



**OPTIMAL SPACECRAFT ATTITUDE CONTROL
USING AERODYNAMIC TORQUES**

THESIS

Michael Luke Gargas, Captain, USAF

AFIT/GA/ENY/07-M08

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

AFIT/GA/ENY/07-M08

**OPTIMAL SPACECRAFT ATTITUDE CONTROL
USING AERODYNAMICS TORQUES**

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Astronautical Engineering

Michael Luke Gargas, BSE

Captain, USAF

March 2007

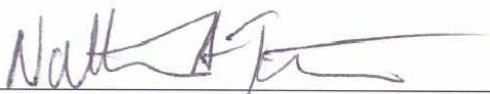
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

**OPTIMAL SPACECRAFT ATTITUDE CONTROL
USING AERODYNAMICS TORQUES**

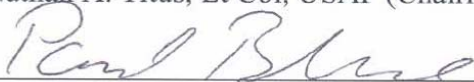
Michael Luke Gargas, BSE

Captain, USAF

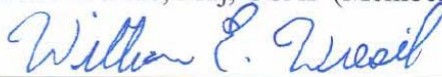
Approved:



Nathan A. Titus, Lt Col, USAF (Chairman)



Paul A. Blue, Maj, USAF (Member)



Dr. William E. Wiesel (Member)

3-7-07

Date

7 May 07

Date

3-7-07

Date

Abstract

This thesis introduces a method of three-axis spacecraft attitude control using only aerodynamic torques. Attitude actuation is achieved using four control panels mounted on the rear of a cubical spacecraft bus. The controller consists of an outer loop using linear state feedback to determine desired control torque and an inner loop to choose appropriate control panel angles. The inner loop uses a Jacobian-based approach to invert the nonlinear relationship between panel angles and generated torque. Controller performance is evaluated via simulations, which show that three-axis control is possible over a range of initial angles and angular rates. The analysis used partial accommodation theory as the basis for aerodynamic torque calculations and assumed a rotating atmosphere with an exponential density profile.

Acknowledgments

I'd like to thank my lovely wife for all her love and support and my boys for making me smile every day. It's been a true pleasure getting to go home from school every day and spend time with you guys.

Thanks also to LtCol Titus for helping me with this thesis. I certainly could not have gotten this to work completely without your help along the way.

Capt M. Luke Gargas

Table of Contents

	Page
Abstract.....	iv
Acknowledgments.....	v
Table of Contents.....	vi
List of Figures.....	viii
I. Introduction.....	1
Background.....	1
Problem Statement.....	1
Research Objectives.....	2
Significance of Research	2
Thesis Overview	2
II. Literature Review.....	4
Chapter Overview.....	4
Early Analyses.....	5
Meirovitch and Wallace Analysis.....	5
Frik Analysis.....	5
Ravindran and Hughes Analysis.....	6
Recent Analyses.....	8
Kumar, Mazanek, and Heck Analysis.....	8
Psiaki Analysis.....	10
III. Methodology.....	13
Chapter Overview.....	13
Spacecraft Geometry & Coordinate Frames.....	13

	Page
Equations of Motion	16
Translation Equations of Motion	16
Rotational Kinematics Equations of Motion.....	17
Rotational Dynamics Equations of Motion.....	19
Aerodynamic Torques	19
Aerodynamic Torque Theory.....	20
Atmospheric Model	25
Aerodynamic Torque Application	27
Control Law	31
Outer Loop Control Law.....	31
Inner Loop Control Law	33
MATLAB Implementation	35
IV. Results.....	37
Chapter Overview	37
Simulation Results	37
Case 1	38
Case 2.....	42
Case 3.....	44
Case 4.....	45
Case 5.....	47
Case 6.....	49
V. Conclusions and Recommendations	52
Conclusions.....	52
Recommendations for Future Research.....	52
Appendix A – MATLAB Simulation Code	56
Appendix B – MATLAB Jacobian Calculation Code	78
Bibliography	80

List of Figures

Figure	Page
Figure 1: Ravindran and Hughes ‘Arrow’ Spacecraft Configuration	6
Figure 2: Kumar, Mazanek, & Heck ‘Wind Vane’ Spacecraft Configuration	9
Figure 3: Psiaki ‘Shuttlecock’ Spacecraft Configuration	11
Figure 4: Spacecraft Geometry	14
Figure 5: Coordinate Systems	15
Figure 6: Specular and Diffuse Molecular Reflection	20
Figure 7: Molecules Incident on an Element of Spacecraft Surface	21
Figure 8: Density Values	26
Figure 9: Analysis Numbering Scheme	27
Figure 10: Spacecraft Dimensions	28
Figure 11: Spacecraft Euler Angles – Case 1 (Uncontrolled)	40
Figure 12: Spacecraft Roll Axis Angular Rates – Case 1 (Uncontrolled)	40
Figure 13: Spacecraft Pitch Axis Angular Rates – Case 1 (Uncontrolled)	41
Figure 14: Spacecraft Yaw Axis Angular Rates – Case 1 (Uncontrolled)	41
Figure 15: Euler Angles – Case 2 (Controlled)	43
Figure 16: Spacecraft Angular Rates – Case 2 (Controlled)	43
Figure 17: Control Panel Angles without Saturation Avoidance – Case 2 (Controlled) ..	44
Figure 18: Control Panel Angles with Saturation Avoidance – Case 3 (Controlled)	45
Figure 19: Euler Angles – Case 4 (Controlled)	46
Figure 20: Control Panel Angles – Case 4 (Controlled)	47
Figure 21: Euler Angles – Case 5 (Controlled)	48
Figure 22: Control Panel Angles – Case 5 (Controlled)	48
Figure 23: Euler Angles – Case 6 (Controlled)	50
Figure 24: Angular Rates – Case 6 (Controlled)	50
Figure 25: Control Panel Angles – Case 6 (Controlled)	51

OPTIMAL SPACECRAFT ATTITUDE CONTROL USING AERODYNAMICS TORQUES

I. Introduction

Background

Within the United States Department of Defense, there is an increased emphasis on the concept of Operationally Responsive Space (Senate Armed Services Committee, 2005:17). It has been proposed that one means to be more responsive is to build smaller, simpler satellites that can, in theory, be launched quickly when a requirement arises (Janicik, 2003:1). Many satellites with significant operational and scientific capability under 100 kg (microsats) have been launched to date, and even smaller satellites (nanosats) are expected to follow. Although significant progress has been made in reducing the size and weight of traditional actuators like reaction wheels and magnetic torquers, alternative methods of attitude control need to be explored. One such alternative is to use the torques resulting from aerodynamic forces acting upon spacecraft in low Earth orbits as a means of attitude control. Aerodynamic torques have been used to generate passive attitude stability in the past and have been proposed as a part of an active control system in combination with magnetic torque rods (Psiaki, 2004:347). This thesis introduces a particular spacecraft geometry which uses aerodynamic torques with an active control system for three-axis attitude control.

Problem Statement

This thesis looks to expand upon previous efforts using aerodynamic torques for attitude control by developing a particular spacecraft geometry and using only aerodynamic torques for three-axis attitude control. Attitude control consists not only of

attitude stabilization to an equilibrium position but also to precision pointing of the spacecraft to a desired offset final orientation.

Research Objectives

The objectives of this research are to develop a spacecraft geometry consisting of a spacecraft bus and control panels, apply partial accommodation theory to calculate aerodynamic torques acting upon this spacecraft in various low Earth orbits, develop a controller to stabilize this spacecraft from a variety of initial orientations and point this spacecraft to a desired final orientation, and demonstrate (via simulation) three-axis attitude control using only aerodynamic torques.

Significance of Research

With the current move toward microsats/nanosats, non-traditional methods of attitude control need to be explored for possible implementation on these spacecraft. This thesis explores using naturally occurring aerodynamic torques for attitude control of spacecraft in low Earth orbits. If proved viable, this method of attitude control could lead to significant weight savings on spacecraft that could be dedicated to data gathering or communications rather than spacecraft attitude control.

Thesis Overview

The following chapters of this thesis are Literature Review, Methodology, Results, and Conclusions and Recommendations. Literature Review covers previously published works discussing the use of aerodynamic torques for attitude control. Methodology covers all aspects of the analysis performed in this thesis. It shows the spacecraft geometry and coordinate frames used, the equations of motion used to model

the motion of the orbiting spacecraft, the model used to derive the aerodynamic torques acting upon the spacecraft, the atmospheric model used, the control law developed to drive the spacecraft to its desired final orientation, and the computer program used for simulation. Results covers the six cases performed to demonstrate the concept of using aerodynamic torques for three-axis spacecraft attitude control. Conclusions and Recommendations covers what can be taken from this research and what can be done to improve upon it in the future.

II. Literature Review

Chapter Overview

This chapter outlines the review of published literature discussing the use of aerodynamic torques as a method of attitude control. This review revealed two distinct time periods when this topic was discussed. The first period was roughly 1966 to 1972 and the second period was roughly 1995 to 2004.

Early analyses focused largely on simple representations of the spacecrafts' orbits and developed stability analyses to verify the concept of utilizing aerodynamic torques as an attitude control method. Apparently, these concepts did not catch on since there was a more than 20 year hiatus in the discussion. Attitude control methods during those 20 years consisted mostly of internal devices such as reaction wheels and magnetic torquers and external devices such as thrusters. Aerodynamic torques were viewed as something that had to be dealt with rather than something that could be taken advantage of.

More recent analyses are much more complicated than those discussed above. They include much more realism in the analyses and develop designs to use aerodynamic torques for attitude control. In fact, one experiment was flown in 1996 which explored using passive aerodynamic torques to control attitude. The following discussion highlights the key points found in the literature. They are divided into Early Analyses and Recent Analyses as discussed above.

Early Analyses

Meirovitch and Wallace Analysis

A 1966 analysis performed by L. Meirovitch and F. B. Wallace, Jr. looked at “the stability of motion of spinning, symmetrical spacecraft under the influence of aerodynamic and gravity gradient torques.” (Meirovitch et al., 1966:2202). This analysis developed the differential equations of motion and performed a stability analysis using Lyapunov’s direct method. A number of simplifying assumptions were made during the development of this analysis: a circular orbit, a uniform gravitational field, no coupling between translational and rotational motion, and a uniform atmospheric density for the entire orbit (1966:2196). This analysis concluded that equilibrium conditions did in fact exist for this particular set-up and developed stability criteria for motion near these equilibrium positions.

Frik Analysis

A 1970 analysis by Martin A. Frik expanded upon the above analysis by generalizing to a rigid body with no restrictions on mass distribution and shape (Frik, 1970:1780). He made the same simplifying assumptions as the Meirovitch and Wallace analysis and also specifically stated another not mentioned above: no rotation of the atmosphere due to the Earth’s rotation (1970:1780-1). Frik concluded that:

“In case of conservative aerodynamic torque, at least one stable equilibrium orientation exists...The shape of the satellite and the location of the center of mass determines whether the aerodynamic torque is conservative. For all bodies of revolution with the center of mass located on the axis of symmetry, the aerodynamic torque is conservative.

Stable equilibrium orientations may also exist if the aerodynamic torque is conservative only in the vicinity of the equilibrium.

Nonconservative aerodynamic torques will in general cause a destabilization of all equilibrium orientations.” (1970:1785).

It was clear from these early analyses that as simplifying assumptions were removed to create more realistic problems, more perturbations were introduced and the possibility of passive aerodynamic stabilization was less likely. Therefore, the expansion of the analyses to include an active controller was an obvious next step. Ravindran and Hughes presented their concept for an active controller in 1972 in the *Journal of Spacecraft and Rockets* in an article entitled “Optimal Aerodynamic Attitude Stabilization of Near-Earth Satellites.” (Ravindran et al., 1972:499-506).

Ravindran and Hughes Analysis

Ravindran’s and Hughes’ analysis used Euler’s Equation for rotational motion with two external torques, a disturbance torque on the spacecraft body and a control torque on the spacecraft control panels (1972:500). A picture of their spacecraft is shown in Figure 1 (1972:499):

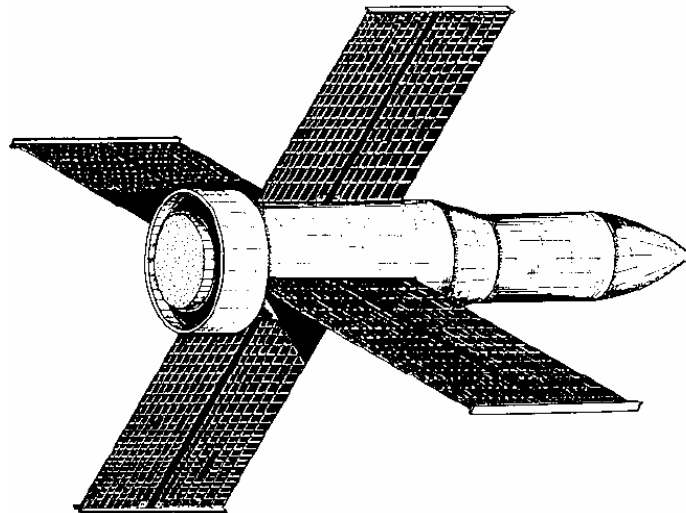


Figure 1: Ravindran and Hughes ‘Arrow’ Spacecraft Configuration

Much like the previously discussed analyses, a number of simplifying assumptions were made: spacecraft in a circular orbit, “time rate of change of the orbital parameters (e.g. due to the Earth’s oblateness and atmospheric effects) are small in comparison with the orbital rate of the satellite”, only periodic diurnal atmospheric density variations are investigated, small deflections of the control panels are not considered, movement of the control panels do not change the spacecraft’s overall moment of inertia, only drag and gravity gradient disturbance torques are considered, a linear analysis is appropriate since external disturbance torques are small, and the Earth’s atmosphere is assumed to rotate with the same angular velocity as the Earth (1972:500).

With these simplifying assumptions in place, Ravindran and Hughes spell out their development of the equations of motion in some detail. Of particular interest to this author is their aerodynamic disturbance torque development. They extend supersonic aerodynamics acting on a flat plat to develop the shear and normal forces acting upon the control panels and then the aerodynamic disturbance torque which they use in their equations of motion. The authors also noted that “the drag acting on the satellite control surfaces, as a direct consequence of using aerodynamic control, leads to a reduction in the life of the satellite.” (1972:501). Therefore, the goal of their analysis is to maintain pointing accuracy while minimizing total drag on the spacecraft.

Ravindran and Hughes linearized their equations of motion and applied feedback control to drive the system to equilibrium. The aim of their analysis was to find a suitable control “which is optimal in some sense and assures asymptotic stability” (1972:502). To define optimal for this analyses, they chose to minimize a quadratic cost function which equally weighted the position away from equilibrium and the control

input. This led to a control law which incorporates the weighting factors, the control input, and the solution of the matrix Riccati equation (1972:502). With the equations of motion derived and the control law set-up, they then propagated their system for a number of different initial conditions. A few of these initial condition variables are 200 km and 300 km orbital altitude, polar and equatorial orbits, 5° and 8° initial offset, and variable atmospheric densities. The trials showed “that the satellite is unstable with [no] control whereas with feedback control all the modes of the satellites rotational motion are damped out in approximately $\frac{1}{2}$ orbit.” (1972:502-4). The authors of this study also looked at the effect of using drag for attitude control on the lifetime of a spacecraft and they concluded that “lifetimes of the order of two to three years are possible.” (1972:506). Apparently, the space community was not too interested in the conclusions of Ravindran and Hughes because very little was written on this topic until the 1990’s.

Recent Analyses

Kumar, Mazanek, and Heck Analysis

In 1995, Renjith Kumar, Daniel Mazanek, and Michael Heck again looked at aerodynamic torques to passively stabilize a spacecraft. Their article entitled “Simulation and Shuttle Hitchhiker Validation of Passive Satellite Aerostabilization” discussed the theoretical basis for the Passive Aerodynamically Stabilized Magnetically Damped Satellite-Satellite Test Unit (PAMS-STU) experiment which was eventually flown from Space Shuttle Endeavour (STS-77) in May 1996. The spacecraft was designed to “characterize and demonstrate passive aerodynamic stabilization and passive magnetic hysteresis damping of attitude rates.” (Kumar et al., 1995:806). Figure 2 shows this

spacecraft's design (this design is comparable to a wind vane in a wind field (Kumar et al., 1996:228)):

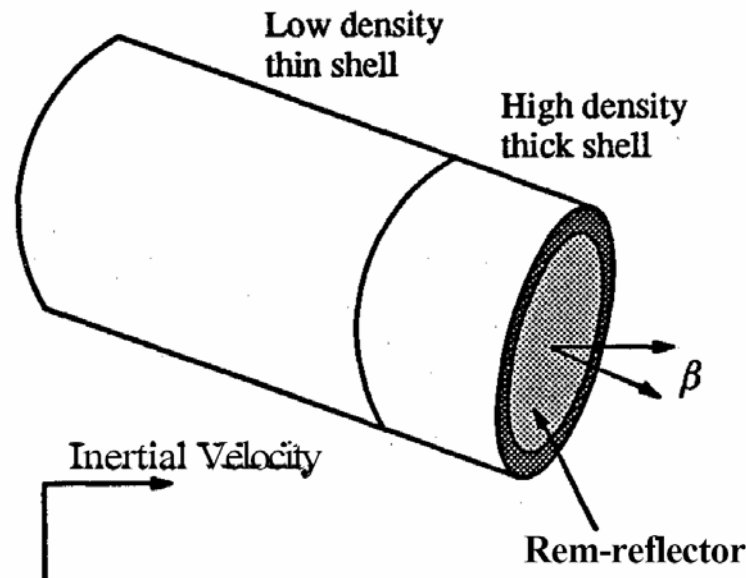


Figure 2: Kumar, Mazanek, & Heck 'Wind Vane' Spacecraft Configuration

This geometry places most of the mass “forward of the geometric center of the satellite, resulting in a center-of-pressure – center-of-mass offset that provides sufficient pitch and yaw aero restoring torques.” (1995:807). Their high-fidelity simulator accounted for the following phenomena:

- “ 1. Free-molecular-flow aerodynamics: accommodation; specular and diffuse reflection of air molecules and shadowing.
2. Jacchia atmospheric model: varying flux and geomagnetic index.
3. Horizontal global winds.
4. Solar radiation pressure: absorption; specular and diffuse reflection of photons with shadowing.
5. Eight-order Earth magnetic field.
6. Magnetic hysteresis rods.
7. Orbital dynamics: altitude decay, regression of ascending node, and changing solar geometry.” (1995:807).

Another important assumption taken from the article is that the spacecraft's orbital dynamics are modeled as a point mass and the orbit is assumed to be circular (1995:809). With these assumptions incorporated into the model, Kumar, Mazanek, and Heck completed numerous simulation runs and showed the concept of aerodynamic stabilization with magnetic damping to be feasible.

On 22 May 1996, PAMS-STU was deployed from Space Shuttle Endeavor tumbling "at a rate of more than 2 degrees/second" (Psiaki, 2004:347). Two proximity rendezvous were performed and video was taken of the spacecraft. After several days of operation, the NASA team concluded that "overall science objectives and demonstrations of aerodynamic satellite stabilization were achieved." (NASA, 1996:12).

Psiaki Analysis

A 2004 analysis done by Mark L. Psiaki built upon the PAMS concept of passive attitude stabilization from aerodynamic torques but included active magnetic damping rather than the passive hysteresis rod damping used on PAMS. His design resembles a badminton shuttlecock and "uses passive aerodynamic drag torques to stabilize pitch and yaw" and active magnetic torque control "stabilizes roll and damps pitch and yaw" (Psiaki, 2004:347). A pictorial of his design is shown in Figure 3 (note the 'feathers' placed to the rear of the spacecraft bus which are the passive drag surfaces) (2004:348). This design was devised to overcome the uncontrolled instability of Ravindran's and Hughes' 'arrow-like' design. Psiaki notes that "this arrow concept has been modified to become a badminton shuttlecock-type design for stabilization of the yaw and pitch axes. A shuttlecock is a better analogy for the aerodynamics of the new system because an arrow relies on lift forces whereas a shuttlecock relies on drag." (2004:347).

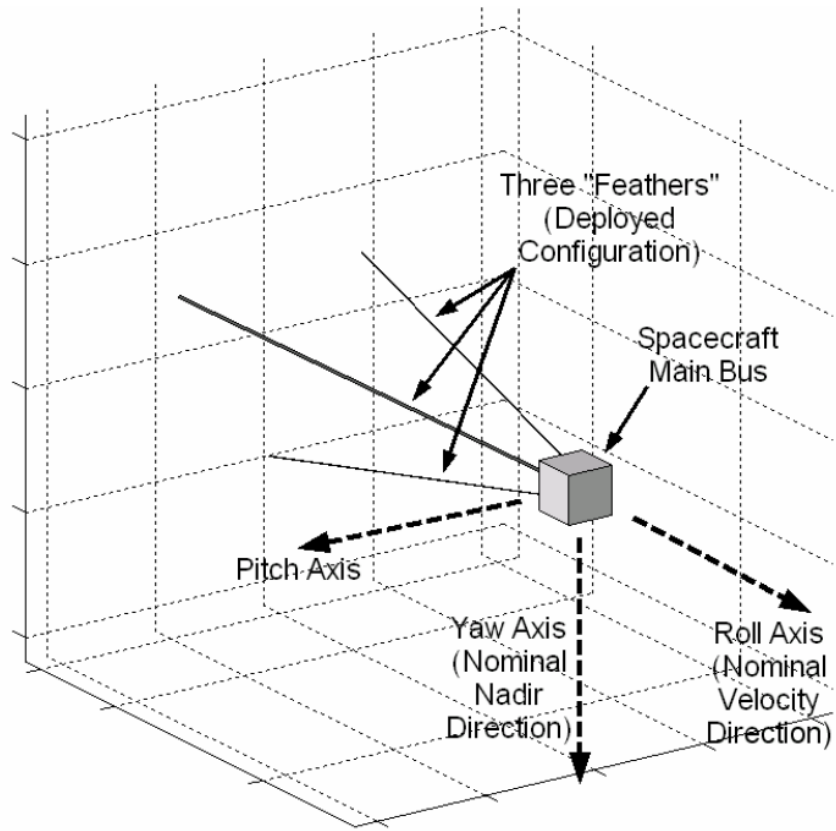


Figure 3: Psiaki 'Shuttlecock' Spacecraft Configuration

This analysis includes a number of assumptions. The orbit propagation uses “Keplerian dynamics and secular J2 effects” but does not include “the effects of drag, solar radiation pressure, periodic J2 terms, higher-order gravity potential terms, sun-moon effects, or tidal effects.” (2004:353). The equations of motion uses Euler’s equation with forcing terms that include drag torques, magnetic coil torques, “the gravity-gradient effect, solar radiation pressure, and radiation pressure from the Earth’s albedo.” (2004:352). The Earth’s atmosphere is assumed to rotate with the Earth with a diurnal density bulge “that elevates the density by a factor of four at the 1400 hours local time at the same latitude as the sun” which drops off “as a two-dimensional Gaussian with a standard deviation of 52 degrees in longitude and 29 degrees in latitude.” (2004:352). Simulations were run for a

‘typical case’ and a ‘challenging case’. The typical case included the center-of-mass of the spacecraft placed perfectly on the roll axis, no collective twist of the spacecraft’s feathers, and an equatorial orbit whereas the challenging case included center-of-mass imbalance, collective twist of the spacecraft’s feathers, and an 87 degree orbital inclination (2004:353). Psiaki summarizes his results as follows”

“The shuttlecock design with feedback can be three-axis stabilized up to 500-km altitude if the controller gains are chosen properly. Atmospheric rotation, magnetometer measurement errors, spacecraft mass imbalance, collective aerodynamic twist of the feathers, solar radiation pressure torque, and parametric resonance from periodic atmospheric density variations can cause steady-state pitch and yaw biases and steady-state oscillations on all three axes. The nominal system considered in this study can be designed to be globally stable below 500 km and to have maximum per-axis pointing errors of 36 degrees or less. These pointing errors decrease with decreases in altitude, the orbital inclination, the mass imbalance, the aerodynamic twist of the feathers, and the magnetometer measurement errors.” (2004:354).

