Spacecraft Attitude Dynamics and Control

Christopher D. Hall

January 16, 2000

Contents

Chapter 1

Introduction

Spacecraft dynamics and control is a rich subject involving a variety of topics from mechanics and control theory. In a first course in dynamics, students learn that the motion of a rigid body can be divided into two types of motion: translational and rotational. For example, the motion of a thrown ball can be studied as the combined motion of the mass center of the ball in a parabolic trajectory and the spinning motion of the ball rotating about its mass center. Thus a first approximation at describing the motion of a ball might be to model the ball as a point mass and ignore the rotational motion. While this may give a reasonable approximation of the motion, in actuality the rotational motion and translational motion are coupled and must be studied together to obtain an accurate picture of the motion. The motion of the mass center of a curve ball is an excellent example of how the rotational and translational motions are coupled.[1] The ball does not follow the parabolic trajectory predicted by analysis of the translational motion, because of the unbalanced forces and moments on the spinning ball.

The study of spacecraft dynamics is similar to the study of baseball dynamics. One first gains an understanding of the translational motion of the mass center using particle dynamics techniques, then the rotational motion is studied. Thus the usual study of spacecraft dynamics begins with a course in orbital dynamics, usually in the junior or senior year, and is followed by a course in attitude dynamics in the senior year or the first year of graduate study. In this book I assume that the student has had a semester of orbital dynamics, but in fact I only use circular orbits, so that a student with some appreciation for dynamics should be able to follow the development without the orbital dynamics background. The basics of orbital dynamics are included in Appendix A.

Another important way to decompose dynamics problems is into kinematics and kinetics. For translational motion, kinematics is the study of the change in position for a given velocity, whereas kinetics is the study of how forces cause changes in velocity. For rotational motion, kinematics is the study of the change in orientation for a given angular velocity, and kinetics is the study of how moments cause changes in the angular velocity. Translational kinematics is relatively easy to learn, since it only involves the motion of a point in three-dimensional space. Rotational kinematics, however, is usually more difficult to master, since it involves the orientation of a reference frame in three-dimensional space.

$$\begin{array}{l}
\overbrace{\text{Force}\quad\text{affects}\quad\text{Velocity}}^{\text{Kinetics}}\quad\text{affects}\quad\text{Position}\\
\text{Moment}\quad\text{affects}\quad\underbrace{\text{Angular Velocity}\quad\text{affects}\quad\text{Orientation}}_{\text{Kinematics}}
\end{array}$$

In this chapter, I begin by describing how attitude dynamics and control arises in the operation of spacecraft. This is followed by a description of the fundamental attitude control concepts that are in widespread use. Finally, I give an overview of the textbook.

1.1 Attitude dynamics and control in operations

Essentially all spacecraft include one or more subsystems intended to interact with or observe other objects. Typically there is one primary subsystem that is known as the payload. For example, the primary mirror on the Hubble Space Telescope is one of many instruments that are used to observe astronomical objects. The communications system on an Intelsat satellite is its payload, and the infrared sensor on board a Defense Support System (DSP) satellite is its payload. In each case, the payload must be pointed at its intended subject with some accuracy specified by the "customer" who purchased the spacecraft. This accuracy is typically specified as an angular quantity; e.g., 1 degree, 10 arcseconds, or 1 milliradian. The attitude control system designer must design the attitude determination and control subsystem (ADCS) so that it can meet the specified accuracy requirements.

It costs more than $10,000 to put a kilogram of mass into low-Earth orbit (LEO), and even more to put it into geostationary orbit (GEO).[2] A typical spacecraft masses about 500 kg, and costs tens to several hundred millions of dollars to design, manufacture, test, and prepare for launch. All of this money is spent to purchase the mission capability of the spacecraft. The attitude control system, propulsion system, launch vehicle, and so forth, are only there so that the mission may be performed effectively. If the mission can be accomplished without an ACS, then that mass can be used to increase the size of the payload, decrease the cost of launch, or in some other way improve the performance or reduce the cost. The bottom line is that the payload and its operation are the raison d'être for the spacecraft. This justifies our spending a little time on describing how the ACS fits into the operations of the payload.

There are many spacecraft payloads, but most fit into one of just two categories: communications and remote sensing. On a communications satellite, the payload comprises the radio transceivers, multiplexers, and antennas that provide the communications capability. Historically, most communications satellites have been in the geostationary belt, and have been either dual-spin or three-axis stabilized. More recently, a host of LEO commsats has been put into orbit, including the 66-satellite Iridium constellation, the 36-satellite OrbComm constellation, and the planned 48-satellite GlobalStar constellation. The Iridia use hydrazine propulsion for three-axis stabilization, whereas the OrbComms are gravity-gradient stabilized. The GlobalStar spacecraft are three-axis stabilized, using momentum wheels, magnetic torquers, and thrusters. In commsats, the mission of the ACS is to keep the spacecraft pointed accurately at the appropriate ground station. The more accurate the ACS, the more tightly focused the radio beam can be, and the smaller the power requirements will be. However, ACS accuracy carries a large price tag itself, so that design trades are necessary.

There are two basic types of remote sensing satellites: Earth-observing and space-observing. An Earth-observing spacecraft could be nadir-pointing, it could be scanning the land or sea in its instantaneous access area, or it could selectively point to and track specific ground targets. In the first case, a passive gravity-gradient stability approach might suffice, whereas in the second and third cases, an active ACS would likely be required, using some combination of momentum wheels, magnetic torquer rods, or thrusters.

A space-observing system could simply point away from the Earth, with an additional requirement to avoid pointing at the sun. This is essentially the ACS requirement for the CATSAT mission being built by the University of New Hampshire[1]. The CATSAT ACS uses momentum wheels and magnetic torquer rods. More complicated space-observing systems require both large-angle slewing capability and highly accurate pointing control. The Hubble Space Telescope is a well-known example. It uses momentum wheels for attitude control, and performs large-angel maenuvers at about the same rate as the minute-hand of a clock. The HST does not use thrusters because the plume would contaminate the sensitive instruments.

One problem with momentum wheels, reaction wheels, and control moment gyros is momentum buildup: external torques such as the gravity gradient torque and the solar radiation pressure will eventually cause the wheel to reach its maximum speed, or the CMG to reach its maximum gimbal angle. Before this happens, the spacecraft must perform an operation called momentum unload, or momentum dump: external torques are applied, using thrusters of magnetic torquer rods, that cause the ACS to decrease the wheels' speeds or the CMGs' gimbal angles. Depending on the spacecraft, this type of "maneuver" may be performed as often as once per orbit.

Another ACS operation involves keeping the spacecraft's solar panels pointing at

[1]Cooperative Astrophysics and Technology SATellite. See Refs. 3 and 4, and the website http://www.catsat.sr.unh.edu/.

the sun. For example, when the HST is pointing at a particular target, it still has a degree of freedom allowing it to rotate about the telescope axis. This rotation can be used to orient the solar panel axis so that it is perpendicular to the direction to the sun. Then the panels are rotated about the panel axis so that they are perpendicular to the sun direction. This maneuver is known as yaw steering.

1.2 Overview of attitude dynamics concepts

The attitude of a spacecraft, i.e., its orientation in space, is an important concept in spacecraft dynamics and control. Attitude motion is approximately decoupled from orbital motion, so that the two subjects are typically treated separately. More precisely, the orbital motion does have a significant effect on the attitude motion, but the attitude motion has a less significant effect on the orbital motion. For this reason orbital dynamics is normally covered first, and is a prerequisite topic for attitude dynamics. In a third course in spacecraft dynamics, the coupling between attitude and orbital motion may be examined more closely. In this course, we will focus on the attitude motion of spacecraft in circular orbits, with a brief discussion of the attitude motion of simple spacecraft in elliptic orbits.

Operationally, the most important aspects of attitude dynamics are attitude determination, and attitude control. The reason for formulating and studying the dynamics problem is so that these operational tasks can be performed accurately and efficiently. Attitude determination, like orbit determination, involves processing observations ("obs") to obtain parameters for describing the motion. As developed in Appendix A, we can determine the orbit of the satellite if we have the range (ρ), range rate ($\dot{\rho}$), azimuth (Az), azimuth rate ($\dot{\mathrm{Az}}$), elevation (El), and elevation rate ($\dot{\mathrm{El}}$) from a known site on the Earth. The deterministic algorithm to compute the six orbital elements from these six measurements is well-known and can be found in most astrodynamics texts. Of course, due to measurement noise, it is not practical to use only six measurements, and statistical methods are normally used, incorporating a large number of observations.

Similarly, we can determine the attitude, which can be described by three parameters such as Euler angles, by measuring the directions from the spacecraft to some known points of interest. In Fig. , we illustrate this concept: The spacecraft has a Sun sensor and an Earth sensor. The two sensors provide vector measurements of the direction from the spacecraft to the sun ($\vec{\mathbf{v}}_s$) and to the Earth ($\vec{\mathbf{v}}_e$). These will normally be unit vectors, so each measurement provides two pieces of information. Thus the two measurements provide four known quantities, and since it only takes three variables to describe attitude, the problem is overdetermined, and statistical methods are required (such as least squares). Actually there is a deterministic method that discards some of the measurements and we will develop it in Chapter 4. A wide variety of attitude determination hardware is in use. The handbook edited by Wertz (Ref. 5) provides a wealth of information on the subject. The more recent text by

Table 1.1: Attitude Control Concepts

Concept	Passive/ Active	Internal/ External	Environmental
Gravity Gradient	P	E	Y
Spin Stabilization	A	I	N
Dual-Spin	A	I	N
Momentum Wheels	A	I	N
Control Moment Gyros	A	I	N
Magnetic Torquer Rods	A	E	Y
Thrusters	A	E	N
Dampers	P or A	I	N

Sidi (Ref. 6) is notable for its appendices on hardware specifications.

Controlling the attitude of a spacecraft is also accomplished using a wide variety of hardware and techniques. The choice of which to use depends on the requirements for pointing accuracy, pointing stability, and maneuverability, as well as on other mission requirements such as cost and lifetime. All attitude control concepts involve the application of torques or moments to the spacecraft. The various methods can be grouped according to whether these torques are passive or active, internal or external, and whether the torques are environmental or not. A reasonably complete list of concepts is shown in Table 1.1.

The most fundamental idea in the study of attitude motion is the reference frame. Throughout the book we will work with several different reference frames, and it is important that you become familiar and comfortable with the basic concept. As a preview, let us consider an example where four different reference frames are used. In Fig. 1.1, we show three reference frames useful in describing the motion of a spacecraft in an equatorial orbit about the Earth. One of the reference frames whose origin is at the center of the Earth is an inertial reference frame with unit vectors $\hat{\mathbf{I}}$, $\hat{\mathbf{J}}$, and $\hat{\mathbf{K}}$. This frame is usually referred to as the Earth-centered inertial (ECI) frame. The $\hat{\mathbf{I}}$ axis points in the direction of the vernal equinox, and the $\hat{\mathbf{I}}\hat{\mathbf{J}}$ plane defines the equatorial plane. Thus the $\hat{\mathbf{K}}$ axis is the Earth's rotation axis, and the Earth spins about $\hat{\mathbf{K}}$ with angular velocity $\omega_\oplus = 2\pi$ radians per sidereal day. In vector form, the angular velocity of the Earth is $\vec{\boldsymbol{\omega}}_\oplus = \omega_\oplus \hat{\mathbf{K}}$.

The other frame centered in the Earth is an Earth-centered, Earth-fixed (ECEF) frame which rotates with respect to the ECI frame with angular velocity $\vec{\boldsymbol{\omega}}_\oplus = \omega_\oplus \hat{\mathbf{K}}$. This frame has unit vectors represented by $\hat{\mathbf{I}}'$, $\hat{\mathbf{J}}'$, and $\hat{\mathbf{K}}'$. Note that $\hat{\mathbf{K}}$ is the same in both the ECI and ECEF frames. The importance of the ECEF frame is that points

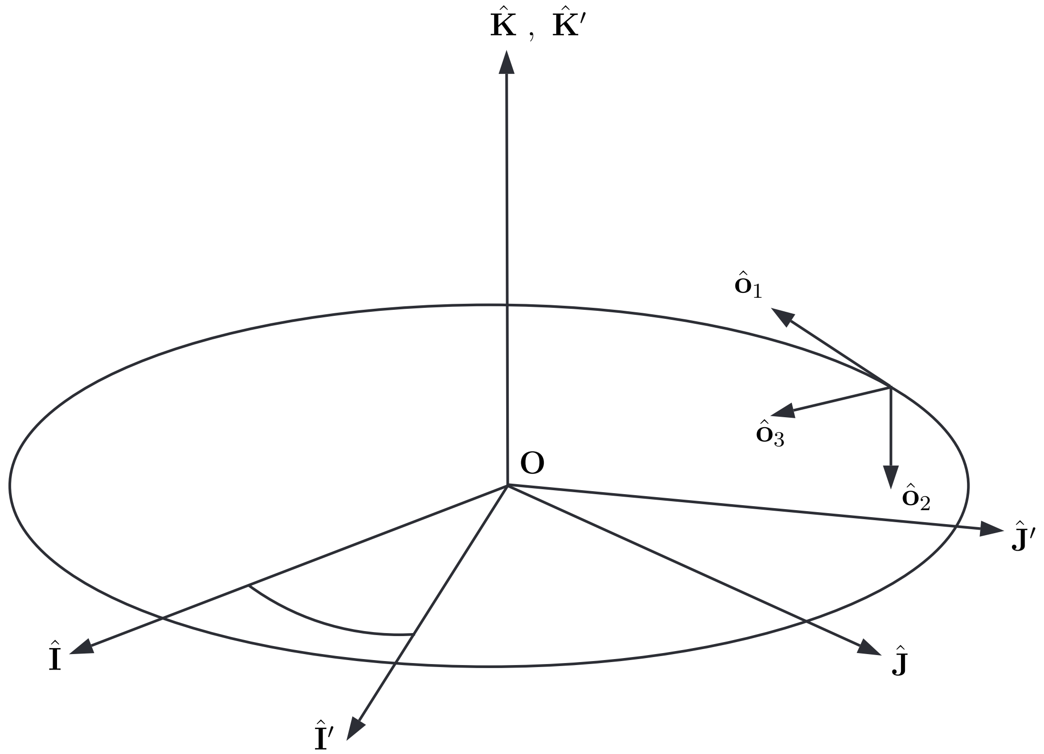

Figure 1.1: Earth-Centered Inertial, Earth-Centered Earth-Fixed, and Orbital Reference Frames for an Equatorial Orbit

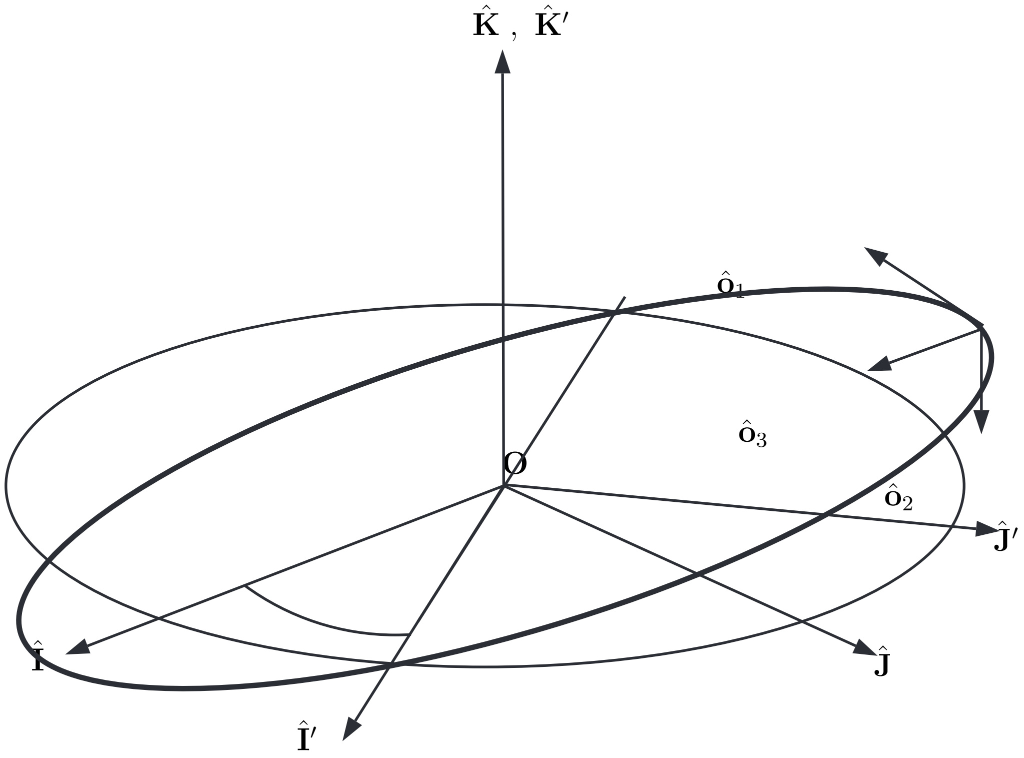

Figure 1.2: Earth-Centered Inertial, Earth-Centered Earth-Fixed, and Orbital Reference Frames for an Inclined Orbit

on the surface of the Earth, such as ground stations and observation targets, are fixed in this frame.

The other frame in the figure has its origin at the mass center of the spacecraft. This point is assumed to be in an orbit (circular or elliptical) about the Earth, thus its motion is given. As drawn in the figure, this orbit is also an equatorial orbit, so that the orbit normal is in the $\hat{\mathbf{K}}$ direction. The origin of this frame is accelerating and so it is not inertial. This frame is called the orbital frame because its motion depends only on the orbit. The unit vectors of the orbital frame are denoted $\hat{\mathbf{o}}_1$, $\hat{\mathbf{o}}_2$, and $\hat{\mathbf{o}}_3$. The direction pointing from the spacecraft to the Earth is denoted by $\hat{\mathbf{o}}_3$, and the direction opposite to the orbit normal is $\hat{\mathbf{o}}_2$. The remaining direction, $\hat{\mathbf{o}}_1$ is defined by $\hat{\mathbf{o}}_1 = \hat{\mathbf{o}}_2 \times \hat{\mathbf{o}}_3$. In the case of a circular orbit, $\hat{\mathbf{o}}_1$ is in the direction of the spacecraft velocity vector. For those familiar with aircraft attitude dynamics, the three axes of the orbital frame correspond to the roll, pitch, and yaw axes, respectively. This reference frame is non-inertial because its origin is accelerating, and because the frame is rotating. The angular velocity of the orbital frame with respect to inertial space is $\vec{\boldsymbol{\omega}}_o = -\omega_o \hat{\mathbf{o}}_2$. The magnitude of the orbital angular velocity is constant only if the orbit is circular, in which case $\omega_o = \sqrt{\mu/r^3}$, where μ is the gravitational parameter, and r is the orbit radius (see Appendix A). If the orbit is not circular, then ω_o varies with time. Note well that $\vec{\boldsymbol{\omega}}_o$ is the angular velocity of the orbital frame with respect to the inertial frame and is determined by the translational, or orbital, dynamics.

Another reference frame of interest is shown in Fig. 1.3 in relation to the orbital frame. This is the body-fixed frame, with basis vectors $\hat{\mathbf{b}}_1$, $\hat{\mathbf{b}}_2$, and $\hat{\mathbf{b}}_3$. Its origin is at the spacecraft mass center, just as with the orbital frame. However, the spacecraft body, or platform, is in general not aligned with the orbital frame. It is the relative orientation between these two reference frames that is central to attitude determination, dynamics, and control. The relative orientation between the body frame and the orbital frame is determined by the satellite's rotational dynamics, which is governed by the kinetic and kinematic equations of motion. The primary purpose of this text is to develop the theory and tools necessary to solve problems involving the motion of the body frame when the orbit is known.

1.3 Overview of the textbook

Most textbooks on this subject begin with some treatment of kinematics and then proceed to a study of a variety of dynamics problems, with some control problems perhaps included. Our approach is similar, but our aim is to spend more time up front, both in motivation of the topics, and in developing an understanding of how to describe and visualize attitude motion. To this beginning, we have an introductory chapter on Space Mission Analysis that will hopefully help readers to develop an appreciation for how attitude dynamics fits into the overall space mission. For a more traditional course, this chapter could be read quickly or even skipped entirely.

Figure 1.3: Orbital and Body Reference Frames

Chapter 3 introduces attitude kinematics, developing the classical topics in some detail, and introducing some new topics that may be used in a second reading. Chapter 4 covers the important topic of attitude determination. This topic is not usually covered in an introductory course, but I believe that mastery of this subject will enhance the student's appreciation for the remaining material. In Chapter 5 we develop the standard equations of motion and relevant results for rigid body dynamics. These topics lead directly to Chapter 6 on satellite attitude dynamics, where we apply basic dynamics principles to a variety of problems. In Chapter 7 we introduce and develop equations of motion for the gyroscopic instruments that are used as sensors in spacecraft attitude control systems. These are used in some simple examples, before proceeding to Chapter 8 where more rigorous development of attitude control problems is presented.

1.4 References and further reading

Dynamics and control of artificial spacecraft has been the subject of numerous texts and monographs since the beginning of the space age. Thomson's book,[7] originally published in 1961, is one of the earliest, and is currently available as a Dover reprint. The book remains a valuable reference, despite its age. Wertz's handbook[5] is perhaps the best reference available on the practical aspects of attitude determination and control. The text by Kaplan[8] treats a wide range of topics in both orbital and attitude dynamics. Kane, Likins, and Levinson[9] present a novel approach to satellite dynamics, using Kane's equations. Hughes' book[10] focuses on modeling and analysis of attitude dynamics problems, and is probably the best systematic and rigorous treatment of these problems. Rimrott's book[11] is similar to Hughes,[10] but uses scalar notation, perhaps making it more accessible to beginning students of the subject. Wiesel[12] covers rigid body dynamics, as well as orbital dynamics and basic rocket dynamics. Agrawal's book[13] is design-oriented, but includes both orbital and attitude dynamics. The brief book by Chobotov[14] covers many of the basics of attitude dynamics and control. Especially useful are the "Recommended Practices" given at the end of each chapter. The books by Griffin and French,[15] Fortescue and Stark,[16] Larson and Wertz,[17] and Pisacane and Moore[18] are all design-oriented, and so present useful information on the actual implementation of attitude determination and control systems, and the interaction of this subsystem with the overall spacecraft. The chapter on attitude in Pisacane and Moore gives an in-depth treatment of the fundamentals, whereas the relevant material in Larson and Wertz is more handbook-oriented, providing useful rules of thumb and simple formulas for sizing attitude determination and control systems. Bryson[19] treats a variety of spacecraft orbital and attitude control problems using the linear-quadratic regulator technique. Sidi[6] is a practice-oriented text, providing many detailed numerical examples, as well as current information on relevant hardware. The new book by Wie[20] furnishes a modern treatment of attitude dynamics and control topics. Finally, an excellent source of information on specific spacecraft is the Mission and Spacecraft Library, located on the World Wide Web at http://leonardo.jpl.nasa.gov/msl/home.html.

Bibliography

[1] Neville de Mestre. The Mathematics of Projectiles in Sport, volume 6 of Australian Mathematical Society Lecture Series. Cambridge University Press, Cambridge, 1990.

[2] John R. London III. LEO on the Cheap — Methods for Achieving Drastic Reductions in Space Launch Costs. Air University Press, Maxwell Air Force Base, Alabama, 1994.

[3] Chris Whitford and David Forrest. The CATSAT attitude control system. In Proceedings of the 12th Annual Conference on Small Satellites, number SSC98-IX-4, Logan, Utah, September 1998.

[4] Ken Levenson and Kermit Reister. A high capability, low cost university satellite for astrophyscial research. In 8th Annual Conference on Small Satellites, Logan, Utah, 1994.

[5] J. R. Wertz, editor. Spacecraft Attitude Determination and Control. D. Reidel, Dordrecht, Holland, 1978.

[6] Marcel J. Sidi. Spacecraft Dynamics and Control: A Practical Engineering Approach. Cambridge University Press, Cambridge, 1997.

[7] W. T. Thomson. Introduction to Space Dynamics. Dover, New York, 1986.

[8] Marshall H. Kaplan. Modern Spacecraft Dynamics & Control. John Wiley & Sons, New York, 1976.

[9] Thomas R. Kane, Peter W. Likins, and David A. Levinson. Spacecraft Dynamics. McGraw-Hill, New York, 1983.

[10] Peter C. Hughes. Spacecraft Attitude Dynamics. John Wiley & Sons, New York, 1986.

[11] F. P. J. Rimrott. Introductory Attitude Dynamics. Springer-Verlag, New York, 1989.

[12] William E. Wiesel. Spaceflight Dynamics. McGraw-Hill, New York, second edition, 1997.

[13] Brij N. Agrawal. Design of Geosynchronous Spacecraft. Prentice-Hall, Englewood Cliffs, NJ, 1986.

[14] V. A. Chobotov. Spacecraft Attitude Dynamics and Control. Krieger Publishing Co., Malabar, FL, 1991.

[15] Michael D. Griffin and James R. French. Space Vehicle Design. AIAA Education Series. American Institute of Aeronautics and Astronautics, Washington, D.C., 1991.

[16] Peter W. Fortescue and John P. W. Stark, editors. Spacecraft Systems Engineering. John Wiley & Sons, Chichester, 1991.

[17] Wiley J. Larson and James R. Wertz, editors. Space Mission Analysis and Design. Microcosm, Inc., Torrance, CA, second edition, 1995.

[18] Vincent L. Pisacane and Robert C. Moore, editors. Fundamentals of Space Systems. Oxford University Press, Oxford, 1994.

[19] Arthur E. Bryson, Jr. Control of Spacecraft and Aircraft. Princeton University Press, Princeton, 1994.

[20] Bong Wie. Space Vehicle Dynamics and Control. AIAA, Reston, Virginia, 1998.

1.5 Exercises

1. What types of attitude control concepts are used by the following spacecraft? If you can, tell what types of sensors and actuators are used in each case.
   - (a) Explorer I
   - (b) Global Positioning System
   - (c) Hubble Space Telescope
   - (d) Intelsat IV
   - (e) Intelsat IX
   - (f) Iridium
   - (g) OrbComm
   - (h) TACSAT I
2. What companies manufacture the following attitude control actuators? List some of the performance characteristics for at least one specific component from each type of actuator.
   - (a) momentum wheels
   - (b) control moment gyros
   - (c) magnetic torquer rods

(d) damping mechanisms

(e) hydrazine thrusters

3. What companies manufacture the following attitude determination sensors? List some of the performance characteristics for at least one specific component from each type of sensor.

   (a) Earth horizon sensors

   (b) magnetometers

   (c) rate gyros

   (d) star trackers

   (e) sun sensors

4. Which control systems rely on naturally occurring fields, and what fields are they?

5. For a circular orbit, what are the directions of the position, velocity, and orbital angular momentum vectors in terms of the orbital frame's base vectors?

6. Repeat Exercise 5 for an elliptical orbit.

7. Make a sketch of the reference frames missing from Fig. 1.3.

Chapter 2

Mission Analysis

As noted in Chapter 1, orbital and attitude dynamics must be considered as coupled. That is to say, the orbital motion of a spacecraft affects the attitude motion, and the attitude motion affects the orbital motion. The attitude $\rightarrow$ orbital coupling is not as significant as the orbital $\rightarrow$ attitude coupling. This fact is traditionally used as motivation to treat the orbit as a given motion and then investigate the attitude motion for a given orbit.

As long as the orbit is circular, the effects on attitude are reasonably straightforward to determine. If the orbit is elliptical, then the effects on the attitude dynamics are more complicated. In this chapter, I develop the basic geometric relationships necessary to investigate the effects of orbital motion on the spacecraft attitude or pointing requirements. The required topics in orbital dynamics are summarized in Appendix A, and some basic spherical geometry terms and relations are given in Appendix B.

I begin by describing the geometric quantities necessary to define pointing and mapping requirements. This space mission geometry is useful for understanding how to visualize attitude motion, and for determining important quantities such as the direction from the spacecraft to the sun. After developing these concepts, I show how a variety of uncertainties lead to errors and how estimates of these errors influence spacecraft design.

2.1 Mission Geometry

In the first approximation, an Earth-orbiting spacecraft follows an elliptical path about the Earth's mass center, and the Earth rotates about its polar axis. This approximation leads to a predictable motion of the satellite over the Earth, commonly illustrated by showing the satellite's *ground track*. Figure 2.1 shows the ground tracks for two satellites: (a) Zarya, the first component of the International Space Station, and satellite in a circular orbit with altitude 500 km, and inclination of 30°, and (b) a satellite in an elliptical orbit with periapsis altitude of 500

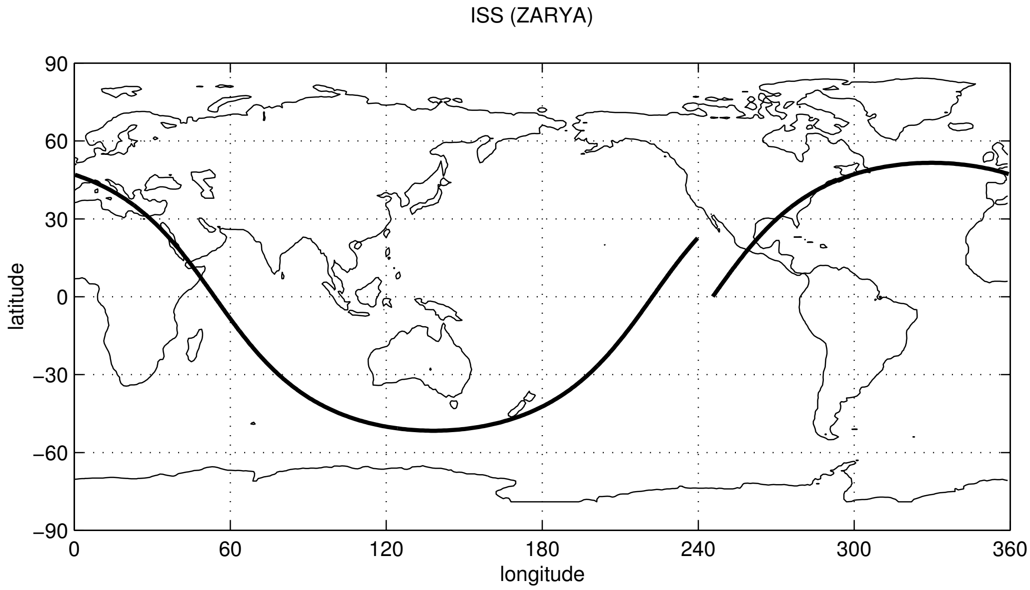

Figure 2.1: Ground Track for the International Space Station

km, apoapsis altitude of about 24,000 km, an eccentricity of 0.63, and inclination of about 26°. Similar plots can be created using Satellite Tool Kit or WinOrbit, which are available on the World Wide Web at http://www.stk.com and http://www.sat-net.com/winorbit/index.html, respectively.

2.1.1 Earth viewed from space

At any instant in time, the point on a ground track is defined as the point of intersection between the surface of the Earth and the line connecting the Earth center and the satellite. This point is called the *sub-satellite point* (SSP). The spacecraft can "see" the sub-satellite point and the area around the SSP. This area is called the *instantaneous access area* (IAA), and is always less than half of the surface area of the Earth. Figures 2.3 and 2.4 illustrate the IAA and related parameters. Figure 2.3 shows the Earth and satellite orbit. The satellite's altitude is denoted by H, so that the distance from the center of the Earth to the satellite is $R = R_\oplus + H$, where $R_\oplus$ is the radius of the Earth.

Decisions about the attitude control system must be made based on the requirements of the spacecraft mission. As described in Chapter 1, ACS requirements generally arise from the need to point the payload or some other subsystem in a particular direction. For example, a communications satellite's antenna must point at its terrestrial counterpart. Similarly, a remote sensing satellite's instrument must point at its subject. More generally, solar panels must point at the sun, and thermal radiators must point away from the sun. Some sensitive optical instruments must avoid pointing near the sun, moon, or earth, as the light from these objects could damage the instruments. Additional requirements include the range of possible pointing

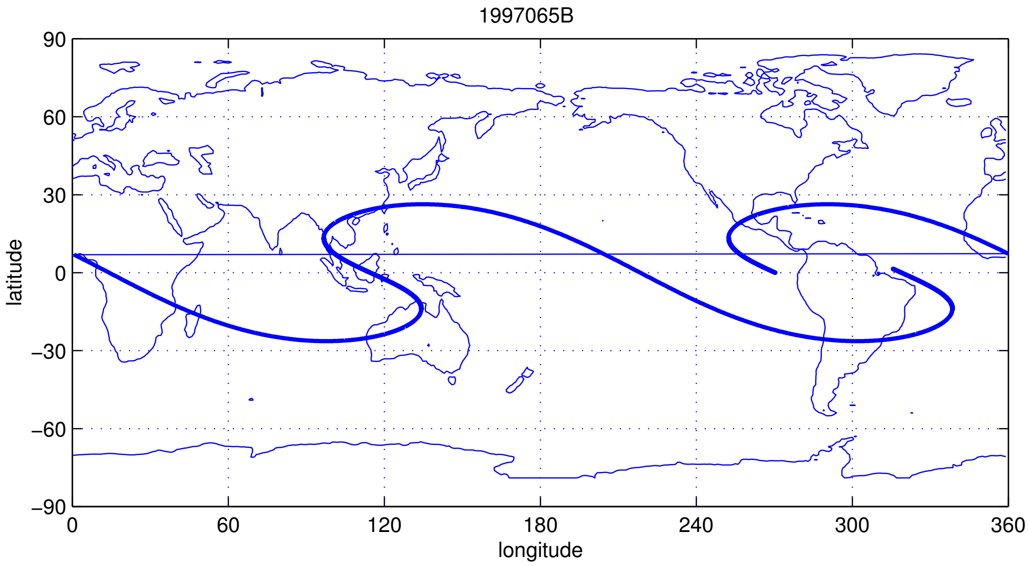

Figure 2.2: Ground Track for the Falcon Gold Satellite

directions, pointing accuracy and stability, pointing knowledge accuracy, and slew rate.

From orbital dynamics (Appendix A), we know how to follow a satellite in its orbit. That is, we can readily compute $\vec{\mathbf{r}}(t)$ and $\vec{\mathbf{v}}(t)$ for a given orbit. We can also compute the orbit ground track in a relatively straightforward manner. The following algorithm computes latitude and longitude of the sub-satellite point (SSP) as functions of time.

