NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

ATTITUDE DETERMINATION OF A THREE-AXIS STABILIZED SPACECRAFT USING STAR SENSORS

by

Jay D. Vogt

December 1999

Thesis Advisor: Hal A. Titus

DTIC QUALITY INSPECTED 3

20000313 028

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 1999	3. REPORT TYPE AND DATES COVERED Engineer's Thesis

4. TITLE AND SUBTITLE Attitude Determination of a Three-Axis Stabilized Spacecraft Using Star Sensors	5. FUNDING NUMBERS
6. AUTHOR(S) Jay D. Vogt	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)* The purpose of this thesis is to investigate the application of a six-state discrete Kalman filter for estimates of angular rates based solely on star sensor data. The satellite is in a Molnyia orbit where orbital angular velocity and orbital angular acceleration are predetermined and stored in the on-board computer; such that they will be available each time a star observation is made. A two-axis star sensor will provide two angles to the estimator whereupon the third "unsensed" angle will be predicted; the rates about all three axes are then estimated. The results show that the rate estimates are accurate to within 10^{-7} r/s, which is equivalent to the data produced by gyroscopes.

14. SUBJECT TERMS: Kalman filter, Molnyia, MATLAB, SIMULINK, Three axis stabilization, Spacecraft, Satellite, Star Sensor, Estimation, Gyroscope	15. NUMBER OF PAGES 86
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-1

Approved for public release; distribution is unlimited

ATTITUDE DETERMINATION OF A THREE-AXIS STABILIZED SPACECRAFT USING STAR SENSORS

Jay D. Vogt

Lieutenant, United States Navy

B.S., University of Colorado, 1991

Submitted in partial fulfillment of the

requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

December 1999

Author: _____

Jay D. Vogt

Approved by: _____

Hal A. Titus, Thesis Advisor

_____

Brij Agrawal, Co-Advisor

_____

Gerald H. Lindsey, Chairman

Department of Aeronautics and Astronautics

ABSTRACT

The purpose of this thesis is to investigate the application of a six-state discrete Kalman filter for estimates of angular rates based solely on star sensor data. The satellite is in a Molnyia orbit where orbital angular velocity and orbital angular acceleration are predetermined and stored in the on-board computer; such that they will be available each time a star observation is made. A two-axis star sensor will provide two angles to the estimator whereupon the third "unsensed" angle will be predicted; the rates about all three axes are then estimated. The results show that the rate estimates are accurate to within 10^{-7} r/s, which is equivalent to the data produced by gyroscopes.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. OVERVIEW 1

II. CONTROL SYSTEM DESIGN SPECIFICATIONS 3

A. SATELLITE SPECIFICATIONS 3

B. SELF-IMPOSED SPECIFICATIONS 3

C. ASSUMPTIONS 3

D. CONTROL SYSTEM DESIGN CONSIDERATIONS 4

III. SPACE ENVIRONMENT 5

A. MOLNYIA ORBIT 5

B. DISTURBANCE TORQUES 8

IV. SATELLITE DYNAMICS AND KINEMATICS 9

A. REFERENCE FRAMES 9

B. STATE SPACE EQUATIONS OF MOTION 10

V. STAR SENSOR 13

A. STAR CHARACTERISTICS 13

B. HOW A STAR SENSOR WORKS 13

C. STAR SENSOR CHARACTERISTICS 14

D. STAR SENSOR OPERATION 14

VI. DISCRETE KALMAN FILTER 17

A. DERIVATION OF THE q MATRIX 18

B. KALMAN ALGORITHM 19

VII. PROPORTIONAL PLUS DERIVATIVE CONTROLLER 25

A. CONTROL LAWS 25

VIII. RESULTS 31

IX. SUMMARY AND CONCLUSION 37

A. SUMMARY 37

B. CONCLUSION 37

APPENDIX A: KINEMATICS 39

APPENDIX B: GRAVITY GRADIENT TORQUES 41

APPENDIX C: DERIVATION OF EQUATIONS OF MOTION 43
APPENDIX D: MATLAB CODE 45
LIST OF REFERENCES 67
INITIAL DISTRIBUTION LIST 69

LIST OF FIGURES

Figure 1: Attitude Control Diagram....2
Figure 2: Orbit Diagram....5
Figure 3: Orbital Characteristics....7
Figure 4: Orbital Reference Frame....10
Figure 5: Discrete Kalman Filter Loop....20
Figure 6: Plant Noise Covariance Elements....22
Figure 7: Error Covariance Elements....22
Figure 8: Controller Gains....28
Figure 9: Controller Rate Gains....28
Figure 10: Satellite Attitude....31
Figure 11: Roll Response with Measurements....32
Figure 12: Pitch Response with Measurements....32
Figure 13: Yaw Response with Measurements....33
Figure 14: Attitude Rates....33
Figure 15: Actual and Estimate Attitude Rates....34
Figure 16: Reaction Wheel Torques....35
Figure 17: Reaction Wheel Momentum....35

LIST OF TABLES

Table 1: Orbit Parameters 6
Table 2: Star Tracker Characteristics 14
Table 3: Star Sensor Measurements 15
Table 4: Kalman Filter Definitions 18

LIST OF SYMBOLS

μ	Gravitational Constant of the Earth
i	Orbit Inclination
r_p	Radius of Perigee
r_a	Radius of Apogee
e	Orbit Eccentricity
p	Semi-Latus Rectum
Π	Orbit Period
ν	True Anomaly
Ω	Orbital Angular Velocity
ϕ	Roll Error
$\dot{\phi}$	Roll Rate
θ	Pitch Error
$\dot{\theta}$	Pitch Rate
ψ	Yaw Error
$\dot{\psi}$	Yaw Rate
$\bar{T}_{gg}$	Gravity Gradient Disturbance Torque Vector
$\bar{T}_{sp}$	Solar Pressure Disturbance Torque Vector
I	Principal Moment of Inertia Tensor
$\bar{h}$	Reaction Wheel Angular Momentum Vector
k_v	Velocity Feedback Gain
k	Position Feedback Gain
ω_n	Natural Frequency
ζ	Damping Factor
ϕ_{ss}	Steady State Roll Error
θ_{ss}	Steady State Pitch Error
ψ_{ss}	Steady State Yaw Error
A	Plant Matrix

B	Control Matrix
$\vec{u}$	Control Input Vector
F	Control Gain Matrix
$\vec{u}_d$	Disturbance Torque Vector Forcing Function
Φ_k	State Transition Matrix

ACKNOWLEDGEMENTS

The author would like to thank the following people for their invaluable guidance and assistance provided in this thesis:

Professor Hal Titus, Naval Postgraduate School

Professor Brij Agrawal, Naval Postgraduate School

I. INTRODUCTION

The space industry has grown at an alarming rate over the last decade and should continue to do so in the future. Information and data received from satellites is generally taken for granted. Only when a critical satellite outage occurs is society rudely reminded of their ever-increasing dependence on these vehicles. For example, the premature and unexpected failure of Galaxy IV temporarily left millions of people without pager service. This dependence translates into big business. Therefore, when a satellite suffers a failure that threatens its life expectancy, every effort will be made to save it.

Recently, the SOHO spacecraft tumbled out of control after suffering from multiple gyroscope failures. As a result, European engineers were forced to create a software package that would over-ride the failed hardware. It took six months to write and test the code, but it was time well spent, as control of the billion dollar spacecraft was once again regained after a successful up-link. SOHO is now able to autonomously maintain proper attitude relative to the sun using its star tracker as the primary control sensor. As a matter of fact, this may be a sign of things to come considering that the reliability of a gyroscope decreases significantly with time. Even though the addition of redundant gyroscopes will serve to increase reliability, it will also increase cost, complexity and mass. SOHO is testimony to the fact that, in certain cases, software is more feasible than hardware.

Manufacturers of gyroscopes will obviously be opposed to the demise of their hardware, but with improvements in both on-board processing and star sensor capabilities, this area of investigation can no longer be ignored. To this end, the purpose of this thesis is to determine the utility of a six-state, discrete Kalman filter in the estimation of satellite attitude.

A. OVERVIEW

In order to test the Kalman filter, an attitude control system must be developed. To help in this development, certain design criteria were mandated while others were self-imposed. Chapter II will summarize these requirements and assumptions. The proposed control flow of the attitude control system is shown below in Figure 1.

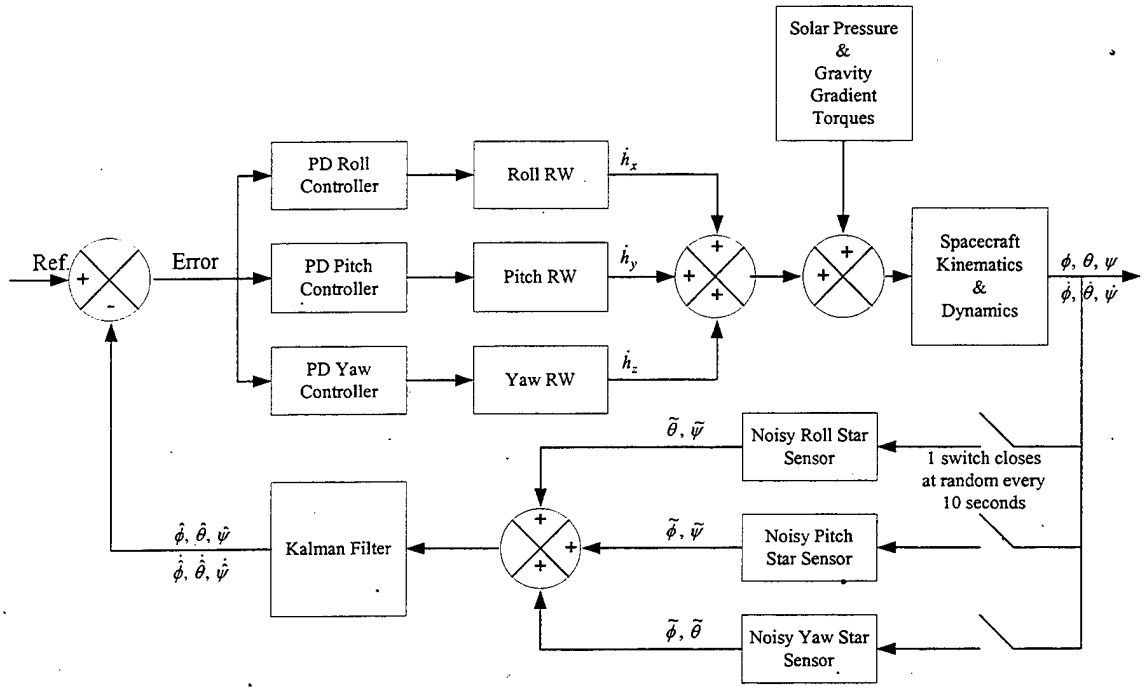

Figure 1: Attitude Control Diagram

Disturbance moments will force the vehicle dynamics and will be briefly discussed in Chapter III; in addition, the orbital equations of motion will also be given in this chapter. The spacecraft equations of motion will be the topic of Chapter IV. The simulation will progress at discrete ten second intervals, which will be the sampling time of each star sensor to be described in Chapter V. Once a star sensor makes a noisy measurement, that information will be sent to the discrete Kalman filter where both attitude and attitude rates will be estimated; Chapter VI will describe this process in more detail. The estimated output of the filter will be the input to the proportional plus derivative controller outlined in Chapter VII. The results and conclusion will follow accordingly.

II. CONTROL SYSTEM DESIGN SPECIFICATIONS

As stated in the introduction, the purpose of this thesis is to develop a suitable attitude estimator based on star sensor measurements. The attitude control system will be designed according to the following parameters.

A. SATELLITE SPECIFICATIONS

- Molnyia orbit
- Three star sensors aligned with the body axes
- ~10 second star sensor sampling time
- 1 star present in star sensor field of view (FOV) at all times
- 1 star sensor chosen at random at each time step
- Roll and pitch inertia=25,000 kg-m^2, yaw inertia=15,000 kg-m^2
- Kalman filter for rate estimation

B. SELF-IMPOSED SPECIFICATIONS

- Nadir pointing to within 0.1° of orbital reference frame
- 4 arc-second noise level for each star sensor
- 3-axis stabilized
- 10-year design life

C. ASSUMPTIONS

- Small angle approximations
- Orbital angular velocity and acceleration known for each sensor measurement
- Negligible cross products of inertia
- Constant solar pressure moments
- Each reaction wheel is independently controlled
- No slewing requirement
- Control law updates performed at 10 second intervals
- Satellite is modeled as a rigid body

D. CONTROL SYSTEM DESIGN CONSIDERATIONS

As can be seen from the previous sections, considerable latitude has been given in the design of this control system. In order to achieve 0.1° pointing accuracy, a zero-momentum system, consisting of three reaction wheels whose momentum vectors coincide with the body axes, will be used. These reaction wheels will each be independently controlled with its own, dedicated proportional plus derivative (PD) controller. Control of the satellite will be difficult at perigee due to its high orbital angular velocity; consequently, the gains of the pitch controller will have to be adjusted accordingly. As disturbance moments cause errors in attitude, off-axis components of reaction wheel angular momentum will cause internal torques that will have to be accounted for. The star sensor described in Section A of this chapter is only able to sense errors about two axes, which means that the Kalman filter will have to predict the error about the "un-sensed" axis. Not only will the filter be used to estimate position and rates, but it will also mitigate sensor noise. Crucial to this design is the assumption that the orbital angular velocity and acceleration are known for each star tracker observation.

III. SPACE ENVIRONMENT

A. MOLNYIA ORBIT

Countries located in high latitudes are forced to place communication satellites in highly eccentric orbits known as Molnyia orbits. These orbits, widely used by the Soviet Union, have the following characteristics: 63.5° inclination, elliptical, 11-12 hour period. Due to the high eccentricity of this orbit, the spacecraft has a long dwell time over the area of interest.

