torque vector is determined through the following equation:

$$\bar{u} = -K_p \bar{\alpha}_e - K_d \bar{\omega}_e \quad (19)^3$$

, where

$$\bar{\alpha}_e = \begin{bmatrix} \phi_1 - \phi_1^{cmd} \\ \theta_2 - \theta_2^{cmd} \\ \psi_3 - \psi_3^{cmd} \end{bmatrix} \quad (20), \quad \bar{\omega}_e = \begin{bmatrix} \omega_x - \omega_x^{cmd} \\ \omega_y - \omega_y^{cmd} \\ \omega_z - \omega_z^{cmd} \end{bmatrix} \quad (21),$$

and K_p and K_d are the proportional and derivative gain matrices, respectively. The gain matrices were chosen based on the performance of the system acquired from previous tests.

Proportional-Integral-Derivative (PID)

The PID controller had only two differences to the PD. First, the position error vector was not determined by first converting the quaternion attitude to Euler angles. The error between the commanded and measured quaternion attitude was determined by the before mentioned matrix. The scalar component of the error vector was excluded while the other three parts were passed on to determine the total error vector calculation.

Second, its error vector determination included an integrator term, altering the calculation to:

$$\bar{u} = -K_p \bar{q}_{e,1-3} - K_i \int \bar{q}_{e,1-3} - K_d \bar{\omega}_e \quad (22)^6$$

, where everything is defined as in the PD controller except the quaternion error vector and the integral gain matrix, K_i . Once again, the gain matrices were chosen based on the performance of the system acquired from previous tests.

Separate Pitch and Roll/Yaw

Starting with the basic equation of motion, circular orbit, small angle, and insignificant higher order terms assumptions are made. This allows the separation of the pitch axis EOM from the roll/yaw EOM. Continuing the derivation, FalconSat-3's plant can eventually be modeled in the Laplace domain. The pitch and roll/yaw characteristic equations in the s-plane respectively are:

$$s^2 + 3(\omega_0^0)^2 \frac{(I_1 - I_3)}{I_2} = 0 \quad (23)^9$$

$$s^4 + (1 + 3k_1 + k_1 k_3)(\omega_0^0)^2 s^2 + 4k_1 k_3 (\omega_0^0)^4 = 0 \quad (24)^9$$

, where $k_1 = \frac{I_2 - I_3}{I_1}$ (25)⁹ and $k_3 = \frac{I_2 - I_1}{I_3}$ (26)⁹.

For the pitch equation, I_1 must be greater than I_3 for purposes of stability. The following criteria must be met to have stability in the roll/yaw equation: $k_1 k_3 > 0$ (27)⁹,

$$1 + 3k_1 + k_1 k_3 > 0 \quad (28)^9, \quad \text{and}$$

$(1 + 3k_1 + k_1 k_3)^2 - 16k_1 k_3 > 0$ (29)⁹. Because of this criteria and the physical properties of FalconSat-3, $I_1 = I_x$, $I_2 = I_y$, and $I_3 = I_z$.

Ultimately, this approach only redefines the system's plant. Other design methods, such as PD and PID, must still be used to create a controller. Greater detail on the derivation of the equations and material contained in this section is found in chapter 6 of Bong Wie's work sited in the References section.

Linear Quadratic Regulator Theory (LQR)

LQR is a modern control, time domain technique that originates with basic state-space modeling:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (30) \quad y(t) = Cx(t) + D \quad (31)$$

, where $x(t)$ is a vector of system states, $u(t)$ is a controller input vector, and $y(t)$ is a vector of outputs. For easier reading, $x(t)$, $u(t)$, and $y(t)$ vectors are no longer explicitly written as functions of time.

LQR begins with the basic feedback control algorithm of:

$$u = -Kx \quad (32)^9$$

, where K is a matrix of gains, which is chosen to minimize a performance index algorithm, such as:

$$J = \frac{1}{2} \int_0^{\infty} (x'Qx + u'Ru)dt \quad (33)^9$$

, where Q is known as the state weighting matrix and R is the control input weighting matrix. Minimizing Equation 33 yields:

$$K = R^{-1}B'S \quad (34)^9$$

, where S is a matrix that satisfies the Riccati equation:

$$0 = A'S + SA - SBR^{-1}B'S + Q \quad (35)^9$$

\dot{B} Controller

The B-dot controller is one of the most common controllers used to stabilize spacecraft with magnetorquers³. Equation 36 can be used to activate the i -axis magnetorquer to damp out the j - and k -axes angular velocity at the same time:

$$M_i = -K_d \dot{B}_i^B \quad (36)^3$$

, where K_d is a positive controller gain and B_i is the i -component of the magnetic field vector in the body coordinate system. In practice, \dot{B}_i^B is obtained by the approximation:

$$\dot{B}_{i,k}^B \approx \frac{B_{i,k}^B - B_{i,k-1}^B}{\Delta t} \quad (37)^3$$

, where Δt is the sampling time. Dr. Hashida recommends smoothing this approximation through Equation 38.

$$\overline{\dot{B}}_{i,k}^B = (1-s)\dot{B}_{i,k}^B + s\overline{\dot{B}}_{i,k-1}^B \quad (38)^9$$

, where \overline{B} is a smoothed measurement and s is a smoothing gain optimized for noise bounded as $0 \leq s \leq 1$. For FalconSat-3, the i -axis corresponds to the pitch body axis.

Spin Rate Controller (Y-Thomson)

A spin rate controller uses the k -magnetorquer to control the spin rate of the i -axis as shown

in Equation 39. The relationship between i , j , and k are displayed in Table 3.

$$M_k = -K_p (\omega_i^{cmd} - \omega_i) \text{sgn}(B_j^B) \quad (39)^3$$

TABLE 3: SPIN RATE CONTROLLER AXES IDENTIFICATION³

	x-spin	y-spin	z-spin
i	1	2	3
j	2	3	1
k	3	1	2

This controller has to be used in conjunction with the B-dot controller because the third axis is disturbed. For FalconSat-3, a Y-Thomson spin is realized by firing the roll axis magnetorquer to control the pitch axis, which inherently disturbs the yaw axis. Then the B-dot controller activates the pitch axis magnetorquer to dampen the remaining rates on the roll and yaw axes.