Algorithm 2.1
Initialize

orbital elements: a, e, i, ω, Ω, ν_0
Greenwich sidereal time at epoch: θ_{g0}
period: $P = 2\pi\sqrt{a^3/\mu}$
number of steps: N
time step: $\Delta t = P/(N-1)$

for $j = 0$ *to* $N-1$

Compute

Greenwich sidereal time: $\theta_g = \theta_{g0} + j\omega_\oplus \Delta t$
position vector: $\mathbf{r}$
latitude: $\delta_s = \sin^{-1}(r_3/r)$
longitude: $L_s = \tan^{-1}(r_2/r_1) - \theta_g$

At any point in its orbit, the spacecraft can see a circular region around the sub-satellite point. This region is known as the instantaneous access area (IAA). The IAA sweeps out a swath as the spacecraft moves in its orbit. Knowing how to determine

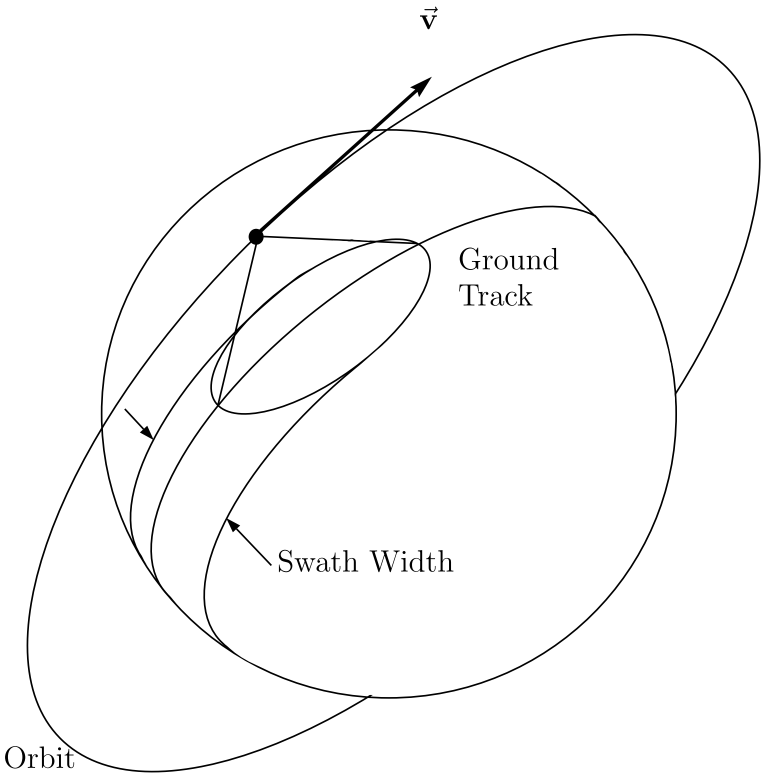

Figure 2.3: Earth Viewed by a Satellite

the SSP, let us develop some useful concepts and algorithms for spacecraft looking at the Earth.

The instantaneous access area is the area enclosed by the small circle on the sphere of the Earth, centered at the SSP and extending to the horizon as seen by the spacecraft. Two angles are evident in this geometry: the Earth central angle, λ_0, and the Earth angular radius, ρ. These are the two non-right angles of a right triangle whose vertices are the center of the Earth, the spacecraft, and the Earth horizon as seen by the spacecraft. Thus the two angles are related by*

$$\rho + \lambda_0 = 90^\circ \tag{2.1}$$

From the geometry of the figure, these angles may be computed from the relation

$$\sin\rho = \cos\lambda_0 = \frac{R_\oplus}{R_\oplus + H} \tag{2.2}$$

*We usually give common angles in degrees; however, most calculations involving angles require radians.

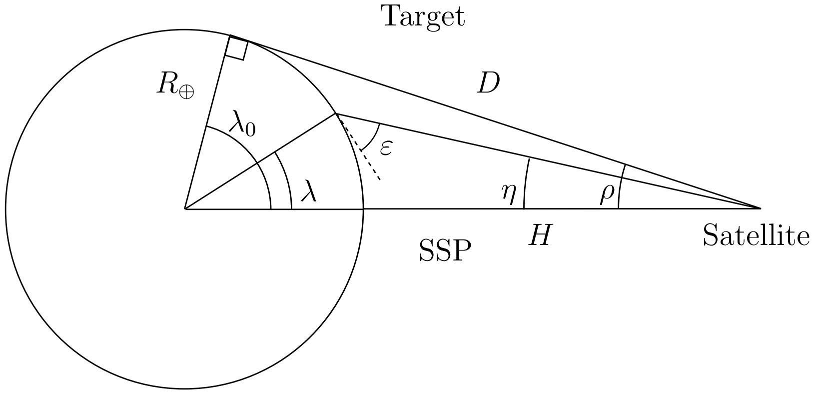

Figure 2.4: Geometry of Earth-viewing

where $R_\oplus$ is the radius of the Earth, and H is the altitude of the satellite. The IAA may be calculated as

$$\text{IAA} = K_A(1 - \cos\lambda_0) \tag{2.3}$$

where

$$\begin{array}{lcll} K_A & = & 2\pi & \text{area in steradians} \\ K_A & = & 20,626.4806 & \text{area in deg}^2 \\ K_A & = & 2.55604187 \times 10^8 & \text{area in km}^2 \\ K_A & = & 7.45222569 \times 10^7 & \text{area in nmi}^2 \end{array}$$

Generally, a spacecraft can point an instrument at any point within its IAA; however, near the horizon, a foreshortening takes place that distorts the view. The operational effects of this distortion are handled by introducing a minimum elevation angle, ε_{min} that reduces the usable IAA. Figure 2.4 shows the relationship between elevation angle, ε, Earth central angle, λ, Earth angular radius, ρ, nadir angle, η, and range, D. These variables are useful for describing the geometry of pointing a spacecraft at a particular target. The target must be within the IAA, and the spacecraft elevation angle ε is measured "up" from the target location to the satellite. Thus if the target is on the horizon, then $\varepsilon = 0$, and if the target is at the SSP, then $\varepsilon = 90°$; therefore ε is always between 0 and 90°. The angle λ is the angle between the position vectors of the spacecraft and the target, so it can be computed using

$$\vec{\mathbf{r}}_s \cdot \vec{\mathbf{r}}_t = r_s r_t \cos\lambda \tag{2.4}$$

where $\vec{\mathbf{r}}_s$ is the position vector of the spacecraft and $\vec{\mathbf{r}}_t$ is the position vector of the target. Clearly $r_s = R_\oplus + H$, and $r_t = R_\oplus$, so

$$\cos\lambda = \frac{\vec{\mathbf{r}}_s \cdot \vec{\mathbf{r}}_t}{R_\oplus(R_\oplus + H)} \tag{2.5}$$

Since we already know the latitude, δ, and longitude, L of both the SSP and the target, we can simplify these expressions, using

$$\vec{\mathbf{r}}_s = (R_\oplus + H)\left(\cos\delta_s \cos L_s \hat{\mathbf{I}}' + \cos\delta_s \sin L_s \hat{\mathbf{J}}' + \sin\delta_s \hat{\mathbf{K}}'\right) \tag{2.6}$$

$$\vec{\mathbf{r}}_t = R_\oplus\left(\cos\delta_t \cos L_t \hat{\mathbf{I}}' + \cos\delta_t \sin L_t \hat{\mathbf{J}}' + \sin\delta_t \hat{\mathbf{K}}'\right) \tag{2.7}$$

Carrying out the dot product in Eq. (2.5), collecting terms, and simplifying, leads to

$$\cos\lambda = \cos\delta_s \cos\delta_t \cos\Delta L + \sin\delta_s \sin\delta_t \tag{2.8}$$

where $\Delta L = L_s - L_t$. Clearly λ must be between 0 and 90°.

Knowing λ, the nadir angle, η, can be found from

$$\tan\eta = \frac{\sin\rho\sin\lambda}{1-\sin\rho\cos\lambda} \tag{2.9}$$

Knowing η and λ, the elevation angle can be determined from the relationship

$$\eta + \lambda + \varepsilon = 90^\circ \tag{2.10}$$

Also, the range, D, to the target may be found using

$$D = R_\oplus \frac{\sin\lambda}{\sin\eta} \tag{2.11}$$

2.2 Summary of Earth Geometry Viewed From Space

The following formulas are useful in dealing with Earth geometry as viewed by a spacecraft. In these formulas, the subsatellite point has longitude L_s and latitude δ_s, and the target point has longitude L_t and latitude δ_t. The notation δ_t' is used to denote the *colatitude* $90^\circ - \delta_t$. The angular radius of the Earth is ρ, the radius of the Earth is R_E, and the satellite altitude is H. The Earth central angle is λ, Az is the azimuth angle of the target relative to the subsatellite point, η is the nadir angle, and ε is the grazing angle or spacecraft elevation angle. This notation is used in Larson and Wertz's *Space Mission Analysis and Design*, 2nd edition, 1992, which is essentially the only reference for this material.

$$\sin\rho = R_E/(R_E + H)$$

$$\text{Spacecraft viewing angles} \quad (L_s, \delta_s, L_t, \delta_t) \mapsto (\lambda, \mathrm{Az}, \eta)$$

$$\Delta L = |L_s - L_t|$$

$$
\begin{aligned}
\cos\lambda &= \sin\delta_s\sin\delta_t + \cos\delta_s\cos\delta_t\cos\Delta L \qquad (\lambda < 180^\circ) \\
\cos \mathrm{Az} &= \frac{\sin\delta_t - \cos\lambda\sin\delta_s}{\sin\lambda\cos\delta_s} \\
\tan\eta &= \frac{\sin\rho\sin\lambda}{1 - \sin\rho\cos\lambda}
\end{aligned}
$$

$$
\begin{aligned}
\text{Earth coordinates} \quad & (L_s, \delta_s, \mathrm{Az}, \eta) \mapsto (\lambda, \delta_t, \Delta L) \\
\cos\varepsilon &= \frac{\sin\eta}{\sin\rho} \\
\lambda &= 90^\circ - \eta - \varepsilon \\
\cos\delta_t' &= \cos\lambda\sin\delta_s + \sin\lambda\cos\delta_s\cos\mathrm{Az} \qquad (\delta_t' < 180^\circ) \\
\cos\Delta L &= \frac{\cos\lambda - \sin\delta_s\sin\delta_t}{\cos\delta_s\cos\delta_t}
\end{aligned}
$$

The Instantaneous Access Area (IAA) is

$$
\mathrm{IAA} = K_A(1 - \cos\lambda_0)
$$

where

$$
\begin{array}{lcll}
K_A &=& 2\pi & \text{area in steradians} \\
K_A &=& 20,626.4806 & \text{area in deg}^2 \\
K_A &=& 2.55604187\times 10^8 & \text{area in km}^2 \\
K_A &=& 7.45222569\times 10^7 & \text{area in nmi}^2
\end{array}
$$

2.3 Error Budget

The attitude or orientation of a spacecraft usually arises as either a pointing problem requiring control, or a mapping problem requiring attitude determination. Example pointing problems would be to control the spacecraft so that a particular instrument (camera, antenna, *etc.*) points at a particular location on the surface of the Earth or at a particular astronomical object of interest. Mapping problems arise when the accurate location of a point being observed is required. In practice, most spacecraft operations involve both pointing and mapping. For example, we may command the spacecraft to point at Blacksburg and take a series of pictures. Afterwards we may need to determine the actual location of a point in one of the pictures.

In order to describe precisely the errors associated with pointing and mapping, we need to define some terms associated with the geometry of an Earth-pointing spacecraft. In this figure, the angle ϵ is the elevation, ϕ is the azimuth angle with respect to the orbital plane, η is the nadir angle, and λ is the Earth central angle. The distance from the center of the Earth to the satellite is R_s, and to the target point is R_T. We define the errors in position as ΔI, ΔC, and ΔR_s, where ΔI is the in-track error, ΔC is the cross-track error, and ΔR_s is the radial error. The instrument

Table 2.1: Sources of Pointing and Mapping Errors[2]

Spacecraft Position Errors		
ΔI	In- or along-track	Displacement along the spacecraft's velocity vector
ΔC	Cross-track	Displacement normal to the spacecraft's orbit plane
ΔR_S	Radial	Displacement toward the center of the Earth (nadir)
Sensing Axis Orientation Errors (in polar coordinates about nadir)		
$\Delta\eta$	Elevation	Error in angle from nadir to sensing axis
$\Delta\phi$	Azimuth	Error in rotation of the sensing axis about nadir
Other Errors		
ΔR_T	Target altitude	Uncertainty i the altitude of the observed object
ΔT	Clock error	Uncertainty in the real observation time

axis orientation error is described by two angles: $\Delta\eta$ and $\Delta\phi$. Two additional error sources are uncertainty of target altitude and clock error: ΔR_T and ΔT.

Based on these error sources, the approximate mapping and pointing errors are described in Tables 2.1 – 2.2. This development follows that in Larson and Wertz.[2]

The errors defined in Table 2.1 lead to mapping and pointing errors as described in Table 2.2.

2.4 References and further reading

Mission analysis is closely related to space systems design. Wertz's handbook[1] provides substantial coverage of all aspects of attitude determination and control systems. The more recent volume edited by Larson and Wertz[2] updates some of the material from Ref. 1, and gives an especially useful treatment of space mission geometry and its influence on design. Brown's text[3] focuses mostly on orbital analysis, but includes a chapter on "Observing the Central Body." The Jet Propulsion Laboratory's *Basics of Space Flight*[4] covers interplanetary mission analysis in detail, but does not provide much on attitude control systems.

Bibliography

[1] J. R. Wertz, editor. *Spacecraft Attitude Determination and Control*. D. Reidel, Dordrecht, Holland, 1978.

[2] Wiley J. Larson and James R. Wertz, editors. *Space Mission Analysis and Design*. Microcosm, Inc., Torrance, CA, second edition, 1995.

Table 2.2: Pointing and Mapping Error Formulas[2]

Source	**Magnitude**	**Magnitude of Mapping Error (km)**	**Magnitude of Pointing Error (rad)**	**Direction of Error**
Attitude Errors:				
Azimuth	$\Delta\phi$ (rad)	$\Delta\phi D \sin\eta$	$\Delta\phi \sin\eta$	Azimuthal
Nadir Angle	$\Delta\eta$ (rad)	$\Delta\eta D / \sin\varepsilon$	$\Delta\eta$	Toward nadir
Position Errors:				
In-track	ΔI (km)	$\Delta I (R_T/R_S) \cos H$	$(\Delta I/D) \sin Y_I$	Parallel to ground track
Cross-track	ΔC (km)	$\Delta C (R_T/R_S) \cos G$	$(\Delta C/D) \sin Y_C$	Perpendicular to ground track
Radial	ΔR_S (km)	$\Delta R_S \sin\eta / \sin\varepsilon$	$(\Delta R_S/D) \sin\eta$	Toward nadir
Other Errors:				
Target altitude	ΔR_T (km)	$\Delta R_T / \tan\varepsilon$	—	Toward nadir
S/C Clock	ΔT (s)	$\Delta T V_e \cos\ \mathrm{lat}$	$\Delta T (V_e/D) \cos\ \mathrm{lat} \sin J$	Parallel to Earth's equator

$\sin H = \sin\lambda \sin\phi$
$\sin G = \sin\lambda \cos\phi$
$V_e = 464$ m/s (Earth rotation velocity at equator)
$\cos Y_I = \cos\phi \sin\eta$
$\cos Y_C = \sin\phi \sin\eta$
$\cos J = \cos\phi_E \cos\varepsilon$, where ϕ_E = azimuth relative to East

[3] Charles D. Brown. *Spacecraft Mission Design.* AIAA, Reston, Virginia, second edition, 1998.

[4] Dave Doody and George Stephan. *Basics of Space Flight Learners' Workbook.* Jet Propulsion Laboratory, Pasadena, 1997. http://www.jpl.nasa.gov/basics.

2.5 Exercises

1. A satellite is in a circular Earth orbit with altitude 500 km. Determine the instantaneous coverage area if the minimum elevation angle is $\epsilon_{\text{min}} = 10°$.
2. Derive Eqs. (2.9) and (2.11), and give a geometrical explanation for Eq. (2.10).

2.6 Problems

1. Make a ground track plot for a satellite with the following two-line element set:

```
COSMOS 2278
1 23087U 94023A   98011.59348139  .00000348  00000-0  21464-3 0  5260
2 23087  71.0176  58.4285 0007185 172.8790 187.2435 14.12274429191907
```

2. The following two-line element set (TLE) is for the International Space Station:

```
ISS (ZARYA)
1 25544U 98067A   99026.49859894 -.00001822  00000-0 -18018-4 0  2532
2 25544  51.5921 190.3677 0004089  55.0982 305.0443 15.56936406 10496
```

   Detailed information about the TLE format can be found in the "handout" on the course webpage: Appendix A: Orbits. Another good source is on the web at http://celestrak.com, which is where I got the information in the appendix, and where I obtained this TLE.

   For this assignment, you should ignore Earth oblateness effects, as well as the "perturbation" terms in the TLE ($\dot{n}$, $\ddot{n}$, and B^*).

   (a) What are the date and the Eastern Standard Time of epoch?

   (b) What are the orbital elements of Zarya? (a, e, i, ω, Ω, and ν_0, with a in km, and angles in degrees)

   (c) What are the latitude and longitude of the sub-satellite point (SSP) at epoch?

   (d) What is the area, in km^2, of the instantaneous access area (IAA) at epoch? What fraction of the theoretical maximum IAA is this area?

(e) If the minimum elevation angle is $\varepsilon_{min} = 10^\circ$, then what is the "reduced" IAA at epoch?

(f) What are the latitude and longitude of the SSP the next time (after epoch) the station passes through apoapsis?

3. **Programming Project.** A useful MatLab function would compute the subsatellite point latitude and LST for a given set of orbital elements, not necessarily circular. A calling format could be

```
[lat,lst] = oe2ssp(oe,dt)
```

The argument `dt` represents a Δt from epoch and could be optional, and it should work if `dt` is a "vector." If `dt` is omitted, then the function should return latitude and LST at epoch. If `dt` is a vector, then the returned values of `lat` and `lon` should be vectors of the same length. The function should also have an optional argument for specifying the gravitational parameter μ, so that the function can be used with different systems of units. How would you need to modify this function so that it provided latitude and longitude instead of latitude and LST? What if you wanted to use it for planning missions about other planets?

Chapter 3

Kinematics

As noted in the Introduction, the study of dynamics can be decomposed into the study of kinematics and kinetics. For the translational motion of a particle of mass m, this decomposition amounts to expressing Newton's second law,

$$m\ddot{\vec{\mathbf{r}}} = \vec{\mathbf{f}} \tag{3.1}$$

a 2nd-order vector differential equation, as the two 1st-order vector differential equations

$$\dot{\vec{\mathbf{r}}} = \vec{\mathbf{p}}/m \tag{3.2}$$
$$\dot{\vec{\mathbf{p}}} = \vec{\mathbf{f}} \tag{3.3}$$

Here $\vec{\mathbf{r}}$ is the position vector of the particle relative to an inertial origin $\mathbf{O}$, $\vec{\mathbf{p}} = m\dot{\vec{\mathbf{r}}}$ (or $m\vec{\mathbf{v}}$) is the linear momentum of the particle, and $\vec{\mathbf{f}}$ is the sum of all the forces acting on the particle. Equation (3.2) is the kinematics differential equation, describing how position changes for a given velocity; *i.e.*, integration of Eq. (3.2) gives $\vec{\mathbf{r}}(t)$. Equation (3.3) is the kinetics differential equation, describing how velocity changes for a given force. It is also important to make clear that the "dot", $\dot{()}$, represents the rate of change of the vector as seen by a fixed (inertial) observer (reference frame).

In the case of rotational motion of a reference frame, the equivalent to Eq. (3.2) is not as simple to express. The purpose of this chapter is to develop the kinematic equations of motion for a rotating reference frame, as well as the conceptual tools for visualizing this motion. In Chapter 4 we describe how the attitude of a spacecraft is determined. In Chapter 5 we develop the kinetic equations of motion for a rigid body.

This chapter begins with the development of attitude representations, including reference frames, rotation matrices, and some of the variables that can be used to describe attitude motion. Then we develop the differential equations that describe attitude motion for a given angular velocity. These equations are equivalent to Eq. (3.2), which describes translational motion for a given translational velocity.

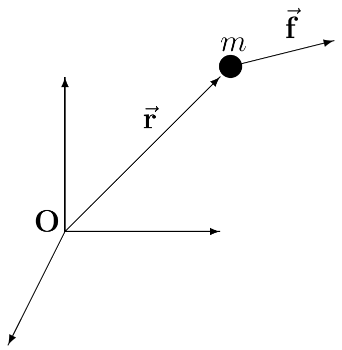

Figure 3.1: Dynamics of a particle

3.1 Attitude Representations

In this section we discuss various representations of the attitude or orientation of spacecraft. We begin by discussing reference frames, vectors, and their representations in reference frames. The problem of representing vectors in different reference frames leads to the development of rotations, rotation matrices, and various ways of representing rotation matrices, including Euler angles, Euler parameters, and quaternions.

3.1.1 Reference Frames

A *reference frame*, or *coordinate system*, is generally taken to be a set of three unit vectors that are mutually perpendicular. An equivalent definition is that a reference frame is a *triad* of *orthonormal* vectors. Triad of course means three, and orthonormal means *ortho*gonal and *normal*. The term orthogonal is nearly synonymous with the term perpendicular, but has a slightly more general meaning when dealing with other sorts of vectors (which we do not do here). The fact that the vectors are normalized means that they are *unit* vectors, or that their lengths are all unity (1) in the units of choice. We also usually use *right-handed* or *dextral* reference frames, which simply means that we order the three vectors in an agreed-upon fashion, as described below.

The reason that reference frames are so important in attitude dynamics is that following the orientation of a reference frame is completely equivalent to following the orientation of a rigid body. Although no spacecraft is perfectly rigid, the rigid body model is a good first approximation for studying attitude dynamics. Similarly, no spacecraft (or planet) is a point mass, but the point mass model is a good first approximation for studying orbital dynamics.

We normally use a triad of unit vectors, denoted by the same letter, with subscripts 1,2,3. For example, an *inertial* frame would be denoted by $\left\{\hat{\mathbf{i}}_1, \hat{\mathbf{i}}_2, \hat{\mathbf{i}}_3\right\}$, an *orbital* frame by $\{\hat{\mathbf{o}}_1, \hat{\mathbf{o}}_2, \hat{\mathbf{o}}_3\}$, and a body- (or spacecraft-) fixed frame by $\left\{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\right\}$. The hats are used to denote that these are unit vectors. We also use the notation $\mathcal{F}_i$, $\mathcal{F}_o$, and $\mathcal{F}_b$, to represent these and other reference frames.

The orthonormal property of a reference frame's base vectors is defined by the dot products of the vectors with each other. Specifically, for a set of orthonormal base vectors, the dot products satisfy

$$\begin{array}{lll} \hat{\mathbf{i}}_1 \cdot \hat{\mathbf{i}}_1 = 1 & \hat{\mathbf{i}}_1 \cdot \hat{\mathbf{i}}_2 = 0 & \hat{\mathbf{i}}_1 \cdot \hat{\mathbf{i}}_3 = 0 \\ \hat{\mathbf{i}}_2 \cdot \hat{\mathbf{i}}_1 = 0 & \hat{\mathbf{i}}_2 \cdot \hat{\mathbf{i}}_2 = 1 & \hat{\mathbf{i}}_2 \cdot \hat{\mathbf{i}}_3 = 0 \\ \hat{\mathbf{i}}_3 \cdot \hat{\mathbf{i}}_1 = 0 & \hat{\mathbf{i}}_3 \cdot \hat{\mathbf{i}}_2 = 0 & \hat{\mathbf{i}}_3 \cdot \hat{\mathbf{i}}_3 = 1 \end{array} \tag{3.4}$$

which may be written more concisely as

$$\hat{\mathbf{i}}_i \cdot \hat{\mathbf{i}}_j = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases} \tag{3.5}$$

or even more concisely as

$$\hat{\mathbf{i}}_i \cdot \hat{\mathbf{i}}_j = \delta_{ij} \tag{3.6}$$

where δ_{ij} is the Kronecker delta, for which Eq. (3.5) may be taken as the definition.

We often find it convenient to collect the unit vectors of a reference frame into a 3×1 column matrix of vectors, and we denote this object by

$$\left\{\hat{\mathbf{i}}\right\} = \left\{ \begin{array}{c} \hat{\mathbf{i}}_1 \\ \hat{\mathbf{i}}_2 \\ \hat{\mathbf{i}}_3 \end{array} \right\} \tag{3.7}$$

This matrix is a rather special object, as its components are unit vectors instead of scalars. Hughes[1] introduced the term *vectrix* to describe this vector matrix. Using this notation, Eq. (3.4) can be written as

$$\left\{\hat{\mathbf{i}}\right\} \cdot \left\{\hat{\mathbf{i}}\right\}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{1} \tag{3.8}$$

which is the 3×3 identity matrix. The superscript T on $\left\{\hat{\mathbf{i}}\right\}$ transposes the matrix from a column matrix (3×1) to a row matrix (1×3).

The *right-handed* or *dextral* property of a reference frame's base vectors is defined by the cross products of the vectors with each other. Specifically, for a right-handed set of *orthonormal* base vectors, the cross products satisfy

$$\begin{array}{lll} \hat{\mathbf{i}}_1 \times \hat{\mathbf{i}}_1 = \vec{\mathbf{0}} & \hat{\mathbf{i}}_1 \times \hat{\mathbf{i}}_2 = \hat{\mathbf{i}}_3 & \hat{\mathbf{i}}_1 \times \hat{\mathbf{i}}_3 = -\hat{\mathbf{i}}_2 \\ \hat{\mathbf{i}}_2 \times \hat{\mathbf{i}}_1 = -\hat{\mathbf{i}}_3 & \hat{\mathbf{i}}_2 \times \hat{\mathbf{i}}_2 = \vec{\mathbf{0}} & \hat{\mathbf{i}}_2 \times \hat{\mathbf{i}}_3 = \hat{\mathbf{i}}_1 \\ \hat{\mathbf{i}}_3 \times \hat{\mathbf{i}}_1 = \hat{\mathbf{i}}_2 & \hat{\mathbf{i}}_3 \times \hat{\mathbf{i}}_2 = -\hat{\mathbf{i}}_1 & \hat{\mathbf{i}}_3 \times \hat{\mathbf{i}}_3 = \vec{\mathbf{0}} \end{array} \tag{3.9}$$

This set of rules may be written more concisely as

$$\hat{\mathbf{i}}_i \times \hat{\mathbf{i}}_j = \varepsilon_{ijk}\hat{\mathbf{i}}_k \tag{3.10}$$

where ε_{ijk} is the *permutation symbol*, defined as

$$\varepsilon_{ijk} = \begin{cases} 1 & \text{for } i,j,k \text{ an even permutation of 1,2,3} \\ -1 & \text{for } i,j,k \text{ an odd permutation of 1,2,3} \\ 0 & \text{otherwise } (\textit{i.e.}, \text{ if any repetitions occur}) \end{cases} \tag{3.11}$$

Equation (3.9) can also be written as

$$\left\{\hat{\mathbf{i}}\right\} \times \left\{\hat{\mathbf{i}}\right\}^T = \left\{ \begin{array}{ccc} \vec{\mathbf{0}} & \hat{\mathbf{i}}_3 & -\hat{\mathbf{i}}_2 \\ -\hat{\mathbf{i}}_3 & \vec{\mathbf{0}} & \hat{\mathbf{i}}_1 \\ \hat{\mathbf{i}}_2 & -\hat{\mathbf{i}}_1 & \vec{\mathbf{0}} \end{array} \right\} = -\left\{\hat{\mathbf{i}}\right\}^\times \tag{3.12}$$

The $\times$ superscript is used to denote a skew-symmetric 3×3 matrix associated with a 3×1 column matrix. Specifically, if $\mathbf{a}$ is a 3×1 matrix of scalars a_i, then

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \Rightarrow \mathbf{a}^\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{3.13}$$

Note that $\mathbf{a}^\times$ satisfies the skew-symmetry property $(\mathbf{a}^\times)^T = -\mathbf{a}^\times$.

3.1.2 Vectors

A *vector* is an abstract mathematical object with two properties: direction and length (or magnitude). Vector quantities that are important in this course include, for example, angular momentum, $\vec{\mathbf{h}}$, angular velocity, $\vec{\omega}$, and the direction to the sun, $\hat{\mathbf{s}}$. Vectors are denoted by a bold letter, with an arrow (hat if a unit vector), and are usually lower case.

Vectors can be expressed in any reference frame. For example, a vector, $\vec{\mathbf{v}}$, may be written in the inertial frame as

$$\vec{\mathbf{v}} = v_1\hat{\mathbf{i}}_1 + v_2\hat{\mathbf{i}}_2 + v_3\hat{\mathbf{i}}_3 \tag{3.14}$$

The scalars, v_1, v_2, and v_3, are the *components* of $\vec{\mathbf{v}}$ expressed in $\mathcal{F}_i$. These components are the dot products of the vector $\vec{\mathbf{v}}$ with the three base vectors of $\mathcal{F}_i$. Specifically,

$$v_1 = \vec{\mathbf{v}} \cdot \hat{\mathbf{i}}_1, \quad v_2 = \vec{\mathbf{v}} \cdot \hat{\mathbf{i}}_2, \quad v_3 = \vec{\mathbf{v}} \cdot \hat{\mathbf{i}}_3 \tag{3.15}$$

Since the $\hat{\mathbf{i}}$ vectors are unit vectors, these components may also be written as

$$v_1 = v\cos\alpha_1, \quad v_2 = v\cos\alpha_2, \quad v_3 = v\cos\alpha_3 \tag{3.16}$$

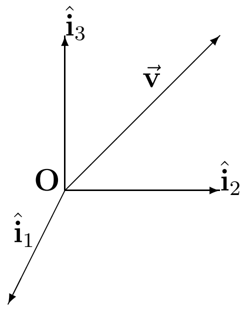

Figure 3.2: Components of a vector

where $v = \|\vec{\mathbf{v}}\|$ is the magnitude or length of $\vec{\mathbf{v}}$, and α_j is the angle between $\vec{\mathbf{v}}$ and $\hat{\mathbf{i}}_j$ for $j = 1, 2, 3$. These cosines are also called the *direction cosines* of $\vec{\mathbf{v}}$ with respect to $\mathcal{F}_i$. We frequently collect the components of a vector $\vec{\mathbf{v}}$ into a column matrix $\mathbf{v}$, with three rows and one column:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{3.17}$$

A bold letter without an overarrow (or hat) denotes such a matrix. Sometimes it is necessary to denote the appropriate reference frame, in which case we use $\mathbf{v}_i$, $\mathbf{v}_o$, $\mathbf{v}_b$, *etc.*

A handy way to write a vector in terms of its components and the base vectors is to write it as the product of two matrices, one the component matrix, and the other a matrix containing the base unit vectors. For example,

$$\vec{\mathbf{v}} = [v_1 \quad v_2 \quad v_3] \left\{ \begin{array}{c} \hat{\mathbf{i}}_1 \\ \hat{\mathbf{i}}_2 \\ \hat{\mathbf{i}}_3 \end{array} \right\} = \mathbf{v}_i^T \left\{ \hat{\mathbf{i}} \right\} \tag{3.18}$$

Recall that the subscript i denotes that the components are with respect to $\mathcal{F}_i$.

Using this notation, we can write $\vec{\mathbf{v}}$ in terms of different frames as

$$\vec{\mathbf{v}} = \mathbf{v}_i^T \left\{ \hat{\mathbf{i}} \right\} = \mathbf{v}_o^T \left\{ \hat{\mathbf{o}} \right\} = \mathbf{v}_b^T \left\{ \hat{\mathbf{b}} \right\} \tag{3.19}$$

and so forth.

There are two types of reference frame problems we encounter in this course. The first involves determining the components of a vector in one frame (say $\mathcal{F}_i$) when the components in another frame (say $\mathcal{F}_b$) are known, and the relative orientation of the two frames is known. The second involves determining the components of a vector in a frame that has been reoriented or rotated. Both problems involve *rotations*, which are the subject of the next section.

3.1.3 Rotations

Suppose we have a vector $\vec{\mathbf{v}}$, and we know its components in $\mathcal{F}_b$, denoted $\mathbf{v}_b$, and we want to determine its components in $\mathcal{F}_i$, denoted $\mathbf{v}_i$. Since

$$\vec{\mathbf{v}} = \mathbf{v}_i^T \left\{\hat{\mathbf{i}}\right\} = \mathbf{v}_b^T \left\{\hat{\mathbf{b}}\right\} \tag{3.20}$$

we seek a way to express $\left\{\hat{\mathbf{i}}\right\}$ in terms of $\left\{\hat{\mathbf{b}}\right\}$, say

$$\left\{\hat{\mathbf{i}}\right\} = \mathbf{R}\left\{\hat{\mathbf{b}}\right\} \tag{3.21}$$

where $\mathbf{R}$ is a 3×3 *transformation* matrix. Then we can write

$$\vec{\mathbf{v}} = \mathbf{v}_i^T \left\{\hat{\mathbf{i}}\right\} = \mathbf{v}_i^T\mathbf{R}\left\{\hat{\mathbf{b}}\right\} = \mathbf{v}_b^T \left\{\hat{\mathbf{b}}\right\} \tag{3.22}$$

Comparing the last two terms in this equation, we see that

$$\mathbf{v}_i^T\mathbf{R} = \mathbf{v}_b^T \tag{3.23}$$

Transposing both sides*, we get

$$\mathbf{R}^T\mathbf{v}_i = \mathbf{v}_b \tag{3.24}$$

Thus, to compute $\mathbf{v}_i$, we just need to determine $\mathbf{R}$ and solve the linear system of equations defined by Eq. (3.24).

If we write the components of $\mathbf{R}$ as R_{ij}, where i denotes the row and j denotes the column, then Eq. (3.21) may be expanded to

$$\hat{\mathbf{i}}_1 = R_{11}\hat{\mathbf{b}}_1 + R_{12}\hat{\mathbf{b}}_2 + R_{13}\hat{\mathbf{b}}_3 \tag{3.25}$$
$$\hat{\mathbf{i}}_2 = R_{21}\hat{\mathbf{b}}_1 + R_{22}\hat{\mathbf{b}}_2 + R_{23}\hat{\mathbf{b}}_3 \tag{3.26}$$
$$\hat{\mathbf{i}}_3 = R_{31}\hat{\mathbf{b}}_1 + R_{32}\hat{\mathbf{b}}_2 + R_{33}\hat{\mathbf{b}}_3 \tag{3.27}$$

Comparing these expressions with Eqs. (3.14–3.16), it is evident that $R_{11} = \hat{\mathbf{i}}_1 \cdot \hat{\mathbf{b}}_1$, $R_{12} = \hat{\mathbf{i}}_1 \cdot \hat{\mathbf{b}}_2$, and in general, $R_{ij} = \hat{\mathbf{i}}_i \cdot \hat{\mathbf{b}}_j$. Using direction cosines, we can write $R_{11} = \cos\alpha_{11}$, $R_{12} = \cos\alpha_{12}$, and in general, $R_{ij} = \cos\alpha_{ij}$, where α_{ij} is the angle between $\hat{\mathbf{i}}_i$ and $\hat{\mathbf{b}}_j$. Thus $\mathbf{R}$ is a matrix of direction cosines, and is frequently referred to

*Recall that to transpose a product of matrices, you reverse the order and transpose each matrix. Thus, $(\mathbf{A}\mathbf{B}^T\mathbf{C})^T = \mathbf{C}^T\mathbf{B}\mathbf{A}^T$.

as the DCM (direction cosine matrix). As with Eq. (3.8), where we have $\left\{\hat{\mathbf{i}}\right\}\cdot\left\{\hat{\mathbf{i}}\right\}^T = \mathbf{1}$, we can also write $\mathbf{R}$ as the dot product of $\left\{\hat{\mathbf{i}}\right\}$ with $\left\{\hat{\mathbf{b}}\right\}^T$, *i.e.*,

$$\mathbf{R} = \left\{\hat{\mathbf{i}}\right\} \cdot \left\{\hat{\mathbf{b}}\right\}^T \tag{3.28}$$

If we know the relative orientation of the two frames, then we can compute the matrix $\mathbf{R}$, and solve Eq. (3.24) to get $\mathbf{v}_i$. As it turns out, it is quite simple to solve this linear system, because the inverse of a direction cosine matrix is simply its transpose. That is,

$$\mathbf{R}^{-1} = \mathbf{R}^T \tag{3.29}$$

To discover this fact, note that it is simple to show that

$$\left\{\hat{\mathbf{b}}\right\} = \mathbf{R}^T \left\{\hat{\mathbf{i}}\right\} \tag{3.30}$$

using the same $\mathbf{R}$ as in Eq. (3.21). Comparing this result with Eq. (3.21), it is clear that Eq. (3.29) is true. A matrix with this property is said to be *orthonormal*, because its rows (and columns) are orthogonal to each other and they all represent unit vectors. This property applied to Eq. (3.24) leads to

$$\mathbf{v}_i = \mathbf{R}\mathbf{v}_b \tag{3.31}$$

Thus $\mathbf{R}$ is the transformation matrix that takes vectors expressed in $\mathcal{F}_b$ and transforms or rotates them into $\mathcal{F}_i$, and $\mathbf{R}^T$ is the transformation that takes vectors expressed in $\mathcal{F}_i$ and transforms them into $\mathcal{F}_b$. We use the notation $\mathbf{R}^{bi}$ to represent the rotation matrix from $\mathcal{F}_i$ to $\mathcal{F}_b$, and $\mathbf{R}^{ib}$ to represent the rotation matrix from $\mathcal{F}_b$ to $\mathcal{F}_i$. Thus

$$\mathbf{v}_b = \mathbf{R}^{bi}\mathbf{v}_i \quad \text{and} \quad \mathbf{v}_i = \mathbf{R}^{ib}\mathbf{v}_b \tag{3.32}$$

The intent of the ordering of b and i in the superscripts is to place the appropriate letter closest to the components of the vector in that frame. The ordering of the superscripts is also related to the rows and columns of $\mathbf{R}$. The first superscript corresponds to the reference frame whose base vector components are in the rows of $\mathbf{R}$, and the second superscript corresponds to the frame whose base vector components are in the columns of $\mathbf{R}$. Similarly the superscripts correspond directly to the dot product notation of Eq. (3.28); *i.e.*, $\mathbf{R}^{ib} = \left\{\hat{\mathbf{i}}\right\} \cdot \left\{\hat{\mathbf{b}}\right\}^T$, and $\mathbf{R}^{bi} = \left\{\hat{\mathbf{b}}\right\} \cdot \left\{\hat{\mathbf{i}}\right\}^T$.