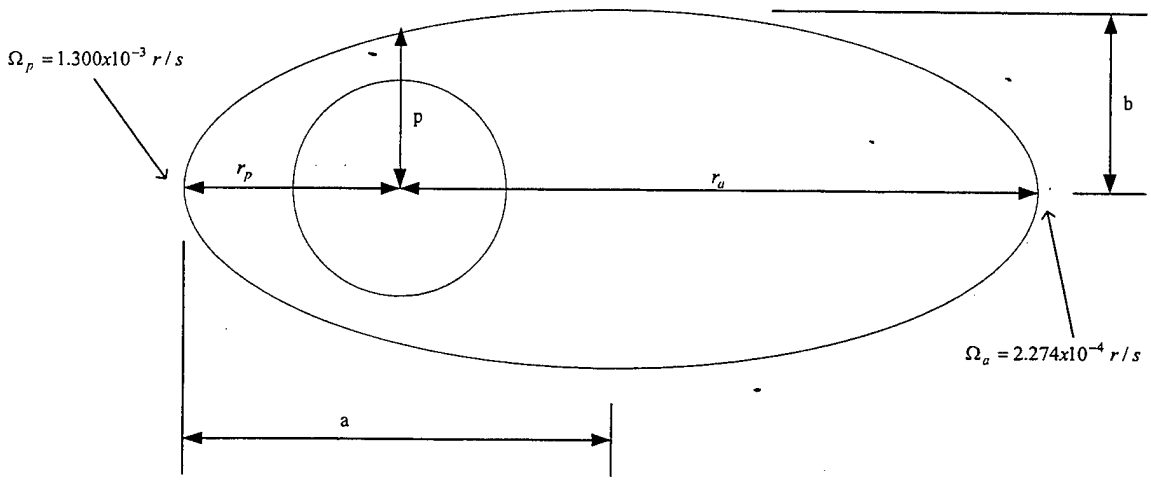

Figure 2: Orbit Diagram

The inclination of this orbit is chosen such that perigee will remain fixed over Antarctica. In order to provide continuous coverage, at least three satellites need to be appropriately phased. Table 1 lists the characteristics of this orbit.

Title	Symbol/Equation	Quantity
Earth's Gravitational Constant	μ	398601 km^3/sec^2
Inclination	i	63.5°
Radius of Perigee	r_p	7378.15 km
Radius of Apogee	r_a	42164.17 km

Semi-Major Axis	$a = \frac{r_p + r_a}{2}$	24771.16 km
Eccentricity	$e = \frac{r_a - r_p}{r_a + r_p}$	0.70215
semi-latus rectum	$p = a(1-e^2)$	12558.62 km
period	$\Pi = 2\pi\sqrt{\frac{a^3}{\mu}}$	38799.86 sec

Table 1: Orbit Parameters

Of particular interest is the orbital angular velocity of the spacecraft at any time, t. These values will be calculated in discrete, ten-second intervals and stored as deterministic values in the on-board computer. The radius, as a function of time, is given by

$$r(t) = \frac{p}{1 + e\cos(\nu(t))} \tag{1}$$

The true anomaly, ν, is just the angle measured from perigee to the satellite's current position. Taking both the first and second derivatives [Ref. 1], the following expressions are obtained

$$\dot{r}(t) = \sqrt{\frac{\mu}{p}} e \sin(\nu(t)) \tag{2}$$

$$\ddot{r}(t) = \sqrt{\frac{\mu}{p}} e \cos(\nu(t))\dot{\nu}(t) \tag{3}$$

The rate of change of true anomaly, $\dot{\nu}$, is just the orbit angular rate of the satellite. Both the orbital angular rate and the orbital angular acceleration can be found in [Ref. 1]. They are given by

$$\dot{v}(t) = \sqrt{\frac{\mu}{p}}\frac{1}{r(t)}\left(1 + e\cos(v(t))\right) \tag{4}$$

$$\ddot{v}(t) = -\sqrt{\frac{\mu}{p}}\frac{1}{r(t)^2}\left[\sqrt{\frac{\mu}{p}}e\sin(v(t))\left(1 + e\cos(v(t))\right) + re\sin(v(t))\dot{v}(t)\right] \tag{5}$$

In order to solve for both the true anomaly and radius at any time, t, it is necessary to convert the above equations into two first order differential equations. If the following variable substitutions are made: $y_1 = r$, $y_2 = \dot{r}$, $y_3 = v$, and $y_4 = \dot{v}$, then they can be substituted in Equations (1) through (5). Once the substitution is made, the resulting first order differential equations can be integrated using a Runge-Kutta integration method. The integration was performed using MATLAB 5.2, and the results are shown in Figure 3.

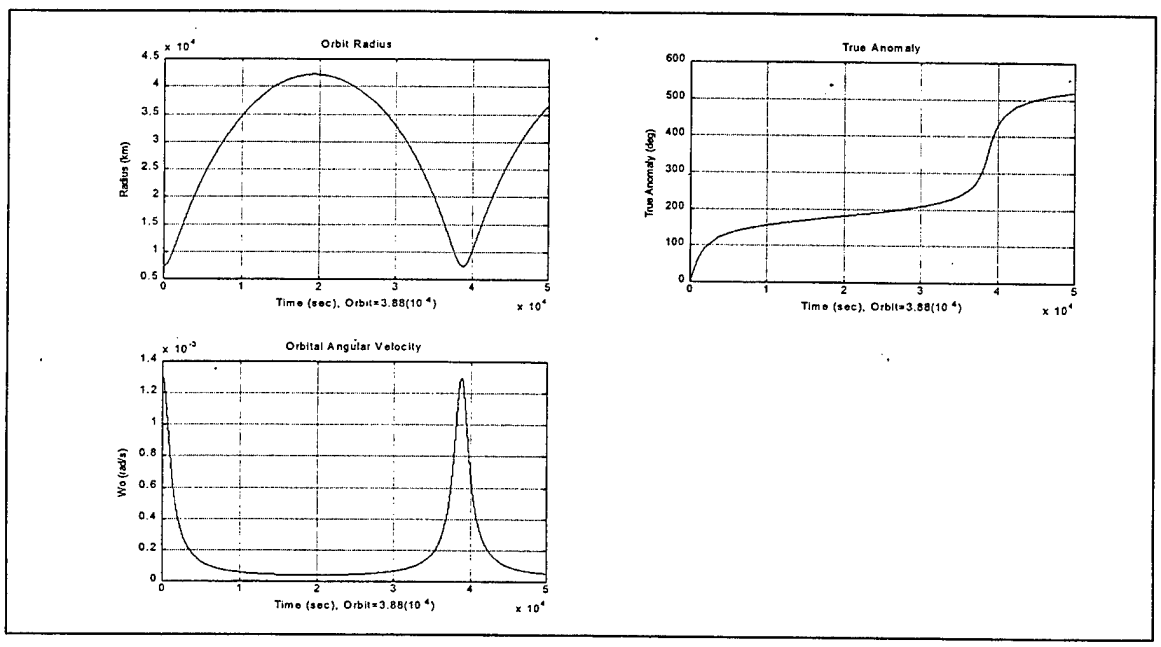

Figure 3: Orbital Characteristics

The simulation begins at perigee and continues for an entire orbit. As can be seen, the orbital angular velocity is nearly constant as the spacecraft dwells near apogee.

B. DISTURBANCE TORQUES

At first glance, it may seem that the vacuum of space is a benign environment. However, this is not true. External disturbance moments will cause errors in the spacecraft's attitude. These errors however, will be kept within the required pointing limits if the attitude control system is properly designed. The four major disturbance moments worth consideration are 1) Solar Pressure 2) Gravity Gradient 3) Magnetic Moment and 4) Aerodynamic. While the last three disturbance moments are significant at perigee, they are insignificant at apogee. Since the satellite will spend the majority of its time at or near apogee, the disturbance torques will be modeled for this case. As a consequence, both magnetic and aerodynamic moments will be discounted. For design simplicity, it will be assumed that the solar pressure moment can be modeled as a constant torque about each body axis. Although gravity gradient moments are relatively insignificant at apogee, they can be incorporated into the satellite equations of motion with minimal effort. These moments, derived in Appendix A, were found to be

$$T_{ggx} = 3\Omega^2 \phi (I_z - I_y)$$

$$T_{ggy} = 3\Omega^2 \theta (I_z - I_x) \tag{6}$$

$$T_{ggz} = 0$$

Since I_z is less than both I_x and I_y, this satellite will be gravity gradient friendly. This could become important during safe mode operations. Although this spacecraft is gravity gradient friendly, the symmetry about the roll and pitch axes compromises yaw stability. As long as the attitude control system remains operational, however, yaw stability will not be a factor.

IV. SATELLITE DYNAMICS AND KINEMATICS

There are many types of transformation methods, the most popular are: direction cosine matrices (DCM), euler angles, and quaternions. Quaternions are popular since they involve only a single rotation about an eigen-axis. This greatly reduces the cumbersome mathematics that is characteristic of the other methods. However, if small angle approximations are made and if second order terms are set to zero, then DCM's are easy to employ; they will be used in this analysis. Transformations from one frame to another are performed to facilitate calculations. For example, the latitude and longitude of stars in the star catalog have all been programmed within a celestial frame, but measurements will be made in the body frame. Therefore, proper attitude determination relies on a simple transformation.

Satellite dynamics refers to the motion of the body when subject to both internal and external disturbance moments. The assumption has been made that the spacecraft will rotate about its principal moments of inertia. This is a reasonable assumption since the off-axis inertias can be significantly reduced by strategic placement of satellite components and hardware.

A. REFERENCE FRAMES

In the field of attitude control, it is often required to express an inertial quantity as a body frame quantity. For example, the inertial angular velocity derived from the Euler moment equations must be expressed in body coordinates and then integrated to get the Euler angles. Three important reference frames are used in the derivation of equations of motion: 1) body frame 2) orbital frame and 3) inertial frame. The origin of these three frames will all be located at the spacecraft's center of mass. In the orbital reference frame, the z-axis points at the center of the Earth, the x-axis points in the satellite's direction of motion, and the y-axis is normal to the orbit plane, completing the right-hand set. The body frame is attached to the spacecraft where the Euler angles represent the deviation of the body frame from the orbital reference frame. On-board sensors measure these Euler angles. The inertial frame remains fixed in Earth space such that the inertial y-axis coincides with the orbital y-axis. An additional reference frame alluded to earlier

is the celestial frame. The z-axis of this frame points north and the x-axis points in the direction of the vernal equinox. Although the x-axis precesses (very slowly), it assumed to be fixed in space.

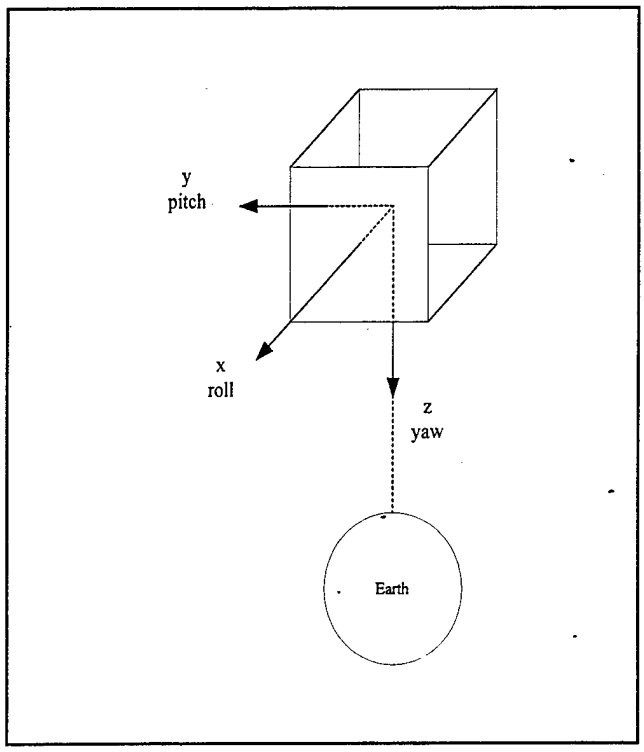

Figure 4: Orbital Reference Frame

B. STATE SPACE EQUATIONS OF MOTION

The equations of motion derived in Appendix C completely describe the motion of the satellite when subjected to both internal and external disturbance moments. If the body accelerations are solved for in Equation (55) the following result is obtained

$$\ddot{\phi} = \left[\frac{-4\Omega^2\left(I_y - I_z\right)\phi + \Omega h_y \phi - \Omega\left(-I_x + I_y - I_z\right)\dot{\psi} + h_y\dot{\psi} - h_z\dot{\theta} + I_x\dot{\Omega}\psi}{I_x}\right] - \left[\frac{\dot{h}_x}{I_x}\right] + \left[\frac{T_{spx} + \Omega h_z}{I_x}\right]$$

$$\ddot{\theta} = \left[\frac{-3\Omega^2\left(I_x - I_z\right)\theta - h_x\dot{\psi} - \Omega h_z\psi - \Omega h_x\phi + h_z\dot{\phi}}{I_y}\right] - \left[\frac{\dot{h}_y}{I_y}\right] + \left[\frac{T_{spy} + I_y\dot{\Omega}}{I_y}\right] \qquad (7)$$

$$\ddot{\psi} = \left[\frac{-\Omega^2\left(-I_x+I_y\right)\psi + \Omega h_y \psi - \Omega\left(I_x - I_y + I_z\right)\dot{\phi} + h_x\dot{\theta} - h_y\dot{\phi} - I_z\dot{\Omega}\phi}{I_z}\right] - \left[\frac{\dot{h}_z}{I_z}\right] + \left[\frac{T_{spz} - \Omega h_x}{I_z}\right]$$

For computational reasons, it is desirable to reduce these second order equations to first order equations by making the following state variable substitutions

$$\bar{x} = [\phi \quad \dot{\phi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi}]^T \tag{8}$$

With these definitions, we can express the satellite dynamic equations into the following matrix form

$$\dot{\bar{x}} = A\bar{x} + B\bar{u} \tag{9}$$

A is the plant matrix and it is given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-4\Omega^2(I_y-I_z)+\Omega h_y}{I_x} & 0 & 0 & \frac{-h_z}{I_x} & \dot{\Omega} & \frac{-\Omega(-I_x+I_y-I_z)+h_y}{I_x} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{-\Omega h_y}{I_y} & \frac{-h_z}{I_y} & \frac{-3\Omega^2(I_x-I_z)}{I_y} & 0 & \frac{-\Omega h_z}{I_y} & \frac{-h_x}{I_y} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\dot{\Omega} & \frac{-\Omega(I_x-I_y+I_z)-h_y}{I_z} & 0 & \frac{h_x}{I_z} & \frac{-\Omega^2(-I_x+I_y)+\Omega h_y}{I_z} & 0 \end{bmatrix} \tag{10}$$

B is the control matrix given by

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

u is going to be the input reaction wheel control, and it will have the following form

$$u = -Fx + u_d \tag{12}$$

F will be the PD controller gain matrix and u_d will represent the summation of the solar pressure moments and the internal reaction wheel moments. F and u_d are given by

$$F = \begin{bmatrix} \frac{k_x}{I_x} & \frac{k_{vx}}{I_x} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{k_y}{I_y} & \frac{k_{vy}}{I_y} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{k_z}{I_z} & \frac{k_{vz}}{I_z} \end{bmatrix} \tag{13}$$

$$u_d = \begin{bmatrix} \frac{T_{spx} + \Omega h_z}{I_x} \\ \frac{T_{spy} + I_y \dot{\Omega}}{I_y} \\ \frac{T_{spz} - \Omega h_x}{I_z} \end{bmatrix} \tag{14}$$

Substituting Equation (14) into Equation (9), the following equation of motion is obtained

$$\dot{\bar{x}} = (A - BF)x + Bu_d \tag{15}$$

Equation (15) is equivalent to the equations of motion derived in Appendix C.