Cross Product Law

The cross product law starts off just like the PD controller, by determining the following error vector:

$$\vec{e} = -K_p \vec{\alpha}_e - K_d \vec{\omega}_e \quad (40)^3$$

, where all the variables are defined as in the PD controller. Instead of taking this error vector directly to the actuators, the cross product law then tries to determine the most favorable magnetorquing vector through Equation 41.

$$\vec{m} = \frac{\vec{e} \times \vec{B}^B}{|\vec{B}^B|^2} \quad (41)^3$$

Finally, the torque vector is determined as:

$$\vec{N}_M = \vec{m} \times \vec{B}^{LO} = \vec{m} \times C^{B/LO} \vec{B}^{LO} \quad (42)^3$$

MPACS (Pulse Width Modulation)

Unlike the other controllers, MPACS is designed for the PPTs and not the magnetorquers. The main control's difference between the two actuators is that a single magnetorquer can be commanded to alter both the direction and magnitude of its magnetic dipole, which becomes a torque after being crossed with the current magnetic field. Two PPTs put along the same axis but in different directions can simulate the direction change but the thrust generated is unalterable.

Because of this limitation, MPACS is limited to controlling FalconSat-3's yaw within a certain range of the commanded orientation and no better. The acceptable distance from zero within which MPACS commands no torque from the PPTs is called the deadband.

The actual algorithm is very simple. If the current yaw angle is outside the deadband, the PPTs are turned on. The sign of firing is always opposite the direction of error.

RESULTS AND DISCUSSION

Table 4 gives an overview of how far research progressed in each area as of the time this paper was written.

TABLE 4: OVERVIEW OF RESEARCH PROGRESS

Control Algorithm	Status
PD	Tested
PID	Tested
Separate Pitch from Roll/Yaw	Designed
LQR	Theoretical
B-dot Controller	Operational
Spin Rate	Operational
Y-Thomson	Operational
Cross Product Law	Operational
MPACS	Operational

The status column is a fairly objective judgment of the current position in that particular area. The chain of progression is as follows: concept, theory, design, implemented, operational, and tested. In concept, a controller is discovered. Next, the algorithms and concept of the controller are delved into. By design, the controller is understood enough to take the algorithms and theory to make a controller that is FalconSat-3 specific. An implemented controller is one that has been modeled in Simulink but is untested. Operational means a controller works in the simulation. Tested is the final stage, where a working controller has undergone a rigorous test battery and its affects on and performance within the model is well understood and documented.

Proportional-Derivative (PD)

The PD controller went through the most extensive test battery during research. While a complete list of the cases performed and their conditions are located in Appendix B, Table 5 shows an abbreviated list of the results.

TABLE 5: PD PERFORMANCE OVERVIEW

K_p	K_d	ang conv? / to cmd?	t_{smax} (s)	rates conv? / to cmd?	t_{smax} (sec)	N_{MTmax} (N-m)
3.5	13	Y/Y	11.3	Y/Y	25.3	1.14
1	1	Y/Y	522	Y/Y	999	0.505
0	1	Y/N	5,146	Y/Y	201	0.0353
1	0	N/N	∞	N/N	∞	∞
0.5	0.5	Y/Y	1,031	Y/Y	1,814	0.317
0	0.5	Y/N	6,110	Y/Y	405	0.0178
0.5	0	N/N	∞	N/N	∞	∞
0.25	0.25	Y/Y	2,390	Y/Y	3,600	0.212
0	0.25	Y/N	4,310	Y/Y	815	0.00936

, where "ang" is body angles, "conv" is converge, "cmd" is commanded, " t_{smax} " is maximum settling time, " N_{MTmax} " is maximum torque demanded from the magnetorquers, "Y" is yes and "N" is no.

The above cases were all run with initial angles equal to twenty degrees and body rates of two degrees per second. The table illustrates each test's proportional and derivative gains, whether the angles and rates converge, whether that convergence is to the commanded angles and rates, the maximum settling time, and maximum required torque. The settling time is defined as the first time the slowest settling axis' orientation stays within five degrees of the commanded orientation, or the first time the slowest settling axis' stays within 0.1 degrees per second of the commanded rate. The table shows how the performance of the controller is a function of the gains. Figures 5 and 6 illustrate the angles and rates, respectively, versus time for the first case in Table 5.

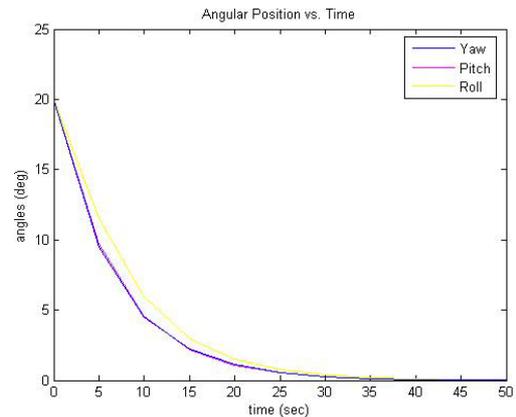


Figure 5: High Gain PD Angular Position vs. Time

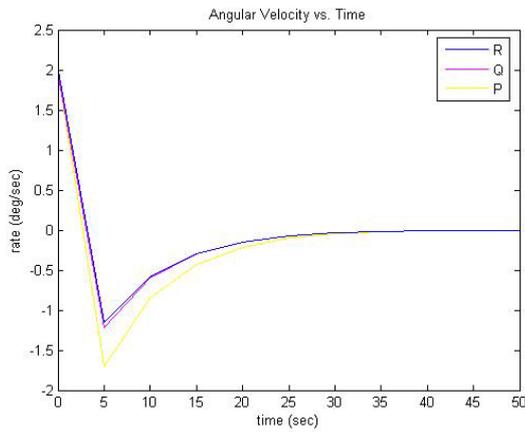


Figure 6: High Gain PD Angular Velocity vs. Time

While these gains produce quick settling times for the system, they demand far too much torque from the magnetorquers, greater than five orders of magnitude more than the estimated amount of torque available. Reducing the gains, increases the settling times but lowers the demanded torque.