Looking again at Eqs. (3.25–3.27), it is clear that the rows of $\mathbf{R}$ are the components of the corresponding $\hat{\mathbf{i}}_i$, expressed in $\mathcal{F}_b$, whereas the columns of $\mathbf{R}$ are the components of the corresponding $\hat{\mathbf{b}}_j$, expressed in $\mathcal{F}_i$. To help remember this relationship, we write the rotation matrix $\mathbf{R}^{ib}$ as follows:

$$\mathbf{R}^{ib} = \begin{bmatrix} \mathbf{i}_{1b}^T \\ \mathbf{i}_{2b}^T \\ \mathbf{i}_{3b}^T \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{1i} & \mathbf{b}_{2i} & \mathbf{b}_{3i} \end{bmatrix} \tag{3.33}$$

Although the discussion here has centered on frames $\left\{\hat{\mathbf{b}}\right\}$ and $\left\{\hat{\mathbf{i}}\right\}$, the development is the same for any two reference frames.

3.1.4 Euler Angles

Computing the nine direction cosines of the DCM is one way to construct a rotation matrix, but there are many others. One of the easiest to visualize is the Euler angle approach. Euler† reasoned that any rotation from one frame to another can be visualized as a sequence of three *simple rotations* about base vectors. Let us consider the rotation from $\mathcal{F}_i$ to $\mathcal{F}_b$ through a sequence of three angles θ_1, θ_2, and θ_3.

We begin with a simple rotation about the $\hat{\mathbf{i}}_3$ axis, through the angle θ_1. We denote the resulting reference frame as $\mathcal{F}_{i'}$, or $\{\hat{\mathbf{i}}'\}$. Using the rules developed above for constructing $\mathbf{R}^{i'i}$, it is easy to show that the correct rotation matrix is

$$\mathbf{R}^{i'i} = \mathbf{R}_3(\theta_1) = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.34}$$

so that

$$\mathbf{v}_{i'} = \mathbf{R}_3(\theta_1)\mathbf{v}_i \tag{3.35}$$

The subscript 3 in $\mathbf{R}_3(\theta_1)$ denotes that this rotation matrix is a "3" rotation about the "3" axis. Note that we could have performed the first rotation about $\hat{\mathbf{i}}_1$ (a "1" rotation) or $\hat{\mathbf{i}}_2$ (a "2" rotation). Thus there are three possibilities for the first simple rotation in an Euler angle sequence. For the second simple rotation, we cannot choose $\hat{\mathbf{i}}'_3$, since this choice would amount to simply adding to θ_1. Thus there are only two choices for the second simple rotation.

We choose $\hat{\mathbf{i}}'_2$ as the second rotation axis, rotate through an angle θ_2, and call the resulting frame $\mathcal{F}_{i''}$, or $\{\hat{\mathbf{i}}''\}$. In this case, the rotation matrix is

$$\mathbf{R}^{i''i'} = \mathbf{R}_2(\theta_2) = \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \tag{3.36}$$

so that

$$\mathbf{v}_{i''} = \mathbf{R}_2(\theta_2)\mathbf{v}_{i'} = \mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)\mathbf{v}_i \tag{3.37}$$

Now $\mathbf{R}^{i''i} = \mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)$ is the rotation matrix transforming vectors from $\mathcal{F}_i$ to $\mathcal{F}_{i''}$.

For the third, and final, rotation, we can use either a "1" rotation or a "3" rotation. We choose a "1" rotation through an angle θ_3, and denote the resulting reference frame

†Leonhard Euler (1707–1783) was a Swiss mathematician and physicist who was associated with the Berlin Academy during the reign of Frederick the Great and with the St Petersburg Academy during the reign of Catherine II. In addition to his many contributions on the motion of rigid bodies, he was a major contributor in the fields of geometry and calculus. Many of our familiar mathematical notations are due to Euler, including e for the natural logarithm base, $f()$ for functions, i for $\sqrt{-1}$, π for π, and Σ for summations. One of my favorites is the special case of Euler's formula: $e^{i\pi} + 1 = 0$, which relates 5 fundamental numbers from mathematics.

$\mathcal{F}_b$, or $\{\hat{\mathbf{b}}\}$. The rotation matrix is

$$\mathbf{R}^{bi''} = \mathbf{R}_1(\theta_3) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{bmatrix} \tag{3.38}$$

so that

$$\mathbf{v}_b = \mathbf{R}_1(\theta_3)\mathbf{v}_{i''} = \mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)\mathbf{v}_i \tag{3.39}$$

Now the matrix transforming vectors from $\mathcal{F}_i$ to $\mathcal{F}_b$ is $\mathbf{R}^{bi} = \mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)$.

For a given rotational motion of a reference frame, if we can keep track of the three Euler angles, then we can track the changing orientation of the frame.

As a final note on Euler angle sequences, recall that there were three axes to choose from for the first rotation, two to choose from for the second rotation, and two to choose from for the third rotation. Thus there are twelve $(3 \times 2 \times 2)$ possible sequences of Euler angles. These are commonly referred to by the axes that are used. For example, the sequence used above is called a "3-2-1" sequence, because we first rotate about the "3" axis, then about the "2" axis, and finally about the "1" axis. It is also possible for the third rotation to be of the same type as the first. Thus we could use a "3-2-3" sequence. This type of sequence (commonly called a symmetric Euler angle set) leads to difficulties when θ_2 is small, and so is not widely used in vehicle dynamics applications.

Example 3.1 *Let us develop the rotation matrix relating the Earth-centered inertial (ECI) frame $\mathcal{F}_i$ and the orbital frame $\mathcal{F}_o$. We consider the case of a circular orbit, with* right ascension of the ascending node *(or RAAN), Ω,* inclination*, i, and* argument of latitude*, u. Recall that argument of latitude is the angle from the ascending node to the position of the satellite, and is especially useful for circular orbits, since* argument of periapsis*, ω, is not defined for circular orbits.*

We denote the ECI frame ($\mathcal{F}_i$) by $\{\hat{\mathbf{i}}\}$, and the orbital frame ($\mathcal{F}_o$) by $\{\hat{\mathbf{o}}\}$. Intermediate frames are designated using primes, as in the Euler angle development above. We use a "3-1-3" sequence as follows: Begin with a "3" rotation about the inertial $\hat{\mathbf{i}}_3$ axis through the RAAN, Ω. This rotation is followed by a "1" rotation about the $\hat{\mathbf{i}}'_1$ axis through the inclination, i. The last rotation is another "3" rotation about the $\hat{\mathbf{i}}''_3$ axis through the argument of latitude, u.

We denote the resulting reference frame by $\{\hat{\mathbf{o}}'\}$, since it is not quite the desired orbital reference frame. Recall that the orbital reference frame for a circular orbit has its three vectors aligned as follows: $\{\hat{\mathbf{o}}_1\}$ is in the direction of the orbital velocity vector (the $\vec{\mathbf{v}}$ direction), $\{\hat{\mathbf{o}}_2\}$ is in the direction opposite to the orbit normal (the $-\vec{\mathbf{h}}$ direction), and $\{\hat{\mathbf{o}}_3\}$ is in the nadir direction (or the $-\vec{\mathbf{r}}$ direction). However, the frame resulting from the "3-1-3" rotation developed above has its unit vectors aligned in the $\vec{\mathbf{r}}$, $\vec{\mathbf{v}}$, and $\vec{\mathbf{h}}$ directions, respectively.

Now, it is possible to go back and choose angles so that the "3-1-3" rotation gives the desired orbital frame; however, it is instructive to see how to use two more rotations to get from the $\{\hat{\mathbf{o}}'\}$ *frame to the* $\{\hat{\mathbf{o}}\}$ *frame. Specifically, if we perform another "3" rotation about* $\{\hat{\mathbf{o}}_3'\}$ *through* $90°$ *and a "1" rotation about* $\{\hat{\mathbf{o}}_1''\}$ *through* $270°$*, we arrive at the desired orbital reference frame. These final two rotations lead to an interesting rotation matrix:*

$$\mathbf{R}^{oo'} = \mathbf{R}_1(270°)\mathbf{R}_3(90°) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \tag{3.40}$$

Careful study of this rotation matrix reveals that its effect is to move the second row to the first row, negate the third row and move it to the second row, and negate the first row and move it to the third row.

So, the rotation matrix that takes vectors from the inertial frame to the orbital frame is

$$\mathbf{R}^{oi} = \mathbf{R}^{oo'}\mathbf{R}_3(u)\mathbf{R}_1(i)\mathbf{R}_3(\Omega) \tag{3.41}$$

which, when expanded, gives

$$\mathbf{R}^{oi} = \begin{bmatrix} -su\,c\Omega - cu\,ci\,s\Omega & -su\,s\Omega + cu\,ci\,c\Omega & cu\,si \\ -si\,s\Omega & si\,c\Omega & -ci \\ -cu\,c\Omega + su\,ci\,s\Omega & -cu\,s\Omega - su\,ci\,c\Omega & -su\,si \end{bmatrix} \tag{3.42}$$

where we have used the letters c *and* s *as abbreviations for* cos *and* sin*, respectively.*

Now, we also need to be able to extract Euler angles from a given rotation matrix. This exercise requires careful consideration of the elements of the rotation matrix and careful application of various inverse trigonometric functions.

Thus, suppose we are given a specific rotation matrix with nine specific numbers. We can extract the three angles associated with $\mathbf{R}^{oi}$ *as developed above as follows:*

$$i = \cos^{-1}(-R_{23}) \tag{3.43}$$

$$u = \tan^{-1}(-R_{33}/R_{13}) \tag{3.44}$$

$$\Omega = \tan^{-1}(-R_{21}/R_{22}) \tag{3.45}$$

3.1.5 Euler's Theorem, Euler Parameters, and Quaternions

The Euler angle sequence approach to describing the relative orientation of two frames is reasonably easy to develop and to visualize, but it is not the most useful approach for spacecraft dynamics. Another of Euler's contributions is the theorem that tells us that only one rotation is necessary to reorient one frame to another. This theorem is known as Euler's Theorem and is formally stated as

> **Euler's Theorem.** The most general motion of a rigid body with a fixed point is a rotation about a fixed axis.

Thus, instead of using three simple rotations (and three angles) to keep track of rotational motion, we only need to use a single rotation (and a single angle) about the "fixed axis" mentioned in the theorem. At first glance, it might appear that we are getting something for nothing, since we are going from three angles to one; however, we also have to know the axis of rotation. This axis, denoted $\hat{\mathbf{a}}$, is called the *Euler axis*, or the *eigenaxis*, and the angle, denoted Φ, is called the *Euler angle*, or the *Euler principal angle*.

For a rotation from $\mathcal{F}_i$ to $\mathcal{F}_b$, about axis $\hat{\mathbf{a}}$ through angle Φ, it is possible to express the rotation matrix $\mathbf{R}^{bi}$, in terms of $\hat{\mathbf{a}}$ and Φ, just as we expressed $\mathbf{R}^{bi}$ in terms of the Euler angles in the previous section. Note that since the rotation is about $\hat{\mathbf{a}}$, the Euler axis vector has the same components in $\mathcal{F}_i$ and $\mathcal{F}_b$; that is,

$$\mathbf{R}^{bi}\mathbf{a} = \mathbf{a} \tag{3.46}$$

and the subscript notation ($\mathbf{a}_i$ or $\mathbf{a}_b$) is not needed. We leave it as an exercise to show that

$$\mathbf{R}^{bi} = \cos\Phi\mathbf{1} + (1-\cos\Phi)\mathbf{a}\mathbf{a}^T - \sin\Phi\mathbf{a}^\times \tag{3.47}$$

where $\mathbf{a}$ is the column matrix of the components of $\hat{\mathbf{a}}$ in either $\mathcal{F}_i$ or $\mathcal{F}_b$. Equation (3.46) provides the justification for the term *eigenaxis* for the Euler axis, since this equation defines $\mathbf{a}$ as the eigenvector of $\mathbf{R}^{bi}$ associated with the eigenvalue 1. A corollary to Euler's Theorem is that every rotation matrix has one eigenvalue that is unity.

Given an Euler axis, $\hat{\mathbf{a}}$, and Euler angle, Φ, we can easily compute the rotation matrix, $\mathbf{R}^{bi}$. We also need to be able to compute the component matrix, $\mathbf{a}$, and the angle Φ, for a given rotation matrix, $\mathbf{R}$. One can show that

$$\begin{aligned}
\Phi &= \cos^{-1}\left[\frac{1}{2}\left(\text{trace}\,\mathbf{R} - 1\right)\right] && (3.48)\\
\mathbf{a}^\times &= \frac{1}{2\sin\Phi}\left(\mathbf{R}^T - \mathbf{R}\right) && (3.49)
\end{aligned}$$

So, we can write a rotation matrix in terms of Euler angles, or in terms of the Euler axis/angle set. There are several other approaches, or *parameterizations of the attitude*, and we introduce one of the most important of these: *Euler parameters*, or *quaternions*.

We define four new variables in terms of $\mathbf{a}$ and Φ.

$$\begin{aligned}
\mathbf{q} &= \mathbf{a}\sin\frac{\Phi}{2} && (3.50)\\
q_4 &= \cos\frac{\Phi}{2} && (3.51)
\end{aligned}$$

The 3×1 matrix $\mathbf{q}$ forms the *Euler axis component* of the quaternion, also called the vector component. The scalar q_4 is called the *scalar component*. Collectively, these four variables are known as a *quaternion*, or as the *Euler parameters*. We use the notation $\bar{\mathbf{q}}$ to denote the 4×1 matrix containing all four variables; that is,

$$\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{q}^T & q_4 \end{bmatrix}^T \tag{3.52}$$

A given $\mathbf{a}$ and Φ correspond to a particular relative orientation of two reference frames. Thus a given $\bar{\mathbf{q}}$ also corresponds to a particular orientation. It is relatively easy to show that the rotation matrix can be written as

$$\mathbf{R} = \left(q_4^2 - \mathbf{q}^T\mathbf{q}\right)\mathbf{1} + 2\mathbf{q}\mathbf{q}^T - 2q_4\mathbf{q}^\times \tag{3.53}$$

We also need to express $\bar{\mathbf{q}}$ in terms of the elements of $\mathbf{R}$:

$$q_4 = \pm\frac{1}{2}\sqrt{1 + \operatorname{trace}\mathbf{R}} \tag{3.54}$$

$$\mathbf{q} = \frac{1}{4q_4}\begin{bmatrix} R_{23} - R_{32} \\ R_{31} - R_{13} \\ R_{12} - R_{21} \end{bmatrix} \tag{3.55}$$

We now have three basic ways to parameterize a rotation matrix: Euler angles, Euler axis/angle, and Euler parameters. Surprisingly there are many other parameterizations, some of which are not named after Euler. However, these three suffice for the topics in this course. To summarize, a rotation matrix can be written as

$$\mathbf{R} = \mathbf{R}_i(\theta_3)\mathbf{R}_j(\theta_2)\mathbf{R}_k(\theta_1) \tag{3.56}$$

$$\mathbf{R} = \cos\Phi\mathbf{1} + (1 - \cos\Phi)\mathbf{a}\mathbf{a}^T - \sin\Phi\mathbf{a}^\times \tag{3.57}$$

$$\mathbf{R} = \left(q_4^2 - \mathbf{q}^T\mathbf{q}\right)\mathbf{1} + 2\mathbf{q}\mathbf{q}^T - 2q_4\mathbf{q}^\times \tag{3.58}$$

The subscripts i, j, k in the Euler angle formulation indicate that any of the twelve Euler angle sequences may be used. That is, using set notation, $k \in \{1, 2, 3\}$, $j \in \{1, 2, 3\}\backslash k$, and $i \in \{1, 2, 3\}\backslash j$.

Before leaving this topic, we need to establish the following rule:

Rotations do not add like vectors.

The Euler axis/angle description of attitude suggests the possibility of representing a rotation by the vector quantity $\Phi\hat{\mathbf{a}}$. Then, if we had two sequential rotations, say $\Phi_1\hat{\mathbf{a}}_1$ and $\Phi_2\hat{\mathbf{a}}_2$, then we might represent the net rotation by the vector sum of these two: $\Phi_1\hat{\mathbf{a}}_1 + \Phi_2\hat{\mathbf{a}}_2$. This operation is not valid, as the following example illustrates. Suppose that Φ_1 and Φ_2 are both 90°, then the vector sum of the two supposed "rotation vectors" would be $\pi/2(\hat{\mathbf{a}}_1 + \hat{\mathbf{a}}_2)$. Since vector addition is commutative, the resulting "rotation vector" does not depend on the order of performing the two rotations. However, it is easy to see that the actual rotation resulting from the two individual rotations does depend on the order of the rotations. Thus the "rotation vector" description of attitude motion is not valid.

3.2 Attitude Kinematics

In the previous sections, we developed several different ways to describe the attitude, or orientation, of one reference frame with respect to another, in terms of attitude variables. The comparison and contrast of rotational and translational motion is summarized in Table 3.1. The purpose of this section is to develop the kinematics

Table 3.1: Comparison of Rotational and Translational Motion

	Variables	Kinematics D.E.s
Translational Motion	(x, y, z)	$\dot{\vec{\mathbf{r}}} = \vec{\mathbf{p}}/m$
Rotational Motion	$(\theta_1, \theta_2, \theta_3)$	?
	$(\mathbf{a}, \Phi)$	?
	$(\mathbf{q}, q_4)$	?

differential equations (D.E.s) to fill in the "?" in Table 3.1. To complete the table, we first need to develop the concept of angular velocity.

3.2.1 Angular Velocity

The easiest way to think about angular velocity is to first consider the simple rotations developed in Section 3.1.4. The first example developed in that section was for a "3-2-1" Euler angle sequence. Thus we are interested in the rotation of one frame, $\mathcal{F}_{i'}$, with respect to another frame, $\mathcal{F}_i$, where the rotation is about the "3" axis (either $\hat{\mathbf{i}}'_3$ or $\hat{\mathbf{i}}_3$). Then, the *angular velocity* of $\mathcal{F}_{i'}$ with respect to $\mathcal{F}_i$ is

$$\vec{\omega}^{i'i} = \dot{\theta}_1 \hat{\mathbf{i}}'_3 = \dot{\theta}_1 \hat{\mathbf{i}}_3 \tag{3.59}$$

Note the ordering of the superscripts in this expression. Also, note that this vector quantity has the same components in either frame; that is,

$$\omega_{i'}^{i'i} = \omega_i^{i'i} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \tag{3.60}$$

This simple expression results because it is a simple rotation. For the "2" rotation from $\mathcal{F}_{i'}$ to $\mathcal{F}_{i''}$, the angular velocity vector is

$$\vec{\omega}^{i''i'} = \dot{\theta}_2 \hat{\mathbf{i}}''_2 = \dot{\theta}_2 \hat{\mathbf{i}}'_2 \tag{3.61}$$

which has components

$$\omega_{i''}^{i''i'} = \omega_{i'}^{i''i'} = \begin{bmatrix} 0 \\ \dot{\theta}_2 \\ 0 \end{bmatrix} \tag{3.62}$$

Finally, for the "1" rotation from $\mathcal{F}_{i''}$ to $\mathcal{F}_b$, the angular velocity vector is

$$\vec{\omega}^{bi''} = \dot{\theta}_3 \hat{\mathbf{b}}_1 = \dot{\theta}_3 \hat{\mathbf{i}}_1'' \tag{3.63}$$

with components

$$\omega_b^{bi''} = \omega_{i''}^{bi''} = \begin{bmatrix} \dot{\theta}_3 \\ 0 \\ 0 \end{bmatrix} \tag{3.64}$$

Thus, the angular velocities for simple rotations are also simple angular velocities.

Now, angular velocity vectors add in the following way: the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$ is equal to the sum of the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_{i''}$, the angular velocity of $\mathcal{F}_{i''}$ with respect to $\mathcal{F}_{i'}$, and the angular velocity of $\mathcal{F}_{i'}$ with respect to $\mathcal{F}_i$. Mathematically,

$$\vec{\omega}^{bi} = \vec{\omega}^{bi''} + \vec{\omega}^{i''i'} + \vec{\omega}^{i'i} \tag{3.65}$$

However, this expression involves vectors, which are mathematically abstract objects. In order to do computations involving angular velocities, we must choose a reference frame, and express all these vectors in that reference frame and add them together. Notice that in Eqs. (3.60,3.62, and 3.64), the components of these vectors are given in different reference frames. To add them, we must transform them all to the same frame. In most attitude dynamics applications, we use the body frame, so for this example, we develop the expression for $\vec{\omega}^{bi}$ in $\mathcal{F}_b$, denoting it ω_b^{bi}.

The first vector on the right hand side of Eq. (3.65) is already expressed in $\mathcal{F}_b$ in Eq. (3.64), so no further transformation is required. The second vector in Eq. (3.65) is given in $\mathcal{F}_{i''}$ and $\mathcal{F}_{i'}$ in Eq. (3.62). Thus, in order to transform $\omega_{i''}^{i''i'}$ (or $\omega_{i'}^{i''i'}$) into $\mathcal{F}_b$, we need to premultiply the column matrix by either $\mathbf{R}^{bi''}$ or $\mathbf{R}^{bi'}$. Both matrices give the exact same result, which is again due to the fact that we are working with simple rotations. Since $\mathbf{R}^{bi''}$ is simpler [$\mathbf{R}^{bi''} = \mathbf{R}_1(\theta_3)$, whereas $\mathbf{R}^{bi'} = \mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)$], we use $\omega_b^{i''i'} = \mathbf{R}^{bi''}\omega_{i''}^{i''i'}$, or

$$\omega_b^{i''i'} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta}_2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \cos\theta_3\ \dot{\theta}_2 \\ -\sin\theta_3\ \dot{\theta}_2 \end{bmatrix} \tag{3.66}$$

Similarly, $\omega_{i'}^{i'i}$ must be premultiplied by $\mathbf{R}^{bi'} = \mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)$ hence

$$\omega_b^{i'i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} = \begin{bmatrix} -\sin\theta_2\ \dot{\theta}_1 \\ \cos\theta_2\ \sin\theta_3\ \dot{\theta}_1 \\ \cos\theta_2\ \cos\theta_3\ \dot{\theta}_1 \end{bmatrix} \tag{3.67}$$

Now we have all the angular velocity vectors of Eq. (3.65) expressed in $\mathcal{F}_b$ and can add them together:

$$
\begin{aligned}
\omega_b^{bi} &= \omega_b^{bi''} + \omega_b^{i''i'} + \omega_b^{i'i} && (3.68)\\
&= \begin{bmatrix} \dot\theta_3 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \cos\theta_3\ \dot\theta_2 \\ -\sin\theta_3\ \dot\theta_2 \end{bmatrix} + \begin{bmatrix} -\sin\theta_2\ \dot\theta_1 \\ \cos\theta_2\ \sin\theta_3\ \dot\theta_1 \\ \cos\theta_2\ \cos\theta_3\ \dot\theta_1 \end{bmatrix} && (3.69)\\
&= \begin{bmatrix} \dot\theta_3 - \sin\theta_2\ \dot\theta_1 \\ \cos\theta_3\ \dot\theta_2 + \cos\theta_2\ \sin\theta_3\ \dot\theta_1 \\ -\sin\theta_3\ \dot\theta_2 + \cos\theta_2\ \cos\theta_3\ \dot\theta_1 \end{bmatrix} && (3.70)\\
&= \begin{bmatrix} -\sin\theta_2 & 0 & 1 \\ \cos\theta_2\ \sin\theta_3 & \cos\theta_3 & 0 \\ \cos\theta_2\ \cos\theta_3 & -\sin\theta_3 & 0 \end{bmatrix} \begin{bmatrix} \dot\theta_1 \\ \dot\theta_2 \\ \dot\theta_3 \end{bmatrix} && (3.71)
\end{aligned}
$$

The last version of this equation is customarily abbreviated as

$$\omega_b^{bi} = \mathbf{S}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{3.72}$$

where $\boldsymbol{\theta} = [\theta_1\ \theta_2\ \theta_3]^T$, and $\mathbf{S}(\boldsymbol{\theta})$ obviously depends on which Euler angle sequence is used. For a given Euler angle sequence, it is relatively straightforward to develop the appropriate $\mathbf{S}(\boldsymbol{\theta})$. Often it is clear what angular velocity vector and reference frame we are working with, and we drop the sub and superscripts on ω.

Thus if the $\dot\theta$'s are known, then they can be integrated to determine the θ's, and then the components of $\vec{\omega}$ can be determined. This integration is entirely analogous to knowing $\dot x$, $\dot y$, and $\dot z$, and integrating these to determine the position x, y, and z. In this translational case however, one usually knows $\dot x$, *etc.*, from determining the *velocity*; *i.e.*, $\mathbf{v} = [\dot x\ \dot y\ \dot z]^T$. The velocity is determined from the kinetics equations of motion as in Eq. (3.3). Similarly, for rotational motion, the kinetics equations of motion are used to determine the angular velocity, which is in turn used to determine the $\dot\theta$'s, not *vice versa*. In the next section, we develop relationships between the time derivatives of the attitude variables and the angular velocity.

3.2.2 Kinematics Equations

We begin by solving Eq. (3.72) for $\dot{\boldsymbol{\theta}}$, which requires inversion of $\mathbf{S}(\boldsymbol{\theta})$. It is straightforward to obtain:

$$\dot{\boldsymbol{\theta}} = \begin{bmatrix} 0 & \sin\theta_3/\cos\theta_2 & \cos\theta_3/\cos\theta_2 \\ 0 & \cos\theta_3 & -\sin\theta_3 \\ 1 & \sin\theta_3\sin\theta_2/\cos\theta_2 & \cos\theta_3\sin\theta_2/\cos\theta_2 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \mathbf{S}^{-1}\omega \tag{3.73}$$

Thus, if we know the ω's as functions of time, and have initial conditions for the three Euler angles, then we can integrate these three differential equations to obtain the

θ's as functions of time. Careful examination of $\mathbf{S}^{-1}$ shows that some of the elements of this matrix become large when θ_2 approaches $\pi/2$, and indeed become infinite when $\theta_2 = \pi/2$. This problem is usually called a *kinematic singularity*, and is one of the difficulties associated with using Euler angles as attitude variables. Even though the angular velocity may be small, the Euler angle rates can become quite large. For a different Euler angle sequence, the kinematic singularity occurs at a different point. For example, with symmetric Euler angle sequences, the kinematic singularity always occurs when the middle angle (θ_2) is 0 or π. Because the singularity occurs at a different point for different sequences, one way to deal with the singularity is to switch Euler angle sequences whenever a singularity is approached. Hughes[1] provides a table of $\mathbf{S}^{-1}$ for all 12 Euler angle sequences. Another difficulty is that it is computationally expensive to compute the sines and cosines necessary to integrate Eq. (3.73).

As we indicated in Section 3.1.5, other attitude variables may be used to represent the orientation of two reference frames, with the Euler axis/angle set, and quaternions, being the most common. We now provide the differential equations relating $(\mathbf{a}, \Phi)$ and $\bar{\mathbf{q}}$ to ω. For the Euler axis/angle set of attitude variables, the differential equations are

$$\dot{\Phi} = \mathbf{a}^T\omega \tag{3.74}$$

$$\dot{\mathbf{a}} = \frac{1}{2}\left[\mathbf{a}^\times - \cot\frac{\Phi}{2}\mathbf{a}^\times\mathbf{a}^\times\right]\omega \tag{3.75}$$

The kinematic singularity in these equations is evidently at $\Phi = 0$ or 2π, both of which correspond to $\mathbf{R} = \mathbf{1}$ which means the two reference frames are identical. Thus it is reasonably straightforward to deal with this singularity. There is, however, the one trig function that must be computed as Φ varies.

For Euler parameters or quaternions, the kinematic equations of motion are

$$\dot{\bar{\mathbf{q}}} = \frac{1}{2}\begin{bmatrix} \mathbf{q}^\times + q_4\mathbf{1} \\ -\mathbf{q}^T \end{bmatrix}\omega = \mathbf{Q}(\bar{\mathbf{q}})\omega \tag{3.76}$$

There are no kinematic singularities associated with $\dot{\bar{\mathbf{q}}}$, and there are no trig functions to evaluate. For these reasons, the quaternion, $\bar{\mathbf{q}}$, is the attitude variable of choice for most satellite attitude dynamics applications.

3.3 Summary

This chapter provides the basic background for describing reference frames, their orientations with respect to each other, and the transformation of vectors from one frame to another.

3.4 Summary of Notation

There are several subscripts and superscripts used in this and preceding chapters. This table summarizes the meanings of these symbols.

Symbol	Meaning
$\vec{\mathbf{v}}$	vector, an abstract mathematical object with direction and length
$\left\{\hat{\mathbf{i}}_1, \hat{\mathbf{i}}_2, \hat{\mathbf{i}}_3\right\}$	the three unit base vectors of a reference frame
$\mathcal{F}_i$	the reference frame with base vectors $\left\{\hat{\mathbf{i}}_1, \hat{\mathbf{i}}_2, \hat{\mathbf{i}}_3\right\}$ typically $\mathcal{F}_i$ denotes an inertial reference frame whereas $\mathcal{F}_b$ denotes a body-fixed frame, and $\mathcal{F}_o$ denotes an orbital reference frame
$\left\{\hat{\mathbf{i}}\right\}$	a column matrix whose 3 elements are the unit vectors of $\mathcal{F}_i$
$\mathbf{v}_i$	a column matrix whose 3 elements are the components of the vector $\vec{\mathbf{v}}$ expressed in $\mathcal{F}_i$
$\mathbf{v}_b$	a column matrix whose 3 elements are the components of the vector $\vec{\mathbf{v}}$ expressed in $\mathcal{F}_b$
$\mathbf{R}^{bi}$	rotation matrix that transforms vectors from $\mathcal{F}_i$ to $\mathcal{F}_b$
$\boldsymbol{\theta}$	a column matrix whose 3 elements are the Euler angles θ_1, θ_2, θ_3
$\vec{\omega}$	an angular velocity vector
$\vec{\omega}^{bi}$	the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$
ω_b^{bi}	the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$ expressed in $\mathcal{F}_b$ typically used in Euler's equations
ω_i^{bi}	the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$ expressed in $\mathcal{F}_i$ not commonly used
ω_a^{bi}	the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$ expressed in $\mathcal{F}_a$ there are applications for this form

3.5 References and further reading

Most satellite attitude dynamics and control textbooks cover kinematics only as a part of the dynamics presentation. Pisacane and Moore[2] is a notable exception, providing a detailed treatment of kinematics before covering dynamics and control. Shuster[3] provided an excellent survey of the many attitude representation approaches, including many interesting historical comments. Wertz's handbook[4] also covers some of this material. Kuipers[5] provides extensive details about quaternions and their applications, but does not include much on spacecraft attitude applications.

Bibliography

[1] Peter C. Hughes. *Spacecraft Attitude Dynamics.* John Wiley & Sons, New York, 1986.

[2] Vincent L. Pisacane and Robert C. Moore, editors. *Fundamentals of Space Systems.* Oxford University Press, Oxford, 1994.

[3] Malcolm D. Shuster. A survey of attitude representations. *Journal of the Astronautical Sciences*, 41(4):439–517, 1993.

[4] J. R. Wertz, editor. *Spacecraft Attitude Determination and Control.* D. Reidel, Dordrecht, Holland, 1978.

[5] Jack B. Kuipers. *Quaternions and Rotation Sequences.* Princeton University Press, Princeton, 1999.

3.6 Exercises

1. Verify the validity of Eq. (3.8) by direct calculation.

2. Develop the rotation matrix for a "3-1-2" rotation from $\mathcal{F}_a$ to $\mathcal{F}_b$. Your result should be a single matrix in terms of θ_1, θ_2, and θ_3 [similar to Eq. (3.42)].

3. Using the relationship between the elements of the quaternion and the Euler angle and axis, verify that Eq. (3.53) and Eq. (3.47) are equivalent.

4. Select two unit vectors and convince yourself that the statement "Rotations do not add like vectors" described on p. 3-12 is true.

5. Develop $\mathbf{S}(\boldsymbol{\theta})$ for a "2-3-1" rotation from $\mathcal{F}_i$ to $\mathcal{F}_b$ so that $\omega_b^{bi} = \mathbf{S}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$. Where is $\mathbf{S}(\boldsymbol{\theta})$ singular?

6. A satellite at altitude 700 km is pointing at a target that is 7° away from nadir (*i.e.*, $\eta = 7°$). What is the range D to the target, and what is the spacecraft elevation angle ε.

7. A satellite in a circular orbit with radius 7000 km is intended to target any point within its instantaneous access area (IAA) for which the elevation angle ε is greater than 5°. What must the range of operation be for the attitude control system?

8. The Tropical Rainfall Measuring Mission is in a circular orbit with altitude 350 km, and inclination $i = 35°$. If it must be able to target any point that is within its IAA and within the tropics, then what must the ACS range of

operations be, and at what minimum elevation angle must the sensor be able to operate?

9. A satellite is in an elliptical orbit with $a = 7000$, $e = 0.1$. Make plots of the IAA as a function of time for one period, using ε_{min} of 0°, 5°, and 10°.