V. STAR SENSOR

Star sensors were chosen because of their high accuracy and because of their ability to determine yaw. Earth horizon sensors can sense only roll and pitch. Even though sun sensors can measure yaw, four or more would be required for continuous information; furthermore, they become ineffective while in Earth's umbra. Historically, star sensors have been used only to update gyroscopic drift estimates, but recent advancements in star sensor technology make it possible to consider practical implementations of attitude determination based solely on star sensor data [Ref. 8].

A. STAR CHARACTERISTICS

For the purpose of this simulation, it will be assumed that there is a uniform distribution of stars in the celestial sphere, such that one star will be present in the sensor FOV at all times. Each star can be classified according to its magnitude and its spectra. The magnitude refers to a star's brightness as seen from Earth. The magnitude of Vega, a bright star in the constellation Lyra, has been assigned a value of zero. All other stars are referenced to Vega logarithmically. As an example, the brightest star outside our solar system, Sirius, has a magnitude of -1.6. Astronomers can distinguish different stars, not only from their magnitude, but also from their unique spectral characteristics. There are seven principal spectral categories: O, B, A, F, G, K, and M and these categories are further subdivided into ten more subgroups, from 0 to 9. [Ref. 3]

B. HOW A STAR SENSOR WORKS

There are three important phases in the determination of attitude using star trackers 1) star identification 2) tracking 3) processing [Ref. 3]. When a satellite completes an orbit, the star sensor will have imaged stars that make up a ring of the celestial sphere. The width of this celestial ring is a function of the sensor FOV. Although accuracy will increase with a small FOV, the time between observations will increase. Once the FOV is decided upon, the celestial latitudes and longitudes of chosen stars, within the sensor celestial ring, are recorded into the star catalog. Therefore, at any point in the orbit, the star sensor will expect to see a certain star. Identification of this

star is made if it falls within a circle of tolerance; if two stars appear in this circle, magnitude and spectral characteristics will separate the two. Once the star is identified, it is tracked until it leaves the FOV. While the sensor is tracking an identified star, however, the measured latitude and longitude is compared with the actual latitude and longitude. This error is sent, via the Kalman filter, to the appropriate controller in the form of a voltage signal, where a corrective torque is subsequently applied.

C. STAR SENSOR CHARACTERISTICS

The star sensor model used in this simulation has been designed in accordance with the specifications outlined in Chapter II. Further assumptions have been made for the sake of completeness, and they will be, in part, consistent with current technology.

Technology	Charged Coupled Device (CCD)
FOV	10° x 10°
Accuracy	~10 arcsec
# Stars in Catalog	4000
Sampling Rate	0.1 Hz (current technology is faster)
Noise	4 arcsec (magnitude=6)
Solar Exclusion Angle	30° w/sun shade

Table 2: Star Tracker Characteristics

The noise level is inherent to the star tracker and it is treated as a zero-mean, Gaussian white sequence.

D. STAR SENSOR OPERATION

Three-axis attitude determination requires two separate line of sights (LOS) with angular separation near 90° for increased accuracy. In this simulation, the optical axis of each star sensor will be aligned with the body axes. At each discrete time step a star sensor will be selected at random for attitude determination. Although this sequence of events permits only one LOS per time step, attitude is readily determined over

consecutive time steps. Since only one star will be in the sensor FOV at any particular time, measurements can only be made about two axes. Table 3 summarizes the angles that can be sensed by each star sensor.

Star Sensor (Direction of Optical Axis)	Angles Sensed
Roll	Pitch, Yaw
Pitch	Roll, Yaw
Yaw	Roll, Pitch

Table 3: Star Sensor Measurements

VI. DISCRETE KALMAN FILTER

A six state discrete Kalman filter has been chosen to estimate both position and rates from noisy star sensor data. The Kalman filter that will be used in the simulation is represented by

$$\begin{aligned} \vec{x}_{k+1} &= \Phi_k \vec{x}_k + \Delta_k \vec{u}_k + \vec{w}_k \\ \vec{z}_k &= H\vec{x}_k + \vec{v}_k \end{aligned} \tag{16}$$

The white sequence, w_k, for the plant has a covariance, Q, while the sensor noise, v_k, has a covariance, R. Noise from the star sensor is affected by the magnitude of the star; a bright star is noisier than a dim star [Ref. 4]. The sensor noise covariance is defined as follows

$$R_k = E\left[\vec{v}_k \vec{v}_k^T\right] \tag{17}$$

Table 4 is a summary of all the matrices and their respective dimensions that will be used in the estimation process.

Symbol	Definition	Size
$\vec{x}_k$	state vector	6 x 1
Φ_k	state transition matrix	6 x 6
Λ_k	deterministic weighting matrix	6 x 3
$\vec{u}_k$	deterministic forcing function	3 x 1
$\vec{w}_k$	plant white noise sequence	6 x 1
Q_k	plant noise covariance matrix	6 x 6
$\vec{z}_k$	measurement vector	3 x 1
H_k	feedback sensitivity matrix	3 x 6
$\vec{v}_k$	sensor white noise sequence	3 x 1

R_k	sensor noise covariance matrix	3 x 3
P_k	error covariance matrix, $E\left[\left(\bar{x}_k - \hat{\bar{x}}\right)\left(\bar{x}_k - \hat{\bar{x}}\right)^T\right]$ (accuracy of estimate)	6 x 6
$\hat{x}_k$	optimal estimate of $\bar{x}_k$ at time, t_k, based on current measurement	6 x 1
$\hat{\bar{x}}_k^-$	estimate of $\bar{x}_k$ at time, t_k, just prior to measurement	6 x 1

Table 4: Kalman Filter Definitions

A. DERIVATION OF THE Q MATRIX

Solving for the covariance of the plant noise is no trivial matter. In this simulation, the Q matrix will vary with each time step. The formal definition of the plant noise covariance is given by

$$Q_k = E\left[\bar{w}_k \bar{w}_k^T\right] \tag{18}$$

It can be shown that Equation (18) must satisfy the following matrix differential equation [Ref. 5]

$$\dot{Q}_k = A_{aug} Q_k + Q_k A_{aug}^T + BWB^T \tag{19}$$

The augmented A matrix is defined from Equation (15) as the quantity, A-BF and the power spectral density matrix associated with the forcing function $\bar{u}$ is denoted by W. The solution to Equation (19) is greatly simplified for the time invariant case. It proceeds as follows

$$\alpha = \begin{bmatrix} -A_{aug} & BWB^T \\ 0 & B^T \end{bmatrix} \Delta t \tag{20}$$

By taking the matrix exponential of Equation (20), the following result is obtained

$$\beta = \begin{bmatrix} \cdots & \chi^{-1} Q_k \\ 0 & \chi^T \end{bmatrix} \tag{21}$$

The upper left partition can be neglected for this analysis. The plant noise covariance matrix can now be determined by multiplying the upper right partition of Equation (21) by χ. This method was first formulated by Van Loan in 1978 and can be found in [Ref. 5].

B. KALMAN ALGORITHM

Before entering the Kalman filter loop, an initial estimate, $\hat{x}_0^-$, and its error covariance, P_0^-, must chosen. For this simulation, these initial conditions were chosen to be

$$\hat{x}_0^- = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$P_0^- = 10^{-12} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

The '-' superscript will represent the predicted estimate while the '^' notation denotes estimation. The discrete Kalman filter is, in essence, just a computer algorithm that derives optimal estimates from discrete measurements. Although there are different forms of the discrete Kalman filter, the most fundamental form starts with the Kalman gain calculation, which is given by

$$G_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{23}$$

The value of this gain matrix will vary with each time step. Next, it is required to update the estimate using star sensor data, and also it is required to determine the accuracy of this new estimate. The equations are given as

$$\hat{\bar{x}}_k = \hat{\bar{x}}_k^- + G_k\left(\bar{z}_k - H_k \hat{\bar{x}}_k^-\right) \tag{24}$$

$$P_k = (I - G_k H_k) P_k^- \tag{25}$$

Figure 5, shown below, illustrates the recursive nature of the discrete Kalman filter. A favorable characteristic of any recursive filter is that there is no need to store past measurements [Ref. 6].

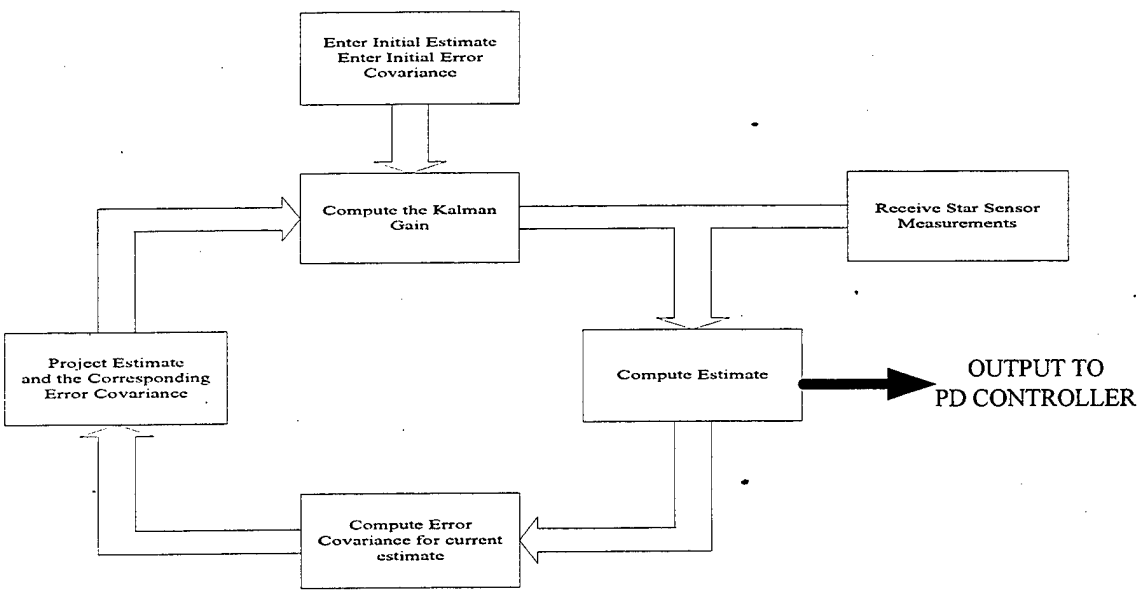

Figure 5: Discrete Kalman Filter Loop

Equation (24) is the actual output of the discrete Kalman filter. It estimates both attitude angles and attitude rates given only star sensor angle information. Not only does it derive rates, but it also mitigates sensor noise effects. Lastly, it is necessary to project ahead and estimate the state for the next time step. The predictive equations are as follows

$$\hat{\bar{x}}_{k+1}^{-} = \Phi_k \hat{\bar{x}}_k + \Delta_k \bar{u}_k \tag{26}$$

$$P_{k+1}^{-} = \Phi_k P_k \Phi_k^T + Q_k \tag{27}$$

It is interesting to note that in Equation (26), the deterministic forcing function has been included. This forcing function consists of known reaction wheel moments, which can be measured by the reaction wheel motor assembly. If this deterministic term is not included, the rate estimator is unable to accurately estimate satellite-rates near perigee.

For the purpose of analysis and proper tuning, it is helpful to look at the time-varying nature of both the Q and P matrices over a period of one orbit. Since the off-diagonal elements of these matrices are small, only the diagonal elements will be examined. These elements are shown in Figure 6 and Figure 7.

