

Spacecraft Attitude Tracking Control

Matthew R. Long

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Aerospace Engineering

Dr. Christopher D. Hall - chair

Dr. Mark R. Anderson

Dr. Frederick H. Lutze

June 1999

Blacksburg, Virginia

Keywords: Spacecraft Dynamics, Target Tracking, Lyapunov Control Theory

Copyright 1999, Matthew R. Long

Spacecraft Attitude Tracking Control

Matthew R. Long

(ABSTRACT)

The problem of reorienting a spacecraft in to acquire a moving target is investigated. The spacecraft is modeled as a rigid body with N axisymmetric wheels controlled by axial torques, and the kinematics are represented by Modified Rodrigues Parameters. The trajectory, denoted the reference trajectory, is one generated by a virtual spacecraft that is identical to the actual spacecraft. The open-loop reference attitude, angular velocity, and angular acceleration tracking commands are constructed so that the solar panel vector is perpendicular to the sun vector during the tracking maneuver. We develop a nonlinear feedback tracking control law, derived from Lyapunov stability and control theory, to provide the control torques for target tracking. The controller makes the body frame asymptotically track the reference motion when there are initial errors in the attitude and angular velocity. A spacecraft model, based on the X-ray Timing Explorer spacecraft, is used to demonstrate the effectiveness of the Lyapunov controller in tracking a given target.

Acknowledgments

I want to thank my advisor, Dr. Chris Hall, for his time and effort that he has expended in teaching me spacecraft dynamics. His expertise and technical insight into spacecraft dynamics has helped me form this research into something worthwhile. Over the last two years, Dr. Hall has enhanced my education through academics, research skills, and his own experiences as an engineer. He has been supportive and patient with me when aspects of the research became frustrating. For this, I am grateful.

I wish to thank Dave Folta and the Guidance, Navigation, and Control Center (GNCC) at NASA's Goddard Spaceflight Center for funding this research project over the past year. Without their support, this project would not have been possible.

Additional support for this project was provided by AI-Solutions in Lanham Maryland. I wish to thank Darrel Conway and Paul Noonan for providing technical insight and excellent trouble shooting tips for FreeFlyer.

Contents

1	Introduction	1
2	Literature Review	5
2.1	Formation Flying Spacecraft	5
2.2	Target Tracking Kinematics	7
2.3	Control Theory Overview	14
2.3.1	Lyapunov Control Theory	14
2.3.2	Controllers Used For Target Tracking	18
3	Equations of Motion	22
3.1	System Model	22
3.1.1	The Virtual Spacecraft Model	26
3.1.2	Kinematics	27

3.2	Orbit Model	29
4	The Reference Trajectory	33
4.1	Derivation of Ideal Pointing Attitude	34
4.2	The Ideal Angular Velocity	46
4.3	The Ideal Angular Acceleration	49
5	Control Laws	52
5.1	Error Kinematics	53
5.2	The Lyapunov Tracking Controller	54
6	Tracking Simulation	60
6.1	XTE Mission Profile	61
6.2	XTE Attitude Control System Overview	62
6.3	XTE Spacecraft Model	64
7	Results and Discussion	69
7.1	Initial Conditions for Tracking Maneuver	70
7.2	Simulation Results	70

8	Summary and Conclusions	95
A	Calculation of Sun Unit Vector	98
	Bibliography	100

List of Figures

2.1	Stable Origin	16
2.2	Asymptotically Stable Origin	17
2.3	Unstable Origin	18
3.1	Gyrostat Model with N -Momentum Wheels	23
3.2	Spacecraft in the Gravitational Field of One Inertial Spherical Primary ⁵	31
4.1	Schematic of Reference Attitude	34
4.2	Panel Axis Orientation	36
4.3	Intermediate Reference Frame “a”	40
4.4	Intermediate Reference Frame “c”	41
4.5	Graphical Illustration of the Yaw-Steering Maneuver	44
7.1	The Case 1 Time History of (a) σ_b and (b) σ_r	72

7.2	The Case 1 Time History of (a) ω_b and (b) ω_r	73
7.3	The Case 1 Time History of (a) $\delta\sigma$ and (b) $\delta\omega$	73
7.4	The Lyapunov Function for Case 1	74
7.5	The Yaw-Steering Condition for Case 1	75
7.6	Case 1 Time Histories of (a) The Momentum Wheel Feedback Control Law in Eq. (5.13) and (b) The Desired Control	76
7.7	The XTE Reaction Wheels Speeds for Case 1	76
7.8	The External Torque for Case 1	77
7.9	The Case 1 Time History of (a) The XTE System Angular Momentum and (b) The Reaction Wheel Angular Momentum	78
7.10	The Case 1 Time History of the Magnitudes of the Four Largest Terms In Eq. (5.13)	79
7.11	The Case 2 Time History of (a) $\delta\sigma$ and (b) $\delta\omega$	80
7.12	The Lyapunov Function for Case 2	81
7.13	The Yaw-Steering Condition for Case 2	82
7.14	The Momentum Wheel Feedback Control Law in Eq. (5.13) for Case 2	83
7.15	The XTE Reaction Wheels Speeds for Case 2	83
7.16	The Case 2 Time History of the Magnitudes of the Four Largest Terms In Eq. (5.13)	84
7.17	The Case 3 Time History of (a) $\delta\sigma$ and (b) $\delta\omega$	85

7.18	The Lyapunov Function for Case 3	86
7.19	The Yaw-Steering Condition for Case 3	86
7.20	The Momentum Wheel Feedback Control Law in Eq. (5.13) for Case 3	87
7.21	The XTE Reaction Wheels Speeds for Case 3	87
7.22	The Case 3 Time History of the Magnitudes of the Four Largest Terms In Eq. (5.13)	88
7.23	The Case 3 Time History of (a) $\delta\sigma$ and (b) $\delta\omega$ Without the Reference Torque	89
7.24	The Linearized Case Time History of (a) $\delta\sigma$ and (b) $\delta\omega$	90
7.25	The Yaw-Steering Condition for Linearized Case	91
7.26	The Lyapunov Function for Linearized Case	92
7.27	The Momentum Wheel Feedback Control Law in Eq. (5.13) for Linearized Case . .	93
7.28	The XTE Reaction Wheels Speeds for the Linearized Case	93

List of Tables

4.1	Known Vectors Used in Deriving Pointing Attitude	35
6.1	XTE Physical Data	66
7.1	XTE Orbital Data	71
7.2	XTE Initial Conditions	71
7.3	The Lyapunov Controller Gains	71
7.4	FLOP Count for Tracking Maneuver Simulation	92

Chapter 1

Introduction

Formation Flying spacecraft is a developing technology that has the potential to expand the capability of future Earth-observing science missions. The development of small low-cost spacecraft has led to the idea of flying several spacecraft in formation to achieve coordinated sensor measurements. A formation is defined as the coordinated motion of a group of vehicles where the vehicle positions relative to each other are important.¹ An example of formation flying is the Earth Observing-1 (EO-1)/Landsat 7 (L-7) mission. The two spacecraft are placed in nearly coplanar orbits with a nominal separation between spacecraft of 450 kilometers.⁹ Both of these spacecraft are to co-observe geographic sites for scientific comparison of the imaging sensors onboard the spacecraft.

Most of the types of science missions proposed, similar to the EO-1/L-7 mission, require coordinated attitude control of each spacecraft in the formation for simultaneous viewing and tracking

of geographic targets on the earth. This attitude control can be broken down into two parts: the relative control of the formation and the individual spacecraft control. The relative control, which is not considered in this research, involves the position and movement of each spacecraft with respect to one another so that the formation can perform mission objectives and requirements. Ideas such as using the Doppler shift⁹ of a spacecraft-to-spacecraft crosslink carrier signal to derive position and separation measurements have been proposed for relative control. Individual spacecraft control is important for maneuvering the spacecraft for data collection, communication, and thermal control, to name a few.

For the data collection aspect of individual spacecraft control, a target is chosen, and the spacecraft is maneuvered so the imaging sensor is able to point at the target. This is usually known as attitude tracking control. There are three key aspects for of attitude tracking control; the pointing, the angular velocity, and the control needed for the attitude and tracking. Tracking control requires that the spacecraft be oriented so that its imaging sensor points directly at the target while it rotates about its body axes to keep the sensor aligned with the target.

This maneuvering is accomplished through a control law that generates control torques to re-orient the spacecraft to the desired attitude and desired angular velocity. When control is performed on three mutually perpendicular spacecraft axes, the spacecraft is known as a three-axis stabilized spacecraft.²⁷ The control torques for a three-axis stabilized spacecraft are also required to compensate for environmental effects such as aerodynamic drag and gravity gradient torques that cause the spacecraft orientation to drift. These control torques could be generated externally by thrusters, internally by momentum wheels, or by a combination of both.

The objective of this research is to develop attitude tracking algorithms and control laws for a three-axis stabilized spacecraft to view and track targets on the rotating earth. We first define the system and orbital models used for a rigid body spacecraft with momentum wheels. We then outline the open-loop computation of the ideal attitude, angular velocity, and angular acceleration commands needed for target alignment and tracking. This target tracking trajectory also incorporates sun tracking commands so that a solar array axis can remain perpendicular the sun direction while a sensor tracks the target. We then develop a nonlinear feedback controller using Lyapunov control theory. The controller generates the necessary axial momentum wheel torques to make the spacecraft body frame track the ideal open-loop trajectory by eliminating initial tracking errors. Finally, simulation results are presented for a simple spacecraft model based on the X-ray Timing Explorer (XTE). These simulation results verify that the feedback control law does drive initial tracking errors asymptotically to zero and that the controller can be linearized. The objective of this research is to develop attitude tracking algorithms and control laws for a three-axis stabilized spacecraft to view and track targets on the rotating earth. We first define the system and orbital models used for a rigid body spacecraft with momentum wheels. We then outline the open-loop computation of the ideal attitude, angular velocity, and angular acceleration commands needed for target alignment and tracking. This target tracking trajectory also incorporates sun tracking commands so that a solar array axis can remain perpendicular the sun direction while a sensor tracks the target. We then develop a nonlinear feedback controller using Lyapunov control theory. The controller generates the necessary axial momentum wheel torques to make the spacecraft body frame track the ideal open-loop trajectory by eliminating initial tracking errors. Finally, sim-

ulation results are presented for a simple spacecraft model based on the X-ray Timing Explorer (XTE). These simulation results verify that the feedback control law does drive initial tracking errors asymptotically to zero and that the controller can be linearized.

Chapter 2

Literature Review

We present a brief overview of previous target tracking simulation and control investigations. Section 2.1 of this chapter discusses attitude and tracking control research that has been done for formation flying. The rest of the chapter deals with the tracking control problem of a single three-axis stabilized spacecraft. Our research is an extension of the control work presented in Sections 2.2–2.3.

2.1 Formation Flying Spacecraft

Formation flying has not been widely investigated in the literature, but some simulation work has been done in the last few years. Folta *et al.*⁶ simulated a two-spacecraft formation using three different types of formation configurations. Each spacecraft carried a nadir-looking imaging instrument, and the three formation configurations differed in separation distances between the

spacecraft and target observation times. The effect that these separations and observation times had on attitude control and instrument field of view (FOV) were analyzed. The attitude control of the formation was not given, but derivations for the attitude, attitude control errors, and field of view coverage areas were presented. The performance of a formation to observe ground targets simultaneously was evaluated by plotting attitude control errors against time and FOV errors in terms of the along-track and cross-track orbit directions. The FOV plots showed which formation configuration had the highest sensor overlap area when viewing a target with and without attitude errors.

Gramling *et al.*⁹ also studied a two-satellite formation. They discussed the Onboard Navigation System (ONS) for relative navigation of the Earth-Observing-1 (EO-1)/Landsat-7 (L-7) formation. Gramling *et al.* also outlined the EO-1/L-7 formation configuration and discussed how the ONS was applied, along with the Global Positioning System (GPS), to make formation control autonomous. They illustrated how the ONS tracking measurements are derived from the Doppler shift of a spacecraft-to-spacecraft crosslink carrier signal. The performance of the ONS was investigated in terms of tracking measurement type and quality, tracking frequency, and the relative orbital geometry of the coplanar formation. Simulation results of the EO-1/L-7 mission were presented in terms of Doppler measurement errors and formation in-track and cross-track errors over time.

Folta and Quinn⁷ extended the work of Gramling *et al.* on the EO-1/L-7 mission concept. They studied formation control of that mission using the three formation types discussed in Ref. 6. An autonomous closed-loop three-axis navigation control algorithm for EO-1/L-7 was presented

for formation maneuvers. The controller allows the spacecraft to perform complex three-axis maneuvers autonomously. Groundtrack and inclination maneuvers were simulated using this algorithm and an autonomous spacecraft simulation package. The algorithm was found to be robust in groundtrack, inclination, and orbit transfer control.

One of the few other papers that present an algorithm for formation control is the work done by Ulybyshev.²⁵ His work pertains to station-keeping of a constellation using a linear-quadratic regulator (LQR) for feedback control. Ulybyshev presented the derivation of the spacecraft equations of motion using the Clohessy-Wiltshire equations. He then derived an LQR controller for formation-keeping and presented an analytical solution for a two-satellite constellation. A twelve-satellite constellation was simulated using the LQR controller. The constellation control was not autonomous, and required data from a mission control center. He showed that the controller minimized the along-track relative displacements between spacecraft and the orbital period displacements relative to a reference circular orbit.

2.2 Target Tracking Kinematics

Tracking a target involves pointing at an object and then moving to stay aligned with that target for a given length of time. Pointing at a target requires a specific attitude that aligns the instrument boresight axis with the target. The spacecraft must then generate the angular velocity to track a target as it moves in its orbit.

The simplest case to track is where the target is inertially fixed, say a star, and the spacecraft is

initially at rest, then the final spacecraft body rates are zero (rest-to-rest maneuver).¹⁰ The Hubble space telescope is one example of a spacecraft that tracks inertially fixed targets. If the target is on the rotating earth, then the spacecraft final angular velocity is not zero.

Hablani¹¹ developed an algorithm to generate the reference trajectory for when the final angular velocity is not zero. He considered a payload rigidly attached to the spacecraft, such that the sensor initially faces in the zenith direction. The tracking commands for this payload orientation were based on an 2-1-3 Euler angle sequence. Hablani used this sequence because the pitch rotation compensates naturally for the once-per-orbit rotation about the pitch axis without any coupling from the subsequent roll rotation.¹²

Defining the initial boresight direction to be along the zenith direction meant that the spacecraft body triad $(\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3)$ was initially aligned with the local-vertical-local-horizontal (LVLH) triad $(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \hat{\mathbf{c}}_3)$ before target acquisition. The desired boresight orientation was defined to be along the negative yaw axis $(\hat{\mathbf{b}}_3)$. The other two spacecraft body axes, $\hat{\mathbf{b}}_1$ and $\hat{\mathbf{b}}_2$, were defined to be along the roll and pitch axis, respectively. To get the target to the focal plane center of the instrument, the spacecraft was then rotated about the $\hat{\mathbf{c}}_2$ axis by a commanded pitch angle θ_{yc} and then about the $\hat{\mathbf{c}}_1$ axis by a commanded roll angle θ_{xc} to acquire the target. The ‘3’ rotation was not needed since a rotation about the sensor boresight axis does not affect target tracking. It only changes the spacecraft’s attitude in the other two body axes.

Following this, Hablani showed that the pitch-roll angle commands were

$$\theta_{yc} = \tan^{-1} [\Leftrightarrow(\mathbf{1} \cdot \hat{\mathbf{c}}_1) / \Leftrightarrow(\mathbf{1} \cdot \hat{\mathbf{c}}_3)] \quad (2.1)$$

$$\theta_{xc} = \sin^{-1}[(\mathbf{l} \cdot \hat{\mathbf{c}}_2)/\|\mathbf{l}\|] \quad (2.2)$$

$$\theta_{zc} = 0 \quad (2.3)$$

where \mathbf{l} is the line-of-sight vector from the spacecraft to the target. The negative signs in the numerator and denominator of Eq. (2.1) were retained to determine the correct quadrant. Also, Hablani noted that the roll command would always be $-\pi/2 \leq \theta_{xc} \leq \pi/2$. The yaw angle θ_{zc} and its rate $\dot{\theta}_{zc}$ were zero since the target vector was aligned with the boresight axis.

The tracking angular velocity and acceleration commands were derived analogously to the position commands. The details of these derivations can be found in Refs 12 and 11. The angular velocity commands were given as

$$\omega_{xc} = (\dot{\mathbf{l}} \cdot \hat{\mathbf{b}}_2)/\|\mathbf{l}\| \quad (2.4)$$

$$\omega_{yc} = (\dot{\mathbf{l}} \cdot \hat{\mathbf{b}}_1)/\|\mathbf{l}\| \quad (2.5)$$

Hablani found the yaw component of the angular velocity by making use of the fact that the spacecraft has three desired angular rates: the mean motion of the spacecraft in a circular orbit $\omega_c \hat{\mathbf{c}}_2$, the pitch rate command $\dot{\theta}_{yc} \hat{\mathbf{c}}_2$, and the roll rate command $\dot{\theta}_{xc} \hat{\mathbf{c}}_1$. The yaw component of the angular velocity became

$$\omega_{zc} = \omega_{yc} \tan \theta_{xc} \quad (2.6)$$

Thus, the inertial yaw rate ω_{zc} was not found to be a second-order quantity even for a small roll angle. Likewise, the commanded angular acceleration was proved to be equal to

$$\dot{\omega}_{xc} = (\ddot{\mathbf{l}} \cdot \hat{\mathbf{b}}_2 + 2\dot{l} \omega_{xc})/\|\mathbf{l}\| + \omega_{yc}\omega_{zc} \quad (2.7)$$

$$\dot{\omega}_{yc} = (\ddot{\mathbf{l}} \cdot \hat{\mathbf{b}}_1 + 2\dot{l} \omega_{yc})/\|\mathbf{l}\| + \omega_{xc}\omega_{zc} \quad (2.8)$$

$$\dot{\omega}_{zc} = \Leftrightarrow \dot{\omega}_{yc} \tan \theta_{xc} \Leftrightarrow \omega_{xc} \omega_{yc} \sec^2 \theta_{xc} \quad (2.9)$$

where \dot{l} is the rate of change of the line-of-sight vector in the body frame. Hablani pointed out that the acceleration commands are useful for feedforward and/or determining the inertial resistance of the spacecraft while tracking a moving target.

Similar to the position, rate, and acceleration commands for the Euler 2-1-3 sequence, Hablani also derived these tracking kinematics for an Euler 1-2-3 sequence. For this case, the boresight axis was initially facing nadir and the spacecraft was rotated through the roll-pitch sequence to acquire the target. The roll-pitch sequence was not found to be as effective for target tracking because it becomes singular at 90° pitch angle, and acquiring a near-earth target causes the pitch angle to cross 90° . The rate and acceleration commands for the yaw axis are also more complicated than the previous definitions because the orbital rate ω_o cannot be expressed as simply about the pitch axis \hat{c}_2 as it is for the pitch-roll sequence. Hablani argued that the roll-pitch sequence was useful, because when the pitch-roll commands are singular, the roll-pitch sequence is not. The results of the roll-pitch derivation can be found in Refs. 10 and 11.

Schaub *et al.*²² also investigated an algorithm for generating commands to track moving targets. Their trajectory was computed by doing maneuvers about a principal axis of rotation using Euler's principal rotation theorem. Euler's principal rotation theorem states that any coordinate frame can be related to another coordinate frame through a single-axis rotation. This single-axis rotation was derived in terms of a minimal three-parameter set known as Modified Rodrigues Parameters (MRPs). The MRP's are defined in terms of the principal rotation axis \hat{e} and the principal

rotation angle ϕ as

$$\boldsymbol{\sigma} = \hat{\mathbf{e}} \tan (\phi/4) \quad (2.10)$$

They are also defined in terms of Euler parameters as

$$\sigma_i = \frac{q_i}{1+q_0} \quad i = 1, 2, 3 \quad (2.11)$$

It can be seen that Modified Rodrigues Parameters are singular at a principal rotation of $\pm 360^\circ$ where $q_0 \rightarrow \pm 1$. Schaub *et al.* noted that MRP's are not unique, and reversing the signs of the \mathbf{q}_i 's in Eq. (2.11) generates a second set of $\boldsymbol{\sigma}_i$'s. This alternative set of MRP's is called the “shadow set,” and it is singular at 0° instead of at $\pm 360^\circ$. Both sets describe the same physical orientation and satisfy the same differential equation of motion, only differing in initial conditions.²² The advantage of using Modified Rodrigues Parameters is that if a singularity is encountered with the original set, one can switch to the shadow set and avoid the singularity. The only effect of switching parameter sets is the discontinuity that occurs at the switching point. The transformation between the “original” and “shadow” set was defined to be

$$\boldsymbol{\sigma}_s = \boldsymbol{\sigma} / \boldsymbol{\sigma}^T \boldsymbol{\sigma} \quad (2.12)$$

Schaub *et al.* explained that the choice in distinguishing between these two sets is purely arbitrary so they chose the switching condition to be $\boldsymbol{\sigma}^T \boldsymbol{\sigma} = 1$. The magnitude of the orientation vector is bounded between $0 \leq \|\boldsymbol{\sigma}\| \leq 1$ which means that the principal rotation angle is restricted to $0^\circ \leq \phi \leq 180^\circ$.

Schaub *et al.* used a principal axis rotational maneuver because it is close to the optimal solution without having to perform expensive calculations. Their target tracking trajectory was derived

for a near-minimum time and a near-minimum fuel maneuver. The optimal control for both of these were a rest-to-rest, or a “bang-bang,” control through a principal angle θ_f . Schaub *et al.* anticipated that the “bang-bang” control would excite significant vibrations in the flexible degrees of freedom and cause errors in the imaging systems. As a result, the control switches were smoothed using cubic splines and a user-controlled “sharpness” parameter α , on the inputs to the trajectory. A set of bang-bang control switch equations for the principal commanded angular acceleration were presented in terms of the sharpness parameter, time, and the maximum principal angular acceleration. A double integration was carried out on these equations to obtain the commanded angular velocity, the commanded angle, and the final maneuver time. Schaub *et al.* noted that $\alpha = 0$ generates the bang-bang instantaneous torque switches and $\alpha = 0.25$ generates the smoothest curve shapes.

Schaub *et al.* also presented a modified version of this optimal trajectory for a final angular velocity at the final maneuver time. The commanded angular velocity and angular accelerations were derived relative to a moving target frame, and then transformed relative to the inertial frame through coordinate frame transformations. Schaub pointed out that when the target frame has zero motion, these equations collapse back to the rest-to-rest case. The final maneuver time must be determined iteratively because the final target position, not the initial, is known in advance.

In addition to target tracking, some work has also been done on computing attitude commands for sun tracking. Kalweit¹⁶ wanted to keep a payload axis nadir pointing while reorienting the spacecraft about the yaw axis so that the solar array normal points towards the sun. The spacecraft attitude was defined relative to the local-vertical-local-horizontal (LVLH) satellite-centered coordinates. The ideal solar array rotation angle and spacecraft yaw angle for sun tracking were derived

for the general case of a satellite moving in an elliptical orbit around a planet. Both the ideal solar array rotation angle and the yaw angle were derived as functions of the satellite inclination, true anomaly, right ascension of the ascending node, and the right ascension of the sun vector in the ecliptic plane.

In conclusion, Hablani¹¹ and Schaub *et al.*²² presented two different approaches for deriving the perfect alignment and target tracking commands. The problem with Hablani's approach is that the ideal tracking commands are constructed only for an Euler angle 2-1-3 sequence with the sensor axis initially facing the zenith direction. There is no flexibility in choosing attitude parameters or where the boresight axis is initially facing. Schaub *et al.*'s optimal tracking trajectory is designed for an eigenaxis maneuver only. Neither one of these algorithms contain sun tracking commands. In Chapter 4, we present an algorithm for determining an attitude independent reference attitude from a rotation matrix that includes sun tracking commands. Constructing the attitude this way provides greater flexibility in the choice of spacecraft attitude parameters.

2.3 Control Theory Overview

This section provides an overview of how a spacecraft can be controlled during a tracking maneuver. Lyapunov control theory is often used for nonlinear systems that are not easily modeled as linear systems. This section provides a description of Lyapunov's second method that is used to determine the stability of a dynamical system.

2.3.1 Lyapunov Control Theory

Stability of control systems is important for controller design. For a linear and time invariant-system, Nyquist and Routh's methods can be used to determine the stability.²¹ However, if the system is nonlinear, or linear but time-varying, then these types of stability methods do not apply.