3.7 Problems

Consider the following rotation matrix $\mathbf{R}^{ab}$ that transforms vectors from $\mathcal{F}_b$ to $\mathcal{F}_a$:

$$\mathbf{R}^{ab} = \begin{bmatrix} 0.45457972 & 0.43387382 & -0.77788868 \\ -0.34766601 & 0.89049359 & 0.29351236 \\ 0.82005221 & 0.13702069 & 0.55564350 \end{bmatrix}$$

1. Use at least two different properties of rotation matrices to convince yourself that $\mathbf{R}^{ab}$ is indeed a rotation matrix. Remark on any discrepancies you notice.

2. Determine the Euler axis $\mathbf{a}$ and Euler principal angle Φ, directly from $\mathbf{R}^{ab}$. Verify your results using the formula for $\mathbf{R}(\mathbf{a}, \Phi)$. Verify that $\mathbf{Ra} = \mathbf{a}$.

3. Determine the components of the quaternion $\bar{\mathbf{q}}$, directly from $\mathbf{R}^{ab}$. Verify your results using the formula for $\mathbf{R}(\bar{\mathbf{q}})$. Verify your results using the relationship between $\bar{\mathbf{q}}$ and $(\mathbf{a}, \Phi)$.

4. Derive the formula for a 2-3-1 rotation. Determine the Euler angles for a 2-3-1 rotation, directly from $\mathbf{R}^{ab}$. Verify your results using the formula you derived.

5. Write a short paragraph discussing the relative merits of the three different representations of $\mathbf{R}$.

6. This problem requires numerical integration of the kinematics equations of motion. You should have a look at the MatLab Appendix.

   Suppose that $\mathcal{F}_b$ and $\mathcal{F}_a$ are initially aligned, so that $\mathbf{R}^{ba}(0) = \mathbf{1}$. At $t = 0$, $\mathcal{F}_b$ begins to rotate with angular velocity $\omega_b = e^{-4t}[\sin t \;\; \sin 2t \;\; \sin 3t]^T$ with respect to $\mathcal{F}_a$.

   Use a 1-2-3 Euler angle sequence and make a plot of the three Euler angles *vs.* time for $t = 0$ to 10 s. Use a sufficiently small step size so that the resulting plot is "smooth."

   Use the quaternion representation and make a plot of the four components of the quaternion *vs.* time time for $t = 0$ to 10 s. Use a sufficiently small step size so that the resulting plot is "smooth."

   Use the results of the two integrations to determine the rotation matrix at $t = 10$ s. Do this using the expression for $\mathbf{R}(\boldsymbol{\theta})$ and for $\mathbf{R}(\bar{\mathbf{q}})$. Compare the results.

Chapter 4

Attitude Determination

Essentially all control systems require two types of hardware components: sensors and actuators. Sensors are used to sense or measure the state of the system, and actuators are used to adjust the state of the system. For example, a typical thermostat used to control room temperature has a thermocouple (sensor) and a connection to the furnace (actuator). The control system compares the reference temperature to the measured temperature and either turns the furnace on or off, depending on the sign of the difference between the two. A spacecraft attitude determination and control system typically uses a variety of sensors and actuators. Because attitude is described by three or more attitude variables, the difference between the desired and measured states is slightly more complicated than for a thermostat, or even for the position of the satellite in space. Furthermore, the mathematical analysis of attitude determination is complicated by the fact that attitude determination is necessarily either underdetermined or overdetermined.

In this chapter, we develop the basic concepts and tools for attitude determination, beginning with attitude sensors and then introducing attitude determination algorithms. We focus here on static attitude determination, where time is not involved in the computations. The more complicated problem of dynamic attitude determination is treated in Chapter 7.

4.1 The Basic Idea

Attitude determination uses a combination of sensors and mathematical models to collect vector components in the body and inertial reference frames. These components are used in one of several different algorithms to determine the attitude, typically in the form of a quaternion, Euler angles, or a rotation matrix. It takes at least two vectors to estimate the attitude. For example, an attitude determination system might use a sun vector, $\hat{\mathbf{s}}$ and a magnetic field vector $\hat{\mathbf{m}}$. A sun sensor measures the components of $\hat{\mathbf{s}}$ in the body frame, $\mathbf{s}_b$, while a mathematical model of the Sun's apparent motion relative to the spacecraft is used to determine the components in the inertial frame, $\mathbf{s}_i$. Similarly, a magnetometer measures the components of $\hat{\mathbf{m}}$ in the body frame, $\mathbf{m}_b$, while a mathematical model of the Earth's magnetic field relative to the spacecraft is used to determine the components in the inertial frame, $\mathbf{m}_i$. An attitude determination algorithm is then used to find a rotation matrix $\mathbf{R}^{bi}$ such that

$$\mathbf{s}_b = \mathbf{R}^{bi}\mathbf{s}_i \quad \text{and} \quad \mathbf{m}_b = \mathbf{R}^{bi}\mathbf{m}_i \tag{4.1}$$

The attitude determination analyst needs to understand how various sensors measure the body-frame components, how mathematical models are used to determine the inertial-frame components, and how standard attitude determination algorithms are used to estimate $\mathbf{R}^{bi}$.

4.2 Underdetermined or Overdetermined?

In the previous section we make claim that at least two vectors are required to determine the attitude. Recall that it takes three independent parameters to determine the attitude, and that a unit vector is actually only two parameters because of the unit vector constraint. Therefore we require three scalars to determine the attitude. Thus the requirement is for more than one and less than two vector measurements. The attitude determination is thus unique in that one measurement is not enough, *i.e.*, the problem is underdetermined, and two measurements is too many, *i.e.*, the problem is overdetermined. The primary implication of this observation is that all attitude determination algorithms are really attitude estimation algorithms.

4.3 Attitude Measurements

There are two basic classes of attitude sensors. The first class makes *absolute* measurements, whereas the second class makes *relative* measurements. Absolute measurement sensors are based on the fact that knowing the position of a spacecraft in its orbit makes it possible to compute the vector directions, with respect to an inertial frame, of certain astronomical objects, and of the force lines of the Earth's magnetic field. Absolute measurement sensors measure these directions with respect to a spacecraft- or body-fixed reference frame, and by comparing the measurements with the known reference directions in an inertial reference frame, are able to determine (at least approximately) the relative orientation of the body frame with respect to the inertial frame. Absolute measurements are used in the static attitude determination algorithms developed in this chapter.

Relative measurement sensors belong to the class of gyroscopic instruments, including the *rate gyro* and the *integrating gyro*. Classically, these instruments have been implemented as spinning disks mounted on gimbals; however, modern technology has brought such marvels as *ring laser gyros*, *fiber optic gyros*, and *hemispherical* *resonator gyros*. Relative measurement sensors are used in the dynamic attitude determination algorithms developed in Chapter 7.

4.3.1 Sun Sensors

We begin with sun sensors because of their relative simplicity and the fact that virtually all spacecraft use sun sensors of some type. The sun is a useful reference direction because of its brightness relative to other astronomical objects, and its relatively small apparent radius as viewed by a spacecraft near the Earth. Also, most satellites use solar power, and so need to make sure that solar panels are oriented correctly with respect to the sun. Some satellites have sensitive instruments that must not be exposed to direct sunlight. For all these reasons, sun sensors are important components in spacecraft attitude determination and control systems.

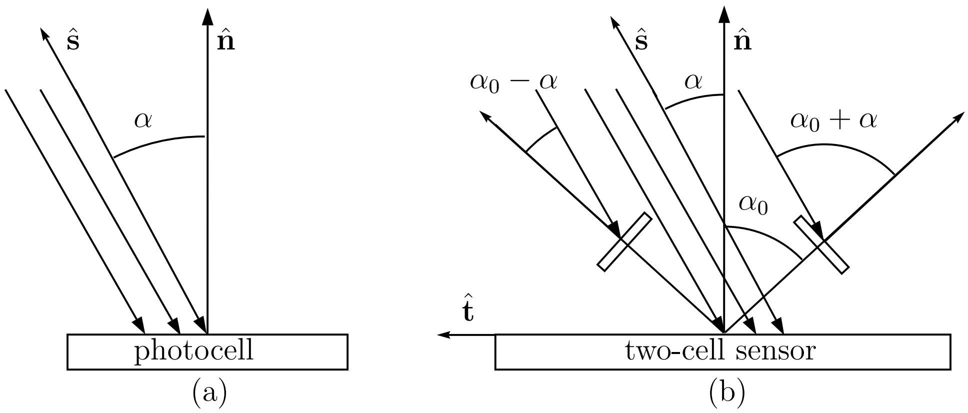

Figure 4.1: Photocells for Sun Sensors. (a) Single photocell. (b) Pair of photocells for measurement of α in $\hat{\mathbf{n}} - \hat{\mathbf{t}}$ plane.

The object of a sun sensor is to provide an approximate unit vector, with respect to the body reference frame, that points towards the sun. We denote this vector by $\hat{\mathbf{s}}$, which can be written as

$$\hat{\mathbf{s}} = \mathbf{s}_i{}^{\mathsf{T}}\{\hat{\mathbf{i}}\} = \mathbf{s}_b{}^{\mathsf{T}}\{\hat{\mathbf{b}}\} \tag{4.2}$$

If the position of the spacecraft in its orbit is known, along with the position of the Earth in its orbit, then $\mathbf{s}_i$ is known. An algorithm to compute $\mathbf{s}_i$ is given at the end of this section.

Two types of sun sensors are available: *analog* and *digital*. Analog sun sensors are based on photocells whose current output is proportional to the cosine of the angle α between the direction to the sun and the normal to the photocell (Fig. 4.1a). That is, the current output is given by

$$I(\alpha) = I(0)\cos\alpha \tag{4.3}$$

from which α can be determined. Denoting the unit normal of the photocell by $\hat{\mathbf{n}}$, we see that

$$\hat{\mathbf{s}} \cdot \hat{\mathbf{n}} = \cos\alpha \tag{4.4}$$

However, knowing α does not provide enough information to determine $\hat{\mathbf{s}}$ completely, since the component of $\hat{\mathbf{s}}$ perpendicular to $\hat{\mathbf{n}}$ remains unknown. Typically, sun sensors combine four such photocells to provide the complete unit vector measurement. Details on specific sun sensors are included in Refs. 1, 2, and 3.

To determine the angle in a specific plane, one normally uses two photocells tilted at an angle α_0 with respect to the normal $\hat{\mathbf{n}}$ of the sun sensor (see the second diagram in Fig. 4.1). This arrangement gives the angle between the sun sensor normal, $\hat{\mathbf{n}}$ and the projection of the sun vector $\hat{\mathbf{s}}$ onto the $\hat{\mathbf{n}} - \hat{\mathbf{t}}$ plane. Then the two photocells generate currents

$$\begin{aligned} I_1(\alpha) &= I(0)\cos(\alpha_0 - \alpha) && (4.5) \\ I_2(\alpha) &= I(0)\cos(\alpha_0 + \alpha) && (4.6) \end{aligned}$$

Taking the difference of these two expressions, we obtain

$$\begin{aligned} \Delta I &= I_2 - I_1 && (4.7) \\ &= I(0)\left[\cos(\alpha_0 + \alpha) - \cos(\alpha_0 - \alpha)\right] && (4.8) \\ &= 2I(0)\sin\alpha_0 \sin\alpha && (4.9) \\ &= C\sin\alpha && (4.10) \end{aligned}$$

where $C = 2I(0)\sin\alpha_0$ is a constant that depends on the electrical characteristics of the photocells and the geometrical arrangement of the two photocells.

Using two appropriately arranged pairs of photocells, we obtain the geometry shown in Fig. 4.2. In this picture, $\hat{\mathbf{n}}_1$ is the normal vector for the first pair of photocells, and $\hat{\mathbf{n}}_2$ is the normal vector for the second pair. The $\hat{\mathbf{t}}$ vector is chosen to define the two planes of the photocell pairs; *i.e.*, $\hat{\mathbf{n}}_1$ and $\hat{\mathbf{t}}$ are as shown in Fig. 4.1(b) for one pair, and $\hat{\mathbf{n}}_2$ and $\hat{\mathbf{t}}$ are for the second pair. Thus $\left\{\hat{\mathbf{n}}_1, \hat{\mathbf{n}}_2, \hat{\mathbf{t}}\right\}$ comprise the three unit vectors of a frame denoted by $\mathcal{F}_s$ (s for sun sensor). The spacecraft designer determines the orientation of this frame with respect to the body frame; thus the orientation matrix $\mathbf{R}^{bs}$ is known. The measurements provide the components of the sun vector in the sun sensor frame, $\mathbf{s}_s$, and the matrix $\mathbf{R}^{bs}$ provides the components in the body frame, $\mathbf{s}_b = \mathbf{R}^{bs}\mathbf{s}_s$.

The two measured angles α_1 and α_2 determine $\mathbf{s}_s$ as follows. We want components of a unit vector, but it is easiest to begin by letting the component in the $\hat{\mathbf{n}}_1$ direction be equal to one. Then the geometry of the arrangement implies that the components in the $\hat{\mathbf{n}}_2$ and $\hat{\mathbf{t}}$ directions are $\tan\alpha_1/\tan\alpha_2$, and $\tan\alpha_1$, respectively. Denoting the components of this non-unit vector by

$$\mathbf{s}_s^* = \left[1 \quad \tan\alpha_1/\tan\alpha_2 \quad \tan\alpha_1\right]^{\mathsf{T}} \tag{4.11}$$

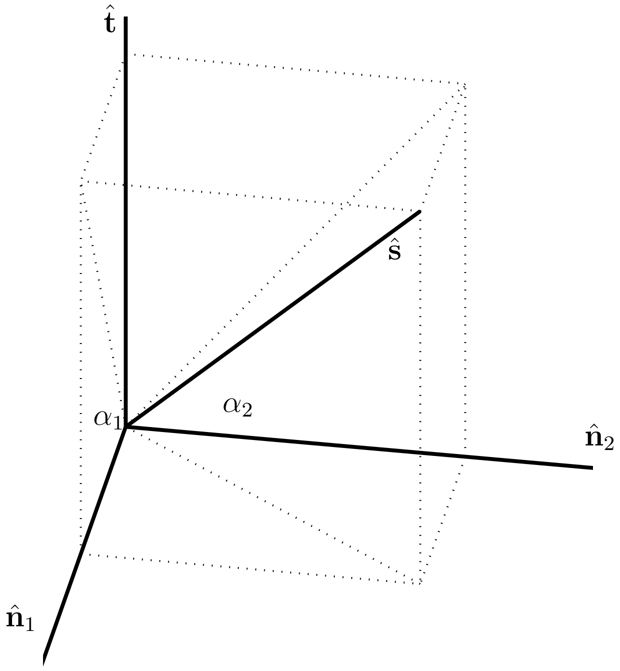

Figure 4.2: Geometry for a four-photocell sun sensor

we obtain the sun vector components by normalizing this vector:

$$\mathbf{s}_s = \frac{[1 \quad \tan\alpha_1/\tan\alpha_2 \quad \tan\alpha_1]^\mathsf{T}}{\sqrt{\mathbf{s}_s^{*\mathsf{T}}\mathbf{s}_s^*}} \tag{4.12}$$

Thus, using a four-photocell sun sensor leads directly to the calculation of a unit sun vector expressed in the sun sensor frame, $\mathcal{F}_s$. As noted above, $\mathbf{R}^{bs}$ is a known constant rotation matrix, and therefore the matrix $\mathbf{s}_b$ is also known.

Example 4.1 *Suppose a sun sensor configured as in Fig. 4.2 gives* $\alpha_1 = 0.9501$ *and* $\alpha_2 = 0.2311$*, both in radians. Then* $\mathbf{s}_s^* = [1 \;\; 5.9441 \;\; 1.3987]^\mathsf{T}$*, which has magnitude* $s^* = 6.1878$*. Thus* $\mathbf{s}_s = [0.1616 \;\; 0.9606 \;\; 0.2260]^\mathsf{T}$*. If the orientation of the sun sensor frame with respect to the body frame is described by the quaternion* $\bar{\mathbf{q}} = [0.1041 \;\; -0.2374 \;\; -0.5480 \;\; 0.7953]^\mathsf{T}$*, then we can compute* $\mathbf{R}^{bs}$ *using Eq. (3.53), and use the result to obtain* $\mathbf{s}_b = [-0.7789 \;\; 0.5920 \;\; 0.2071]^\mathsf{T}$*.*

A mathematical model for the direction to the sun in the inertial frame is required. That is, we need to know $\mathbf{s}_i$ as well as $\mathbf{s}_b$. A convenient algorithm is developed in Ref. 4, and is presented here. The algorithm requires the current time, expressed as the Julian date, and returns the position vector of the Sun in the Earth-centered inertial reference frame.

We first present two algorithms for computing the Julian date. The first algorithm determines the year, month, day, hour, minute and second from a two-line element set epoch. The TLE epoch is in columns 19–32 of line 1 (see Appendix A), and is of the form `yyddd.ffffffff`, where `yy` is the last two digits of the year, `ddd` is the day of the year (with January 1 being 001), and `.ffffffff` is the fraction of that day. Thus, `01001.50000000` is noon universal time on January 1, 2001.

Algorithm 4.1 *Given the epoch from a two-line element set, compute the year, month, day, hour, minute and second.*

Extract `yy`, `ddd`, *and* `.ffffffff`.

The year is either 19`yy` *or 20*`yy`*, and for the next six decades the appropriate choice should be evident. Since old two-line element sets should normally not be used for current operations, the 19*`yy` *form is of historical interest only.*

The day, `ddd`*, begins with* `001` *as January 1, 20*`yy`.

Algorithm 4.2 *Given year, month, day, hour, minute, second, compute the Julian date, JD.*

$$JD = 367\,year - INT\left\{\frac{7\left[year + INT\left(\frac{month+9}{12}\right)\right]}{4}\right\} + INT\left(\frac{275\,month}{9}\right) + day$$
$$+1,721,013.5 + \frac{hour}{24} + \frac{minute}{1440} + \frac{s}{86,400}$$

Algorithm 4.3 *Given the Julian Date, JD, compute the distance to the sun, s^*, and the unit vector direction to the sun in the Earth-centered inertial frame, $\mathbf{s}_i$.*

$$\begin{aligned}
T_{UT1} &= \frac{JD_{UT1} - 2,451,545.0}{36,525} \\
\lambda_{M_{Sun}} &= 280.460\,618\,4^\circ + 36,000.770\,053\,61 T_{UT1} \\
& \quad \textit{mean longitude of Sun} \\
\text{Let } T_{TDB} &\approx T_{UT1} \\
M_{Sun} &= 357.527\,723\,3^\circ + 35,999.050\,34 T_{TDB} \\
& \quad \textit{mean anomaly of Sun} \\
\lambda_{ecliptic} &= \lambda_{M_{Sun}} + 1.914\,666\,471^\circ \sin M_{Sun} + 0.918\,994\,643 \sin 2M_{Sun} \\
& \quad \textit{ecliptic longitude of Sun}
\end{aligned}$$

$$
\begin{aligned}
s^* &= 1.000\,140\,612 - 0.016\,708\,617 \cos M_{Sun} - 0.000\,139\,589 \cos 2M_{Sun} \\
&\quad \textit{distance to Sun in AUs} \\
\varepsilon &= 23.439\,291^\circ - 0.013\,004\,2 T_{TDB} \\
\mathbf{s}_i &= \begin{bmatrix} \cos \lambda_{ecliptic} \\ \cos \varepsilon \sin \lambda_{ecliptic} \\ \sin \varepsilon \sin \lambda_{ecliptic} \end{bmatrix}
\end{aligned}
$$

4.3.2 Magnetometers

A vector magnetometer returns a vector measurement of the Earth's magnetic field in a magnetometer-fixed reference frame. As with sun sensors, the orientation of the magnetometer frame with respect to the spacecraft body frame is determined by the designers. Therefore, we can assume that the magnetometer provides a measurement of the magnetic field in the body frame, $\mathbf{m}_b^*$, which we can normalize to obtain $\mathbf{m}_b$. We also need a mathematical model of the Earth's magnetic field so that we can determine $\mathbf{m}_i$ based on the time and the spacecraft's position. More details can be found in Refs. 1 and 5.

Using a simple tilted dipole model of the Earth's magnetic field, we can write the components of the magnetic field in the ECI frame as

$$
\mathbf{m}_i^* = \frac{R_\oplus^3 H_0}{r^3} \left[3 \mathbf{d}_i{}^\mathsf{T} \hat{\mathbf{r}}_i \hat{\mathbf{r}}_i - \mathbf{d}_i \right] \tag{4.13}
$$

$$
\mathbf{m}_i^* = \frac{R_\oplus^3 H_0}{r^3} \begin{bmatrix} 3(\mathbf{d}^\mathsf{T}\hat{\mathbf{r}})\hat{r}_1 - \sin\theta_m' \cos\alpha_m \\ 3(\mathbf{d}^\mathsf{T}\hat{\mathbf{r}})\hat{r}_2 - \sin\theta_m' \sin\alpha_m \\ 3(\mathbf{d}^\mathsf{T}\hat{\mathbf{r}})\hat{r}_3 - \cos\theta_m' \end{bmatrix} \tag{4.14}
$$

where the vector $\hat{\mathbf{d}}$ is the unit dipole direction, with components in the inertial frame:

$$
\mathbf{d}_i = \begin{bmatrix} \sin\theta_m' \cos\alpha_m \\ \sin\theta_m' \sin\alpha_m \\ \cos\theta_m' \end{bmatrix} \tag{4.15}
$$

The vector $\hat{\mathbf{r}}$ is a unit vector in the direction of the position vector of the spacecraft.* The constants $R_\oplus = 6378$ km and $H_0 = 30,115$ nT are the radius of the Earth and a constant characterizing the Earth's magnetic field, respectively. In these expressions,

$$
\alpha_m = \theta_{g0} + \omega_\oplus t + \phi_m' \tag{4.16}
$$

where θ_{g0} is the Greenwich sidereal time at epoch, $\omega_\oplus$ is the average rotation rate of the Earth, t is the time since epoch, and θ_m' and ϕ_m' are the coelevation and East longitude of the dipole. Current values of these angles are $\theta_m' = 196.54^\circ$ and $\phi_m' = 108.43^\circ$.

*We normally only use the "hat" to denote unit vectors and omit the symbol when referring to the components of a vector in a specific frame. However, here we use $\hat{\mathbf{r}}$ and $\hat{\mathbf{r}}_i$ since $\mathbf{r}$ and $\mathbf{r}_i$ normally denote the position vector and its components in $\mathcal{F}_i$.

Table 4.1: Sensor Accuracy Ranges

Sensor	Accuracy	Characteristics and Applicability
Magnetometers	1.0° (5000 km alt) 5.0° (200 km alt)	Attitude measured relative to Earth's local magnetic field. Magnetic field uncertainties and variability dominate accuracy. Usable only below $\sim$ 6,000 km.
Earth sensors	0.05° (GEO) 0.1° (LEO)	Horizon uncertainties dominate accuracy. Highly accurate units use scanning.
Sun sensors	0.01°	Typical field of view ±30°
Star sensors	2 arc-sec	Typical field of view ±6°
Gyroscopes	0.001 deg/hr	Normal use involves periodically resetting reference.
Directional antennas	0.01° to 0.5°	Typically 1% of the antenna beamwidth
Adapted from Ref. 6		

4.3.3 Sensor Accuracy

Attitude determination sensors vary widely in expense, complexity, reliability and accuracy. Some of the accuracy characteristics are included in Table 4.1.

4.4 Deterministic Attitude Determination

Determining the attitude of a spacecraft is equivalent to determining the rotation matrix describing the orientation of the spacecraft-fixed reference frame, $\mathcal{F}_b$, with respect to a known reference frame, say an inertial frame, $\mathcal{F}_i$. That is, attitude determination is equivalent to determining $\mathbf{R}^{bi}$. Although there are nine numbers in this direction cosine matrix, as we showed in Chapter 3, it only takes three numbers to determine the matrix completely. Since each measured unit vector provides two pieces of information, it takes at least two different measurements to determine the attitude. In fact, this results in an overdetermined problem, since we have three unknowns and four known quantities.

We begin with two measurement vectors, such as the direction to the sun and the direction of the Earth's magnetic field. We denote the actual vectors by $\hat{\mathbf{s}}$ and $\hat{\mathbf{m}}$, respectively. The measured components of the vectors, with respect to the body frame, are denoted $\mathbf{s}_b$ and $\mathbf{m}_b$, respectively. The known components of the vectors in the inertial frame are $\mathbf{s}_i$ and $\mathbf{m}_i$. Ideally, the rotation matrix, or attitude matrix, $\mathbf{R}^{bi}$, satisfies

$$\mathbf{s}_b = \mathbf{R}^{bi}\mathbf{s}_i \quad \text{and} \quad \mathbf{m}_b = \mathbf{R}^{bi}\mathbf{m}_i \tag{4.17}$$

Unfortunately, since the problem is overdetermined, it is not generally possible to find such an $\mathbf{R}^{bi}$. The simplest deterministic attitude determination algorithm is based on discarding one piece of this information; however, this approach does not simply amount to throwing away one of the components of one of the measured directions. The algorithm is known as the *Triad* algorithm, because it is based on constructing two *triads* of orthonormal unit vectors using the vector information that we have. The two triads are the components of the same reference frame, denoted $\mathcal{F}_t$, expressed in the body and inertial frames.

This reference frame is constructed by assuming that one of the body/inertial vector pairs is correct. For example, we could assume that the sun vector measurement is exact, so that when we find the attitude matrix, the first of Eqs. (4.17) is satisfied exactly. We use this direction as the first base vector of $\mathcal{F}_t$. That is,

$$\hat{\mathbf{t}}_1 = \hat{\mathbf{s}} \tag{4.18}$$

$$\mathbf{t}_{1b} = \mathbf{s}_b \tag{4.19}$$

$$\mathbf{t}_{1i} = \mathbf{s}_i \tag{4.20}$$

We then construct the second base vector of $\mathcal{F}_t$ as a unit vector in the direction perpendicular to the two observations. That is,

$$\hat{\mathbf{t}}_2 = \hat{\mathbf{s}} \times \hat{\mathbf{m}} \tag{4.21}$$

$$\mathbf{t}_{2b} = \frac{\mathbf{s}_b^{\times}\mathbf{m}_b}{|\mathbf{s}_b^{\times}\mathbf{m}_b|} \tag{4.22}$$

$$\mathbf{t}_{2i} = \frac{\mathbf{s}_i^{\times}\mathbf{m}_i}{|\mathbf{s}_i^{\times}\mathbf{m}_i|} \tag{4.23}$$

Note that we are in effect assuming that the measurement of the magnetic field vector is less accurate than the measurement of the sun vector. The third base vector of $\mathcal{F}_t$ is chosen to complete the triad:

$$\hat{\mathbf{t}}_3 = \hat{\mathbf{t}}_1 \times \hat{\mathbf{t}}_2 \tag{4.24}$$

$$\mathbf{t}_{3b} = \mathbf{t}_{1b}^{\times}\mathbf{t}_{2b} \tag{4.25}$$

$$\mathbf{t}_{3i} = \mathbf{t}_{1i}^{\times}\mathbf{t}_{2i} \tag{4.26}$$

Now, we construct two rotation matrices by putting the $\mathbf{t}$ vector components into the columns of two 3×3 matrices. The two matrices are

$$[\mathbf{t}_{1b}\ \ \mathbf{t}_{2b}\ \ \mathbf{t}_{3b}] \quad \text{and} \quad [\mathbf{t}_{1i}\ \ \mathbf{t}_{2i}\ \ \mathbf{t}_{3i}] \tag{4.27}$$

Comparing these matrices with Eq. (3.33), it is evident that they are $\mathbf{R}^{bt}$ and $\mathbf{R}^{it}$, respectively. Now, to obtain the desired attitude matrix, $\mathbf{R}^{bi}$, we simply form

$$\mathbf{R}^{bi} = \mathbf{R}^{bt}\mathbf{R}^{ti} = [\mathbf{t}_{1b}\ \ \mathbf{t}_{2b}\ \ \mathbf{t}_{3b}]\,[\mathbf{t}_{1i}\ \ \mathbf{t}_{2i}\ \ \mathbf{t}_{3i}]^{\mathsf{T}} \tag{4.28}$$

Equation (4.28 completes the Triad algorithm.

Example 4.2 *Suppose a spacecraft has two attitude sensors that provide the following measurements of the two vectors* $\hat{\mathbf{v}}_1$ *and* $\hat{\mathbf{v}}_2$*:*

$$\mathbf{v}_{1b} = [0.8273 \;\; 0.5541 \;\; -0.0920]^{\mathsf{T}} \tag{4.29}$$
$$\mathbf{v}_{2b} = [-0.8285 \;\; 0.5522 \;\; -0.0955]^{\mathsf{T}} \tag{4.30}$$

These vectors have known inertial frame components of

$$\mathbf{v}_{1i} = [-0.1517 \;\; -0.9669 \;\; 0.2050]^{\mathsf{T}} \tag{4.31}$$
$$\mathbf{v}_{2i} = [-0.8393 \;\; 0.4494 \;\; -0.3044]^{\mathsf{T}} \tag{4.32}$$

Applying the Triad algorithm, we construct the components of the vectors $\hat{\mathbf{t}}_j, j = 1, 2, 3$ *in both the body and inertial frames:*

$$\mathbf{t}_{1b} = [0.8273 \;\; 0.5541 \;\; -0.0920]^{\mathsf{T}} \tag{4.33}$$
$$\mathbf{t}_{2b} = [-0.0023 \;\; 0.1671 \;\; 0.9859]^{\mathsf{T}} \tag{4.34}$$
$$\mathbf{t}_{3b} = [0.5617 \;\; -0.8155 \;\; 0.1395]^{\mathsf{T}} \tag{4.35}$$

and

$$\mathbf{t}_{1i} = [-0.1517 \;\; -0.9669 \;\; 0.2050]^{\mathsf{T}} \tag{4.36}$$
$$\mathbf{t}_{2i} = [0.2177 \;\; -0.2350 \;\; -0.9473]^{\mathsf{T}} \tag{4.37}$$
$$\mathbf{t}_{3i} = [0.9641 \;\; -0.0991 \;\; 0.2462]^{\mathsf{T}} \tag{4.38}$$

Using these results with Eq. (4.28), we obtain the approximate rotation matrix

$$\mathbf{R}^{bi} = \begin{bmatrix} 0.4156 & -0.8551 & 0.3100 \\ -0.8339 & -0.4943 & -0.2455 \\ 0.3631 & -0.1566 & -0.9185 \end{bmatrix} \tag{4.39}$$

Applying this rotation matrix to $\mathbf{v}_{1i}$ *gives* $\mathbf{v}_{1b}$ *exactly, because we used this condition in the formulation; however, applying it to* $\mathbf{v}_{2i}$ *does not give* $\mathbf{v}_{2b}$ *exactly. If we know* a priori *that sensor 2 is more accurate than sensor 1, then we can use* $\hat{\mathbf{v}}_2$ *as the exact measurement, hopefully leading to a more accurate estimate of* $\mathbf{R}^{bi}$.

4.5 Statistical Attitude Determination

If more than two observations are available, and we want to use all the information, we can use a statistical method. In fact, since we discarded some information from the two observations in developing the Triad algorithm, the statistical method also provides a (hopefully) better estimate of $\mathbf{R}^{bi}$ in that case as well.

Suppose we have a set of N unit vectors $\hat{\mathbf{v}}_k$, $k = 1, ..., N$. For each vector, we have a sensor measurement in the body frame, $\mathbf{v}_{kb}$, and a mathematical model of the components in the inertial frame, $\mathbf{v}_{ki}$. We want to find a rotation matrix $\mathbf{R}^{bi}$, such that

$$\mathbf{v}_{kb} = \mathbf{R}^{bi}\mathbf{v}_{ki} \tag{4.40}$$

for each of the N vectors. Obviously this set of equations is overdetermined if $N \geq 2$, and therefore the equation cannot, in general, be satisfied for each $k = 1, ..., N$. Thus we want to find a solution for $\mathbf{R}^{bi}$ that in some sense minimizes the overall error for the N vectors.

One way to state the problem is: find a matrix $\mathbf{R}^{bi}$ that minimizes the *loss function*†:

$$J\left(\mathbf{R}^{bi}\right) = \frac{1}{2}\sum_{k=1}^{N} w_k \left|\mathbf{v}_{kb} - \mathbf{R}^{bi}\mathbf{v}_{ki}\right|^2 \tag{4.41}$$

In this expression, J is the loss function to be minimized, k is the counter for the N observations, $\hat{\mathbf{v}}_k$ is the k^{th} vector being measured, $\mathbf{v}_{kb}$ is the matrix of measured components in the body frame, and $\mathbf{v}_{ki}$ is the matrix of components in the inertial frame as determined by appropriate mathematical models. This loss function is a sum of the squared errors for each vector measurement. If the measurements and mathematical models are all perfect, then Eq. (4.40) will be satisfied for all N vectors and $J = 0$. If there are any errors or noisy measurements, then $J > 0$. The smaller we can make J, the better the approximation of $\mathbf{R}^{bi}$.

In this section we present three different methods for solving this minimization problem: an iterative numerical solution based on Newton's method; an exact method known as the "q-method;" and an efficient approximation of the q-method known as QUEST (QUaternion ESTimator).

4.5.1 Numerical solution

We can use a systematic algorithm that converges to a rotation matrix giving a good estimate of the attitude. The algorithm requires an initial matrix $\mathbf{R}_0^{bi}$ and iteratively improves it to minimize J. However, recall that the components of a rotation matrix cannot be changed independently. That is, even though there are nine numbers in the matrix, there are constraints that must be satisfied, and, as we know, only *three* numbers are required to specify the rotation matrix completely (*e.g.*, Euler angles). Thus, there are actually only three variables that we need to determine. We could use quaternions or the Euler axis/angle set for attitude variables, but then we would need to incorporate the constraints $\bar{\mathbf{q}}^{\mathsf{T}}\bar{\mathbf{q}} = 1$, or $\mathbf{a}^{\mathsf{T}}\mathbf{a} = 1$, while trying to minimize J. Thus, even though there are trigonometric functions involved, it is probably advantageous to use the Euler angles in this application. Thus, we may write

$$J(\mathbf{R}) = J(\boldsymbol{\theta}) = J(\mathbf{a}, \Phi) = J(\bar{\mathbf{q}}) \tag{4.42}$$

†This problem was first posed by Wahba,[7] and forms the basis of most attitude determination algorithms.

and restate the problem as: find the attitude that minimizes $J(\cdot)$.

Minimization of a function requires taking its derivative and setting the derivative equal to zero, then solving for the unknown variable(s). To minimize the loss function, we must recognize that the unknown variable is multi-dimensional, so that the derivative of J with respect to the unknowns is an $n_p \times 1$ matrix of partial derivatives. For example, if we use Euler angles as the attitude variables, then the minimization is:

$$\frac{\partial J}{\partial \theta_1} = 0 \tag{4.43}$$

$$\frac{\partial J}{\partial \theta_2} = 0 \tag{4.44}$$

$$\frac{\partial J}{\partial \theta_3} = 0 \tag{4.45}$$

If we use quaternions, the minimization is:

$$\frac{\partial J}{\partial q_1} = 0 \tag{4.46}$$

$$\frac{\partial J}{\partial q_2} = 0 \tag{4.47}$$

$$\frac{\partial J}{\partial q_3} = 0 \tag{4.48}$$

$$\frac{\partial J}{\partial q_4} = 0 \tag{4.49}$$

subject to the constraint that $\bar{\mathbf{q}}^{\mathsf{T}}\bar{\mathbf{q}} = 1$. Incorporating the constraint into the minimization involves the addition of a Lagrange multiplier.