Another trend noticed is that solely controlling the angles, having only a proportional controller, merely bounds the angular rates, as seen in Figure 7.

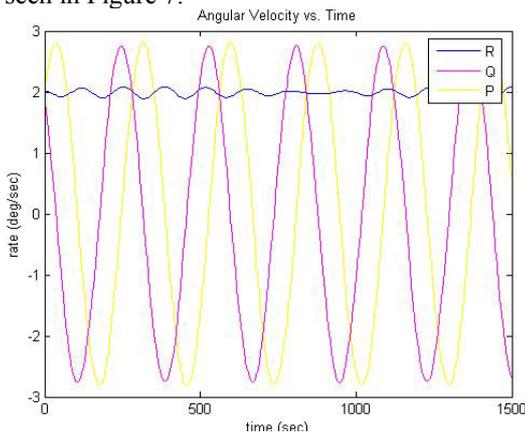


Figure 7: P Controller Angular Velocity vs. Time

More useful, having only a derivative gain, requires less torque for a given gain value than having both gains set. Taking this one step further, a PD controller can be simulated that starts off as just a derivative controller then switches to a PD controller after the rates have been reduced so far. If the right balance is reached, this reduces the necessary torque and increases the performance as seen in Figures 8 and 9.

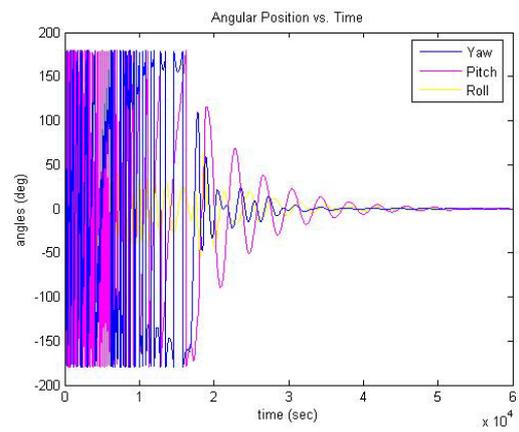


Figure 8: D/DP Controller Ang. Position vs. Time

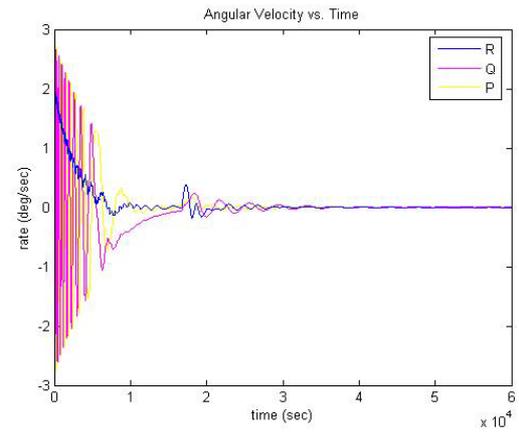


Figure 9: D/DP Controller Ang. Velocity vs. Time

Table 6 compares the performance of this D/DP controller versus any combination of its gains.

TABLE 6: D/DP PERFORMANCE SUMMARY

K_p	K_d	ang conv? / to cmd?	max t_s (s)	rates conv? / to cmd?	t_{smax} (sec)	N_{MTmax} (N-m)
0/1E-5	5E-4/4/1E-3	Y/Y	4.0E4	Y/Y	2.2E4	2.9E-5
0	5E-4	N/N	∞	Y/Y	1.5E4	2.4E-5
1E-5	1E-3	Y/Y	3.4E4	Y/Y	1.6E4	7.4E-5

Proportional-Integral-Derivative (PID)

The PID controller underwent a fairly extensive test battery during research. While a complete list of the cases performed and their conditions are located in Appendix C, Table 7 shows an abbreviated list of the results.

TABLE 7: PID PERFORMANCE SUMMARY

K_p	K_i	K_d	ang conv?	rates conv?	t_{smax} (s)	N_{MTmax} (N-m)
3.5	2.5E-4	13	Yes	Yes	25.4	1.14
0.035	2.5E-4	0.13	Yes	Yes	234.4	0.012
0.035	2.5E-4	0.013	No	No	∞	∞
0.035	2.5E-4	1.3	Yes	Yes	134.0	0.051
0.35	2.5E-4	0.13	Yes	Yes	200.0	0.075
0.0035	2.5E-7	0.013	Yes	Yes	1902	2.52E-3

Just as with PD, all the above cases had initial angles of twenty degrees and rates of two degrees per second. Figures 10 and 11 demonstrate the angles and rates versus time for the first case, which look almost identical to the similar PD case because the integral gain is so small.

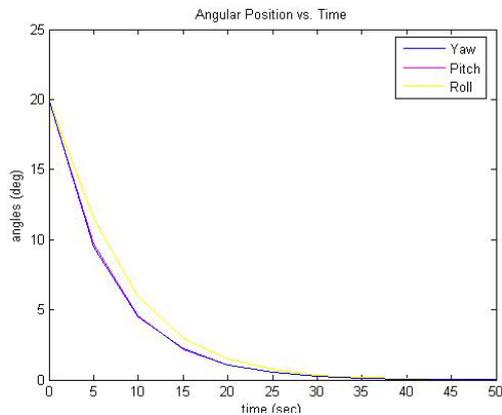


Figure 10: High Gain PID Angular Position vs. Time

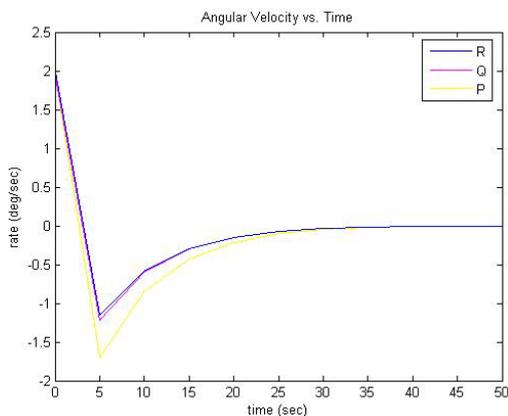


Figure 11: High Gain PID Angular Velocity vs. Time

As with PD, smaller gains demand less torque but have longer settling times. Among the trends noticed in the data collected, the integral derivative seemed to destabilize the system. Only by making it almost insignificant in comparison to the others, about 1000 times less, did the PID work as a controller and by that point it acted almost exactly like a PD controller.