One method that is frequently used for determining the stability of nonlinear and/or time-varying systems is Lyapunov's second, or direct method. In 1892, A.M. Lyapunov presented two methods for determining the stability of dynamic systems described by ordinary differential equations.²¹ The first method, which will not be discussed here, determines whether or not a system has local stability at the point of interest, and uses the solutions to the differential equations in the stability analysis. Lyapunov's second method allows the stability of a system to be determined without solving the state equations. This is a nice feature since solving nonlinear and time-varying systems can be difficult.

In place of solving the differential equations to determine stability, a scalar function $V(\mathbf{x})$ is used along with the ode's to answer the stability question. This scalar, energy-like function is known as a Lyapunov function. There are no general procedures for choosing a Lyapunov function, which means that there is no unique Lyapunov function for a given system. Meirovitch¹⁹ pointed out that Lyapunov's second method for a general dynamical system is more of a philosophical approach because of the flexibility in choosing the Lyapunov function.

There are three main types of stability associated with Lyapunov's direct method: stability, asymptotic stability, and instability. Figure 2.1 graphically illustrates the stability of the origin

in phase space where x_1 is the coordinate variable and x_2 is the velocity. The curves represent constant energy contours with $V(\mathbf{x}) > 0$, which says that the origin is a stable equilibrium point. Meirovitch¹⁹ noted that Lyapunov's direct method is an extension to the energy methods used to determine a system's equations of motion, with the Lyapunov function being the counterpart to the potential energy function.

Figure 2.2 shows an asymptotically stable system where the distance between the origin and the instantaneous state \mathbf{x} is continually decreasing ($\dot{V}(\mathbf{x}) < 0$ causing $\mathbf{x} \rightarrow \mathbf{0}$). As a result, the system energy is decreasing. Figure 2.3 illustrates the unstable case for $\dot{V}(\mathbf{x}) > 0$ as \mathbf{x} increases. The trajectory spirals away from the origin, thus making the system energy increase and the solution diverge from the equilibrium point.

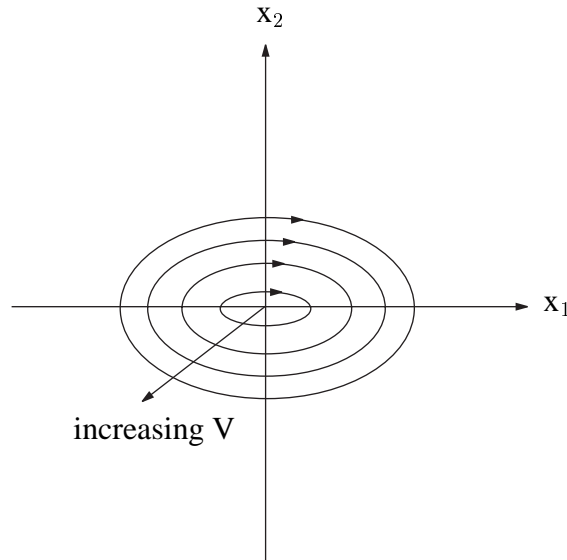


Figure 2.1: Stable Origin

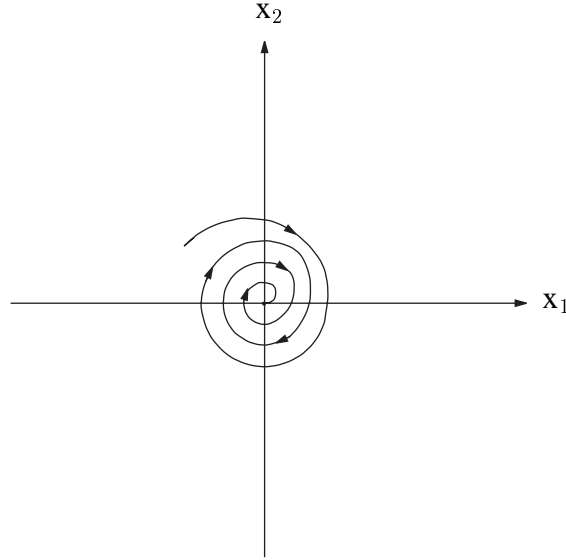


Figure 2.2: Asymptotically Stable Origin

To determine if a Lyapunov function falls under one of the cases presented in the previous figures, the sign definiteness of the function and its time derivative are tested according to the following theorems:^{19,20}

Theorem (1). If there exists a positive definite function ($V(\mathbf{x}) > 0$) whose time derivative is semi-negative definite ($\dot{V}(\mathbf{x}) \leq 0$), or identically zero along every trajectory of $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$, then the trivial solution, $\mathbf{x} = \mathbf{0}$, is stable.

Theorem (2). If there exists a positive definite function ($V(\mathbf{x}) > 0$) whose time derivative is negative definite ($\dot{V}(\mathbf{x}) < 0$) along every trajectory of $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$, then the trivial solution is asymptotically stable.

Theorem (3). If the trivial solution is asymptotically stable (Theorem 2), and if $V(\mathbf{x})$ is radially unbounded (i.e. $V(\mathbf{x}) \rightarrow \infty$ as $\|\mathbf{x}\| \rightarrow \infty$), then the trivial solution is globally asymptotically stable.

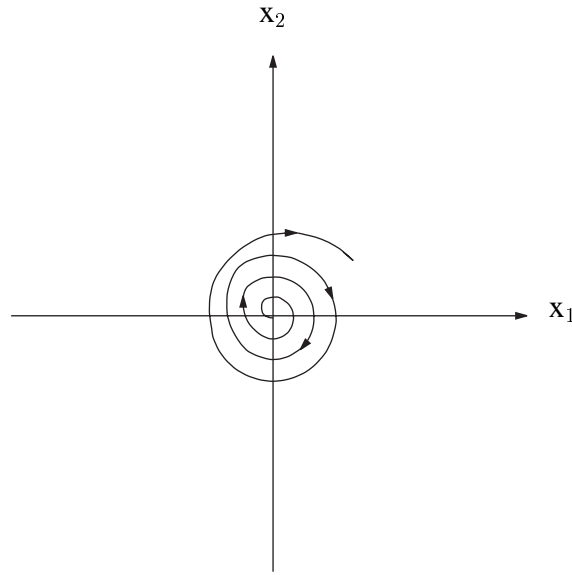


Figure 2.3: Unstable Origin

Theorem (4). If there exists for the system a function $V(\mathbf{x})$ whose time derivative $\dot{V}(\mathbf{x})$ is positive definite along every trajectory of $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$ and $V(\mathbf{x}) > 0$ for arbitrarily small values of \mathbf{x} , then the trivial solution is unstable.

These theorems give sufficient conditions for determining the stability of a system. Mohler²⁰ also pointed out for theorem 3 that if $V(\mathbf{x}) \rightarrow \infty$ as $\|\mathbf{x}\| \rightarrow \infty$ and $\dot{V}(\mathbf{x}) \leq 0$ then the solution to $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$ cannot grow without bound. The system then becomes Lagrange stable.

These stability definitions are used along with $\dot{V}(\mathbf{x})$ to develop the required control law. The control law is constructed from the terms that appear in $\dot{V}(\mathbf{x})$ so that the time derivative of the Lyapunov function satisfies the above stability definitions. The next section illustrates how a Lyapunov control law is applied to the target tracking problem.

2.3.2 Controllers Used For Target Tracking

Schaub *et al.*²² developed a nonlinear tracking control law using Lyapunov's second method outlined in the previous section. Their system model was a flexible spacecraft with three reaction wheels aligned along the three body axes. Their nonlinear tracking control law was developed to assure that their previously discussed optimal reference trajectory was tracked asymptotically in the absence of external torques, by generating the necessary reaction wheel torques to drive the spacecraft to the reference attitude and the reference body rates.

To get their feedback control law, Schaub *et al.* started with the following Lyapunov function:

$$V = \frac{1}{2} \delta \boldsymbol{\omega}^T \mathbf{J} \delta \boldsymbol{\omega} + 2K \ln(1 + \boldsymbol{\sigma}^T \boldsymbol{\sigma}) \quad (2.13)$$

where $\delta \boldsymbol{\omega}$ is the error in the body angular velocity, \mathbf{J} is the matrix containing the spacecraft and the transverse reaction wheel inertia, K is a scalar gain for the attitude error feedback, and $\boldsymbol{\sigma}$ is the attitude error between the body and the ideal body frame to be tracked in terms of Modified Rodrigues Parameters. To guarantee global asymptotic stability, Schaub *et al.* chose their controller so that the first derivative of Eq. (2.13) is negative definite. They noted that the first derivative of V does not guarantee any stability if external torques are present, just that $\delta \boldsymbol{\omega}$ will remain bounded. As a result, if external torques are present, $\delta \boldsymbol{\omega}$ still decays to zero. However, the attitude error $\boldsymbol{\sigma}$ decays to small finite offset, which makes their system Lagrange stable.

An extension of this Lyapunov controller work was carried out by Hall *et al.*¹⁵ Their control laws were designed for a rigid body spacecraft that used both thrusters and momentum wheels to track given attitude motions. The thrusters are used for “coarse” attitude control, and the momen-

tum wheels provide the “fine” control to eliminate tracking errors. Their ideal tracking trajectory is generated by a “virtual” rigid body spacecraft that has the same inertia properties as the real spacecraft. The reference attitude, angular velocity, and angular acceleration are assumed known. The desired, control torque, \mathbf{g}_R is chosen to be provided by the thrusters in the form of a “bang-bang” command.

Using the Lyapunov function in Eq. (2.13), Hall *et al.* developed three tracking controllers; 1) The first controller uses thruster torques to implement the bang-bang control law, since thrusters generally are not capable of providing continuously varying torques¹⁵ and momentum wheels are used to correct tracking errors. The thruster control is:

$$\mathbf{g}_e = \mathbf{g}_R \quad (2.14)$$

The feedback control law for the momentum wheels was shown to satisfy:

$$\begin{aligned} \mathbf{A}\mathbf{g}_a &= \mathbf{h}_b^\times \mathbf{J}^{-1}(\mathbf{h}_b \Leftrightarrow \mathbf{A}\mathbf{h}_a) + \mathbf{g}_R \\ &\Leftrightarrow \mathbf{J}\boldsymbol{\omega}_b^\times \boldsymbol{\delta}\boldsymbol{\omega} \Leftrightarrow \mathbf{J}\mathbf{R}^{br}(\boldsymbol{\delta}\boldsymbol{\sigma})\mathbf{I}^{-1}\mathbf{h}_R^\times \mathbf{I}^{-1}\mathbf{h}_R \\ &\Leftrightarrow \mathbf{J}\mathbf{R}^{br}(\boldsymbol{\delta}\boldsymbol{\sigma})\mathbf{I}^{-1}\mathbf{g}_R + k_1\boldsymbol{\delta}\boldsymbol{\omega} + k_2\boldsymbol{\delta}\boldsymbol{\sigma} \end{aligned} \quad (2.15)$$

where \mathbf{h}_b is the body angular momentum, \mathbf{h}_a is the angular momenta of the wheels, \mathbf{g}_e is the $N \times 1$ matrix of external thruster torques, and \mathbf{g}_a is the $N \times 1$ matrix of the internal reference axial torques. The superscript \times denotes a skew-symmetric matrix which is defined in Section 3.1. The matrix \mathbf{A} contains the axial unit vectors of the momentum wheels, and the matrices \mathbf{I} and \mathbf{J} are defined to be the inertia of the rigid body and the rigid body with momentum wheels, respectively. The matrix $\mathbf{R}^{br}(\boldsymbol{\delta}\boldsymbol{\sigma})$ is the rotation matrix from the reference to the actual body frame, which also

yields the attitude tracking error $\delta\sigma$. 2) The second controller uses the thruster torques exclusively if no initial condition errors ($\omega_B(0) = \omega_R(0), \sigma_B(0) = \sigma_R(0)$) are present. The thruster control law reduced to the same form as in Eq. (2.14), and the momentum wheel controller became

$$\mathbf{A}\mathbf{g}_a = (\mathbf{A}\mathbf{h}_a)^\times \omega_R \quad (2.16)$$

where ω_R is the reference angular velocity. 3) The last controller assumes that the thrusters can generate continuous control profiles. The thruster control law is chosen to implement the feedforward plus the nonlinear portion¹⁵ of the control law through the thrusters as

$$\mathbf{g}_e = \mathbf{h}_b^\times \mathbf{J}^{-1}(\mathbf{h}_b \Leftrightarrow \mathbf{A}\mathbf{h}_a) + \mathbf{J}\omega_b^\times \delta\omega + \mathbf{J}\mathbf{R}^{br}(\delta\sigma)\dot{\omega}_R \quad (2.17)$$

As a result, the momentum wheel controller is chosen to be a linear feedback control law in the form

$$\mathbf{A}\mathbf{g}_a = k_1\delta\omega + k_2\delta\sigma \quad (2.18)$$

Hall *et al.* proved, in the same fashion as Schaub *et al.*,²² that all three sets of controllers were globally asymptotically stable in tracking given reference motions.

Having presented Lyapunov control schemes for target tracking, we extend the work done by Hall *et al.*¹⁵ Our control scheme is virtually identical to theirs, but we do not use any thrusters for attitude control. We develop a controller that uses momentum wheels for attitude control since thrusters can expel gases that might interfere with delicate optics on a spacecraft. We begin by presenting the equations of motion used in deriving our feedback controller.

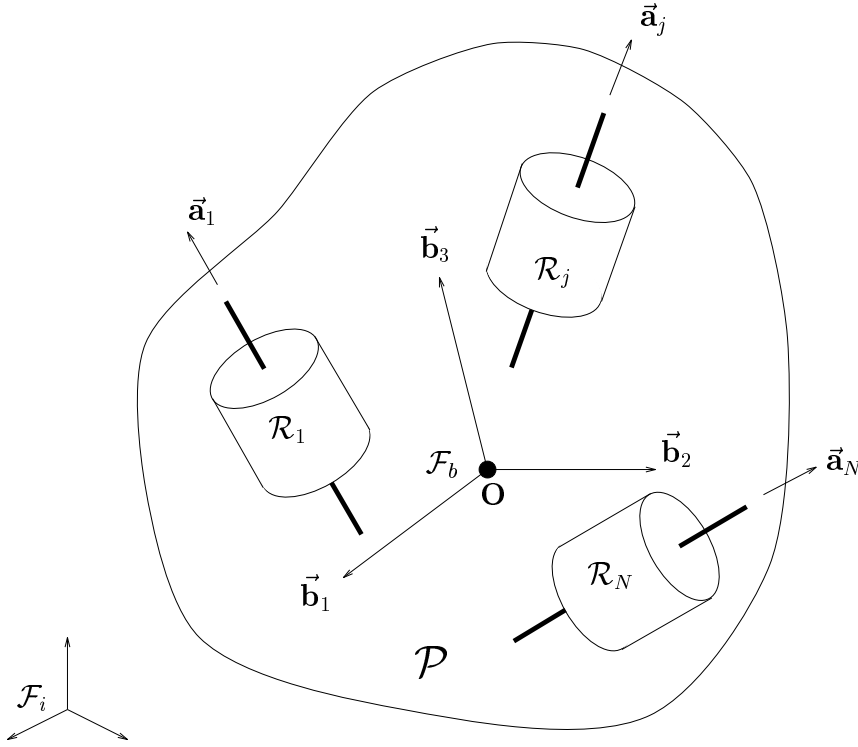
Chapter 3

Equations of Motion

This chapter presents the equations of motion for a rigid body spacecraft with momentum wheels. The dynamics in Section 3.1 are Euler's rotational equations, and the kinematics are presented in terms of Modified Rodrigues Parameters. These equations are presented for the actual spacecraft and the fictitious spacecraft that generates the reference trajectory. Mathematical models for the spacecraft's orbit and environmental effects are presented in Section 3.2.

3.1 System Model

In this section, a system model is presented for use in developing tracking control algorithms. The equations of motion presented here follow the notation developed in Hughes⁵ and Hall.^{13,14} We consider a rigid spacecraft P , shown in Figure 3.1, with N rigid axisymmetric momentum wheels R_i , $i = 1, \dots, N$. The wheels have an arbitrary, but fixed orientation with respect to the body. Let

Figure 3.1: Gyrostat Model with N -Momentum Wheels

\mathcal{F}_b denote the body frame with the origin at the center of mass of the system $P + \sum_{i=1}^N R_i$, and \mathcal{F}_i denote the inertial frame. When a rigid spacecraft has one or more rigid axisymmetric wheels spinning about their axes of symmetry, then the system is known as a gyrostat.¹⁴

Let \mathbf{I} represent the moment of inertia of the system, including the momentum wheels, $\mathbf{I}_s = \text{diag}\{I_{s1}, \dots, I_{sN}\}$ denote the axial moments of inertia of the momentum wheels, and the wheel axial vectors \mathbf{a}_j are represented by a $3 \times N$ matrix $\mathbf{A} = [\mathbf{a}_1 \cdots \mathbf{a}_N]$. We do not assume that \mathcal{F}_b is a principal frame, so \mathbf{I} is not necessarily a diagonal matrix. As developed in Refs. 14 and 13, we express all vectors and tensors in a platform-fixed, nonprincipal frame, designated as the “pseudo-principal” frame. Without losing generality, we let \mathcal{F}_b represent the pseudo-principal frame, which

is chosen so that the inertia-like matrix $\mathbf{J} = \mathbf{I} \Leftrightarrow \mathbf{A}\mathbf{I}_s\mathbf{A}^T$ is diagonal. The 3×1 system angular momentum vector in \mathcal{F}_b is defined as

$$\mathbf{h}_b = \mathbf{I}\boldsymbol{\omega}_b + \mathbf{A}\mathbf{I}_s\boldsymbol{\omega}_s \quad (3.1)$$

where $\boldsymbol{\omega}_b$ and $\boldsymbol{\omega}_s$ are the angular velocities of the body and the wheels, respectively. The $N \times 1$ matrix of the axial angular momenta of the wheels \mathbf{h}_a , is defined for the N -wheel gyrostat as

$$\mathbf{h}_a = \mathbf{I}_s\mathbf{A}^T\boldsymbol{\omega}_b + \mathbf{I}_s\boldsymbol{\omega}_s \quad (3.2)$$

The dynamics of the gyrostat are Euler's rotational equations of motion which come from the following equation from analytical dynamics¹⁹

$$\dot{\mathbf{v}} = \overset{\circ}{\mathbf{v}} + \boldsymbol{\omega}^\times \mathbf{v} \quad (3.3)$$

where \mathbf{v} represents any vector expressed in a frame with angular velocity $\boldsymbol{\omega}$. The “ \circ ” denotes differentiation of \mathbf{v} with respect to a moving coordinate frame. Replacing \mathbf{v} with \mathbf{h} yields $\overset{\circ}{\mathbf{h}}$, which is the rate of change of the angular momentum relative to \mathcal{F}_b , and $\dot{\mathbf{h}}$ which becomes the external torque acting on the system. Solving for the rate of change of the angular momentum in the body frame results in Eq. (3.4) which is written in matrix form as

$$\dot{\mathbf{h}}_b = \Leftrightarrow \boldsymbol{\omega}_b^\times \mathbf{h}_b + \mathbf{g}_e \quad (3.4)$$

$$\dot{\mathbf{h}}_a = \mathbf{g}_a \quad (3.5)$$

where \mathbf{g}_e is the column vector of external torques that act on the body, and \mathbf{g}_a represents the $N \times 1$ matrix of the internal axial control torques applied by the platform to the momentum wheels. The

notation ω_b^\times represents a skew-symmetric matrix form of a vector, i.e.

$$\omega_b = \begin{bmatrix} \omega_{1b} \\ \omega_{2b} \\ \omega_{3b} \end{bmatrix} \Leftrightarrow \omega_b^\times = \begin{bmatrix} 0 & \Leftrightarrow\omega_{3b} & \omega_{2b} \\ \omega_{3b} & 0 & \Leftrightarrow\omega_{1b} \\ \Leftrightarrow\omega_{2b} & \omega_{1b} & 0 \end{bmatrix} \quad (3.6)$$

so that $\Leftrightarrow\omega_b^\times \mathbf{h}_b$ is the matrix equivalent to $\Leftrightarrow\omega_b \times \mathbf{h}_b$. Comparing Eqs. (3.1) and (3.2), and using the definition for \mathbf{J} , the angular momentum of the body can be written as

$$\begin{aligned} \mathbf{h}_b &= \mathbf{I}\omega_b + \mathbf{A}(\mathbf{h}_a \Leftrightarrow \mathbf{I}_s \mathbf{A}^T \omega_b) \\ &= \mathbf{I}\omega_b + \mathbf{A}\mathbf{h}_a \Leftrightarrow \mathbf{A}\mathbf{I}_s \mathbf{A}^T \omega_b \\ &= (\mathbf{J} + \mathbf{A}\mathbf{I}_s \mathbf{A}^T) \omega_b + \mathbf{A}\mathbf{h}_a \Leftrightarrow \mathbf{A}\mathbf{I}_s \mathbf{A}^T \omega_b \\ &= \mathbf{J}\omega_b + \mathbf{A}\mathbf{h}_a \end{aligned} \quad (3.7)$$

From this, the body angular velocity can be expressed as

$$\omega_b = \mathbf{J}^{-1}(\mathbf{h}_b \Leftrightarrow \mathbf{A}\mathbf{h}_a) \quad (3.8)$$

Substituting Eq. (3.8) into Eq. (3.4) yields the final expressions for the equations of motion:

$$\dot{\mathbf{h}}_b = \mathbf{h}_b^\times \mathbf{J}^{-1}(\mathbf{h}_b \Leftrightarrow \mathbf{A}\mathbf{h}_a) + \mathbf{g}_e \quad (3.9)$$

$$\dot{\mathbf{h}}_a = \mathbf{g}_a \quad (3.10)$$

The external torques, \mathbf{g}_e , are comprised of environmental torques, and possibly control torques using thrusters or magnetic torquer rods. The wheel torques, \mathbf{g}_a , are comprised of control torques applied by the motors, and possible friction torques. In this thesis, we assume that the gravity gradient torques are the only environmental torques present, and the motor torques are the only

control torques used to maneuver the spacecraft. The next section shows how Eqs. (3.4) and (3.5) are used to define the fictitious spacecraft model and the reference control torque \mathbf{g}_{ar} .

3.1.1 The Virtual Spacecraft Model

The desired trajectory to be tracked comes from the trajectory generated by a “virtual” spacecraft in a reference coordinate frame. Let \mathcal{F}_r represent this reference frame which is fixed at the center of mass of this virtual spacecraft as in Fig. 3.1. In Ref. 15, the “virtual” spacecraft is assumed to be a rigid body. Here, we assume that the virtual spacecraft is a gyrostat, with the same properties as the real spacecraft.

Since the virtual spacecraft has the same inertial and wheel parameters as the real spacecraft, the reference frame dynamics are the same as in Section 3.1, except that the subscript b is replaced with r :

$$\dot{\mathbf{h}}_r = \mathbf{h}_r^\times \mathbf{J}^{-1}(\mathbf{h}_r \Leftrightarrow \mathbf{A}\mathbf{h}_{ar}) + \mathbf{g}_e \quad (3.11)$$

$$\dot{\mathbf{h}}_{ar} = \mathbf{g}_{ar} \quad (3.12)$$

where $\mathbf{h}_r = \mathbf{J}\boldsymbol{\omega}_r + \mathbf{A}\mathbf{h}_{ar}$. The external torque \mathbf{g}_e remains the same, but we need to solve for the virtual spacecraft’s axial torque \mathbf{g}_{ar} . The torque \mathbf{g}_{ar} is the torque that would generate the desired trajectory in the absence of initial condition errors. The torque \mathbf{g}_{ar} comes from first noting that $\dot{\mathbf{h}}_r$ can also be expressed by differentiating \mathbf{h}_r to get

$$\dot{\mathbf{h}}_r = \mathbf{J}\dot{\boldsymbol{\omega}}_r + \mathbf{A}\dot{\mathbf{h}}_{ar} = \mathbf{J}\dot{\boldsymbol{\omega}}_r + \mathbf{A}\mathbf{g}_{ar} \quad (3.13)$$

Equating Eqs. (3.13) and (3.11) yields the following expression for the desired axial control torque

$\mathbf{A}\mathbf{g}_{ar}$:

$$\mathbf{A}\mathbf{g}_{ar} = \mathbf{h}_r^\times \mathbf{J}^{-1}(\mathbf{h}_r \Leftrightarrow \mathbf{A}\mathbf{h}_{ar}) + \mathbf{g}_e \Leftrightarrow \mathbf{J}\dot{\boldsymbol{\omega}}_r \quad (3.14)$$

where $\boldsymbol{\omega}_r = \mathbf{J}^{-1}(\mathbf{h}_r \Leftrightarrow \mathbf{A}\mathbf{h}_{ar})$ is the desired angular velocity for target tracking that will be derived in the next chapter.