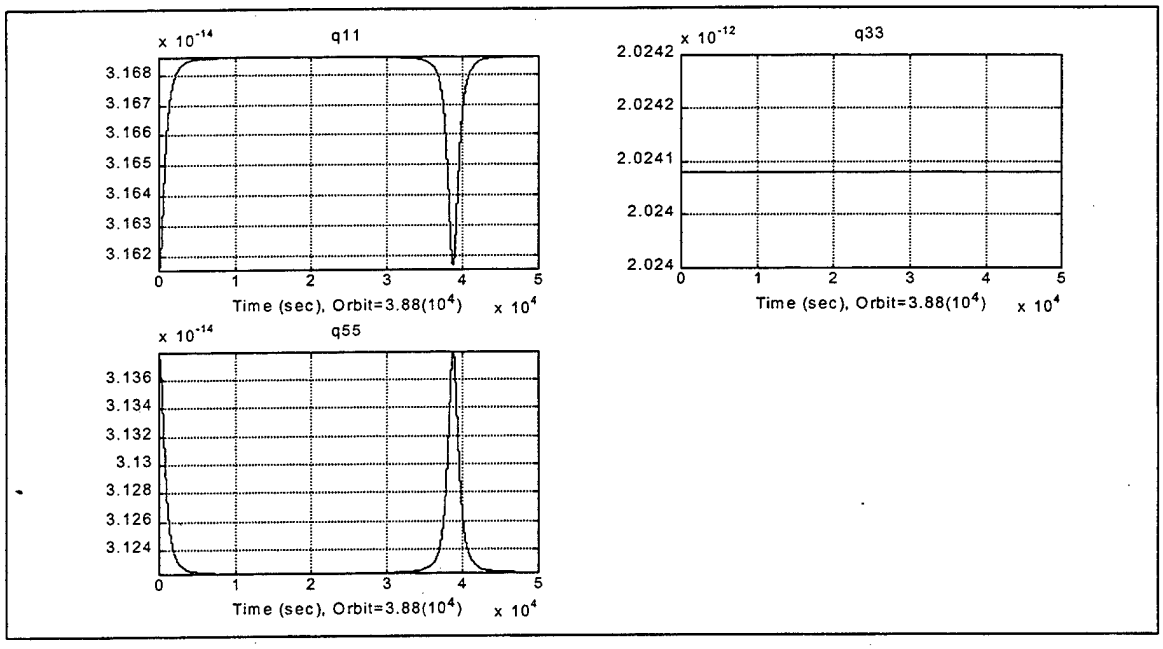

Figure 6: Plant Noise Covariance Elements

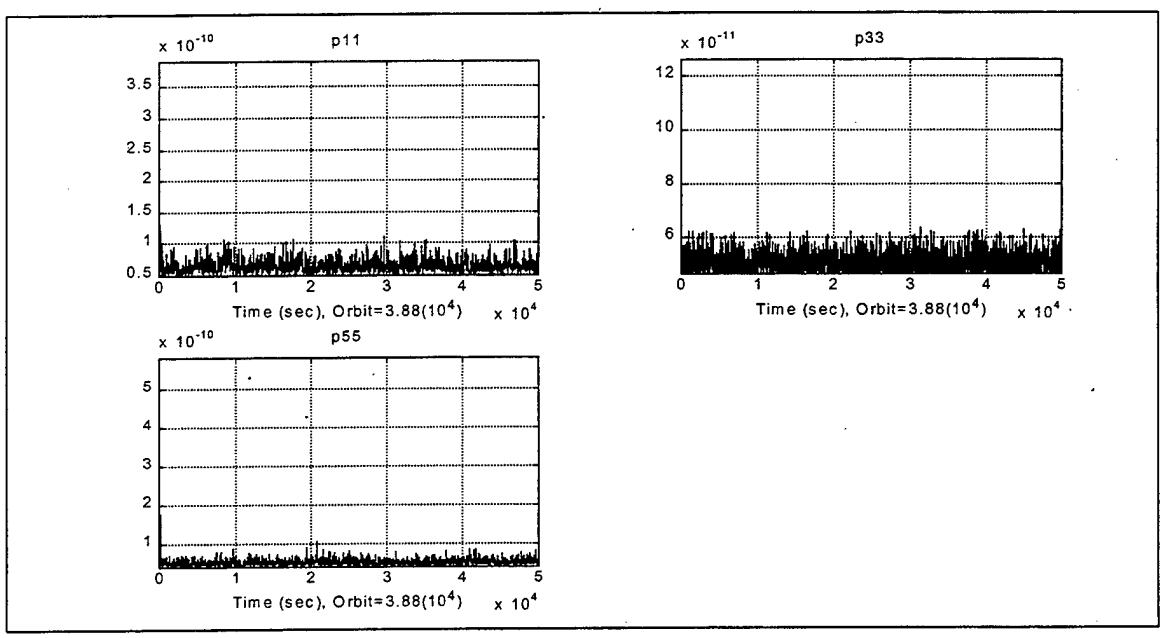

Figure 7: Error Covariance Elements

If Q is decreased, the filter will have a tendency to track the predicted estimate. On the other hand, if Q is increased, the filter will track the measurements. It is important to find the right balance because if Q is too low, then the filter will have a hard

time tracking when or if the satellite maneuvers, but if Q is too high, pointing accuracy will suffer from sensor noise effects. The results of the Kalman filter will be shown in Chapter VII.

VII. PROPORTIONAL PLUS DERIVATIVE CONTROLLER

A PD controller was chosen because of its simplicity. Although this type of controller does not reduce or correct steady state errors, the gains can be adjusted to ensure the error is within acceptable limits. A separate controller will be assigned to each reaction wheel.

A. CONTROL LAWS

Many types of control laws are available which can conceivably satisfy this satellite's pointing requirements. Some common control laws are 1) proportional 2) proportional plus derivative 3) proportional plus integral plus derivative and 4) optimal. Each of these controllers has its own unique characteristics; however, as long as the controller maintains proper spacecraft attitude, exotic controllers will not be required. In fact, it will be shown that the gains of a simple PD controller can be adjusted to minimize overshoot and settling time. Each of the reaction wheels in this spacecraft will have its own PD controller and they are represented as

$$\begin{aligned} \dot{h}_x &= k_{vx}\dot{\phi} + k_x\phi \\ \dot{h}_y &= k_{vy}\dot{\theta} + k_y\theta \\ \dot{h}_z &= k_{vz}\dot{\psi} + k_z\psi \end{aligned} \tag{28}$$

These control laws are expressed as the rate of change of reaction wheel angular momentum, or reaction wheel torque, and they are part of the feedback loop. As can be seen, these internal torque equations are a function of the measured Euler angles and rates. The gain constants are found by substituting Equation (28) into Equation (55), which is located in Appendix C. If the resulting set of equations is completely de-coupled, and the Laplace transform is taken, the following result is obtained

$$\frac{\Phi(s)}{T_x(s)} = \frac{\frac{1}{I_x}}{s^2 + \frac{k_{vx}}{I_x}s + \frac{4\Omega^2(I_y - I_x) - \Omega h_y + k_x}{I_x}}$$

$$\frac{\Theta(s)}{T_y(s)} = \frac{\frac{1}{I_y}}{s^2 + \frac{k_{vy}}{I_y}s + \frac{3\Omega^2(I_x - I_z) + k_y}{I_y}} \tag{29}$$

$$\frac{\Psi(s)}{T_z(s)} = \frac{\frac{1}{I_z}}{s^2 + \frac{k_{vz}}{I_z}s + \frac{\Omega^2(-I_x + I_y) - \Omega h_y + k_z}{I_z}}$$

For this particular analysis, it is assumed that the orbital angular velocity is locally constant. The objective is to determine suitable position and rate feedback gains that will increase spacecraft robustness. The nominal characteristic equation for any second order system has the following form

$$\Lambda(s) = s^2 + 2\omega_n \zeta s + \omega_n^2 \tag{30}$$

The natural frequency is denoted as ω_n and ζ is the damping factor, which will be chosen to be one. Each of the denominators in Equation (29) will be equated to Equation (30). Solving for the coefficients, the result is two equations and three unknowns. The third equation makes use of the final value theorem [Ref. 7], and it is given by the following

$$f(\infty) = \lim_{t \to \infty} f(t) = \lim_{s \to 0} sF(s) \tag{31}$$

The pointing requirements for this satellite require a steady state pointing accuracy of 0.1° about each axis. By applying the final value theorem to Equation (29) and assuming

that the external disturbance torques can be approximated as a step input, position feedback gains can be determined from the following equations

$$k_x = \frac{T_x - 4\Omega^2 (I_y - I_x)\phi_{ss} + \Omega h_y \phi_{ss}}{\phi_{ss}}$$

$$k_y = \frac{T_y - 3\Omega^2 (I_x - I_z)\theta_{ss}}{\theta_{ss}} \tag{32}$$

$$k_z = \frac{T_z - \Omega^2 (-I_x + I_y)\psi_{ss} + \Omega h_y \psi_{ss}}{\psi_{ss}}$$

The 'ss' subscript denotes steady state and the design torques represent a worst case scenario. It can be seen from Equation (32) that the position feedback gains are not constant; they will vary as a function of orbital position. The natural frequency for roll, pitch and yaw can now be determined by taking the square root of the last term in the denominator in Equation (29). Once this is found, the velocity feedback gains can be calculated from the following expressions

$$k_{vx} = 2\omega_{nx} I_x$$

$$k_{vy} = 2\omega_{ny} I_y \tag{33}$$

$$k_{vz} = 2\omega_{nz} I_z$$

In a similar manner to the position feedback gains, the velocity feedback gains also vary with time. Figure 8 and Figure 9 each depict the time varying nature of the PD gains over one orbit, specifically during perigee. As expected the pitch and pitch rate gains are much higher than the roll and yaw gains.

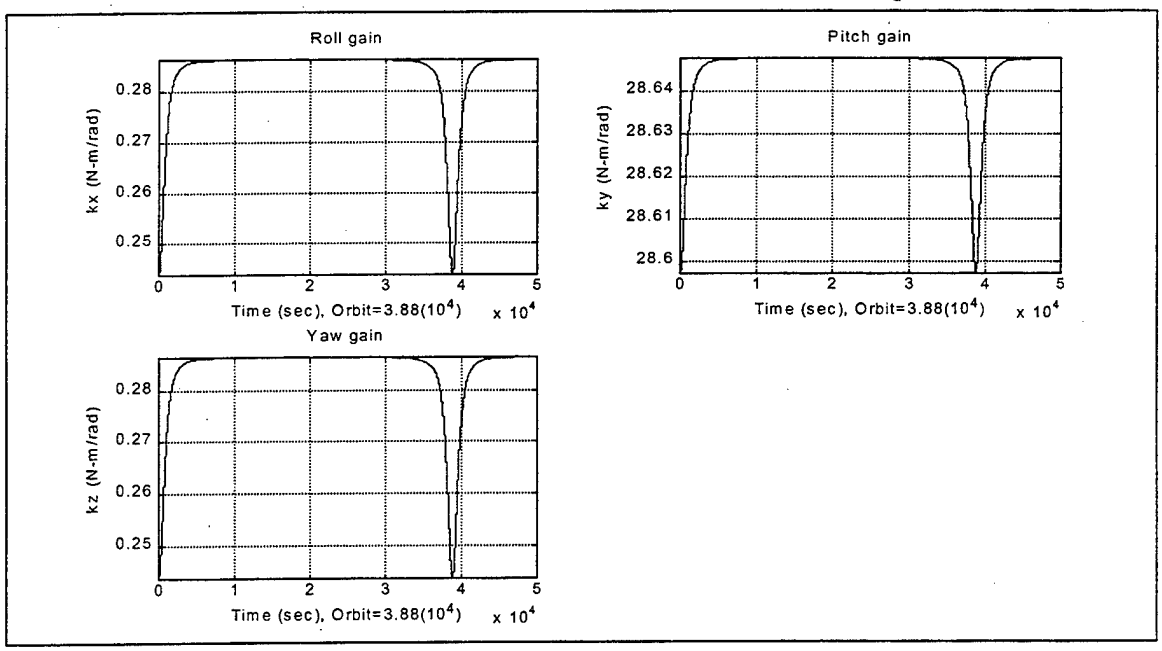

Figure 8: Controller Gains

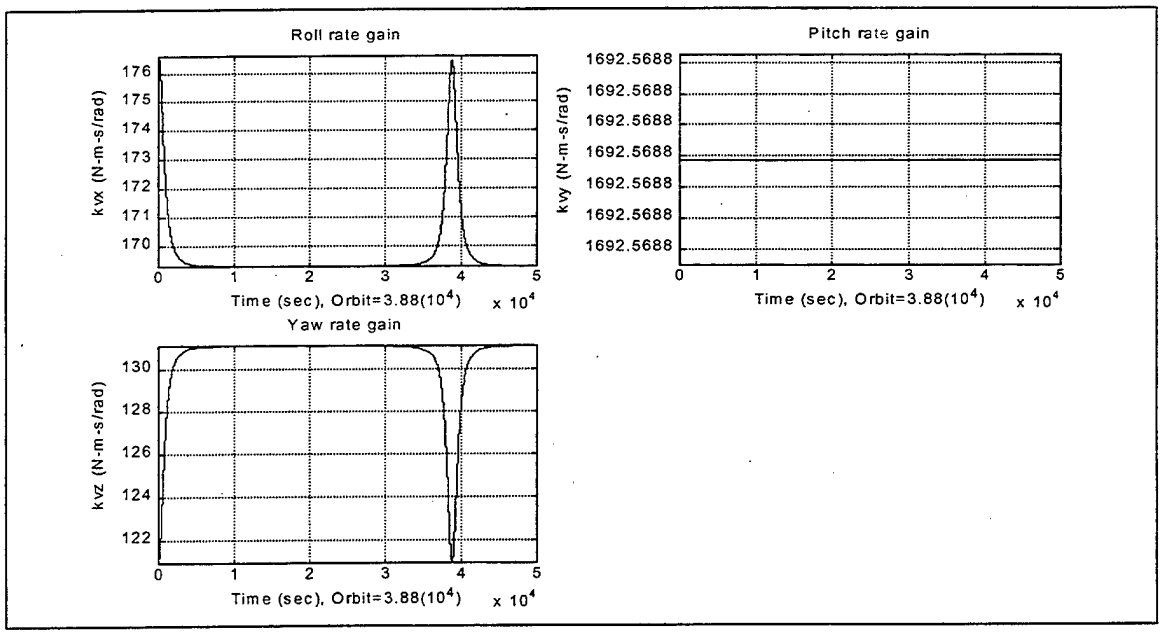

Figure 9: Controller Rate Gains

In the above analysis, many assumptions were made in order to achieve suitable gains, but as long as these gains minimize overshoot and decrease settling time, then they are acceptable. For clarification, the design disturbance torques used in Equation (32)

represent the aggregate, worst-case expected torques, both internal and external. If the magnitude of this torque is exceeded, then the pointing accuracy of this model will suffer. For example, the worst-case torque about the y-axis is estimated to be higher than the torques about the other two axes since the reaction wheel will have to exert a considerable torque at perigee, in order to keep the spacecraft nadir pointing.

VIII. RESULTS

The results of the simulation prove that it is possible to design a satellite control system without rate gyroscopes. Figure 10 is a plot, using only star trackers, of satellite attitude over a period of one orbit. Although it is difficult to see from this figure, both actual and estimated satellite attitude about all three axes have been plotted.