Separate Pitch from Roll/Yaw

As previously mentioned, the separating pitch from roll/yaw method merely redefines the plant equation into the Laplace domain. Plugging FalconSat-3's inertias and orbital velocity into Wie's equations is a simple task. This was the extent reached for this method during research but the next steps are evident from past experience. After determining the

characteristic equation, there are a plethora of controller design tools, such as root locus plots, to help construct a controller that is initially more optimized than one obtained by guessing gains. This is a great method to start from as an undergraduate student where so many simple optimization tools are available. For example, this method can take advantage of second order approximations that graph performance specifications in the Laplace domain to also augment design.

However, this is just a starting point. The s-plane characteristic equations make several assumptions, including circular orbit, truncating higher order terms, and replacing trigonometric functions with their small angle approximations. While not necessarily bad or good, their affect on the accuracy of the plant equations needs to be tested, which makes this seemingly simple method less attractive. Also, once the requirements are put into ADCS terms, such as settling time, rising time, and so forth, they are easy to graph on the s-plane. However, taking real world requirements and putting them in ADCS terms is difficult. For example, FalconSat-3's requirements are listed at the beginning of this paper and there are no clear relationships between these and the basic controller performance characteristics.

Linear Quadratic Regulator Theory

LQR was never taken past a theoretical stage, so the discussion is based on the consulted sources without validation from the simulation.

To this theory's credit, a properly derived K , where Q and R are suitably chosen, guarantees closed loop stability³. Some other advantages of LQR is that it can be used for three axes control, is easy to apply to multiple-input-multiple-output systems, and can be modified to work for full-state feedback and state estimation.

On the other hand, LQR is far too complex to be intuitive to the beginning user. If the designed controller failed to work properly, knowing even the general location of the error would not be possible. Also, this theory creates a computationally intensive controller. For example, a six-state system, three angles and three rates, requires the inversion of a 6x6 matrix and multiplication by 12x12 matrices that might be complex numbers. As such, LQR controllers are not usable by 186 or even 386 satellite on board computers³.

B-dot

This controller never made it into extensive testing so the effects of altering the controller gain and smoothing constant on varying initial conditions are still unknown. Figure 12 shows the results of the simulation using only the B-dot controller with a controller gain of 2×10^6 , a smoothing gain of 0.5, sampling time of five seconds, initial angles of zero degrees, and initial rates of two degrees per second. A picture of the orientation versus time is unnecessary as this controller only tries to realign the rates.

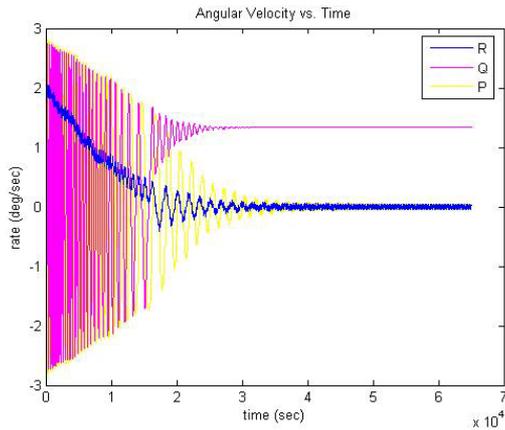


Figure 12: Angular Velocity vs. Time for B-dot Controller

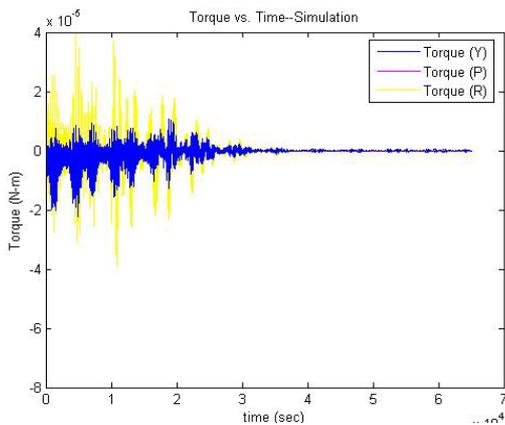


Figure 13: Torque vs. Time for B-dot Controller

First inspection shows a controller that performs as expected but very slowly. By only using the pitch axis magnetorquer, the B-dot controller brings the roll and yaw axes rates to zero while steadying the pitch axis to some constant value. While seemingly slower than PD or PID, a look at the torque versus time graph, Figure 13, reveals a very efficient controller. Note that since the torque is a function of the magnetorquer's dipole crossed with the magnetic field, the created torque is

purely on the roll and yaw axes when the pitch magnetorquer is the only one firing.

While it has a maximum setline time of around 36,100 seconds, Figure 13 shows that the B-dot controller commanded a maximum torque below 4×10^{-5} N-m, much lower than interpolated data from the PD and/or PID test battery results indicates.

Spin Rate / Y-Thomson

Like the B-dot, this controller never underwent extensive testing so the effects of changing the controller gain are still unknown. Figure 14 shows the results of the simulation using only a y-spin rate controller with a controller gain of 50, initial angles of zero degrees, and initial rates of two degrees per second. Once again, the angular position graph is not represented as the spin rate controller only attempts to bring the rates back to the commanded values.