The kinematics for the virtual spacecraft are also parameterized by Modified Rodrigues Parameters (MRP's). As with the gyrostad dynamics, the kinematics are the same for the reference and actual body attitude, and are distinguished by the subscripts r and b , respectively. The kinematics in terms of MRP's are derived in Section 3.1.2.

3.1.2 Kinematics

Following Refs. 22 and 15, the kinematic equations are written in terms of Modified Rodrigues Parameters (MRP's). We use MRP's because they are a minimal representation of the spacecraft attitude, and because singularities can be avoided using MRP's. As discussed in Section 2.2, the singularity associated with the MRP's at $\pm 360^\circ$ can be avoided by switching to the shadow set of MRP's. If we had used a set of Euler angles, singularities would be computationally more expensive. Recall that Hablani¹¹ pointed out that, for the 1-2-3 set of Euler angles, the pitch angle does cross the 90° singularity for target tracking. Quaternions (Euler parameters) could have also been used, since they have no singularities associated with them. However, they require a fourth parameter and a once redundant set of equations ($q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$) to avoid singularities.

The definition of the MRP's are repeated here for convenience as

$$\boldsymbol{\sigma} = \hat{\mathbf{e}} \tan(\phi/4) \quad (3.15)$$

where $\hat{\mathbf{e}}$ is the principal rotation axis and ϕ is the principal rotation angle. As in Ref. 22, we define the transformation from the regular set to the shadow set as

$$\boldsymbol{\sigma}_s = \Leftrightarrow \boldsymbol{\sigma} / \boldsymbol{\sigma}^T \boldsymbol{\sigma} \quad (3.16)$$

As in Ref. 22, the switching condition is defined to be $\boldsymbol{\sigma}^T \boldsymbol{\sigma} = 1$. This condition causes the magnitude of the orientation vector to be bounded between $0 \leq \|\boldsymbol{\sigma}\| \leq 1$, which means that the principal rotation angle is restricted to $\Leftrightarrow 180^\circ \leq \phi \leq 180^\circ$. We also define the direction-cosine matrix,²³ or a rotation matrix, that transforms a vector from one coordinate frame to another:

$$\mathbf{R} = \mathbf{1} + \frac{4(1 \Leftrightarrow \boldsymbol{\sigma}^T \boldsymbol{\sigma})}{(1 + \boldsymbol{\sigma}^T \boldsymbol{\sigma})^2} \boldsymbol{\sigma}^\times + \frac{8}{(1 + \boldsymbol{\sigma}^T \boldsymbol{\sigma})^2} (\boldsymbol{\sigma}^\times)^2 \quad (3.17)$$

where $\mathbf{1}$ is the 3×3 identity matrix and $\boldsymbol{\sigma}^\times$ is a skew-symmetric matrix defined earlier. To get the attitude vector from the rotation matrix \mathbf{R} , we adopt the following transformation:²²

$$\begin{aligned} \beta_0 &= \sqrt{(\text{tr } \mathbf{R} + 1)/2} \\ \sigma_1 &= (\mathbf{R}_{2,3} \Leftrightarrow \mathbf{R}_{3,2}) / [4\beta_0(1 + \beta_0)] \\ \sigma_2 &= (\mathbf{R}_{3,1} \Leftrightarrow \mathbf{R}_{1,3}) / [4\beta_0(1 + \beta_0)] \\ \sigma_3 &= (\mathbf{R}_{1,2} \Leftrightarrow \mathbf{R}_{2,1}) / [4\beta_0(1 + \beta_0)] \end{aligned} \quad (3.18)$$

where tr represents the trace of a matrix and β_0 is an Euler parameter. Note that β_0 is defined so that $\beta_0 \geq 0$, which guarantees $0 \leq \|\boldsymbol{\sigma}\| \leq 1$.

The kinematic differential equation for the attitude is given by the following

$$\dot{\sigma} = \mathbf{G}(\sigma)\omega \quad (3.19)$$

where the matrix $\mathbf{G}(\sigma)$ is defined as

$$\mathbf{G}(\sigma) = \frac{1}{2} \left(\mathbf{1} + \sigma^\times + \sigma^T \sigma \Leftrightarrow \frac{1 + \sigma^T \sigma}{2} \mathbf{1} \right) \quad (3.20)$$

It should be noted that the actual implementation of the reference trajectory in Chapter 4 does not require the solution to Eq. (3.19) for σ_r . Since the reference attitude is constructed through rotation matrices, we use Eq. (3.18) to determine the reference attitude and the attitude error vector. However, Eq. (3.19) is used in the derivation of the control law in Chapter 5. Having now completely described the equations of motion for a gyrostat, we proceed to define the orbital equations of motion in Section 3.2.

3.2 Orbit Model

For simplicity, we use the two-body equation of motion to describe a circular, or Keplerian orbit. The two-body equation is only an approximation of orbital dynamics, and for actual simulations should not be used because information such as the earth's oblateness, perturbations from other planets, aerodynamic drag, and solar radiation pressure torques is lost.

The two-body equation²⁶ is derived from Newton's law of gravitation, and is for a system containing two celestial bodies, or a planet and a spacecraft. Our system assumes that the earth is the primary attracting body and a spacecraft is the secondary body. By defining the gravitational

constant as $\mu = Gm_{\oplus}$, where m_{\oplus} is the mass of the earth, the mass of the spacecraft is assumed to be negligible compared to the earth. The earth and satellite are also assumed to be spherically symmetrical. As a result, both the earth and the spacecraft can be treated as a point mass. In addition, there are no external or internal forces acting on the system other than the gravitational forces which act along the line joining the two bodies. With these assumptions, the two-body orbital equation of motion is defined to be

$$\ddot{\mathbf{r}}_s = \Leftrightarrow \frac{\mu}{\|\mathbf{r}_s\|^3} \mathbf{r}_s \quad (3.21)$$

where \mathbf{r}_s is the position vector from the center of the earth to the center of mass of the spacecraft in \mathcal{F}_i . Note that for near-earth spacecraft, the geocentric equatorial system serves as the inertial system, and for interplanetary spacecraft the heliocentric system is used as an inertial system. The two-body equation can also be used to describe satellite orbits around other planets by changing the value of μ and defining the inertial system to be referenced from the planet's center.

Having defined the orbital equations of motion, we now turn our attention to the external torque \mathbf{g}_e . The only external torque that we assume present in Eqs. (3.9) and (3.14) is the uncontrolled gravity gradient torque. The consequent variations in the specific gravitational force over a spacecraft body leads, in general, to a torque about the body mass center.⁵ In contrast, if the gravitational field were uniform, then the center of mass would be at the same location as the center of gravity, and the gravitational torque about the mass center would be zero. The following four assumptions⁵ greatly simplify the gravity gradient torque expression.

- (a) Only one celestial primary need be considered.

- (b) This primary possesses a spherically symmetrical mass distribution.
- (c) The spacecraft is small compared to its distance from the mass center of the primary.
- (d) The spacecraft consists of a single body.

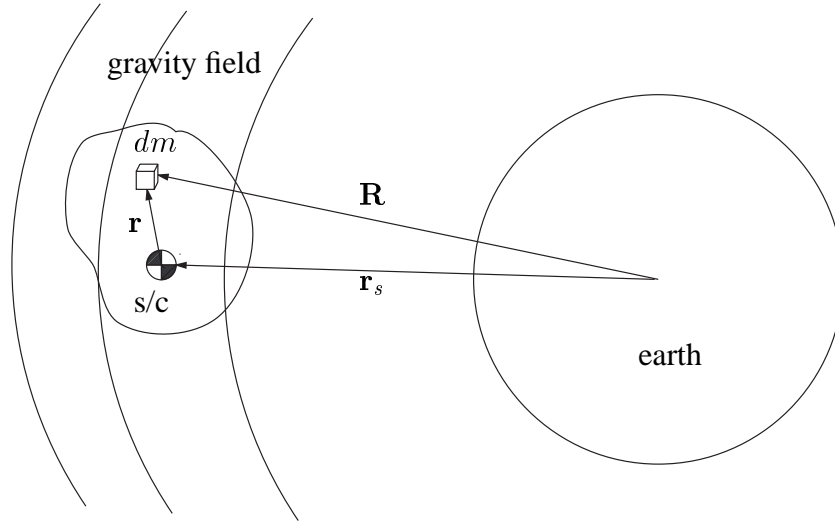


Figure 3.2: Spacecraft in the Gravitational Field of One Inertial Spherical Primary⁵

Using these four assumptions, the gravity gradient torque over \mathcal{P} , referenced at the center of mass, is

$$\mathbf{g}_e = \oint_{\mathcal{P}} \frac{\mathbf{r} \times \mathbf{R}}{R^3} dm \quad (3.22)$$

where \mathbf{r} is the vector from the spacecraft center of mass to dm and \mathbf{R} is the vector from the center of the earth to dm as shown in Fig. 3.2. Expanding Eq. (3.22) and applying the previous four assumptions, the vector form of the gravity gradient torque is found to be

$$\mathbf{g}_e = 3 \frac{\mu}{\|\mathbf{r}_s\|^3} \hat{\mathbf{o}}_3^\times \mathbf{I} \hat{\mathbf{o}}_3 \quad (3.23)$$

where $\hat{\mathbf{o}}_3$ is a unit vector of the form

$$\hat{\mathbf{o}}_3 = \frac{\mathbf{r}_s}{\|\mathbf{r}_s\|} \quad (3.24)$$

This vector is defined to be the “nadir” vector which points to the location on the earth directly beneath the spacecraft.

We have presented equations of motion that are used to model the attitude dynamics of a spacecraft in a circular orbit. The dynamics are modeled for an actual spacecraft, and its fictitious counterpart, that have N axisymmetric momentum wheels for attitude control and kinematics defined in terms of the Modified Rodrigues Parameters. The orbit is modeled using the two-body equation of motion with gravity gradient torque as the only environmental torque. The two-body approximation is sufficient to generate the necessary spacecraft position and velocity vectors without requiring expensive computations. Chapter 4 illustrates how the results from the orbital equations are used in determining the reference trajectory of the virtual spacecraft.

Chapter 4

The Reference Trajectory

Tracking a moving target requires a body orientation that allows some instrument fixed in the body to point at a target, and the required rotation and translation of the body to stay pointed at the moving target. Since we are dealing with a spacecraft assumed to be moving in a specified orbit, we only concern ourselves with the pointing and rotational maneuvers needed for target tracking. In addition, we also require that the attitude be constructed to satisfy solar power requirements by performing a yaw-steering maneuver¹⁶ about the sensor axis. We define the pointing attitude and the angular velocity that allows the spacecraft body reference frame \mathcal{F}_r to track the target and the sun as the “reference” trajectory. The reference trajectory is computed in an open-loop manner from known spacecraft position, velocity, sensor boresight, solar panel, and sun vectors.

4.1 Derivation of Ideal Pointing Attitude

Pointing at a target requires a specific attitude to make the position vector from the target to the spacecraft co-linear with the instrument boresight as illustrated in Fig. 4.1. The instrument axis

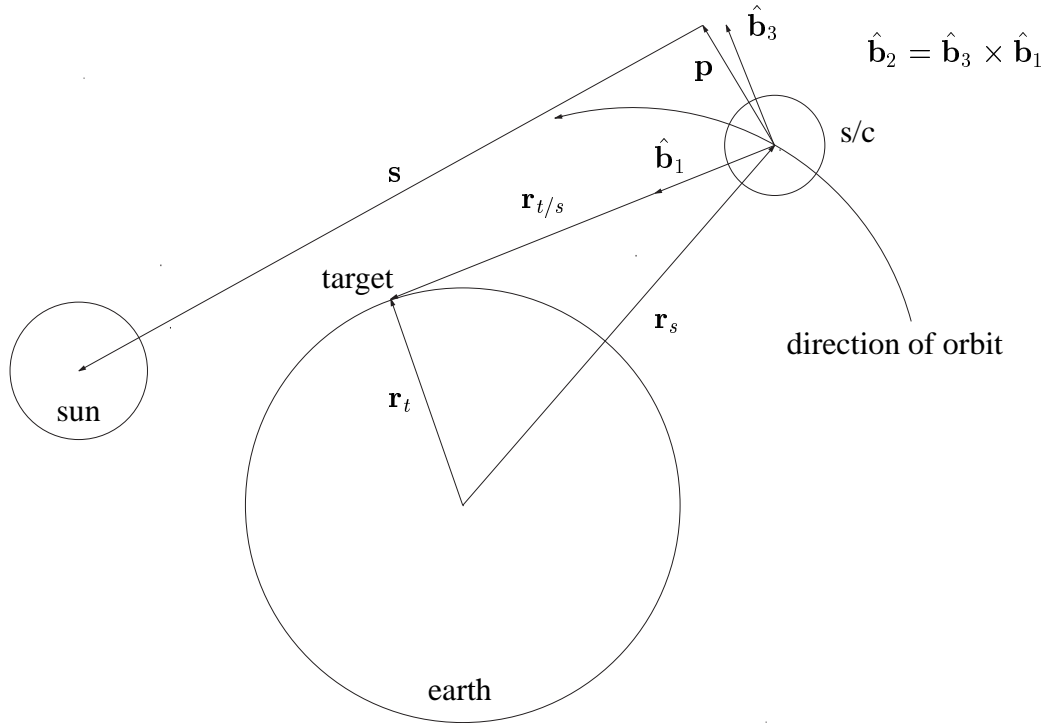


Figure 4.1: Schematic of Reference Attitude

can be any unit vector fixed in \mathcal{F}_b , which is defined to be the same in \mathcal{F}_r . Here, we define the instrument, or boresight, axis to be along the “1” direction in \mathcal{F}_b and \mathcal{F}_r :

$$\mathbf{a}_b = \mathbf{a}_r = [1 \ 0 \ 0]^T \quad (4.1)$$

From Figure 4.1, it can easily be seen that the target position vector with respect to the space-

craft, $\mathbf{r}_{t/s}$, can be written as

$$\mathbf{r}_{t/s} = \mathbf{r}_t \Leftrightarrow \mathbf{r}_s \quad (4.2)$$

where \mathbf{r}_t represents the position vector from the center of the earth to the spacecraft. The vector \mathbf{r}_s is assumed known from orbital data, and the target position vector is expressed in \mathcal{F}_i as

$$\mathbf{r}_{ti} = R_{\oplus} [\cos(\delta_t) \cos(\theta_{GST} + L_t) \quad \cos(\delta_t) \sin(\theta_{GST} + L_t) \quad \sin(\delta_t)]^T \quad (4.3)$$

where δ_t and L_t are the latitude and longitude of the target, respectively, θ_{GST} is Greenwich sidereal time measured from a given epoch, and R_{\oplus} is the radius of the earth. Table 4.1 illustrates the coordinate frames that these vectors are known. The sun vector, \mathbf{s} , is known from the algorithm

Table 4.1: Known Vectors Used in Deriving Pointing Attitude

	\mathcal{F}_b	\mathcal{F}_o	\mathcal{F}_i
\mathbf{r}_s	\Leftrightarrow	\checkmark	\checkmark
\mathbf{r}_t	\Leftrightarrow	\checkmark	\checkmark
\mathbf{a}	\checkmark	\checkmark	\checkmark
\mathbf{p}	\checkmark	\Leftrightarrow	\Leftrightarrow
\mathbf{s}	\Leftrightarrow	\checkmark	\checkmark

outlined in Appendix A that computes the sun unit vector in \mathcal{F}_i from the true longitude and the obliquity of the ecliptic of date. The solar panel vector, illustrated in Fig. 4.2, is the axis about which the panel can rotate. The goal of the yaw-steering maneuver is to align the sun vector with the vector normal to the solar panel in Fig. 4.2 by rotating the panel about its axis. This allows the solar cells to generate the maximum amount of electricity for the spacecraft. However,

we are ignoring the rotation of the panels and assuming that they are fixed about \mathbf{p} to simplify the yaw-steering maneuver and the construction of the reference attitude. Instead, we are rotating the spacecraft to satisfy the power requirements. We are also ignoring any dynamic effects of the flexible solar arrays on the system. The panel vector is defined in \mathcal{F}_b and \mathcal{F}_r as an arbitrary unit vector.

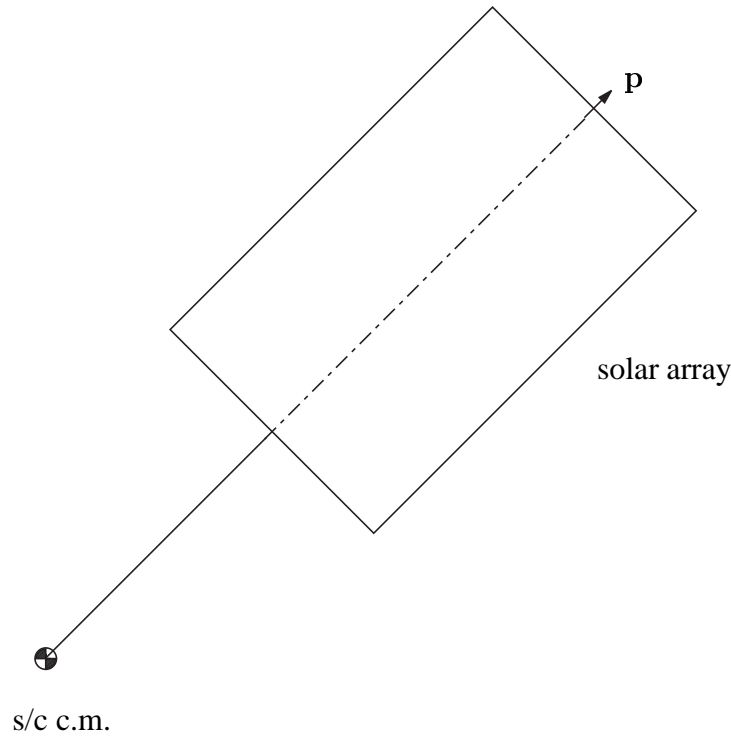


Figure 4.2: Panel Axis Orientation

To find the required pointing attitude, we notice that Eq. (4.3) can also be written as

$$\mathbf{r}_{ti} = \mathbf{r}_{si} + \mathbf{R}^{ir} D \mathbf{a}_r \quad (4.4)$$

where D represents the range from the spacecraft to the target. The rotation matrix \mathbf{R}^{ri} yields the reference attitude, σ_r , with respect to the inertial frame. The reference attitude is represented

in terms of Modified Rodrigues Parameters (MRP's). Since the pointing attitude is in terms of a rotation matrix, \mathbf{R}^{ri} , it is coordinate independent. As a result, the attitude can be represented using any set of attitude parameters other than MRP's. The problem is to determine \mathbf{R}^{ri} and since \mathbf{R}^{oi} is known, the real problem is to determine \mathbf{R}^{ro} while satisfying the yaw-steering condition:

$$\mathbf{s}^T \mathbf{p} = 0 \quad (4.5)$$

From Table 4.1, it is clear that not every vector is known in all of the coordinate frames. If \mathbf{r}_s and \mathbf{a}_r were both known in \mathcal{F}_r and \mathcal{F}_o , this suggests using the TRIAD algorithm² to construct \mathbf{R}^{ro} .

The TRIAD algorithm is used to determine an approximation of the rotation matrix from one coordinate system to another by constructing a rotation matrix from vectors in different coordinate frames. The word TRIAD can be thought of as the word “triad,” or an acronym for TRIaxial Attitude Determination. Let \mathbf{w}_{1x} and \mathbf{w}_{2x} represent the column vectors in some coordinate system \mathcal{F}_x , and let \mathbf{v}_{1y} and \mathbf{v}_{2y} denote the column vectors in some other coordinate frame \mathcal{F}_y . Mathematically, the TRIAD algorithm for constructing the base vectors for \mathcal{F}_x becomes

$$\hat{\mathbf{r}}_{1x} = \mathbf{w}_{1x} / \|\mathbf{w}_{1x}\| \quad (4.6)$$

$$\hat{\mathbf{r}}_{2x} = \frac{\hat{\mathbf{r}}_{1x}^\times \mathbf{w}_{2x}}{\|\hat{\mathbf{r}}_{1x}^\times \mathbf{w}_{2x}\|} \quad (4.7)$$

$$\hat{\mathbf{r}}_{3x} = \hat{\mathbf{r}}_{1x}^\times \hat{\mathbf{r}}_{2x} \quad (4.8)$$

$$\mathbf{R}^{rx} = [\hat{\mathbf{r}}_{1x} \hat{\mathbf{r}}_{2x} \hat{\mathbf{r}}_{3x}]^T \quad (4.9)$$

and for \mathcal{F}_y , the base unit vectors become

$$\hat{\mathbf{r}}_{1y} = \mathbf{v}_{1y} / \|\mathbf{v}_{1y}\| \quad (4.10)$$

$$\hat{\mathbf{r}}_{2y} = \frac{\hat{\mathbf{r}}_{1y}^\times \mathbf{V}_{2y}}{\|\hat{\mathbf{r}}_{1y}^\times \mathbf{V}_{2y}\|} \quad (4.11)$$

$$\hat{\mathbf{r}}_{3y} = \hat{\mathbf{r}}_{1y}^\times \hat{\mathbf{r}}_{2y} \quad (4.12)$$

$$\mathbf{R}^{yr} = [\hat{\mathbf{r}}_{1y} \hat{\mathbf{r}}_{2y} \hat{\mathbf{r}}_{3y}] \quad (4.13)$$

where r is a dummy variable used to represent the base unit vectors in each frame. The rotation matrix \mathbf{R} that transforms a vector from \mathcal{F}_{x_1} to \mathcal{F}_y is constructed as:

$$\mathbf{R}^{yx} = \mathbf{R}^{yr} \mathbf{R}^{rx} \quad (4.14)$$

If two vectors are known in two coordinate frames, the TRIAD algorithm provides a conceptually simple way of constructing the desired rotation matrix.

However, the TRIAD algorithm can not be used here to compute \mathbf{R}^{ro} , or \mathbf{R}^{ri} directly since \mathbf{r}_s is not known in \mathcal{F}_r . Any two vectors, known both in \mathcal{F}_r and \mathcal{F}_o , could also be used in the TRIAD algorithm to construct \mathbf{R}^{ro} . Table 4.1 illustrates that there are no two vectors that are immediately known in these two coordinate frames. The TRIAD method can be used to construct each component that leads to the rotation matrix \mathbf{R}^{ro} . As a result, two intermediate frames, \mathcal{F}_a and \mathcal{F}_c , are defined below that allows us to use the TRIAD algorithm with the known vectors in Table 4.1. This leads to the construction of the rotation matrices \mathbf{R}^{ao} and \mathbf{R}^{rc} , which are used in determining \mathbf{R}^{ro} .

Once \mathbf{R}^{ro} is known, the target pointing attitude of the virtual spacecraft is then given by the product of the following rotation matrices:

$$\mathbf{R}^{ri} = \mathbf{R}^{ro} \mathbf{R}^{oi} \quad (4.15)$$

where the rotation matrix component, \mathbf{R}^{oi} , is constructed from the known orbit of the spacecraft and hence, the inertial position and velocity vectors, \mathbf{r}_{si} and \mathbf{v}_{si} , respectively. These vectors are used in the TRIAD algorithm to construct \mathbf{R}^{oi} as

$$\hat{\mathbf{o}}_{3i} = \frac{\mathbf{r}_{si}}{\|\mathbf{r}_{si}\|} \quad (4.16)$$

$$\hat{\mathbf{o}}_{2i} = \frac{\mathbf{r}_{si}^\times \mathbf{v}_{si}}{\|\mathbf{r}_{si}^\times \mathbf{v}_{si}\|} \quad (4.17)$$

$$\hat{\mathbf{o}}_{1i} = \frac{\hat{\mathbf{o}}_{2i}^\times \hat{\mathbf{o}}_{3i}}{\|\hat{\mathbf{o}}_{2i}^\times \hat{\mathbf{o}}_{3i}\|} \quad (4.18)$$

$$\mathbf{R}^{oi} = [\hat{\mathbf{o}}_{1i} \ \hat{\mathbf{o}}_{2i} \ \hat{\mathbf{o}}_{3i}]^T \quad (4.19)$$

This form of \mathbf{R}^{oi} can also be used for elliptical orbits because the variation of \mathbf{r}_s for an elliptical orbit is accounted for in Eqs. (4.16)–(4.18). We use \mathbf{R}^{oi} to rotate the sun vector and boresight vector into the orbital frame. We calculate \mathbf{a} in the inertial frame using

$$\mathbf{a}_i = \frac{\mathbf{r}_{t/si}}{\|\mathbf{r}_{t/si}\|} \quad (4.20)$$

This expression is then rotated into \mathcal{F}_o for use in constructing \mathbf{R}^{ao} .