III. Methodology

Chapter Overview

This chapter outlines a method of three-axis attitude control using only aerodynamic torques. The spacecraft designed and analyzed in this thesis is a low Earth orbiting satellite which uses aerodynamic torques to control the spacecraft's orientation relative to the spacecraft's orbital path. Attitude actuation is achieved using four control panels mounted on the rear of a cubical spacecraft bus. An outer/inner loop controller is developed to drive the spacecraft to the desired orientation. The outer loop uses quaternion feedback reorientation to determine the desired control torque, and the inner loop uses a Jacobian-based approach to choose appropriate control panel angles and incorporates saturation avoidance to maximize control authority available for future maneuvers. The analysis uses partial accommodation theory as the basis for aerodynamic torque calculations and models the Earth's atmosphere as a rotating with an exponential density profile.

Spacecraft Geometry & Coordinate Frames

To facilitate a simulation-based investigation of using aerodynamic torques for attitude control, a specific spacecraft geometry was established consisting of a cubical bus and four control panels as show in Figure 4. The control panels were placed to the rear of the center-of-mass, similar to a badminton shuttlecock, to provide passive stability about the pitch and yaw axes. Their default orientation is $\theta_{c_1} = \theta_{c_2} = -45^\circ$ and

$\theta_{c_3} = \theta_{c_4} = 45^\circ$ as shown in Figure 4 and each panel is allowed to rotate $\pm 45^\circ$ from this default orientation to produce the desired control torques.

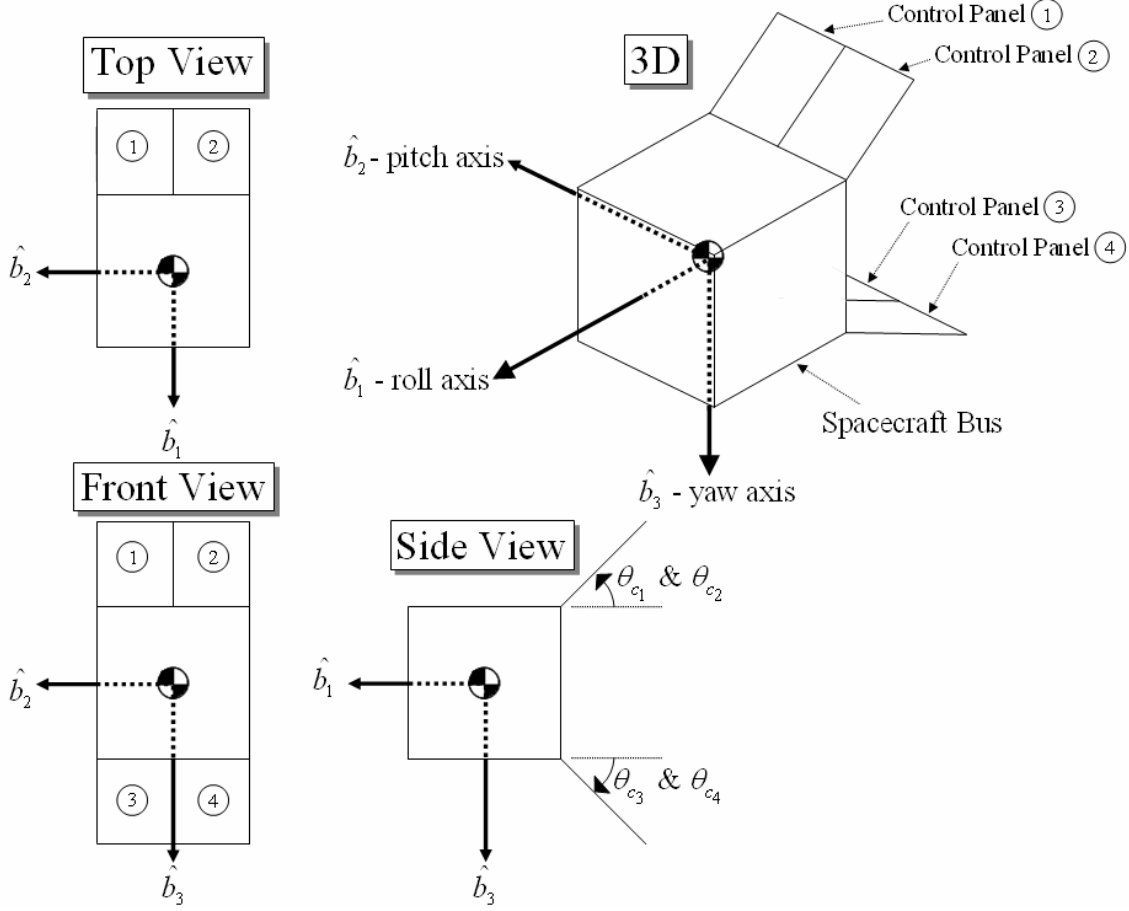


Figure 4: Spacecraft Geometry

An orthogonal principle axis body coordinate system $(\hat{b}_1, \hat{b}_2, \hat{b}_3)$ is used in the analysis, where $\hat{b}_1$ is the roll axis, $\hat{b}_2$ is the pitch axis, and $\hat{b}_3$ is the yaw axis. In a nominal orientation, the $\hat{b}_1$ axis points along the spacecraft's velocity vector ($\hat{o}_1$), and the $\hat{b}_3$ axis points in the nadir direction ($\hat{o}_3$). The orbital coordinate frame $(\hat{o}_1, \hat{o}_2, \hat{o}_3)$ is a rotating frame where the $\hat{o}_1$ vector points along the spacecraft's orbital path, the $\hat{o}_3$ vector points

to the center of the inertial frame, and the $\hat{o}_2$ vector completes the orthogonal coordinate system. It should be noted that this orbital frame is only good for circular orbits (which is assumed in this analysis). The inertial coordinate system selected is an Earth-Center Inertial frame ($\hat{I}$, $\hat{J}$, $\hat{K}$). The $\hat{I}$ vector points to the vernal equinox, the $\hat{K}$ vector points out the Earth's north pole, and the $\hat{J}$ vector completes the orthogonal system. These coordinate frames, along with the body frame introduced above, are shown in Figure 5. Other items noted in Figure 5 are the spacecraft's position vector ($\vec{R}$), the spacecraft velocity vector ($\vec{V}$), the orbital inclination (i), the right ascension of the ascending node (Ω), the argument of perigee (ω), and the initial true anomaly (v_o). (Petty, 2006):

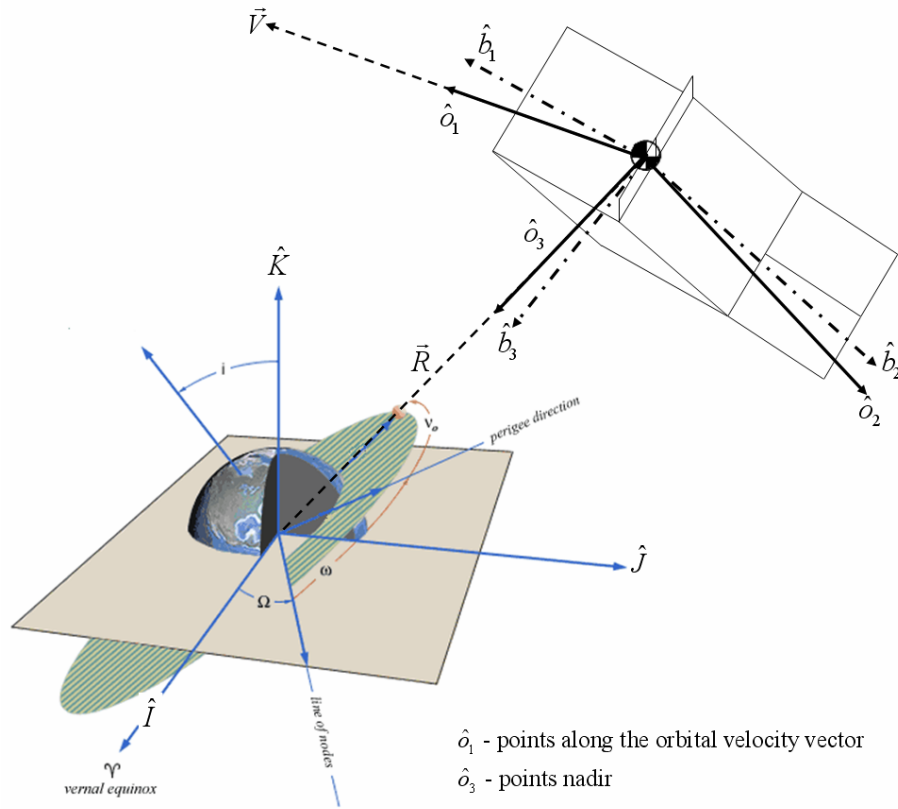


Figure 5: Coordinate Systems

It is often useful to perform calculations in the spacecraft's body frame. Therefore, transformations are necessary to translate measurements in one coordinate frame into another. To transform measurements in the orbital frame to the spacecraft's principle axis body frame an Euler 1-2-3 rotation sequence is used to produce the following rotation matrix (Hughes, 2004:19):

$$R_{ORB \rightarrow PRIN} = \begin{bmatrix} c_2 c_3 & s_1 s_2 c_3 + s_3 c_1 & -c_1 s_2 c_3 + s_3 s_1 \\ -c_2 s_3 & -s_1 s_2 s_3 + c_3 c_1 & c_1 s_2 s_3 + c_3 s_1 \\ s_2 & -s_1 c_2 & c_1 c_2 \end{bmatrix} \quad (3.1)$$

where: $s_i \triangleq \sin \theta_i$ $c_i \triangleq \cos \theta_i$

Due to the orbital motion assumptions made for this analysis, the transformation matrix from the Earth-Center Inertial to the Orbital Frame was not necessary.

Equations of Motion

To completely describe the motion of the spacecraft being analyzed in this thesis, it was necessary to define equations for the translational motion of the spacecraft as well as equations of motion for the rotational kinematics and rotational dynamics of the spacecraft.

Translation Equations of Motion

The orbital dynamics used for this analysis assume strictly 2-body motion (the spacecraft and the Earth) and assume a circular, non-decaying orbit. The timeframe required to stabilize and point the spacecraft is relatively short in terms of a spacecraft's lifetime so the long term effects of drag on the spacecraft's orbit are ignored in this analysis. The spacecraft altitude was chosen for each simulation and the velocity of the spacecraft was then calculated via this relationship (Wiesel, 1997:70):

$$V_{circular} = \sqrt{\frac{\mu}{R}} \quad (3.2)$$

where:

μ : Earth's Gravitational Constant ($3.986 \times 10^{14} \text{ m}^3 / \text{sec}^2$)

R : Orbital Altitude + Earth's Radius ($6.378 \times 10^6 \text{ m}$)

To analyze the motion of a spacecraft orbiting the Earth, a starting position must be specified. For the analysis performed here, all spacecraft begin coincident with the $\hat{I}$ axis (i.e. $\nu_o = 0^\circ$) and true anomaly (ν) propagates as the spacecraft orbits.

To calculate how ν propagates, the mean motion (n) of spacecraft is first calculated (Wiesel, 1997:59):

$$n_{circular} = \sqrt{\frac{\mu}{R^3}} \quad (3.3)$$

Then, ν at any time is calculated:

$$\nu(t) = n_{circular} t + \nu_o \quad (3.4)$$

where: t : time (seconds)

Another useful property of the orbit to calculate is the orbital period (T) (Wiesel, 1997:58):

$$T_{circular} = 2\pi \sqrt{\frac{R^3}{\mu}} \quad (3.5)$$

Rotational Kinematics Equations of Motion

There are three Euler Angles ($\theta_1, \theta_2, \theta_3$) which describe the orientation of the spacecraft relative to the orbital frame. Although Euler Angles provide a more intuitive description of attitude, quaternions were used in this analysis to avoid the possibility of

encountering singularities. To derive the initial quaternions for a simulation, it was necessary to use the rotation matrix from the orbital frame to the principal axis frame ($R_{ORB \rightarrow PRIN}$) derived earlier (3.1). The quaternions are derived from this rotation matrix by the following relationships (the subscripts in the q equation below refer to the row-column position in the $R_{ORB \rightarrow PRIN}$ rotation matrix) (Wie, 1998:319):

$$q_4 = \pm \frac{1}{2} \sqrt{1 + \text{trace}(R_{ORB \rightarrow PRIN})} \quad (3.6)$$

$$\underline{q} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix} = \frac{1}{4q_4} \begin{bmatrix} R_{ORB \rightarrow PRIN_{23}} - R_{ORB \rightarrow PRIN_{32}} \\ R_{ORB \rightarrow PRIN_{31}} - R_{ORB \rightarrow PRIN_{13}} \\ R_{ORB \rightarrow PRIN_{12}} - R_{ORB \rightarrow PRIN_{21}} \end{bmatrix} \quad (3.7)$$

With the initial Euler Angles converted to quaternions, the rotation matrix ($R_{ORB \rightarrow PRIN}$) can be written in terms of quaternions (1998:319):

$$R_{ORB \rightarrow PRIN} = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_2q_1 - q_3q_4) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 + q_1q_4) \\ 2(q_3q_1 + q_2q_4) & 2(q_3q_2 - q_1q_4) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \quad (3.8)$$

The kinematic differential equations for quaternions are (1998:327):

$$\dot{\underline{q}} = \frac{1}{2} (\underline{q}_4 \vec{\omega} - \vec{\omega}^\times \underline{q}) \quad (3.9)$$

$$\dot{q}_4 = -\frac{1}{2} (\vec{\omega}^T \underline{q}) \quad (3.10)$$

where: $\vec{\omega}$: spacecraft's angular velocity vector

Note that $\vec{\omega}^\times$ is a skew symmetric matrix populated with the components of ω :

$$\vec{\omega}^\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (3.11)$$

Rotational Dynamics Equations of Motion

To describe the rotational motion of the spacecraft, Euler's Rotational Equations of Motion were used (1998:341):

$$I\dot{\vec{\omega}} + \vec{\omega} \times I\vec{\omega} = M \quad (3.12)$$

This is the general form of Euler's Equation where I is the moment of inertia of the body being analyzed and M is the external moment acting on the body. The spacecraft moment of inertia is assumed to be that of the spacecraft bus and the control panels are of lightweight construction stiff enough to generate the required aerodynamic torques but not massive enough to effect the overall moment of inertia of the spacecraft. The external moments acting on the spacecraft being analyzed here are the aerodynamic disturbance torque ($\vec{g}_d$) on the spacecraft bus and the aerodynamic control torque ($\vec{g}_c$) being generated by the control surfaces. Making this substitution into Euler's Equation and solving for $\dot{\vec{\omega}}$ gives the dynamics differential equations of motion used in this analysis (Ravindran, 1972:500):

$$\dot{\vec{\omega}} = -I^{-1}\vec{\omega} \times I\vec{\omega} + I^{-1}\vec{g}_d + I^{-1}\vec{g}_c \quad (3.13)$$

These equations are generally written in body frame coordinates (and this analysis uses this convention).

Aerodynamic Torques

Aerodynamic torques occur when the spacecraft impacts gas molecules in the Earth's upper atmosphere and those particles transfer momentum to the spacecraft. The mechanism of that momentum transfer is not perfectly understood, but a number of useful models exist. In this paper, aerodynamic torques are modeled using partial

accommodation theory. The following sections derive the equations used to calculate aerodynamic torques on a spacecraft, describe the atmospheric model used, and then apply those equations to the specific geometry of the spacecraft being analyzed here.

Aerodynamic Torque Theory

The topic of aerodynamic torques is a topic of much uncertainty. There are two extremes when discussing how particles impact a surface and how momentum is transferred from those collisions: specular and diffuse.

“Specular reflection is essentially a deterministic concept: each molecule bounces off the surface with no change in energy. The angle of reflection equals the angle of incidence, and the incoming velocity, the outgoing velocity, and the surface normal are coplanar. The momentum transfer is therefore normal to the surface and equals twice the normal component of the incoming momentum. As it happens, very few molecules experience specular reflection. More often, the incoming molecule becomes at least partially accommodated to the surface. This suggests the other limiting case: in the diffuse reflection model, the incoming molecule is completely accommodated to the surface. It loses all “memory” of its incoming direction and energy; it mingles with other molecules in the layer of surface contamination and eventually leaves with a probabilistic kinetic energy characteristic of the surface temperature and a probabilistic direction governed by a “cosine” distribution.” (Hughes, 2004:250).

These two concepts are depicted in Figure 6 (2004:249):

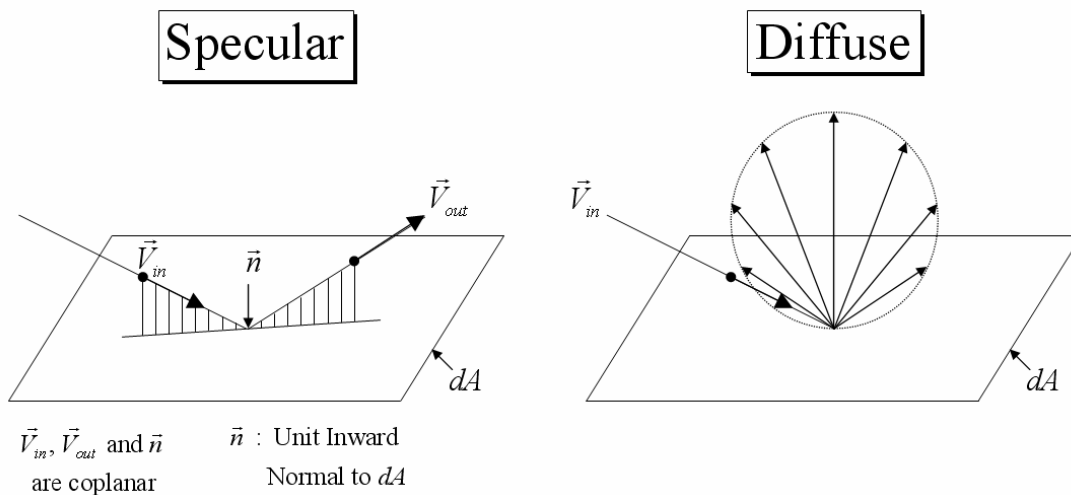


Figure 6: Specular and Diffuse Molecular Reflection

As the passage above says, reality falls somewhere between specular and diffuse.

Therefore, this analysis assumes partial surface accommodation of the incoming particles.

The following is adapted from Peter C. Hughes' book *Spacecraft Attitude Dynamics* and develops the equations used to calculate the disturbance and control torques on the spacecraft being studied here (2004:248-255).

Consider the force acting on an element of the spacecraft surface (dA) as shown in Figure 7:

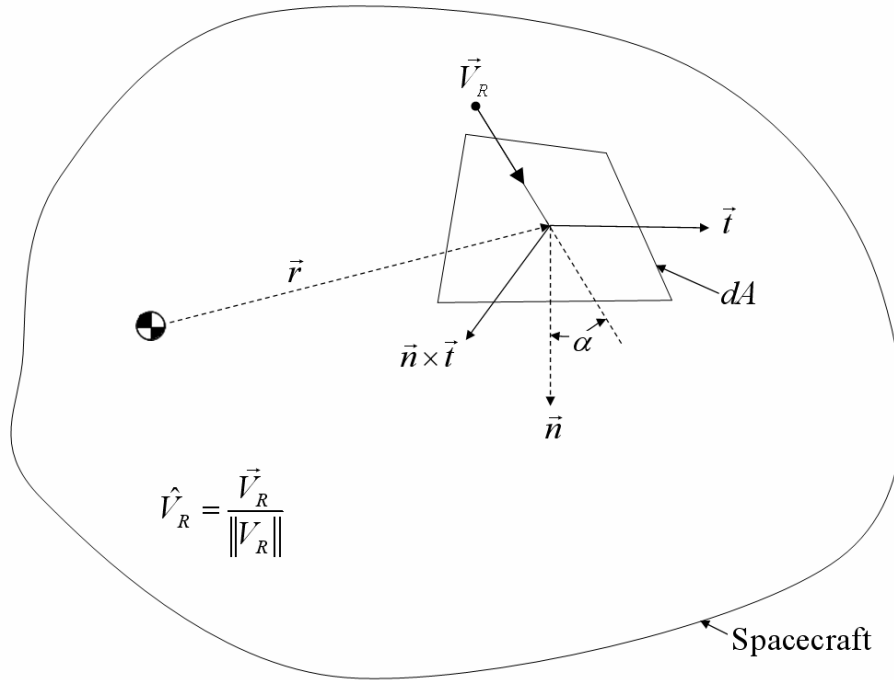


Figure 7: Molecules Incident on an Element of Spacecraft Surface

Using partial accommodation theory, the force imparted upon dA by the local atmosphere will have both components in the inward facing normal direction ($\vec{n}$) and components in the tangential direction ($\vec{t}$). So, the elemental force can be broken into a normal component ($d\vec{f}_n$) and a tangential component ($d\vec{f}_t$):

$$d\vec{f} = d\vec{f}_n + d\vec{f}_t \quad (3.14)$$

It is assumed that the relative velocity between the spacecraft and the atmosphere is dominated by the orbital velocity and rotation of the atmosphere, therefore, the random thermal motion of the individual gas molecules may be ignored and a local atmosphere velocity vector ($\vec{V}_R$) can be calculated to represent the magnitude and direction of the incoming molecules. The local atmosphere velocity unit vector ($\hat{V}_R$) combined with the surface inward facing normal ($\vec{n}$) is used to compute an angle of incidence (α) using a dot product:

$$\cos \alpha = \hat{V}_R \cdot \vec{n} \quad (3.15)$$

The projected area acted upon by the local atmosphere is $dA \cos \alpha$ in the normal direction and $dA \sin \alpha$ in the tangential direction. The momentum flux acting through the projected area is the force imparted to dA . Therefore, the force imparted to dA is:

$$d\vec{f} = \rho V_R^2 \cos \alpha \hat{V}_R dA + \rho V_R^2 \sin \alpha \hat{V}_R dA \quad (3.16)$$

where:

ρ : local atmospheric density

Before proceeding with this development, it is necessary to introduce a couple of concepts. First, not all surfaces of an orbiting spacecraft will impact upper atmosphere molecules. Some of the spacecraft will be shadowed and should not be considered in the force and torque calculations. In particular, if $\cos \alpha \geq 0$ that element is exposed to the flow while if $\cos \alpha < 0$ that element is shadowed. Therefore, the Heaviside function ($H(\cdot)$) is introduced to account for this in force and torque calculations (i.e. if $\cos \alpha \geq 0$, then $H(\cos \alpha) = 1$ and $H(\cos \alpha) = 0$ otherwise).