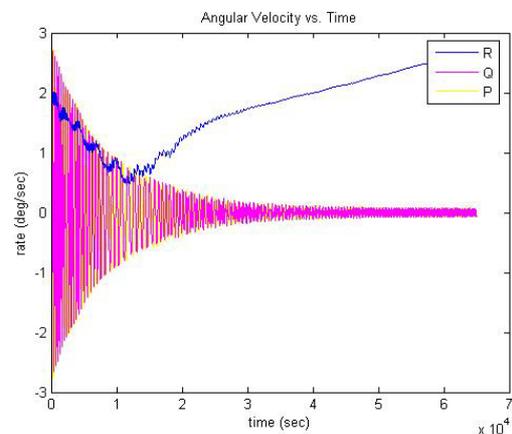


Figure 14: Angular Velocity vs. Time for Y-Spin Rate Controller

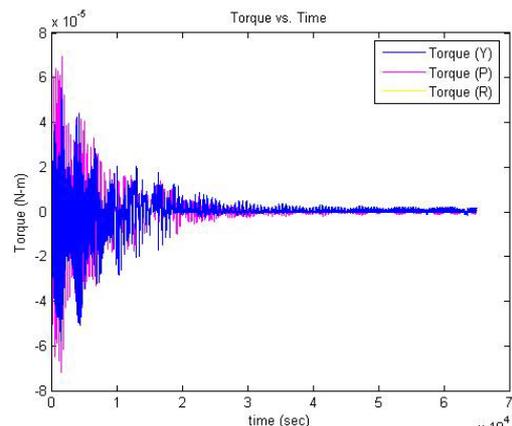


Figure 15: Torque vs. Time for Y-Spin Rate Controller

As expected, using the roll axis magnetorquer a y-spin rate controller starts constructively affecting all three axes. However, at some point it starts to disrupt the yaw axis, which is where the B-dot controller should be switched

on. Figure 15 shows the controller's torque profile versus time, maxing out around $7 \cdot 10^{-5}$ N-m. As predicted, the torque only acts on the pitch and yaw axes.

To create system stability, the y-spin controller must be used in conjunction with the B-dot controller. Incorporating both the B-dot and y-spin rate controllers requires several conditional statements. When the task is completed the resulting controller is a X-, Y-, or Z- Thomson, depending on the desired spin axis. For Falconsat-3, a Y-Thomson is required. Utilizing the same gains as in the previous individual implementations, Figure 16 shows the simulated rates over time

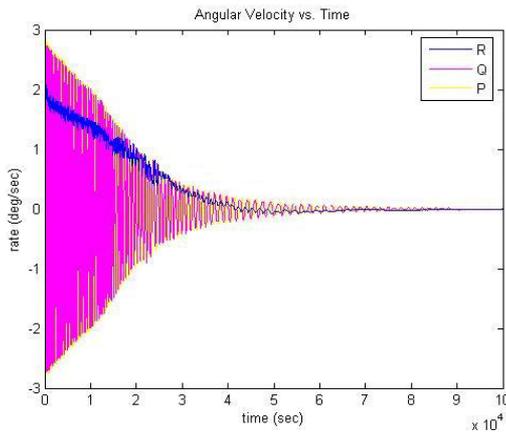


Figure 16: Angular Velocity vs. Time for Y-Thomson Controller

Cross Product Law

At the time of this paper, the Cross Product Law Controller has only recently been made operational. Like the B-dot, this controller never underwent extensive testing so the effects of changing the controller gains are still unknown. Figures 17 and 18 show the results of the simulation using a cross product law controller with a proportional gain of one and a derivative gain of two. The initial angles are zero degrees and the initial rates are two degrees per second. The angular position is shown for this controller as it should stabilize these back to zero.

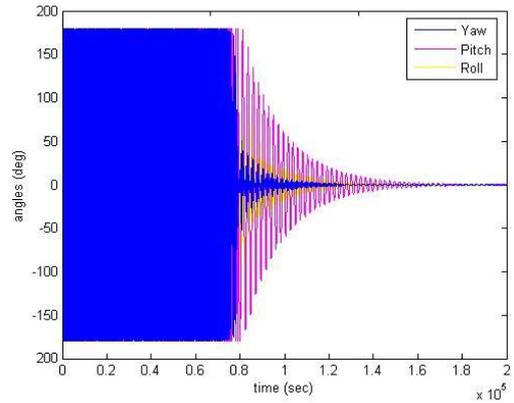


Figure 17: Angular Position vs. Time for Cross Product Law Controller

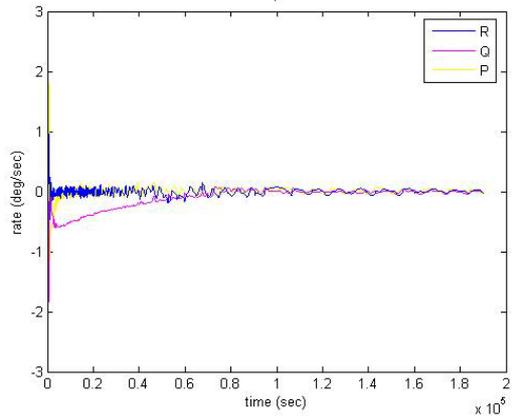


Figure 18: Angular Velocity vs. Time for Cross Product Law Controller

Micro Propulsion Attitude Control System

Simulations of this system focused on the yaw axis as this would be the easiest axis for the PPTs to affect. The implemented system has only undergone operational and not performance testing. For the first simulation, the initial orientation was ten degrees off commanded with PPT thrust levels at $4 \cdot 10^{-4}$ N-m, a deadband of two and one-half degrees, a thrust period of one second, and a duty cycle of 25% or 0.25 seconds. While the thrust level is orders of magnitude above the expectable thrust from the PPTs on Falconsat-3, the condition was merely chosen to test for operability. Figure 19 illustrates the thrust level, thrust period, and duty cycle for easier understanding of terms.