Before calculating \mathbf{R}^{ao} , we must define the two intermediate frames \mathcal{F}_a and \mathcal{F}_c . The coordinate frame, \mathcal{F}_a , is a “body-carried” frame, centered at the virtual spacecraft’s center of mass, that rotates relative to the orbital frame. Frame “a” differs from \mathcal{F}_o since $\hat{\mathbf{a}}_1$ points at the target, whereas $\hat{\mathbf{o}}_1$ points in the direction of the spacecraft’s velocity vector. Frame “a” is related to orbital frame through the sun vector, \mathbf{s} , and the boresight axis, \mathbf{a} . Because these two vectors are known in \mathcal{F}_o , we use them in the TRIAD method to construct \mathbf{R}^{ao} . Figure 4.3 shows how \mathbf{s} and \mathbf{a} are defined in \mathcal{F}_a . Note that \mathcal{F}_a is defined so that \mathbf{s} lies in the 1-2 plane of \mathcal{F}_a . This feature is used to simplify the yaw-steering maneuver, presented below, which relates \mathcal{F}_a to \mathcal{F}_c .

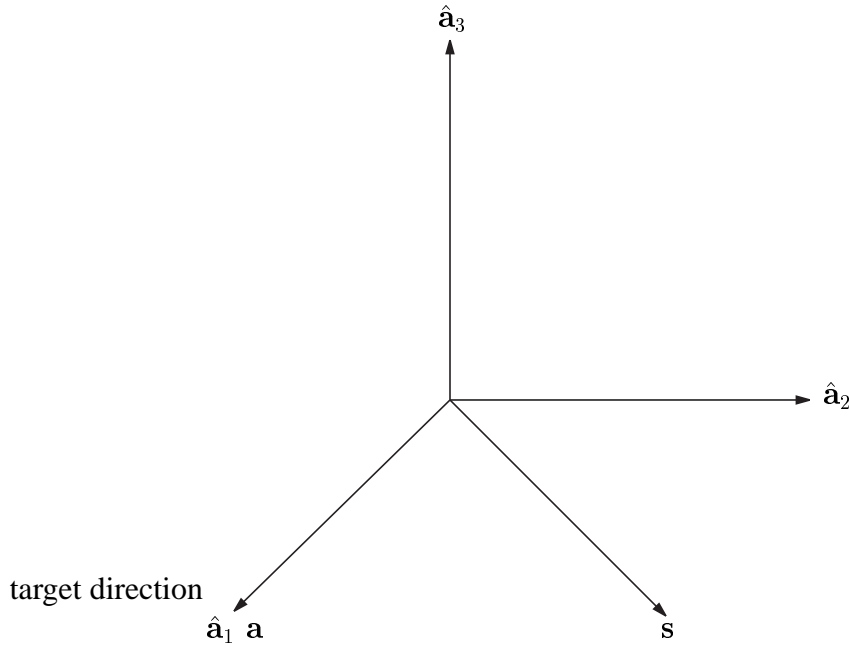


Figure 4.3: Intermediate Reference Frame “a”

The frame, \mathcal{F}_c , is a body frame centered at the virtual spacecraft’s center of mass. Since \mathcal{F}_c is fixed relative to \mathcal{F}_r , it does not rotate with respect to \mathcal{F}_r . Frame “c” differs from \mathcal{F}_r only by a rotation about the boresight axis, \mathbf{a} , since \mathbf{a} is defined to lie along $\hat{\mathbf{r}}_1$. As a result, these two frames are related by the known vectors \mathbf{p} and \mathbf{a} . Figure 4.4 illustrates how \mathbf{p} and \mathbf{a} are defined in \mathcal{F}_a . As in the case of the sun vector, \mathbf{p} also lies in the 1-2 plane of \mathcal{F}_c , which also simplifies the yaw-steering maneuver.

Using \mathcal{F}_a and \mathcal{F}_c , along with \mathcal{F}_b , \mathcal{F}_o , and \mathcal{F}_i , allows us to use the TRIAD algorithm to construct

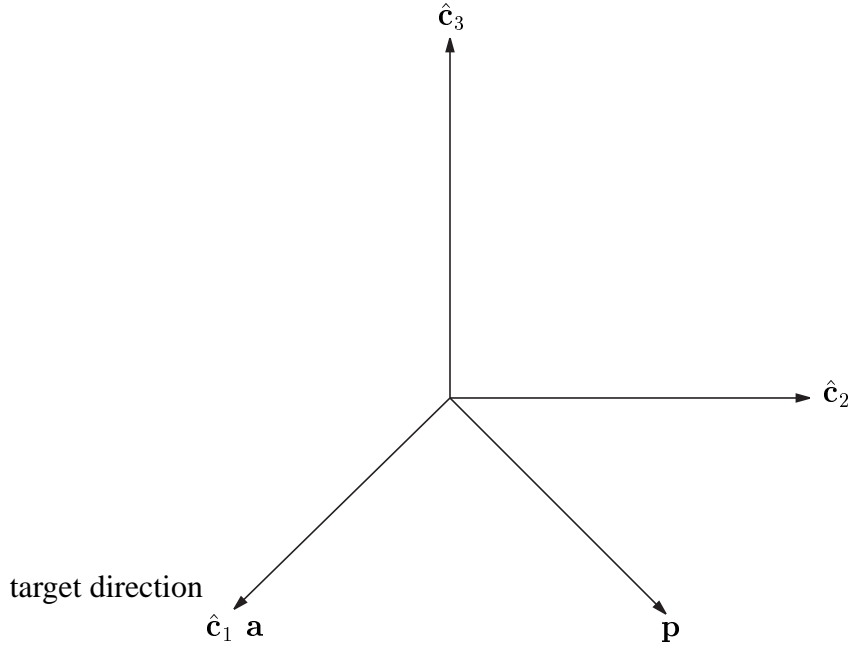


Figure 4.4: Intermediate Reference Frame “c”

the attitude in the following form:

$$\mathbf{R}^{ro} = \mathbf{R}^{rc} \mathbf{R}^{ca} \mathbf{R}^{ao} \quad (4.21)$$

Since we know the vectors \mathbf{a} and \mathbf{p} in \mathcal{F}_r , and \mathbf{s} and \mathbf{a} in \mathcal{F}_o , we solve for \mathbf{R}^{ri} by approaching the expression in Eq. (4.21) from the right and the left side until we reach \mathbf{R}^{ca} . Starting on the right side, \mathbf{R}^{ao} is constructed using Eqs. (4.7)–(4.9):

$$\hat{\mathbf{a}}_{1o} = \mathbf{a}_o \quad (4.22)$$

$$\hat{\mathbf{a}}_{3o} = \frac{\mathbf{a}_o^\times \mathbf{s}_o}{\|\mathbf{a}_o^\times \mathbf{s}_o\|} \quad (4.23)$$

$$\hat{\mathbf{a}}_{2o} = \hat{\mathbf{a}}_{3o}^\times \hat{\mathbf{a}}_{1o} \quad (4.24)$$

$$\mathbf{R}^{ao} = [\hat{\mathbf{a}}_{1o} \ \hat{\mathbf{a}}_{2o} \ \hat{\mathbf{a}}_{3o}]^T \quad (4.25)$$

We then proceed from the left side of Eq. (4.21) and construct the rotation matrix \mathbf{R}^{rc} in the same way as \mathbf{R}^{ao} :

$$\hat{\mathbf{c}}_{1r} = \mathbf{a}_r \quad (4.26)$$

$$\hat{\mathbf{c}}_{3r} = \frac{\mathbf{a}_b^\times \mathbf{p}_r}{\|\mathbf{a}_r^\times \mathbf{p}_r\|} \quad (4.27)$$

$$\hat{\mathbf{c}}_{2r} = \hat{\mathbf{c}}_{3r}^\times \hat{\mathbf{c}}_{1r} \quad (4.28)$$

$$\mathbf{R}^{rc} = [\hat{\mathbf{c}}_{1r} \ \hat{\mathbf{c}}_{2r} \ \hat{\mathbf{c}}_{3r}] \quad (4.29)$$

We now know all of the rotation matrices in Eq. (4.21) up to \mathbf{R}^{ca} . We use \mathbf{R}^{ca} to perform the yaw-steering maneuver, so that the sun vector is perpendicular to the solar panel axis.

To determine the rotation matrix from \mathcal{F}_c to \mathcal{F}_a , we use the prescribed orthogonality condition between the sun vector and the solar panel axis. Equation (4.5) can be expressed as

$$\mathbf{s}_a^T \mathbf{R}^{ac} \mathbf{p}_c = 0 \quad (4.30)$$

The matrix \mathbf{R}^{ca} is not constructed using a the TRIAD method, but from a careful analysis of the kinematics that result from definition of \mathcal{F}_c and \mathcal{F}_a . The rotation matrix \mathbf{R}^{ac} is first defined as the dot product between the base unit vectors of \mathcal{F}_a and \mathcal{F}_c , which can be expressed as

$$\mathbf{R}^{ac} = \begin{bmatrix} \hat{\mathbf{a}}_1 \cdot \hat{\mathbf{c}}_1 & \hat{\mathbf{a}}_1 \cdot \hat{\mathbf{c}}_2 & \hat{\mathbf{a}}_1 \cdot \hat{\mathbf{c}}_3 \\ \hat{\mathbf{a}}_2 \cdot \hat{\mathbf{c}}_1 & \hat{\mathbf{a}}_2 \cdot \hat{\mathbf{c}}_2 & \hat{\mathbf{a}}_2 \cdot \hat{\mathbf{c}}_3 \\ \hat{\mathbf{a}}_3 \cdot \hat{\mathbf{c}}_1 & \hat{\mathbf{a}}_3 \cdot \hat{\mathbf{c}}_2 & \hat{\mathbf{a}}_3 \cdot \hat{\mathbf{c}}_3 \end{bmatrix} \quad (4.31)$$

Recall that, from Figs. 4.3 and 4.4, we defined \mathbf{a} to lie along the “1” direction. Clearly, $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{c}}_1$ are the same vector; therefore $\hat{\mathbf{a}}_1 \cdot \hat{\mathbf{c}}_1 = 1$. By definition, the unit vectors $\hat{\mathbf{a}}_2$ and $\hat{\mathbf{a}}_3$ are perpendicular

to $\hat{\mathbf{a}}_1$, so they are also perpendicular to $\hat{\mathbf{c}}_1$. The same is true for $\hat{\mathbf{a}}_1$, which is also perpendicular to $\hat{\mathbf{c}}_2$ and $\hat{\mathbf{c}}_3$. As a result, \mathbf{R}^{ac} is a “1” rotation and Eq. (4.30) becomes

$$\mathbf{R}^{ac} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_{ac} & \sin \theta_{ac} \\ 0 & \Leftrightarrow \sin \theta_{ac} & \cos \theta_{ac} \end{bmatrix} \quad (4.32)$$

where θ_{ac} is the angle that will satisfy Equation (4.30), which can also be written as

$$\mathbf{s}_a^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_{ac} & \sin \theta_{ac} \\ 0 & \Leftrightarrow \sin \theta_{ac} & \cos \theta_{ac} \end{bmatrix} \mathbf{p}_c = 0 \quad (4.33)$$

Recall that we have defined \mathcal{F}_a and \mathcal{F}_c so that s_{3a} and p_{3c} are zero, and Eq. (4.33) expands as

$$s_{1a}p_{1c} + s_{2a}p_{2c}\cos\theta_{ac} = 0 \quad (4.34)$$

The angle θ_{ac} , which satisfies the yaw-steering condition, is

$$\cos \theta_{ac} = \Leftrightarrow \frac{s_{1a}p_{1c}}{s_{2a}p_{2c}} \quad (4.35)$$

If s_{3a} and p_{3c} were not zero, Eq. (4.35) would be a transcendental equation involving both $\cos \theta_{ac}$ and $\sin \theta_{ac}$ terms. Solving this equation would require an iterative method, such as Newton-Raphson, to determine the yaw-steering angle. The rotation matrix \mathbf{R}^{ac} is then calculated by Eq. (4.32). The ideal target pointing attitude \mathbf{R}^{ri} is then constructed by multiplying together the rotation matrices found in Eq. (4.15).

It would seem that the yaw-steering condition allows for a fully automated way of meeting the solar array power requirements while tracking a target. However, this is only true when the

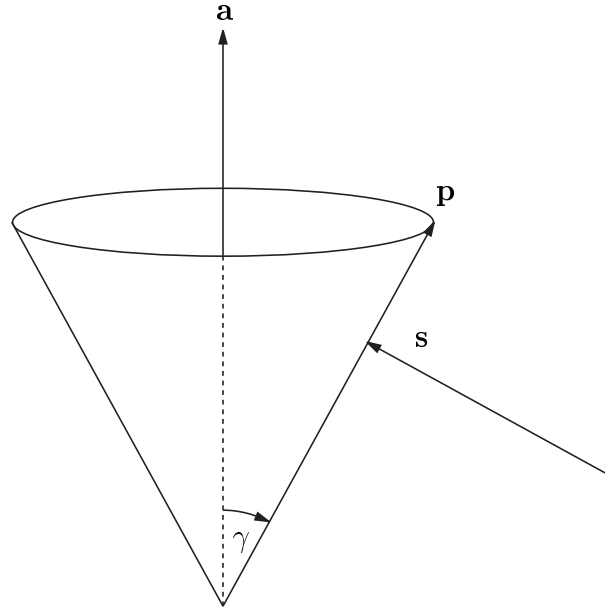


Figure 4.5: Graphical Illustration of the Yaw-Steering Maneuver

sensor axis \mathbf{a} is perpendicular to the panel vector \mathbf{p} . For certain panel orientations, Eq. (4.35) has numerical singularities associated with it. When the panel vector is not perpendicular to the boresight axis, the panel vector traces out an imaginary cone during the yaw-steering maneuver, as shown in Fig. 4.5. The singularity occurs when the sun vector can not be made perpendicular to the cone surface. Looking at Fig. 4.5, it is obvious that the yaw-steering maneuver can not be performed when either \mathbf{s} or \mathbf{p} is aligned with \mathbf{a} , which is evident in Eq. (4.35) where $\cos \theta_{ac} \rightarrow \infty$ as either $s_{2a} \rightarrow 0$ or $p_{2c} \rightarrow 0$.

We examine Eq. (4.35) for the cases where $0 < \gamma \leq 90^\circ$ in Fig. 4.5 to determine when singularities occur. Clearly, the limiting case for Eq. (4.35) is when $|\cos \theta_{ac}| = 1$, and singularities appear when the right side of Eq. (4.35) exceeds this value. No singularities will occur if the right

hand side is less than unity:

$$\left\| \frac{s_{1a}p_{1c}}{s_{2a}p_{2c}} \right\| < 1 \quad (4.36)$$

which can also be written as

$$\left\| \frac{s_{1a}}{s_{2a}} \right\| < \left\| \frac{p_{2c}}{p_{1c}} \right\| \quad (4.37)$$

In short, the yaw-steering maneuver can be performed if either $s_{1a} < s_{2a}$, $p_{1c} < p_{2c}$, or both are satisfied. The yaw-steering maneuver can always be performed if the \mathbf{a} is perpendicular to \mathbf{p} , which says that $p_{1c} = 0$. As a result, the yaw-steering maneuver is always a 90° rotation about the boresight axis.

Once we know \mathbf{R}^{ri} , we can determine the reference attitude, σ_r , using Eq. (3.18). The vectors in Table 4.1 can now be expressed in all of the coordinate frames. We proceed to develop in the next section the required angular velocity that is necessary to track a target using the rotation matrices developed for each frame.

4.2 The Ideal Angular Velocity

We develop angular velocity commands by differentiating the attitude expressions in the previous section and then computing ω_r^{ri} from the vector sum of each intermediate angular velocity expression.

We begin by differentiating Eq. (4.2) to get

$$\dot{\mathbf{r}}_{t/s} = \dot{\mathbf{r}}_t \Leftrightarrow \dot{\mathbf{r}}_s \quad (4.38)$$

where $\dot{\mathbf{r}}_s$ is simply the known spacecraft velocity vector and $\dot{\mathbf{r}}_t$ is given by $\boldsymbol{\omega}_e^\times \mathbf{r}_t$, where $\boldsymbol{\omega}_e$ is the angular velocity of the earth.

We begin by defining the angular velocity of \mathcal{F}_o with respect to \mathcal{F}_i . Since we are assuming that the orbit is circular, the angular velocity is known, which is just the mean motion of the orbit expressed about the negative orbit normal ($\Leftrightarrow \hat{\mathbf{o}}_2$) as

$$\boldsymbol{\omega}_o^{oi} = \begin{bmatrix} 0 & \Leftrightarrow (\sqrt{\mu / \|\mathbf{r}_s^T \mathbf{r}_s\|^3}) & 0 \end{bmatrix}^T \quad (4.39)$$

where the superscript oi denotes the angular velocity of \mathcal{F}_o with respect to \mathcal{F}_i .

Since \mathcal{F}_a and \mathcal{F}_o change with time, we need to calculate the angular velocity $\boldsymbol{\omega}_a^{ao}$. It is not difficult to show⁵ that $\boldsymbol{\omega}_a^{ao}$ can be calculated based on the differentiation of \mathbf{R}^{ao} as

$$\boldsymbol{\omega}_a^{ao\times} = \Leftrightarrow \overset{\circ}{\mathbf{R}}^{ao} \mathbf{R}^{oa} \quad (4.40)$$

where “ \circ ” denotes differentiation with respect to a moving coordinate frame. The matrix $\overset{\circ}{\mathbf{R}}^{ao}$ is found by first rewriting Eqs. (4.22)-(4.24) as

$$D_1 \hat{\mathbf{a}}_{1o} = \mathbf{r}_{t/so} \quad (4.41)$$

$$D_2 \hat{\mathbf{a}}_{3o} = \mathbf{r}_{t/so}^\times \mathbf{s}_o \quad (4.42)$$

$$\hat{\mathbf{a}}_{2o} = \hat{\mathbf{a}}_{3o}^\times \hat{\mathbf{a}}_{1o} \quad (4.43)$$

where the scalar quantities D_1 and D_2 are given by $\|\mathbf{r}_{t/so}\|$ and $\|\mathbf{r}_{t/so}^\times \mathbf{s}_o\|$, respectively. Differentiating the above equations with respect to time results in the following

$$\dot{\hat{\mathbf{a}}}_{1o} = \frac{\dot{\mathbf{r}}_{t/so} \Leftrightarrow \dot{D}_1 \hat{\mathbf{a}}_{1o}}{D_1} \quad (4.44)$$

$$\dot{\hat{\mathbf{a}}}_{3o} = \frac{\dot{\mathbf{r}}_{t/so}^\times \mathbf{s}_o \Leftrightarrow \dot{D}_2 \hat{\mathbf{a}}_{3o}}{D_2} \quad (4.45)$$

$$\dot{\hat{\mathbf{a}}}_{2o} = \dot{\hat{\mathbf{a}}}_{3o}^\times \hat{\mathbf{a}}_{1o} + \hat{\mathbf{a}}_{3o}^\times \dot{\hat{\mathbf{a}}}_{1o} \quad (4.46)$$

which have been simplified by assuming that the sun vector, calculated in Appendix A, slowly varies in the inertial frame, and can be considered constant in \mathcal{F}_i . As a result, $\dot{\mathbf{s}}_o$ is zero. The rates of change of D_1 and D_2 are

$$\dot{D}_1 = \frac{\dot{\mathbf{r}}_{t/so}^T \mathbf{r}_{t/so}}{D_1} \quad (4.47)$$

$$\dot{D}_2 = \frac{(\dot{\mathbf{r}}_{t/so}^\times \mathbf{s}_o)^T (\mathbf{r}_{t/so}^\times \mathbf{s}_o)}{D_2} \quad (4.48)$$

These derivatives have been computed with respect to \mathcal{F}_i . We need the derivatives with respect to the moving orbital frame, which we find using Eq. (3.3):

$$\overset{\circ}{\hat{\mathbf{a}}}_{1o} = \dot{\hat{\mathbf{a}}}_{1o} \Leftrightarrow \boldsymbol{\omega}_o^{oi \times} \mathbf{a}_{1o} \quad (4.49)$$

$$\overset{\circ}{\hat{\mathbf{a}}}_{3o} = \dot{\hat{\mathbf{a}}}_{3o} \Leftrightarrow \boldsymbol{\omega}_o^{oi \times} \mathbf{a}_{3o} \quad (4.50)$$

$$\overset{\circ}{\hat{\mathbf{a}}}_{2o} = \overset{\circ}{\hat{\mathbf{a}}}_{3o}^\times \hat{\mathbf{a}}_{1o} + \hat{\mathbf{a}}_{3o}^\times \overset{\circ}{\hat{\mathbf{a}}}_{1o} \quad (4.51)$$

and then the rate of change of \mathbf{R}^{ao} with respect to \mathcal{F}_o is simply

$$\overset{\circ}{\mathbf{R}}^{ao} = \begin{bmatrix} \overset{\circ}{\hat{\mathbf{a}}}_{1o} & \overset{\circ}{\hat{\mathbf{a}}}_{2o} & \overset{\circ}{\hat{\mathbf{a}}}_{3o} \end{bmatrix}^T \quad (4.52)$$

Using Eq. (4.40), the angular velocity $\boldsymbol{\omega}_a^{ai}$ is then found to be

$$\boldsymbol{\omega}_a^{ai} = \boldsymbol{\omega}_a^{ao} + \mathbf{R}^{ao} \boldsymbol{\omega}_o^{oi} \quad (4.53)$$

It can be seen from Eq. (4.33) that $\boldsymbol{\omega}_a^{ca}$ is simply

$$\boldsymbol{\omega}_a^{ca} = \begin{bmatrix} \Leftrightarrow \dot{\theta}_{ac} & 0 & 0 \end{bmatrix}^T \quad (4.54)$$

and $\dot{\theta}_{ac}$ is found by taking a time derivative of Eq.(4.34) to yield

$$\dot{\theta}_{ac} = \frac{\dot{s}_{1a}p_{1c} + \dot{s}_{2a}p_{2c} \cos \theta_{ac}}{s_{2a}p_{2c} \sin \theta_{ac}} \quad (4.55)$$

where the derivative of the sun vector with respect to \mathcal{F}_a is given by

$$\overset{\circ}{\mathbf{s}}_a = \Leftrightarrow \boldsymbol{\omega}_a^{ai \times} \mathbf{s}_a \quad (4.56)$$

The frame \mathcal{F}_c has a fixed orientation with respect to \mathcal{F}_r , so $\boldsymbol{\omega}_r^{rc} = \mathbf{0}$. As a result, the desired tracking body rate vector $\boldsymbol{\omega}_r^{ri}$ is constructed by adding Eqs. (4.53) and (4.54), and then rotating them into \mathcal{F}_r :

$$\boldsymbol{\omega}_r^{ri} = \mathbf{R}^{ra} (\boldsymbol{\omega}_a^{ca} + \boldsymbol{\omega}_a^{ai}) \quad (4.57)$$

where \mathbf{R}^{ra} is the rotation matrix from \mathcal{F}_a to \mathcal{F}_r and is found from the previous section to be

$$\mathbf{R}^{ra} = \mathbf{R}^{rc} [\mathbf{R}^{ac}]^T \quad (4.58)$$

Once the ideal angular velocity is computed, the target tracking trajectory with yaw-steering is known. To make the virtual spacecraft follow this trajectory, momentum wheels are used for maneuvering. The next section develops the angular acceleration commands that are used in determining the virtual spacecraft's control torque.

4.3 The Ideal Angular Acceleration

Once the angular velocities are known in each of the coordinate frames, we compute the desired angular accelerations. The accelerations are needed to compute the reference axial wheel torque

\mathbf{g}_{ar} to generate the desired trajectory. The angular acceleration $\dot{\omega}_r^{ri}$ is constructed analogously to the angular velocity ω_r^{ri} by differentiating Eqs. (4.38)–(4.56). The acceleration commands are found by taking a time derivative of Eq. (4.38)

$$\ddot{\mathbf{r}}_{t/s} = \ddot{\mathbf{r}}_t \Leftrightarrow \ddot{\mathbf{r}}_s \quad (4.59)$$

where $\ddot{\mathbf{r}}_s$ is simply the two-body equation of motion defined in Chapter 3. The inertial acceleration of the target $\ddot{\mathbf{r}}_{ti}$ is

$$\ddot{\mathbf{r}}_{ti} = \omega_e^\times \dot{\mathbf{r}}_{ti} \quad (4.60)$$

where $\ddot{\mathbf{r}}_{ti}^{\circ\circ}$ is zero since \mathcal{F}_i does not rotate.