The next concept is a molecular exit velocity (V_b). This captures the velocity of those particles leaving the surface “with a probabilistic kinetic energy characteristic of the surface temperature and a probabilistic direction governed by a “cosine” distribution.” (2004:250). “According to the kinetic theory of gases, V_b is related to the surface temperature T_b as follows:

$$V_b = \left(\frac{\pi R T_b}{2 m} \right)^{1/2} \quad (3.17)$$

where R is the universal gas constant ($R = 8.314 \times 10^3 \text{ J/kg} \cdot \text{mole} \cdot ^\circ \text{C}$) and m is the molecular weight of the gas.”

Returning to the elemental force development, assume for a moment that all the molecules impacting the spacecraft are fully accommodated and reemitted diffusely. The diffuse normal and tangential elemental force on dA is then:

$$d\vec{f}_n^{(Diffuse)} = H(\cos \alpha) \rho V_R \cos \alpha [V_R \cos \alpha + V_b] \vec{n} dA \quad (3.18)$$

$$d\vec{f}_t^{(Diffuse)} = H(\cos \alpha) \rho V_R^2 \sin \alpha \cos \alpha \vec{t} dA \quad (3.19)$$

Assume now, the other extreme, specular reflection of incoming molecules. That is, molecules reflect with no change in speed and impart twice the normal component of the incoming molecules. The specular normal and tangential elemental force on dA is then:

$$d\vec{f}_n^{(Specular)} = 2 H(\cos \alpha) \rho V_R^2 \cos^2 \alpha \vec{n} dA \quad (3.20)$$

$$d\vec{f}_t^{(Specular)} = \vec{0} \quad (3.21)$$

Reality falls somewhere between these two extremes. Two factors are introduced to make these calculations more closely reflect reality: the normal accommodation

coefficient (σ_n) and the tangential accommodation coefficient (σ_t). Therefore, the

elemental normal and tangential force equations can be rewritten as:

$$d\vec{f}_n = \sigma_n d\vec{f}_n^{(Diffuse)} + (1 - \sigma_n) d\vec{f}_n^{(Specular)} \quad (3.22)$$

$$d\vec{f}_t = \sigma_t d\vec{f}_t^{(Diffuse)} + (1 - \sigma_t) d\vec{f}_t^{(Specular)} \quad (3.23)$$

Substituting the diffuse equations (3.18) and (3.19) and specular equations (3.20) and

(3.21) into the above relationships (3.22) and (3.23), realizing that $\vec{t} \sin \alpha = \hat{V}_R - \vec{n} \cos \alpha$,

and combining the equations per $d\vec{f} = d\vec{f}_n + d\vec{f}_t$ the partial accommodation elemental

force equation is:

$$d\vec{f} = H(\cos \alpha) \rho V_R^2 \cos \alpha \left\{ \left[(2 - \sigma_n - \sigma_t) \cos \alpha + \sigma_n \left(\frac{V_b}{V_R} \right) \right] \vec{n} + \sigma_t \hat{V}_R \right\} dA \quad (3.24)$$

To find the total force imparted upon the spacecraft by the local atmosphere, integrate

over the surface of the spacecraft which produces:

$$\vec{f} = \rho V_R^2 \left[\sigma_t A_p \hat{V}_R + \sigma_n \left(\frac{V_b}{V_R} \right) \vec{A}_p + (2 - \sigma_n - \sigma_t) \vec{A}_{pp} \right] \quad (3.25)$$

with A_p , $\vec{A}_p$, and $\vec{A}_{pp}$ defined as:

$$A_p \triangleq \oint\!\!\!\oint H(\cos \alpha) (\cos \alpha) dA \quad (3.26)$$

$$\vec{A}_p \triangleq \oint\!\!\!\oint H(\cos \alpha) (\cos \alpha) \vec{n} dA \quad (3.27)$$

$$\vec{A}_{pp} \triangleq \oint\!\!\!\oint H(\cos \alpha) (\cos^2 \alpha) \vec{n} dA \quad (3.28)$$

The torque can be calculated by crossing the distance from the center-of-mass to the

center of surface element ($\vec{r}$) and the force vector acting on that surface element ($d\vec{f}$)

and integrating over the entire surface:

$$\vec{g} = \oint\!\!\!\oint \vec{r} \times d\vec{f} = \rho V_R^2 \left[\sigma_t A_p \vec{c}_p \times \hat{V}_R + \sigma_n \left(\frac{V_b}{V_R} \right) \vec{G}_p + (2 - \sigma_n - \sigma_t) \vec{G}_{pp} \right] \quad (3.29)$$

with $\vec{G}_p$ and $\vec{G}_{pp}$ defined as:

$$\vec{G}_p \triangleq \oint\!\!\!\oint H(\cos \alpha)(\cos \alpha)(\vec{r} \times \vec{n}) dA \quad (3.30)$$

$$\vec{G}_{pp} \triangleq \oint\!\!\!\oint H(\cos \alpha)(\cos^2 \alpha)(\vec{r} \times \vec{n}) dA \quad (3.31)$$

$\vec{c}_p$ is the center of pressure vector from the spacecraft's center-of-mass and is defined by this equation:

$$A_p \vec{c}_p \triangleq \oint\!\!\!\oint H(\cos \alpha)(\cos \alpha) \vec{r} dA \quad (3.32)$$

Atmospheric Model

In this analysis, it was assumed that the only external torque on the spacecraft was due to aerodynamic forces. Local atmospheric particles impact the spacecraft with a particular magnitude and direction ($\vec{V}_R$) and density (ρ) defined by an atmospheric model which assumes a rotating Earth atmosphere with an exponential density profile.

The local atmosphere velocity vector ($\vec{V}_R$) for a rotating Earth atmosphere is described by the following equation (Ravindran, 1972:500):

$$\vec{V}_R = V \left[1 - \left(\frac{\omega_E R}{V} \right) \cos i \right] R_{ORB \rightarrow PRIN} \begin{pmatrix} -1 \\ \left(\frac{\omega_E R}{V} \right) \sin i \cos \nu \\ 0 \end{pmatrix} \quad (3.33)$$

where:

V : magnitude of $\vec{V}$

R : magnitude of $\vec{R}$

ω_E : Earth's rotation rate (7.27×10^{-5} radians/second)

Other disturbances such as solar radiation pressure, Earth magnetic variations and oblateness, and day/night/solar activity atmospheric density variations were neglected.

Aerodynamic density is an area of significant uncertainty and constant change.

For this analysis, an exponential model was used which is described by (Vallado, 2001:534-537):

$$\rho = \rho_o \exp \left[-\frac{h_{ellip} - h_o}{H} \right] \quad (3.34)$$

where:

ρ_o : reference density h_o : reference altitude
 h_{ellip} : actual altitude H : scale height

Values for h_{ellip} is defined by the orbit chosen in any simulation run while values for ρ_o , h_o , and H are obtained from the following table (2001:537):

Altitude h_{ellip} (km)	Base Altitude h_o (km)	Nominal Density ρ_o (kg/m ³)	Scale Height H (km)	Altitude h_{ellip} (km)	Base Altitude h_o (km)	Nominal Density ρ_o (kg/m ³)	Scale Height H (km)
0–25	0	1.225	7.249	150–180	150	2.070×10^{-9}	22.523
25–30	25	3.899×10^{-2}	6.349	180–200	180	5.464×10^{-10}	29.740
30–40	30	1.774×10^{-2}	6.682	200–250	200	2.789×10^{-10}	37.105
40–50	40	3.972×10^{-3}	7.554	250–300	250	7.248×10^{-11}	45.546
50–60	50	1.057×10^{-3}	8.382	300–350	300	2.418×10^{-11}	53.628
60–70	60	3.206×10^{-4}	7.714	350–400	350	9.518×10^{-12}	53.298
70–80	70	8.770×10^{-5}	6.549	400–450	400	3.725×10^{-12}	58.515
80–90	80	1.905×10^{-5}	5.799	450–500	450	1.585×10^{-12}	60.828
90–100	90	3.396×10^{-6}	5.382	500–600	500	6.967×10^{-13}	63.822
100–110	100	5.297×10^{-7}	5.877	600–700	600	1.454×10^{-13}	71.835
110–120	110	9.661×10^{-8}	7.263	700–800	700	3.614×10^{-14}	88.667
120–130	120	2.438×10^{-8}	9.473	800–900	800	1.170×10^{-14}	124.64
130–140	130	8.484×10^{-9}	12.636	900–1000	900	5.245×10^{-15}	181.05
140–150	140	3.845×10^{-9}	16.149	1000–	1000	3.019×10^{-15}	268.00

Figure 8: Density Values

Aerodynamic Torque Application

To apply partial accommodation theory to the spacecraft geometry being used in this analysis, it is first necessary to introduce a numbering system for the spacecraft's exposed surfaces (i.e. spacecraft bus surfaces and control panel surfaces) and to introduce general dimensions so that radius vectors from the spacecraft's center-of-mass to the surface's geometric center ($\vec{r}_{surface}$) and the inward-facing normal vectors ($\vec{n}_{surface}$) can be defined in general for every surface. There are ten exposed surfaces for the spacecraft designed here. The number scheme for those ten surfaces is (also see Figure 9):

- Surface 1: $\hat{b}_1$ axis side of spacecraft bus (front)
- Surface 2: $\hat{b}_2$ axis side of spacecraft bus (left)
- Surface 3: Opposite $\hat{b}_2$ axis side of spacecraft bus (right)
- Surface 4: Opposite $\hat{b}_3$ axis side of spacecraft bus (top)
- Surface 5: $\hat{b}_3$ axis side of spacecraft bus (bottom)
- Surface 6: Opposite $\hat{b}_1$ axis side of spacecraft bus (back)
- Surface 7: Top-Left Control Panel (front & rear of panel)
- Surface 8: Top-Right Control Panel (front & rear of panel)
- Surface 9: Bottom-Left Control Panel (front & rear of panel)
- Surface 10: Bottom-Right Control Panel (front & rear of panel)

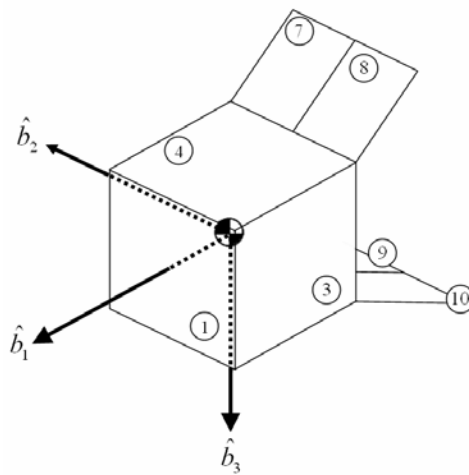


Figure 9: Analysis Numbering Scheme

The general dimensions used in this analysis are shown in Figure 10. The dimension d is the distance from the spacecraft bus' center-of-mass to the edge of the cube. The center-of-mass is assumed to be placed perfectly at the geometric center of the spacecraft bus. The dimension L is half the length of a control panel. The dimension f is the horizontal distance from the center-of-mass of the spacecraft to the center line of the control panels. The dimension w is the width of the control panels.

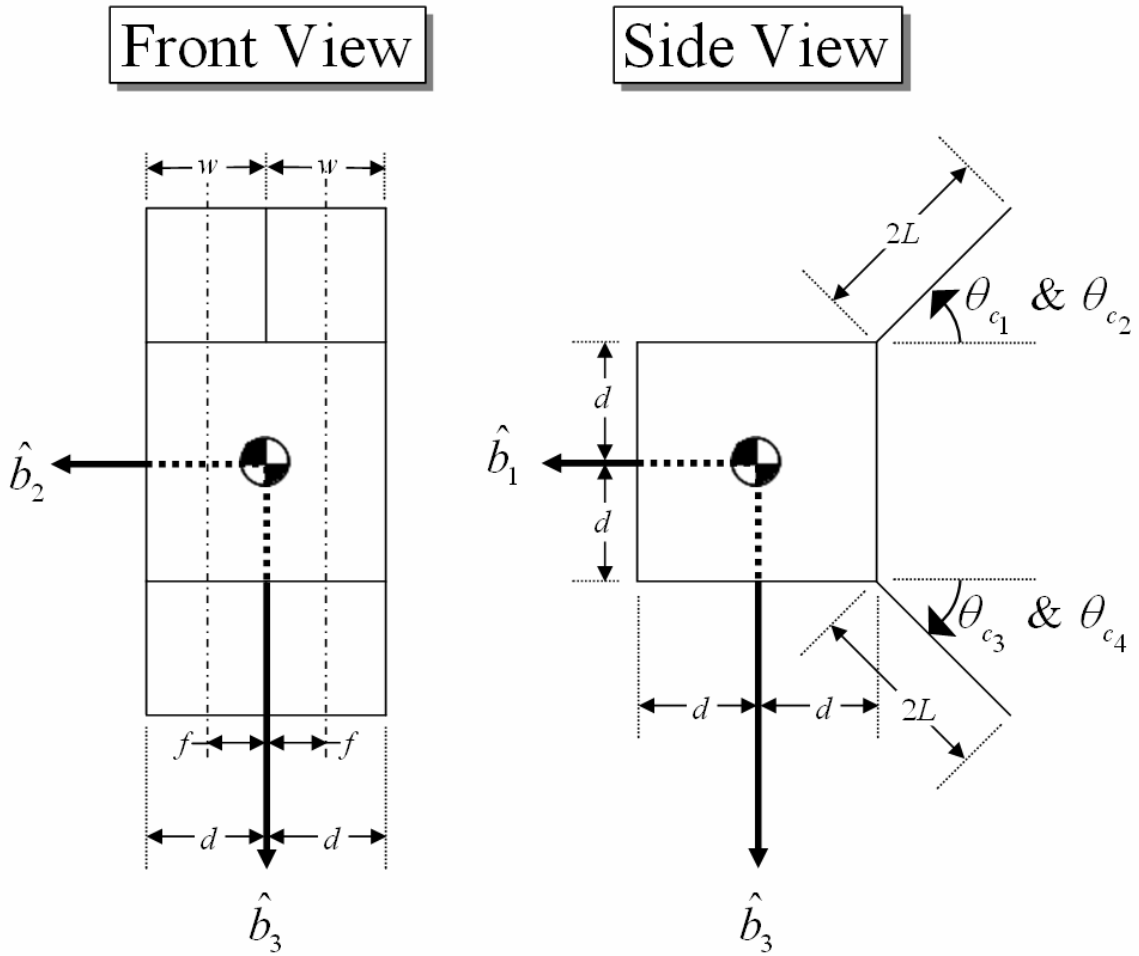


Figure 10: Spacecraft Dimensions

With these general dimensions defined, the following $\vec{r}_{surface}$ and $\vec{n}_{surface}$ vectors were defined (inward facing normals were defined for the front and rear of each control panel so that off nominal configurations could be analyzed):

Spacecraft Bus Vectors

$$\begin{aligned} \vec{r}_1 &= \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} & \vec{n}_1 &= \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} & \vec{r}_2 &= \begin{bmatrix} 0 \\ d \\ 0 \end{bmatrix} & \vec{n}_2 &= \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} & \vec{r}_3 &= \begin{bmatrix} 0 \\ -d \\ 0 \end{bmatrix} & \vec{n}_3 &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ \vec{r}_4 &= \begin{bmatrix} 0 \\ 0 \\ -d \end{bmatrix} & \vec{n}_4 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \vec{r}_5 &= \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} & \vec{n}_5 &= \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} & \vec{r}_6 &= \begin{bmatrix} -d \\ 0 \\ 0 \end{bmatrix} & \vec{n}_6 &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

Control Panel Vectors

$$\begin{aligned} \vec{r}_7 &= \begin{bmatrix} -d - L \cos(\theta_{c_1}) \\ f \\ -d + L \cos(\theta_{c_1}) \end{bmatrix} & \vec{r}_8 &= \begin{bmatrix} -d - L \cos(\theta_{c_2}) \\ -f \\ -d + L \cos(\theta_{c_2}) \end{bmatrix} \\ \vec{n}_{7,front} &= \begin{bmatrix} \sin(\theta_{c_1}) \\ 0 \\ \cos(\theta_{c_1}) \end{bmatrix} = -\vec{n}_{7,rear} & \vec{n}_{8,front} &= \begin{bmatrix} \sin(\theta_{c_2}) \\ 0 \\ \cos(\theta_{c_2}) \end{bmatrix} = -\vec{n}_{8,rear} \end{aligned}$$

$$\begin{aligned} \vec{r}_9 &= \begin{bmatrix} -d - L \cos(\theta_{c_3}) \\ f \\ d + L \cos(\theta_{c_3}) \end{bmatrix} & \vec{r}_{10} &= \begin{bmatrix} -d - L \cos(\theta_{c_4}) \\ -f \\ d + L \cos(\theta_{c_4}) \end{bmatrix} \\ \vec{n}_{9,front} &= \begin{bmatrix} -\sin(\theta_{c_3}) \\ 0 \\ -\cos(\theta_{c_3}) \end{bmatrix} = -\vec{n}_{9,rear} & \vec{n}_{10,front} &= \begin{bmatrix} -\sin(\theta_{c_4}) \\ 0 \\ -\cos(\theta_{c_4}) \end{bmatrix} = -\vec{n}_{10,rear} \end{aligned}$$

With $\vec{r}_{surface}$ and $\vec{n}_{surface}$ defined, the partial accommodation aerodynamic torque equations can be applied to every surface (with ρ and $\vec{V}_R$ being defined for the orbit selected):

$$\vec{g}_{surface} = \rho V_R^2 \left[\sigma_t A_p \vec{c}_{p_{surface}} \times \hat{V}_R + \sigma_n \left(\frac{V_b}{V_R} \right) \vec{G}_p + (2 - \sigma_n - \sigma_t) \vec{G}_{pp} \right] \quad (3.35)$$

with:

$$\alpha_{surface} = \cos^{-1} \left(\hat{V}_R \bullet \vec{n}_{surface} \right) \quad (3.36)$$

$$A_p \triangleq H \left(\cos \alpha_{surface} \right) \left(\cos \alpha_{surface} \right) A_{surface} \quad (3.37)$$

$$\vec{c}_{p_{surface}} = \vec{r}_{surface} \quad (3.38)$$

$$\vec{G}_p \triangleq H \left(\cos \alpha_{surface} \right) \left(\cos \alpha_{surface} \right) \left(\vec{r}_{surface} \times \vec{n}_{surface} \right) A_{surface} \quad (3.39)$$

$$\vec{G}_{pp} \triangleq H \left(\cos \alpha_{surface} \right) \left(\cos^2 \alpha_{surface} \right) \left(\vec{r}_{surface} \times \vec{n}_{surface} \right) A_{surface} \quad (3.40)$$

$\vec{c}_{p_{surface}} = \vec{r}_{surface}$ for all spacecraft surfaces since only flat surfaces are considered and

spacecraft bus shadowing of the control panels is not incorporated (i.e. flow is not cut off to some/all of the control panel surfaces by the spacecraft bus). Although the exact value of σ_t and σ_n depend on spacecraft materials, incident angle, molecular flow velocity, and surface temperature, approximate values are available in Hughes. Both coefficients are assumed to be 0.8 in this analysis. Also, Hughes notes that V_b is typically 5 % of V_R at altitudes of interest, therefore, V_b is assumed to be 5 % of V_R in this analysis.

Finally, the aerodynamic disturbance torque ($\vec{g}_d$) on the spacecraft bus and the aerodynamic control torque ($\vec{g}_c$) being generated by the control panels can be calculated by summing each surfaces contribution:

$$\vec{g}_d = \sum_{n=1}^6 \vec{g}_{surface, n} \quad (3.41)$$

$$\vec{g}_c = \sum_{k=7}^{k=10} g_{c_{control\ panel\ front, k}} + \sum_{k=7}^{k=10} g_{c_{control\ panel\ rear, k}} \quad (3.42)$$

Control Law

Attitude control is achieved using outer and inner feedback loops. The control law for the outer loop computes a desired control torque based on orientation error and angular velocity using an approach called quaternion feedback reorientation. The inner loop then controls the panel angles to achieve the desired torque. The control panel angles will be driven at rates much higher than the expected spacecraft angular velocities, so it is assumed that the independent stability of each loop will result in an overall stable controller. Simulations support this assumption, as will be shown in the cases later in this thesis.

Referring back to the rotational equations of motion (3.13):

$$\dot{\vec{\omega}} = -I^{-1}\vec{\omega}^\times I\vec{\omega} + I^{-1}\vec{g}_d + I^{-1}\vec{g}_c$$

The disturbance torque on the spacecraft bus ($\vec{g}_d$) is determined by the orientation of the spacecraft and the control torque ($\vec{g}_c$) is determined by the orientation of the control panels as well as the orientation of the spacecraft.

Outer Loop Control Law

The outer loop control law used to determine the desired control torque is quaternion feedback reorientation (Wie, 1998:402-4):

$$\vec{u}_{desired} = -K \underline{q}_e - C \vec{\omega} \quad (3.43)$$

where $\underline{q}_e$ is the first three components of the error quaternion ($\vec{q}_e$) (to be defined below), $\vec{\omega}$ are the measured body angular rates, and K and C are positive definite gain matrices (this analysis uses scalar gains multiplied by an identity matrix). The error quaternion

$(\vec{q}_e)$ is calculated using the commanded attitude quaternion $(q_{1_c}, q_{2_c}, q_{3_c}, q_{4_c})$ and the current calculated attitude quaternions (q_1, q_2, q_3, q_4) via the following equation:

$$\vec{q}_e = \begin{bmatrix} q_{1_e} \\ q_{2_e} \\ q_{3_e} \\ q_{4_e} \end{bmatrix} = \begin{bmatrix} q_{4_c} & q_{3_c} & -q_{2_c} & -q_{1_c} \\ -q_{3_c} & q_{4_c} & q_{1_c} & -q_{2_c} \\ q_{2_c} & -q_{1_c} & q_{4_c} & -q_{3_c} \\ q_{1_c} & q_{2_c} & q_{3_c} & q_{4_c} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (3.44)$$

The actual torque generated by the control panels $(\vec{g}_c)$ depends on the control panel angles and the spacecraft orientation. The inner loop controller ensures that $\vec{g}_c$ tracks $\vec{u}_{desired}$.