4.5.2 Review of minimization

If we want to find the minimum of a function of one variable, say $\min f(x)$, we would solve $F(x) = f'(x) = 0$. A widely used method of solving such an equation is Newton's method. The method is based on the Taylor series of $F(x)$ about the current estimate, x_n, which we assume is close to the correct answer, denoted x^*, with the difference between x^* and x_n denoted Δx. That is,

$$F(x^*) = F(x_n + \Delta x) = F(x_n) + \frac{\partial F}{\partial x}(x_n)\Delta x + \mathcal{O}(\Delta x^2) \tag{4.50}$$

Since $F(x^*) = 0$, and we have assumed we are close (implying $\Delta x^2 \approx 0$), we can solve this equation for Δx, giving

$$\Delta x = -\left[\frac{\partial F}{\partial x}(x_n)\right]^{-1} F(x_n) \tag{4.51}$$

Thus, a hopefully closer estimate is

$$x_{n+1} = x_n - \left[\frac{\partial F}{\partial x}(x_n)\right]^{-1} F(x_n) \tag{4.52}$$

We can continue applying these *Newton steps* until $\Delta x \to 0$, or until $F \to 0$. Usually the *stopping conditions* used with Newton's method use a combination of both these criteria.

Because J depends on more than one variable, we use the multivariable version of Newton's method. In this case the Newton step is

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \left[\frac{\partial \mathbf{F}}{\partial \mathbf{x}}(\mathbf{x}_n)\right]^{-1} \mathbf{F}(\mathbf{x}_n) \tag{4.53}$$

where the bold-face variables are column matrices, and the -1 superscript indicates a matrix inverse rather than "one over."

4.5.3 Application to minimizing J

Comparing Eqs. (4.52) and (4.53), what are $\mathbf{F}$ and $\partial\mathbf{F}/\partial\mathbf{x}$ for this problem? In the single-variable case, $F(x)$ is simply $f'(x)$, and $\partial F/\partial x$ is simply $f''(x)$. In the multivariable case, $\mathbf{F}(\mathbf{x})$ is given by

$$\mathbf{F} = \frac{\partial J}{\partial \mathbf{x}} = \left[\frac{\partial J}{\partial x_1} \quad \frac{\partial J}{\partial x_2} \quad \frac{\partial J}{\partial x_3}\right]^{\mathsf{T}} \tag{4.54}$$

where $x_j = \theta_j$ is the j^{th} of the three Euler angles. The expression $\partial\mathbf{F}/\partial\mathbf{x}$ represents a 3×3 *Jacobian* matrix whose elements are

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial^2 J}{\partial x_1^2} & \frac{\partial^2 J}{\partial x_1 \partial x_2} & \frac{\partial^2 J}{\partial x_1 \partial x_3} \\ \frac{\partial^2 J}{\partial x_2 \partial x_1} & \frac{\partial^2 J}{\partial x_2^2} & \frac{\partial^2 J}{\partial x_2 \partial x_3} \\ \frac{\partial^2 J}{\partial x_3 \partial x_1} & \frac{\partial^2 J}{\partial x_3 \partial x_2} & \frac{\partial^2 J}{\partial x_3^2} \end{bmatrix} \tag{4.55}$$

Note that this Jacobian matrix is symmetric.

For a given Euler angle sequence, we could compute the partial derivatives in $\mathbf{F}$ and $\partial\mathbf{F}/\partial\mathbf{x}$ explicitly. However, because the trigonometric functions in the rotation matrix expand dramatically as a result of the product rule, we normally compute the derivatives numerically. Typically a central or one-sided finite difference scheme is used. For example, the term $\partial J/\partial x_1$ would be computed by calculating $J(\mathbf{x})$ for a particular set of x_1, x_2, and x_3. Then a small value would be added to x_1, say, $x_1 \to x_1 + \delta x_1$. Then

$$\frac{\partial J}{\partial x_1} \approx \frac{J(x_1 + \delta x_1, x_2, x_3) - J(x_1, x_2, x_3)}{\delta x_1} \tag{4.56}$$

This approach is applied to compute all the derivatives, including the second derivatives in the Jacobian. For example, to compute the second derivative $\partial^2 J/(\partial x_1 \partial x_2)$, we denote $\partial J/\partial x_1$ by J_{x_1}, and use the following:

$$\frac{\partial^2 J}{\partial x_1 \partial x_2} \approx \frac{J_{x_1}(x_1, x_2 + \delta x_2, x_3) - J_{x_1}(x_1, x_2, x_3)}{\delta x_2} \tag{4.57}$$

In practice, there are several issues that must be addressed. An excellent reference for this type of calculation is the text by Dennis and Schnabel.[8]

This numerical approach is unnecessary for solving this problem because an analytical solution exists, which we develop n the next subsection. However, this technique for solving a nonlinear problem is so useful that we have included it as an example.

4.5.4 q-Method

One elegant method of solving for the attitude which minimizes the loss function $J(\mathbf{R}^{bi})$ is the q-method.[1] We begin by expanding the loss function as follows:

$$\begin{aligned} J &= \frac{1}{2}\sum w_k(\mathbf{v}_{kb} - \mathbf{R}^{bi}\mathbf{v}_{ki})^\mathsf{T}(\mathbf{v}_{kb} - \mathbf{R}^{bi}\mathbf{v}_{ki}) && (4.58)\\ &= \frac{1}{2}\sum w_k(\mathbf{v}_{kb}{}^\mathsf{T}\mathbf{v}_{kb} + \mathbf{v}_{ki}{}^\mathsf{T}\mathbf{v}_{ki} - 2\mathbf{v}_{kb}{}^\mathsf{T}\mathbf{R}^{bi}\mathbf{v}_{ki}) && (4.59)\end{aligned}$$

The vectors are assumed to be normalized to unity, so the first two terms satisfy $\mathbf{v}_{kb}{}^\mathsf{T}\mathbf{v}_{kb} = \mathbf{v}_{ki}{}^\mathsf{T}\mathbf{v}_{ki} = 1$. Therefore, the loss function becomes

$$J = \sum w_k(1 - \mathbf{v}_{kb}{}^\mathsf{T}\mathbf{R}^{bi}\mathbf{v}_{ki}) \tag{4.60}$$

Minimizing $J(\mathbf{R})$ is the same as minimizing $J'(\mathbf{R}) = -\sum w_k\mathbf{v}_{kb}{}^\mathsf{T}\mathbf{R}^{bi}\mathbf{v}_{ki}$ or *maximizing* the gain function

$$g(\mathbf{R}) = \sum w_k\mathbf{v}_{kb}{}^\mathsf{T}\mathbf{R}^{bi}\mathbf{v}_{ki} \tag{4.61}$$

The key to solving this optimization problem is to restate the problem in terms of the quaternion $\bar{\mathbf{q}} = \begin{bmatrix}\mathbf{q}^\mathsf{T} & q_4\end{bmatrix}^\mathsf{T}$, for which

$$\mathbf{R} = (q_4^2 - \mathbf{q}^\mathsf{T}\mathbf{q})\mathbf{1} + 2\mathbf{q}\mathbf{q}^\mathsf{T} - 2q_4\mathbf{q}^\times \tag{4.62}$$

Since three parameters are the minimum required to uniquely determine attitude, any four-parameter representation of attitude has a single constraint relating the parameters. For quaternions, the constraint is

$$\bar{\mathbf{q}}^\mathsf{T}\bar{\mathbf{q}} = 1 \tag{4.63}$$

The gain function, $g(\mathbf{R})$, may be rewritten in terms of the quaternion instead of the rotation matrix.[9] This substitution leads to the form,

$$g(\bar{\mathbf{q}}) = \bar{\mathbf{q}}^\mathsf{T}\mathbf{K}\bar{\mathbf{q}} \tag{4.64}$$

where $\mathbf{K}$ is a 4 x 4 matrix given by

$$\mathbf{K} = \begin{bmatrix} \mathbf{S} - \sigma\mathbf{I} & \mathbf{Z} \\ \mathbf{Z}^{\mathsf{T}} & \sigma \end{bmatrix} \tag{4.65}$$

with

$$\begin{aligned}
\mathbf{B} &= \sum_{k=1}^{N} w_k \left(\mathbf{v}_{kb} \mathbf{v}_{ki}{}^{\mathsf{T}} \right) && (4.66) \\
\mathbf{S} &= \mathbf{B} + \mathbf{B}^{\mathsf{T}} && (4.67) \\
\mathbf{Z} &= \begin{bmatrix} B_{23} - B_{32} & B_{31} - B_{13} & B_{12} - B_{21} \end{bmatrix}^{\mathsf{T}} && (4.68) \\
\sigma &= \mathsf{tr}\,[\mathbf{B}] && (4.69)
\end{aligned}$$

To maximize the gain function, we take the derivative with respect to $\bar{\mathbf{q}}$, but since the quaternion elements are not independent the constraint must also be satisfied. Adding the constraint to the gain function with a Lagrange multiplier yields a new gain function,

$$g'(\bar{\mathbf{q}}) = \bar{\mathbf{q}}^{\mathsf{T}} \mathbf{K} \bar{\mathbf{q}} - \lambda \bar{\mathbf{q}}^{\mathsf{T}} \bar{\mathbf{q}} \tag{4.70}$$

Differentiating this gain function shows that $g'(\bar{\mathbf{q}})$ has a stationary value when

$$\mathbf{K}\bar{\mathbf{q}} = \lambda \bar{\mathbf{q}} \tag{4.71}$$

This equation is easily recognized as an eigenvalue problem. The optimal attitude is thus an eigenvector of the $\mathbf{K}$ matrix. However, there are four eigenvalues and they each have different eigenvectors. To see which eigenvalue corresponds to the optimal eigenvector (quaternion) which maximizes the gain function, recall

$$\begin{aligned}
g(\bar{\mathbf{q}}) &= \bar{\mathbf{q}}^{\mathsf{T}} \mathbf{K} \bar{\mathbf{q}} && (4.72) \\
&= \bar{\mathbf{q}}^{\mathsf{T}} \lambda \bar{\mathbf{q}} && (4.73) \\
&= \lambda \bar{\mathbf{q}}^{\mathsf{T}} \bar{\mathbf{q}} && (4.74) \\
&= \lambda && (4.75)
\end{aligned}$$

The largest eigenvalue of $\mathbf{K}$ maximizes the gain function. The eigenvector corresponding to this largest eigenvalue is the least-squares optimal estimate of the attitude.

There are many methods for directly calculating the eigenvalues and eigenvectors of a matrix, or approximating them. The q-method involves solving the eigenvalue/vector problem directly, but as seen in the next section, QUEST approximates the largest eigenvalue and solves for the corresponding eigenvector.

Example 4.3 *We use a two-sensor satellite to demonstrate the q-method. First, we use the Triad algorithm to generate an attitude estimate which we compare with the known attitude as well as with the q-method result.*

Given two vectors known in the inertial frame:

$$\mathbf{v}_{1i} = \begin{bmatrix} 0.2673 \\ 0.5345 \\ 0.8018 \end{bmatrix} \quad \mathbf{v}_{2i} = \begin{bmatrix} -0.3124 \\ 0.9370 \\ 0.1562 \end{bmatrix} \tag{4.76}$$

The "known" attitude is defined by a 3-1-3 Euler angle sequence with a 30° *rotation for each angle. The true attitude is represented by the rotation matrix between the inertial and body frames:*

$$\mathbf{R}^{bi}_{exact} = \begin{bmatrix} 0.5335 & 0.8080 & 0.2500 \\ -0.8080 & 0.3995 & 0.4330 \\ 0.2500 & -0.4330 & 0.8660 \end{bmatrix} \tag{4.77}$$

If the sensors measured the two vectors without error, then in the body frame the vectors would be:

$$\mathbf{v}_{1b_{exact}} = \begin{bmatrix} 0.7749 \\ 0.3448 \\ 0.5297 \end{bmatrix} \quad \mathbf{v}_{2b_{exact}} = \begin{bmatrix} 0.6296 \\ 0.6944 \\ -0.3486 \end{bmatrix} \tag{4.78}$$

Sensor measurements are not perfect however, and to model this uncertainty we introduce some error into the body-frame sensor measurements. A uniformly distributed random error is added to the sensor measurements, with a maximum value of $\pm 5^\circ$. *The two "measured" vectors are:*

$$\mathbf{v}_{1b} = \begin{bmatrix} 0.7814 \\ 0.3751 \\ 0.4987 \end{bmatrix} \quad \mathbf{v}_{2b} = \begin{bmatrix} 0.6163 \\ 0.7075 \\ -0.3459 \end{bmatrix} \tag{4.79}$$

Using the Triad algorithm and assuming $\mathbf{v}_1$ *is the "exact" vector, the satellite attitude is estimated by :*

$$\mathbf{R}^{bi}_{triad} = \begin{bmatrix} 0.5662 & 0.7803 & 0.2657 \\ -0.7881 & 0.4180 & 0.4518 \\ 0.2415 & -0.4652 & 0.8516 \end{bmatrix} \tag{4.80}$$

A useful approach to measuring the value of the attitude estimate makes use of the orthonormal nature of the rotation matrix; i.e., $\mathbf{R}^{\mathsf{T}}\mathbf{R} = \mathbf{1}$. *Since the Triad algorithm's estimate is not perfect, we compare the following to the identity matrix:*

$$\mathbf{R}^{bi^{\mathsf{T}}}_{triad}\mathbf{R}^{bi}_{exact} = \begin{bmatrix} 0.9992 & 0.03806 & 0.0094 \\ -0.0378 & 0.9989 & -0.0268 \\ -0.0104 & 0.02645 & 0.9996 \end{bmatrix} \tag{4.81}$$

This new matrix is the rotation matrix from the exact attitude to the attitude estimated by the Triad algorithm. The principal Euler angle of this matrix, and *therefore the attitude error of the estimate, is $\Phi = 2.72°$. For later comparison, the loss function for this rotation matrix is $J = 7.3609 \times 10^{-4}$.*

Using the q-method with the same inertial and measured vectors produces the $\mathbf{K}$ *matrix:*

$$\mathbf{K} = \begin{bmatrix} -1.1929 & 0.8744 & 0.9641 & 0.4688 \\ 0.8744 & 0.5013 & 0.3536 & -0.4815 \\ 0.9641 & 0.3536 & -0.5340 & 1.1159 \\ 0.4688 & -0.4815 & 1.1159 & 1.2256 \end{bmatrix} \tag{4.82}$$

Each measurement is equally weighted in the loss function. The largest eigenvalue and corresponding eigenvector of this matrix are:

$$\lambda_{max} = 1.9996 \tag{4.83}$$

$$\bar{\mathbf{q}} = \begin{bmatrix} 0.2643 \\ -0.0051 \\ 0.4706 \\ 0.8418 \end{bmatrix} \tag{4.84}$$

The corresponding rotation matrix is:

$$\mathbf{R}_q^{bi} = \begin{bmatrix} 0.5570 & 0.7896 & 0.2575 \\ -0.7951 & 0.4173 & 0.4402 \\ 0.2401 & -0.4499 & 0.8602 \end{bmatrix} \tag{4.85}$$

We determine the accuracy of this solution by computing the Euler angle of $\mathbf{R}_q^{bi\mathsf{T}} \mathbf{R}_{exact}^{bi}$. *For the q-method estimate of attitude, the attitude error and loss function values are:*

$$\Phi = 1.763° \tag{4.86}$$

$$J = 3.6808 \times 10^{-4} \tag{4.87}$$

As expected, the q-method finds the attitude matrix which minimizes the loss function. The attitude error is also lower than for the Triad algorithm. However, other measurement errors could create a case where the attitude error is actually larger for the q-method, even though it minimizes the loss function. We return to this point after the QUEST algorithm example.

4.5.5 QUEST

The q-method provides an optimal least-squares estimate of the attitude, given vector measurements in the body frame and information on those same vectors in some reference (often inertial) frame. The key to the method is to solve for eigenvalues and eigenvectors of the **K** matrix. While the eigenproblem may be solved easily using Matlab or other modern tools, the solution is numerically intensive. On-board computing requirements are a concern for satellite designers, so a more efficient way of solving the eigenproblem is needed. The QUEST algorithm provides a "cheaper" way to estimate the solution to the eigenproblem.[9,10]

Recall that the least-squares optimal attitude minimizes the loss function

$$J = \frac{1}{2}\sum_{k=1}^{N} w_k |\mathbf{v}_{kb} - \mathbf{R}^{bi}\mathbf{v}_{ki}|^2 \tag{4.88}$$

$$J = \sum w_k (1 - \mathbf{v}_{kb}{}^{\mathsf{T}}\mathbf{R}^{bi}\mathbf{v}_{ki}) \tag{4.89}$$

and maximizes the gain function

$$g = \sum w_k \mathbf{v}_{kb}{}^{\mathsf{T}}\mathbf{R}^{bi}\mathbf{v}_{ki} \tag{4.90}$$

$$g = \lambda_{opt} \tag{4.91}$$

Rearranging these two expressions provides a useful result:

$$\lambda_{opt} = \sum w_k - J \tag{4.92}$$

For the optimal eigenvalue, the loss function should be small. Thus a good approximation for the optimal eigenvalue is

$$\lambda_{opt} \approx \sum w_k \tag{4.93}$$

For many applications this approximation may be accurate enough. Reference 9 includes a Newton-Raphson method which uses the approximate eigenvalue as an initial guess. However, for sensor accuracies of 1° or better the accuracy of a 64-bit word is exceeded with just a single Newton-Raphson iteration.

Once the optimal eigenvalue has been estimated, the corresponding eigenvector must be calculated. The eigenvector is the quaternion which corresponds to the optimal attitude estimate. One way is to convert the quaternion in the eigenproblem to Rodriguez parameters, defined as

$$\mathbf{p} = \frac{\bar{\mathbf{q}}}{q_4} = \mathbf{a}\tan\frac{\Phi}{2} \tag{4.94}$$

The eigenproblem is rearranged as

$$\mathbf{p} = [(\lambda_{opt} + \sigma)\mathbf{1} - \mathbf{S}]^{-1}\,\mathbf{Z} \tag{4.95}$$

Taking the inverse in this expression is also a computationally intensive operation. Again, Matlab does it effortlessly, but solving for the inverse is not necessary. An efficient approach is to use Gaussian elimination or other linear system methods to solve the equation:

$$[(\lambda_{opt} + \sigma)\mathbf{1} - \mathbf{S}]\,\mathbf{p} = \mathbf{Z} \tag{4.96}$$

Once the Rodriguez parameters are found, the quaternion is calculated by

$$\bar{\mathbf{q}} = \frac{1}{\sqrt{1+\mathbf{p}^{\mathsf{T}}\mathbf{p}}}\begin{bmatrix}\mathbf{p}\\ 1\end{bmatrix} \tag{4.97}$$

One problem with this approach is that the Rodriguez parameters become singular when the rotation is π radians. Shuster and Oh have developed a method of sequential rotations which avoids this singularity.[9]

Example 4.4 *We repeat Example 4.3 using the QUEST method. Recall that the vector measurements are equally weighted, so we use a weighting vector of:*

$$\mathbf{w} = \begin{bmatrix}1\\ 1\end{bmatrix} \tag{4.98}$$

Using $\lambda_{opt} \approx \sum w_k = 2$*, the QUEST method produces an attitude estimate of*

$$\mathbf{R}^{bi}_{QUEST} = \begin{bmatrix} 0.5571 & 0.7895 & 0.2575\\ -0.7950 & 0.4175 & 0.4400\\ 0.2399 & -0.4499 & 0.8603 \end{bmatrix} \tag{4.99}$$

For the QUEST estimate of attitude, the attitude error and loss function values are:

$$\Phi = 1.773^\circ \tag{4.100}$$

$$J = 3.6810\times 10^{-4} \tag{4.101}$$

The QUEST method produces a rotation matrix which has a slightly larger loss function value, but without solving the entire eigenproblem. The actual attitude error of the estimate is comparable to that obtained using the q-method.

These least-squares estimates of attitude find the rotation matrix that minimizes the given loss function. *These methods do not guarantee that the actual attitude error is a minimum.* The actual attitude error may or may not be smaller than the Triad algorithm, or any other proposed method. In this two-vector example, with certain combinations of measurements errors, the Triad algorithm's actual attitude error may be less than the q-method's error. However, this example is different from the on-orbit attitude determination problem in that the actual attitude is known. In the real situation *the attitude error is never actually known.*

In general, using all the available vector information, as the least-squares methods do, provides a more consistently accurate result than the Triad algorithm. Recall that Triad uses only two vector measurements and assumes one is exactly correct. For systems with more than two sensors the least-squares methods clearly make better use of all available information.

4.6 Summary

The attitude determination problem is complicated by the fact that it is necessarily either underdetermined or overdetermined. The static attitude determination problem involves using two or more sensors to measure the components of distinct reference vectors in the body frame, and using mathematical models to calculate the components of the same reference vectors in an inertial frame. These vectors are then used in an algorithm to estimate the attitude in the form of one of the equivalent attitude representations, usually a rotation matrix, a set of Euler angles, or a quaternion. The simplest algorithm is the Triad algorithm, which only uses two reference vectors. More accurate algorithms are based on minimizing Wahba's loss function. While an analytical solution to this minimization problem exists (the q-method), an approximation is useful for finding a numerical solution (QUEST).

4.7 References and further reading

The handbook edited by Wertz[1] is the most complete reference on this material. The recent textbook by Sidi[3] includes useful appendices on attitude determination sensors and control actuators, but has little coverage of attitude determination algorithms. The space systems textbook edited by Pisacane and Moore[2] includes an excellent chapter on attitude determination and control written by Malcolm Shuster, the originator of the QUEST algorithm.[10] Star trackers are described in detail in the now-dated monograph by Quasius and McCanless.[11] Gyroscopic instruments are covered in Chapter 7.

Bibliography

[1] J. R. Wertz, editor. *Spacecraft Attitude Determination and Control.* D. Reidel, Dordrecht, Holland, 1978.

[2] Vincent L. Pisacane and Robert C. Moore, editors. *Fundamentals of Space Systems.* Oxford University Press, Oxford, 1994.

[3] Marcel J. Sidi. *Spacecraft Dynamics and Control: A Practical Engineering Approach.* Cambridge University Press, Cambridge, 1997.

[4] David A. Vallado. *Fundamentals of Astrodynamics and Applications.* McGraw-Hill, New York, 1997.

[5] Kristin L. Makovec. A nonlinear magnetic controller for nanosatellite applications. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2001.

[6] Wiley J. Larson and James R. Wertz, editors. *Space Mission Analysis and Design*. Microcosm, Inc., Torrance, CA, second edition, 1995.

[7] G. Wahba. A Least-Squares Estimate of Satellite Attitude, Problem 65.1. *SIAM Review*, pages 385–386, July 1966.

[8] J. E. Dennis, Jr. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[9] M. D. Shuster and S. D. Oh. Three-Axis Attitude Determination from Vector Observations. *Journal of Guidance and Control*, 4(1):70–77, 1981.

[10] Malcolm D. Shuster. In quest of better attitudes. In *Proceedings of the 11th AAS/AIAA Space Flight Mechanics Meeting*, pages 2089–2117, Santa Barbara, California, February 11-14 2001.

[11] Glen Quasius and Floyd McCanless. *Star Trackers and Systems Design*. Spartan Books, Washington, 1966.

4.8 Exercises

1. Compute $\mathbf{s}_i$ at epoch for the International Space Station position given in the following TLE:

```
ISS (ZARYA)
1 25544U 98067A   00256.59538941  .00002703  00000-0  29176-4 0   674
2 25544  51.5791  53.5981 0005510  45.6001 359.2109 15.67864156103651
```

2. Compute $\mathbf{m}_i$ at periapsis and apoapsis following epoch for the Molniya satellite orbit described in the following TLE:

```
MOLNIYA 1-91
1 25485U 98054A   00300.78960173  .00000175  00000-0  40203-2 0  6131
2 25485  63.1706 206.3462 7044482 281.6461  12.9979  2.00579102 15222
```

3. Develop Eq. (4.11).

4.9 Problems

1. A spacecraft has four attitude sensors, sensing four unit vectors (directions), $\hat{\mathbf{v}}_k$, $k = 1, 2, 3, 4$. These could be, for example, a sun sensor, an Earth horizon sensor, a star tracker, and a magnetometer. We know that the first sensor ($k = 1$) is

more accurate than the others, but we don't know the relative accuracy of the other three. At an instant in time, the four vectors measured by the sensors have body frame components

$$\mathbf{v}_{1b} = \begin{bmatrix} 0.8273 \\ 0.5541 \\ -0.0920 \end{bmatrix} \quad \mathbf{v}_{2b} = \begin{bmatrix} -0.8285 \\ 0.5522 \\ -0.0955 \end{bmatrix} \quad \mathbf{v}_{3b} = \begin{bmatrix} 0.2155 \\ 0.5522 \\ 0.8022 \end{bmatrix} \quad \mathbf{v}_{4b} = \begin{bmatrix} 0.5570 \\ -0.7442 \\ -0.2884 \end{bmatrix}$$

At the same time, the four vectors are determined to have inertial frame components

$$\mathbf{v}_{1i} = \begin{bmatrix} -0.1517 \\ -0.9669 \\ 0.2050 \end{bmatrix} \quad \mathbf{v}_{2i} = \begin{bmatrix} -0.8393 \\ 0.4494 \\ -0.3044 \end{bmatrix} \quad \mathbf{v}_{3i} = \begin{bmatrix} -0.0886 \\ -0.5856 \\ -0.8000 \end{bmatrix} \quad \mathbf{v}_{4i} = \begin{bmatrix} 0.8814 \\ -0.0303 \\ 0.5202 \end{bmatrix}$$

Because of the inaccuracies of the instruments, these vectors may not actually be unit vectors, so you should normalize them in your calculations.

(a) Use the Triad algorithm to obtain 3 different estimates of the attitude ($\mathbf{R}^{bi}$), using $\hat{\mathbf{v}}_1$ as the "exact" vector, and $\hat{\mathbf{v}}_2$, $\hat{\mathbf{v}}_3$, and $\hat{\mathbf{v}}_4$ as the second vector.

(b) Compute the error $J(\mathbf{R}^{bi})$ for each of the three estimates, using all four measurements in the calculation of J, and using weights $w_2 = w_3 = w_4 = 1$.

(c) Using the data and your calculations, make an educated ranking of sensors 2, 3, and 4 in terms of their expected accuracy.

2. Using the vectors in Problem 1, compute $\mathbf{R}^{bi}$ using Triad and the first two vectors, with $\mathbf{v}_1$ as the exact vector. Compute $\mathbf{R}_q^{bi}$ using the q-method with only vectors 1 and 2. Determine the principal Euler angle describing the difference between these two estimates.

3. Suppose a two-sensor spacecraft has one perfect sensor and one sensor that always gives a vector that is $1°$ off of the correct vector, but in an unknown direction. Let $\phi \in [0, 2\pi]$ be the angle describing the direction of this $1°$ error. Write a computer program that computes the principal Euler angle Φ_e describing the error of a given estimate. Plot Φ_e *vs.* ϕ for Triad and for the q-method, for at least three significantly different actual rotations. Discuss the results.

4. Develop a Triad-like algorithm using $\mathbf{w}_1 = \mathbf{v}_1 + \mathbf{v}_2$ and $\mathbf{w}_1 = \mathbf{v}_1 - \mathbf{v}_2$. Compare the performance of the resulting algorithm with that of the standard Triad algorithm.

5. Devise an example where $J = 0$, but the actual attitude error is non-zero.

4.10 Projects

1. Develop a subroutine implementing the deterministic attitude determination algorithm outlined in Section 4.5. The format for calling the subroutine should be something like `Rbi = triad(v1b,v2b,v1i,v2i)`.

2. Develop a subroutine implementing the numerical algorithm for estimating $\mathbf{R}^{bi}$ in terms of Euler angles. Use two of the measurements to compute an initial estimate using Triad. The function call should be of the form `theta = optest(vi,vb,w)`, where `vi` is a $3 \times N$ matrix of the reference vectors expressed in $\mathcal{F}_i$, `vb` is a $3 \times N$ matrix of the reference vectors expressed in $\mathcal{F}_b$, and `w` is a $1 \times N$ matrix of the weights.

3. Develop a subroutine implementing the q-method. The function call should be of the form `qopt = qmethod(vi,vb,w)`, where `vi` is a $3 \times N$ matrix of the reference vectors expressed in $\mathcal{F}_i$, `vb` is a $3 \times N$ matrix of the reference vectors expressed in $\mathcal{F}_b$, and `w` is a $1 \times N$ matrix of the weights.

4. Develop a subroutine implementing QUEST. The function call should be of the form `qopt = quest(vi,vb,w)`, where `vi` is a $3 \times N$ matrix of the reference vectors expressed in $\mathcal{F}_i$, `vb` is a $3 \times N$ matrix of the reference vectors expressed in $\mathcal{F}_b$, and `w` is a $1 \times N$ matrix of the weights.

5. Conduct a literature review on the subject of attitude determination. What algorithms have been developed since QUEST was introduced? What algorithms are used on current spacecraft?

Chapter 5

Rigid Body Dynamics

In Chapter 3 we developed the equations of motion for attitude kinematics. The main results of that chapter involve the description of attitude motion using attitude variables, such as rotation matrices, Euler angles, Euler axis/angle sets, or quaternions. Knowing the values of any of these variables at a particular instant of time allows the attitude analyst to visualize the orientation of one reference frame with respect to another. Examples of mission-oriented analysis include determining the location of a target on the Earth, the direction to the Sun, and the local direction of the Earth's magnetic field.

We also developed in Chapter 3 the differential equations describing how these variables depend on the angular velocity of the reference frame. We did not investigate how the angular velocity, $\vec{\omega}$, is determined, and that is the subject of this chapter.

In orbital dynamics, we typically model a spacecraft as a point mass or as a solid, rigid sphere, ignoring the rotational motion of the spacecraft about its mass center. This model is an excellent simplification for studying the motion of a satellite in its orbit — it works well for describing the motion of Europa about Jupiter, of Jupiter about the Sun, and of artificial satellites about the Earth. For artificial satellites, however, we usually need to know the orientation of the satellite with some degree of accuracy, so we need a more detailed model than the point mass model. A real spacecraft, of course, is a complicated piece of machinery with moving parts, flexible appendages, and partially filled fluid containers. However, even with all this complexity, a useful model is the *rigid body*, and that is the model we develop in this chapter.

We begin with a rigorous development of the equations of motion for a rigid body, both translational and rotational. Because the torques applied to spacecraft are typically quite small, we find it useful to consider the special case of torque-free motion. As noted above, all spacecraft include non-rigid elements. The most notable effect of non-rigidity is the effect of energy dissipation, which has important implications for spacecraft design. We also present analysis of two important examples of rigid body dynamics: the Lagrange top, and the unbalanced rotor.

5.1 Newton's Second Law

Newton's laws are applicable to a particle, or point mass, m. As noted previously, Newton's Second Law may be expressed as one second-order vector differential equation

$$m\ddot{\vec{\mathbf{r}}} = \vec{\mathbf{f}} \tag{5.1}$$

or as two first-order vector differential equations

$$\dot{\vec{\mathbf{p}}} = \vec{\mathbf{f}} \tag{5.2}$$

$$\dot{\vec{\mathbf{r}}} = \vec{\mathbf{p}}/m \tag{5.3}$$

Here $\vec{\mathbf{r}}$ is the position vector of the particle with respect to an inertially fixed point, $\vec{\mathbf{p}}$ is the linear momentum of the particle, and $\vec{\mathbf{f}}$ is the total applied force acting on the particle. Either of these equations describe the dynamics of a particle. Integration of these equations requires selection of specific coordinates or position variables. For example, with respect to an inertial reference frame, we might represent $\vec{\mathbf{r}}$ and $\vec{\mathbf{f}}$ by

$$\vec{\mathbf{r}} = r_1\hat{\mathbf{i}}_1 + r_2\hat{\mathbf{i}}_2 + r_3\hat{\mathbf{i}}_3 = \mathbf{r}^{\mathsf{T}}\left\{\hat{\mathbf{i}}\right\} \tag{5.4}$$

$$\vec{\mathbf{f}} = f_1\hat{\mathbf{i}}_1 + f_2\hat{\mathbf{i}}_2 + f_3\hat{\mathbf{i}}_3 = \mathbf{f}^{\mathsf{T}}\left\{\hat{\mathbf{i}}\right\} \tag{5.5}$$

In this case, the scalar second-order equations of motion are

$$m\ddot{r}_1 = f_1 \tag{5.6}$$

$$m\ddot{\mathbf{r}} = \mathbf{f} \quad \Leftrightarrow \quad m\ddot{r}_2 = f_2 \tag{5.7}$$

$$m\ddot{r}_3 = f_3 \tag{5.8}$$

and the scalar first-order equations of motion are

$$\dot{p}_1 = f_1 \tag{5.9}$$

$$\dot{\mathbf{p}} = \mathbf{f} \quad \Leftrightarrow \quad \dot{p}_2 = f_2 \tag{5.10}$$

$$\dot{p}_3 = f_3 \tag{5.11}$$

$$\dot{r}_1 = p_1/m \tag{5.12}$$

$$\dot{\mathbf{r}} = \mathbf{p}/m \quad \Leftrightarrow \quad \dot{r}_2 = p_2/m \tag{5.13}$$

$$\dot{r}_3 = p_3/m \tag{5.14}$$

One approach to developing the equations of motion of a spacecraft is to model the spacecraft as a system of point masses, m_i, $i = 1, ..., n$ and then develop the set of $3n$ second-order equations of motion or the set of $6n$ first-order equations of motion. This approach involves summations of the mass particles. For example, the total mass is $m = \sum_{i=1}^{n} m_i$, and the total applied force is $\vec{\mathbf{f}} = \sum_{i=1}^{n} \vec{\mathbf{f}}_i$. However, if we are to model a complicated spacecraft as a system of particles, the number n will be large and the system of equations will be quite unwieldy. In the following section, we develop the rigid body as an approximate mathematical model for spacecraft attitude motion.

5.2 The Rigid Body Model

Here we introduce the concept of a continuum of differential mass elements, dm. A differential mass element is quite a different concept from a mass particle. Whereas a mass particle has zero volume but finite mass, a differential mass element has infinitesimal volume and infinitesimal mass. The mathematical approach to collecting mass particles is summation, whereas the approach to collecting differential mass elements is integration.

A common problem in studying rigid body motion is the problem of integrating over the volume of the body. This operation may have a scalar integrand, a vector integrand, or a tensor integrand, and all three of these types of integrands are important in rigid body motion. Integrating over the volume of the body is a triple integral operation, but we use the following shorthand notation:

$$\int_{\mathcal{B}} f(\vec{\mathbf{r}})\, dV = \int_{r_{1min}}^{r_{1max}} \int_{r_{2min}(r_1)}^{r_{2max}(r_1)} \int_{r_{3min}(r_1,r_2)}^{r_{3max}(r_1,r_2)} f(\vec{\mathbf{r}})\, dr_3\, dr_2\, dr_1 \tag{5.15}$$

Let us consider two examples, using a rectangular prism shape with a reference frame coinciding with one corner of the prism as shown in Fig. 5.1. The first example is the scalar integral corresponding to the total mass, and the second is the vector integral corresponding to the first moment of inertia.