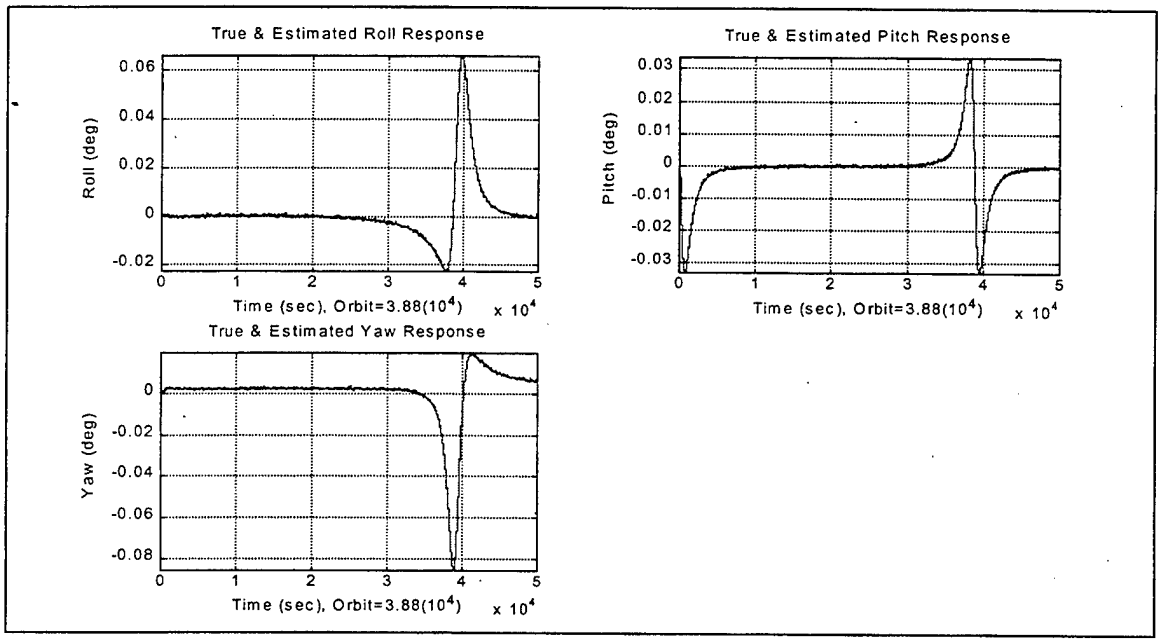

Figure 10: Satellite Attitude

In order to analyze Figure 10, a close-in look at a small portion of each curve is shown in Figure 11, Figure 12, and Figure 13. If the star sensors were ideal, the measurements (triangles) would all fall on the actual attitude, but since there is noise, they are randomly dispersed. The non-measurements in Figure 11 occur when the x-axis star sensor is selected; roll angles can not be sensed, as a result, they are assigned a value of zero. The Kalman filter makes a prediction in this case. It can be seen from the figures below that the Kalman filter cuts through the noise and tracks the attitude effectively. As alluded to earlier, if Q is decreased, the estimates will be weighted in favor of the predicted estimate and if Q is increased the estimates will be weighted in favor of the measurements.

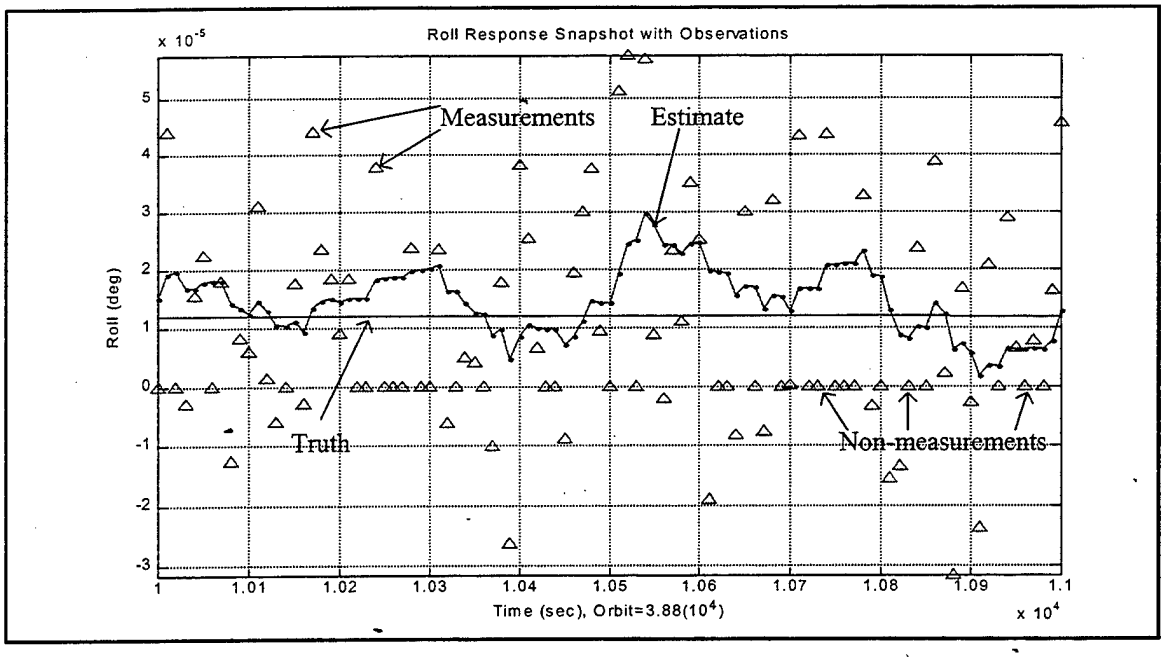

Figure 11: Roll Response with Measurements

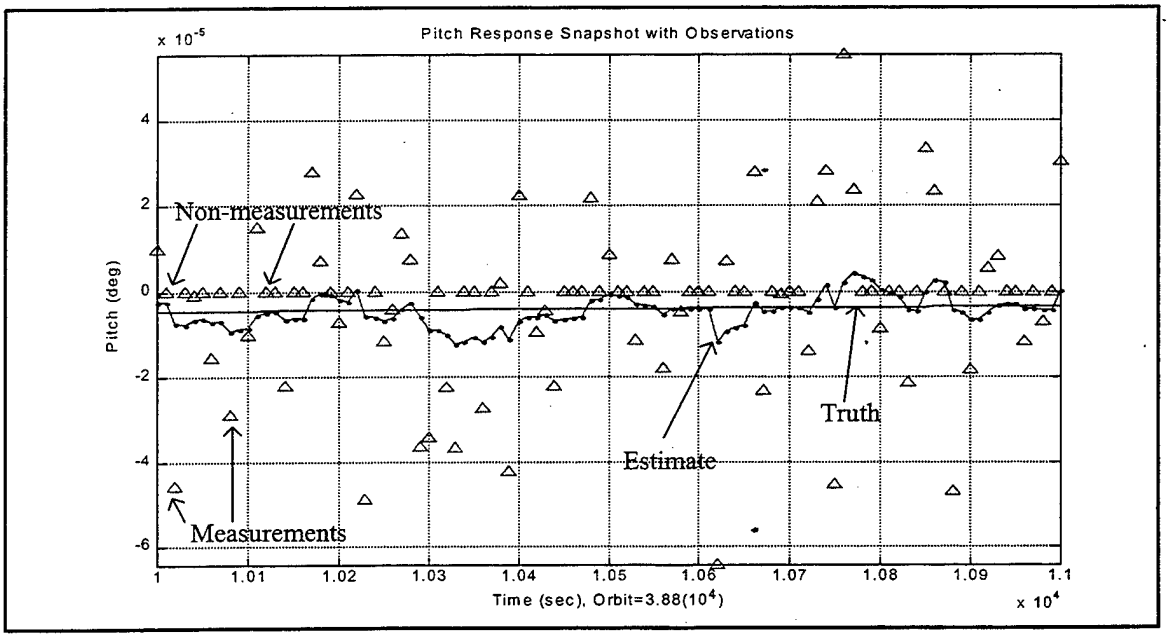

Figure 12: Pitch Response with Measurements

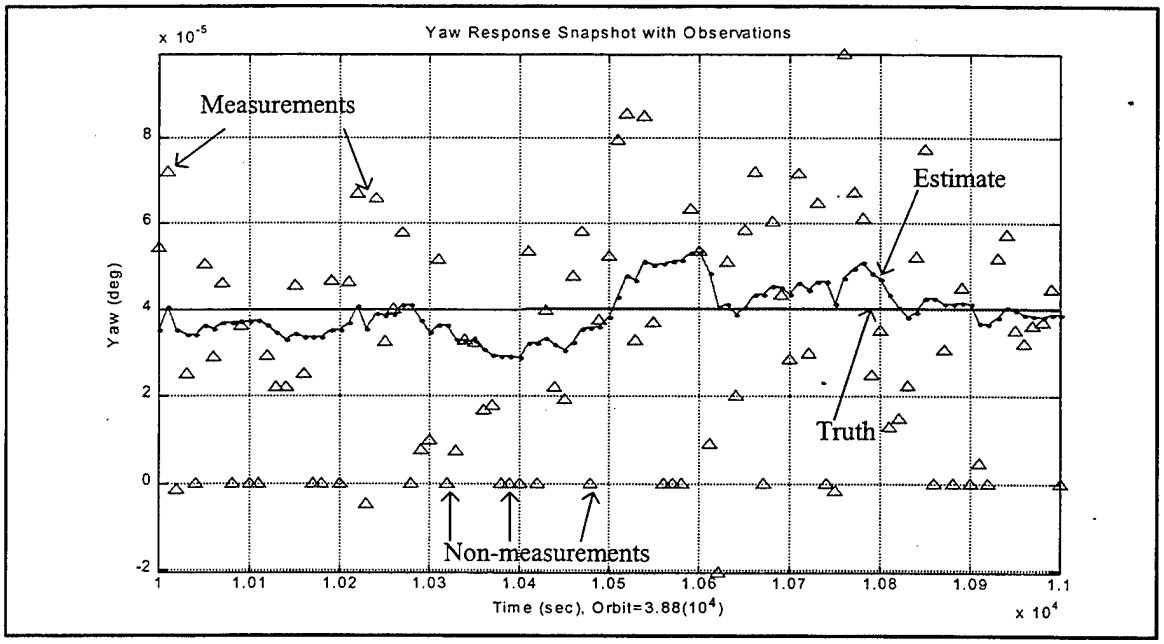

Figure 13: Yaw Response with Measurements

Attitude rates are estimated from star tracker data. These rates traditionally come from rate gyroscopes, but as seen in Figure 14, the estimated rates are very accurate.

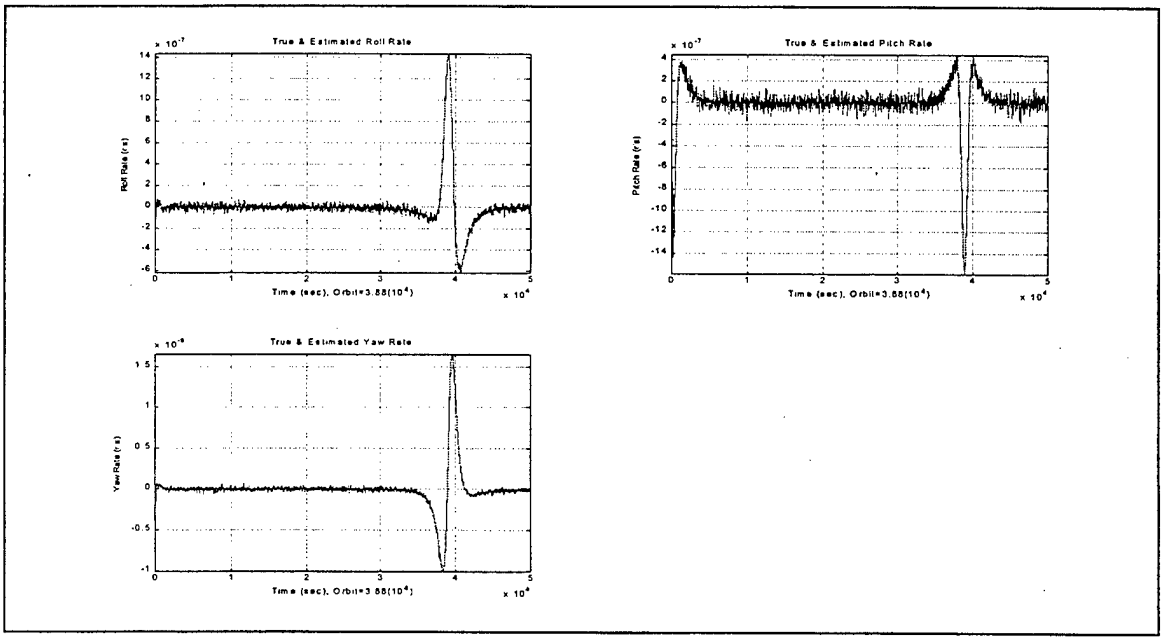

Figure 14: Attitude Rates

Similar to Figure 10, two curves are actually plotted in Figure 14. Figure 15 is a close-in look that shows that the estimate is accurate to within 10^{-7} rad/s. No measurements are shown since star sensors can only measure angles, not rates.