Stability of the outer loop can be shown asymptotically stable via Lyapunov's direct method when commanded to the origin $(\vec{q}_c = [0 \ 0 \ 0 \ \pm 1]^T)$ with (Wie et al., 1989:375-80):

$$K = k I_{3 \times 3}$$

$$C = \text{diagonal}(c_1, c_2, c_3)$$

k and c_i : positive scalar constants

Simulations performed in this analysis command to the origin and offset orientations within the vicinity of the origin. Therefore, the simulations commanded to the origin have a shown stable outer loop, and simulations commanded to something other than the origin have an assumed stable outer loop. Simulations support this assumption.

Inner Loop Control Law

The inner loop control law uses methods similar to those used in robotic manipulator control (Titus, 1998:69-71). At any moment, the error between the generated torque ($\vec{g}_c$) and the desired torque ($\vec{u}_{desired}$) can be written:

$$\vec{e} = \vec{g}_c - \vec{u}_{desired}$$

Differentiating the torque error velocity yields:

$$\dot{\vec{e}} = \dot{\vec{g}}_c - \dot{\vec{u}}_{desired} \quad (3.45)$$

On the time scale of interest in the inner loop, it is assumed that the desired control torque is constant, therefore, $\dot{\vec{u}}_{desired} = 0$ and:

$$\dot{\vec{e}} = \dot{\vec{g}}_c \quad (3.46)$$

The control torque is a non-linear function of spacecraft orientation and control panel orientation. Although it is a straightforward computation to find $\vec{g}_c$, given the current spacecraft orientation and control panel angles, the inverse is not easily found due to the cascading nature of the calculations. To overcome this difficulty, $\vec{g}_c$ can be linearized at the current state and the resulting Jacobian (A) can be used for the necessary inverse relationship. A linearized version of the control torque velocity is:

$$\dot{\vec{g}}_c = A \dot{\vec{\theta}}_c \quad (3.47)$$

where the Jacobian (A) is:

$$A = \begin{bmatrix} \frac{\partial \vec{g}_c}{\partial \theta_{c_1}} & \frac{\partial \vec{g}_c}{\partial \theta_{c_2}} & \frac{\partial \vec{g}_c}{\partial \theta_{c_3}} & \frac{\partial \vec{g}_c}{\partial \theta_{c_4}} \end{bmatrix} \quad (3.48)$$

Applying proportional feedback control to the error velocity yields:

$$\dot{\vec{e}} = -k_f \vec{e} = -k_f (\vec{g}_c - \vec{u}_{desired}) \quad (3.49)$$

Stability of the inner loop is assured as long as k_f is positive as the error will be driven to zero over time. Substituting (3.47) and (3.49) into (3.46) gives:

$$A \dot{\vec{\theta}}_c = -k_f (\vec{g}_c - \vec{u}_{desired}) \quad (3.50)$$

The needed value from this relationship is $\dot{\vec{\theta}}_c$ as it can be numerically integrated to calculate new control panel orientations that drive the spacecraft toward the commanded orientation. Since the A matrix is not square, it has no inverse. However, the pseudo-inverse ($A^\#$) may be used to find the minimum norm solution for $\dot{\vec{\theta}}_c$. This solution represents the least possible movement of the control panels to achieve the desired torque, giving the inner loop control law:

$$\dot{\vec{\theta}}_c = -k_f A^\# (\vec{g}_c - \vec{u}_{desired}) \quad (3.51)$$

where:

$$A^\# = (A^T A)^{-1} A^T \quad (3.52)$$

However, this control law may not be optimal in the sense of spacecraft operations as it can drive the control panels toward saturation. Saturation avoidance is added to the inner loop control law by adding null motion (i.e. non-torque producing motion) to keep the control panels near their preferred $\pm 45^\circ$ default orientation (Kuhns, 1994:2892-3). To add saturation avoidance to (3.51), we first define a control panel angle error ($\vec{\theta}_{c,error}$):

$$\vec{\theta}_{c,error} \triangleq \vec{\theta}_{c,preferred} - \vec{\theta}_c \quad (3.53)$$

where:

$$\begin{aligned} \vec{\theta}_{c,preferred} &= [-45^\circ \quad -45^\circ \quad 45^\circ \quad 45^\circ]^T \\ \vec{\theta}_c &= [\theta_{c_1} \quad \theta_{c_2} \quad \theta_{c_3} \quad \theta_{c_4}]^T \end{aligned}$$

Saturation avoidance is accomplished by projecting the control panel angle error onto the right null-space of A and adding the proper amount to the minimum norm solution. This will result in convergence of the control panel angle trajectories to the preferred orientation angles. Therefore, the inner loop control law with saturation avoidance is (P is the null-space projection matrix and k_p is a positive gain):

$$\dot{\vec{\theta}}_c = -k_f A^\# (\vec{g}_c - \vec{u}_{desired}) + k_p P \vec{\theta}_{c,error} \quad (3.54)$$

where:

$$P = I - A^\# A \quad (3.55)$$

With the stability of the outer and inner loops established independently, spacecraft stability with the combined outer/inner loop controller is likely but not guaranteed. Without a robust stability analysis for the combined outer/inner loop controller (such as Lyapunov's direct method), extensive simulations are necessary to convince the control designer of the outer/inner loop controller's stability before considering operational use of the controller.

MATLAB Implementation

The equations described in this thesis were coded in MATLAB so that this spacecraft's design and controller performance could be evaluated via simulation. MATLAB's ODE45 differential equation solver was used for these simulations with the states being spacecraft quaternions ($\vec{q}$), spacecraft angular rates ($\vec{\omega}$), and control panel angles (θ_{c_1} , θ_{c_2} , θ_{c_3} , and θ_{c_4}). The initial conditions for the spacecraft quaternions and spacecraft angular rates were defined for the case being analyzed and the initial conditions for the control panel angles were the preferred control panel angles

($\theta_{c_1} = \theta_{c_2} = -45^\circ$ and $\theta_{c_3} = \theta_{c_4} = 45^\circ$). Saturation limits were placed on their movements to they stayed within $\pm 45^\circ$ of their initial condition. A commanded quaternion was calculated based on the desired final Euler Angles and was passed to the solver for use in the outer loop control law. The time frame for each simulation was selected based on the expected number of orbital periods required for maneuvering to the commanded final position. The simulation code is shown in Appendix A.

The Jacobian matrix used in the inner loop control law was evaluated at every time step so a symbolic Jacobian was necessary for inclusion in the simulation code. This symbolic Jacobian was calculated via MATLAB's symbolic toolbox by the code shown in Appendix B.

IV. Results

Chapter Overview

Simulations were performed for a variety of cases to investigate the effectiveness of the actuators and the controller. The cases explored uncontrolled behavior of the spacecraft (to confirm the validity of assumptions made in this analysis), controlled performance from a variety of initial conditions and desired final orientations, and the impact of including saturation avoidance in the inner loop control law. Uncontrolled simulations verified assumptions made during the development of this analysis, controlled simulations showed that three-axis control is possible over a range of initial angles and angular rates, and saturation avoidance inclusion in the inner loop control law does keep the control panels in an orientation which maximizes control authority for future maneuvers.

Simulation Results

As stated in the chapter overview, simulations looked at uncontrolled and controlled performance of the spacecraft designed and the control law developed in this thesis. The uncontrolled simulation looks to confirm the assumption of passive stability about the pitch and yaw axes. Passive stability is assumed because the center-of-pressure is placed to the rear of the center-of-mass (similar to Psiaki's shuttlecock design).

To complete these simulations a number of parameters were held constant while others were varied. Constant parameters were:

orbital altitude = 300 km
spacecraft mass = 10 kg
spacecraft bus dimensions = 0.5 m × 0.5 m × 0.5 m
control panel dimensions = 0.25 m × 0.4 m

Variable parameters were:

spacecraft initial Euler Angles: $\theta_{1,initial}$, $\theta_{2,initial}$, $\theta_{3,initial}$
spacecraft commanded final Euler Angles: $\theta_{1,final}$, $\theta_{2,final}$, $\theta_{3,final}$
spacecraft initial rotation rates: ω_1 , ω_2 , ω_3
orbital inclination: i
quaternion feedback reorientation gains: K & C
minimum norm solution gain: k_f
saturation avoidance gain: k_p

Case 1

This case explored the uncontrolled performance of the spacecraft and looks to confirm the assumption of passive stability about the pitch and yaw axes. The spacecraft in this case is placed in an equatorial orbit in a significant tumble and at a significant initial Euler Angle offset. The parameters selected for this case are shown below.

$$\begin{array}{ll}
\theta_{1,initial} = -13^\circ & \\
\theta_{2,initial} = 101^\circ & K = 0 \\
\theta_{3,initial} = -26^\circ & C = 0 \\
\theta_{1,final} = \theta_{2,final} = \theta_{3,final} = 0^\circ & k_f = 0 \\
\omega_1 = -3.0 \text{ deg/sec} & k_p = 0 \\
\omega_2 = -2.0 \text{ deg/sec} & i = 0^\circ \\
\omega_3 = 2.5 \text{ deg/sec} &
\end{array}$$

(Final Euler Angles are shown with a commanded value of 0° , but this value is really not applicable for the uncontrolled case at hand and is only included so MATLAB will run without error.)

Spacecraft performance for this uncontrolled simulation was as expected. The spacecraft reoriented itself with the roll axis pointed primarily along the spacecraft's orbit velocity vector and passive stability about the pitch and yaw axes was demonstrated. Figure 11 shows the spacecraft's Euler Angles near the end of the 1000 uncontrolled orbits. During the time period shown, the pitch and yaw axes' Euler Angles are wobbling about zero with max values less than 10° . The roll axis' Euler Angle, however, continues to flip from -90° and 90° as the spacecraft continually rotates about that axis. Figure 12, Figure 13, and Figure 14 show uncontrolled angular rates for the roll, pitch, and yaw axes, respectively, for the full 1000 orbits. Interestingly, the pitch and yaw axes demonstrate slight damping over the 1000 orbits (as shown in Figure 13 and Figure 14). The roll axis angular rates decrease as the spacecraft reorients itself from the initial tumble (see Figure 12), but those rates level off at -0.01 rad/sec. The case confirmed that the model used in this analysis: demonstrates passive pitch and yaw axis stability, injects slight damping of the pitch and yaw axis angular rates, and allows the roll axis to rotate undamped in an uncontrolled configuration.

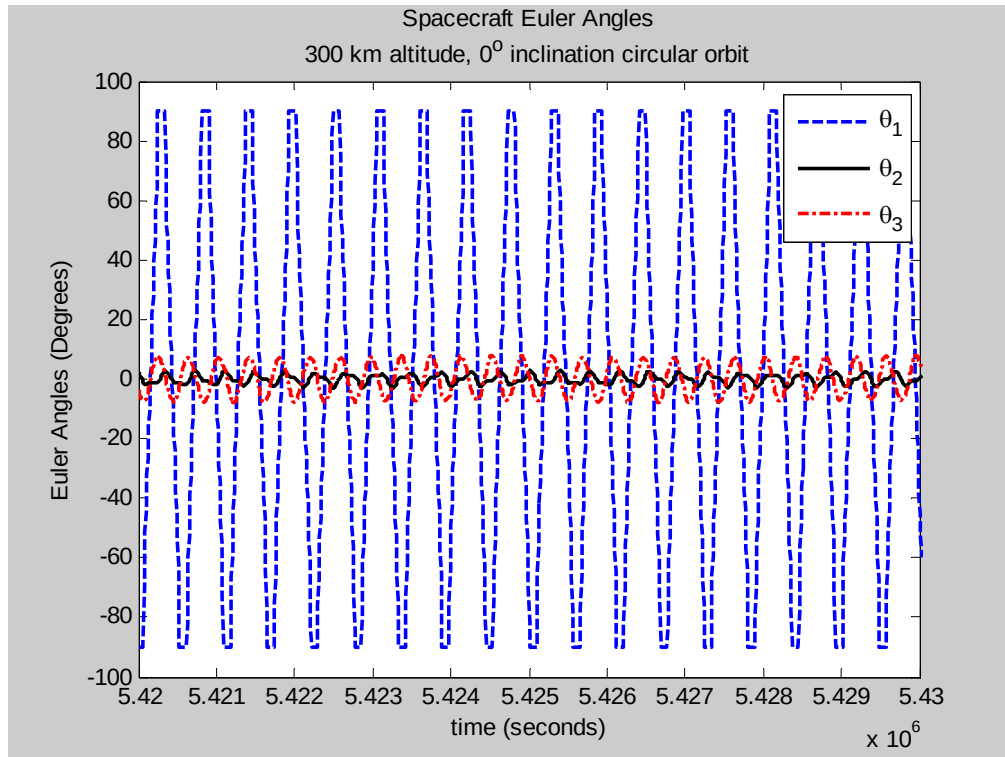


Figure 11: Spacecraft Euler Angles – Case 1 (Uncontrolled)

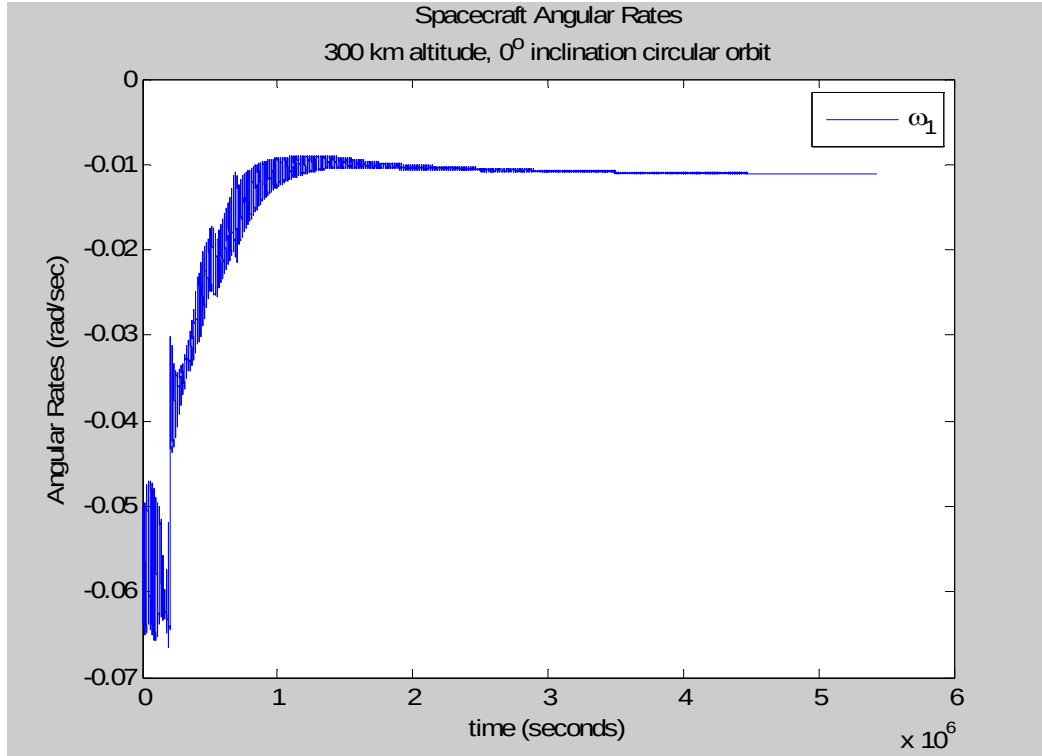


Figure 12: Spacecraft Roll Axis Angular Rates – Case 1 (Uncontrolled)

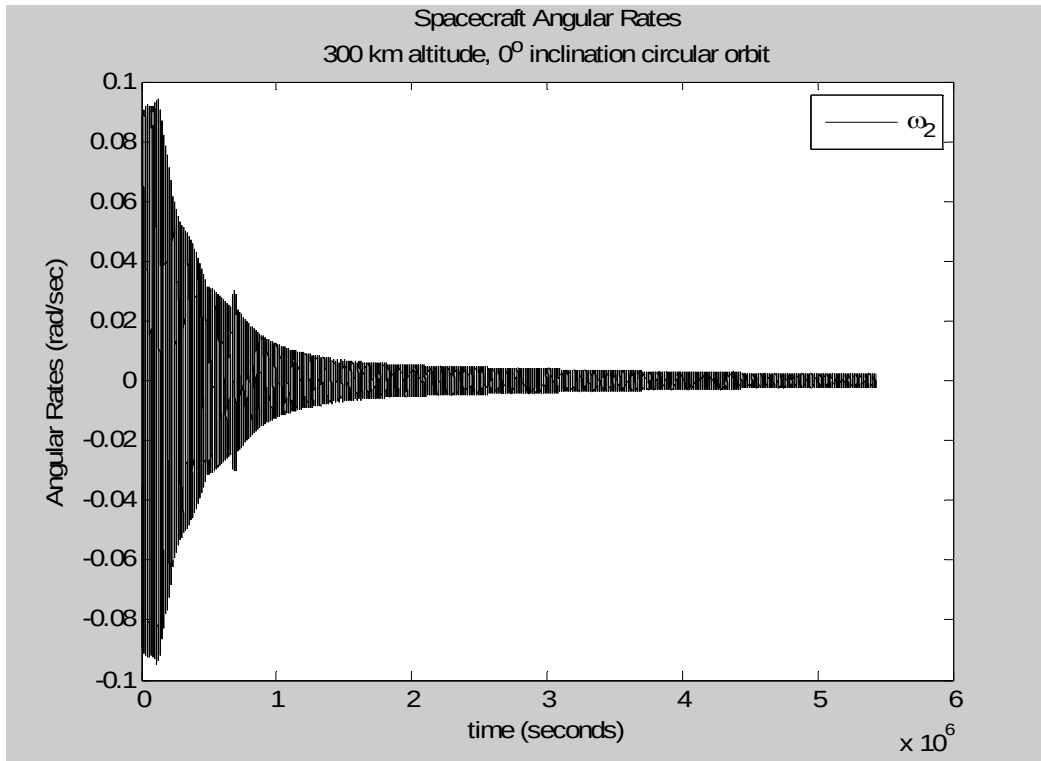


Figure 13: Spacecraft Pitch Axis Angular Rates – Case 1 (Uncontrolled)

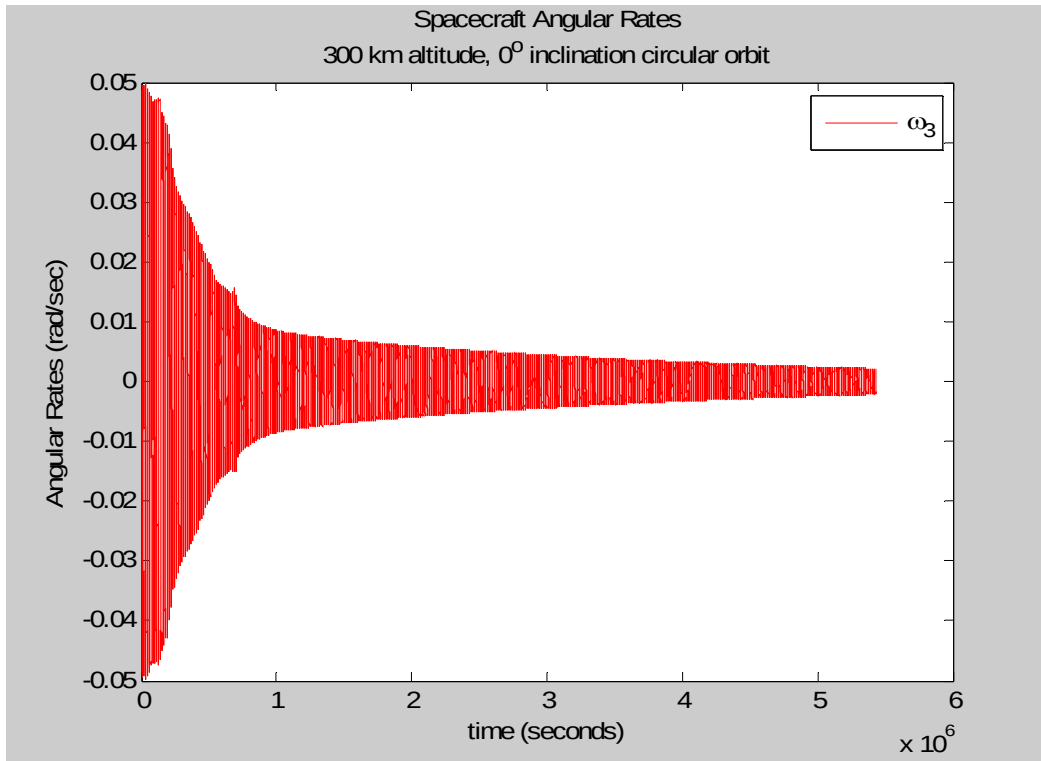


Figure 14: Spacecraft Yaw Axis Angular Rates – Case 1 (Uncontrolled)

Case 2

This case looked at a non-tumbling spacecraft in an equatorial orbit commanded to equilibrium without saturation avoidance. Equilibrium in this thesis is defined as $\theta_{1,final} = \theta_{2,final} = \theta_{3,final} = 0^\circ$. This case is somewhat intuitive since the interaction between the rotating atmosphere and the orbital motion does not cause a time varying atmospheric velocity vector ($\vec{V}_R$), as is the case for inclined orbits. The parameters selected for this simulation were:

$$\begin{array}{ll}
 \theta_{1,initial} = 7^\circ & K = 0.001 \\
 \theta_{2,initial} = 5^\circ & C = 0.2 \\
 \theta_{3,initial} = -5^\circ & k_f = 0.02 \\
 \theta_{1,final} = \theta_{2,final} = \theta_{3,final} = 0^\circ & k_p = 0 \\
 \omega_1 = \omega_2 = \omega_3 = 0 \text{ deg/sec} & i = 0^\circ
 \end{array}$$

The time history of spacecraft orientation, rotation rate, and control panel angles are given in Figure 15, Figure 16, and Figure 17, respectively. Figure 15 shows that the desired final orientation was achieved in approximately 2000 seconds. Of particular interest in this case is the final orientation of the control panels as shown in Figure 17. For this fairly benign case, it was expected that the panel angles would remain constant and in symmetric positions ($\theta_{c_4} = -\theta_{c_1}$ and $\theta_{c_3} = -\theta_{c_2}$), but it should be possible for the spacecraft to maintain this attitude with control panels at the preferred angle positions ($\pm 45^\circ$). This is important to maximize control authority in readiness for future maneuvers. The addition of the saturation avoidance term rectifies this problem, as shown in Case 3.