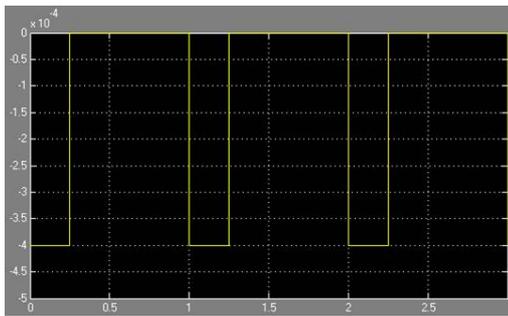


Figure 19: Thrust vs. Time (s) for MPACS/PPTs

Note the thrust level is shown as the outlined magnitude but opposite direction.

To check a hard-limit cycle, the simulation was run with no other torques and the results are illustrated by Figure 20. To check a soft-limit cycle, a simulation was run with a constant disturbance torque of 4×10^{-5} N-m, with the results shown in Figure 21.

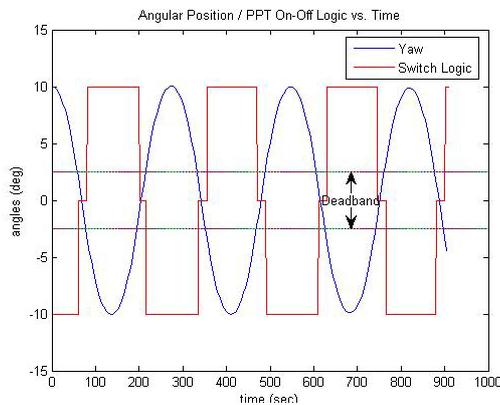


Figure 20: Hard-Limit Cycle for MPACS/PPTs

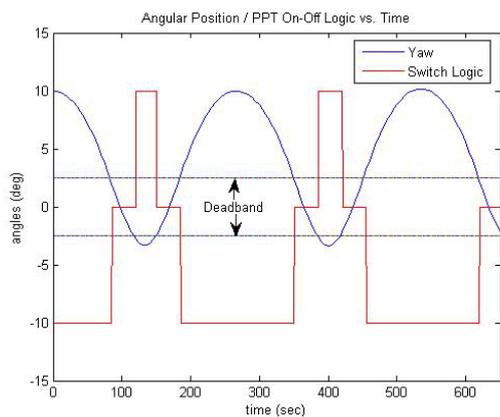


Figure 21: Soft-Limit Cycle for MPACS/PPTs

In both figures, the switch logic represents the PPT firing direction commanded by MPACS, where 10 equals “positive on”, -10 equals “negative on”, and 0 is “off”. In both, the deadband is 2.5 degrees off both sides of zero and MPACS is seen to at best act as a

proportional controller. For the hard-limit cycle, the PPTs are used less efficiently as they have to fire equally on both sides of the deadband. The soft-limit is more efficient as the disturbance torque prevents the yaw axis from going very far past the negative side of the deadband.

CONCLUSIONS

Because different pieces of the research finished at different stages of completion, quantitative comparison is difficult. However, qualitatively, there are a number of worthwhile conclusions seen.

Overall, PD control was very simple. Having only two gains that directly created the torque vector, it was easy to understand the effect of each gain. It was then easier to alter the gains to obtain the desired performance. However, the gains were merely chosen to obtain the desired results through educated guessing. No method was discovered to generate gain values that maximized performance for a given torque limit.

The PID controller is only slightly more complicated than the PD controller. Its third gain makes it less intuitive than the PD controller, but the effect of the gains is rather evident after a few tests. However, for the slight addition in complexity, the tests ran did not show any increase in performance over the PD and, once again, no method was discovered to maximize the gains for performance and torque limits.

Separating pitch from roll/yaw is a great method to start from as this method is taught at the undergraduate level. It can take advantage of second order approximations that graph performance specifications in the Laplace domain and other tools to augment design. However, this really is just a starting point. The s-plane characteristic equations make several assumptions, whose effects need to be quantitatively tested. Also, while requirements in ADCS terms are easy to graph in the s-plane, no simple relationships between real world requirements and those in ADCS terms were found.

As previously mentioned, LQR’s main advantages is that it guarantees closed loop stability when properly designed and can be easily modified for MIMO. Nevertheless, it is far too complex to be intuitive to the beginning user and is far too computationally intensive for FalconSat-3.

It is easy to see why a B-dot controller is a great start to commissioning a satellite from tip-off. It requires no attitude or attitude rate knowledge, only magnetometer measurements. Despite its low demands, the B-dot controller seems to be able to obtain excellent performance. However, this conclusion was made on the few test cases ran.

A spin rate controller is simple and able to constructively influence all three axes for a time and two for all time. Unfortunately, gaining control of the roll and pitch, in FalconSat-3's case, disturbs the yaw axis. This prohibits the spin rate controller from being used on its own. To create system stability, it had to be used in conjunction with the B-dot controller.

According to consulted sources, a cross product law controller is the only one that constructively incorporates a gravity gradient with magnetorquers for three-axes stabilization. It is more complicated than the B-dot or spin rate controllers, as it requires output from the attitude estimator as well as sensor data, but is able to align the orbital and body coordinate frames, which is necessary for the payloads during the satellite's operational lifetime. Further testing will lead to more optimized gains and better performance for FalconSat-3

Assuming, the PPTs are strong enough to do attitude maneuvers, they are limited to being used for proportional control. To obtain proper convergence to the commanded, they will have to be used in conjunction with another controller system and the magnetorquers.

In the end, while many of the control algorithms could be used to meet FalconSat-3's attitude control requirements, proper integration of the B-dot, spin rate, and cross product law controllers, when properly incorporated, should yield the best balance between competing performance characteristics.

RECOMMENDATIONS FOR FUTURE RESEARCH

Appendix A contains a picture of the current Simulink model and Figure 22 gives a simplified overview of the simulation's pieces.

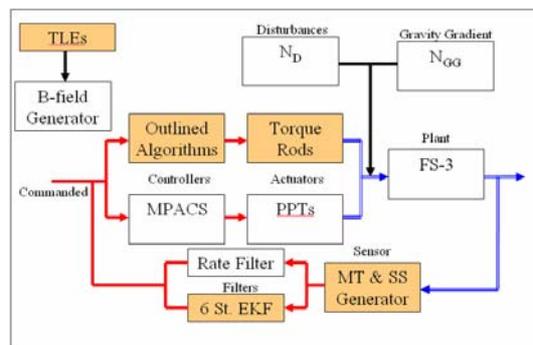


Figure 22: Simulation Overview

The colored blocks represent pieces of the simulation that are the most improperly implemented. Once fixed, proper analysis should be carried out so the different controllers' performances can be more quantitatively compared.