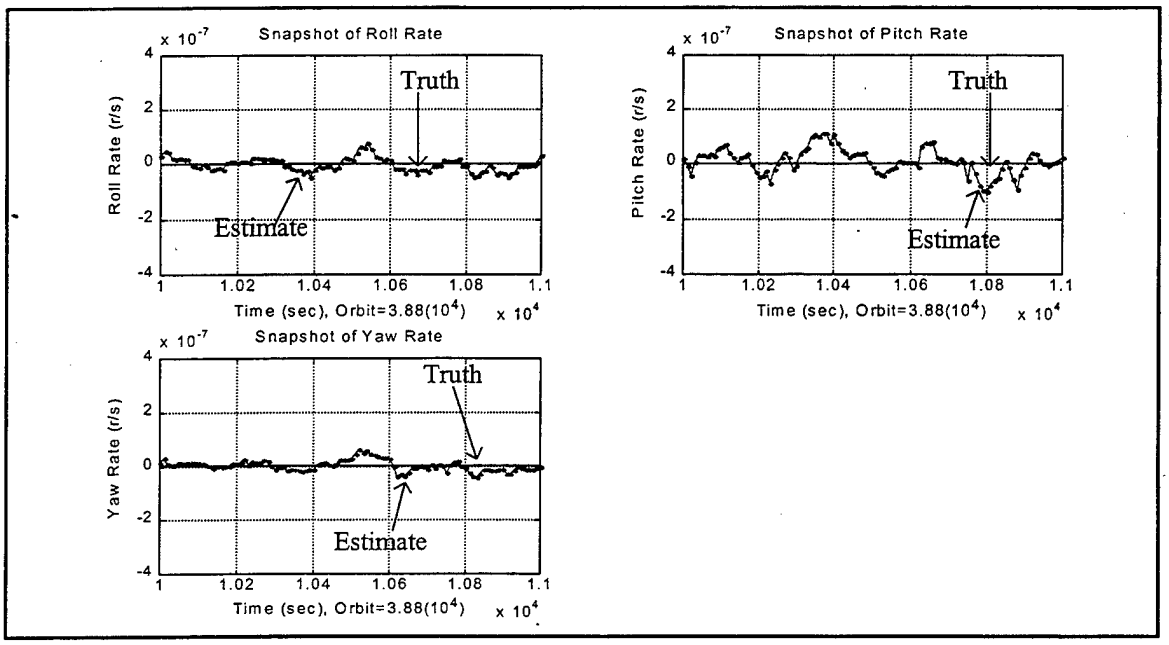

Figure 15: Actual and Estimate Attitude Rates

The output of the Kalman filter is fed into the PD controllers. The torque applied to the reaction wheels over a period of one orbit is shown in Figure 16. As expected, considerable torque is applied to the pitch wheel at perigee. The angular momentum of the pitch wheel at perigee is therefore also high; this can be seen in Figure 17.

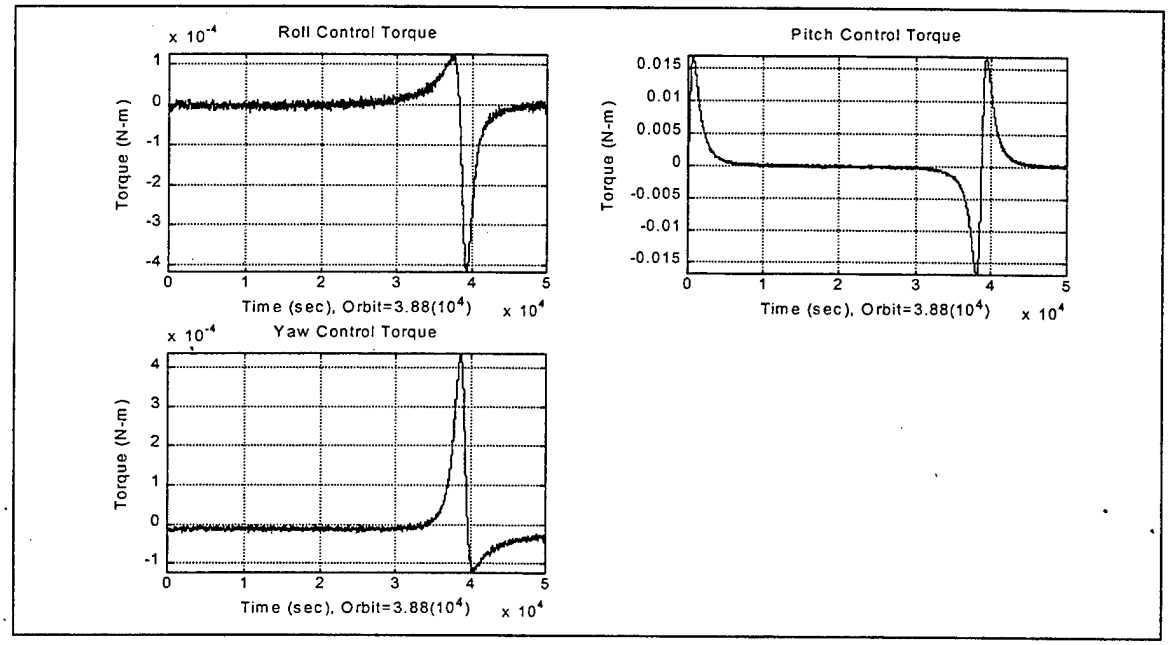

Figure 16: Reaction Wheel Torques

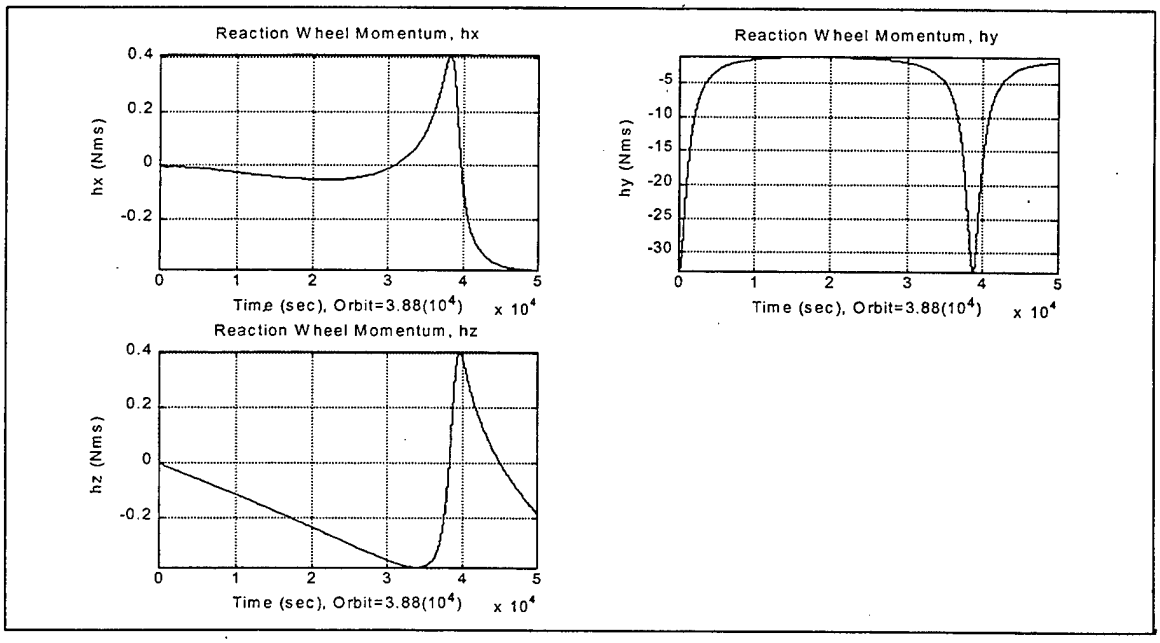

Figure 17: Reaction Wheel Momentum

If there is an attitude error about any axis, internal torques will arise due to the cross coupling of reaction wheel angular momentum components. That is why, in Figure 16,

the magnitude of the torque spikes in both roll and yaw at perigee. The reaction wheels work against each other until the satellite achieves the proper attitude.

IX. SUMMARY AND CONCLUSION

A. SUMMARY

In summary, the equations of motion that were derived in Appendix A were transformed into a state space equation. This state space equation was discretized resulting in a difference equation. This difference equation, the plant model, consisted of a state transition matrix and a deterministic control matrix. Star trackers provided two-axis measurements to the Kalman filter at a rate of 0.1 Hz. The output of the filter was fed into three PD controllers, which counteracted disturbance torques.

B. CONCLUSION

From the previous chapter, it was proven that a discrete Kalman filter is effective in the estimation of body rates from noisy sensor data. The results show that rates can be estimated to within 10^{-7} r/s, which is as good as any gyroscope. These results, however, are based on small angle approximations. The next step is to develop a quaternion for the large angle acquisition phase of the spacecraft. In order to handle the non-linear nature of the quaternion, an extended Kalman filter will have to be implemented.

In this simulation, everything was calculated in ten second intervals. An additional loop needs to be included in the control flow diagram that will speed up the rate updates. A star does not have to be identified in order to calculate rates, it only needs to be identified for position updates. Furthermore, processing delays were not included in this analysis. The satellite was modeled as a rigid body. Further studies will need to study the dynamics of the solar arrays, antennas and other appendages.

The technology for rate estimation, using only star trackers for attitude updates, has reached the level where it is more feasible than using both gyroscopes and star trackers for attitude determination.

APPENDIX A: KINEMATICS

Three reference frames will be used in the derivation of equations of motion 1) Inertial 2) Orbital and 3) Body. Transformation between coordinate systems will be done using direction cosine matrices (DCM). These matrices are given by

$$C(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix}$$

$$C(\theta) = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \tag{34}$$

$$C(\psi) = \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The orbital reference frame is oriented such that the x-axis points in the direction of the velocity vector, the z-axis points towards the center of the Earth and the y-axis completes the right-hand set. It is desired to keep the body frame aligned with the orbital frame. The transformation from the orbital reference frame to the body frame is given by the following 3-2-1 transformation

$$^{b}C^{o} = C(\psi)C(\theta)C(\phi) = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\phi s\psi + s\phi s\theta c\psi & c\phi c\psi + s\phi s\theta s\psi & s\phi c\theta \\ s\phi s\psi + c\phi s\theta c\psi & -s\phi c\psi + c\phi s\theta s\psi & c\theta c\phi \end{bmatrix} \tag{35}$$

The orbital reference frame rotates at a rate of $\Omega(t)$ with respect to the inertial frame, or

$$^{i}\vec{\omega}^{o} = -\Omega\hat{o}_2 \tag{36}$$

In order to perform angular momentum calculations, it is required to express the inertial angular velocity in body coordinates. The inertial angular velocity is represented by

$$ {}^{i}\vec{\omega}^{b} = {}^{i}\vec{\omega}^{o} + {}^{o}\vec{\omega}^{b} \tag{37} $$

The angular velocity of the orbital frame with respect to the inertial frame, expressed in body coordinates is

$$ {}^{i}\vec{\omega}_{b}^{o} = -{}^{b}C^{o}\Omega\hat{o}_{2} \tag{38} $$

The angular velocity of the body frame with respect to the orbital frame, expressed in body coordinates is

$$ {}^{o}\vec{\omega}_{b}^{b} = \dot{\phi}\hat{b}_{1} + C(\phi)C(\theta)\dot{\theta}\hat{n}_{2} + {}^{b}C^{o}\dot{\psi}\hat{o}_{3} \tag{39} $$

The $\hat{n}_2$ unit vector belongs to an intermediate reference frame. If Equation (38) and Equation (39) are substituted into Equation (37), the following result is obtained

$$ {}^{i}\vec{\omega}^{b} = (\dot{\phi} - \Omega\psi)\hat{b}_{1} + (\dot{\theta} - \Omega)\hat{b}_{2} + (\dot{\psi} + \Omega\phi)\hat{b}_{3} \tag{40} $$

Equation (40) is a simplified expression where small angle approximations were used and second order terms were neglected.

APPENDIX B: GRAVITY GRADIENT TORQUES

In a Molnyia orbit, gravity gradient moments will be greatest at perigee. The gravity gradient torque is given by [Ref. 1]

$$\vec{T}_{gg} = \int \vec{r} x \vec{a}_g dm \tag{41}$$

The gravitational acceleration is

$$\vec{a}_g = -GM_{\oplus} \frac{\vec{R} + \vec{r}}{\left|\vec{R} + \vec{r}\right|^3} \tag{42}$$

$\vec{R}$ is the distance to the center of mass of the satellite measured from the center of the Earth and it is given by

$$\vec{R}_o = R\hat{o}_3 \tag{43}$$

Equation (43), expressed in body coordinates is

$$\vec{R}_b = {}^bC^o \vec{R}_o \tag{44}$$

For now, $\vec{R}_b$ will be written as

$$\vec{R}_b = X\hat{b}_1 + Y\hat{b}_2 + Z\hat{b}_3 \tag{45}$$

Taking the cross product

$$\vec{r} x \vec{R} = (yZ - zY)\hat{b}_1 + (zX - xZ)\hat{b}_2 + (xY - yX)\hat{b}_3 \tag{46}$$

From the binomial theorem, the following expression is obtained

$$\left|\vec{R}+\vec{r}\right|^{-3} = \frac{1}{R^3} - 3\frac{xX + yY + zZ}{R^5} \tag{47}$$

It can be shown that the orbital angular velocity is just

$$\Omega = \sqrt{\frac{GM_{\oplus}}{R^3}} \tag{48}$$

Equation (46), Equation (47), and Equation (48) can be substituted into Equation (41) to get the following expression

$$\begin{aligned} T_{ggx} &= -3\Omega^2[\phi(I_y - I_z) - \theta I_{xy} - I_{yz}] \\ T_{ggy} &= -3\Omega^2[\theta(I_x - I_z) + \phi I_{xy} + I_{xz}] \\ T_{ggz} &= 3\Omega^2(\phi I_{xz} + \theta I_{yz}) \end{aligned} \tag{49}$$

These three equations were derived using small angle approximations and neglecting second order terms.