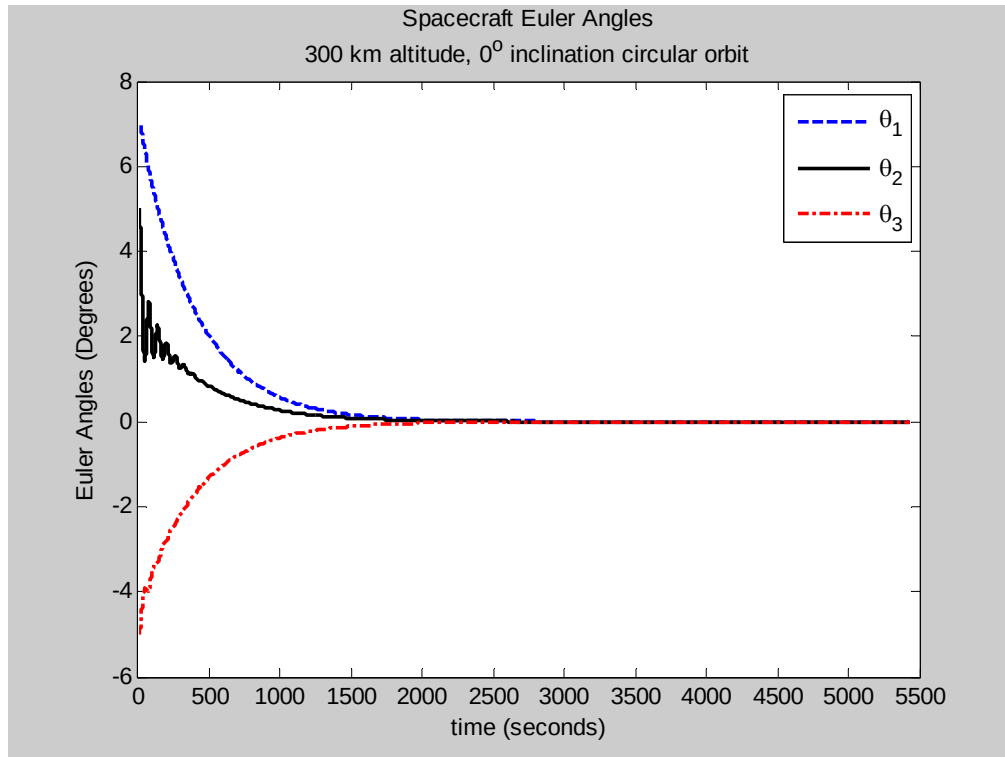


Figure 15: Euler Angles – Case 2 (Controlled)

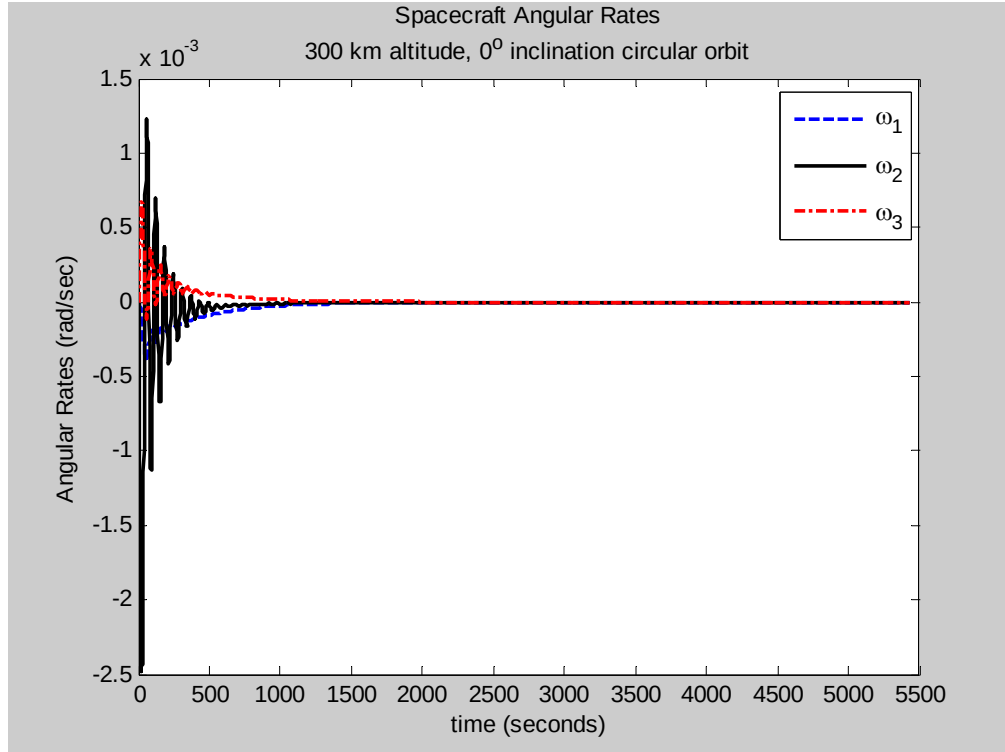


Figure 16: Spacecraft Angular Rates – Case 2 (Controlled)

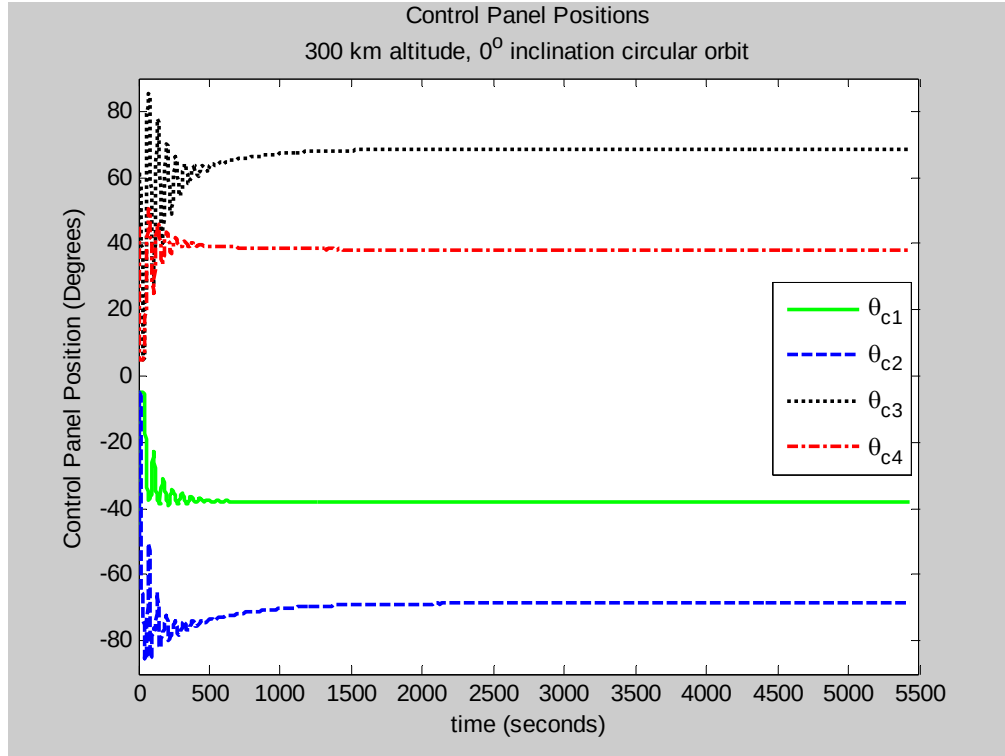


Figure 17: Control Panel Angles without Saturation Avoidance – Case 2 (Controlled)

Case 3

This case looks at Case 2 again but with saturation avoidance enabled with $k_p = 0.02$. The spacecraft orientation and angular rates are very similar to Case 2 but the control panel angles differ as expected. As shown in Figure 18, the control panels now reorient themselves to the desired final orientation of $\pm 45^\circ$ while still properly orienting the spacecraft.

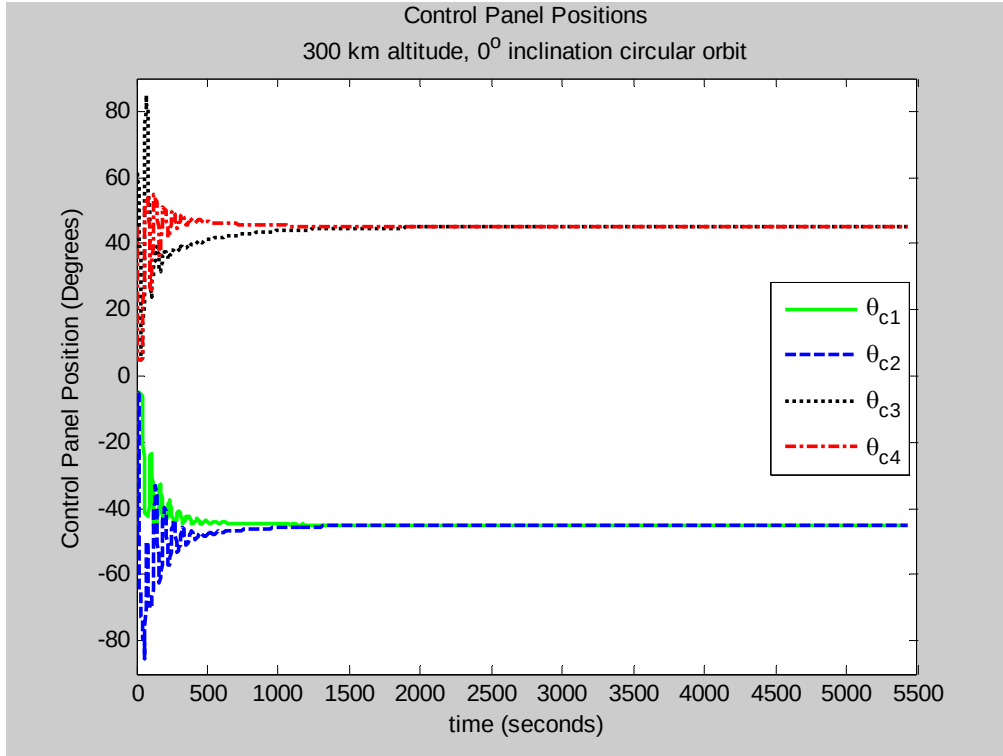


Figure 18: Control Panel Angles with Saturation Avoidance – Case 3 (Controlled)

Case 4

This case once again looks at a spacecraft in a non-tumbling equatorial orbit. This case includes saturation avoidance and a commanded final offset. Here are the parameters used for this case:

$$\begin{aligned}
 \theta_{1,initial} &= 7^\circ \\
 \theta_{2,initial} &= 5^\circ \\
 \theta_{3,initial} &= -5^\circ \\
 \theta_{1,final} &= -3^\circ \\
 \theta_{2,final} &= 2^\circ \\
 \theta_{3,final} &= 4^\circ \\
 \omega_1 = \omega_2 = \omega_3 &= 0 \text{ deg/sec}
 \end{aligned}
 \qquad
 \begin{aligned}
 K &= 0.001 \\
 C &= 0.2 \\
 k_f &= 0.02 \\
 k_p &= 0.02 \\
 i &= 0^\circ
 \end{aligned}$$

As shown in Figure 19, the desired final orientation was again achieved in approximately 2000 seconds. The angular rates settled out as expected. As shown in Figure 20, the control panels tended toward the desired final orientation but do not return to exactly $\pm 45^\circ$ because an offset is required to hold the desired final orientation.

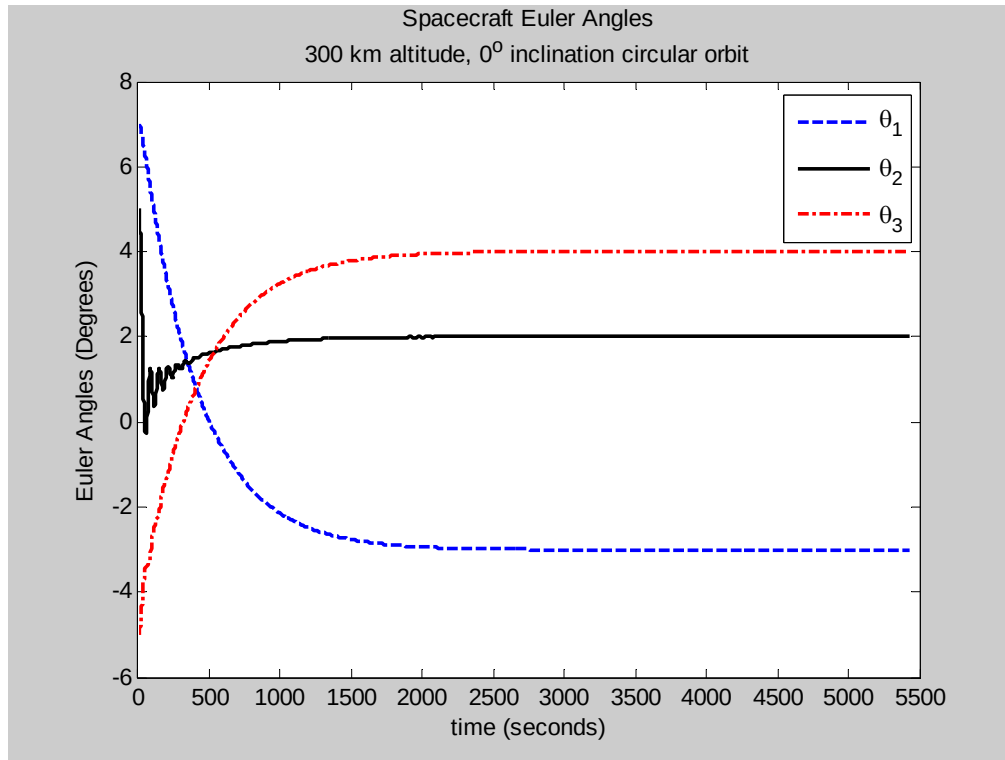


Figure 19: Euler Angles – Case 4 (Controlled)

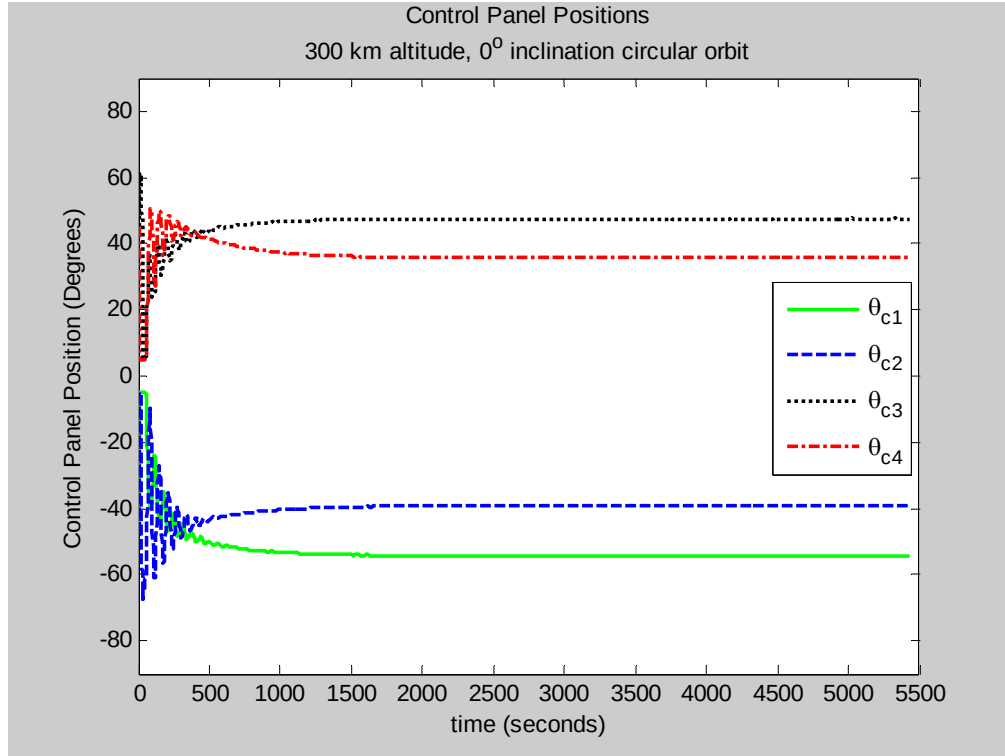


Figure 20: Control Panel Angles – Case 4 (Controlled)

Case 5

This case builds upon Case 4 by adding orbital inclination into the simulation ($i = 45^\circ$). All other parameters are the same as those used in Case 4. As Figure 21 shows, the Euler angle time history is very similar to Case 4 and converges to the desired final orientation in approximately the same time. The control panel angle plot (Figure 22) differs as expected as the control panels have to adjust while encountering the rotating atmosphere to maintain the proper final orientation. Also, note that the control panels angles trend toward the $\pm 45^\circ$ desired orientation.

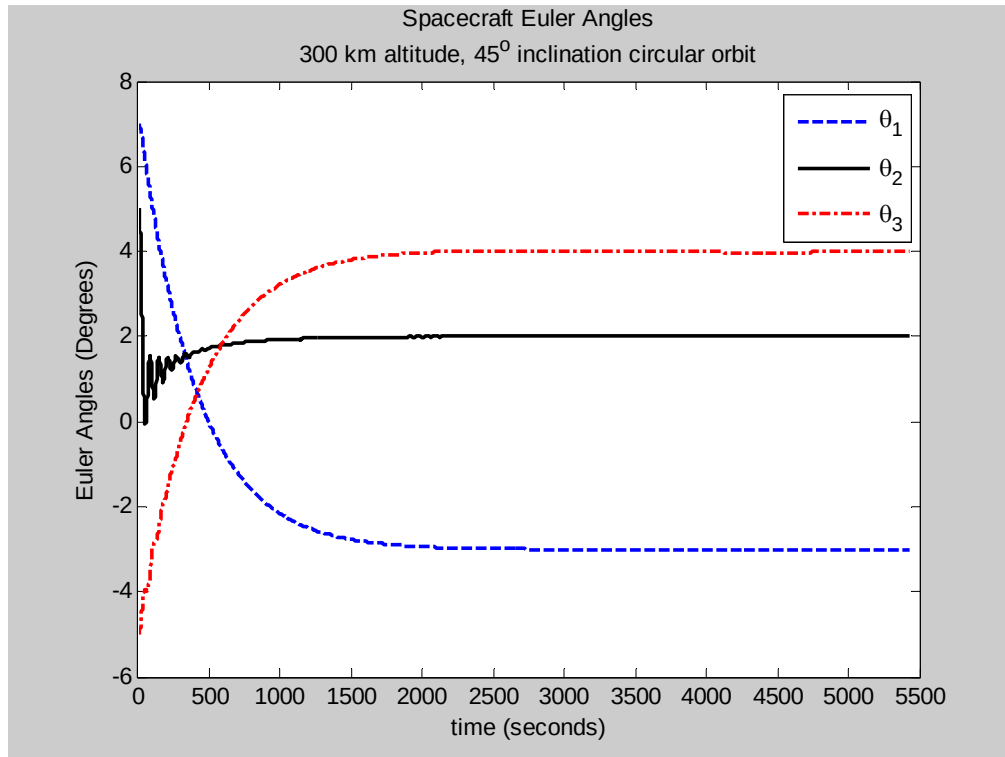


Figure 21: Euler Angles – Case 5 (Controlled)

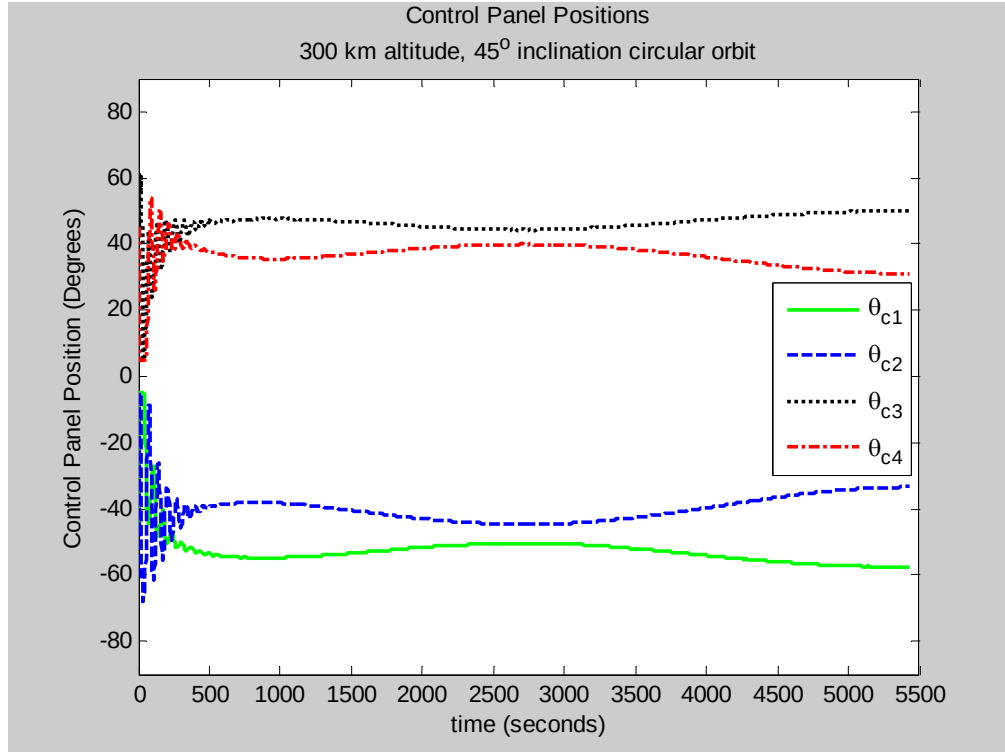


Figure 22: Control Panel Angles – Case 5 (Controlled)

Case 6

This case explores the ability of the spacecraft in an equatorial orbit to recover from a slight tumble and achieve a commanded final offset. The parameters selected for this case are:

$$\begin{aligned}\theta_{1,\text{initial}} &= 7^\circ \\ \theta_{2,\text{initial}} &= 5^\circ & K &= 0.001 \\ \theta_{3,\text{initial}} &= -5^\circ & C &= 0.2 \\ \theta_{1,\text{final}} &= -3^\circ & k_f &= 0.02 \\ \theta_{2,\text{final}} &= 2^\circ & k_p &= 0.02 \\ \theta_{3,\text{final}} &= 4^\circ & i &= 0^\circ \\ \omega_1 = \omega_2 = \omega_3 &= 0.1 \text{ deg/sec}\end{aligned}$$

This case achieves the desired final orientation in approximately 3000 seconds as shown in Figure 23, but it does so with significant control panel activity and rotation about the roll axis. The spacecraft rotation rates are relatively small throughout this maneuver as shown in Figure 24. Not much can be gleaned from the control panel angle plot when viewing the full period of data, therefore, Figure 25 shows the control panel angle history only during the time of tumble recovery. The control panels are driven to/near saturation for much of the time of tumble recovery as the controller brings down the angular rates. Once the angular rates are decreased, the control panels perform much like Cases 1-5 and tend toward their default $\pm 45^\circ$ orientation after reaching the desired final orientation.