After defining the accelerations of the position vectors, we start determining what the angular accelerations of the coordinate frames. The angular acceleration $\dot{\omega}_o^{oi}$ is found by differentiating Eq. (4.39) to get

$$\dot{\omega}_o^{oi} = \begin{bmatrix} 0 & 1.5 \left(\dot{\mathbf{r}}_s^T \mathbf{r}_s + \mathbf{r}_s^T \dot{\mathbf{r}}_s \right) \left(\sqrt{\mu / \|\mathbf{r}_s^T \mathbf{r}_s\|^5} \right) & 0 \end{bmatrix}^T \quad (4.61)$$

This equation is only true when the orbit is elliptical, since \mathbf{r}_s varies as the spacecraft moves in an elliptical orbit. However, we are using a circular orbit for this thesis, so Eq. (4.61) is zero because $\|\mathbf{r}_s\|$ is constant.

The next component that is needed is the angular acceleration of \mathcal{F}_a with respect to \mathcal{F}_o . The angular acceleration $\dot{\omega}_a^{ao}$ is found by differentiating Eq. (4.40) which yields:

$$\dot{\omega}_a^{ao\times} = \Leftrightarrow \ddot{\mathbf{R}}^{ao} \mathbf{R}^{oa} \Leftrightarrow \dot{\mathbf{R}}^{ao} \dot{\mathbf{R}}^{oa} \quad (4.62)$$

The second derivative of \mathbf{R}^{ao} is found by differentiating Eqs. (4.49)–(4.51) to get

$$\ddot{\mathbf{a}}_{1o}^{\circ\circ} = \ddot{\mathbf{a}}_{1o} \Leftrightarrow 2\omega_o^{oi\times} \dot{\mathbf{a}}_{1o} \Leftrightarrow \dot{\omega}_o^{oi\times} (\omega_o^{oi\times} \mathbf{a}_{1o}) \Leftrightarrow \dot{\omega}_o^{oi\times} \mathbf{a}_{1o} \quad (4.63)$$

$$\overset{\circ\circ}{\hat{\mathbf{a}}}_{3o} = \ddot{\mathbf{a}}_{3o} \Leftrightarrow 2\omega_o^{oi\times} \dot{\mathbf{a}}_{3o} \Leftrightarrow \omega_o^{oi\times} (\omega_o^{oi\times} \mathbf{a}_{3o}) \Leftrightarrow \dot{\omega}_o^{oi\times} \mathbf{a}_{3o} \quad (4.64)$$

$$\overset{\circ\circ}{\hat{\mathbf{a}}}_{2o} = \overset{\circ\circ\times}{\hat{\mathbf{a}}}_{3o} \overset{\circ}{\hat{\mathbf{a}}}_{1o} + 2 \overset{\circ\times}{\hat{\mathbf{a}}}_{3o} \overset{\circ}{\hat{\mathbf{a}}}_{1o} + \overset{\circ\circ}{\hat{\mathbf{a}}}_{3o}^\times \overset{\circ\circ}{\hat{\mathbf{a}}}_{1o} \quad (4.65)$$

$$\overset{\circ\circ}{\mathbf{R}}^{ao} = \begin{bmatrix} \overset{\circ\circ}{\hat{\mathbf{a}}}_{1o} & \overset{\circ\circ}{\hat{\mathbf{a}}}_{2o} & \overset{\circ\circ}{\hat{\mathbf{a}}}_{3o} \end{bmatrix}^T \quad (4.66)$$

where the inertial derivatives of the unit vectors are found by differentiating Eqs. (4.44)–(4.45):

$$\ddot{\mathbf{a}}_{1o} = \frac{\ddot{\mathbf{r}}_{t/so} \Leftrightarrow 2\dot{D}_1 \dot{\mathbf{a}}_{1o} \Leftrightarrow \ddot{D}_1 \hat{\mathbf{a}}_{1o}}{D_1} \quad (4.67)$$

$$\ddot{\mathbf{a}}_{3o} = \frac{\ddot{\mathbf{r}}_{t/so}^\times \mathbf{s}_o \Leftrightarrow 2\dot{D}_2 \dot{\mathbf{a}}_{3o} \Leftrightarrow \ddot{D}_2 \hat{\mathbf{a}}_{3o}}{D_2} \quad (4.68)$$

$$\ddot{D}_1 = \frac{(\ddot{\mathbf{r}}_{t/so} \cdot \mathbf{r}_{t/so}) + (\dot{\mathbf{r}}_{t/so} \cdot \dot{\mathbf{r}}_{t/so}) \Leftrightarrow \dot{D}_1^2}{D_1} \quad (4.69)$$

$$\ddot{D}_2 = \frac{(\ddot{\mathbf{r}}_{t/so}^\times \mathbf{s}_o) \cdot (\mathbf{r}_{t/so}^\times \mathbf{s}_o) + (\dot{\mathbf{r}}_{t/so}^\times \mathbf{s}_o) \cdot (\dot{\mathbf{r}}_{t/so}^\times \mathbf{s}_o) \Leftrightarrow \dot{D}_2^2}{D_2} \quad (4.70)$$

and then $\dot{\omega}_a^{ai}$ becomes

$$\dot{\omega}_a^{ai} = \overset{\circ}{\omega}_a^{ao} + \mathbf{R}^{ao} \dot{\omega}_o^{oi} \quad (4.71)$$

Likewise, $\overset{\circ}{\omega}_a^{ca}$ is found by differentiating the expression in Eq. (4.54) where

$$\ddot{\theta}_{ac} = \frac{\ddot{s}_{1a} p_{1c} \Leftrightarrow 2\dot{\theta}_{ac} \dot{s}_{2a} p_{2c} \sin \theta_{ac} \Leftrightarrow \dot{\theta}_{ac}^2 s_{2a} p_{2c} \cos \theta_{ac} + \ddot{s}_{2a} p_{2c} \cos \theta_{ac}}{s_{2a} p_{2c} \sin \theta_{ac}} \quad (4.72)$$

and the acceleration of the sun vector with respect to \mathcal{F}_a is given by

$$\overset{\circ\circ}{\mathbf{s}}_a = \Leftrightarrow 2\omega_a^{ai\times} \overset{\circ}{\mathbf{s}}_a \Leftrightarrow \omega_a^{ai\times} (\omega_a^{ai\times} \mathbf{s}_a) \quad (4.73)$$

The desired angular acceleration becomes

$$\dot{\omega}_r^{ri} = \mathbf{R}^{ra} (\overset{\circ}{\omega}_a^{ca} + \dot{\omega}_a^{ai}) \quad (4.74)$$

where as before, \mathbf{R}^{ra} is the rotation matrix from \mathcal{F}_a to \mathcal{F}_r .

Like the desired angular velocity vector ω_r^{ri} , the desired acceleration vector $\dot{\omega}_r^{ri}$ is constructed from knowing $\overset{\circ}{\omega}_a^{ca}$ and $\dot{\omega}_a^{ai}$ in \mathcal{F}_r using Eq. (4.58). Having now found \mathbf{R}^{ri} , ω_r^{ri} , and $\dot{\omega}_r^{ri}$, we can completely describe the desired trajectory that the real spacecraft needs to obtain in order to track a target. In the next chapter, we show how this open-loop reference trajectory is used in the derivation of a control law that will asymptotically drive any initial tracking errors in the body frame to zero.

Chapter 5

Control Laws

To track a target, we develop a nonlinear feedback control law that allows the actual spacecraft to perform a rotational maneuver to reorient itself with the attitude and angular velocities presented in Chapter 4. The spacecraft uses momentum wheels to generate control torques to perform the tracking maneuver. The control law is derived using Lyapunov stability and control theory for the wheel control torque, \mathbf{g}_a . In addition to providing control torques, the control law in Section 5.2 compensates for the effects of the gravity gradient torque. Like the controllers developed in Ref. 15, this controller asymptotically stabilizes the tracking errors in the attitude and angular velocity. The definitions of these errors are presented in Section 5.1, which are used, along with the state equations, in the development of the controller.

5.1 Error Kinematics

At the start of a rotational tracking maneuver, the actual body frame and the reference frame are different. This difference is termed the tracking error between the two frames. Tracking errors between \mathcal{F}_b and \mathcal{F}_r can be calculated in terms of the attitude, angular velocity, angular acceleration, angular momenta, and control torques. We only consider the attitude error and the angular velocity error for the derivation of our control law. The attitude tracking error is constructed from the rotation matrix assembled from the actual body frame attitude vector and the rotation matrix assembled from the ideal pointing attitude. The attitude error is defined by

$$\mathbf{R}^{br}(\delta\sigma) = \mathbf{R}^{bi}(\sigma_b)\mathbf{R}^{ir}(\sigma_r) \quad (5.1)$$

where $\mathbf{R}^{br}(\delta\sigma)$ is the rotation matrix from the reference frame \mathcal{F}_r to the body frame \mathcal{F}_b , and $\delta\sigma$ is the error in the attitude between the frames \mathcal{F}_b and \mathcal{F}_r .

The error in the angular velocity is simply the difference of the actual spacecraft's and the virtual spacecraft's angular velocities. The tracking error of the angular velocity is expressed in \mathcal{F}_b as

$$\delta\omega = \omega_b \Leftrightarrow \mathbf{R}^{br}(\delta\sigma)\omega_r \quad (5.2)$$

Using Eq. (3.19), the differential equation for the error kinematics becomes

$$\delta\dot{\sigma} = \mathbf{G}(\delta\sigma)\delta\omega \quad (5.3)$$

As pointed out in Section 3.1.2, there are multiple ways of computing the attitude and attitude error vectors. In the actual implementation, computational expenses are reduced by using Eq. (5.1)

instead of Eq. (5.3) to determine $\delta\sigma$. As mentioned earlier, Eq. (5.3) is important for the derivation of the control law in the next section.

5.2 The Lyapunov Tracking Controller

The feedback momentum wheel controller is derived from Lyapunov's direct method. We use the following Lyapunov function candidate¹⁵

$$\begin{aligned} V &= \frac{1}{2} \delta\omega^T \mathbf{K} \delta\omega + 2k_2 \ln(1 + \delta\sigma^T \delta\sigma) \\ &= \frac{1}{2} \left(\omega_b \Leftrightarrow \mathbf{R}^{br}(\delta\sigma) \omega_r \right)^T \mathbf{K} \left(\omega_b \Leftrightarrow \mathbf{R}^{br}(\delta\sigma) \omega_r \right) + 2k_2 \ln(1 + \delta\sigma^T \delta\sigma) \end{aligned} \quad (5.4)$$

where \mathbf{K} and k_2 are positive gain constants so that V is positive definite. This Lyapunov function is also radially unbounded in terms of the tracking errors $\delta\sigma$ and $\delta\omega$. Recall that the definition of radial unboundedness from Section 2.3.1 states that $V(\mathbf{x}) \rightarrow \infty$ as $\|\mathbf{x}\| \rightarrow \infty$.

To develop the control law, we take the derivative of Eq. (5.4) to get the following equation for \dot{V} :

$$\dot{V} = \left[\dot{\omega}_b \Leftrightarrow \dot{\mathbf{R}}^{br}(\delta\sigma) \omega_r \Leftrightarrow \mathbf{R}^{br}(\delta\sigma) \dot{\omega}_r \right]^T \mathbf{K} \delta\omega + 4k_2 \frac{\delta\sigma^T \delta\dot{\sigma}}{1 + \delta\sigma^T \delta\sigma} \quad (5.5)$$

We substitute Eqs. (3.9) and (3.14) into Eq. (5.5) to get

$$\begin{aligned} \dot{V} &= \left[\mathbf{J}^{-1} \left(\mathbf{h}_b^\times \omega_b + \mathbf{g}_e \Leftrightarrow \mathbf{A} \mathbf{g}_a \right) \Leftrightarrow \omega_b^\times \delta\omega \Leftrightarrow \mathbf{R}^{br}(\delta\sigma) \mathbf{J}^{-1} \left(\mathbf{h}_r^\times \omega_r + \mathbf{g}_e \Leftrightarrow \mathbf{A} \mathbf{g}_{ar} \right) \right]^T \mathbf{K} \delta\omega \\ &\quad + 4k_2 \frac{\delta\sigma^T \delta\dot{\sigma}}{1 + \delta\sigma^T \delta\sigma} \end{aligned} \quad (5.6)$$

where we have made use of the fact proved in Ref. 15

$$\frac{d\mathbf{R}^{br}(\delta\sigma)}{dt} \omega_r = \omega_b^\times \delta\omega \quad (5.7)$$

Equation (5.3) is now substituted into the second term of Eq. (5.6) to get:

$$\begin{aligned}\dot{V} = & \left[\mathbf{J}^{-1} \left(\mathbf{h}_b^\times \boldsymbol{\omega}_b + \mathbf{g}_e \Leftrightarrow \mathbf{A} \mathbf{g}_a \right) \Leftrightarrow \boldsymbol{\omega}_b^\times \delta \boldsymbol{\omega} \Leftrightarrow \mathbf{R}^{br}(\delta \boldsymbol{\sigma}) \mathbf{J}^{-1} \left(\mathbf{h}_r^\times \boldsymbol{\omega}_r + \mathbf{g}_e \Leftrightarrow \mathbf{A} \mathbf{g}_{ar} \right) \right]^T \mathbf{K} \delta \boldsymbol{\omega} \\ & + 4k_2 \frac{\delta \boldsymbol{\sigma}^T \mathbf{G}(\delta \boldsymbol{\sigma})}{1 + \delta \boldsymbol{\sigma}^T \delta \boldsymbol{\sigma}} \delta \boldsymbol{\omega}\end{aligned}\quad (5.8)$$

The last term in Eq. (5.8) can be shown to reduce to $k_2 \delta \boldsymbol{\sigma}^T \delta \boldsymbol{\omega}$ due to a identity found in Ref. 22.

Letting $\mathbf{K} = \mathbf{J}$ in Eq. (5.8) and making use of Eq. (3.8), the final equation for the derivative of V becomes:

$$\begin{aligned}\dot{V} = & \Leftrightarrow \left[\Leftrightarrow \mathbf{h}_b^\times \mathbf{J}^{-1} (\mathbf{h}_b \Leftrightarrow \mathbf{A} \mathbf{h}_a) \Leftrightarrow \mathbf{g}_e + \mathbf{A} \mathbf{g}_a + \mathbf{J} \boldsymbol{\omega}_b^\times \delta \boldsymbol{\omega} \right. \\ & + \mathbf{J} \mathbf{R}^{br}(\delta \boldsymbol{\sigma}) \mathbf{J}^{-1} \mathbf{h}_r^\times \mathbf{J}^{-1} (\mathbf{h}_r \Leftrightarrow \mathbf{A} \mathbf{h}_{ar}) + \mathbf{J} \mathbf{R}^{br}(\delta \boldsymbol{\sigma}) \mathbf{J}^{-1} \mathbf{g}_e \\ & \left. \Leftrightarrow \mathbf{J} \mathbf{R}^{br}(\delta \boldsymbol{\sigma}) \mathbf{J}^{-1} \mathbf{A} \mathbf{g}_{ar} \Leftrightarrow k_2 \delta \boldsymbol{\sigma} \right]^T \delta \boldsymbol{\omega}\end{aligned}\quad (5.9)$$

Note that Eq. (5.9) is almost identical to the expression of \dot{V} found in Ref. 15. The difference lies in the angular momentum expression for the virtual spacecraft. Our version of \dot{V} is more complicated because the virtual spacecraft is a gyrostat, and not just a rigid body as in Ref. 15.

Having now found \dot{V} , we want to choose the control torques $\mathbf{A} \mathbf{g}_a$ so that \dot{V} is negative semi-definite. The reason for doing this is explained below. We first rewrite Eq. (5.9) as

$$\dot{V} = \Leftrightarrow (\mathbf{A} \mathbf{g}_a + \mathbf{z})^T \delta \boldsymbol{\omega} \quad (5.10)$$

where \mathbf{z} is set equal to all of the bracketed terms in Eq. (5.9) except for $\mathbf{A} \mathbf{g}_a$. The control is selected so that all of the terms in Eq. (5.9) cancel, making \dot{V} semi-negative definite in the form

$$\dot{V} = \Leftrightarrow k_1 \delta \boldsymbol{\omega}^T \delta \boldsymbol{\omega} \leq 0 \quad (5.11)$$

where k_1 is a positive gain constant. The control torque that yields the expression for \dot{V} in Eq. (5.11) becomes:

$$\mathbf{A}\mathbf{g}_a = k_1\delta\boldsymbol{\omega} \Leftrightarrow \mathbf{z} \quad (5.12)$$

After substituting in the expression for \mathbf{z} , the control torque is rewritten as:

$$\begin{aligned} \mathbf{A}\mathbf{g}_a &= \mathbf{h}_b^\times \mathbf{J}^{-1}(\mathbf{h}_b \Leftrightarrow \mathbf{A}\mathbf{h}_a) + \mathbf{g}_e \Leftrightarrow \mathbf{J}\boldsymbol{\omega}_b^\times \delta\boldsymbol{\omega} \\ &\Leftrightarrow \mathbf{J}\mathbf{R}^{br}(\delta\boldsymbol{\sigma})\mathbf{J}^{-1}\mathbf{h}_r^\times \mathbf{J}^{-1}(\mathbf{h}_r \Leftrightarrow \mathbf{A}\mathbf{h}_{ar}) \Leftrightarrow \mathbf{J}\mathbf{R}^{br}(\delta\boldsymbol{\sigma})\mathbf{J}^{-1}\mathbf{g}_e \\ &+ \mathbf{J}\mathbf{R}^{br}(\delta\boldsymbol{\sigma})\mathbf{J}^{-1}\mathbf{A}\mathbf{g}_{ar} + k_1\delta\boldsymbol{\omega} + k_2\delta\boldsymbol{\sigma} \end{aligned} \quad (5.13)$$

As found in Ref. 15, the control law in Eq. (5.13) guarantees perfect tracking, i.e., $\boldsymbol{\omega}_b(t) = \boldsymbol{\omega}_r(t)$ and $\boldsymbol{\sigma}_b(t) = \boldsymbol{\sigma}_r(t)$ for all $t \geq 0$ if the initial condition errors are zero, i.e., $\delta\boldsymbol{\omega}(0) = \delta\boldsymbol{\sigma}(0) = 0$. As a result, all of the terms on the right hand side of Eq. (5.13) containing the error terms are zero, $\mathbf{R}^{br}(\delta\boldsymbol{\sigma})$ becomes the identity matrix, and the remaining terms cancel out. The control law becomes reduces to the reference torque

$$\mathbf{A}\mathbf{g}_a = \mathbf{A}\mathbf{g}_{ar} \quad (5.14)$$

where the reference control torque is determined from Eq. (3.14).

There are two problems associated with this feedback controller. The first one, and the most important, is what happens when $\dot{V} = 0$. Looking at Eq. (5.11), it is obvious that $\dot{V} = 0$ when $\delta\boldsymbol{\omega} = \mathbf{0}$, and that the attitude error $\delta\boldsymbol{\sigma}$ has no effect whatsoever on the sign-definiteness of \dot{V} . As a result, the stability of the controller for this case can only be proved by invoking Lasalle's Theorem¹⁷ which is stated in the following form:

Let $\Omega \subset D$ be a compact (closed and bounded) set with the property that every solution of $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$ which starts in Ω remains for all future time in Ω . Let $V: \Omega \rightarrow \mathbb{R}$ be a continuously differentiable function such that $\dot{V} \leq 0$ in Ω . Let E be the set of all points in Ω where $\dot{V} = 0$. Let M be the largest invariant set in E . Then every solution starting in Ω approaches M as $t \rightarrow \infty$.

In our problem, the set of points in E where $\dot{V} = 0$ occurs when $\delta\omega = \mathbf{0}$ and when $\delta\sigma$ is not zero. The largest invariant set M is the origin, or when $\delta\sigma = \delta\omega = \mathbf{0}$. LaSalle's theorem states that the solution to the differential equations in Chapter 3, assuming that they are in the set Ω , will approach the reference trajectory when $\dot{V} = 0$. The Lyapunov function, V , spirals towards the trivial solution resulting from the controller being globally asymptotically stable. Reference 17 shows that the controller is asymptotically stable when $\dot{V} = 0$ and V is radially unbounded when the solutions to the differential equations can not remain in the set where $\delta\omega$ is zero and $\delta\sigma$ is nonzero. Maneuvering the spacecraft to acquire a target requires some angular velocity, causing some $\delta\omega$ during the maneuver.

The second difficulty arises when more than 3 wheels are modeled in the spacecraft. Looking at Eq. (5.13), the $3 \times N$ wheel position vector matrix \mathbf{A} is on the left hand side of the equation. In actual implementation of this control law, \mathbf{A} has to be inverted. However, this matrix is not square for cases involving more than three wheels, and is not invertible. Then the pseudo-inverse method¹⁸ must be used. The pseudo-inverse has three forms associated with it. The first form assumes that \mathbf{A} has linearly independent columns, then $\mathbf{A}^T \mathbf{A}$ is invertible and

$$\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \quad (5.15)$$

If \mathbf{A}^T has linearly independent columns, then $\mathbf{A}\mathbf{A}^T$ is invertible and the pseudo-inverse becomes

$$\mathbf{A}^\dagger = (\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}^T \quad (5.16)$$

When neither \mathbf{A} nor \mathbf{A}^T has independent columns, then Singular Value Decomposition⁸ has to be used to obtain \mathbf{A}^\dagger :

$$\mathbf{A}^\dagger = \mathbf{V}_t \mathbf{S}_t \mathbf{U}^T \quad (5.17)$$

where t denotes that the last $n \Leftrightarrow m$ zero columns of \mathbf{S} and \mathbf{V} have been discarded. Note that, if \mathbf{A} is invertible, then the inverse and the psuedo-inverse are the same.

We close this chapter by providing a brief summary of some of the different control schemes that have been developed using Lyapunov's direct method. Schaub *et al.*²² developed a reaction wheel controller using the Lyapunov function in Eq. (5.4). That controller is similar to Eq. (5.13), but is only globally aysmptotically stable in the absence of external torques. If there are external torques, then their controller is only Lagrange stable. The gains are selected by linearizing the error dynamics and applying the root-locus method. The reference trajectory was developed for a “bang-bang” near-minimum time eigenaxis maneuver. Hall *et al.*¹⁵ developed two types of control laws using the same Lyapunov function. The first type of control law is for reorienting the whole spacecraft during a pointing maneuver using thrusters. A second type of control law for momentum wheels is used for “fine” pointing and the elimination of tracking errors. The control laws were shown to be globally asymptotically stable. The reference trajectory comes from a virtual spacecraft that is assumed to be just a rigid body spacecraft with no momentum wheels. Finally, the feedback controller presented here uses momentum wheels for both spacecraft reorientation

and tracking. Equation (5.13) is virtually identical to the momentum wheel control law developed in Ref. 15 except that the gravity gradient torque is assumed to be the only external torque present. The control law remains globally asymptotically stable with external torques according to Lasalle's theorem. In Chapters 6 and 7, the stability of the controller is demonstrated using a numerical simulation.

Chapter 6

Tracking Simulation

We demonstrate the effectiveness of the control law in Section 5.2 by simulating a target tracking scenario with the spacecraft models and tracking algorithms presented in Chapters 3 and 4. In this chapter, we present the gyrostat model used in the simulation. The X-ray Timing Explorer (XTE) spacecraft is used as the baseline spacecraft model. Our gyrostat model is a simplified version of the XTE spacecraft because the actual dynamics of the XTE complicate a numerical simulation. We begin by first presenting an overview of the XTE spacecraft through its science objectives and its attitude control system. This leads to the simplified XTE model, and the assumptions behind it, in Section 6.3.

6.1 XTE Mission Profile

The X-ray Timing Explorer (XTE) satellite falls under the Explorer-class missions developed by NASA.³ The XTE spacecraft was designed and built by NASA's Goddard Space Flight Center, and was launched on a Delta launch vehicle in December 1995. After its launch, it was commissioned as the Rossi X-ray Timing Explorer⁴ because Bruno Rossi was the scientist who led the team that discovered the first nonsolar source of X-rays in 1962. Here, we will refer to it as the XTE. Its primary science objective was to perform astronomical measurements in the X-ray portion of the electro-magnetic spectrum.³ It was designed for a nominal 3 year mission to study the structure and dynamics of compact X-ray sources, neutron stars, white dwarfs, and other stellar objects with X-ray energy emissions. To study these sources, three X-ray instruments were placed on the XTE. These include the Proportional Counter Array (PCA), the High Energy X-ray Timing Experiment (HEXTE), and the All Sky Monitor (ASM). All three of these instruments perform continuous X-ray sky surveys and measure short (as low as 10 μ sec) as well as long (days to year) variations in energy and spectra of X-ray targets of interest.³ The PCA is sensitive to X-ray energies from 2 to 60 keV, whereas the HEXTE is sensitive from 20 to 200 keV. These two instruments combined constitute a powerful X-ray telescope. The ASM scans the sky looking for any new X-ray activity in the 2 to 20 keV region once every 90-min orbit. As a result, the attitude control system, summarized in the subsequent sections, is designed to ensure that these instruments can perform their tasks.