APPENDIX C: DERIVATION OF EQUATIONS OF MOTION

When determining the attitude of a satellite, it is helpful to translate everything into the body coordinate system since on-board sensors will detect errors with respect to the body frame. Transformations are done by a variety of techniques, but the most elementary method is known as the direction cosine matrix. From Appendix A, it was shown that

$$ {}^{i}\vec{\omega}^{b} = (\dot{\phi} - \Omega\psi)\hat{b}_1 + (\dot{\theta} - \Omega)\hat{b}_2 + (\dot{\psi} - \Omega\phi)\hat{b}_3 \tag{50} $$

This result was obtained using small angle approximations where the symbol ϕ represents the error in roll, θ represents the error in pitch, and ψ represents the error in yaw. The total spacecraft angular momentum can be separated into two vectors: 1) angular momentum of the spacecraft body and 2) angular momentum of the reaction wheels, and it is given by the following expression

$$ \vec{H} = \vec{H}_b + \vec{H}_w \tag{51} $$

Assuming cross products of inertia are negligible, the following expression is obtained

$$ \vec{H}_b = I\,{}^{i}\vec{\omega}^{b} \tag{52} $$

It is important to note that when calculating the angular momentum of the satellite about its center of mass, inertial angular rates must be used rather than body rates. Substituting Equation (50) and Equation (52) into Equation (51), total spacecraft angular momentum is found to be

$$ \vec{H} = (I_x\dot{\phi} - I_x\Omega\psi + h_x)\hat{b}_1 + (I_y\dot{\theta} - I_y\Omega + h_y)\hat{b}_2 + (I_z\dot{\psi} + I_z\Omega\phi + h_z)\hat{b}_3 \tag{53} $$

The following relation will be used to determine the Euler moment equations [Ref. 8]

$$\frac{d}{dt}^{i}\vec{H} = \frac{d}{dt}^{b}\vec{H} + {}^{i}\vec{\omega}^{b} x \vec{H} \tag{54}$$

If second order terms are neglected and gravity gradient moments derived in Appendix B are incorporated, it can be shown that the Euler equations for this spacecraft are just

$$T_x = I_x\ddot{\phi} + 4\Omega^2(I_y - I_z)\phi - \Omega h_y\phi - \Omega h_z + \Omega(-I_x + I_y - I_z)\dot{\psi} - h_y\dot{\psi} + h_z\dot{\theta} - I_x\dot{\Omega}\psi + \dot{h}_x$$

$$T_y = I_y\ddot{\theta} + 3\Omega^2(I_x - I_z)\theta + h_x\dot{\psi} + \Omega h_z\psi + \Omega h_x\phi - h_z\dot{\phi} - I_y\dot{\Omega} + \dot{h}_y \tag{55}$$

$$T_z = I_z\ddot{\psi} + \Omega^2(-I_x + I_y)\psi - \Omega h_y\psi + \Omega h_x + \Omega(I_x - I_y + I_z)\dot{\phi} - h_x\dot{\theta} + h_y\dot{\phi} + I_z\dot{\Omega}\phi + \dot{h}_z$$

These equations completely describe the motion of the spacecraft when subject to external disturbance torques. The rate of change of angular momentum of each reaction wheel will be used to counteract the disturbance moments, thereby maintaining the required pointing accuracy. Solving for the Euler angles, however, is no trivial task; all three differential equations are second order and coupled together. If the cross products of inertia are not negligible, the equations of motion become

$$T_{x1} = T_x + (\ddot{\theta} - \dot{\Omega} - 3\Omega^2\theta)I_{xy} + (\Omega^2\psi + \ddot{\psi} + \dot{\Omega}\phi)I_{xz} + (-2\Omega\dot{\theta} - 2\Omega^2)I_{yz}$$

$$T_{y1} = T_y + (-2\Omega\dot{\psi} + 2\Omega^2\phi + \ddot{\phi} - \dot{\Omega}\psi)I_{xy} + 3\Omega^2 I_{xz} + (2\dot{\phi}\Omega - \Omega^2\psi + \ddot{\psi} + \dot{\Omega}\phi)I_{yz} \tag{56}$$

$$T_{z1} = T_z + (2\Omega\dot{\theta} - \Omega^2)I_{xy} + (-2\Omega^2\phi + \ddot{\phi} - \dot{\Omega}\psi)I_{xz} + (\ddot{\theta} - \dot{\Omega} - 3\Omega^2\theta)I_{yz}$$