Ultimately, the best controller(s) should be implemented and the life of the satellite, from tip-off to decommissioning, properly simulated, so the best gains can be found through further testing.

ACKNOWLEDGEMENTS

I would like to thank everyone at the United States Air Force Academy and Surrey Space Centre that worked to make my summer internship in England possible. I would also like to express my personal gratitude to the following individuals that made the time in England not only tangibly productive but an outstanding personal and professional maturing experience: Dr. Vaios Lappas for his direction and supervision over the program, Dr. Yoshi Hashida for his expert advice, and Major David Richie and Lieutenant James Valpiani for their tireless patience in dealing with my endless stream of questions on a more than hourly basis.

REFERENCES

- [1] Anderson, Andrew. *No Bad Attitudes!: FalconSat-3 Attitude Determination and Control System Simulation and Analysis*. United States Air Force Academy: 18 August 2003.
- [2] Hale, Matthew J. *FS3 Kalman Filter Overview*. United States Air Force Academy: 6 April 2005.
- [3] Hashida, Yoshi. *ADCS for Future UoSat Standard Platform Revision 2: Surrey Satellite Technology Limited Internal Technical Note*. Guildford. Surrey Satellite Technology Limited: 24 August 2004.
- [4] Johnson, Andrea. *Analysis of the Feasibility of Demonstrating Pulsed Plasma*

Thrusters on FalconSat-3. United States Air Force Academy: Summer 2004.

[5] Kaplan, Marshall H. *Modern Spacecraft Dynamics and Control*. New York. John Wiley and Sons: 1976.

[6] Lappas, Vaivos J. *A Control Moment Gyro (CMG) Based Attitude Control System (ACS) For Agile Small Satellites*. Guildford. Surrey Space Centre: October 2002.

[7] Steyn, W. H. *A Multi-mode Attitude Determination and Control System for Small Satellites*. University of Stellenbosch: December 1995.

[8] Wertz, James R. and Wiley J. Larson, ed. *Space Mission Analysis and Design*. 3rd ed. Space Technology Library. El Segundo, CA.

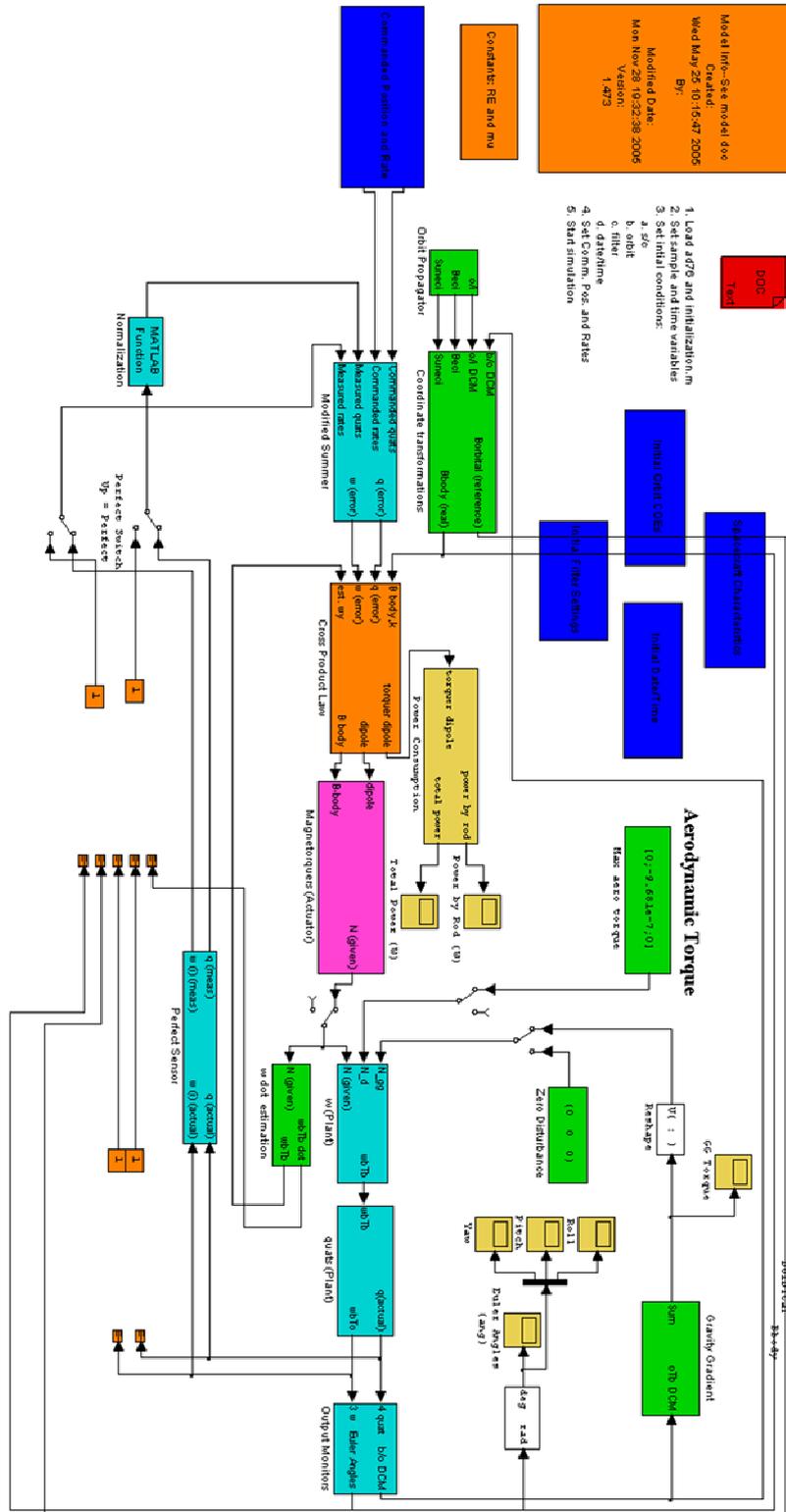
Microcosm Press and Kluwer Academic Publishers: 1999.