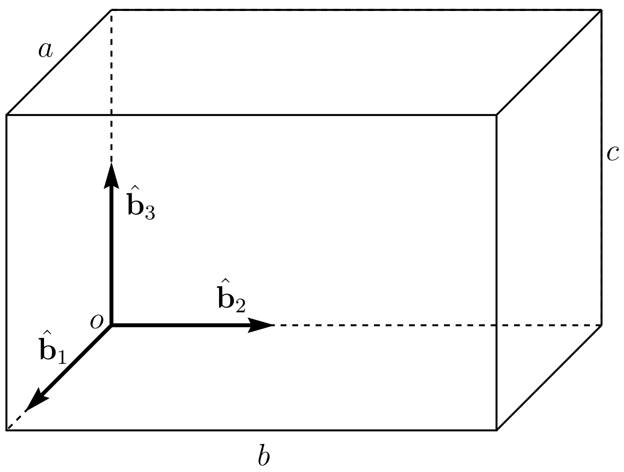

Figure 5.1: A rectangular prism for computing mass and first moment of inertia

Total mass. To compute the total mass of a rigid body, we need to know the density of the body, which we denote by μ. Units for μ are mass/length^3; for example, aluminum has density $\mu = 2.85 \times 10^3$ kg/m^3. In general, the density could vary from point to point within the body, or it could vary as a function of time. Thus, generally

$\mu = \mu(\vec{\mathbf{r}}, t)$. However, we generally assume that μ is constant, both spatially and temporally. Also, we usually use $dm = \mu\, dV$ as the variable of integration, even though m is not an independent variable. The meaning should always be clear. The total mass may then be expressed as

$$m = \int_{\mathcal{B}} \mu\, dV = \int_{\mathcal{B}} dm \tag{5.16}$$

Carrying out the indicated operations for the prism in Fig. 5.1, we obtain the expected result:

$$m = \int_0^a \int_0^b \int_0^c \mu\, dr_3\, dr_2\, dr_1 = \mu abc \tag{5.17}$$

Note that sometimes the total mass is called the *zeroth moment of inertia*, because it is a generalization of the first and second moments of inertia.

First moment of inertia. This vector quantity is defined as

$$\vec{\mathbf{c}}^o = \int_{\mathcal{B}} {}_o\vec{\mathbf{r}}\, dm \tag{5.18}$$

Clearly this quantity depends on the origin from which $\vec{\mathbf{r}}$ is measured, which motivates use of the *left* subscript o to denote that $\vec{\mathbf{r}}$ is measured *from* o, and the superscript o to denote that $\vec{\mathbf{c}}$ is the first moment *about* o. The matrix version of this equation is

$$\mathbf{c}^o = \int_{\mathcal{B}} {}_o\mathbf{r}\, dm \tag{5.19}$$

To compute the first moment, we must choose a reference frame, and the components depend on that choice of frame. For this example, we use the body frame and corner origin of Fig. 5.1. Thus, $\vec{\mathbf{c}} = \mathbf{c}_b{}^{\mathsf{T}}\{\hat{\mathbf{b}}\}$, $\vec{\mathbf{r}} = \mathbf{r}_b{}^{\mathsf{T}}\{\hat{\mathbf{b}}\}$, and

$$\mathbf{c}_b^o = \int_0^a \int_0^b \int_0^c \mu\, [r_1\ r_2\ r_3]^{\mathsf{T}}\, dr_3\, dr_2\, dr_1 = \frac{\mu}{2}\left[a^2bc\ \ ab^2c\ \ abc^2\right]^{\mathsf{T}} \tag{5.20}$$

Center of mass. We can use this result to determine the center of mass, c, which is defined as the point about which the first moment of inertia is zero; *i.e.* $\vec{\mathbf{c}}^c = \vec{\mathbf{0}}$. Mathematically, this definition leads to

$$\vec{\mathbf{c}}^c = \int_{\mathcal{B}} {}_c\vec{\mathbf{r}}\, dm = \vec{\mathbf{0}} \tag{5.21}$$

The position vector from o to any point may be written as

$${}_o\vec{\mathbf{r}} = {}^c_o\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}} \tag{5.22}$$

where ${}^c_o\vec{\mathbf{r}}$ is the position vector of c with respect to o, which is what we want to find, and ${}_c\vec{\mathbf{r}}$ is the position vector from c to a point in the body. The matrix version of this expression is

$${}_o\mathbf{r} = {}^c_o\mathbf{r} + {}_c\mathbf{r} \tag{5.23}$$

where we omit the subscript b since the reference frame is clear. Substituting this expression into Eq. (5.19), we find

$$\mathbf{c}^o = \int_{\mathcal{B}} \left({}^c_o\mathbf{r} + {}_c\mathbf{r}\right) \, dm = \int_{\mathcal{B}} {}^c_o\mathbf{r} \, dm = {}^c_o\mathbf{r} \int_{\mathcal{B}} dm = m\,{}^c_o\mathbf{r} \tag{5.24}$$

Thus,

$${}^c_o\mathbf{r} = \frac{1}{m}\mathbf{c}^o = \frac{\mu}{2m}\left[a^2bc \;\; ab^2c \;\; abc^2\right]^{\mathsf{T}} \tag{5.25}$$

Using the expression for the mass m developed in Eq. (5.17), we obtain

$${}^c_o\mathbf{r} = \frac{1}{2}\left[a \;\; b \;\; c\right]^{\mathsf{T}} \tag{5.26}$$

which is obvious from the figure. However, this same procedure is applicable to more complicated geometric figures and to cases where μ is not constant.

We now continue with the development of equations of motion for a rigid body, beginning with the linear momentum equations, and then developing the angular momentum equations.

Linear momentum. The linear momentum of a rigid body is defined as

$$\vec{\mathbf{p}} = \int_{\mathcal{B}} \vec{\mathbf{v}} \, dm \tag{5.27}$$

where $\vec{\mathbf{v}}$ is the time derivative of the position vector $\vec{\mathbf{r}}$ of a differential mass element $dm = \mu \, dV$. This position vector must be measured from an inertial origin, and the derivative is taken with respect to inertial space. The position vector can be written as ${}^o_O\vec{\mathbf{r}} + {}_o\vec{\mathbf{r}}$, where ${}^o_O\vec{\mathbf{r}}^o$ is the position vector from the inertial origin *to* o and ${}_o\vec{\mathbf{r}}$ is the vector *from* o to a point in the body.

First we must ask, *What is the velocity* $\vec{\mathbf{v}}$ *of a differential mass element?* We describe the motion of the body frame by the velocity of the origin, ${}^o_O\vec{\mathbf{v}}$ and the angular velocity of the body frame with respect to the inertial frame, $\vec{\omega}^{bi}$. Then the velocity of a mass element is $\vec{\mathbf{v}} = {}^o_O\vec{\mathbf{v}} + \vec{\omega}^{bi} \times {}_o\vec{\mathbf{r}}$. Thus the linear momentum is

$$\vec{\mathbf{p}} = \int_{\mathcal{B}} \left({}^o_O\vec{\mathbf{v}} + \vec{\omega}^{bi} \times {}_o\vec{\mathbf{r}}\right) \, dm \tag{5.28}$$

The velocity of the origin and the angular velocity of the frame do not depend on the particular differential mass element, so they are constant with respect to the integration, so that the momentum becomes

$$\vec{\mathbf{p}} = m\,{}^o_O\vec{\mathbf{v}} + \vec{\omega}^{bi} \times \int_{\mathcal{B}} {}_o\vec{\mathbf{r}} \, dm \tag{5.29}$$

The integral on the right can be recognized as the first moment of inertia about the origin, so that

$$\vec{\mathbf{p}} = m\,{}^O_o\vec{\mathbf{v}} + \vec{\omega}^{bi} \times \vec{\mathbf{c}}^o \tag{5.30}$$

Thus, if o is the mass center c, then the linear momentum is simply $\vec{\mathbf{p}} = m{}^c_O\vec{\mathbf{v}}$.

Now, we can extend Newton's second law, $\dot{\vec{\mathbf{p}}} = \vec{\mathbf{f}}$ to the rigid body case as follows. The total force acting on a differential element of mass is

$$d\vec{\mathbf{f}} = \dot{\vec{\mathbf{v}}}\, dm \tag{5.31}$$

Integrating both sides of this expression over the body, we get

$$\int_{\mathcal{B}} d\vec{\mathbf{f}} = \int_{\mathcal{B}} \dot{\vec{\mathbf{v}}}\, dm = \dot{\vec{\mathbf{p}}} \tag{5.32}$$

Before integrating the differential forces, we first distinguish between the external applied forces, $d\vec{\mathbf{f}}_{ext}$, and the internal forces due to the other differential mass elements, $d\vec{\mathbf{f}}_{int}$. Thus

$$\int_{\mathcal{B}} \left(d\vec{\mathbf{f}}_{ext} + d\vec{\mathbf{f}}_{int}\right) = \dot{\vec{\mathbf{p}}} \tag{5.33}$$

By Newton's third law, the internal forces are equal and opposite, so that the integral $\int_{\mathcal{B}} d\vec{\mathbf{f}}_{int} = \vec{\mathbf{0}}$. Thus

$$\int_{\mathcal{B}} d\vec{\mathbf{f}}_{ext} = \dot{\vec{\mathbf{p}}} \tag{5.34}$$

We define the integral of the external forces by

$$\vec{\mathbf{f}} = \int_{\mathcal{B}} d\vec{\mathbf{f}}_{ext} \tag{5.35}$$

so that the translational equations of motion for a rigid body are

$$\vec{\mathbf{p}} = m{}^o_O\vec{\mathbf{v}} + \vec{\omega}^{bi} \times \vec{\mathbf{c}}^o \tag{5.36}$$

$$\vec{\mathbf{f}} = \dot{\vec{\mathbf{p}}} \tag{5.37}$$

These are vector equations of motion. To integrate them, we need to choose a reference frame and work with the components of the appropriate matrices in that frame. For example, we usually work with a body-fixed reference frame, $\mathcal{F}_b$, whose angular velocity with respect to inertial space is $\vec{\omega}^{bi}$. To differentiate a vector expressed in a rotating reference frame, we must recall the rule:

$$\frac{d}{dt}\left[\left\{\hat{\mathbf{b}}\right\}^{\mathsf{T}}\mathbf{a}\right] = \left\{\hat{\mathbf{b}}\right\}^{\mathsf{T}}\left[\dot{\mathbf{a}} + \omega^{\times}\mathbf{a}\right] \tag{5.38}$$

Applying this rule to Eqs. (5.36) and (5.37), we get

$$\mathbf{p} = m{}^o_O\mathbf{v} + \omega^{bi\times}\mathbf{c}^o \tag{5.39}$$

$$\mathbf{f} = \dot{\mathbf{p}} + \omega^{bi\times}\mathbf{p} \tag{5.40}$$

Note that the velocity term ${}^o_O\mathbf{v}$ may also involve using Eq. (5.38).

5.3 Euler's Law for Moment of Momentum

Now we develop the rotational equations of motion for a rigid body. One approach to developing these equations is to develop an expression for the angular momentum, and then apply Newton's second law. Another approach is to apply Euler's Law

$$\dot{\vec{\mathbf{h}}} = \vec{\mathbf{g}} \tag{5.41}$$

where $\vec{\mathbf{h}}$ is the angular momentum about the mass center, and $\vec{\mathbf{g}}$ is the net applied moment about the mass center. One advantage of the approach based on Newton's second law is that it also yields valid expressions when the moments are taken about some point other than the mass center. This generality is frequently useful in spacecraft dynamics problems where there are flexible components resulting in a movable mass center.

As we develop in the remainder of this section, the angular momentum can be expressed in $\mathcal{F}_b$ as $\mathbf{h} = \mathbf{I}\omega$, where $\mathbf{I}$ is the moment of inertia matrix, and ω is the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$ (*i.e.*, $\omega = \omega^{bi}$). Since $\mathbf{I}$ is constant in the body frame, we want to express the equations of motion in that frame (so that we do not have to deal with $\dot{\mathbf{I}}$ terms), and the body frame is rotation. Recalling Eq. (5.38), then, Eq. (5.41) in body-frame components is written:

$$\dot{\mathbf{h}} + \omega^{\times}\mathbf{h} = \mathbf{g} \tag{5.42}$$

Since $\mathbf{h} = \mathbf{I}\omega$, and since $\dot{\mathbf{I}} = \mathbf{0}$ *in the body frame*, we can rewrite this equation as

$$\dot{\omega} = -\mathbf{I}^{-1}\omega^{\times}\mathbf{I}\omega + \mathbf{I}^{-1}\mathbf{g} \tag{5.43}$$

If a *principal* reference frame is used, then $\mathbf{I}$ is diagonal; *i.e.*, $\mathbf{I} = \mathrm{diag}(I_1, I_2, I_3)$. This matrix equation can then be expanded to obtain the "standard" version of Euler's equations for the rotational motion of a rigid body:

$$\dot{\omega}_1 = \frac{I_2 - I_3}{I_1}\omega_2\omega_3 + \frac{g_1}{I_1} \tag{5.44}$$

$$\dot{\omega}_2 = \frac{I_3 - I_1}{I_2}\omega_1\omega_3 + \frac{g_2}{I_2} \tag{5.45}$$

$$\dot{\omega}_3 = \frac{I_1 - I_2}{I_3}\omega_1\omega_2 + \frac{g_3}{I_3} \tag{5.46}$$

Therefore, to determine the motion of a rigid body, we need to simultaneously integrate Eqs. (5.43) and one of the sets of kinematics differential equations developed in Chapter 3 and summarized in § 3.2.2.

In the remainder of this chapter we further develop these equations for the kinetics of a rigid body.

*Angular Momentum and Moment of Momentum**. There are actually at least three quantities that are commonly called *angular momentum* in the literature. They are all related, but have slightly different definitions. We define two different vector quantities: *angular momentum* and *momentum of momentum.*

The *moment of momentum*, $\vec{\mathbf{H}}^o$, of a rigid body about a point o is defined as

$$\vec{\mathbf{H}}^o = \int_{\mathcal{B}} {}_o\vec{\mathbf{r}} \times \vec{\mathbf{v}}\, dm \tag{5.47}$$

where ${}_o\vec{\mathbf{r}}$ is the position vector of a differential mass element from point o, and $\vec{\mathbf{v}}$ is the velocity of the mass element with respect to inertial space. Note that the quantity $\vec{\mathbf{v}}\, dm$ is in fact the momentum of the differential mass element, and by taking the cross product with the moment arm ${}_o\vec{\mathbf{r}}$, we are forming the moment of momentum about o.

The *angular momentum*, $\vec{\mathbf{h}}^o$, of a rigid body about a point o is defined as

$$\vec{\mathbf{h}}^o = \int_{\mathcal{B}} {}_o\vec{\mathbf{r}} \times {}_o\dot{\vec{\mathbf{r}}}\, dm \tag{5.48}$$

where the velocity $\vec{\mathbf{v}}$ appearing in $\vec{\mathbf{H}}^o$ has been replaced with the time derivative of the moment arm vector itself. Note that ${}_o\dot{\vec{\mathbf{r}}}\, dm$ is not the momentum of the differential mass element, but it is "momentum-like;" *i.e.*, it has the correct units.

We subsequently derive expressions for $\dot{\vec{\mathbf{H}}}^o$ and $\dot{\vec{\mathbf{h}}}^o$, but first we develop expressions for $\vec{\mathbf{H}}^o$ and $\vec{\mathbf{h}}^o$ involving the angular velocity of the body reference frame and the moment of inertia tensor.

Beginning with Eq. (5.47), we use the fact that the velocity of the differential mass element dm may be written as

$$\vec{\mathbf{v}} = {}^o_O\vec{\mathbf{v}} + \vec{\omega} \times {}_o\vec{\mathbf{r}} \tag{5.49}$$

where ${}^o_O\vec{\mathbf{v}}$ is the velocity of the point o, $\vec{\omega}$ is the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$, and ${}_o\vec{\mathbf{r}}$ is the position vector from o to the mass element. With this expression for $\vec{\mathbf{v}}$, Eq. (5.47) may be written as

$$\vec{\mathbf{H}}^o = \int_{\mathcal{B}} {}_o\vec{\mathbf{r}} \times \left[{}^o_O\vec{\mathbf{v}} + \vec{\omega} \times {}_o\vec{\mathbf{r}}\right] dm \tag{5.50}$$

The integrand may be expanded so that

$$\vec{\mathbf{H}}^o = \int_{\mathcal{B}} \left[{}_o\vec{\mathbf{r}} \times {}^o_O\vec{\mathbf{v}}\right] dm + \int_{\mathcal{B}} \left[{}_o\vec{\mathbf{r}} \times (\vec{\omega} \times {}_o\vec{\mathbf{r}})\right] dm \tag{5.51}$$

*An excellent development of this material can be found in the out-of-print text by Peter Likins.[1] Professor Likins studied with the well-known dynamicist Thomas Kane at Stanford, and subsequently taught space mechanics at UCLA with D. Lewis Mingori and Robert Roberson. While at UCLA he also coined the term "dual-spin spacecraft." He went on to be Dean of Engineering at Columbia University and was President of Lehigh University for several years before becoming President of the University of Arizona.

The triple vector product in the second integral can be expanded using the vector identity

$$\vec{\mathbf{a}} \times \left(\vec{\mathbf{b}} \times \vec{\mathbf{c}}\right) = \vec{\mathbf{b}}\left(\vec{\mathbf{a}} \cdot \vec{\mathbf{c}}\right) - \vec{\mathbf{c}}\left(\vec{\mathbf{a}} \cdot \vec{\mathbf{b}}\right) \tag{5.52}$$

In this case $\vec{\mathbf{a}} = \vec{\mathbf{c}} = {}_o\vec{\mathbf{r}}$, and $\vec{\mathbf{b}} = \vec{\omega}$, so that the expression becomes

$$\int_{\mathcal{B}} \left[{}_o\vec{\mathbf{r}} \times \left(\vec{\omega} \times {}_o\vec{\mathbf{r}}\right)\right] dm = \int_{\mathcal{B}} \left[\vec{\omega}\left({}_o\vec{\mathbf{r}} \cdot {}_o\vec{\mathbf{r}}\right) - {}_o\vec{\mathbf{r}}\left(\vec{\omega} \cdot {}_o\vec{\mathbf{r}}\right)\right] dm \tag{5.53}$$

We can reverse the order of the second dot product, and *we can omit the parentheses in the integrand.* That is, we write

$$\vec{\omega}\left({}_o\vec{\mathbf{r}} \cdot {}^o\vec{\mathbf{r}}\right) = {}_o\vec{\mathbf{r}} \cdot {}_o\vec{\mathbf{r}}\,\vec{\omega} \tag{5.54}$$

and

$${}_o\vec{\mathbf{r}}\left(\vec{\omega} \cdot {}_o\vec{\mathbf{r}}\right) = {}_o\vec{\mathbf{r}}{}_o\vec{\mathbf{r}} \cdot \vec{\omega} \tag{5.55}$$

Whenever we write three vectors in the form $\vec{\mathbf{a}}\vec{\mathbf{b}}\cdot\vec{\mathbf{c}}$, parentheses are implied around the two vectors with the $\cdot$ between them. Thus, $\vec{\mathbf{a}}\vec{\mathbf{b}} \cdot \vec{\mathbf{c}} = \vec{\mathbf{a}}\left(\vec{\mathbf{b}} \cdot \vec{\mathbf{c}}\right)$, and $\vec{\mathbf{a}} \cdot \vec{\mathbf{b}}\vec{\mathbf{c}} = \left(\vec{\mathbf{a}} \cdot \vec{\mathbf{b}}\right)\vec{\mathbf{c}}$. There is no ambiguity in this notation, and we discuss it in more detail later.

Now we introduce a mathematical object, $\vec{\mathbf{1}}$, defined by the following properties:

$$\vec{\mathbf{1}} \cdot \vec{\mathbf{v}} = \vec{\mathbf{v}} \cdot \vec{\mathbf{1}} = \vec{\mathbf{v}} \tag{5.56}$$

for any vector $\vec{\mathbf{v}}$. Like the $\vec{\mathbf{a}}\vec{\mathbf{b}} \cdot \vec{\mathbf{c}}$ notation, this notation is simple, unambiguous, and convenient for working with angular momentum and moment of momentum. We use this operator in the expression in Eq. (5.54):

$${}_o\vec{\mathbf{r}} \cdot {}_o\vec{\mathbf{r}}\,\vec{\omega} = {}_o\vec{\mathbf{r}} \cdot {}_o\vec{\mathbf{r}}\,\vec{\mathbf{1}} \cdot \vec{\omega} = {}_o r^2 \vec{\mathbf{1}} \cdot \vec{\omega} \tag{5.57}$$

We can now write the moment of momentum vector as

$$\vec{\mathbf{H}}^o = \int_{\mathcal{B}} \left[{}_o\vec{\mathbf{r}} \times {}^o_O\vec{\mathbf{v}}\right] dm + \int_{\mathcal{B}} \left[({}_o r)^2 \vec{\mathbf{1}} \cdot \vec{\omega} - {}_o\vec{\mathbf{r}}{}_o\vec{\mathbf{r}} \cdot \vec{\omega}\right] dm \tag{5.58}$$

With respect to the variables of integration, the two vectors ${}^o_O\vec{\mathbf{v}}^o$ and $\vec{\omega}$ are both constants (note that they are not constants with respect to time, though). Thus these two terms can be brought outside of the integral sign to get

$$\vec{\mathbf{H}}^o = \int_{\mathcal{B}} {}_o\vec{\mathbf{r}}\,dm \times {}^o_O\vec{\mathbf{v}} + \int_{\mathcal{B}} \left[({}_o r)^2 \vec{\mathbf{1}} - {}_o\vec{\mathbf{r}}{}_o\vec{\mathbf{r}}\right] dm \cdot \vec{\omega} \tag{5.59}$$

The first integral is the first moment of inertia about o, and the second integral is a mathematical object called the second moment of inertia about o. We denote this object by

$$\vec{\mathbf{I}}^o = \int_{\mathcal{B}} \left[{}_o r^2 \vec{\mathbf{1}} - {}_o\vec{\mathbf{r}}{}_o\vec{\mathbf{r}}\right] dm \tag{5.60}$$

We develop the moment of inertia in more detail below. For now, we simply note that it is a constant property of a rigid body.

Now we can write the moment of momentum as

$$\vec{\mathbf{H}}^o = \vec{\mathbf{c}}^o \times {}^o_O\vec{\mathbf{v}} + \vec{\mathbf{I}}^o \cdot \vec{\omega} \tag{5.61}$$

Note that if we choose the mass center of the rigid body as the origin, *i.e.*, $o \equiv c$, then $\vec{\mathbf{c}}^c = \vec{\mathbf{0}}$, and $\vec{\mathbf{H}}^c = \vec{\mathbf{I}}^c \cdot \vec{\omega}$.

Similar operations lead to the following expression for the angular momentum:

$$\vec{\mathbf{h}}^o = \vec{\mathbf{I}}^o \cdot \vec{\omega} \tag{5.62}$$

Note that when $o \equiv c$, the distinction between angular momentum and moment of momentum vanishes. This equivalence is part of the reason that these two expressions are sometimes confused in the literature. The careful student of spacecraft dynamics always clearly states which vector is being used.

In what follows, we work with the special case where $o \equiv c$, and we use the angular momentum.

Second Moment of Inertia. The integral in Eq. (5.60) is an important quantity called the *second moment of inertia tensor.* If we express the vector ${}_c\vec{\mathbf{r}}$ in a body-fixed frame, say $\mathcal{F}_b$, then we can develop a matrix version of $\vec{\mathbf{I}}$. That is, we write ${}_c\vec{\mathbf{r}} = {}_c\mathbf{r}_b^T\{\vec{\mathbf{b}}\} = \{\vec{\mathbf{b}}\}^{\mathsf{T}}{}_c\mathbf{r}_b$, and substitute one or the other of these expressions into Eq. (5.60). This substitution leads to

$$\vec{\mathbf{I}}^c = \left\{\vec{\mathbf{b}}\right\}^{\mathsf{T}} \int_{\mathcal{B}} \left[{}_c\mathbf{r}_b^T\,{}_c\mathbf{r}_b\mathbf{1} - {}_c\mathbf{r}_b\,{}_c\mathbf{r}_b^T\right] dm \left\{\vec{\mathbf{b}}\right\} \tag{5.63}$$

The integral is the *moment of inertia matrix*, denoted

$$\mathbf{I}_b^c = \int_{\mathcal{B}} \left[{}_c\mathbf{r}_b^T\,{}_c\mathbf{r}_b\mathbf{1} - {}_c\mathbf{r}_b\,{}_c\mathbf{r}_b^T\right] dm \tag{5.64}$$

so that

$$\vec{\mathbf{I}}^c = \left\{\vec{\mathbf{b}}\right\}^{\mathsf{T}}\mathbf{I}_b^c\left\{\vec{\mathbf{b}}\right\} \tag{5.65}$$

or

$$\mathbf{I}_b^c = \left\{\vec{\mathbf{b}}\right\} \cdot \vec{\mathbf{I}}^c \cdot \left\{\vec{\mathbf{b}}\right\}^{\mathsf{T}} \tag{5.66}$$

Note that the moment of inertia matrix depends explicitly on the choice of point about which moments are taken (c here) and on the choice of body-fixed reference frame in which the inertia tensor is expressed. This frame-dependence is equivalent to the fact that the components of a vector depend on the choice of reference frame.

We frequently need to be able to change the reference point or to change the reference frame. The two tools we need to be able to do make these changes are the *parallel axis theorem* and the *change of vector basis theorem.*

Parallel Axis Theorem. Suppose we know the moment of inertia matrix about the mass center, c, expressed in a particular reference frame, and we want the moment of inertia matrix about a different point o, defined by $\mathbf{r}^{o/c}$. The former is denoted $\mathbf{I}^c$ and the latter is denoted $\mathbf{I}^o$ (we omit the subscript, since the reference frame is the same for both).

Returning to the definition of the inertia matrix in Eq. (5.64), we express the term ${}_c\mathbf{r}$ as

$$ {}_c\mathbf{r} = {}^o_c\mathbf{r} + {}_o\mathbf{r} \tag{5.67}$$

Upon substituting this expression into Eq. (5.64), we obtain

$$ \mathbf{I}^c = \int_{\mathcal{B}} \left[{}_o\mathbf{r}^{\mathsf{T}}{}_o\mathbf{r}\mathbf{1} - {}_o\mathbf{r}_o\mathbf{r}^T \right] dm - m\left[{}^c_o\mathbf{r}^T{}^c_o\mathbf{r}\mathbf{1} - {}^c_o\mathbf{r}^c_o\mathbf{r}^T \right] \tag{5.68}$$

The integral in this expression is recognized as the moment of inertia matrix about the point o (expressed in $\mathcal{F}_b$). Thus, the parallel axis theorem may be written as

$$ \mathbf{I}^o = \mathbf{I}^c + m\left[{}^c_o\mathbf{r}^T{}^c_o\mathbf{r}\mathbf{1} - {}^c_o\mathbf{r}^c_o\mathbf{r}^T \right] \tag{5.69}$$

This theorem may also be written as

$$ \mathbf{I}^o = \mathbf{I}^c - m{}^c_o\mathbf{r}^{\times}{}^c_o\mathbf{r}^{\times} \tag{5.70}$$

where we have made use of the result of the identity

$$ \mathbf{a}^{\times}\mathbf{a}^{\times} = \mathbf{1} - \mathbf{a}\mathbf{a}^{\times} \tag{5.71}$$

which is left as an exercise.

Change Of Vector Basis Theorem. The parallel axis theorem is used to change the reference point about which the moment of inertia is taken. The *change of vector basis theorem* is used to express the moment of inertia matrix in a different reference frame, but about the same reference point. For this development, it makes no difference whether the reference point is the mass center or an arbitrary point o, so we omit the superscript to simplify the notation.

Suppose we know the inertia matrix with respect to a certain body-fixed reference frame, $\mathcal{F}_b$, and we want to know it with respect to a different body-fixed reference frame, $\mathcal{F}_a$. Ordinarily, we know the relative attitude between these two reference frames, usually expressed as a rotation matrix, $\mathbf{R}^{ab}$, that takes vectors from $\mathcal{F}_b$ to $\mathcal{F}_a$. That is,

$$ \mathbf{v}_a = \mathbf{R}^{ab}\mathbf{v}_b \tag{5.72}$$

We want to develop an equivalent expression involving $\mathbf{R}^{ab}$, $\mathbf{I}_a$, and $\mathbf{I}_b$.

Again, we begin with Eq. (5.64), substituting $\mathbf{R}^{ba}\mathbf{r}_a$ for $\mathbf{r}_b$. Upon multiplying through and factoring out the rotation matrices (which do not depend on the independent variable of integration), we find that

$$ \mathbf{I}_b = \mathbf{R}^{ba} \int_{\mathcal{B}} \left[\mathbf{r}_a^T\mathbf{r}_a\mathbf{1} - \mathbf{r}_a\mathbf{r}_a^T \right] dm\, \mathbf{R}^{ab} \tag{5.73}$$

The integral may be immediately recognized as $\mathbf{I}_a$, so that

$$\mathbf{I}_b = \mathbf{R}^{ba}\mathbf{I}_a\mathbf{R}^{ab} \tag{5.74}$$

Using the properties of rotation matrices, this identity may also be written as

$$\mathbf{I}_a = \mathbf{R}^{ab}\mathbf{I}_b\mathbf{R}^{ba} \tag{5.75}$$

Equations (5.74) and (5.75) are the tensor equivalents of Eq. (5.72). Note the ordering of the sub- and superscripts.

Principal Axes. One of the most important applications of the change of basis vector theorem is the use of a rigid body's *principal axes.* The principal axes comprise a body-fixed reference frame for which the moment of inertia matrix is diagonal; *i.e.*, the moment of inertia matrix takes the form

$$\mathbf{I} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \tag{5.76}$$

That such a reference frame always exists is a consequence of the fact that the moment of inertia tensor is *symmetric*; *i.e.*,

$$\vec{\mathbf{u}} \cdot \vec{\mathbf{I}} \cdot \vec{\mathbf{v}} = \vec{\mathbf{v}} \cdot \vec{\mathbf{I}} \cdot \vec{\mathbf{u}} \tag{5.77}$$

for any vectors $\vec{\mathbf{u}}$ and $\vec{\mathbf{v}}$. This fact means that *in any reference frame* the moment of inertia matrix is symmetric; *i.e.*, $\mathbf{I} = \mathbf{I}^{\mathsf{T}}$. A well-known theorem states that for any symmetric matrix, its eigenvalues are real, and its eigenvectors are orthogonal. For the moment of inertia matrix, these eigenvalues are the principal moments of inertia, and the eigenvectors provide the directions of the principal axes.

If we know $\mathbf{I}$ in a non-principal reference frame $\mathcal{F}_a$, and want to know $\mathbf{I}$ in principal reference frame, $\mathcal{F}_b$, we must find the rotation matrix $\mathbf{R}^{ba}$, which requires the establishment and solution of an eigenvalue problem. Beginning with Eq. (5.75), we post-multiply by $\mathbf{R}^{ab}$ to obtain

$$\mathbf{I}_a\mathbf{R}^{ab} = \mathbf{R}^{ab}\mathbf{I}_b \tag{5.78}$$

Recalling that the columns of $\mathbf{R}^{ab}$ are the components of the unit vectors of $\mathcal{F}_b$ expressed in $\mathcal{F}_a$, and that the principal moment of inertia matrix $\mathbf{I} = \mathrm{diag}[I_1, I_2, I_3]$, we can rewrite Eq. (5.78) as

$$\mathbf{I}_a \left[\mathbf{b}_{1a}\ \ \mathbf{b}_{2a}\ \ \mathbf{b}_{3a}\right] = \left[I_1\mathbf{b}_{1a}\ \ I_2\mathbf{b}_{2a}\ \ I_3\mathbf{b}_{3a}\right] \tag{5.79}$$

or as the three separate equations

$$\mathbf{I}_a\mathbf{b}_{1a} = I_1\mathbf{b}_{1a} \qquad \mathbf{I}_a\mathbf{b}_{2a} = I_2\mathbf{b}_{2a} \qquad \mathbf{I}_a\mathbf{b}_{3a} = I_3\mathbf{b}_{3a} \tag{5.80}$$

Each of these three equations is in the form

$$\mathbf{I}_a \mathbf{x} = \lambda \mathbf{x} \tag{5.81}$$

which is recognized as the eigenvalue problem, where solutions $(\lambda, \mathbf{x})$ are eigenvalues and eigenvectors of $\mathbf{I}_a$. In general, since $\mathbf{I}_a$ is a 3×3 matrix, there are three eigenvalues, $\lambda_i, i = 1, 2, 3$, each with an associated eigenvector, $\mathbf{x}_i$. Furthermore, since $\mathbf{I}_a$ is a real symmetric matrix, all its eigenvalues are real, and the associated eigenvectors are mutually orthogonal. The eigenvalues of $\mathbf{I}_a$ are the principal moments of inertia of the body, and the eigenvectors are the components of the principal axes expressed in $\mathcal{F}_a$.

5.4 Summary of Notation

There are several subscripts and superscripts used in this and preceding chapters. This table summarizes the meanings of these symbols.

Symbol	Meaning
$\vec{\mathbf{c}}^o$	first moment of inertia *relative to* or *about* o (a vector)
$\mathbf{c}_b^o$	first moment of inertia *relative to* or *about* o expressed in $\mathcal{F}_b$ (a 3×1 matrix)
${}_o\vec{\mathbf{r}}$	position vector from o to a point
${}_O^o\vec{\mathbf{r}}$	position vector from an inertial origin (O) to point o
${}_o^c\vec{\mathbf{r}}$	position vector from o to point c
${}_O^o\vec{\mathbf{v}}$	velocity vector of point o with respect to an inertial origin
$\vec{\mathbf{H}}^o$	moment of momentum of a rigid body about point o
$\vec{\mathbf{h}}^o$	angular momentum of a rigid body about point o
$\vec{\omega}^{bi}$	angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$
$\mathbf{R}^{bi}$	rotation matrix of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$
$\vec{\mathbf{1}}$	the identity tensor
$\mathbf{1}$	the identity matrix
$\vec{\mathbf{I}}^o$	the moment of inertia of a rigid body with respect to o (a tensor)
$\mathbf{I}_b^o$	the moment of inertia of a rigid body with respect to o expressed in $\mathcal{F}_b$ (a 3×3 matrix)

5.5 References and further reading

The textbook by Likins[1] has one of the best developments of rigid body dynamics that I have seen. Unfortunately the book is out of print. If you are interested in the history of Eq. (5.41), I highly recommend Truesdell's essay *Whence the Law of Moment of Momentum?*, which appears in Ref. 2. The advanced dynamics book by Meirovitch[3] develops rigid body equations of motion, and has two chapters of space-related applications. Hughes's[4] textbook gives a rigorous development of the equations of motion for a variety of spacecraft dynamics problems, beginning with a thorough treatment of the basics of rigid body dynamics. Synge and Griffith[5] is one of the few elementary mechanics texts that treats the elliptic function solution of the torque-free rigid body equations of motion. Goldstein's[6] advanced dynamics book is primarily aimed at the particle physics community; however, he provides an excellent treatment of the basics of rigid body kinematics and kinetics. Wiesel[7] develops rigid body dynamics from the point of view of spacecraft dynamics.