6.2 XTE Attitude Control System Overview

The Attitude Control System (ACS) is designed to meet the above science objectives by using an assortment of sensors, actuators, electronics, and control system software. These system controls are housed in the Attitude Control Electronics (ACE). The 80C85 microprocessor provides command decoding, telemetry encoding, and 1773 bus support, and houses the autonomous, independent sun-pointed safehold mode controls algorithm.³ All of the attitude control subsystem software, except the safehold mode, is located in the redundant 80386 microprocessor. This processor interfaces with the ACE to receive sensor signals, issue torque commands to wheel and magnetic actuators, and to receive star tracker data over the ACS bus.

The XTE uses a low noise gyro/star tracker system to determine attitude. The Inertial Reference Unit (IRU) provides 3-axis attitude data which is combined optimally in a Kalman filter with measurements taken from two Charge-Coupled-Device (CCD) star trackers. The trackers derive roll, pitch, and yaw information by taking multiple star measurements. The trackers are mounted on the XTE in a V-shaped configuration with their boresight offset 4.95° from the XTE instrument boresight (spacecraft roll axis, or $\hat{\mathbf{b}}_1$). There are also two fine sun sensors that can be used to update the Kalman Filter when stars are not available. Sun sensors are used during earth occultations or during extended pointing at a star-sparse region of the celestial sphere. There are also 8 coarse sun sensors located on the spacecraft. These sensors are only used to initiate sun acquisition after launch vehicle separation, and when the XTE spacecraft transitions into B-dot sun pointing mode. B-dot is a control law that relies on magnetic coils or torque rods as control actuators. It is based on

the measurement of the rate of change of body-fixed magnetometer signals. The B-dot sun pointing mode occurs when momentum management fails and the reaction wheels become saturated. The magnetometer and torquer bars are used to dump the extra momentum via the B-dot control law, while the coarse sun sensors keep the spacecraft pointed in the general direction of the sun.

In addition to these sensors, the XTE spacecraft uses reaction wheel actuators to counteract disturbance torques and to slew the spacecraft. The four reaction wheels are mounted in a pyramid arrangement with a 45° angle. In the event of a wheel failure, any three wheels can be used without losing the mission. Long term momentum management is accomplished using three redundant three-axis magnetometer sensors and three doubly-wound magnetic torquer bar actuators.³

The reaction wheels are used for two types of slewing required for the science missions: stellar pointing and fixed rate target slew. Long term inertial pointing to a single astronomical object is accomplished in the stellar pointing mode. The fixed rate target slew mode is a large-angle, eigen-axis maneuver to move the spacecraft $\hat{\mathbf{b}}_1$ axis from one stellar object to another (rest-to-rest maneuver). This mode is used to maneuver between pre-planned targets and to acquire occasional targets of opportunity quickly. As a result, the rest-to-rest eigen-axis maneuver is optimized for near-minimum time.²⁸ The biggest constraints are the saturation limits of the rate gyros and reaction wheels, the maximum slew rate, and the maximum available control torque.

The control torque for the roll, pitch, and yaw axes are computed using a standard (PID) control law such as the one presented in Ref. 28. A Lyapunov function is used to generate the control for a near-minimum time eigenaxis maneuver. Actuator and sensor saturation limits are accounted for in the controller. The quaternion error between the actual and commanded attitude is computed from

the Kalman Filter, which is driven to zero by the PID controller. Having given some background information on the X-ray Timing Explorer, we now present our simplified XTE system model used in the computer simulation.

6.3 XTE Spacecraft Model

The simplified XTE model is based on the data presented in Refs. 3 and 28. We simplify the model by making a few assumptions:

(1). All attitude estimation using the gyro/star trackers, sun sensors, and the Kalman Filters is ignored. The initial attitude and angular velocity data are assumed known so that the XTE spacecraft can “see” and acquire a target. Modified Rodrigues Parameters (MRP’s) are used in place of quaternions as the attitude parameters.

(2). The All Sky Monitor (ASM) and the High Energy X-ray Timing Experiment (HEXTE) dynamically interact with the spacecraft. Both of these instruments are gimballed to allow single or two axis large angle motion. This angular motion, coupled with the very large instrument inertias, imparts a large exchange of angular momentum on the spacecraft which the ACS system can not fully handle.³ This momentum exchange causes a degradation in the XTE stability (jitter) performance. As a result, we assume that all instruments are fixed within the body and can not independently rotate. We also assume that there are no flexible modes, such as solar arrays, that can be excited during the maneuver to degrade the stability. The only component of the solar arrays that we are concerned with is the solar panel axis in the body, which is used in computing

the reference trajectory.

(3). The XTE spacecraft is a large spacecraft with moments of inertia of the body $\mathbf{I} = \text{diag}(6292, 5477, 2687) \text{ kg-m}^2$. In addition, the spacecraft's most stringent attitude control performance requirement is the pitch and yaw stability (jitter) specification of 30 arc-sec, 3σ . The XTE inertia and pointing requirements result in slew times for target tracking on the order of 10-15 minutes. This fact is illustrated in Ref. 28 for the XTE near-minimum-time eigenaxis maneuver. Since we are using the XTE for tracking targets on the earth, the spacecraft has to be able to quickly acquire targets while they are above the horizon. Slew times on the order of 10-15 minutes means that spacecraft will be far past the target when target acquisition has been achieved. To remedy this we assume that the XTE has smaller moments of inertia for faster maneuvering. We define the new moments of inertia of the XTE system as:

$$\mathbf{I} = \text{diag}(320, 300, 275) \text{ kg-m}^2 \quad (6.1)$$

where \mathbf{I} is the inertia matrix of the XTE body and reaction wheels.

(4). The target is not an astronomical object that is inertially fixed, but a geographic point located on the moving Earth. At the end of the tracking maneuver, the spacecraft will rotate with some final angular velocity instead of being zero.

Having stated our assumptions, we can now present our simplified XTE model. Our spacecraft is modeled as a gyrostatt with four identical reaction wheels, like the XTE, mounted in a pyramid arrangement with a 45° angle. Table 6.1 illustrates some of the physical parameters that we use for the XTE. The actual axial moments of inertia for the XTE's reaction wheels were unavailable, but

the angular momentum and the mass of the wheels were specified in Ref. 3 to be 50 N-m-s and 18.3 kg, respectively. We found that an ITOS series reaction wheel, found in Ref. 27, approximately matches the available XTE reaction wheel data. The wheel inertias are listed in Table. 6.1. The

Table 6.1: XTE Physical Data

Item	Value	Units
\mathbf{J}	$\text{diag}\{320, 300, 275\}$	$kg\text{-}m^2$
\mathbf{I}_s	$\text{diag}\{14.43, 14.43, 14.43, 14.43\}$	$kg\text{-}m^2$
\mathbf{p}	$[0 \ 1 \ 0]^T$	
\mathbf{a}_b	$[1 \ 0 \ 0]^T$	
Pyramid Angle	45	deg

maximum wheel torque for the XTE is $T_{sat} = 0.3$ N-m, and the maximum slew rate is $\dot{\theta}_{max} = 0.2$ deg/s. Note that because we are not using the exact reaction wheel inertias, these wheel constraints are not exact. The reaction wheels are located in the body by

$$\mathbf{A} = \begin{bmatrix} \sqrt{2}/2 & \Leftrightarrow\sqrt{2}/2 & \sqrt{2}/2 & \Leftrightarrow\sqrt{2}/2 \\ \sqrt{2}/2 & 0 & \Leftrightarrow\sqrt{2}/2 & 0 \\ 0 & \Leftrightarrow\sqrt{2}/2 & 0 & \sqrt{2}/2 \end{bmatrix} \quad (6.2)$$

Obviously, this is not square, so the pseudo-inverse discussed in Chapter 5 is used to invert this matrix within the simulation.

In addition to the wheels, the panel vectors, \mathbf{p} , for the solar arrays are defined in the XTE to be along $\pm\hat{\mathbf{b}}_2$. We only consider the axis along $\hat{\mathbf{b}}_2$, because if one panel axis is perpendicular to the sun, then so is the other. We also use the HEXTE instrument in the tracking simulation because

it is mounted facing the $\hat{\mathbf{b}}_1$ direction in agreement with our sensor axis definition. As pointed out in the previous section, we have assumed that the HEXTE is fixed within the body and can not rotate. The HEXTE boresight axis \mathbf{a}_b is defined in Table 6.1. Recall from Chapter 4 that the virtual spacecraft has the same physical properties as the actual spacecraft. The boresight axis and the panel vector are also defined along the “1” and “2” directions, respectively, in the reference frame.

Clearly, we are leaving out a lot of attitude control systems and important dynamic effects with our assumptions. To get the complete dynamic behavior of the XTE, we would need to account for all of the interactions between the instruments and the space environment with the XTE’s attitude control system, and model the attitude estimation algorithms properly. However, our goal is to demonstrate the characteristics and the effectiveness of the Lyapunov controller with a realistic spacecraft model. Therefore, we can assume a simple model without losing any insight on the behavior of a closed-loop feedback controller in a dynamic system which is illustrated in Chapter 7.

Chapter 7

Results and Discussion

This chapter explores some of the characteristics exhibited by a Lyapunov control law through numerical simulations. A simplified model of the X-ray Timing spacecraft (XTE) is used in modeling a target tracking simulation. We want our XTE spacecraft model to align its High-Energy X-ray Timing Experiment (HEXTE) sensor at Cape Canveral, Florida for X-ray imaging in the 20 to 200 keV range. For the purposes of this discussion, we are assuming that the XTE is a remote sensing spacecraft when in reality it is not since the XTE scans for X-ray emissions coming from celestial bodies such as white dwarfs and neutron stars. We demonstrate the effectiveness of the Lyapunov controller acquiring a target in a relatively short time. The effect of the gain constants on eliminating tracking errors is investigated. By picking sufficiently large gains, we determine whether or not the controller can be linearized, and if it is robust.

7.1 Initial Conditions for Tracking Maneuver

Setting up the XTE simulation first requires that the orbit be modeled. The orbit is defined to be circular, with no environmental effects other than the gravity gradient torque. The two-body equation of motion given by Eq. (3.21) is used to simulate the XTE orbit around a spherical Earth. The orbit parameters were taken from a Two Line Element set (TLE) for an epoch of October 2, 1998 as shown in Table 7.1. Given this particular orbit, we want the XTE spacecraft to acquire and track Cape Canaveral, Florida ($\delta_t = 28.467^\circ N$, $L_t = 80.467^\circ W$) starting from a sub-satellite point located at $15.219^\circ N$ latitude and $79.752^\circ W$ longitude as defined by the TLE.

The initial conditions on the XTE and reaction wheels are listed in Table 7.2 where the wheel speeds were chosen to ensure that the axial angular momentum was 50 N-m-s. The initial attitude was chosen by letting the body frame be “off-nadir.” This means that $\hat{\mathbf{b}}_3$ and $\hat{\mathbf{o}}_3$ are misaligned by 40.511° . Now that we have defined all of our initial conditions for the XTE, we produce the simulation results of the tracking maneuver in the next section.

7.2 Simulation Results

Three sets of simulations with the Lyapunov controller were run using a representative range of values for k_1 and k_2 . Table 7.3 summarizes the gain values used in simulating a tracking maneuver. The results of the first simulation are presented in Figs. 7.1–7.10. Figure 7.1 contains the time

Table 7.1: XTE Orbital Data

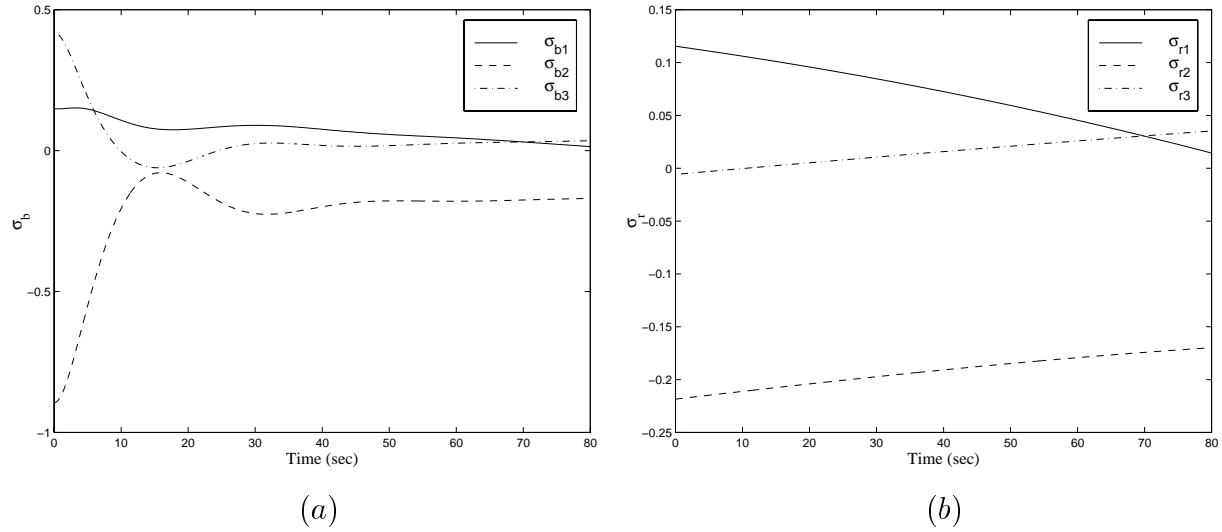
Item	Value	Units
Altitude	600	<i>km</i>
Eccentricity	0.0	
Inclination	22.9844	<i>deg</i>
Arg. of Perigee	0.2319	<i>deg</i>
RAAN	112.5017	<i>deg</i>
True Anomaly	402.0117	<i>deg</i>
Epoch	10/2/98, 14:38:08	

Table 7.2: XTE Initial Conditions

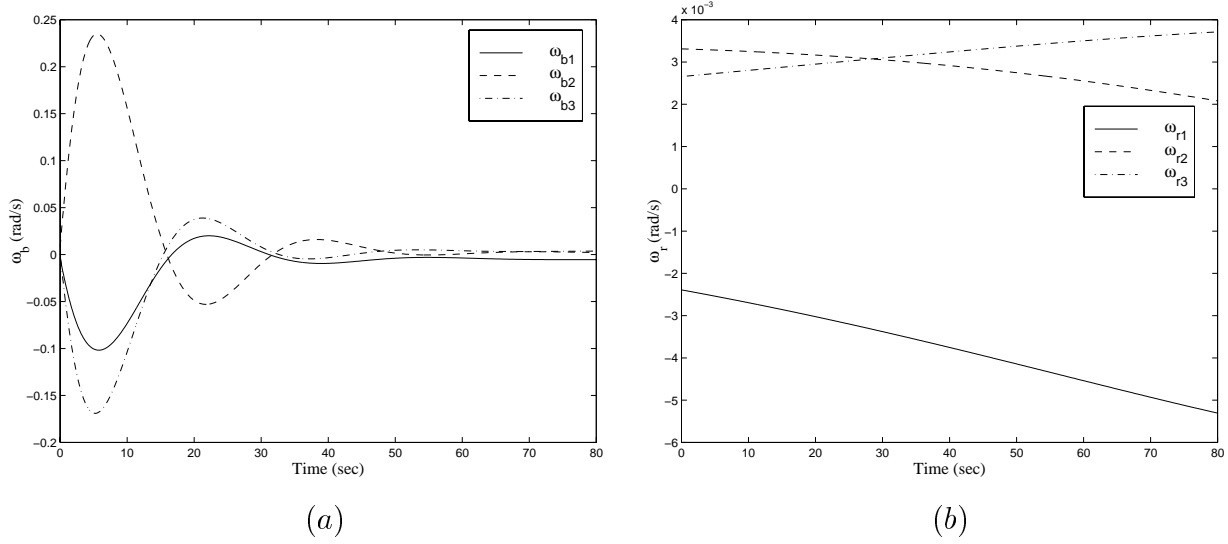
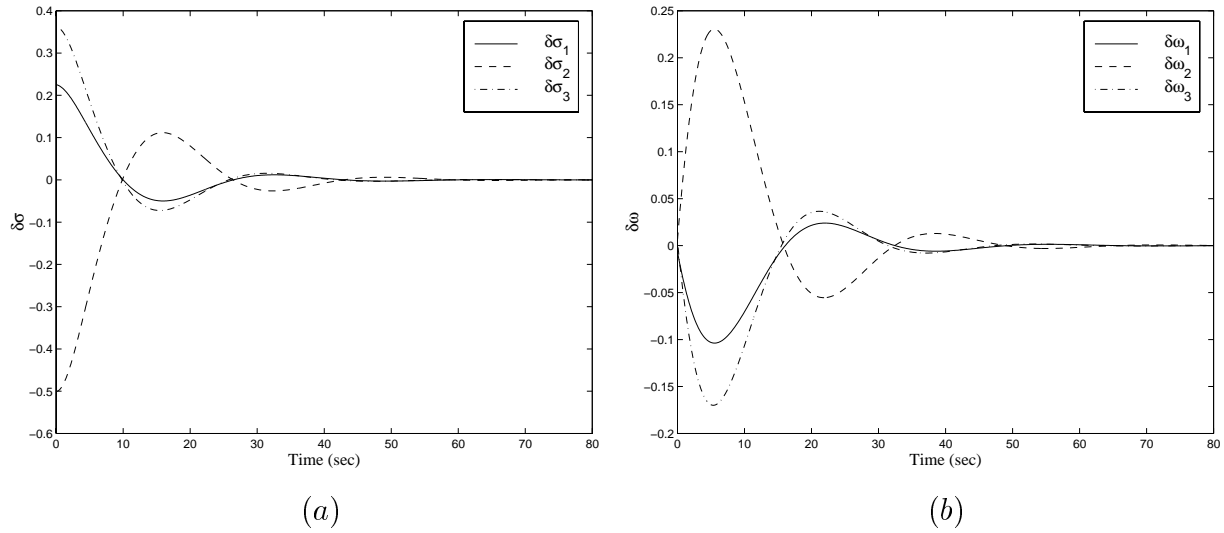
Item	Value	Units
$\sigma_b(0)$	$[0.14904 \quad \Leftrightarrow 0.89515 \quad 0.42011]^T$	
$\sigma_r(0)$	$[0.11549 \quad \Leftrightarrow 0.21848 \quad \Leftrightarrow 0.00580]^T$	
$\omega_b(0)$	$[\Leftrightarrow 0.0040 \quad \Leftrightarrow 0.00854 \quad 0.00090]^T$	<i>rad/sec</i>
$\omega_r(0)$	$[\Leftrightarrow 0.00239 \quad 0.00331 \quad 0.00266]^T$	<i>rad/sec</i>
$\omega_s(0)$	$[3.34650 \quad 3.34650 \quad 3.34650 \quad 3.34650]^T$	<i>rad/sec</i>

Table 7.3: The Lyapunov Controller Gains

Cases	k_1	k_2
Case 1:	50	50
Case 2:	10	10
Case 3:	120	120

Figure 7.1: The Case 1 Time History of (a) σ_b and (b) σ_r

histories of the actual and reference body attitude of the XTE. The initial conditions were given in Chapter 6. The attitude starts off with some oscillations during the target acquisition phase, and damp out as the errors are eliminated during the tracking part of the maneuver. Both of the figures show that by the end of the simulation $\sigma_b(t) = \sigma_r(t)$. The reference trajectory attitude is approximately linear for the maneuver, and so is the actual attitude once errors have vanished. As pointed out in Ref. 22, the MRP's behave linearly because we specified the principal rotation angle to be $\Leftrightarrow 180^\circ \leq \phi \leq 180^\circ$. The corresponding angular velocities for the actual and virtual spacecraft are shown in Fig. 7.2. A careful inspection yields that the actual angular velocity was driven to the same value as the reference angular velocity. There are large oscillations that are damped out after 50 seconds. The large platform angular velocity suggests that the XTE was pointed far from the target, and that there were large tracking errors in the system. We look at the error plots to verify this.

Figure 7.2: The Case 1 Time History of (a) ω_b and (b) ω_r Figure 7.3: The Case 1 Time History of (a) $\delta\sigma$ and (b) $\delta\omega$

The tracking errors $\delta\sigma$ and $\delta\omega$ for the first simulation are shown in Fig. 7.3. Clearly, the Lyapunov controller drives both sets of tracking errors to zero asymptotically in about 65 seconds. This is also verified by Fig. 7.4, that shows V approaching the stable origin. Asymptotic stability causes the trajectory to spiral towards the origin as $V \rightarrow 0$. The initial attitude error in Fig. 7.3(a) is indeed substantially large, especially in σ_2 because we chose initial conditions that were not close

to the reference trajectory. Thus, $\delta\omega_2$ increases until the large attitude error in σ_2 is eliminated. Similar effects occur for the other two body axes in Fig. 7.3. From Fig. 7.3(b), the XTE model seems to have no initial angular velocity error. There is some error present, but it is small because we defined the XTE platform to rotate with small angular velocity on the order of 10^{-3} rad/s.

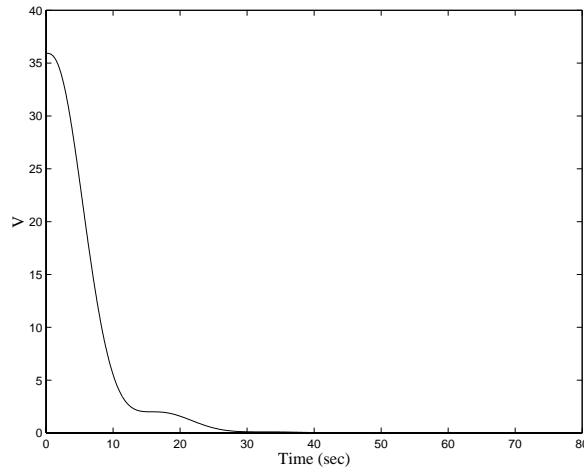


Figure 7.4: The Lyapunov Function for Case 1

In addition to driving the tracking errors asymptotically to zero, the feedback controller also drives the “error” between the sun vector and the solar panel vector to zero. Recall that the reference trajectory was designed to keep the sun vector perpendicular to the panel vector while the target was being tracked (yaw-steering). Figure 7.5 plots the dot product between the sun and the panel vector over time. As the tracking errors are eliminated for Case 1, \mathbf{s} and \mathbf{p} become perpendicular at about the same time the target has been acquired (65 seconds). This is useful because as sensors use power for data collection, spacecraft power requirements can be maintained without the use of batteries. This is only true if the spacecraft is not in eclipse.

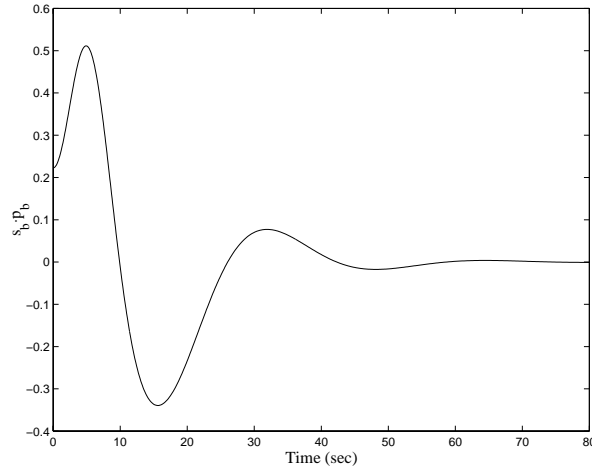


Figure 7.5: The Yaw-Steering Condition for Case 1

Maneuvering the XTE spacecraft to do yaw-steering or track a target requires control torques generated by the controller. The Lyapunov controller also drives the initial errors between the actual and reference control torques to zero. Figure 7.6 shows the time histories of the actual and reference control torques. The XTE's control torques are large initially, but approach the reference torques when the tracking errors are eliminated. The reference control torques are small, and fall within the maximum wheel torque constraint of 0.3 N-m.