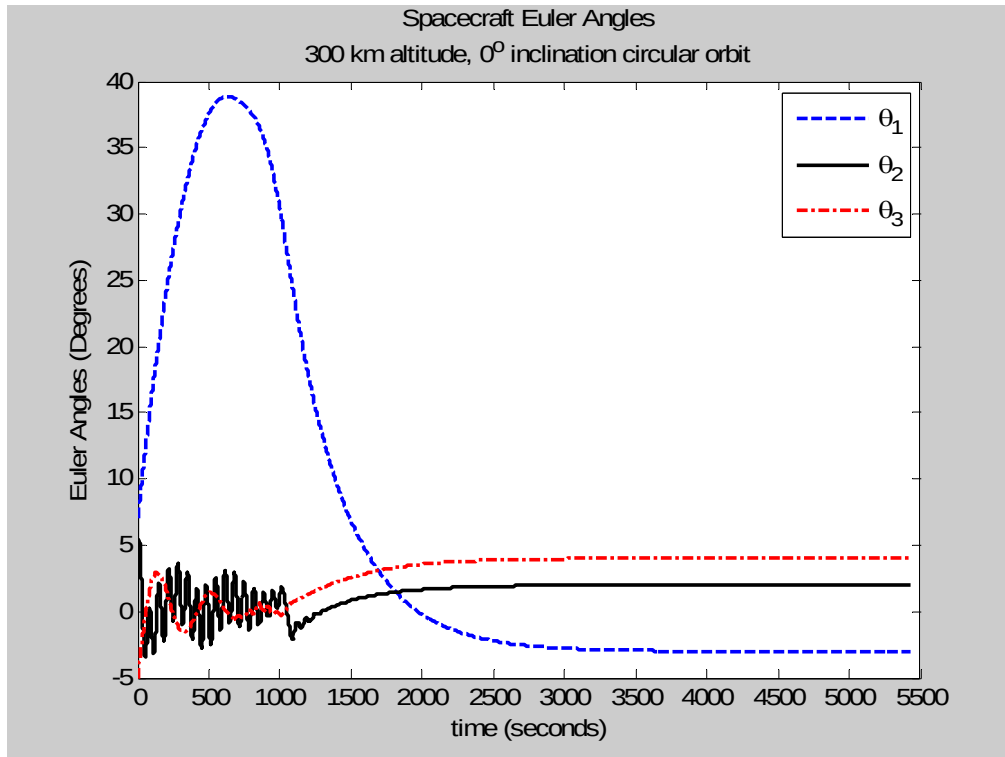


Figure 23: Euler Angles – Case 6 (Controlled)

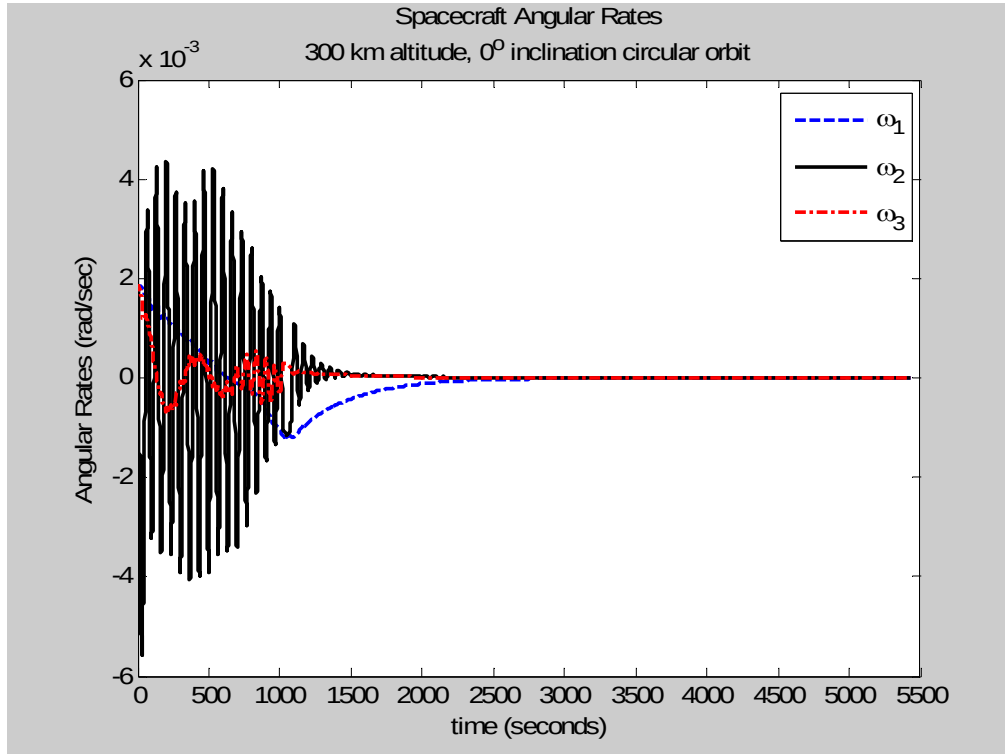


Figure 24: Angular Rates – Case 6 (Controlled)

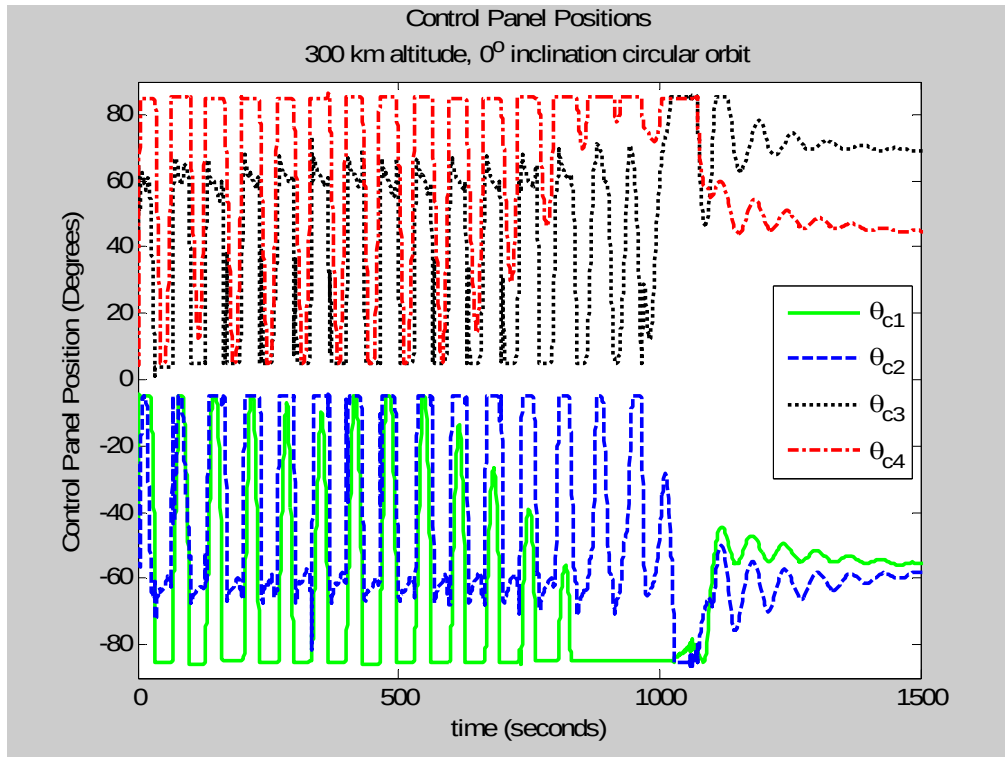


Figure 25: Control Panel Angles – Case 6 (Controlled)

V. Conclusions and Recommendations

Conclusions

This paper introduced a method of three-axis attitude control using only aerodynamic torques. Attitude actuation is achieved using four control panels mounted on the rear of a cubical spacecraft bus. The controller consisted of an outer loop using linear state feedback to determine desired control torque and an inner loop to choose appropriate drag panel angles. The inner loop used a Jacobian-based approach to invert the nonlinear relationship between panel angles and generated torque. Controller performance was evaluated via simulations, which showed that three-axis control is possible over a range of initial angles and angular rates. Uncontrolled performance was explored and the spacecraft designed showed passive stability and damping about the pitch and yaw axes. The analysis used partial accommodation theory as the basis for aerodynamic torque calculations and assumed a rotating atmosphere with an exponential density profile.

Recommendations for Future Research

The spacecraft designed in this thesis has been shown capable of recovering from a slight tumble, but more severe tumbles are certainly realistic and deserved further exploration. Also, a number of simplifying assumptions were made during the development of this analysis that warrant mention for further study. These are especially important if the concept of using aerodynamic torques for attitude control extends beyond the math problem (as it is treated in this thesis) to a physical attitude control system on a spacecraft. Here's a list of topics that could be explored by future researchers. I've

guessed at ranking them in order of importance with 1 being the most important and so on:

1. Expanding controller's ability to recover from a tumble – Case 1 showed passive stability and damping of the yaw and pitch axes in an uncontrolled mode, but the roll axes did not demonstrate these. Also, Case 6 showed the controller's ability to recover from a slight tumble. It's my contention that a combination of uncontrolled and controlled results from these two cases could be merged to expand the spacecraft's tumbling recovery ability. By this I mean, a tumbling spacecraft could be allowed to tumble (uncontrolled) until the angular rates decreased to an acceptable level and the controller could be turned on to reach the desired final orientation. Some attention will have to be paid to the roll axis as it has not demonstrated damping during a prolonged tumble. This researcher attempted this concept with limited success (partly due to the ad hoc method used for the 'turn on' decision and partly due to computers running out of memory two-days into simulations). The greatest tumble rate I have been able to demonstrate recovery from is $\omega_1 = \omega_2 = \omega_3 = 0.2$ deg/sec with all other parameters the same as Case 6.
2. Spacecraft bus shadowing of the control panels – We've assumed in this analysis that the spacecraft bus does not shield the control panels from impacting upper atmosphere particles. This is not really a great assumption and a model that accounts for it would be beneficial to understanding the usefulness of using aerodynamic torques for attitude control.

3. Rate limiting control panel movements – The analysis placed no limits on the rate of control panel movement and the MATLAB solver took advantage of this by moving the control panels quickly to bring down angular rates while recovering from a tumble. Hardware will have limits and a model that incorporates this fact will be useful.
4. Systematic method of determining controller gains – The gains used in this thesis were, quite honestly, determined by guessing and checking. Formal methods for gain determination exist and would be a useful addition to this analysis especially as orbits are extended beyond the 300 km altitude orbit with an exponential density, rotating Earth atmosphere.
5. Use of realistic atmospheric density values – The atmospheric density model used in this analysis was a simple, exponential distribution and provided good ballpark numbers to demonstrate the concept of using aerodynamic torques for attitude control. The Earth's atmosphere's density does vary significantly during orbit (e.g. day/night variation) and a number of good models exist to model this. The controller used in this thesis relies on a good estimate of the atmospheric density to make control panel orientation decisions, therefore, its performance under more realistic density conditions warrants further study.
6. Decaying orbit – This analysis assumed a non-decaying orbit, but the very fact that drag is being used for attitude control will cause the orbit to decay. The impact of this fact was assumed small in this analysis needs to be understood should the concept be made operational.

7. Use of realistic moment of inertia – The spacecraft bus' center-of-mass was assumed to be placed perfectly at its geometric center and the control panels were assumed massless yet stiff enough to generate the control torques. Obviously, a real spacecraft will likely have some mass imbalance and control panels will have mass that should be accounted for with a realistic moment of inertia for use in Euler's Equation.
8. Expand beyond partial accommodation theory – It was assumed that the random thermal motion of the upper atmosphere particles was negligible in this analysis, that V_b is 5 % of V_R , and that $\sigma_n = \sigma_t = 0.8$. The impact of making these assumptions is unknown but models exist for each to better reflect reality. Extending the analysis to include this would be an interesting project to find if the design is sensitive to any of these variables.

Appendix A – MATLAB Simulation Code

```
%          Capt M. LUKE GARGASZ
%    OPTIMAL SPACECRAFT ATTITUDE CONTROL USING AERODYNAMIC TORQUES
%          MARCH 2007
%    MASTER'S THESIS: AFIT/GA/ENY/07-M08
%
%    Contact email: luke_gargasz@hotmail.com

function[out]=Thesis();      close all;clear all;clc;

global Case TrueAnomalyInitial Inclination OrbitalRadius

%% CHOOSE CASE TO RUN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Case = 3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Case 1 is an extreme tumbling/initial offset to explore the uncommanded performance of the spacecraft
% CAUTION: TAKES A LONG TIME TO RUN
% Case 2 is a non-tumbling spacecraft in equatorial orbit without saturation avoidance commanded to
equilibrium
% Case 3 is a non-tumbling spacecraft in equatorial orbit with saturation avoidance commanded to
equilibrium
% Case 4 is a non-tumbling spacecraft in equatorial orbit with saturation avoidance commanded to final
offset
% Case 5 is a non-tumbling spacecraft in inclined orbit with saturation avoidance commanded to final
offset
% Case 6 is a slightly tumbling spacecraft in equatorial orbit with saturation avoidance commanded to final
offset
% CAUTION: TAKES A LONG TIME TO RUN

% Orbit Propagation begin coincident with I vector
TrueAnomalyInitial = deg2rad(0);

% PARAMETERS ASSIGNED BASED ON THE CASE CHOSEN ABOVE
if Case == 1;
    % Orbital Inclination
    Inclination = deg2rad(0);
    % Initial Euler Angles
    Theta1 = deg2rad(-13); Theta2 = deg2rad(101); Theta3 = deg2rad(-26);
    % Final Euler Angles
    Theta1C = deg2rad(0); Theta2C = deg2rad(0); Theta3C = deg2rad(0);
    % Initial Rotation Rates
    Omega1 = deg2rad(-3); Omega2 = deg2rad(-2); Omega3 = deg2rad(2.5);
    % Orbital Altitude in kilometers
    OrbitalAltitude = 300;
    % Number of Orbital Periods to Propagate
    NumberOfPeriods = 1000;
elseif Case == 2 || Case == 3;
    % Orbital Inclination
    Inclination = deg2rad(0);
    % Initial Euler Angles
    Theta1 = deg2rad(7); Theta2 = deg2rad(5); Theta3 = deg2rad(-5);
```

```

% Final Euler Angles
Theta1C = deg2rad(0); Theta2C = deg2rad(0); Theta3C = deg2rad(0);
% Initial Rotation Rates
Omega1 = deg2rad(0); Omega2 = deg2rad(0); Omega3 = deg2rad(0);
% Orbital Altitude in kilometers
OrbitalAltitude = 300;
% Number of Orbital Periods to Propagate
NumberOfPeriods = 1;
elseif Case == 4
% Orbital Inclination
Inclination = deg2rad(0);
% Initial Euler Angles
Theta1 = deg2rad(7); Theta2 = deg2rad(5); Theta3 = deg2rad(-5);
% Final Euler Angles
Theta1C = deg2rad(-3); Theta2C = deg2rad(2); Theta3C = deg2rad(4);
% Initial Rotation Rates
Omega1 = deg2rad(0); Omega2 = deg2rad(0); Omega3 = deg2rad(0);
% Orbital Altitude in kilometers
OrbitalAltitude = 300;
% Number of Orbital Periods to Propagate
NumberOfPeriods = 1;
elseif Case == 5
% Orbital Inclination
Inclination = deg2rad(45);
% Initial Euler Angles
Theta1 = deg2rad(7); Theta2 = deg2rad(5); Theta3 = deg2rad(-5);
% Final Euler Angles
Theta1C = deg2rad(-3); Theta2C = deg2rad(2); Theta3C = deg2rad(4);
% Initial Rotation Rates
Omega1 = deg2rad(0); Omega2 = deg2rad(0); Omega3 = deg2rad(0);
% Orbital Altitude in kilometers
OrbitalAltitude = 300;
% Number of Orbital Periods to Propagate
NumberOfPeriods = 1;
elseif Case == 6
% Orbital Inclination
Inclination = deg2rad(0);
% Initial Euler Angles
Theta1 = deg2rad(7); Theta2 = deg2rad(5); Theta3 = deg2rad(-5);
% Final Euler Angles
Theta1C = deg2rad(-3); Theta2C = deg2rad(2); Theta3C = deg2rad(4);
% Initial Rotation Rates
Omega1 = deg2rad(0.1); Omega2 = deg2rad(0.1); Omega3 = deg2rad(0.1);
% Orbital Altitude in kilometers
OrbitalAltitude = 300;
% Number of Orbital Periods to Propagate
NumberOfPeriods = 1;
end

%% Set the initial control panel angles
ThC1 = deg2rad(-45); ThC2 = deg2rad(-45);
ThC3 = deg2rad( 45); ThC4 = deg2rad( 45);

%% Calculate the initial quaternion

```

% ROTATION MATRIX - ORBITAL TO PRINCIPAL AXES - BODY-THREE 1-2-3 ROTATION

```
c1 = cos(Theta1); s1 = sin(Theta1);
c2 = cos(Theta2); s2 = sin(Theta2);
c3 = cos(Theta3); s3 = sin(Theta3);
```

```
R_ORB_PRIN = [ c2*c3  s1*s2*c3+s3*c1  -c1*s2*c3+s3*s1;
               -c2*s3  -s1*s2*s3+c3*c1  c1*s2*s3+c3*s1;
               s2      -s1*c2      c1*c2    ];
```

% Calculate quaternion components

```
q4 = .5*sqrt(1+trace(R_ORB_PRIN));
q1 = 1/(4*q4)*(R_ORB_PRIN(2,3)-R_ORB_PRIN(3,2));
q2 = 1/(4*q4)*(R_ORB_PRIN(3,1)-R_ORB_PRIN(1,3));
q3 = 1/(4*q4)*(R_ORB_PRIN(1,2)-R_ORB_PRIN(2,1));
```

%% Calculate the commanded quaternion

% ROTATION MATRIX - ORBITAL TO PRINCIPAL AXES - BODY-THREE 1-2-3 ROTATION

```
c1c = cos(Theta1C); s1c = sin(Theta1C);
c2c = cos(Theta2C); s2c = sin(Theta2C);
c3c = cos(Theta3C); s3c = sin(Theta3C);
```

```
R_ORB_PRIN = [ c2c*c3c  s1c*s2c*c3c+s3c*c1c  -c1c*s2c*c3c+s3c*s1c;
               -c2c*s3c  -s1c*s2c*s3c+c3c*c1c  c1c*s2c*s3c+c3c*s1c;
               s2c      -s1c*c2c      c1c*c2c    ];
```

% Calculate quaternion components

```
q4C = .5*sqrt(1+trace(R_ORB_PRIN));
q1C = 1/(4*q4C)*(R_ORB_PRIN(2,3)-R_ORB_PRIN(3,2));
q2C = 1/(4*q4C)*(R_ORB_PRIN(3,1)-R_ORB_PRIN(1,3));
q3C = 1/(4*q4C)*(R_ORB_PRIN(1,2)-R_ORB_PRIN(2,1));
```

```
qc = [q1C q2C q3C q4C]';
```

%% Calculate the orbital period

```
mu = 398600e9; % m^3/sec^2
```

```
OrbitalRadius = 6378000 + OrbitalAltitude*1000; % meters -- 6378 km earth radius + orbit altitude in km
```

```
T = 2*pi*sqrt(OrbitalRadius^3/mu); % Only good for circular orbits
```

%% ODE45 Call

```
t_init = 0; t_final = NumberOfPeriods*T;
```

```
x0 = [Omega1 Omega2 Omega3 q1 q2 q3 q4 ThC1 ThC2 ThC3 ThC4]';
```

```
[t,x]=ode45(@eom,[t_init t_final],x0,[],qc);
```

%% Plot Results

```
w = x(:,1:3); q = x(:,4:7); ThC = x(:,8:11);
```

% Back Euler Angles out of Quaternions

```
for n = 1:length(t)
```

```

R_ORB_PRIN = [ 1-2*(q(n,2)^2+q(n,3)^2) 2*(q(n,1)*q(n,2)+q(n,3)*q(n,4)) 2*(q(n,1)*q(n,3)-
q(n,2)*q(n,4));
2*(q(n,2)*q(n,1)-q(n,3)*q(n,4)) 1-2*(q(n,1)^2+q(n,3)^2)
2*(q(n,2)*q(n,3)+q(n,1)*q(n,4));
2*(q(n,3)*q(n,1)+q(n,2)*q(n,4)) 2*(q(n,3)*q(n,2)-q(n,1)*q(n,4)) 1-
2*(q(n,1)^2+q(n,2)^2)];
theta2 = asin(R_ORB_PRIN(3,1));
theta1 = asin(-R_ORB_PRIN(3,2)/cos(theta2));
theta3 = asin(-R_ORB_PRIN(2,1)/cos(theta2));
theta(n,1:3) = [theta1 theta2 theta3];
n = n+1;
end

```

% Select the plots of interest for the Case being analyzed

```

if Case == 1
    plot(t,w(:,1),'b-','LineWidth',0.5);
    title({'Spacecraft Angular Rates';[' ',num2str((OrbitalRadius-6378000)/1000), ' km altitude,
',num2str(rad2deg(Inclination)), '^o inclination circular orbit']});
    legend('\omega_1'); xlabel('time (seconds)'); ylabel('Angular Rates (rad/sec)'); figure;
    plot(t,w(:,2),'k-','LineWidth',0.5);
    title({'Spacecraft Angular Rates';[' ',num2str((OrbitalRadius-6378000)/1000), ' km altitude,
',num2str(rad2deg(Inclination)), '^o inclination circular orbit']});
    legend('\omega_2'); xlabel('time (seconds)'); ylabel('Angular Rates (rad/sec)'); figure;
    plot(t,w(:,3),'r-','LineWidth',0.5);
    title({'Spacecraft Angular Rates';[' ',num2str((OrbitalRadius-6378000)/1000), ' km altitude,
',num2str(rad2deg(Inclination)), '^o inclination circular orbit']});
    legend('\omega_3'); xlabel('time (seconds)'); ylabel('Angular Rates (rad/sec)');
elseif Case == 2 || Case == 3 || Case == 4 || Case == 5 || Case == 6
    plot(t,rad2deg(theta(:,1)),'b- ',t,rad2deg(theta(:,2)),'k- ',t,rad2deg(theta(:,3)),'r- ','LineWidth',1.75);
    title({'Spacecraft Euler Angles';[' ',num2str((OrbitalRadius-6378000)/1000), ' km altitude,
',num2str(rad2deg(Inclination)), '^o inclination circular orbit']});
    legend('\theta_1','\theta_2','\theta_3');
    xlabel('time (seconds)'); ylabel('Euler Angles (Degrees)'); xlim([0 5500]); figure;
    plot(t,w(:,1),'b- ',t,w(:,2),'k- ',t,w(:,3),'r- ','LineWidth',1.75);
    title({'Spacecraft Angular Rates';[' ',num2str((OrbitalRadius-6378000)/1000), ' km altitude,
',num2str(rad2deg(Inclination)), '^o inclination circular orbit']});
    legend('\omega_1','\omega_2','\omega_3');
    xlabel('time (seconds)'); ylabel('Angular Rates (rad/sec)'); xlim([0 5500]); figure;
    plot(t,rad2deg(ThC(:,1)),'g- ',t,rad2deg(ThC(:,2)),'b- ',t,rad2deg(ThC(:,3)),'k- ',t,rad2deg(ThC(:,4)),'r-
','LineWidth',1.75);
    title({'Control Panel Positions';[' ',num2str((OrbitalRadius-6378000)/1000), ' km altitude,
',num2str(rad2deg(Inclination)), '^o inclination circular orbit']});
    legend('\theta_c_1','\theta_c_2','\theta_c_3','\theta_c_4','Location','East');
    xlabel('time (seconds)'); ylabel('Control Panel Position (Degrees)'); xlim([0 5500]); ylim([-90 90]);
end