Bibliography

[1] Peter W. Likins. *Elements of Engineering Mechanics.* McGraw-Hill, New York, 1973.

[2] C. Truesdell. *Essays in the History of Mechanics.* Springer-Verlag, New York, 1968.

[3] Leonard Meirovitch. *Methods of Analytical Dynamics.* McGraw-Hill, New York, 1970.

[4] Peter C. Hughes. *Spacecraft Attitude Dynamics.* John Wiley & Sons, New York, 1986.

[5] John L. Synge and Byron A. Griffith. *Principles of Mechanics.* McGraw-Hill, New York, third edition, 1959.

[6] Herbert Goldstein. *Classical Mechanics.* Addison-Wesley, Reading, MA, second edition, 1980.

[7] William E. Wiesel. *Spaceflight Dynamics.* McGraw-Hill, New York, second edition, 1997.

5.6 Exercises

1. Develop the scalar first-order differential equations of motion for the case where $\vec{\mathbf{r}}$ and $\vec{\mathbf{p}}$ are expressed in terms of spherical coordinates.

2. Develop the scalar first-order differential equations of motion for the case where $\vec{\mathbf{r}}$ and $\vec{\mathbf{p}}$ are expressed in terms of cylindrical coordinates.

3. Use the first-order vector differential equations for a system of n point masses to develop two first-order vector differential equations for the system's mass center.

4. Explicitly carry out the integrations required to compute the total mass, first moment, and second moment of inertia matrix for the rectangular prism used in § 5.2.

5. Show that $\vec{\mathbf{b}}_1\vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_2\vec{\mathbf{b}}_2 + \vec{\mathbf{b}}_3\vec{\mathbf{b}}_3$ satisfies the definition of $\vec{\mathbf{1}}$.

6. Prove Eq. (5.57).

7. Develop the equation $\vec{\mathbf{h}}^o = \vec{\mathbf{I}}^o \cdot \omega$.

8. Show that Eq. (5.71) is true.

9. Beginning with the definition of kinetic energy,
$$T = \frac{1}{2}\int_{\mathcal{B}} \vec{\mathbf{r}} \times \vec{\mathbf{v}}\, dm$$
show that
$$T = \frac{1}{2}\vec{\omega} \cdot \vec{\mathbf{I}} \cdot \vec{\omega} = \frac{1}{2}\omega^{\mathsf{T}}\mathbf{I}\omega$$
Be sure to distinguish between vectors and tensors, and their representations in reference frames.

5.7 Problems

1. For the body shown in Fig. 5.2, determine the volume, V, mass, m, mass center position relative to o, and the first, $\mathbf{c}^o$, and second, $\mathbf{I}^o$, moments of inertia about o. Use the parallel axis theorem to determine the second moments of inertia about the mass center. Develop these expressions analytically (*i.e.*, in terms of a, b, c, *etc.*), and then use the following numbers to compute numerical values: $a = 1$ m, $b = 3$ m, $c = 2$ m, $\mu = 10$ kg/m^3, $\rho = 0.4$ m.

2. A spacecraft is comprised of 3 rigid bodies: a *cylinder*, a *rod*, and a *panel*, as shown in Fig. 5.3. The $\mathcal{F}_{b'}$ frame shown is at the center of mass of the cylinder with the $\hat{\mathbf{b}}_3'$ axis parallel to the cylinder's symmetry axis, and the $\hat{\mathbf{b}}_2'$ axis parallel to the rod axis. The cylinder has a diameter of 1 m, a height of 2 m, and a mass density of 100 kg/m^3. The rod has a length of 2 m and a mass of 1 kg. The panel has length 2 m, width 0.5 m, and mass 4 kg. The panel is rotated about the rod axis through an angle $\theta = 87°$; *i.e.*, if $\theta = 0$, then the panel would be in the $\hat{\mathbf{b}}_2'\hat{\mathbf{b}}_3'$ plane, and if $\theta = 90°$, then the panel would be parallel to the $\hat{\mathbf{b}}_1'\hat{\mathbf{b}}_2'$ plane.

   (a) Find the mass center of the system, relative to the $\{\hat{\mathbf{b}}'\}$ frame shown in the sketch. Give your answer as
   $$\vec{\mathbf{r}}^c = x\hat{\mathbf{b}}_1' + y\hat{\mathbf{b}}_2' + z\hat{\mathbf{b}}_3'$$

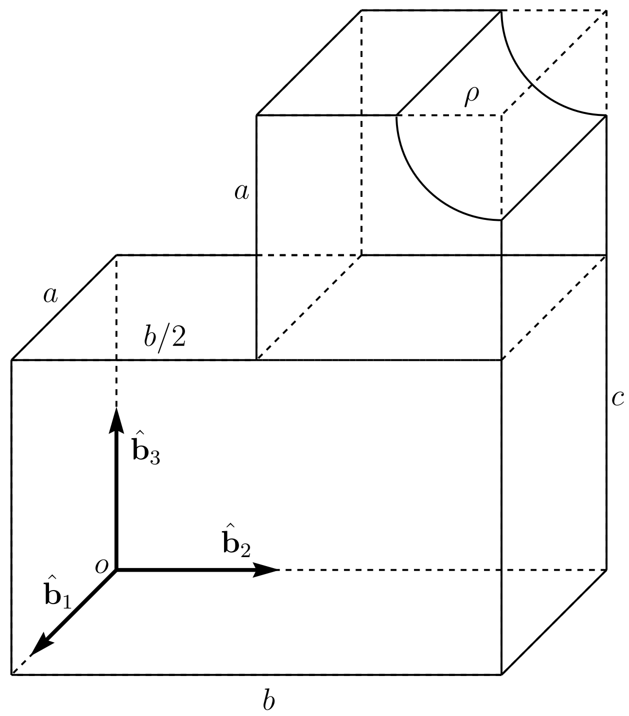

Figure 5.2: Composite rigid body

where x, y, and z are lengths given in meters with at least 5 significant digits.

(b) Find the moment of inertia matrix of the system, about the mass center of the cylinder, and relative to $\mathcal{F}_{b'}$.

(c) Find the moment of inertia matrix of the system about the mass center of the system, and relative to $\mathcal{F}_{b'}$.

(d) Find the principal axes and principal moments of inertia of the system. That is, find the rotation matrix $\mathbf{R}^{bb'}$ that takes vectors from $\mathcal{F}_{b'}$ to $\mathcal{F}_b$. You may find the Matlab function `[V,D] = eig(I)` useful in solving this problem.

(e) Suppose this spacecraft is in an orbit with the TLE given below. (What spacecraft is this really the TLE for?) Further suppose that at epoch, the attitude of $\mathcal{F}_b$ with respect to the orbital frame is given by the identity matrix (*i.e.*, the two reference frames are aligned). Determine the latitude and longitude of the subsatellite point, and of the "targets" pointed to by the $\hat{\mathbf{b}}_3'$ axis and the vector $\hat{\mathbf{n}}_p$ which is normal to the panel. How far apart are these two targets, measured in km as an arc on the surface of the Earth?

```
1 25544U 98067A   99077.49800042  .00020263  00000-0  26284-3 0  4071
```

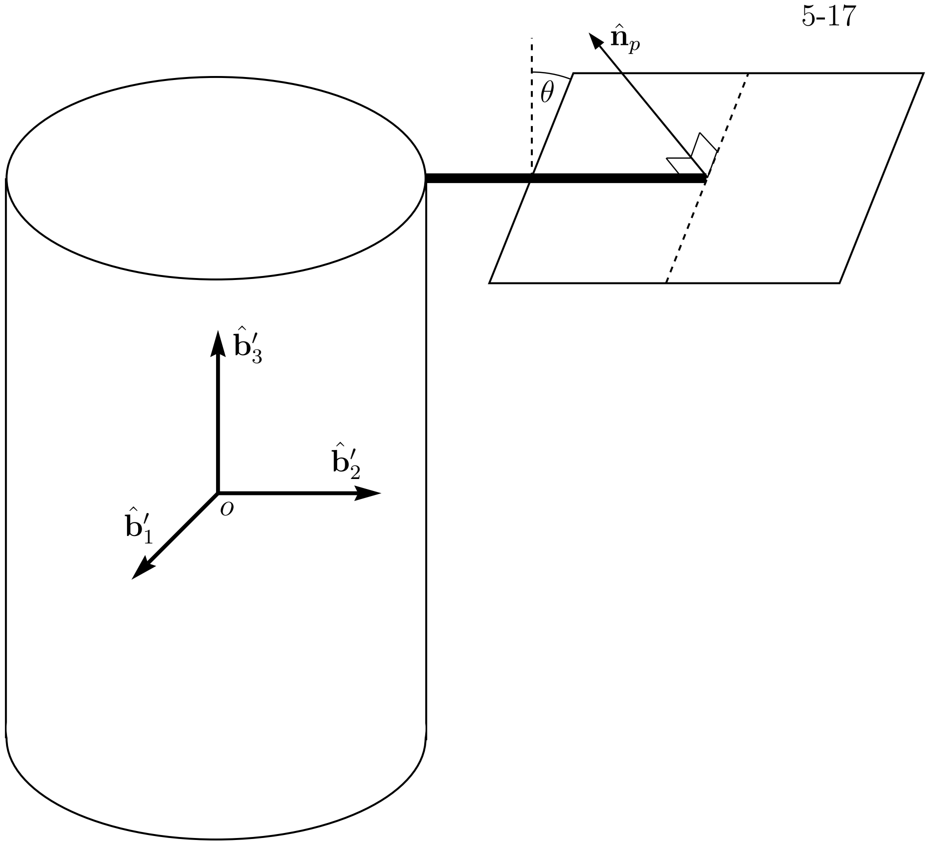

Figure 5.3: Cylinder+rod+panel

```
2 25544  51.5921 294.4693 0004426 235.8661 124.1905 15.58818692 18441
```

Chapter 6

Satellite Attitude Dynamics

Chapter 5 develops the fundamental topics required to study the rotational motion of rigid bodies. These fundamentals are relevant to a wide variety of engineering problems, and especially to the study of the attitude motion of spacecraft. An important step in specializing the rigid body equations of motion for a particular application is to develop a thorough understanding of the torques $\mathbf{g}$ that appear in the equations. In this chapter, we develop the environmental torques that affect spacecraft motion, and investigate some of the standard problems of interest to spacecraft engineers.

6.1 Environmental Torques

Important environmental torques affecting satellite attitude dynamics include gravity gradient, magnetic, aerodynamic, and solar radiation pressure torques. We develop expressions for these torques in this section, and in subsequent sections we study the effects of the torques on attitude motion.

Currently this section only includes the development of the gravity gradient torque.

6.1.1 Gravity Gradient Torque

We assume a rigid spacecraft in orbit about a spherical primary, and every differential mass element of the body is subject to Newton's Universal Gravitational Law:

$$d\vec{\mathbf{f}}_g = -\frac{GM\,dm}{r^2}\hat{\mathbf{e}}_r \tag{6.1}$$

where G is the universal gravitational constant, M is the mass of the spherical primary, dm is the mass of a mass element of the body in orbit, r is the radial distance from the mass center of the primary to the mass element, and $\hat{\mathbf{e}}_r$ is a unit vector from the mass center of the primary to the mass element. Note that $\hat{\mathbf{e}}_r$ for the mass center may be expressed as one of the unit vectors in the orbital frame, $\mathcal{F}_o$, as $\hat{\mathbf{e}}_r = -\hat{\mathbf{o}}_3$.

Furthermore, we find it useful to write the position vector to a differential mass element as the sum of the position vector from the primary to the mass center of the body and the vector from the mass center of the body to the differential mass element:

$$\vec{\mathbf{r}} = {}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}} \tag{6.2}$$

This force acts on every particle in the body, so that we compute the force on the body by integrating this differential force over the body. Thus

$$\vec{\mathbf{f}}_g = -\int_{\mathcal{B}} \frac{GM}{r^2}\hat{\mathbf{e}}_r \, dm \tag{6.3}$$

which can be written as

$$\vec{\mathbf{f}}_g = -\int_{\mathcal{B}} \frac{GM}{r^3}\vec{\mathbf{r}} \, dm \tag{6.4}$$

or, expanding the position vector using Eq. (6.2),

$$\vec{\mathbf{f}}_g = -\int_{\mathcal{B}} \frac{GM}{|{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}|^3}({}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}) \, dm \tag{6.5}$$

In the integrand, the vector ${}_c\vec{\mathbf{r}}$ is the only variable that depends on the differential mass element. However, in general, this integral cannot be computed in closed form. The usual approach is to assume that the radius of the orbit is much greater than the size of the body, *i.e.*, $|{}^c_O\vec{\mathbf{r}}| \gg |{}_c\vec{\mathbf{r}}|$, and expanding the integrand as follows:

$$\begin{aligned}
\frac{{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}}{|{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}|^3} &= \frac{{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}}{[({}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}) \cdot ({}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}})]^{3/2}} && (6.6)\\
&= \frac{{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}}{[{}^c_O\vec{\mathbf{r}} \cdot {}^c_O\vec{\mathbf{r}} + 2{}^c_O\vec{\mathbf{r}} \cdot {}_c\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}} \cdot {}_c\vec{\mathbf{r}}]^{3/2}} && (6.7)\\
&= \frac{{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}}{[{}^c_Or^2 + 2{}^c_O\vec{\mathbf{r}} \cdot {}_c\vec{\mathbf{r}} + {}_cr^2]^{3/2}} && (6.8)\\
&= \frac{{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}}{{}^c_Or^3\,[1 + 2{}^c_O\vec{\mathbf{r}} \cdot {}_c\vec{\mathbf{r}}/{}^c_Or^2 + {}_cr^2/{}^c_Or^2]^{3/2}} && (6.9)
\end{aligned}$$

Now, since $|{}^c_O\vec{\mathbf{r}}| \gg |{}_c\vec{\mathbf{r}}|$, expansion in a Taylor series leads to

$$\begin{aligned}
\frac{{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}}{|{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}|^3} &= \frac{{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}}{{}^c_Or^3}\left(1 - 3\frac{{}^c_O\vec{\mathbf{r}} \cdot {}_c\vec{\mathbf{r}}}{{}^c_Or^2} + H.O.T.\right) && (6.10)\\
&\approx \frac{{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}}{{}^c_Or^3}\left(1 - 3\frac{{}^c_O\vec{\mathbf{r}} \cdot {}_c\vec{\mathbf{r}}}{{}^c_Or^2}\right) && (6.11)\\
&\approx \frac{{}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}}}{{}^c_Or^3} && (6.12)
\end{aligned}$$

Substituting the last of these approximations in to the volume integral for $\vec{\mathbf{f}}_g$ leads to

$$\vec{\mathbf{f}}_g = -\int_{\mathcal{B}} \frac{GM\,({}^c_O\vec{\mathbf{r}} + {}_c\vec{\mathbf{r}})}{{}^c_O r^3}\,dm \tag{6.13}$$

$$= -\frac{GMm}{{}^c_O r^3}\,{}^c_O\vec{\mathbf{r}} \tag{6.14}$$

Finally, letting $\vec{\mathbf{r}}$ denote the position vector from the primary to the mass center of the orbiting body, and applying Newton's Second Law, we obtain:

$$\ddot{\vec{\mathbf{r}}} + \frac{GM}{r^3}\vec{\mathbf{r}} = \vec{\mathbf{0}} \tag{6.15}$$

This equation is recognizable as the familiar two-body vector equation of orbital motion, in which the constant GM is usually written as the gravitational parameter μ.

We also need to develop the moment about the mass center due to the gravitational forces. This moment may be expressed as an integral over the body:

$$\vec{\mathbf{g}}^c_g = -\int_{\mathcal{B}} {}_c\vec{\mathbf{r}} \times d\vec{\mathbf{f}}_g \tag{6.16}$$

Applying the same assumptions as those used to compute the approximate force, we obtain an expression for the approximate moment about the mass center:

$$\vec{\mathbf{g}}^c_g = 3\frac{GM}{r^3}\hat{\mathbf{o}}_3 \times \vec{\mathbf{I}}^c \cdot \hat{\mathbf{o}}_3 \tag{6.17}$$

Expressed in a body-fixed reference frame, this gravity-gradient torque is

$$\mathbf{g}^c_{gb} = 3\frac{GM}{r^3}\mathbf{o}^\times_{3b}\mathbf{I}^c_b\mathbf{o}_{3b} \tag{6.18}$$

Note that $\mathbf{o}_{3b}$ is the third column of the rotation matrix $\mathbf{R}^{bo}$. Usually the facts that the moment and moment of inertia are about the mass center and that the vectors are expressed in $\mathcal{F}_b$ are understood and we can simplify the notation to

$$\mathbf{g}_g = 3\frac{GM}{r^3}\mathbf{o}^\times_3\mathbf{I}\mathbf{o}_3 \tag{6.19}$$

This torque affects the motion of all orbiting bodies, and in the next section we investigate the resulting steady motions and their stability.

6.2 Gravity Gradient Stabilization

For a rigid satellite in a central gravitational field, the equations of motion may be approximated as

$$\ddot{\vec{\mathbf{r}}} + \frac{GM}{r^3}\vec{\mathbf{r}} = \vec{\mathbf{0}} \tag{6.20}$$

$$\dot{\vec{\mathbf{h}}} = 3\frac{GM}{r^3}\hat{\mathbf{o}}_3 \times \vec{\mathbf{I}} \cdot \hat{\mathbf{o}}_3 \tag{6.21}$$

where $\vec{\mathbf{r}}$ is the position vector of the mass center of the body with respect to the center of the gravitational primary, and $\vec{\mathbf{h}}$ is the angular momentum of the body about its mass center. The vector $\hat{\mathbf{o}}_3$ is the nadir vector; *i.e.*, $\hat{\mathbf{o}}_3 = -\vec{\mathbf{r}}/r$.

The first equation is simply the two-body equation of motion for a point mass orbiting a spherical primary, and its solution is well known. The second equation is the vector expression of Euler's equations for the specific case of a gravity gradient torque (the right-hand side). As usual, we want to write this equation in terms of the principal body-frame components of these vectors. Thus, the rotational equations of motion for a rigid body subject only to gravitational forces and moments can be written as

$$\mathbf{I}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}^\times \mathbf{I}\boldsymbol{\omega} + 3\frac{GM}{r^3}\mathbf{o}_3^\times \mathbf{I}\mathbf{o}_3 \tag{6.22}$$

Since r appears in this equation, the solution to the orbital (translational) equations of motion must be used in solving the attitude (rotational) equations of motion. Also, suitable kinematics equations must be used with these equations, since the components of $\mathbf{o}_3$ in $\mathcal{F}_b$ are the elements of the third column of $\mathbf{R}^{bo}$, which may be written as $\mathbf{R}^{bi}\mathbf{R}^{io}$.

Now, let us investigate the possibility of equilibrium attitude motion of a satellite in a circular orbit, in which case r is constant, and the term $3GM/r^3$ is $3\omega_c^2$, where ω_c is the orbital angular velocity, or mean motion. Recall that the orbital frame, $\mathcal{F}_o$ is a frame with $\hat{\mathbf{o}}_1$ in the direction of the velocity vector, $\hat{\mathbf{o}}_2$ in the direction of the negative orbit normal, and $\hat{\mathbf{o}}_3$ in the nadir direction (*i.e.*, Earth-pointing). The angular velocity of this frame with respect to the inertial frame may be expressed in the orbital frame as $\boldsymbol{\omega}_o^{oi} = [0, -\omega_c, 0]^T$. The possibility of an Earth-orbiting satellite with the body frame always aligned with the orbital frame has obvious operational advantages, and we investigate this equilibrium in the following section.

If the two frames, $\mathcal{F}_b$ and $\mathcal{F}_o$, are aligned, then $\mathbf{R}^{bo} = \mathbf{1}$, so that $\mathbf{o}_3 = [0, 0, 1]^T$ in $\mathcal{F}_b$. Furthermore, $\boldsymbol{\omega}^{bo} = \mathbf{0}$, so that $\boldsymbol{\omega}^{bi} = \boldsymbol{\omega}^{oi} = [0, -\omega_c, 0]^T$. If we substitute this angular velocity into the right-hand side of Eq. (6.22), we find that $\dot{\boldsymbol{\omega}} = \mathbf{0}$, so this motion is in fact an equilibrium motion. However, just as the rigid pendulum has two equilibrium motions, straight down ($\theta = 0$) and straight up (($\theta = \pi$), with different stability properties, there is a variety of equilibrium motions for the rigid body. Specifically, there are three different principal axes that could be aligned with the $\hat{\mathbf{o}}_1$ axis, and for each of these there are two remaining principal axes that could be aligned with the $\hat{\mathbf{o}}_2$ axis. Thus there are six different types of equilibrium motions possible (actually there are 12 if you count the $-\hat{\mathbf{b}}_1$ axis as being different from the $+\hat{\mathbf{b}}_1$ axis). The way we think about these possibilities is the same for the spin-stability of an asymmetric body: we investigate small motions of $\mathcal{F}_b$ with respect to $\mathcal{F}_o$, and see how the ordering of the principal moments of inertia I_1, I_2, and I_3 affect the stability properties.

To investigate the stability of small motions in this case, we must include the kinematics. We use a 1-2-3 rotation sequence from $\mathcal{F}_o$ to $\mathcal{F}_b$. That is, $\mathbf{R}^{bo} =$

$\mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1)$. Expanding this expression gives

$$\mathbf{R}^{bo} = \begin{bmatrix} c_2c_3 & s_1s_2c_3 + c_1s_3 & s_1s_3 - c_1s_2c_3 \\ -s_2s_3 & c_1c_3 - s_1s_2s_3 & s_1c_3 + c_1s_2s_3 \\ s_2 & -s_1c_2 & c_1c_2 \end{bmatrix} \tag{6.23}$$

If we make the assumption that all the angles are small, *i.e.*, $\sin\theta_i \approx \theta_i$, $\cos\theta_i \approx 1$, and $\theta_i\theta_j \approx 0$, then this rotation matrix becomes

$$\mathbf{R}^{bo} \approx \mathbf{1} - \boldsymbol{\theta}^{\times} \tag{6.24}$$

Thus $\mathbf{o}_3 = [-\theta_2, \theta_1, 1]^T$ in the body frame. This approximation may be used immediately to compute the gravity gradient torque (for small angles) as

$$\mathbf{g}_{gg} = 3\omega_c^2\mathbf{o}_3^{\times}\mathbf{I}\mathbf{o}_3 = 3\omega_c^2 \begin{bmatrix} (I_3 - I_2)\theta_1 \\ (I_3 - I_1)\theta_2 \\ 0 \end{bmatrix} \tag{6.25}$$

Assuming small angles and small angular rates, the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_o$ is simply $\boldsymbol{\omega}^{bo} = \dot{\boldsymbol{\theta}}$. Note that this result requires developing $\boldsymbol{\omega} = \mathbf{S}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$ and carrying out the small-angle approximations.

The $\boldsymbol{\omega}$ that appears in the equations of motion (6.22) is the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$, which is

$$\boldsymbol{\omega}^{bi} = \boldsymbol{\omega}^{bo} + \boldsymbol{\omega}^{oi} \tag{6.26}$$

and both vectors must be expressed in $\mathcal{F}_b$. Thus

$$\boldsymbol{\omega}^{bi} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 1 & \theta_3 & -\theta_2 \\ -\theta_3 & 1 & \theta_1 \\ \theta_2 & -\theta_1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -\omega_c \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 - \omega_c\theta_3 \\ \dot{\theta}_2 - \omega_c \\ \dot{\theta}_3 + \omega_c\theta_1 \end{bmatrix} \tag{6.27}$$

In Euler's equations, we also need $\dot{\boldsymbol{\omega}}$, which is easily computed as

$$\dot{\boldsymbol{\omega}} = \begin{bmatrix} \ddot{\theta}_1 - \omega_c\dot{\theta}_3 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 + \omega_c\dot{\theta}_1 \end{bmatrix} \tag{6.28}$$

Now, if we substitute Eqs. (6.25), (6.27), and (6.28) into Eq. (6.22), we obtain the following three coupled linear ordinary differential equations:

$$I_1\ddot{\theta}_1 + (I_2 - I_3 - I_1)\omega_c\dot{\theta}_3 - 4(I_3 - I_2)\omega_c^2\theta_1 = 0 \tag{6.29}$$

$$I_2\ddot{\theta}_2 + 3\omega_c^2(I_1 - I_3)\theta_2 = 0 \tag{6.30}$$

$$I_3\ddot{\theta}_3 + (I_3 + I_1 - I_2)\omega_c\dot{\theta}_1 + (I_2 - I_1)\omega_c^2\theta_3 = 0 \tag{6.31}$$

Note that the pitch equation (θ_2) is decoupled from the roll (θ_1) and yaw (θ_3) equations, and is of the form $\ddot{x} + kx = 0$, so that the pitch motion is stable if $I_1 > I_3$. That is, the Earth-pointing axis, $\hat{\mathbf{b}}_3$, cannot be the major axis. If the other two equations of motion are unstable, then the pitch motion does not necessarily remain small, since the nonlinear terms are no longer negligible.

To investigate the stability of the coupled roll-yaw motion, we write the two equations (6.29,6.31) in the matrix second-order form:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{G}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{0} \tag{6.32}$$

where $\mathbf{x} = [\theta_1 \;\; \theta_3]^T$ and the three 2×2 matrices are defined as

$$\mathbf{M} = \begin{bmatrix} I_1 & 0 \\ 0 & I_3 \end{bmatrix}, \tag{6.33}$$

$$\mathbf{G} = (I_1 + I_3 - I_2)\omega_c \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \tag{6.34}$$

$$\mathbf{K} = \omega_c^2 \begin{bmatrix} 4(I_2 - I_3) & 0 \\ 0 & (I_2 - I_1) \end{bmatrix} \tag{6.35}$$

These three matrices are normally called the *mass matrix*, the *gyroscopic damping matrix*, and the *stiffness matrix*, respectively.

Just as we did for the spin stability of a rigid body, we assume an exponential form for the solution to this system of differential equations. Specifically, we seek solutions of the form:

$$\mathbf{x} = e^{\lambda t}\mathbf{c} \tag{6.36}$$

where λ is an eigenvalue, and $\mathbf{c}$ is a 2×1 matrix containing the arbitrary constants that must be determined based on the initial conditions. We need the two derivatives of this form of the solution: $\dot{\mathbf{x}} = \lambda e^{\lambda t}\mathbf{c} = \lambda\mathbf{x}$, and $\ddot{\mathbf{x}} = \lambda^2 e^{\lambda t}\mathbf{c} = \lambda^2\mathbf{x}$. Substituting these into the differential equation and collecting terms, we obtain

$$e^{\lambda t}\left[\lambda^2\mathbf{M} + \lambda\mathbf{G} + \mathbf{K}\right]\mathbf{c}_o = \mathbf{0} \tag{6.37}$$

Since $e^{\lambda t} \neq 0$, we can divide both sides by $e^{\lambda t}$, leading to the matrix equation

$$\left[\lambda^2\mathbf{M} + \lambda\mathbf{G} + \mathbf{K}\right]\mathbf{c}_o = \mathbf{0} \tag{6.38}$$

which is of the form $\mathbf{A}\mathbf{c}_o = \mathbf{0}$. A well-known result of linear algebra states that this equation can only have non-trivial solutions ($\mathbf{c}_o \neq \mathbf{0}$) if the matrix $\mathbf{A}$ is singular. Another well-known result of linear algebra is that a matrix is singular if and only if its determinant is zero. Thus, we can determine the eigenvalues λ by setting the determinant of the matrix equal to zero, and solving for λ:

$$\begin{vmatrix} \lambda^2 I_1 + 4\omega_c^2(I_2 - I_3) & -\lambda\omega_c(I_1 + I_3 - I_2) \\ \lambda\omega_c(I_1 + I_3 - I_2) & \lambda^2 I_3 + \omega_c^2(I_2 - I_1) \end{vmatrix} = 0 \tag{6.39}$$

which expands to give the characteristic polynomial

$$\lambda^4 I_1 I_3 + \lambda^2 \omega_c^2 [I_1(I_2 - I_1) + 4I_3(I_2 - I_3) + (I_1 + I_3 - I_2)^2] + 4\omega_c^4 (I_2 - I_1)(I_2 - I_3) = 0 \quad (6.40)$$

Dividing by $I_1 I_3 \omega_c^4$, and defining the two inertia parameters $k_1 = (I_2 - I_3)/I_1$, $k_3 = (I_2 - I_1)/I_3$, the characteristic equation may be more conveniently written as

$$\left(\frac{\lambda}{\omega_c}\right)^4 + (1 + 3k_1 + k_1 k_3)\left(\frac{\lambda}{\omega_c}\right)^2 + 4k_1 k_3 = 0 \quad (6.41)$$

This equation is a quadratic polynomial in the variable $s = (\lambda/\omega_c)^2$, which can be written as

$$s^2 + b_1 s + b_0 = 0 \quad (6.42)$$

In order for the motion to be stable (oscillatory), the eigenvalues (λ's) must be pure imaginary, so s must be negative. The necessary conditions for s to be real and negative are

$$b_0 > 0 \qquad b_1 > 0 \qquad b_1^2 - 4b_2 > 0 \quad (6.43)$$

Recall that for pitch stability, we need $I_1 > I_3$, which is equivalent to $k_1 > k_3$. The four stability conditions may then be expressed as

$$\begin{aligned} k_1 &> k_3 & (6.44)\\ k_1 k_3 &> 0 & (6.45)\\ 1 + 3k_1 + k_1 k_3 &> 0 & (6.46)\\ (1 + 3k_1 + k_1 k_3)^2 - 16 k_1 k_3 &> 0 & (6.47) \end{aligned}$$

We can show that the parameters k_1 and k_2 (called Smelt parameters[1]) are bounded between ± 1. Thus we can construct a stability diagram in the $k_1 k_3$ plane, as shown in Figure 6.1.

In the stability diagram shown in Fig. 6.1, the curves indicate the stability boundaries. The curves labeled I, III, and IV, correspond to the first, third, and fourth of the conditions listed in Eqs. (6.44–6.47). The second condition rules out the second and fourth quadrants of the plane. Thus, the only regions corresponding to stable configurations are the two labeled "Lagrange" and "D^2." The symbol D^2 refers to the "DeBra-Delp" region first reported in Ref. 1. The Lagrange region was established by Lagrange* in his studies of the equilibrium motions of the moon.

*Joseph Louis Lagrange (1736 - 1813) was a major contributor to the field of mechanics, developing the calculus of variations and a new way of writing the equations of motion, now known as Lagrange's equations.

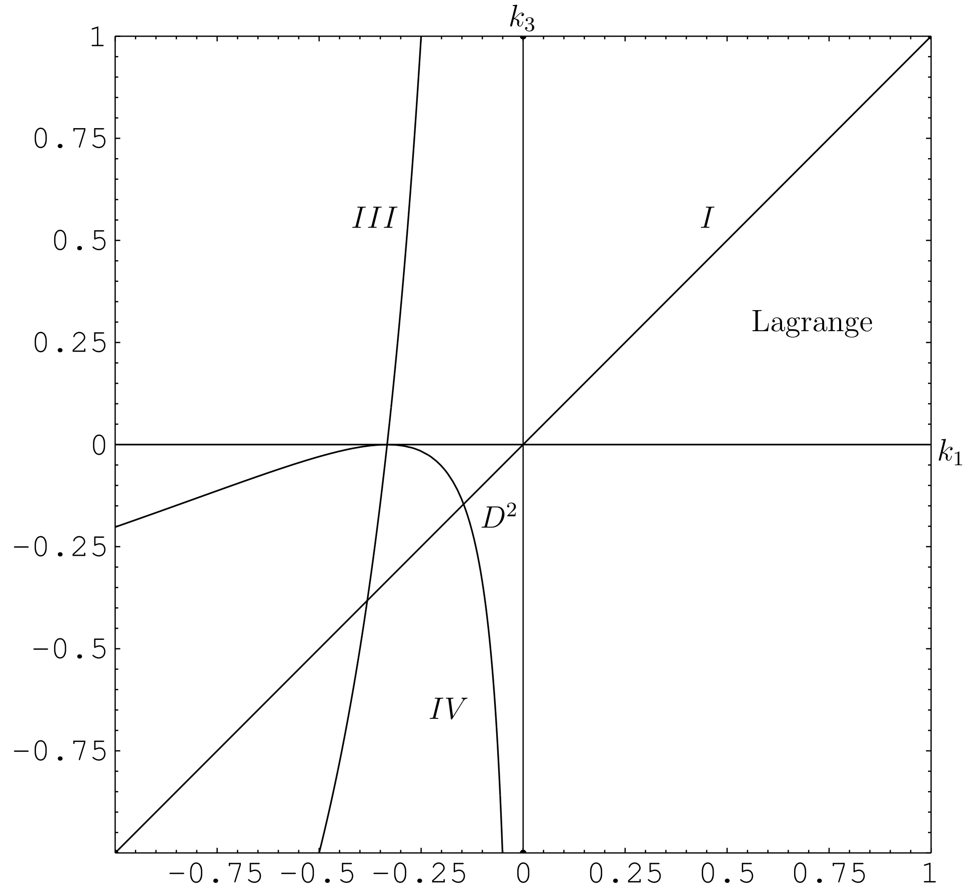

Figure 6.1: The Smelt Parameter Plane

6.3 Spin Stabilization

6.4 Dual-Spin Stabilization

In Chapter 5, we develop the well-known result that a rigid body is stable in spin about either the major or minor axis, but is unstable in spin about the intermediate axis. We also found that in the presence of energy dissipation the major axis spin becomes asymptotically stable whereas the minor axis spin becomes unstable. This latter result is the reason for spin-stabilized spacecraft being shaped more like tuna cans than asparagus cans. In this section, we develop two important results regarding dual-spin stabilization.

Dual-spin stabilization applies to spacecraft with two components that are spinning relative to each other. Typically one body is spinning relatively fast and the other is spinning relatively slow. The first result involves only rigid bodies, and concludes that a spinning wheel can be used to stabilize spin about *any* axis. The second result involves an energy sink analysis and concludes that a minor axis spin is stable if the energy dissipation on the rapidly spinning component is "smaller" than the energy dissipation on the slowly spinning component.

We begin with the equations of motion for a rigid body with an embedded axisymmetric flywheel. Note that the flywheel's motion does not affect the moment of inertia matrix of the system. That is, if we lock the flywheel relative to the body and compute the moment of inertia matrix and principal axes of the system, these moments of inertia are unchanged by allowing the flywheel to rotate about its symmetry axis. We use a body-fixed, principal reference frame, $\mathcal{F}_b$, which has angular velocity relative to inertial space denoted $\vec{\boldsymbol{\omega}}$, with body frame components $\boldsymbol{\omega}$. The flywheel's spin axis is assumed (without loss of generality) to be aligned parallel to the spacecraft's $\hat{\mathbf{b}}_3$ axis. The flywheel has spin axis moment of inertia denoted by I_w. The flywheel's angular velocity relative to the body frame may be written as $\vec{\boldsymbol{\omega}}^{wb} = \Omega_w \hat{\mathbf{b}}_3$. Expressed in the body frame, this vector has components $\boldsymbol{\omega}^{wb} = \Omega_w [0, 0, 1]^T$. Note that this angular velocity is the relative velocity as would be measured by a tachometer. The inertial angular velocity of the flywheel is $\vec{\boldsymbol{\omega}}^{wi} = \vec{\boldsymbol{\omega}} + \vec{\boldsymbol{\omega}}^{wb}$. In the following development we assume that Ω_w is constant.