On the other hand, the XTE torques initially violate that constraint. This also can be seen from the large wheel speeds in Fig. 7.7. Because of the large initial attitude error in the “2” direction, ω_{s2} reaches 7 rad/s in less than 10 seconds. If our XTE model was actually used for a science mission, we could not track a target this quickly. If the maneuver could be performed, the slew

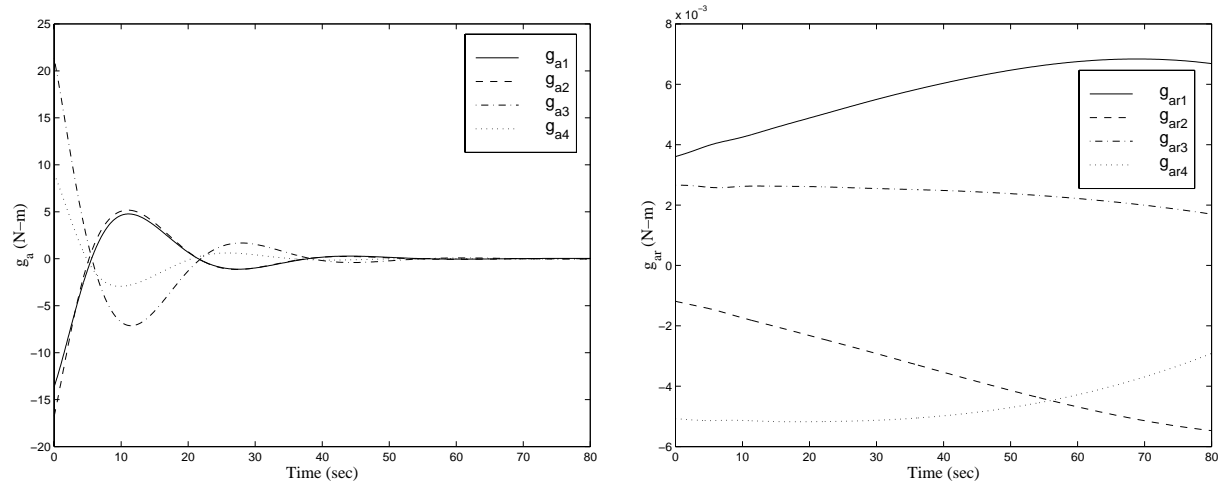


Figure 7.6: Case 1 Time Histories of (a) The Momentum Wheel Feedback Control Law in Eq. (5.13) and (b) The Desired Control

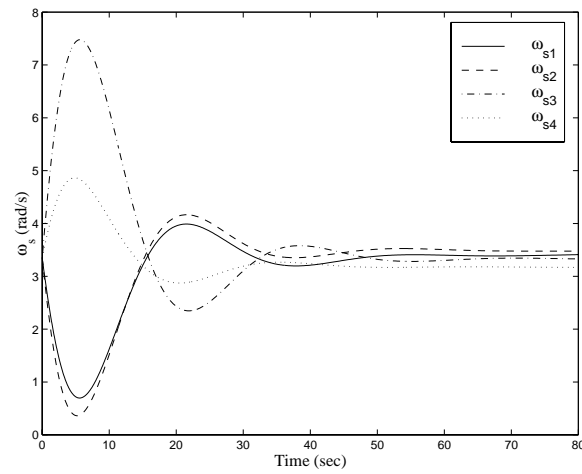


Figure 7.7: The XTE Reaction Wheels Speeds for Case 1

time would have to be longer to stay within wheel limits.

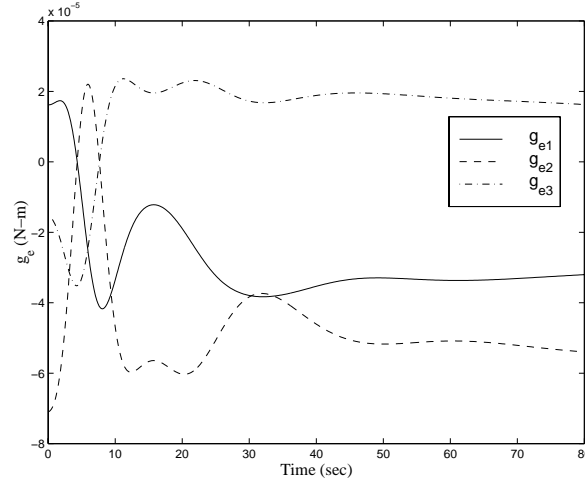


Figure 7.8: The External Torque for Case 1

Along with the controllable torques, the external torques are shown in Fig. 7.8. In our system model, the external torques are only the gravity gradient torques. As expected, these torques are extremely small for this tracking simulation. From Fig. 7.8, they vary during the acquisition of the target, but become constant during the tracking phase. In our system, they have a small, almost negligible, effect on the attitude control system.

Going from torques to momenta, we show the XTE system and axial angular momentum in Fig. 7.9. The XTE system initially has large momentum, and then loses it while acquiring the target. The momentum gradually increases towards the end of the simulation. The axial angular momentum starts with the prescribed value of 50 N-m-s and then increases to peak values, shown in Fig. 7.9(b), in about 5 seconds.

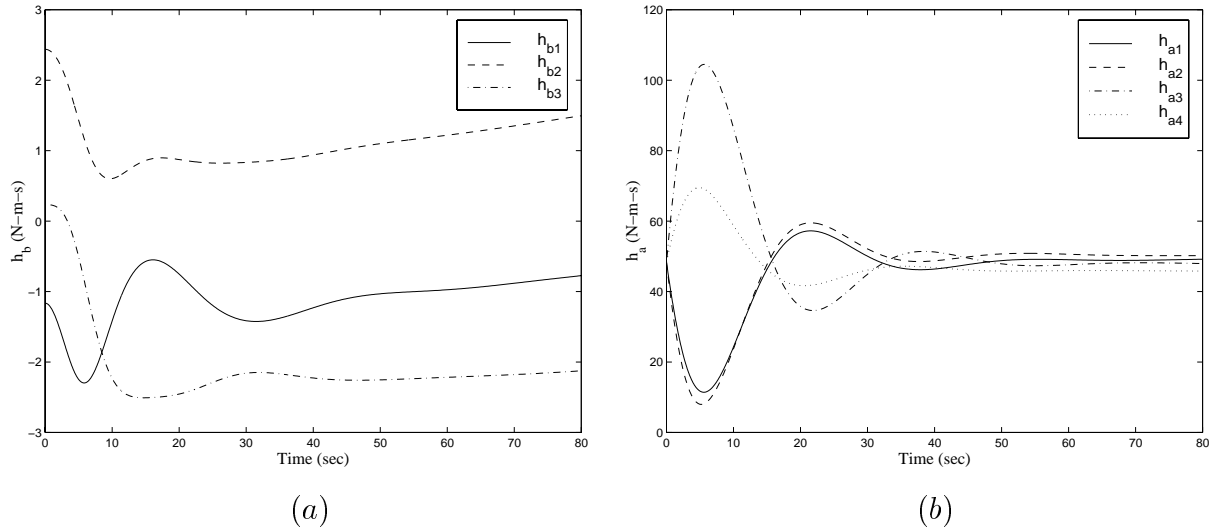


Figure 7.9: The Case 1 Time History of (a) The XTE System Angular Momentum and (b) The Reaction Wheel Angular Momentum

All of the plots of attitude, angular velocity, tracking errors, control torques, and angular momentum have relatively small terms associated with them. There are initial jumps in these during the maneuver, but overall the momenta, torques, and angular velocities, and errors remain small, especially after the target has been acquired. The expression for the controller in Eq. (5.13) is a function of all of these terms plus the gains k_1 and k_2 . Do the gains cause the last two terms to become large enough making the first six terms in Eq. (5.13) negligible? We try to answer this by plotting time histories of the magnitudes of the components of the control law, which is illustrated in Fig. 7.10. The last two terms, which are the gain terms, are much larger than the rest throughout the simulation. Components 1 and 3 are the only other two components that compare to the components 7 and 8 because the first and third terms are functions of the body angular momentum, wheel angular momentum, and, angular velocity. From Figs. 7.2, 7.3(b), and 7.9, these terms are fairly

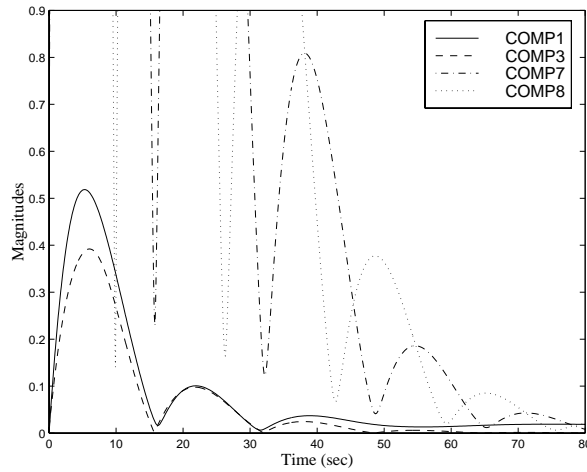


Figure 7.10: The Case 1 Time History of the Magnitudes of the Four Largest Terms In Eq. (5.13)

large at the start of the tracking maneuver and until the target is acquired. After target acquisition, these terms approach the reference values and cause the decay of the magnitudes in Fig. 7.10. The time histories of the remaining four normalized controller terms never exceed a value of 0.013, which is generated by the sixth term in the controller, and are not shown in Fig. 7.10 for clarity. Even though components 1 and 3 are relatively significant, it would seem that the control law can be approximated linearly by the last two terms. To verify this, we present two more cases varying the gains to see if this trend still occurs.

The results of case two are shown in Figs. 7.11–7.16. The same target and initial conditions are used for this simulation, and the gains are decreased to the values of Case 2 shown in Table 7.3. We omit the figures that are not affected by a change in the gains. The figures that are affected by gains are tracking errors, control torques, Lyapunov function, yaw-steering, and wheel speeds. Figure 7.11 shows the tracking errors for the second case. The most obvious feature is the increase

in the maneuver time. Decreasing the gains causes the tracking errors to decay at a slower rate. This is because the derivative of the Lyapunov function, which is a function of k_1 and $\delta\omega$, is not a large negative value which decreases the stability of the controller. As a result, the Lyapunov function in Fig. 7.12 approaches the origin at a slower rate than in the first case. The Lyapunov function is not as smooth as it was in Case 1, which also suggests that some stability has been lost by using smaller gains.

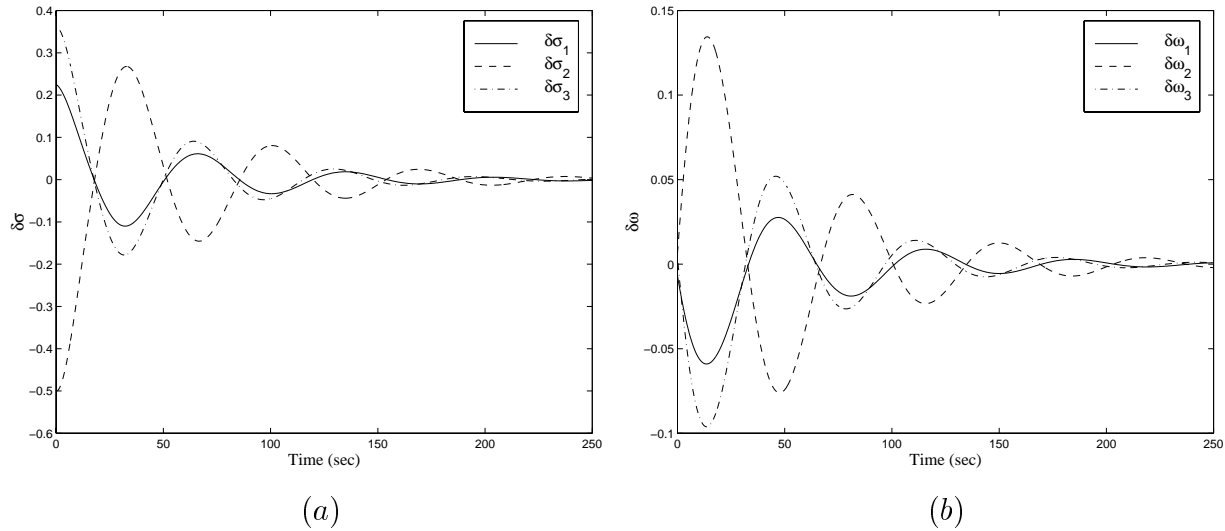


Figure 7.11: The Case 2 Time History of (a) $\delta\sigma$ and (b) $\delta\omega$

The smaller gains also affect the yaw-steering. As predicted, the sun vector and panel vector do not become perpendicular until about 250 seconds. The oscillations are more frequent and decay slower as the spacecraft converges on the yaw-steering condition. As seen in the previous case, the yaw-steering condition is satisfied with the elimination of the tracking errors in Fig. 7.11. Looking at Fig. 7.13, the spacecraft quickly goes through large yaw-steering angles during the first portion

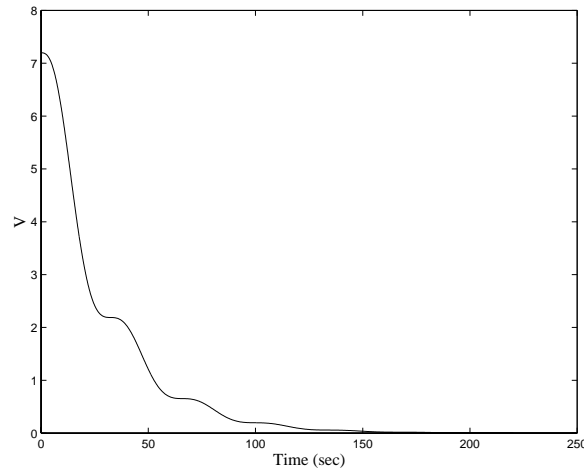


Figure 7.12: The Lyapunov Function for Case 2

of the tracking maneuver, which suggests that the spacecraft is quickly switching its direction of rotation about the boresight axis. These large angles, implied by the peak values in Fig. 7.13, indicate that the spacecraft is not as stable as it was in Case 1. If we had modeled solar arrays in Case 2, vibrations from these flexible appendages would result and interfere with the spacecraft's instruments and attitude control system.

Like the yaw-steering condition, both the control torques and wheel speeds take a long time to “settle” on their target tracking values. Figure 7.14 is the time history of the actual control torque generated by the controller for Case 2. Initially the control torque is large to compensate for the large initial tracking errors. Figure 7.15 shows the corresponding wheel speeds. Like the control torque, the wheel speeds are at their largest values initially and converge the reference wheel speeds at the end of the maneuver. The most interesting feature of these two plots is that the peak values are smaller than for the first case. The oscillatory motion of each is more pronounced than in the

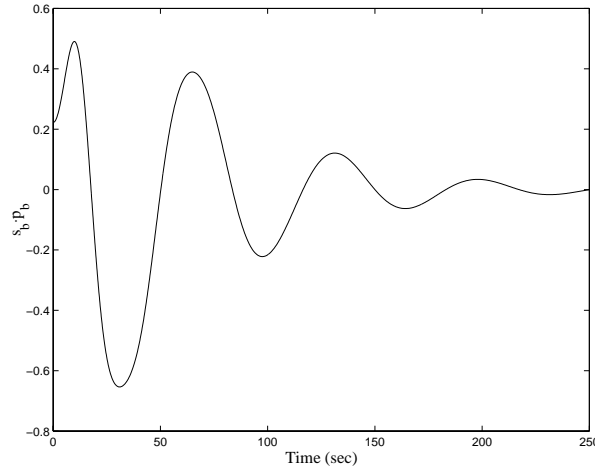


Figure 7.13: The Yaw-Steering Condition for Case 2

first case, but their amplitudes are lower than the first case. This results from the fact that the small gain terms within the feedback controller result in smaller control torque outputs. Even though the wheels provide less torque in Case 2, they still exceed the XTE wheel torque constraints.

Figure 7.16 illustrates the time histories of the normalized terms of the feedback controller. Although the gains are smaller in Case 2, the last two terms in the controller still dominate. Again, components 1 and 3 are still the only other significant terms in the control law. It is interesting to note that the other four normalized terms in Eq. (5.13) never exceed the value of 0.013 for Case 2. Figure 7.16 indicates that even with small gains, ($k_1 = k_2 = 10$), the feedback controller can be linearized.

The last case that we consider is Case 3, where the gains are made sufficiently large. As

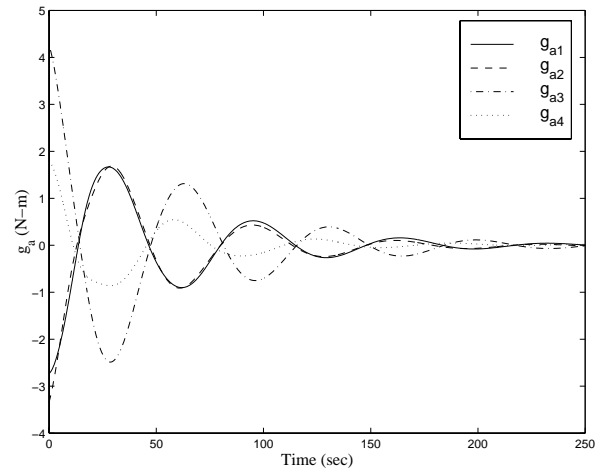


Figure 7.14: The Momentum Wheel Feedback Control Law in Eq. (5.13) for Case 2

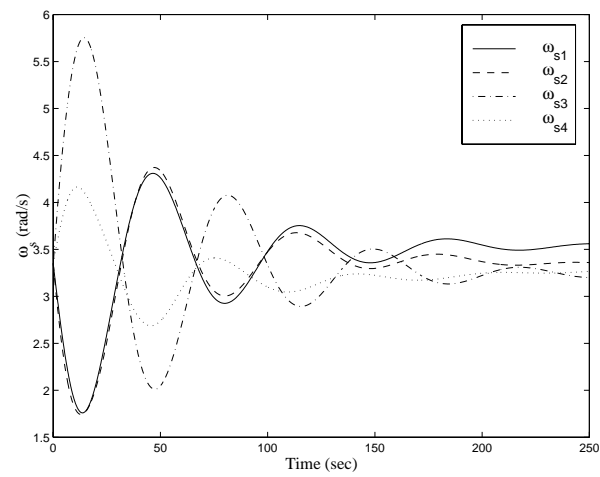


Figure 7.15: The XTE Reaction Wheels Speeds for Case 2

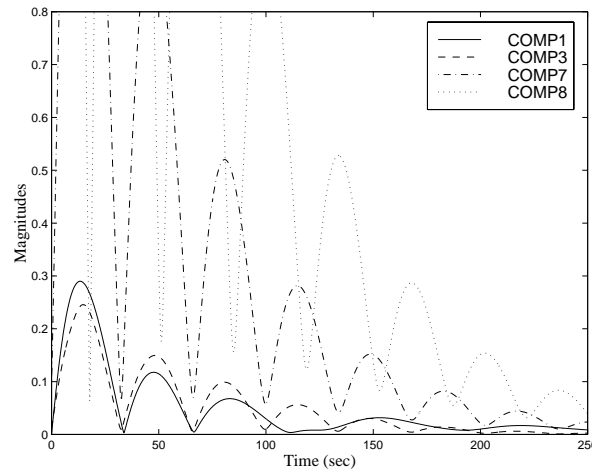


Figure 7.16: The Case 2 Time History of the Magnitudes of the Four Largest Terms In Eq. (5.13)

one might expect, increasing the gains increases the decay rate of the tracking errors as shown in Fig. 7.17. The larger gains increase the stability of the control law which is seen from the decrease in oscillations as the spacecraft converges on the reference trajectory. The increased damping also causes the Lyapunov function to approach the stable origin rapidly as shown in Fig 7.18. As a result of the larger gains, our XTE model acquired the target and performed the yaw-steering maneuver in Fig. 7.19 in 25 seconds. However, a quicker convergence time means that the control torques and reaction wheel speeds will be higher. This is indeed the case as shown in Figs. 7.20 and 7.21. Higher torques and wheel speeds also means that maximum wheel and torque constraints can be easily exceeded for fast tracking maneuvers.

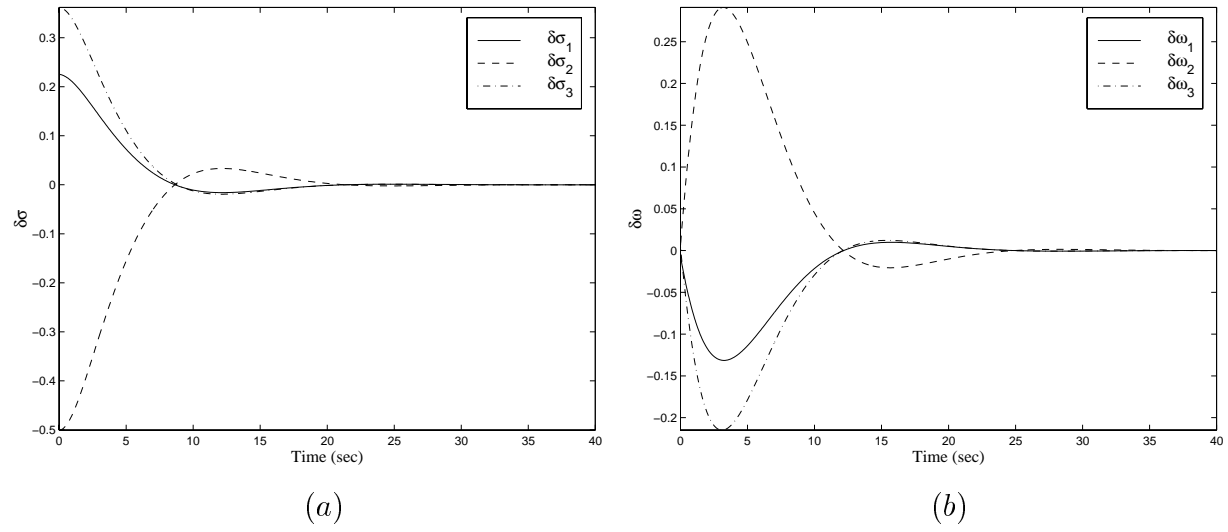
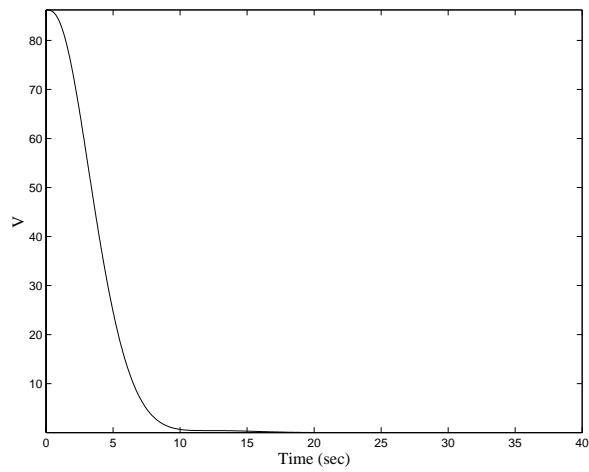
Figure 7.17: The Case 3 Time History of (a) $\delta\sigma$ and (b) $\delta\omega$ 

Figure 7.18: The Lyapunov Function for Case 3

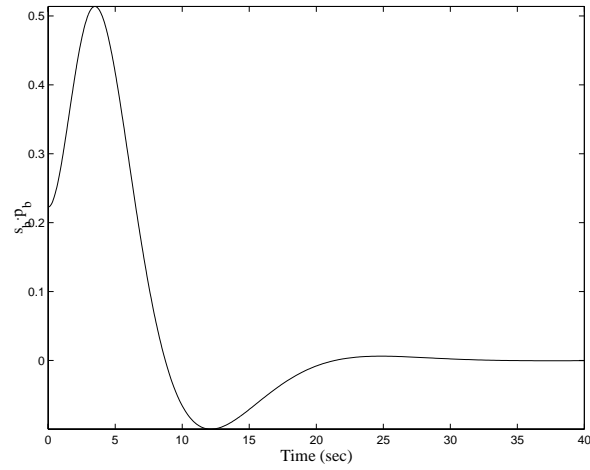


Figure 7.19: The Yaw-Steering Condition for Case 3

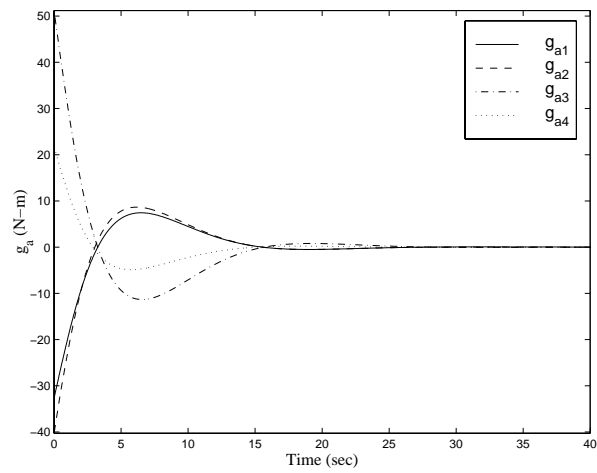


Figure 7.20: The Momentum Wheel Feedback Control Law in Eq. (5.13) for Case 3

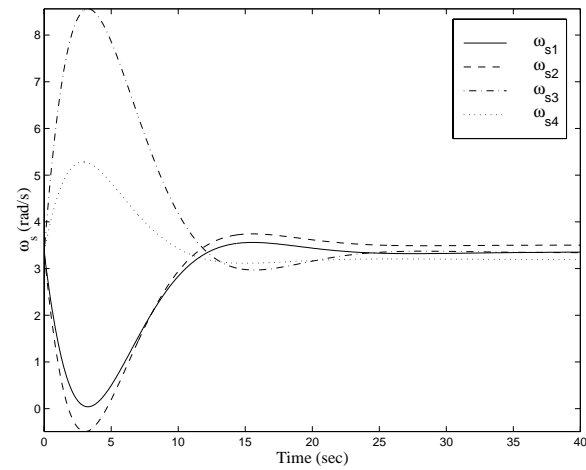


Figure 7.21: The XTE Reaction Wheels Speeds for Case 3

Guessing from the higher control torque, the magnitudes of the gain terms in the Eq. (5.13) should be much larger than the other terms. Figure 7.22 confirms this, and also shows that components 1,3, 7 and 8 follow the same trend found in Figs. 7.10 and 7.16. It is interesting to note that

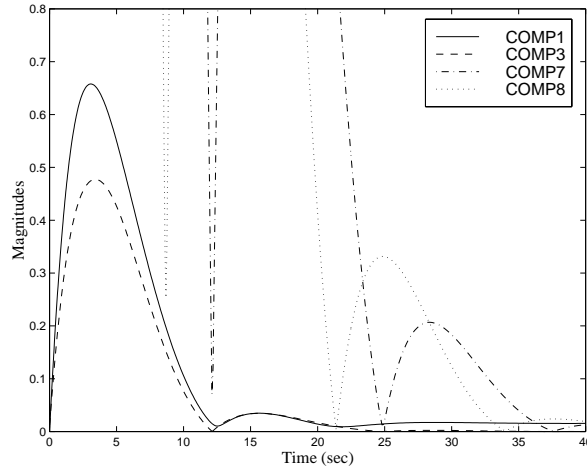


Figure 7.22: The Case 3 Time History of the Magnitudes of the Four Largest Terms In Eq. (5.13)

the other magnitudes also never exceed the maximum value of 0.013 during the maneuver. This reoccurring trend can be explained by looking at the second, fourth, fifth, and sixth components of Eq. (5.13). The four terms are functions of the external gravity gradient torque, the reference control torque, and the reference momenta. The gravity gradient torque is a relatively small value since our maneuver times are short. The reference momenta and control torques are small because the virtual spacecraft is assumed to be already pointing at the target and requires only a rotational maneuver to keep it tracking the target. These terms also have no dependence on the gains which means that they remain relatively constant for all three simulation cases.