```

out=[t,rad2deg(theta),w,q,rad2deg(ThC)]; % DATA FOR OUTPUT TO THE WORKSPACE

%% Equations of Motion Function

```
function [xdot] = eom(t,x,qc)
```

global Case TrueAnomalyInitial Inclination OrbitalRadius

% Unpack variables

```
w = [x(1); x(2); x(3)];
```

```

q1 = x(4); q2 = x(5); q3 = x(6); q4 = x(7); q = [q1; q2; q3];
ThC1 = x(8); ThC2 = x(9); ThC3 = x(10); ThC4 = x(11);

```

```

%% CALCULATE ATMOSPHERIC DENSITY

```

```

% 300 km orbit values taken from Vallado % this needs to be changed to a lookup table so the orbital
altitude can be varied

```

```

rho_not = 2.418e-11;
h = (OrbitalRadius-6378000)/1000;
h_not = 300;
H = 53.628;

```

```

rho = rho_not*exp((h-h_not)/H);

```

```

%% ROTATION MATRIX - ORBITAL TO PRINCIPAL AXES - BODY-THREE 1-2-3 ROTATION

```

```

R_ORB_PRIN = [1-2*(q2^2+q3^2) 2*(q1*q2+q3*q4) 2*(q1*q3-q2*q4);
               2*(q2*q1-q3*q4) 1-2*(q1^2+q3^2) 2*(q2*q3+q1*q4);
               2*(q3*q1+q2*q4) 2*(q3*q2-q1*q4) 1-2*(q1^2+q2^2)];

```

```

%% Calculate the local atmospheric velocity vector

```

```

% Constants

```

```

we = 7.27E-5; % radians/sec
mu = 398600e9; % m^3/sec^2

```

```

% Calculate the Orbital Velocity

```

```

OrbitalVelocity = sqrt(mu/OrbitalRadius); % m/s % only good for circular orbits

```

```

%Calculate the true anomaly

```

```

mean_motion = sqrt(mu/OrbitalRadius^3);
nu = mean_motion*t+TrueAnomalyInitial;

```

```

R = OrbitalRadius; V = OrbitalVelocity;

```

```

VR = V*(1-(we*R/V)*cos(Inclination))*R_ORB_PRIN*[-1; (we*R/V)*sin(Inclination)*cos(nu); 0];

```

```

VRunit = VR/norm(VR);

```

```

Vb = 0.05*VR; % This is just an estimate and can be expanded

```

```

%% Spacecraft Measurements

```

```

% Spacecraft Bus Measurements

```

```

d = .25; % distance from Center of Mass to the edge of the spacecraft bus
Acube = 2*d*2*d; % Area of one side of the cube

```

```

% Control Panel Measurements

```

```

L = .2; % half the length of the control panels
f = .25/2; % distance from the center of mass to the center of the control panels
width = .25; % width of the control panels
Apanel = width*2*L; % Area of one control panel

```

```

%% Spacecraft Bus Geometry

```

```

% Front Side (+ x-axis side)
r1 = [d;0;0]; n1 = [-1;0;0];
alpha1 = acos(dot(VRunit,n1));

% Left Side (+ y-axis side)
r2 = [0;d;0]; n2 = [0;-1;0];
alpha2 = acos(dot(VRunit,n2));

% Right Side
r3 = [0;-d;0]; n3 = [0;1;0];
alpha3 = acos(dot(VRunit,n3));

% Top Side
r4 = [0;0;-d]; n4 = [0;0;1];
alpha4 = acos(dot(VRunit,n4));

% Bottom Side (+ z-axis side)
r5 = [0;0;d]; n5 = [0;0;-1];
alpha5 = acos(dot(VRunit,n5));

% Back Side
r6 = [-d;0;0]; n6 = [1;0;0];
alpha6 = acos(dot(VRunit,n6));

%% Control Panel Geometry

% Control Panel 7 -- Top / positive y-axis side
r7 = [-(d+L*cos(ThC1)); f; -d+L*sin(ThC1)];
n7f = [sin(ThC1); 0; cos(ThC1)];
n7r = -n7f;
alpha7f = acos(dot(VRunit,n7f));
alpha7r = acos(dot(VRunit,n7r));

% Control Panel 8 -- Top / negative y-axis side
r8 = [-(d+L*cos(ThC2)); -f; -d+L*sin(ThC2)];
n8f = [sin(ThC2); 0; cos(ThC2)];
n8r = -n8f;
alpha8f = acos(dot(VRunit,n8f));
alpha8r = acos(dot(VRunit,n8r));

% Control Panel 9 -- Bottom / positive y-axis side
r9 = [-(d+L*cos(ThC3)); f; (d+L*sin(ThC3))];
n9f = [-sin(ThC3); 0; -cos(ThC3)];
n9r = -n9f;
alpha9f = acos(dot(VRunit,n9f));
alpha9r = acos(dot(VRunit,n9r));

% Control Panel 10 -- Bottom / negative y-axis side
r10 = [-(d+L*cos(ThC4)); -f; (d+L*sin(ThC4))];
n10f = [-sin(ThC4); 0; -cos(ThC4)];
n10r = -n10f;
alpha10f = acos(dot(VRunit,n10f));
alpha10r = acos(dot(VRunit,n10r));

```

%% Calculate Ap

```
Ap1 = heaviside(cos(alpha1 ))*cos(alpha1 )*Acube;  
Ap2 = heaviside(cos(alpha2 ))*cos(alpha2 )*Acube;  
Ap3 = heaviside(cos(alpha3 ))*cos(alpha3 )*Acube;  
Ap4 = heaviside(cos(alpha4 ))*cos(alpha4 )*Acube;  
Ap5 = heaviside(cos(alpha5 ))*cos(alpha5 )*Acube;  
Ap6 = heaviside(cos(alpha6 ))*cos(alpha6 )*Acube;  
Ap7f = heaviside(cos(alpha7f ))*cos(alpha7f )*Apanel;  
Ap7r = heaviside(cos(alpha7r ))*cos(alpha7r )*Apanel;  
Ap8f = heaviside(cos(alpha8f ))*cos(alpha8f )*Apanel;  
Ap8r = heaviside(cos(alpha8r ))*cos(alpha8r )*Apanel;  
Ap9f = heaviside(cos(alpha9f ))*cos(alpha9f )*Apanel;  
Ap9r = heaviside(cos(alpha9r ))*cos(alpha9r )*Apanel;  
Ap10f = heaviside(cos(alpha10f ))*cos(alpha10f )*Apanel;  
Ap10r = heaviside(cos(alpha10r ))*cos(alpha10r )*Apanel;
```

%% Calculate Cp

% if/else statement necessary to account for Ap being zero and then trying to divide by zero

```
if Ap1 <= 0; Cp1 = [0;0;0];  
else Cp1 = heaviside(cos(alpha1 ))*cos(alpha1 )*Acube*r1/Ap1; end
```

```
if Ap2 <= 0; Cp2 = [0;0;0];  
else Cp2 = heaviside(cos(alpha2 ))*cos(alpha2 )*Acube*r2/Ap2; end
```

```
if Ap3 <= 0; Cp3 = [0;0;0];  
else Cp3 = heaviside(cos(alpha3 ))*cos(alpha3 )*Acube*r3/Ap3; end
```

```
if Ap4 <= 0; Cp4 = [0;0;0];  
else Cp4 = heaviside(cos(alpha4 ))*cos(alpha4 )*Acube*r4/Ap4; end
```

```
if Ap5 <= 0; Cp5 = [0;0;0];  
else Cp5 = heaviside(cos(alpha5 ))*cos(alpha5 )*Acube*r5/Ap5; end
```

```
if Ap6 <= 0; Cp6 = [0;0;0];  
else Cp6 = heaviside(cos(alpha6 ))*cos(alpha6 )*Acube*r6/Ap6; end
```

```
if Ap7f <= 0; Cp7f = [0;0;0];  
else Cp7f = heaviside(cos(alpha7f ))*cos(alpha7f )*Apanel*r7/Ap7f; end  
if Ap7r <= 0; Cp7r = [0;0;0];  
else Cp7r = heaviside(cos(alpha7r ))*cos(alpha7r )*Apanel*r7/Ap7r; end
```

```
if Ap8f <= 0; Cp8f = [0;0;0];  
else Cp8f = heaviside(cos(alpha8f ))*cos(alpha8f )*Apanel*r8/Ap8f; end  
if Ap8r <= 0; Cp8r = [0;0;0];  
else Cp8r = heaviside(cos(alpha8r ))*cos(alpha8r )*Apanel*r8/Ap8r; end
```

```
if Ap9f <= 0; Cp9f = [0;0;0];  
else Cp9f = heaviside(cos(alpha9f ))*cos(alpha9f )*Apanel*r9/Ap9f; end  
if Ap9r <= 0; Cp9r = [0;0;0];  
else Cp9r = heaviside(cos(alpha9r ))*cos(alpha9r )*Apanel*r9/Ap9r; end
```

```

if Ap10f <= 0; Cp10f = [0;0;0];
else Cp10f = heaviside(cos(alpha10f))*cos(alpha10f)*Apanel*r10/Ap10f; end
if Ap10r <= 0; Cp10r = [0;0;0];
else Cp10r = heaviside(cos(alpha10r))*cos(alpha10r)*Apanel*r10/Ap10r; end

```

%% Calculate Gp

```

Gp1 = heaviside(cos(alpha1 ))*cos(alpha1 )*Acube* cross(r1, n1);
Gp2 = heaviside(cos(alpha2 ))*cos(alpha2 )*Acube* cross(r2, n2);
Gp3 = heaviside(cos(alpha3 ))*cos(alpha3 )*Acube* cross(r3, n3);
Gp4 = heaviside(cos(alpha4 ))*cos(alpha4 )*Acube* cross(r4, n4);
Gp5 = heaviside(cos(alpha5 ))*cos(alpha5 )*Acube* cross(r5, n5);
Gp6 = heaviside(cos(alpha6 ))*cos(alpha6 )*Acube* cross(r6, n6);
Gp7f = heaviside(cos(alpha7f))*cos(alpha7f)*Apanel*cross(r7, n7f);
Gp7r = heaviside(cos(alpha7r))*cos(alpha7r)*Apanel*cross(r7, n7r);
Gp8f = heaviside(cos(alpha8f))*cos(alpha8f)*Apanel*cross(r8, n8f);
Gp8r = heaviside(cos(alpha8r))*cos(alpha8r)*Apanel*cross(r8, n8r);
Gp9f = heaviside(cos(alpha9f))*cos(alpha9f)*Apanel*cross(r9, n9f);
Gp9r = heaviside(cos(alpha9r))*cos(alpha9r)*Apanel*cross(r9, n9r);
Gp10f = heaviside(cos(alpha10f))*cos(alpha10f)*Apanel*cross(r10,n10f);
Gp10r = heaviside(cos(alpha10r))*cos(alpha10r)*Apanel*cross(r10,n10r);

```

%% Calculate Gpp

```

Gpp1 = heaviside(cos(alpha1 ))*cos(alpha1 )^2*Acube* cross(r1, n1);
Gpp2 = heaviside(cos(alpha2 ))*cos(alpha2 )^2*Acube* cross(r2, n2);
Gpp3 = heaviside(cos(alpha3 ))*cos(alpha3 )^2*Acube* cross(r3, n3);
Gpp4 = heaviside(cos(alpha4 ))*cos(alpha4 )^2*Acube* cross(r4, n4);
Gpp5 = heaviside(cos(alpha5 ))*cos(alpha5 )^2*Acube* cross(r5, n5);
Gpp6 = heaviside(cos(alpha6 ))*cos(alpha6 )^2*Acube* cross(r6, n6);
Gpp7f = heaviside(cos(alpha7f))*cos(alpha7f)^2*Apanel*cross(r7, n7f);
Gpp7r = heaviside(cos(alpha7r))*cos(alpha7r)^2*Apanel*cross(r7, n7r);
Gpp8f = heaviside(cos(alpha8f))*cos(alpha8f)^2*Apanel*cross(r8, n8f);
Gpp8r = heaviside(cos(alpha8r))*cos(alpha8r)^2*Apanel*cross(r8, n8r);
Gpp9f = heaviside(cos(alpha9f))*cos(alpha9f)^2*Apanel*cross(r9, n9f);
Gpp9r = heaviside(cos(alpha9r))*cos(alpha9r)^2*Apanel*cross(r9, n9r);
Gpp10f = heaviside(cos(alpha10f))*cos(alpha10f)^2*Apanel*cross(r10,n10f);
Gpp10r = heaviside(cos(alpha10r))*cos(alpha10r)^2*Apanel*cross(r10,n10r);

```

%% Calculate the disturbance torque and control torque

% ACCOMODATION COEFFICIENTS - estimates

```
sig_t = 0.8; sig_n = 0.8;
```

```

gd1 = rho*norm(VR)^2*(sig_t*Ap1* cross(Cp1,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp1 +(2-
sig_n-sig_t)*Gpp1);
gd2 = rho*norm(VR)^2*(sig_t*Ap2* cross(Cp2,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp2 +(2-
sig_n-sig_t)*Gpp2);
gd3 = rho*norm(VR)^2*(sig_t*Ap3* cross(Cp3,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp3 +(2-
sig_n-sig_t)*Gpp3);
gd4 = rho*norm(VR)^2*(sig_t*Ap4* cross(Cp4,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp4 +(2-
sig_n-sig_t)*Gpp4);
gd5 = rho*norm(VR)^2*(sig_t*Ap5* cross(Cp5,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp5 +(2-
sig_n-sig_t)*Gpp5);
gd6 = rho*norm(VR)^2*(sig_t*Ap6* cross(Cp6,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp6 +(2-
sig_n-sig_t)*Gpp6);

```



```

gc7f = rho*norm(VR)^2*(sig_t*Ap7f* cross(Cp7f,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp7f +(2-
sig_n-sig_t)*Gpp7f);
gc7r = rho*norm(VR)^2*(sig_t*Ap7r* cross(Cp7r,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp7r +(2-
sig_n-sig_t)*Gpp7r);
gc8f = rho*norm(VR)^2*(sig_t*Ap8f* cross(Cp8f,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp8f +(2-
sig_n-sig_t)*Gpp8f);
gc8r = rho*norm(VR)^2*(sig_t*Ap8r* cross(Cp8r,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp8r +(2-
sig_n-sig_t)*Gpp8r);
gc9f = rho*norm(VR)^2*(sig_t*Ap9f* cross(Cp9f,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp9f +(2-
sig_n-sig_t)*Gpp9f);
gc9r = rho*norm(VR)^2*(sig_t*Ap9r* cross(Cp9r,VRunit) +sig_n*(norm(Vb)/norm(VR))*Gp9r +(2-
sig_n-sig_t)*Gpp9r);
gc10f = rho*norm(VR)^2*(sig_t*Ap10f*cross(Cp10f,VRunit)+sig_n*(norm(Vb)/norm(VR))*Gp10f+(2-
sig_n-sig_t)*Gpp10f);
gc10r = rho*norm(VR)^2*(sig_t*Ap10r*cross(Cp10r,VRunit)+sig_n*(norm(Vb)/norm(VR))*Gp10r+(2-
sig_n-sig_t)*Gpp10r);

```

```

% Disturbance torque on the spacecraft bus

```

```

gd = gd1 + gd2 + gd3 + gd4 + gd5 + gd6;

```

```

% Control torque generated by the control panels

```

```

gc = gc7f + gc7r + gc8f + gc8r + gc9f + gc9r + gc10f + gc10r;

```

```

%% Select Gains for Case Being Analyzed

```

```

if Case == 1 % Uncontrolled Case

```

```

    K = 0;

```

```

    C = 0;

```

```

    kf = 0;

```

```

    kp = 0;

```

```

elseif Case == 2 % Controlled Case without Saturation Avoidance

```

```

    K = 0.001*eye(3);

```

```

    C = 0.2*eye(3);

```

```

    kf = 0.02;

```

```

    kp = 0.0;

```

```

elseif Case == 3 || Case == 4 || Case == 5 || Case == 6 % Controlled Cases with Saturation Avoidance

```

```

    K = 0.001*eye(3);

```

```

    C = 0.2*eye(3);

```

```

    kf = 0.02;

```

```

    kp = 0.02;

```

```

end

```

```

%% OUTER LOOP

```

```

% Define Attitude Error Quaternion

```

```

qe4 = [ qc(4) qc(3) -qc(2) -qc(1);...
        -qc(3) qc(4) qc(1) -qc(2);...
        qc(2) -qc(1) qc(4) -qc(3);...
        qc(1) qc(2) qc(3) qc(4)]*[q; q4];

```

```

qe3 = [qe4(1) qe4(2) qe4(3)]';

```

```

% OUTER LOOP CONTROL LAW - Quaternion Feedback Reorientation

```

```

u = -K*qe3-C*w;

```

```

%% INNER LOOP

```

% These definitions are necessary due to the symbolic Jacobian taken in derivatives.m

```
normVR = norm(VR);
normVb = norm(Vb);
VR1 = VR(1); VR2 = VR(2); VR3 = VR(3);
```

% Linearized control torque (gc) with respect to ThC1, ThC2, ThC3 & ThC4

```
A = [
rho*normVR^2*(sig_t*dirac(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/
/normVR)*cos(ThC1)-
conj(VR3/normVR)*sin(ThC1))*(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*Apanel
el*(f*VR3/normVR-(-
d+L*sin(ThC1))*VR2/normVR)+sig_t*heaviside(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*co
s(ThC1))*(conj(VR1/normVR)*cos(ThC1)-conj(VR3/normVR)*sin(ThC1))*Apanel*(f*VR3/normVR-(-
d+L*sin(ThC1))*VR2/normVR)-
sig_t*heaviside(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/normVR)*si
n(ThC1)+conj(VR3/normVR)*cos(ThC1))*Apanel*L*cos(ThC1)*VR2/normVR+sig_n*normVb/normVR
*dirac(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/normVR)*cos(ThC1)
-
conj(VR3/normVR)*sin(ThC1))*(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*Apanel
*f*cos(ThC1)+sig_n*normVb/normVR*heaviside(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*
cos(ThC1))*(conj(VR1/normVR)*cos(ThC1)-conj(VR3/normVR)*sin(ThC1))*Apanel*f*cos(ThC1)-
sig_n*normVb/normVR*heaviside(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(con
j(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*Apanel*f*sin(ThC1)+(2-sig_n-
sig_t)*dirac(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/normVR)*cos(Th
C1)-
conj(VR3/normVR)*sin(ThC1))*(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))^2*Ap
anel*f*cos(ThC1)+2*(2-sig_n-
sig_t)*heaviside(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/normVR)*s
in(ThC1)+conj(VR3/normVR)*cos(ThC1))*Apanel*f*cos(ThC1)*(conj(VR1/normVR)*cos(ThC1)-
conj(VR3/normVR)*sin(ThC1))-(2-sig_n-
sig_t)*heaviside(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/normVR)*s
in(ThC1)+conj(VR3/normVR)*cos(ThC1))^2*Apanel*f*sin(ThC1))+rho*normVR^2*(sig_t*dirac(conj(V
R1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(-
conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))*(-conj(VR1/normVR)*sin(ThC1)-
conj(VR3/normVR)*cos(ThC1))*Apanel*(f*VR3/normVR-(-
d+L*sin(ThC1))*VR2/normVR)+sig_t*heaviside(-conj(VR1/normVR)*sin(ThC1)-
conj(VR3/normVR)*cos(ThC1))*(-
conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))*Apanel*(f*VR3/normVR-(-
d+L*sin(ThC1))*VR2/normVR)-sig_t*heaviside(-conj(VR1/normVR)*sin(ThC1)-
conj(VR3/normVR)*cos(ThC1))*(-conj(VR1/normVR)*sin(ThC1)-
conj(VR3/normVR)*cos(ThC1))*Apanel*L*cos(ThC1)*VR2/normVR-
sig_n*normVb/normVR*dirac(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(-
conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))*(-conj(VR1/normVR)*sin(ThC1)-
conj(VR3/normVR)*cos(ThC1))*Apanel*f*cos(ThC1)-sig_n*normVb/normVR*heaviside(-
conj(VR1/normVR)*sin(ThC1)-conj(VR3/normVR)*cos(ThC1))*(-
conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))*Apanel*f*cos(ThC1)+sig_n*normVb/n
ormVR*heaviside(-conj(VR1/normVR)*sin(ThC1)-conj(VR3/normVR)*cos(ThC1))*(-
conj(VR1/normVR)*sin(ThC1)-conj(VR3/normVR)*cos(ThC1))*Apanel*f*sin(ThC1)-(2-sig_n-
sig_t)*dirac(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(-
conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))*(-conj(VR1/normVR)*sin(ThC1)-
conj(VR3/normVR)*cos(ThC1))^2*Apanel*f*cos(ThC1)-2*(2-sig_n-sig_t)*heaviside(-
conj(VR1/normVR)*sin(ThC1)-conj(VR3/normVR)*cos(ThC1))*(-conj(VR1/normVR)*sin(ThC1)-
conj(VR3/normVR)*cos(ThC1))*Apanel*f*cos(ThC1))*(-
conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))+2*(2-sig_n-sig_t)*heaviside(-
conj(VR1/normVR)*sin(ThC1)-conj(VR3/normVR)*cos(ThC1))*(-conj(VR1/normVR)*sin(ThC1)-
conj(VR3/normVR)*cos(ThC1))^2*Apanel*f*sin(ThC1)),
```


[illegible]