APPENDIX D: MATLAB CODE

```
%%%%%%%%%%%%%%%%%%MAIN PROGRAM %%%%%%%%%%%%%%%%%%
%
% This code simulates a 3-axis stabilized spacecraft in a Molnyia orbit. The
% satellite is designed to be nadir pointing to within 0.1 degrees about each
% axis. The simulation starts with the satellite at perigee and progresses in
% discrete, ten second time intervals for an entire orbit. Three orthogonal
% star sensors, each aligned with the body axes, sense attitude errors caused
% by various disturbance torques. Due to limited onboard processing
% capabilities, only one star sensor can make an observation at each time step;
% this star sensor is selected at random. These measurements, however, are
% corrupted by additive white noise. A six-state discrete Kalman filter is
% used to both diminish sensor noise effects and estimate rates. The data from
% the Kalman filter is then fed back to three independent, proportional plus
% derivative controllers thus completing the control loop. This code calls three
% functions: ssorbit, ssgains, and ssmatrix.
%
%
%
Ix=25000;
Iy=25000;                                   % Moments of inertia
Iz=15000;
%
mu=398601;                                  % Gravitational constant
rp=7378.15;                                 % Radius of perigee
ra=42164.17;                                % Radius of apogee
a=(rp+ra)/2;                                % Semi-major axis
e=(ra-rp)/(ra+rp);                          % Eccentricity
p=a*(1-e^2);                                % Semi-latus rectum
%
Tspx=1e-5;
```

```
Tspy=1e-5;                                  % Solar pressure moments
Tspz=1e-5;
%
r0=rp;                                      % Radius at t=0 (perigee)
v0=sqrt(2*mu/r0-mu/a);                      % Velocity at t=0
omega0=v0/r0;                               % Orbital angular rate at t=0
hy0=-omega0*Iy;                             % RW angular momentum at t=0
%
t0=0;
dt=10;
tf=50000;
tspan=[t0:dt:tf];                           % Time span
yo=[r0 0 0 omega0];                         % Initial conditions
options = odeset('RelTol',1e-6);            % Accuracy of convergence
[t,y]=ode45('ssorbit',tspan,yo,options,e,mu,p);  % Integration
%
s=size(t);
kmax=s(1,1)+1;
%
x=zeros(6,kmax);                            % Plant array
xkk=zeros(6,kmax);                          % Estimation array
xkkm1=zeros(6,kmax);                        % Prediction array
z=zeros(2,kmax);                            % Measurement array
zx=zeros(1,kmax);
zy=zeros(1,kmax);
zz=zeros(1,kmax);
h=zeros(3,kmax);                            % RW angular momentum array
Tc=zeros(3,kmax);                           % RW torque array
time=zeros(1,kmax);                         % Sampling time array
P=1e-12*eye(6);                             % Initial error covariance
h(:,1)=[0 hy0 0]';                          % Initial RW angular momentum
```

```
%
%
%
for i=1:kmax-1
%
%
%
  [F,k,Wo]=ssgains(y(i,:)',h(:,i),Ix,Iy,Iz);              % PD controller gains
  [A,B,Wdot]=ssmatrix(y(i,:)',h(:,i),mu,p,e,Ix,Iy,Iz);% Plant and control matrices
  Aaug=A-B*F;                                             % Augmented plant matrix
  [phik,delk]=c2d(Aaug,B,dt);                             % Discrete A and B matrices
  ud=[(Tspx+Wo*h(3,i))/Ix Tspy/Iy+Wdot ...
      (Tspz-Wo*h(1,i))/Iz]';                              % Disturbance torques
  W=1e-14*diag([.01 1 .01]);                              % Power spectral density
  Aq=[-Aaug B*W*B';zeros(6) Aaug']*dt;
  Bq=expm(Aq);
  phiq=Bq(7:12,7:12)';
  Q=phiq*Bq(1:6,7:12);                                    % Plant noise covariance
  c=rand;                                                 % Random number generator
%
  if c<=.3333                                             % 1 STAR SENSOR/TIME STEP
    N=2e-5;                                               % Roll sensor noise
    R=N^2*eye(2);                                         % Roll sensor covariance
    H=[0 0 1 0 0 0;0 0 0 0 1 0];                          % Pitch & yaw observations
    z(:,i)=H*x(:,i)+N*randn*ones(2,1);                    % Measurement with sensor noise
    zy(i)=z(1,i);                                         % Observation in pitch
    zz(i)=z(2,i);                                         % Observation in yaw
  elseif c>.3333 & c<=.6666
    N=2e-5;                                               % Pitch sensor noise
    R=N^2*eye(2);                                         % Pitch sensor covariance
    H=[1 0 0 0 0 0;0 0 0 0 1 0];                          % Roll & yaw observations
```

```
    z(:,i)=H*x(:,i)+N*randn*ones(2,1);          % Measurement with sensor noise
    zx(i)=z(1,i);                               % Observation in roll
    zz(i)=z(2,i);                               % Observaton in yaw
  else
    N=2e-5;                                     % Yaw sensor noise
    R=N^2*eye(2);                               % Yaw sensor covariance
    H=[1 0 0 0 0 0;0 0 1 0 0 0];                % Roll & pitch observations
    z(:,i)=H*x(:,i)+N*randn*ones(2,1);          % Measurement with sensor noise
    zx(i)=z(1,i);                               % Observation in roll
    zy(i)=z(2,i);                               % Observation in pitch
  end
%                                               % 3 STAR SENSOR/TIMESTEP
%     Nx=2e-5;
%     Ny=2e-5;                                  % Nominal star sensor noise
%     Nz=2e-5;
%     Rx=4e-10*eye(2);
%     Ry=4e-10*eye(2);                          % Sensor noise covariance
%     Rz=4e-10*eye(2);
%     Hx=[0 0 1 0 0 0;0 0 0 0 1 0];             % x-axis star sensor
%     Hy=[1 0 0 0 0 0;0 0 0 0 1 0];             % y-axis star sensor
%     Hz=[1 0 0 0 0 0;0 0 1 0 0 0];             % z-axis star sensor
%     x(:,i+1)=phik*x(:,i)+delk*ud;             % Plant
%
%     Gx=P*Hx'*inv(Hx*P*Hx'+Rx);                % Initial Kalman gain
%     zx(:,i)=Hx*x(:,i)+Nx*randn*ones(2,1);     % Noisy y and z measurements
%     xkk1=xkkm1(:,i)+Gx*(zx(:,i)-Hx*xkkm1(:,i)); % Initial estimate
%     P1=(eye(6)-Gx*Hx)*P;                      % Initial error covariance
%
%     Gy=P1*Hy'*inv(Hy*P1*Hy'+Ry);              % Updated Kalman gain
%     zy(:,i)=Hy*x(:,i)+Ny*randn*ones(2,1);     % Noisy x and z measurements
%     xkk2=xkk1+Gy*(zy(:,i)-Hy*xkk1);           % Updated estimate
```

```
%     P2=(eye(6)-Gy*Hy)*P1;                          % Updated error covariance
%
%     Gz=P2*Hz'*inv(Hz*P2*Hz'+Rz);                   % Final Kalman gain
%     zz(:,i)=Hz*x(:,i)+Nz*randn*ones(2,1);          % Noisy x and y measurements
%     xkk(:,i)=xkk2+Gz*(zz(:,i)-Hz*xkk2);            % Final estimate
%    Pk=(eye(6)-Gz*Hz)*P2;                           % Final error covariance
%
  x(:,i+1)=phik*x(:,i)+delk*ud;                      % Plant
  G=P*H'*inv(H*P*H'+R);                              % Kalman gain
  xkk(:,i)=xkkm1(:,i)+G*(z(:,i)-H*xkkm1(:,i));       % Current estimate
%
  if zx(i)==0                                        % No roll measurement case
    xkk(1:2,i)=xkkm1(1:2,i);                         % Roll & roll rate prediction
  elseif zy(i)==0                                    % No pitch measurement case
    xkk(3:4,i)=xkkm1(3:4,i);                         % Pitch & pitch rate prediction
  elseif zz(i)==0                                    % No yaw measurement case
    xkk(5:6,i)=xkkm1(5:6,i);                         % Yaw & yaw rate prediction
  end
%
  Pk=(eye(6)-G*H)*P;                                 % Current error covariance
  xkkm1(:,i+1)=phik*xkk(:,i)+delk*ud;                % Future estimate
  P=phik*Pk*phik'+Q;                                 % Future error covariance
  Tc(:,i)=-k*xkk(:,i);                               % Control torques (using xkk)
  h(:,i+1)=h(:,i)+Tc(:,i)*dt;                        % RW angular momentum
  aeig(:,i)=eig(A);                                  % Eigenvalues, no controller
  augeig(:,i)=eig(Aaug);                             % Eigenvalues, PD controller
%
  kx(i)=k(1,1);                                      % Roll gain
  kvx(i)=k(1,2);                                     % Roll rate gain
  ky(i)=k(2,3);                                      % Pitch gain
  kvy(i)=k(2,4);                                     % Pitch rate gain
```

```
  kz(i)=k(3,5);                          % Yaw gain
  kvz(i)=k(3,6);                         % Yaw rate gain
  q11(i)=Q(1,1);
  q33(i)=Q(3,3);                         % Elements of Q
  q55(i)=Q(5,5);
  p11(i)=P(1,1);
  p33(i)=P(3,3);                         % Elements of P
  p55(i)=P(5,5);
  time(i+1)=time(i)+dt;                  % Time steps
%
%
%
end
%
%
%
%
%
%
%%%%%%%%%%%%%%%%%%%% RESULTS %%%%%%%%%%%%%%%%%%%%%%
r=1001:1101;
t1=time(r);
t2=time(1:kmax-1);
%
%
%
figure(1)
subplot(221)
plot(time,x(1,:)*180/pi)
title('Roll Response using Ideal Gyros/Sensors')
xlabel('Time (sec), Orbit=3.88(10^4)')
```

```
ylabel('Roll (deg)')
axis tight
grid;
%
subplot(222)
plot(time,x(3,:)*180/pi)
title('Pitch Response using Ideal Gyros/Sensors')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Pitch (deg)')
axis tight
grid;
%
subplot(223)
plot(time,x(5,:)*180/pi)
title('Yaw Response using Ideal Gyros/Sensors')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Yaw (deg)')
axis tight
grid;
%
%
%
figure(2)
subplot(221)
plot(t,y(:,1))
title('Orbit Radius')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Radius (km)')
grid;
%
subplot(222)
```

```
plot(t,y(:,3)*180/pi)
title('True Anomaly')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('True Anomaly (deg)')
grid;
%
subplot(223)
plot(t,y(:,4))
title('Orbital Angular Velocity')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Wo (rad/s)')
grid;
%
%
%
figure(3)
subplot(221)
plot(time,Tc(1,:))
title('Roll Control Torque')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Torque (N-m)')
axis tight
grid;
%
subplot(222)
plot(time,Tc(2,:))
title('Pitch Control Torque')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Torque (N-m)')
axis tight
grid;
```

```
%
subplot(223)
plot(time,Tc(3,:))
title('Yaw Control Torque')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Torque (N-m)')
axis tight
grid;
%
%
%
figure(4)
subplot(221)
plot(time,h(1,:))
title('Reaction Wheel Momentum, hx')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('hx (Nms)')
axis tight
grid;
%
subplot(222)
plot(time,h(2,:))
title('Reaction Wheel Momentum, hy')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('hy (Nms)')
axis tight
grid;
%
subplot(223)
plot(time,h(3,:))
title('Reaction Wheel Momentum, hz')
```

```
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('hz (Nms)')
axis tight
grid;
%
%
%
figure(5)
subplot(221)
plot(t2,kx)
title('Roll gain')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('kx (N-m/rad)')
axis tight
grid;
%
subplot(222)
plot(t2,ky)
title('Pitch gain')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('ky (N-m/rad)')
axis tight
grid;
%
subplot(223)
plot(t2,kz)
title('Yaw gain')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('kz (N-m/rad)')
axis tight
grid;
```

```
%
%
%
figure(6)
subplot(221)
plot(t2,kvx)
title('Roll rate gain')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('kvx (N-m-s/rad)')
axis tight
grid;
%
subplot(222)
plot(t2,kvy)
title('Pitch rate gain')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('kvy (N-m-s/rad)')
axis tight
grid;
%
subplot(223)
plot(t2,kvz)
title('Yaw rate gain')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('kvz (N-m-s/rad)')
axis tight
grid;
%
%
%
figure(7)
```

```
subplot(221)
plot(time,x(1,:)*180/pi,'-',time,xkk(1,:)*180/pi)
title('True & Estimated Roll Response')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Roll (deg)')
axis tight
grid;
%
subplot(222)
plot(time,x(3,:)*180/pi,'-',time,xkk(3,:)*180/pi)
title('True & Estimated Pitch Response')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Pitch (deg)')
axis tight
grid;
%
subplot(223)
plot(time,x(5,:)*180/pi,'-',time,xkk(5,:)*180/pi)
title('True & Estimated Yaw Response')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Yaw (deg)')
axis tight
grid;
%
%
%
figure(8)
subplot(221)
plot(time,x(2,:),'-',time,xkk(2,:))
title('True & Estimated Roll Rate')
xlabel('Time (sec), Orbit=3.88(10^4)')
```

```
ylabel('Roll Rate (r/s)')
axis tight
grid;
%
subplot(222)
plot(time,x(4,:),'-',time,xkk(4,:))
title('True & Estimated Pitch Rate')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Pitch Rate (r/s)')
axis tight
grid;
%
subplot(223)
plot(time,x(6,:),'-',time,xkk(6,:))
title('True & Estimated Yaw Rate')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Yaw Rate (r/s)')
axis tight
grid;
%
%
%
figure(9)
subplot(221)
plot(t1,x(1,r),t1,xkk(1,r),'.-')
title('Snapshot of Roll Response')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Roll (deg)')
axis([min(t1) max(t1) -3e-5 6e-5]);
grid;
%
```

```
subplot(222)
plot(t1,x(3,r),t1,xkk(3,r),'.-')
title('Snapshot of Pitch Response')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Pitch (deg)')
axis([min(t1) max(t1) -3e-5 3e-5]);
grid;
%
subplot(223)
plot(t1,x(5,r),t1,xkk(5,r),'.-')
title('Snapshot of Yaw Response')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Yaw (deg)')
axis([min(t1) max(t1) 1e-5 8e-5]);
grid;
%
%
%
figure(10)
subplot(221)
plot(t1,x(2,r),t1,xkk(2,r),'.-')
title('Snapshot of Roll Rate')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Roll Rate (r/s)')
axis([min(t1) max(t1) -4e-7 4e-7]);
grid;
%
subplot(222)
plot(t1,x(4,r),t1,xkk(4,r),'.-')
title('Snapshot of Pitch Rate')
xlabel('Time (sec), Orbit=3.88(10^4)')
```

```
ylabel('Pitch Rate (r/s)')
axis([min(t1) max(t1) -4e-7 4e-7]);
grid;
%
subplot(223)
plot(t1,x(6,r),t1,xkk(6,r),'.-')
title('Snapshot of Yaw Rate')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Yaw Rate (r/s)')
axis([min(t1) max(t1) -4e-7 4e-7]);
grid;
%
%
%
figure(11)
subplot(221)
plot(t2,q11)
title('q11')
xlabel('Time (sec), Orbit=3.88(10^4)')
axis tight
grid;
%
subplot(222)
plot(t2,q33)
title('q33')
xlabel('Time (sec), Orbit=3.88(10^4)')
axis([min(t2) max(t2) 2.0240e-12 2.0242e-12]);
grid;
%
subplot(223)
plot(t2,q55)
```

```
title('q55')
xlabel('Time (sec), Orbit=3.88(10^4)')
axis tight
grid;
%
%
%
figure(12)
subplot(221)
plot(t2,p11)
title('p11')
xlabel('Time (sec), Orbit=3.88(10^4)')
axis tight
grid;
%
subplot(222)
plot(t2,p33)
title('p33')
xlabel('Time (sec), Orbit=3.88(10^4)')
axis tight
grid;
%
subplot(223)
plot(t2,p55)
title('p55')
xlabel('Time (sec), Orbit=3.88(10^4)')
axis tight
grid;
%
%
%
```

```
figure(13)
plot(t1,x(1,r),t1,xkk(1,r),'.-',t1,zx(r),'^')
title('Roll Response Snapshot with Observations')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Roll (deg)')
axis tight
grid
%
%
%
figure(14)
plot(t1,x(3,r),t1,xkk(3,r),'.-',t1,zy(r),'^')
title('Pitch Response Snapshot with Observations')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Pitch (deg)')
axis tight
grid
%
%
%
figure(15)
plot(t1,x(5,r),t1,xkk(5,r),'.-',t1,zz(r),'^')
title('Yaw Response Snapshot with Observations')
xlabel('Time (sec), Orbit=3.88(10^4)')
ylabel('Yaw (deg)')
axis tight
grid
%
%
%
figure(16)
```

```
subplot(211)
plot(aeig,'.')
title('Root Locus, No Controller')
xlabel('Real')
ylabel('Imaginary')
grid
%
subplot(212)
plot(augeig,'.')
title('Root Locus, PD Controller')
xlabel('Real')
ylabel('Imaginary')
axis([-.035 .035 -3e-3 3e-3])
grid
%%%%%%%%%%%%%%%%%%%%% END MAIN  %%%%%%%%%%%%%%%%%%%%
%
%
%
function [F,k,Wo]=ssgains(y,h,Ix,Iy,Iz)
% This function computes non-optimal gains for each PD controller. Both the
% position and rate gains will be time varying. In order to keep the satellite
% nadir pointing at perigee, it is expected that the pitch gains will be much
% higher than the roll and yaw gains. The F matrix is used, in part, to augment
% the A matrix. The resulting augmented plant matrix has all six eigenvalues in
% the left hand plane, which is a requirement for system stability. The k matrix
% is used to calculate the reaction wheel control torques.
%
%
%
Wo=y(4,:);                          % Orbital angular velocity
hy=h(2,:);                          % Angular momentum of y RW
```

```
%
Tx=5e-4;
Ty=5e-2;                                    % Design torques
Tz=5e-4;
%
ssphi=.1*pi/180;
sstheta=.1*pi/180;                          % Allowable Steady State Errors
sspsi=.1*pi/180;
%
kx=(Tx-4*Wo^2*(Ix-Iy)*ssphi+Wo*hy*ssphi)/ssphi;
wnx=sqrt(4*Wo^2*(Iy-Iz)/Ix+kx/Ix);          % Roll gains & natural freq.
kvx=2*wnx*Ix;
%
ky=(Ty-3*Wo^2*(Ix-Iz)*sstheta)/sstheta;
wny=sqrt(3*Wo^2*(Ix-Iz)/Iy+ky/Iy);          % Pitch gains & natural freq.
kvy=2*wny*Iy;
%
kz=(Tz-Wo^2*(-Ix+Iy)*sspsi+Wo*hy*sspsi)/sspsi;
wnz=sqrt(Wo^2*(-Ix+Iy)/Iz+kz/Iz);           % Yaw gains & natural freq.
kvz=2*wnz*Iz;
%
%
%
F=[kx/Ix kvx/Ix 0 0 0 0;...
   0 0 ky/Iy kvy/Iy 0 0;...
   0 0 0 0 kz/Iz kvz/Iz];                   % Controller gain matrix
%
k=[kx kvx 0 0 0 0;...
   0 0 ky kvy 0 0;...
   0 0 0 0 kz kvz];                         % Gain matrix
%%%%%%%%%%%%%% END FUNCTION SSGAIN %%%%%%%%%%%%%%%%%%
```

```
%
%
%
function ydot=ssorbit(t,y,FLAG,e,mu,p)
% This function solves two first order differential equations for radius
% and true anomaly using a Runge-Kutta integration scheme.  In order to
% have orbital angular velocity and acceleration available for each star
% sensor measurement, the differential equations were solved at fixed,
% discrete time step. The duration of the time steps is ten seconds,
% same as the star sensor sampling time.
%
%
%
r=y(1,:);                                   % Orbital radius
rdot=y(2,:);                                % Rate of change of radius
nu=y(3,:);                                  % True anomaly
Wo=y(4,:);                                  % Orbital angular velocity
%
% Output
ydot(1,:)=rdot;
ydot(2,:)=sqrt(mu/p)*e*cos(nu)*Wo;
ydot(3,:)=Wo;
ydot(4,:)=-sqrt(mu/p)/r^2*(r*e*sin(nu)*Wo...
  +(1+e*cos(nu))*sqrt(mu/p)*e*sin(nu));
%%%%%%%%%%%%%%%%% END FUNCTION SSORBIT %%%%%%%%%%%%%%
%
%
%
function [A,B,Wdot]=ssmatrix(y,h,mu,p,e,Ix,Iy,Iz)
% This function computes the system matrices for the state space equations.
% Both the A and B matrices are time varying since they both include
```

```
% orbital angular velocity and orbital angular acceleration terms.  It is
% assumed that these values can be pre-calculated and stored in the satellite
% computer.  According to engineers at the Aerospace corporation, this is not
% an unreasonable assumption.  Each time a star sensor observation is made,
% associated with that measurement is a dedicated orbital angular velocity
% and acceleration.
%
%
%
r=y(1,:);                                   % Orbital radius
nu=y(3,:);                                  % True anomaly
Wo=y(4,:);                                  % Orbital angular velocity
%
hx=h(1,:);
hy=h(2,:);                                  % RW angular momentum
hz=h(3,:);
%
Wdot=-sqrt(mu/p)/r^2*(r*e*sin(nu)*Wo...
  +(1+e*cos(nu))*sqrt(mu/p)*e*sin(nu));     % Orbital angular acceleration
%
A=[0 1 0 0 0 0;...
    (-4*Wo^2*(Iy-Iz)+Wo*hy)/Ix 0 0 ...
    -hz/Ix Wdot (-Wo*(-Ix+Iy-Iz)+hy)/Ix;...
  0 0 0 1 0 0;...
  -Wo*hx/Iy hz/Iy -3*Wo^2*(Ix-Iz)/Iy 0 ...
  -Wo*hz/Iy -hx/Iy;...
  0 0 0 0 0 1;...
  -Wdot (-Wo*(Ix-Iy+Iz)-hy)/Iz 0 hx/Iz ...
  (-Wo^2*(-Ix+Iy)+Wo*hy)/Iz 0];             % Plant matrix
B=[0 0 0;1 0 0;0 0 0;0 1 0;0 0 0;0 0 1];     % Control Matrix
%%%%%%%%%%%%%% END FUNCTION SSMATRIX %%%%%%%%%%%%%%
```

LIST OF REFERENCES

1. Weisel, William E., *Spaceflight Dynamics 2nd ed.*, McGraw-Hill Companies, Inc., New York, NY, 1997.
2. Gai, E., Daly, K., Harrison, J., Lemos, L.K., "Star-Sensor-Based Satellite Attitude/Attitude Rate Estimator", *Journal of Guidance and Control*, Vol. 8, No. 5, Sept.-Oct. 1985, pp. 560-565.
3. Sidi, Marcel J., *Spacecraft Dynamics and Control*, Cambridge University Press, New York, NY, 1997.
4. Podgorski, W.A., Lemos, L.K., Cheng, J., and Daly, K.C., "Gyroless Attitude Determination and Control System for Advanced Environment Satellites", AIAA Paper 82-1614, Aug. 1982.
5. Brown, Robert G., *Introduction to Random Signals and Applied Kalman Filtering 3rd ed.*, John Wiley & Sons, Inc., New York, NY, 1997.
6. Gelb, Arthur, *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.
7. Ogata, Katsuhiko, *Modern Control Engineering 3rd ed.*, Prentice-Hall, Inc., Upper Saddle River, NJ, 1997.
8. Agrawal, Brij N., *Design of Geosynchronous Spacecraft*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1986.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd. STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Department Chairman, Code AA 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, CA 93943-5000

4. Department of Electrical and Computer Engineering 3
   ATTN: Professor Hal A. Titus, Code EC/Ts
   Naval Postgraduate School
   Monterey, CA 93943-5121

5. Department of Aeronautics and Astronautics 1
   ATTN: Professor Brij Agrawal, Code AA/Ag
   Naval Postgraduate School
   Monterey, CA 93943-5000

6. Department of Aeronautics and Astronautics 1
   ATTN: Professor Barry Leonard, Code AA/Ln
   Naval Postgraduate School
   Monterey, CA 93943-5000

7. LT Jay D. Vogt, USN 2
   c/o Vogt Family
   831 Marble Dr.
   Ft. Collins, CO 80526