The fact that the magnitude of the sixth term in the controller never exceeds 0.013 suggests

that this term can be neglected. Neglecting this term implies that the reference control torque is not needed for the control law to function properly. Figure 7.23 shows the error plots for Case 3 without the reference control torque. Comparing Figs. 7.23 and 7.17, it can be seen that both plots

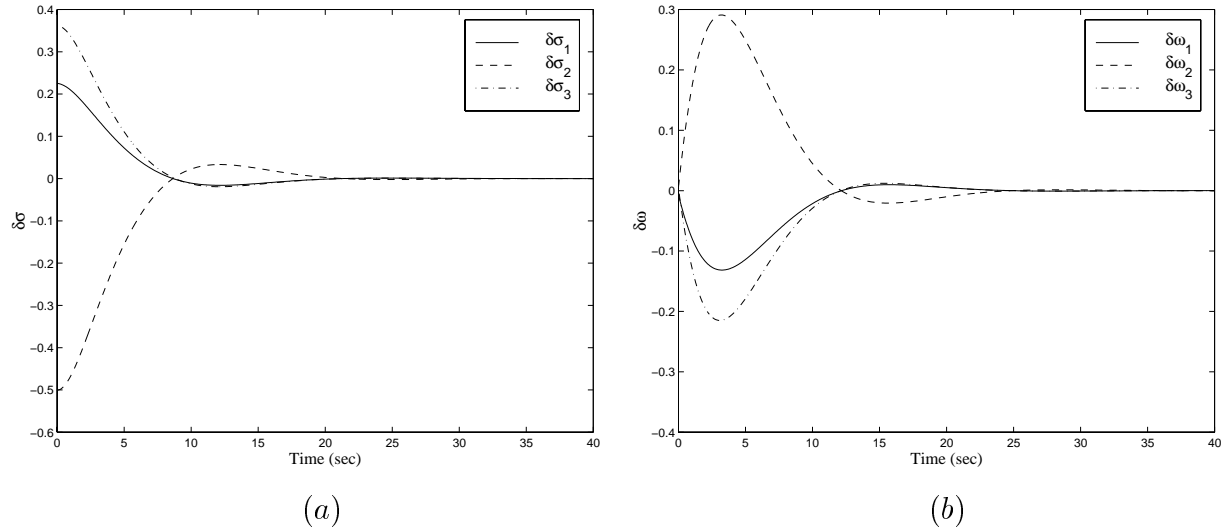


Figure 7.23: The Case 3 Time History of (a) $\delta\sigma$ and (b) $\delta\omega$ Without the Reference Torque

are virtually identical. The reference control torque does not affect the actual control torque, or the tracking maneuver. However, neglecting the reference control torque does not mean that we can completely neglect the reference tracking trajectory. The reference trajectory is needed to compute $\delta\sigma$ and $\delta\omega$ for the controller to work properly.

Finally, we briefly consider the cases where $k_1 > k_2$ and $k_1 < k_2$. When $k_1 > k_2$, the tracking errors are driven asymptotically to zero with virtually no oscillations. This results from the fact that \dot{V} is only a function of k_1 . A larger k_1 means that \dot{V} is a larger negative number, and the controller drives the tracking errors to zero with less oscillations. For the case where $k_1 < k_2$, there are more oscillations and the tracking errors take longer to damp out. The derivative \dot{V} becomes a smaller

negative number and the stability decreases.

Having run three simulations that show that the gain terms outweigh any other parameter in the controller, we can approximate it by neglecting the first six terms in Eq. (5.13). The feedback control becomes:

$$\mathbf{A} \mathbf{g}_a = k_1 \delta \boldsymbol{\omega} + k_2 \delta \boldsymbol{\sigma} \quad (7.1)$$

Figures 7.24– 7.28 show the results of the linearized controller using the Case 3 gain constants in Eq. (7.1). All of the figures show that the linearized controller is virtually identical to the nonlinear

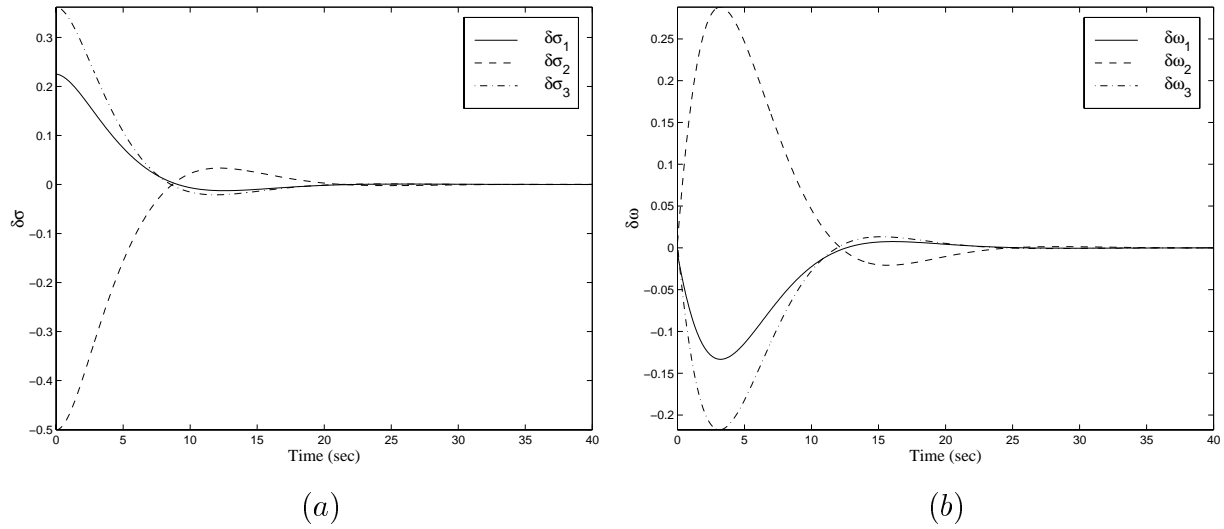


Figure 7.24: The Linearized Case Time History of (a) $\delta \boldsymbol{\sigma}$ and (b) $\delta \boldsymbol{\omega}$

version for sufficiently large gain constants. Looking at Eq. (7.1), it seems that the control torque would be equal to zero if there were no initial tracking errors ($\delta \boldsymbol{\sigma}(0) = \delta \boldsymbol{\omega}(0) = \mathbf{0}$). This is not the case because the error terms are never identically zero. Throughout this thesis, we have

made approximations and simplifications on the spacecraft and orbital equations of motion. We

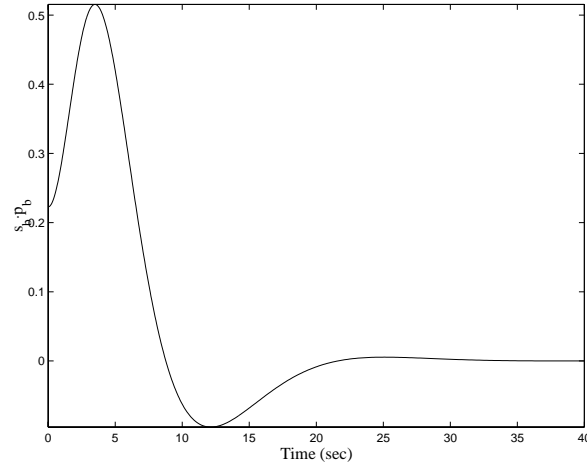


Figure 7.25: The Yaw-Steering Condition for Linearized Case

have also used these approximated equations of motion in a Runge-Kutta 45 numerical integration scheme to perform the simulation. The combination of using approximated equations and numerical integration introduces some round-off and truncation error in the numerical results. Therefore, the error terms computed during the simulation approach and oscillate around zero with values on the order of 10^{-3} resulting in a finite control torque. The control torque from the linearized controller leads to simulation results virtually identical to the ones achieved using the nonlinear feedback controller.

In addition to producing comparable control torques, the linearized control law also requires less Floating Point Operations (FLOPS) than the nonlinear control law. Table 7.4 shows the FLOP count for each of the three simulation cases. The linearized control law does not dramatically reduce the FLOPs needed to perform the tracking maneuvers. However, it does reduce some of the computational expense for a numerical tracking simulation.

Table 7.4: FLOP Count for Tracking Maneuver Simulation

Controller	Case 1	Case 2	Case 3
Nonlinear	5297068	8333708	4471861
Linear	5093485	8094053	4272244

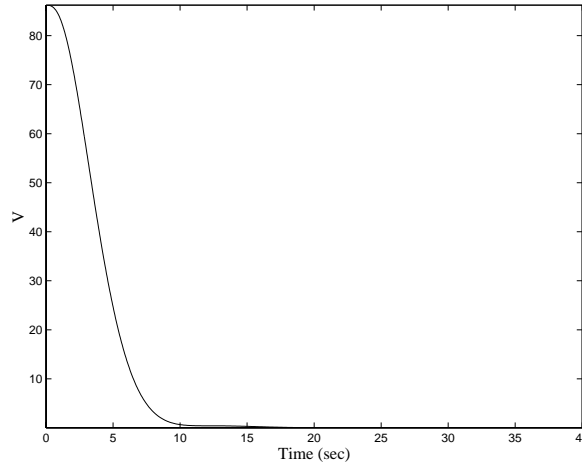


Figure 7.26: The Lyapunov Function for Linearized Case

We have shown sample tracking maneuvers for three different sets of scalar gains. The gains determined how fast the spacecraft acquired the target. The larger the gains, the faster the tracking errors were driven asymptotically to zero. The terms in the controller containing the gain constants dominated all other terms even for small values of k_1 and k_2 . As a result, the nonlinear feedback controller can be approximated by the linear error terms in the controller. The linearized controller performs just as well as the nonlinear controller and has less computational expenses associated with it. The conclusions drawn about this controller, as well as the reference trajectory, are summarized in the next chapter.

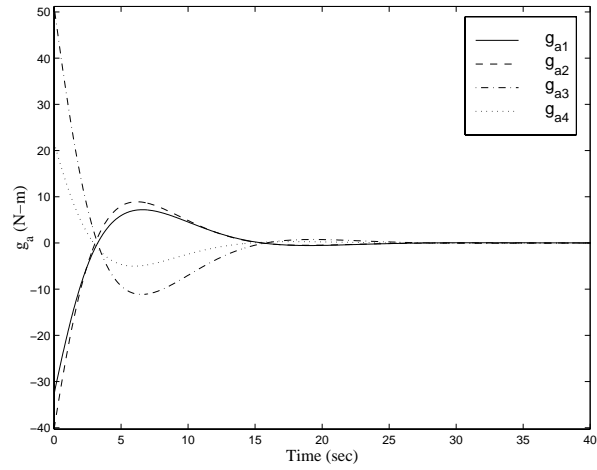


Figure 7.27: The Momentum Wheel Feedback Control Law in Eq. (5.13) for Linearized Case

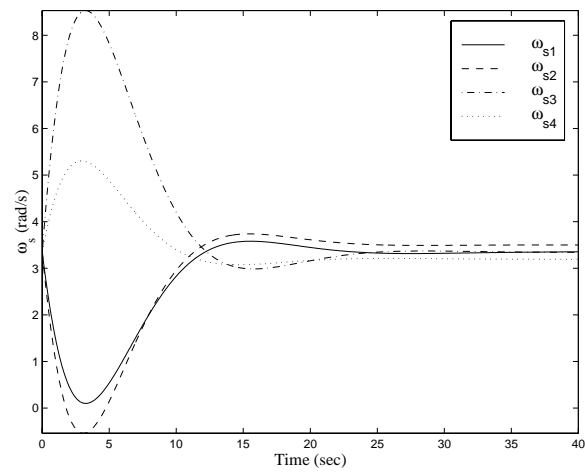


Figure 7.28: The XTE Reaction Wheels Speeds for the Linearized Case

Chapter 8

Summary and Conclusions

A method for computing a multiaxis target tracking trajectory is developed that also allows the solar panel vector to remain perpendicular to the sun vector during a tracking maneuver. The actual spacecraft is made to track this reference trajectory through a Lyapunov control law that drives initial tracking errors asymptotically to zero. The control law generates internal torques provided by momentum wheels for tracking rotational maneuvers. We now can draw conclusions based on these results.

Constructing the reference trajectory in a TRIAD-like manner is advantageous because only known vectors are needed. Only the spacecraft position, velocity, sensor, panel, and sun vectors are used to construct the reference motion. Since the position vectors are specific to a particular target, the reference trajectory is unique for each tracking problem. The reference attitude is constructed in the form of rotation matrix, which makes it coordinate independent. An attitude

coordinate independent reference attitude gives greater flexibility for spacecraft maneuver design. Other authors have developed similar algorithms, but they were designed for a particular type of attitude parameters and rotational maneuver and lack flexibility for use in various mission scenarios. Computing the reference angular velocity and acceleration by differentiating the attitude is straightforward, but involves some algebraic manipulation to compute the derivatives with respect to the rotating coordinate frames.

The sun tracking commands are automated, but numerical singularities occur causing the yaw-steering maneuver to break down. These singularities occur whenever the sun vector and sensor axis are co-linear, or when the panel vector orientation is less than 45° . Singularities appear both in the yaw-steering attitude, and the yaw-angle rate calculation. Both sets of singularities occur when the yaw-steering angle approaches 180° . However, the rate singularity numerically occurs when the sine term approaches zero, but is not investigated because the yaw-steer angle singularity dominates.

Using the reference commands, the Lyapunov controller drives the actual spacecraft orientation to the tracking attitude with the required tracking angular velocity. Lyapunov's second, or direct, method is frequently used for determining feedback control laws. It may or not be the best method to pick such a controller for several reasons. One reason is because the choice of the Lyapunov function is arbitrary. The other reason is that, once a suitable Lyapunov function is found, the derivative is usually complicated. Such is the case for our Lyapunov function. As a result, the control law itself becomes complicated and computationally expensive to perform.

However, the controller derived here does prove itself useful in tracking maneuvers. Using a

spacecraft model based on a simplified version of the XTE spacecraft, tracking maneuvers are carried out for different sets of gains. Tracking control is sensitive to the value of the gains. With gain values greater than 50, the tracking maneuvers can be quickly performed. Gain values less than 50 cause the stability of the controller to degrade because the derivative of the Lyapunov function is chosen as a function of k_1 . The error terms in the controller dominate the others regardless of the gain values. As a result, the error terms can be used as a linear approximation to the nonlinear control law. Simulation results involving the linear controller agree well with the results using the nonlinear controller. The linearized control law also results in less computational expenses, making the attitude tracking simulation more efficient. Based on the simulation results, one can conclude that the linearized controller is robust, and always yields the correct tracking control torque for gain parameters that are on the order of 10 or higher.

In conclusion, the reference trajectory algorithm and the feedback control law were found to be feasible for a mission that requires tracking of targets on the earth. The reference trajectory and control algorithms presented here are suitable for an individual spacecraft, or for each spacecraft in a formation. However, the yaw-steering singularity problem needs to be investigated further for use in actual tracking problems. The linearized control law, because of its simplicity and robustness, is feasible for use in a spacecraft's ACS. Gain selection also needs to be investigated to for the control law to be useful for spacecraft control.

Appendix A

Calculation of Sun Unit Vector

We present the calculations for the sun unit vector which are used in computing the reference trajectory in Chapter 4. The equations for computing the direction of the sun are taken from the *Astronomical Almanac*²⁴ and Wertz.²⁷ The mean elements of the orbit of the Sun are referred to the mean equinox and ecliptic of date coordinate systems. The mean motion of the sun is given by the following:

$$L = 279.611371^\circ + 0.9856473354t + \delta L \quad (\text{A.1})$$

$$M_\odot = 356.930969^\circ + 0.98560028t \quad (\text{A.2})$$

where L is the mean longitude of the sun, measured in the ecliptic plane from the mean equinox of date, and M_\odot is the mean anomaly of the sun. The time t is the current time in seconds since 1957 September 1, 0 hours UT (Universal Time). Time is referenced from this date since the sun data in Eqs (A.2-A.5) was derived for that year. The term δL is the correction added to the mean

longitude to get the true longitude:

$$\delta L = 1.918^\circ \sin(M_\odot) + 0.0201^\circ \sin(2M_\odot) \quad (\text{A.3})$$

The sun unit vector is computed from the true longitude as

$$\mathbf{s}_i = [\cos(L) \cos(E) \quad \sin(L) \cos(E) \quad \sin(L) \sin(E)]^T \quad (\text{A.4})$$

where E is the obliquity of the ecliptic of date with respect to the mean equator of date given by

$$E = 23.441242^\circ \Leftrightarrow (3.5626 \times 10^{-7})t \quad (\text{A.5})$$

where E is represented in degrees. Representing the sun vector in \mathcal{F}_i allows the calculations involving the sun vector in Chapter 4 to be simplified. A reasonable assumption made on the sun vector is that it is constant in the inertial frame. As result, all derivatives of \mathbf{s} are zero.

BIBLIOGRAPHY

- [1] Adams, J., Robertson, A., Zimmerman, K., and How, J., “Technologies for Spacecraft Formation Flying,” In *Proceedings of the Institute of Navigation GPS*, pp. 1321–1330, Kansas City, Missouri, September 1996.
- [2] Bar-Itzhack, I. Y. and Harman, R. R., “Optimized TRIAD Algorithm for Attitude Determination,” *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1996, pp. 208–211.
- [3] Bauer, F. H., Femiano, M. D., and Mosier, G. E., “Attitude Control System Conceptual Design for the X-ray Timing Explorer,” In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 236–249, Washington, D.C., 1992.
- [4] Day, R. M., “XTE: Astronomy with Autonomy,” *Aerospace America*, Vol. 34, No. 4, 1996, pp. 34–39.
- [5] Hughes, P. C., *Spacecraft Attitude Dynamics*, John Wiley & Sons, New York, 1986.
- [6] Folta, D., Bordi, F., and Scolese, C., “Considerations On Formation Flying Separations For Earth Observing Satellite Missions,” In *Advances in Astronautical Sciences*, Vol. 79, No. 2, 1992, pp. 803–822.
- [7] Folta, D. C. and Quinn, D. A., “An Algorithm For Enhanced Formation Flying of Satellites In Low Earth Orbit,” In *Proceedings of the American Institute of Physics Space Technology and Applications International Forum*, pp. 803–822, 1998.

- [8] Ford, K. A. and Hall, C. D., "Singular Direction Avoidance using Control Moment Gyros,"
To appear in the Journal of Guidance, Control and Dynamics.
- [9] Gramling, C. J., Lee, T., Niklewski, D. J., and Long, A. C., "Relative Navigation For Autonomous Formation Flying Of Spacecraft," In *Proceedings of the AAS/AIAA 1997 Astrodynamics Specialist Conference*, pp. 1–16, August 1997.
- [10] Hablani, H. B., "Design of a Payload Pointing Control System for Tracking Moving Objects,"
Journal of Guidance, Control, and Dynamics, Vol. 12, No. 3, 1989, pp. 365–374.
- [11] Hablani, H. B., "Multiaxis Tracking and Attitude Control of Flexible Spacecraft with Reaction Jets," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 4, 1994, pp. 831–839.
- [12] Hablani, H. B., "Target Acquisition, Tracking, Spacecraft Attitude Control, and Vibration Suppression with IPFM Reaction Jet Controllers," In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, pp. 1118–1137, Hilton Head, SC, August 1992.
- [13] Hall, C. D., "Escape From Gyrostat Trap States," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 3, 1998, pp. 421–426.
- [14] Hall, C. D., "Spinup Dynamics of Gyrostats," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, 1995, pp. 1177–1183.
- [15] Hall, C. D., Tsiotras, P., and Shen, H., "Tracking Rigid Body Motion Using Thrusters and Momentum Wheels," In *1998 AIAA/AAS Astrodynamics Conference*, pp. 302–307, August 1998.

- [16] Kalweit, C. C., "Optimum Yaw Motion for Satellites with a Nadir-Pointing Payload," *Journal of Guidance, Control, and Dynamics*, Vol. 6, No. 1, 1983, pp. 47–52.
- [17] Khalil, H. K., *Nonlinear Systems*, Macmillan, New York, 1992.
- [18] Levine, W. S., editor, *The Control Handbook*, CRC Press, Boca Raton, Florida, 1996.
- [19] Meirovitch, L., *Methods of Analytical Dynamics*, McGraw-Hill, New York, 1970.
- [20] Mohler, R. R., *Nonlinear Systems, Volume I Dynamics and Control*, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [21] Ogata, K., *Modern Control Engineering*, Prentice-Hall, Englewood Cliffs, New Jersey, third edition, 1991.
- [22] Schaub, H., Robinett, R. D., and Junkins, J. L., "Globally Stable Feedback Laws for Near-Minimum-Fuel and Near-Minimum-Time Pointing Maneuvers for a Landmark-Tracking Spacecraft," *Journal of the Astronautical Sciences*, Vol. 44, No. 4, 1996, pp. 443–466.
- [23] Shuster, M. D., "A Survey of Attitude Representations," *Journal of the Astronautical Sciences*, Vol. 41, No. 4, 1993, pp. 439–517.
- [24] United States Naval Observatory, *The Astronomical Almanac*, U.S. Government Printing Office, Washington, D.C., 1999.
- [25] Ulybyshev, Y., "Long-Term Formation Keeping of Satellite Constellation Using Linear-Quadratic Controller," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 109–115.

- [26] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, McGraw-Hill, New York, 1997.
- [27] Wertz, J. R., editor, *Spacecraft Attitude Determination and Control*, D. Reidel, Dordrecht, Holland, 1978.
- [28] Wie, B. and Lu, J., “Feedback Control Logic for Spacecraft Eigenaxis Rotations Under Slew Rate and Control Constraints,” *Journal of Guidance, Control and Dynamics*, Vol. 18, No. 6, 1995, pp. 1372–1379.

Vita

Matthew R. Long recieved is B.S. degree in Engineering Science and Mechanics from the Pennsylvania State University in 1997. While there, he worked under the direction of Dr. Larry Strait at the Penn State Applied Research Lab doing undergraduate research in Materials Science. After receiving his B.S. degree, he went to Virginia Tech to pursue a M.S. in Aerospace Engineering. Under the direction of Dr. Chris Hall, he worked on spacecraft dyamics and control problems. His field of specialization is attitude dynamics and control.