[9] Wie, Bong. *Space Vehicle Dynamics and Control*. U of Arizona. Reston. American Institute of Aeronautics and Astronautics, Inc: 1998.

Cadet First Class Paul Tisa is a senior cadet at the USAF Academy. He is an Astronautical Engineering major and has been working with the USAFA Small Satellite Program since January 2005. For eight weeks last summer, he worked with SSTL Staff at Surrey Space Centre in Guildford, UK.

APPENDICES

Appendix A: Simulation Picture



Appendix B: PD Test Battery

No.	time (sec)	samp (sec)	Initial Angles (°)			Init'l Rates			LO (%/s)	Comm. Angles (°) [R P Y]	Comm. Rates (%/s) LO [P Q R]	Nc	
			Roll	Pitch	Yaw	P	Q	R				Kp	Kd
0	10000	10	0	0	0	0	0	0	[0 0 0]	[0 0 0]	0	0	
1	1000	10	20	20	20	2	2	2	[0 0 0]	[0 0 0]	0	0	
2	10000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-1	1	
3	10000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	0	1	
4	500	0.5	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-1	0	
5	10000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.5	0.5	
6	10000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	0	0.5	
7	100	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.5	0	
8	10000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.25	0.25	
9	10000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	0	0.25	
10	10	10	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.025	0	
11	10000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.1	0.1	
12	10000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	0	0.1	
13	10	10	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.1	0	
14	20000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.05	0.05	
15	20000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	0	0.05	
16	10	10	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.05	0	
17	40000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.025	0.025	
18	40000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	0	0.025	
19	10	10	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.025	0	
20	100000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.01	0.01	
21	100000	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	0	0.01	
22	10	10	20	20	20	2	2	2	[0 0 0]	[0 0 0]	-0.01	0	
23	5.00E+5	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	⊙	⊙	
24	5.00E+5	1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	♣	♣	
25	5.00E+5	10	20	20	20	2	2	2	[0 0 0]	[0 0 0]	♥	♥	
26	5.00E+5	20	20	20	20	2	2	2	[0 0 0]	[0 0 0]	♦	♦	
27	3.00E+6	40	20	20	20	2	2	2	[0 0 0]	[0 0 0]	♠	♠	
28	1.50E+6	50	60	60	60	0	0	0	[0 0 0]	[0 0 0]	♠	♠	
29	1.50E+6	50	-60	-60	-60	0	0	0	[0 0 0]	[0 0 0]	♠	♠	
30	1.50E+6	50	75	60	45	0	0	0	[0 0 0]	[0 0 0]	♠	♠	
31	1.50E+6	50	60	45	75	0	0	0	[0 0 0]	[0 0 0]	♠	♠	
32	1.50E+6	50	45	75	60	0	0	0	[0 0 0]	[0 0 0]	♠	♠	
33	3.00E+6	10	0	0	0	6	6	6	[0 0 0]	[0 0 0]	♠	♠	
34	3.00E+6	20	0	0	0	-6	-6	-6	[0 0 0]	[0 0 0]	♠	♠	

not performed
<1E-7 ≈ 0

Notes:				
	qk		wk	
⊙	time = 0	error < .001	t = 0	error < .005
♣	0	1.00E-04	5.00E-02	5.00E-03
♥	0	1.00E-03	5.00E-03	5.00E-04
♦	0	1.00E-03	5.00E-03	2.00E-03
♠	0	1.00E-04	5.00E-03	5.00E-04

Appendix C: PID Test Battery

No.	Initial Angles (°)			Initial Rates LO (°/s)			Comm. Angles (°) [R P Y]	Comm. Rates (°/s) LO [P Q R]	Nc		
	Roll	Pitch	Yaw	P	Q	R			Kp	Ki	Kd
0	0	0	0	0	0	0	[0 0 0]	[0 0 0]	0	0	0
1	20	20	20	2	2	2	[0 0 0]	[0 0 0]	0	0	0
2	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E+00	2.50E-04	1.30E+01
3	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-02	2.50E-04	1.30E-01
4	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-02	2.50E-05	1.30E-01
5	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-02	2.50E-03	1.30E-01
6	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-02	2.50E-04	1.30E-02
7	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-02	2.50E-04	1.30E+00
8	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-03	2.50E-04	1.30E-01
9	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-01	2.50E-04	1.30E-01
10	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-03	2.50E-07	1.30E-02
11	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-03	2.50E-05	1.30E-02
12	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-04	2.50E-07	1.30E-02
13	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-03	2.50E-08	1.30E-02
14	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-03	2.50E-07	1.30E-03
15	20	20	20	2	2	2	[0 0 0]	[0 0 0]	3.50E-05	2.50E-09	1.30E-04
16	60	60	60	0	0	0	[0 0 0]	[0 0 0]	3.50E-05	2.50E-09	1.30E-04
17	-60	-60	-60	0	0	0	[0 0 0]	[0 0 0]	3.50E-05	2.50E-09	1.30E-04
18	75	60	45	0	0	0	[0 0 0]	[0 0 0]	3.50E-05	2.50E-09	1.30E-04
19	60	45	75	0	0	0	[0 0 0]	[0 0 0]	3.50E-05	2.50E-09	1.30E-04
20	45	75	60	0	0	0	[0 0 0]	[0 0 0]	3.50E-05	2.50E-09	1.30E-04
21	0	0	0	6	6	6	[0 0 0]	[0 0 0]	3.50E-05	2.50E-09	1.30E-04
22	0	0	0	-6	-6	-6	[0 0 0]	[0 0 0]	3.50E-05	2.50E-09	1.30E-04

<1E-7 ≈ 0