[illegible]

$$\begin{aligned}
& (VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*Apanel*L*sin(ThC1)*VR2/normVR- \\
& sig_n*normVb/normVR*dirac(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/normVR)*cos(ThC1)- \\
& conj(VR3/normVR)*sin(ThC1))*(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*Apanel*f*sin(ThC1)- \\
& sig_n*normVb/normVR*heaviside(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/normVR)*cos(ThC1)- \\
& conj(VR3/normVR)*sin(ThC1))*Apanel*f*sin(ThC1)- \\
& sig_n*normVb/normVR*heaviside(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/normVR)*sin(ThC1)+ \\
& conj(VR3/normVR)*cos(ThC1))*Apanel*f*cos(ThC1)-(2-sig_n- \\
& sig_t)*dirac(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/normVR)*cos(ThC1)- \\
& conj(VR3/normVR)*sin(ThC1))* \\
& conj(VR3/normVR)*sin(ThC1))*(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))^2*Apanel*f*sin(ThC1)-2*(2-sig_n- \\
& sig_t)*heaviside(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/normVR)*sin(ThC1)+ \\
& conj(VR3/normVR)*cos(ThC1))*Apanel*f*sin(ThC1)*(conj(VR1/normVR)*cos(ThC1)- \\
& conj(VR3/normVR)*sin(ThC1))-(2-sig_n- \\
& sig_t)*heaviside(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))*(conj(VR1/normVR)*sin(ThC1)+ \\
& conj(VR3/normVR)*cos(ThC1))^2*Apanel*f*cos(ThC1)+rho*normVR^2*(sig_t*dirac(conj(VR1/normVR)*sin(ThC1)+ \\
& conj(VR3/normVR)*cos(ThC1))* \\
& conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))*(-conj(VR1/normVR)*sin(ThC1)- \\
& conj(VR3/normVR)*cos(ThC1))*Apanel*((-d-L*cos(ThC1))*VR2/normVR- \\
& f*VR1/normVR)+sig_t*heaviside(-conj(VR1/normVR)*sin(ThC1)-conj(VR3/normVR)*cos(ThC1))*(- \\
& conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))*Apanel*((-d- \\
& L*cos(ThC1))*VR2/normVR-f*VR1/normVR)+sig_t*heaviside(-conj(VR1/normVR)*sin(ThC1)- \\
& conj(VR3/normVR)*cos(ThC1))*(-conj(VR1/normVR)*sin(ThC1)- \\
& conj(VR3/normVR)*cos(ThC1))*Apanel*L*sin(ThC1)*VR2/normVR+sig_n*normVb/normVR*dirac(conj(VR1/normVR)*sin(ThC1)+ \\
& conj(VR3/normVR)*cos(ThC1))* \\
& (-conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))*(-conj(VR1/normVR)*sin(ThC1)- \\
& conj(VR3/normVR)*cos(ThC1))*Apanel*f*sin(ThC1)+sig_n*normVb/normVR*heaviside(- \\
& conj(VR1/normVR)*sin(ThC1)-conj(VR3/normVR)*cos(ThC1))* \\
& (-conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))*Apanel*f*sin(ThC1)+sig_n*normVb/normVR*heaviside(- \\
& conj(VR1/normVR)*sin(ThC1)-conj(VR3/normVR)*cos(ThC1))* \\
& (-conj(VR1/normVR)*sin(ThC1)-conj(VR3/normVR)*cos(ThC1))*Apanel*f*cos(ThC1)+(2-sig_n- \\
& sig_t)*dirac(conj(VR1/normVR)*sin(ThC1)+conj(VR3/normVR)*cos(ThC1))* \\
& (-conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))*(-conj(VR1/normVR)*sin(ThC1)- \\
& conj(VR3/normVR)*cos(ThC1))^2*Apanel*f*sin(ThC1)+2*(2-sig_n-sig_t)*heaviside(- \\
& conj(VR1/normVR)*sin(ThC1)-conj(VR3/normVR)*cos(ThC1))*(-conj(VR1/normVR)*sin(ThC1)- \\
& conj(VR3/normVR)*cos(ThC1))*Apanel*f*sin(ThC1))* \\
& (-conj(VR1/normVR)*cos(ThC1)+conj(VR3/normVR)*sin(ThC1))+(2-sig_n-sig_t)*heaviside(- \\
& conj(VR1/normVR)*sin(ThC1)-conj(VR3/normVR)*cos(ThC1))*(-conj(VR1/normVR)*sin(ThC1)- \\
& conj(VR3/normVR)*cos(ThC1))^2*Apanel*f*cos(ThC1)), \\
& rho*normVR^2*(sig_t*dirac(conj(VR1/normVR)*sin(ThC2)+conj(VR3/normVR)*cos(ThC2))*(conj(VR1/normVR)*cos(ThC2)- \\
& conj(VR3/normVR)*sin(ThC2))- \\
& conj(VR3/normVR)*sin(ThC2))*(conj(VR1/normVR)*sin(ThC2)+conj(VR3/normVR)*cos(ThC2))*Apanel*((-d- \\
& L*cos(ThC2))*VR2/normVR+f*VR1/normVR)+sig_t*heaviside(conj(VR1/normVR)*sin(ThC2)+conj(VR3/normVR)*cos(ThC2))* \\
& (conj(VR1/normVR)*cos(ThC2)-conj(VR3/normVR)*sin(ThC2))*Apanel*((-d- \\
& L*cos(ThC2))*VR2/normVR+f*VR1/normVR)+sig_t*heaviside(conj(VR1/normVR)*sin(ThC2)+conj(VR3/normVR)*cos(ThC2))* \\
& (conj(VR1/normVR)*sin(ThC2)+conj(VR3/normVR)*cos(ThC2))*Apanel*L*sin(ThC2)*VR2/normVR+sig_n*normVb/normVR*dirac(conj(VR1/normVR)*sin(ThC2)+ \\
& conj(VR3/normVR)*cos(ThC2))* \\
& (conj(VR1/normVR)*cos(ThC2)-conj(VR3/normVR)*sin(ThC2))* \\
& conj(VR3/normVR)*sin(ThC2))*(conj(VR1/normVR)*sin(ThC2)+conj(VR3/normVR)*cos(ThC2))*Apanel*f*sin(ThC2)+sig_n*normVb/normVR*heaviside(conj(VR1/normVR)*sin(ThC2)+ \\
& conj(VR3/normVR)*cos(ThC2))* \\
& (conj(VR1/normVR)*cos(ThC2)-conj(VR3/normVR)*sin(ThC2))- \\
& conj(VR3/normVR)*sin(ThC2))*Apanel*f*sin(ThC2)+sig_n*normVb/normVR*heaviside(conj(VR1/normVR)*sin(ThC2)+conj(VR3/normVR)*cos(ThC2))* \\
& (conj(VR1/normVR)*sin(ThC2)+conj(VR3/normVR)*cos(ThC2))*(conj(VR1/normVR)*sin(ThC2)+conj(VR3/normVR)*cos(ThC2))*c
\end{aligned}$$

[illegible]

```

(ThC4)*VR2/normVR+sig_n*normVb/normVR*dirac(conj(VR1/normVR)*sin(ThC4)+conj(VR3/normVR)*cos(ThC4))*(conj(VR1/normVR)*cos(ThC4)-conj(VR3/normVR)*sin(ThC4))*(conj(VR1/normVR)*sin(ThC4)+conj(VR3/normVR)*cos(ThC4))*Apanel*f*sin(ThC4)+sig_n*normVb/normVR*heaviside(conj(VR1/normVR)*sin(ThC4)+conj(VR3/normVR)*cos(ThC4))*(conj(VR1/normVR)*cos(ThC4)-conj(VR3/normVR)*sin(ThC4))*Apanel*f*sin(ThC4)+sig_n*normVb/normVR*heaviside(conj(VR1/normVR)*sin(ThC4)+conj(VR3/normVR)*cos(ThC4))*(conj(VR1/normVR)*sin(ThC4)+conj(VR3/normVR)*cos(ThC4))*Apanel*f*cos(ThC4)+(2-sig_n-sig_t)*dirac(conj(VR1/normVR)*sin(ThC4)+conj(VR3/normVR)*cos(ThC4))*(conj(VR1/normVR)*cos(ThC4)-conj(VR3/normVR)*sin(ThC4))*(conj(VR1/normVR)*sin(ThC4)+conj(VR3/normVR)*cos(ThC4))^2*Apanel*f*sin(ThC4)+2*(2-sig_n-sig_t)*heaviside(conj(VR1/normVR)*sin(ThC4)+conj(VR3/normVR)*cos(ThC4))*(conj(VR1/normVR)*sin(ThC4)+conj(VR3/normVR)*cos(ThC4))*Apanel*f*sin(ThC4)*(conj(VR1/normVR)*cos(ThC4)-conj(VR3/normVR)*sin(ThC4)))+(2-sig_n-sig_t)*heaviside(conj(VR1/normVR)*sin(ThC4)+conj(VR3/normVR)*cos(ThC4))*(conj(VR1/normVR)*sin(ThC4)+conj(VR3/normVR)*cos(ThC4))^2*Apanel*f*cos(ThC4)];

```

% Saturation Avoidance Logic Definitions

```

thetas = [ThC1; ThC2; ThC3; ThC4];
thetas_p = [deg2rad(-45); deg2rad(-45); deg2rad(45); deg2rad(45)];
theta_error = thetas_p - thetas;
P = eye(4)-pinv(A)*A;

```

% INNER LOOP CONTROL LAW - Minimum Norm Solution + Saturation Avoidance

```

ThCdot = -kf*pinv(A)*(gc-u) + kp*P*theta_error;

```

%% Control Panel Saturation Logic - restricts movements of control panels to +/- 45 degrees from preferred orientation

```

if ThC1 >= deg2rad(-5) & ThCdot(1) >= 0;
    ThCdot(1) = 0;
elseif ThC1 <= deg2rad(-85) & ThCdot(1) <= 0;
    ThCdot(1) = 0; end

if ThC2 >= deg2rad(-5) & ThCdot(2) >= 0;
    ThCdot(2) = 0;
elseif ThC2 <= deg2rad(-85) & ThCdot(2) <= 0;
    ThCdot(2) = 0; end

if ThC3 >= deg2rad(85) & ThCdot(3) >= 0;
    ThCdot(3) = 0;
elseif ThC3 <= deg2rad(5) & ThCdot(3) <= 0;
    ThCdot(3) = 0; end

if ThC4 >= deg2rad(85) & ThCdot(4) >= 0;
    ThCdot(4) = 0;
elseif ThC4 <= deg2rad(5) & ThCdot(4) <= 0;
    ThCdot(4) = 0; end

```

%% Dynamics

```

% Define Moment of Inertia (assumes no addition from control surfaces)

```

```

m = 10; B = 1/6*m*(2*d)^2;
I = [B 0 0; 0 B 0; 0 0 B];

```

```

% Euler Equation of Rotational Motion
wdot = inv(I)*gd + inv(I)*gc - inv(I)*skew(w)*I*w;

%% Kinematics - Quaternion Differential Equations
qdot = 1/2*(q4*w - skew(w)*q); q4dot = -1/2*transpose(w)*q;

%% Return the state vector
xdot = [wdot(1) wdot(2) wdot(3) qdot(1) qdot(2) qdot(3) q4dot ThCdot(1) ThCdot(2) ThCdot(3)
ThCdot(4)]';

```

Appendix B – MATLAB Jacobian Calculation Code

```
%          Capt M. LUKE GARGASZ
%    OPTIMAL SPACECRAFT ATTITUDE CONTROL USING AERODYNAMIC TORQUES
%          MARCH 2007
%          MASTER'S THESIS: AFIT/GA/ENY/07-M08
%
%          Contact email: luke_gargasz@hotmail.com

% Derivatives.m

close all;clear all;clc;

syms ThC1 ThC2 ThC3 ThC4 d L f VR1 VR2 VR3 Apanel rho sig_t sig_n normVR normVb

VR = [VR1; VR2; VR3]; VRunit = VR/normVR; Vb = 0.05*VR;

% Control Panel 7 -- Top / positive y-axis side
r7 = [-(d+L*cos(ThC1)); f; -d+L*sin(ThC1)];
n7f = [sin(ThC1); 0; cos(ThC1)];
n7r = -n7f;
alpha7f = acos(dot(VRunit,n7f));
alpha7r = acos(dot(VRunit,n7r));

% Control Panel 8 -- Top / negative y-axis side
r8 = [-(d+L*cos(ThC2)); -f; -d+L*sin(ThC2)];
n8f = [sin(ThC2); 0; cos(ThC2)];
n8r = -n8f;
alpha8f = acos(dot(VRunit,n8f));
alpha8r = acos(dot(VRunit,n8r));

% Control Panel 9 -- Bottom / positive y-axis side
r9 = [-(d+L*cos(ThC3)); f; (d+L*sin(ThC3))];
n9f = [-sin(ThC3); 0; -cos(ThC3)];
n9r = -n9f;
alpha9f = acos(dot(VRunit,n9f));
alpha9r = acos(dot(VRunit,n9r));

% Control Panel 10 -- Bottom / negative y-axis side
r10 = [-(d+L*cos(ThC4)); -f; (d+L*sin(ThC4))];
n10f = [-sin(ThC4); 0; -cos(ThC4)];
n10r = -n10f;
alpha10f = acos(dot(VRunit,n10f));
alpha10r = acos(dot(VRunit,n10r));

%% Calculate Ap
Ap7f = heaviside(cos(alpha7f))*cos(alpha7f)*Apanel;
Ap7r = heaviside(cos(alpha7r))*cos(alpha7r)*Apanel;
Ap8f = heaviside(cos(alpha8f))*cos(alpha8f)*Apanel;
Ap8r = heaviside(cos(alpha8r))*cos(alpha8r)*Apanel;
Ap9f = heaviside(cos(alpha9f))*cos(alpha9f)*Apanel;
Ap9r = heaviside(cos(alpha9r))*cos(alpha9r)*Apanel;
```

```
Ap10f = heaviside(cos(alpha10f))*cos(alpha10f)*Apanel;
Ap10r = heaviside(cos(alpha10r))*cos(alpha10r)*Apanel;
```

```
%% Calculate Cp
```

```
Cp7f = heaviside(cos(alpha7f))*cos(alpha7f)*Apanel*r7/Ap7f;
Cp7r = heaviside(cos(alpha7r))*cos(alpha7r)*Apanel*r7/Ap7r;
Cp8f = heaviside(cos(alpha8f))*cos(alpha8f)*Apanel*r8/Ap8f;
Cp8r = heaviside(cos(alpha8r))*cos(alpha8r)*Apanel*r8/Ap8r;
Cp9f = heaviside(cos(alpha9f))*cos(alpha9f)*Apanel*r9/Ap9f;
Cp9r = heaviside(cos(alpha9r))*cos(alpha9r)*Apanel*r9/Ap9r;
Cp10f = heaviside(cos(alpha10f))*cos(alpha10f)*Apanel*r10/Ap10f;
Cp10r = heaviside(cos(alpha10r))*cos(alpha10r)*Apanel*r10/Ap10r;
```

```
%% Calculate Gp
```

```
Gp7f = heaviside(cos(alpha7f))*cos(alpha7f)*Apanel*cross(r7,n7f);
Gp7r = heaviside(cos(alpha7r))*cos(alpha7r)*Apanel*cross(r7,n7r);
Gp8f = heaviside(cos(alpha8f))*cos(alpha8f)*Apanel*cross(r8,n8f);
Gp8r = heaviside(cos(alpha8r))*cos(alpha8r)*Apanel*cross(r8,n8r);
Gp9f = heaviside(cos(alpha9f))*cos(alpha9f)*Apanel*cross(r9,n9f);
Gp9r = heaviside(cos(alpha9r))*cos(alpha9r)*Apanel*cross(r9,n9r);
Gp10f = heaviside(cos(alpha10f))*cos(alpha10f)*Apanel*cross(r10,n10f);
Gp10r = heaviside(cos(alpha10r))*cos(alpha10r)*Apanel*cross(r10,n10r);
```

```
%% Calculate Gpp
```

```
Gpp7f = heaviside(cos(alpha7f))*cos(alpha7f)^2*Apanel*cross(r7, n7f);
Gpp7r = heaviside(cos(alpha7r))*cos(alpha7r)^2*Apanel*cross(r7, n7r);
Gpp8f = heaviside(cos(alpha8f))*cos(alpha8f)^2*Apanel*cross(r8, n8f);
Gpp8r = heaviside(cos(alpha8r))*cos(alpha8r)^2*Apanel*cross(r8, n8r);
Gpp9f = heaviside(cos(alpha9f))*cos(alpha9f)^2*Apanel*cross(r9, n9f);
Gpp9r = heaviside(cos(alpha9r))*cos(alpha9r)^2*Apanel*cross(r9, n9r);
Gpp10f = heaviside(cos(alpha10f))*cos(alpha10f)^2*Apanel*cross(r10,n10f);
Gpp10r = heaviside(cos(alpha10r))*cos(alpha10r)^2*Apanel*cross(r10,n10r);
```

```
%% Calculate the control torque
```

```
gc7f = rho*normVR^2*(sig_t*Ap7f* cross(Cp7f,VRunit) +sig_n*(normVb/normVR)*Gp7f +(2-sig_n-
sig_t)*Gpp7f);
gc7r = rho*normVR^2*(sig_t*Ap7r* cross(Cp7r,VRunit) +sig_n*(normVb/normVR)*Gp7r +(2-sig_n-
sig_t)*Gpp7r);
gc8f = rho*normVR^2*(sig_t*Ap8f* cross(Cp8f,VRunit) +sig_n*(normVb/normVR)*Gp8f +(2-sig_n-
sig_t)*Gpp8f);
gc8r = rho*normVR^2*(sig_t*Ap8r* cross(Cp8r,VRunit) +sig_n*(normVb/normVR)*Gp8r +(2-sig_n-
sig_t)*Gpp8r);
gc9f = rho*normVR^2*(sig_t*Ap9f* cross(Cp9f,VRunit) +sig_n*(normVb/normVR)*Gp9f +(2-sig_n-
sig_t)*Gpp9f);
gc9r = rho*normVR^2*(sig_t*Ap9r* cross(Cp9r,VRunit) +sig_n*(normVb/normVR)*Gp9r +(2-sig_n-
sig_t)*Gpp9r);
gc10f = rho*normVR^2*(sig_t*Ap10f*cross(Cp10f,VRunit)+sig_n*(normVb/normVR)*Gp10f+(2-sig_n-
sig_t)*Gpp10f);
gc10r = rho*normVR^2*(sig_t*Ap10r*cross(Cp10r,VRunit)+sig_n*(normVb/normVR)*Gp10r+(2-sig_n-
sig_t)*Gpp10r);
```

```
gc = gc7f+gc7r + gc8f+gc8r + gc9f+gc9r + gc10f+gc10r;
```

```
A = jacobian(gc,[ThC1 ThC2 ThC3 ThC4])
```


Bibliography

- Frik, Martin A. "Attitude Stability of Satellites Subjected to Gravity Gradient and Aerodynamic Torques." *AIAA Journal*: 1780-1785 (October 1970).
- Kumar, Renjith R., Daniel D. Mazanek, and Michael L. Heck. "Simulation and Shuttle Hitchhiker Validation of Passive Satellite Aerostabilization," *Journal of Spacecraft and Rockets*: 806-811 (September-October 1995).
- Kumar, Renjith R., Daniel D. Mazanek, and Michael L. Heck. "Parametric and Classical Resonance in Passive Satellite Aerostabilization," *Journal of Spacecraft and Rockets*: 228-234 (March-April 1996).
- Hughes, Peter C. *Spacecraft Attitude Dynamics*. Mineola NY: Dover Publications, 2004.
- Janicik, Jeffery L. "Implementing Standard Microsatellites for Responsive Space." *Proceedings of the 1st Responsive Space Conference*. April 2003.
- Kuhns, Mark D. and Armando A. Rodriquez. "Singularity Avoidance Control Laws for a Multiple CMG Spacecraft Attitude Control System." *Proceedings of the American Control Conference*: 2892-3 (June 1994).
- Meirovitch, L. and F. B. Wallace, Jr. "On the Effect of Aerodynamic and Gravitational Torques on the Attitude Stability of Satellites." *AIAA Journal*: 2196-2202 (December 1966).
- NASA. "STS-77 Space Shuttle Mission Report." June 1996.
- Petty, John I. "Orbital Elements." *Classical Orbital Elements Picture*. n. pag. <http://www.spaceflight.nasa.gov/realdata/elements/graphs.html>. 4 January 2006.
- Psiaki, Mark L. "Nanosatellite Attitude Stabilization Using Aerodynamics and Active Magnetic Torquing." *Journal of Guidance, Control, and Dynamics*: 347-355 (May-June 2004).
- Ravindran, R. and P. C. Hughes. "Optimal Aerodynamic Attitude Stabilization of Near-Earth Satellites." *Journal of Spacecraft and Rockets*: 499-506 (July 1972).
- Senate Armed Services Committee. *Statement of General Lance W. Lord, Commander, Air Force Space Command, before the Senate Armed Services Committee, Strategic Forces Subcommittee, United States Senate on March 16, 2005*.

Vallado, David A. *Fundamentals of Astrodynamics and Applications*. El Segundo CA: Microcosm Press, 2001.

Wie, B., Weiss, H., Araposthathis, A. “Quaternion Feedback Regulator for Spacecraft Eigenaxis Rotations.” *Journal of Guidance, Control, and Dynamics*: 375-380 (May-June 1989).

Wie, Bong. *Space Vehicle Dynamics and Control*. Reston VA: American Institute of Aeronautics and Astronautics, Inc., 1998.

Wiesel, William E. *Spaceflight Dynamics*. Boston MA: Irwin/McGraw-Hill, 1997.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 074-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, 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 the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 22 Mar 07		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) August 2005 - March 2007	
4. TITLE AND SUBTITLE Optimal Spacecraft Attitude Control Using Aerodynamic Torques				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Gargas, Michael Luke, Captain, USAF				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GA/ENY/07-M08	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Space Vehicles Directorate, Air Force Research Laboratory 3550 Aberdeen Ave SE Kirtland AFB, NM 87117 Dr. Thomas Lovell DSN 263-4132				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/VSES	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This thesis introduces a method of three-axis spacecraft attitude control using only aerodynamic torques. Attitude actuation is achieved using four control panels mounted on the rear of a cubical spacecraft bus. The controller consists of an outer loop using linear state feedback to determine desired control torque and an inner loop to choose appropriate control panel angles. The inner loop uses a Jacobian-based approach to invert the nonlinear relationship between panel angles and generated torque. Controller performance is evaluated via simulations, which show that three-axis control is possible over a range of initial angles and angular rates. The analysis used partial accommodation theory as the basis for aerodynamic torque calculations and assumed a rotating atmosphere with an exponential density profile.					
15. SUBJECT TERMS attitude control systems, aerodynamic drag, control systems, stability					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Nathan A. Titus, Lt Col, USAF
U	U	U	UU	91	19b. TELEPHONE NUMBER (Include area code) (937) 255-6565, ext 4597 (Nathan.Titus@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18