The total angular momentum of the system is simply the angular momentum due to the rigid body motion, plus the axial angular momentum contributed by the flywheel. In the body frame, the angular momentum has components:

$$\mathbf{h} = \mathbf{I}\boldsymbol{\omega} + I_w \boldsymbol{\omega}^{wb} \tag{6.48}$$

Now we can apply Euler's law of moment of momentum, $\dot{\vec{\mathbf{h}}} = \vec{\mathbf{g}}$.Since $\vec{\mathbf{h}}$ is expressed in a rotating reference frame, the resulting differential equation is

$$\dot{\mathbf{h}} + \boldsymbol{\omega}^{\times}\mathbf{h} = \mathbf{g} \tag{6.49}$$

Now substituting the expression for $\mathbf{h}$ into this equation and collecting terms leads to the following coupled system of equations:

$$\dot{\omega}_1 = \frac{I_2 - I_3}{I_1}\omega_2\omega_3 - \frac{I_w}{I_1}\Omega_w\omega_2 + \frac{g_1}{I_1} \tag{6.50}$$

$$\dot{\omega}_2 = \frac{I_3 - I_1}{I_2}\omega_1\omega_3 + \frac{I_w}{I_2}\Omega_w\omega_1\frac{g_2}{I_2} \tag{6.51}$$

$$\dot{\omega}_3 = \frac{I_1 - I_2}{I_3}\omega_1\omega_2\frac{g_3}{I_3} \tag{6.52}$$

We restrict attention here to the case of torque-free motion, $\mathbf{g} = \mathbf{0}$. If the body frame has angular velocity $\boldsymbol{\omega} = [0, 0, \Omega]^T$, then $\dot{\boldsymbol{\omega}} = \mathbf{0}$. That is, steady spin about the $\hat{\mathbf{b}}_3$ axis is an equilibrium motion. If the wheel's relative angular velocity is zero, *i.e.*, $\Omega_w = 0$, then the rigid body result holds: the spin is stable if the $\hat{\mathbf{b}}_3$ axis is the major or minor axis, unstable if it is the intermediate axis. We now want to determine how the wheel affects this stability result.

To determine stability of this steady motion, we linearize the equations of motion, by letting $\boldsymbol{\omega} = \boldsymbol{\omega}_e + \delta\boldsymbol{\omega}$, where $\boldsymbol{\omega}_e$ is the equilibrium motion of interest. On substituting this expression into the equations of motion, and noting that $\dot{\boldsymbol{\omega}}_e = \mathbf{0}$, we obtain

$$\dot{\delta\omega_1} = \frac{I_2 - I_3}{I_1}\Omega\delta\omega_2 - \frac{I_w}{I_1}\Omega_w\delta\omega_2 \tag{6.53}$$

$$\dot{\delta\omega_2} = \frac{I_3 - I_1}{I_2}\Omega\delta\omega_1 + \frac{I_w}{I_2}\Omega_w\delta\omega_1 \tag{6.54}$$

$$\dot{\delta\omega_3} = 0 \tag{6.55}$$

Notice that there are no $\delta\omega_3$'s in these equations, and that $\dot{\delta\omega_3} = 0$, so it is constant.

The $\dot{\delta\omega_1}$ and $\dot{\delta\omega_2}$ equations can be rewritten as

$$\dot{\delta\omega_1} = \left(\frac{I_2 - I_3}{I_1}\Omega - \frac{I_w}{I_1}\Omega_w\right)\delta\omega_2 \tag{6.56}$$

$$\dot{\delta\omega_2} = \left(\frac{I_3 - I_1}{I_2}\Omega + \frac{I_w}{I_2}\Omega_w\right)\delta\omega_1 \tag{6.57}$$

If we differentiate the $\dot{\delta\omega_1}$ equation and substitute the $\dot{\delta\omega_2}$ equation into the result, we obtain the following second-order, constant-coefficient, linear ordinary differential equation:

$$\ddot{\delta\omega_1} + \left(\frac{I_3 - I_2}{I_1}\Omega + \frac{I_w}{I_1}\Omega_w\right)\left(\frac{I_3 - I_1}{I_2}\Omega + \frac{I_w}{I_2}\Omega_w\right)\delta\omega_1 \tag{6.58}$$

This equation is of the form $\ddot{x} + kx = 0$, and as we know, solutions are stable and oscillatory (sines and cosines of $\sqrt{k}t$) if $k > 0$, and are unstable (exponentials of $\pm\sqrt{-k}t$) if $k < 0$. Therefore, to determine stability, we just need to know whether the term

$$k = \left(\frac{I_3 - I_2}{I_1}\Omega - \frac{I_w}{I_1}\Omega_w\right)\left(\frac{I_3 - I_1}{I_2}\Omega + \frac{I_w}{I_2}\Omega_w\right) \tag{6.59}$$

is positive or negative. To analyze this term, we factor out the positive quantities, $1/I_1$, $1/I_2$, and Ω, so that

$$k = \frac{\Omega^2}{I_1 I_2}\left(I_3 - I_2 + I_w\hat{\Omega}_w\right)\left(I_3 - I_1 + I_w\hat{\Omega}_w\right) \tag{6.60}$$

where $\hat{\Omega}_w = \Omega_w/\Omega$. For the case where $\hat{\Omega}_w = 0$, the condition reduces to the familiar asymmetric rigid body conditions for stability: $I_3 > I_1$ and $I_3 > I_2$, or $I_3 < I_1$ and $I_3 < I_2$. However, for the case where the wheel is spinning relative to the body, we get the conditions

$$I_3 > I_2 - I_w\hat{\Omega}_w \quad \text{and} \quad I_3 > I_1 - I_w\hat{\Omega}_w \tag{6.61}$$

or

$$I_3 < I_2 - I_w\hat{\Omega}_w \quad \text{and} \quad I_3 < I_1 - I_w\hat{\Omega}_w \tag{6.62}$$

Thus, the $\hat{\mathbf{b}}_3$ axis does not have to be the major axis. For example, if $\hat{\Omega}_w$ is "large enough," then the expressions $I_1 - I_w\hat{\Omega}_w$ and $I_2 - I_w\hat{\Omega}_w$ can both be made negative, in which case the first set of conditions is satisfied. Likewise, if the wheel is spinning in the opposite direction as the body, then $\hat{\Omega}_w < 0$, and for sufficiently high spin rate, the two expressions $I_1 - I_w\hat{\Omega}_w$ and $I_2 - I_w\hat{\Omega}_w$ are both positive and can be made greater than I_3, so that the second set of conditions are satisfied.

Example 6.1 *Consider a rigid body with flywheel, with $I_1 = 300$, $I_2 = 400$, $I_3 = 350$, $I_w = 10$ (all in kg m^2), which we would like to have spinning about $\hat{\mathbf{b}}_3$ at 2π rad/s (60 RPM). Clearly $\hat{\mathbf{b}}_3$ is the intermediate axis, so this motion is unstable with the wheel locked. How fast does the wheel have to be spinning in order to stabilize this motion?*

In the first of the two sets of stability conditions, since $I_2 > I_1$, if the first condition is satisfied, the second condition is satisfied as well. Thus we need to have

$$I_3 > I_2 - I_w\hat{\Omega}_w \tag{6.63}$$

which we can solve for Ω_w to obtain

$$\Omega_w > \frac{I_2 - I_3}{I_w}\Omega \tag{6.64}$$

Substituting the given numbers into this expression gives

$$\Omega_w = 10\pi \ rad/s = 300 \ RPM \tag{6.65}$$

Thus if we spin the wheel faster than 300 RPM, the intermediate axis spin is stable.

One thing to be careful of here is that for $\mathbf{g} = \mathbf{0}$, the total angular momentum magnitude is constant. Say we decide to spin the wheel at 400 RPM. Then the total angular momentum of the operating point is $I_3\Omega + I_w\Omega_w = 2618$ kg m^2/ s. Thus before we spin up the wheel, the initial total angular momentum must have this same magnitude.

6.5 Summary

This chapter presents the basics of satellite attitude dynamics and stability analysis.

6.6 References and further reading

Thomson,[2] Kaplan,[3] Hughes,[4] Rimrott,[5] Chobotov,[6] Wiesel,[7] Sidi[8]

Bibliography

[1] D. B. DeBra and R. H. Delp. Rigid body attitude stability and natural frequencies in a circular orbit. *Journal of the Astronautical Sciences*, 8:14–17, 1961.

[2] W. T. Thomson. *Introduction to Space Dynamics.* Dover, New York, 1986.

[3] Marshall H. Kaplan. *Modern Spacecraft Dynamics & Control.* John Wiley & Sons, New York, 1976.

[4] Peter C. Hughes. *Spacecraft Attitude Dynamics.* John Wiley & Sons, New York, 1986.

[5] F. P. J. Rimrott. *Introductory Attitude Dynamics.* Springer-Verlag, New York, 1989.

[6] V. A. Chobotov. *Spacecraft Attitude Dynamics and Control.* Krieger Publishing Co., Malabar, FL, 1991.

[7] William E. Wiesel. *Spaceflight Dynamics.* McGraw-Hill, New York, second edition, 1997.

[8] Marcel J. Sidi. *Spacecraft Dynamics and Control: A Practical Engineering Approach.* Cambridge University Press, Cambridge, 1997.

6.7 Exercises

6.8 Problems

Appendix A

Orbits

As discussed in the Introduction, a good first approximation for satellite motion is obtained by assuming the spacecraft is a point mass or spherical body moving in the gravitational field of a spherical planet. This leads to the classical *two-body problem*. Since we use the term *body* to refer to a spacecraft of finite size (as in *rigid body*), it may be more appropriate to call this the *two-particle* problem, but I will use the term two-body problem in its classical sense.

The basic elements of orbital dynamics are captured in Kepler's three laws which he published in the 17th century. His laws were for the orbital motion of the planets about the Sun, but are also applicable to the motion of satellites about planets. The three laws are:

1. The orbit of each planet is an ellipse with the Sun at one focus.

2. The line joining the planet to the Sun sweeps out equal areas in equal times.

3. The square of the period of a planet is proportional to the cube of its mean distance to the sun.

The first law applies to most spacecraft, but it is also possible for spacecraft to travel in parabolic and hyperbolic orbits, in which case the period is infinite and the 3rd law does not apply. However, the 2nd law applies to all two-body motion. Newton's 2nd law and his law of universal gravitation provide the tools for generalizing Kepler's laws to non-elliptical orbits, as well as for proving Kepler's laws.

A.1 Equations of Motion and Their Solution

In the two-body problem, one assumes a system of only two bodies that are both spherically symmetric or are both point masses. In this case the equations of motion may be developed as

$$\ddot{\vec{\mathbf{r}}} + \frac{\mu}{r^3}\vec{\mathbf{r}} = \vec{\mathbf{0}} \tag{A.1}$$

where $\mu = G(m_1 + m_2)$ and $\vec{\mathbf{r}}$ is the relative position vector as shown in Fig. A.1. The parameter G is Newton's universal gravitational constant, which has the value $G = 6.67259 \times 10^{-11}\text{m}^3\text{kg}^{-1}\text{s}^{-2}$. The scalar r is the magnitude of $\vec{\mathbf{r}}$, *i.e.*, $r = \|\vec{\mathbf{r}}\|$. Since normally $m_1 \gg m_2$, $\mu \approx Gm_1$, and this is the value usually given in astrodynamics tables.

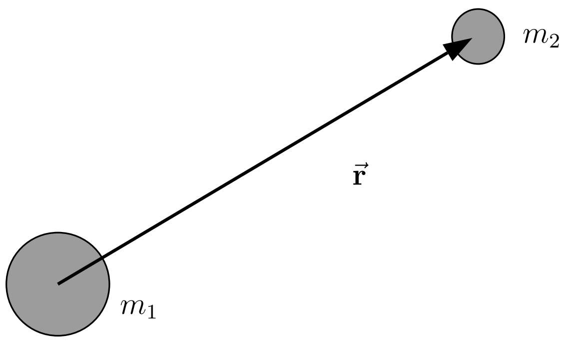

Figure A.1: Relative position vector in the two-body problem.

There are several ways to derive the conserved quantities associated with two-body motion. These constants of the motion are also called first integrals or conservation laws. Here I simply state these conservation laws and interpret them physically and graphically. First is conservation of energy,

$$\mathcal{E} = \frac{1}{2}v^2 - \frac{\mu}{r} = \text{constant} \tag{A.2}$$

where $v = \|\vec{\mathbf{v}}\| = \|\dot{\vec{\mathbf{r}}}\|$. Note that $\mathcal{E}$ is the energy per unit mass: $v^2/2$ is the kinetic energy and $-\mu/r$ is the potential energy. As I discuss more below, energy is closely related to the size of the orbit.

The second constant of motion is the angular momentum vector

$$\vec{\mathbf{h}} = \vec{\mathbf{r}} \times \vec{\mathbf{v}} \tag{A.3}$$

Again, this is the angular momentum per unit mass. The fact that $\vec{\mathbf{h}}$ is constant in direction means that $\vec{\mathbf{r}}$ and $\vec{\mathbf{v}}$ always lie in the plane $\perp$ to $\vec{\mathbf{h}}$. This plane is called the *orbital plane* and it is fixed in inertial space in the two-body problem. It is relatively straightforward to show that conservation of angular momentum is equivalent to

Kepler's second law, and from it can be derived the mathematical version of Kepler's third law for elliptical orbits:

$$\mathcal{P} = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{A.4}$$

where $\mathcal{P}$ is the orbital period and a is the semimajor axis of the ellipse.

The third constant is also a vector quantity, sometimes called the *eccentricity vector*

$$\vec{\mathbf{e}} = \frac{1}{\mu}\left[\left(v^2 - \frac{\mu}{r}\right)\vec{\mathbf{r}} - (\vec{\mathbf{r}} \cdot \vec{\mathbf{v}})\,\vec{\mathbf{v}}\right] \tag{A.5}$$

Note that $\vec{\mathbf{e}}$ is in the orbital plane, since it is a linear combination of $\vec{\mathbf{r}}$ and $\vec{\mathbf{v}}$. In fact, $\vec{\mathbf{e}}$ points in the direction of periapsis, the closest point of the orbit to the central body. Also, $e = \|\vec{\mathbf{e}}\|$ is the eccentricity of the orbit.

Using these constants, it is possible to develop a solution to Eq. (A.1) in the form

$$r = \frac{p}{1 + e\cos\nu} \tag{A.6}$$

where ν is an angle called the *true anomaly*, and $p = h^2/\mu$ is the *semilatus rectum* of the conic section. Equation (A.6) is the mathematical expression of Kepler's second law. This relation describes how the radius changes as the satellite moves but it does not tell how ν varies with time. To determine $\nu(t)$, we must solve Kepler's equation

$$M = n(t - t_0) = E - e\sin E \tag{A.7}$$

where E is the *eccentric anomaly*, M is the *mean anomaly*, $n = \sqrt{\mu/a^3}$ is the *mean motion*, and t_0 is the time of periapsis passage.

A satellite's orbit is normally described in terms of its *orbital elements*. The "classical" orbital elements are semimajor axis, a, eccentricity, e, inclination, i, right ascension of the ascending node, Ω, argument of periapsis, ω, and true anomaly of epoch, ν_0. The last of these six orbital elements is sometimes replaced by mean anomaly at epoch, M_0, or by the time of periapsis passage, T_p.

A.2 Orbit Determination and Prediction

We often need to be able to determine the orbit of a spacecraft or other space object. The primary motivation for orbit determination is to be able to predict where the object will be at some later time. This problem was posed and solved in 1801 by Carl Friedrich Gauss in order to predict the reappearance of the newly discovered minor planet, Ceres. He also invented the method of least squares which is described below.

In practice, orbit determination is accomplished by processing a sequence of observations in an orbit determination algorithm. Since it takes six numbers (elements) to describe an orbit, the algorithm is deterministic if and only if there are six independent numbers in the observation data. If there are fewer than six numbers, then the problem is underdetermined and there is an infinite number of solutions. If there are more than six numbers in the observations, then the problem is overdetermined, and in general no solution exists. In the overdetermined case, a statistical method such as least squares is used to find an orbit that is as close as possible to fitting all the observations. In this section, we first present some deterministic orbit determination algorithms, then a statistical algorithm, and finally we discuss the orbit prediction problem.

A.2.1 Solving Kepler's Equation

```
%  function E = EofMe(M,e,tol)
%
%  This function computes E as a function of M and e.
function E = EofMe(M,e,tol)
   if ( nargin<3 ), tol=1e-11; end
    En  = M;
    En1 = En - (En-e*sin(En)-M)/(1-e*cos(En));
    while ( abs(En1-En) > tol )
        En = En1;
        En1 = En - (En-e*sin(En)-M)/(1-e*cos(En));
    end;
    E = En1;
```

A.2.2 Deterministic Orbit Determination

In the best of all possible worlds, we might be able to measure $\vec{\mathbf{r}}$ and $\vec{\mathbf{v}}$ directly, in which case we would have a set of obs comprising six numbers, three in each vector. With this set of observations, it is straightforward to develop an algorithm that computes the orbital elements as a function of $\vec{\mathbf{r}}$ and $\vec{\mathbf{v}}$. This algorithm is developed in numerous orbit dynamics texts, and here we simply present the mathematical algorithm and a MatLab function that implements the algorithm.

```
%
%  rv2oe.m  r,v,mu -> orbital elements
%
%  oe = rv2oe(rv,vv,mu)
%       rv  = [X Y Z]'    vv = [v1 v2 v3]'   in IJK frame
%       oe = [a e i Omega omega nu0]
%
function oe = rv2oe(rv,vv,mu)
%
% first do some preliminary calculations
%
    K  = [0;0;1];
```

```
    hv = cross(rv,vv);
    nv = cross(K,hv);
    n  = sqrt(nv'*nv);
    h2 = (hv'*hv);
    v2 = (vv'*vv);
    r  = sqrt(rv'*rv);
    ev = 1/mu *( (v2-mu/r)*rv - (rv'*vv)*vv );
    p  = h2/mu;
%
% now compute the oe's
%
    e  = sqrt(ev'*ev);      % eccentricity
    a  = p/(1-e*e);         % semimajor axis
    i  = acos(hv(3)/sqrt(h2));  % inclination
    Om = acos(nv(1)/n);     % RAAN
    if ( nv(2) < 0-eps )        % fix quadrant
        Om = 2*pi-Om
    end;
    om = acos(nv'*ev/n/e);      % arg of periapsis
    if ( ev(3) < 0 )        % fix quadrant
        om = 2*pi-om;
    end;
    nu = acos(ev'*rv/e/r);      % true anomaly
    if ( rv'*vv < 0 )       % fix quadrant
        nu = 2*pi-nu;
    end;
    oe = [a e i Om om nu];      % assemble "vector"
```

A.3 Two-Line Element Sets

Once an orbit has been determined for a particular space object, we might want to communicate that information to another person who is interested in tracking the object. The standard format for communicating the orbit of an Earth-orbiting object is the two-line element set, or TLE. An example TLE for a Russian spy satellite is

```
COSMOS 2278 1 23087U 94023A
98011.59348139  .00000348  00000-0  21464-3 0  5260 2 23087
71.0176  58.4285 0007185 172.8790 187.2435 14.12274429191907
```

Note that the TLE actually has three lines.

Documentation for the TLE format is available at the website `http://celestrak.com` and is reproduced here for your convenience. This website also includes current TLEs for many satellites.

```
Data for each satellite consists of three lines in the following format:
```

```
AAAAAAAAAAAAAAAAAAAAAAAA
1 NNNNNU NNNNNAAA NNNNN.NNNNNNNN +.NNNNNNNN +NNNNN-N +NNNNN-N N NNNNN
2 NNNNN NNN.NNNN NNN.NNNN NNNNNNN NNN.NNNN NNN.NNNN NN.NNNNNNNNNNNNN

Line 0 is a twenty-four character name (to be consistent with the
name length in the NORAD SATCAT).

Lines 1 and 2 are the standard Two-Line Orbital Element Set Format
identical to that used by NORAD and NASA. The format description
is:
Line 1
Column   Description
01       Line Number of Element Data
03-07    Satellite Number
08       Classification (U=Unclassified)
10-11    International Designator (Last two digits of launch year)
12-14    International Designator (Launch number of the year)
15-17    International Designator (Piece of the launch)
19-20    Epoch Year (Last two digits of year)
21-32    Epoch (Day of the year and fractional portion of the day)
34-43    First Time Derivative of the Mean Motion
45-52    Second Time Derivative of Mean Motion (decimal point assumed)
54-61    BSTAR drag term (decimal point assumed)
63       Ephemeris type
65-68    Element number
69       Checksum (Modulo 10)
         (Letters, blanks, periods, plus signs = 0; minus signs = 1)

Line 2
Column   Description
01       Line Number of Element Data
03-07    Satellite Number
09-16    Inclination [Degrees]
18-25    Right Ascension of the Ascending Node [Degrees]
27-33    Eccentricity (decimal point assumed)
35-42    Argument of Perigee [Degrees]
44-51    Mean Anomaly [Degrees]
53-63    Mean Motion [Revs per day]
64-68    Revolution number at epoch [Revs]
69       Checksum (Modulo 10)


All other columns are blank or fixed.

Example:
```

```
NOAA 14
1 23455U 94089A   97320.90946019  .00000140  00000-0  10191-3 0  2621
2 23455  99.0090 272.6745 0008546 223.1686 136.8816 14.11711747148495

For further information, see "Frequently Asked Questions: Two-Line
Element Set Format" in the Computers & Satellites column of
Satellite Times, Volume 4 Number 3.
```

A Matlab function to compute the orbital elements given a TLE is provided below.

```
% function [oe,epoch,yr,M,E,satname] = TLE2oe(fname);
%       fname is a filename string for a file containing
%             a two-line element set (TLE)
%       oe is a 1/6 matrix containing the orbital elements
%             [a e i Om om nu]
%       yr is the two-digit year
%       M  is the mean anomaly at epoch
%       E  is the eccentric anomaly at epoch
%       satname is the satellite name
%
% Calls Newton iteration function file EofMe.m

function [oe,epoch,yr,M,E,satname] = TLE2oe(fname);

% Open the file up and scan in the elements

fid = fopen(fname, 'rb');
A   = fscanf(fid,'%13c%*s',1);
B   = fscanf(fid,'%d%6d%*c%5d%*3c%2d%f%f%5d%*c%*d%5d%*c%*d%d%5d',[1,10]);
C   = fscanf(fid,'%d%6d%f%f%f%f%f%f',[1,8]);
fclose(fid);
satname=A;

% The value of mu is for the earth
mu = 3.986004415e5;

%  Calculate 2-digit year (Oh no!, look out for Y2K bug!)

yr = B(1,4);

% Calculate epoch in julian days
epoch = B(1,5);
%ndot = B(1,6);
% n2dot = B(1,7);
```

```
% Assign variables to the orbital elements
i = C(1,3)*pi/180;          % inclination
Om = C(1,4)*pi/180;         % Right Ascension of the Ascending Node
e = C(1,5)/1e7;             % Eccentricity
om = C(1,6)*pi/180;         % Argument of periapsis
M = C(1,7)*pi/180;          % Mean anomaly
n = C(1,8)*2*pi/(24*3600);  % Mean motion

% Calculate the semi-major axis
a = (mu/n^2)^(1/3);

% Calculate the eccentric anomaly using mean anomaly
E = EofMe(M,e,1e-10);

% Calculate true anomaly from eccentric anomaly
cosnu = (e-cos(E)) / (e*cos(E)-1);
sinnu = ((a*sqrt(1-e*e)) / (a*(1-e*cos(E))))*sin(E);
nu = atan2(sinnu,cosnu);
if (nu<0), nu=nu+2*pi; end

% Return the orbital elements in a 1x6 matrix
oe = [a e i Om om nu];
```

A.4 Summary

This appendix is intended to give a brief overview of orbital mechanics so that the problems involving two-line element sets can be completed. Here is a summary of some basic formulas and constants.

Two-body problem relationships

$$
\begin{aligned}
r &= a(1-e^2)/(1+e\cos\nu) & p &= a(1-e^2) = h^2/\mu \\
\vec{\mathbf{h}} &= \vec{\mathbf{r}}\times\vec{\mathbf{v}} & h &= rv\cos\phi \\
\mathrm{TP} &= 2\pi\sqrt{a^3/\mu} & \mathcal{E} &= v^2/2-\mu/r \\
v_c &= \sqrt{\mu/r} & v_{\mathrm{esc}} &= \sqrt{2\mu/r} \\
\vec{\mathbf{e}} &= \tfrac{1}{\mu}\left[(v^2-\mu/r)\,\vec{\mathbf{r}}-(\vec{\mathbf{r}}\cdot\vec{\mathbf{v}})\vec{\mathbf{v}}\right] & e &= (r_a-r_p)/(r_a+r_p) \\
\vec{\mathbf{n}} &= \hat{\mathbf{K}}\times\vec{\mathbf{h}}
\end{aligned}
$$

Orbital elements

a	semimajor axis	b	$=$	$\sqrt{ap}$	semiminor axis
e	eccentricity				
i	inclination	$h\cos i$	$=$	$\vec{\mathbf{h}}\cdot\hat{\mathbf{K}}$	
Ω	RAAN	$n\cos\Omega$	$=$	$\vec{\mathbf{n}}\cdot\hat{\mathbf{I}}$	check n_j
ω	argument of periapsis	$ne\cos\omega$	$=$	$\vec{\mathbf{n}}\cdot\vec{\mathbf{e}}$	check e_k
ν	true anomaly	$er\cos\nu$	$=$	$\vec{\mathbf{e}}\cdot\vec{\mathbf{r}}$	check $\vec{\mathbf{r}}\cdot\vec{\mathbf{v}}$

Perifocal frame

$$\vec{\mathbf{v}} = \sqrt{\frac{\mu}{p}}\left[-\sin\nu\,\hat{\mathbf{P}} + (e+\cos\nu)\,\hat{\mathbf{Q}}\right]$$

Rotation matrices

PQW to IJK

$$\mathbf{R} = \begin{bmatrix} \cos\Omega\cos\omega - \sin\Omega\sin\omega\cos i & -\cos\Omega\sin\omega - \sin\Omega\cos\omega\cos i & \sin\Omega\sin i \\ \sin\Omega\cos\omega + \cos\Omega\sin\omega\cos i & -\sin\Omega\sin\omega + \cos\Omega\cos\omega\cos i & -\cos\Omega\sin i \\ \sin\omega\sin i & \cos\omega\sin i & \cos i \end{bmatrix}$$

Constants and other parameters

Heliocentric	**Geocentric**
1 AU (or 1 DU) $= 1.4959965\times 10^8$ km	1 DU ($R_\oplus$) $= 6378.145$ km
1 TU $= 5.0226757\times 10^6$ sec	1 TU $= 806.8118744$ sec
$\mu_\odot = 1.3271544\times 10^{11}$ km^3/s^2	$\mu_\oplus = 3.986012\times 10^5$ km^3/s^2

A.5 References and further reading

One good reference for this material also has the advantage of being the least expensive. Bate, Mueller, and White[1] is an affordable Dover paperback that includes most of the basic information about orbits that you'll need to know. Another inexpensive book is Thomson's *Introduction to Space Dynamics*,[2] which covers orbital dynamics as well as attitude dynamics, rocket dynamics, and some advanced topics from analytical dynamics. Other recent texts include Prussing and Conway,[3] Vallado,[4] and Wiesel.[5]

Bibliography

[1] Roger R. Bate, Donald D. Mueller, and Jerry E. White. *Fundamentals of Astrodynamics*. Dover, New York, 1971.

[2] W. T. Thomson. *Introduction to Space Dynamics*. Dover, New York, 1986.

[3] John E. Prussing and Bruce A. Conway. *Orbital Mechanics*. Oxford University Press, Oxford, 1993.

[4] David A. Vallado. *Fundamentals of Astrodynamics and Applications.* McGraw-Hill, New York, 1997.

[5] William E. Wiesel. *Spaceflight Dynamics.* McGraw-Hill, New York, second edition, 1997.

A.6 Exercises

1. The Earth rotates about its axis with an angular velocity of about 2π radians per day (or $360°$/day). Approximately how fast does the Sun rotate about its axis?

2. Approximately how much of the Solar System's mass does the Earth contribute?

3. What are the approximate minimum and maximum communication delays between the Earth and Mars?

4. What are the Apollo asteroids?

5. What are the latitude and longitude of Blacksburg, Virginia. Give your answer in degrees, minutes, and seconds to the nearest second, and in radians to an equivalent accuracy. Also, what is the distance on the surface of the earth equivalent to 1 second of longitude error at the latitude of Blacksburg? Give your answer in kilometers. You may assume a spherical earth with a radius of 6378 km.

6. Describe the main features of the Earth's orbit about the Sun and its rotational motion.

7. Imagine that you are standing at the front center of our classroom (which you can assume to be the "origin"), facing the rear of the room (which you can assume to be North). What are the azimuth and elevation of a point 100 meters to the North, 50 meters to the East, and 10 meters above the floor of the classroom?

8. What time does this class meet in UT1?

9. Given two vectors, $\vec{\mathbf{r}}_1 = 3\,\hat{\mathbf{I}} + 4\,\hat{\mathbf{J}} + 5\,\hat{\mathbf{K}}$ and $\vec{\mathbf{r}}_2 = 4\,\hat{\mathbf{I}} - 5\,\hat{\mathbf{J}} - 2\,\hat{\mathbf{K}}$, compute the following quantities. Use a reasonably accurate sketch to illustrate the geometry of the required calculations.

   a) $\vec{\mathbf{r}}_1 + \vec{\mathbf{r}}_2$ **b)** $\vec{\mathbf{r}}_1 - \vec{\mathbf{r}}_2$

   c) $\vec{\mathbf{r}}_1 \cdot \vec{\mathbf{r}}_2$ **d)** $\vec{\mathbf{r}}_1 \times \vec{\mathbf{r}}_2$

   e) a unit vector perpendicular to both $\vec{\mathbf{r}}_1$ and $\vec{\mathbf{r}}_2$

10. Make a plot of the gravitational acceleration magnitude *vs.* radius for a satellite in Earth orbit. The abscissa should be the orbital radius in units of Earth radii (use 6378 km) from 1 to 10, and the ordinate should be acceleration in units of *gees* (1 "gee" = 9.81 m/s^2). Your plot should be labeled and should include labeled markers to indicate space station altitude, GlobalStar altitude, GPS altitude and geostationary altitude.

    Discuss the significance of the term "zero-gravity" commonly used to describe the near-Earth orbital environment.

11. A 100 kg spacecraft is in a circular orbit ($e = 0$) at 400 km altitude. What are the orbital period [s], total mechanical energy [W], specific mechanical energy [W/kg], and specific angular momentum [km^2/s^2]?

    What is the instantaneous access area of the satellite [km^2]?

    If the spacecraft is in an equatorial orbit ($i = 0$) and the position vector is $\vec{\mathbf{r}} = 6778\,\hat{\mathbf{J}}$ [km], then what is the velocity vector [km/s]? Make a supporting sketch.

12. A spacecraft is in a geostationary orbit. Compute the acceleration magnitude [m/s^2] due to the gravitational attraction of Earth, Moon, Mars, Sun, and Jupiter. You will need to make some assumptions about the relative positions of the spacecraft and the primary bodies. You should choose the closest approach (worst case) and you can assume circular orbits for all the planets.

    Discuss the significance of the two-body assumptions used in developing Keplerian motion.

A.7 Problems

1. An orbit with parameter $p = 2$ DU connects two position vectors:

$$\vec{\mathbf{r}}_1 = 3\,\hat{\mathbf{I}}\,\mathrm{DU} \qquad\qquad \vec{\mathbf{r}}_2 = 4\,\hat{\mathbf{J}}\,\mathrm{DU}$$

    Assuming a "long-way" trajectory, how long does a spacecraft take to get from $\vec{\mathbf{r}}_1$ to $\vec{\mathbf{r}}_2$?

    $t =$ ____________________________________ [TU]

2. A mission to Mars begins in a parking orbit at 500 km altitude above the Earth. Assume the parking orbit is in the plane of the ecliptic, and that Earth and Mars are in coplanar circular orbits.

    (a) 5 points. Determine the angular separation required between Earth and Mars at the time of initiating a heliocentric Hohmann transfer.

$\theta =$ ______________________________ [rad]

(b) 5 points. Determine the Δv required to get from the same heliocentric circular orbit that the Earth is on to the heliocentric orbit of Mars, using a Hohmann transfer.

$\Delta v =$ ______________________________ [km/s]

(c) 10 points. Determine the actual Δv required to get from the Earth parking orbit to the heliocentric transfer orbit.

$\Delta v =$ ______________________________ [km/s]

(d) 5 points. Assuming that the maneuver takes place on the first day of spring, determine the sidereal time of the position at which the actual Δv takes place.

$\theta_{\Delta v} =$ ______________________________ [degrees]

3. The Mars Global Surveyor used an aerobraking maneuver to reduce the energy in the orbit until it reached a low Mars orbit suitable for its surface-mapping mission (400 km altitude circular orbit). Useful information about Mars is in your textbook on pp. 360–361.

   The aerobraking maneuver is equivalent to applying a Δv at each periapsis to reduce the periapsis speed, thereby reducing the energy and lowering the apoapsis altitude, without appreciably affecting periapsis altitude.

   (a) 20 points. Assume that the periapsis altitude is 110 km. If the apoapsis altitude in one orbit is 20,000 km, and the apoapsis altitude in the next orbit is 19,500 km, determine what the *effective* Δv at periapsis was (in km/s).

   $\Delta v =$ ______________________________ [km/s]

   (b) 5 points. Assuming that the effective Δv is the same at each periapsis passage, determine the number of orbits it will take to lower the apoapsis to $\leq$ 400 km.

   $N =$ ______________________________ [orbits]

Appendix C

MatLab

This appendix is intended to help you understand numerical integration and to put it into practice using Matlab's `ode45` function.

C.1 Integration

We want to *integrate* ordinary differential equations (ODEs) of the form

$$\dot{x} = f(x,t), \quad \text{with initial conditions} \;\; x(0) = x_0 \tag{C.1}$$

where the *state* x may be a scalar or an $N \times 1$ matrix, and t is a scalar. This problem is known as an *initial value problem*, the variable t is the *independent variable* (usually time), and the state x is the *dependent variable*. The result of integration is to find the solution $x(t)$ that satisfies the differential equation and initial conditions given in Eq. (C.1).

The function $f(x,t)$ is frequently referred to as the *right-hand side* of the ODE, and most numerical algorithms involve repetitive calculation of the right-hand side for many values of x and t. Since $f(x,t)$ is the rate of change of the dependent variable x with respect to the independent variable t, we can think of the right-hand side as the *slope* of $x(t)$, even if x is a multi-dimensional state vector.

C.2 Some Example Problems of the Form of Eq. (C.1)

C.2.1 A simple first-order ODE

One of the simplest possible examples is

$$\dot{x} = a, \quad \text{with initial conditions} \;\; x(0) = b \tag{C.2}$$

where a and b are constants, and x is a scalar. This equation is easily integrated *analytically* to obtain

$$x(t) = at + b \tag{C.3}$$

which defines a straight line in the (t, x) plane, with slope a and intercept b. Notice how the right-hand side is in fact the slope of $x(t)$, as mentioned in the previous section.

C.2.2 A slightly less simple first-order ODE

Instead of a constant right-hand side, let the right-hand side have a t-dependence:

$$\dot{x} = at, \quad \text{with initial conditions} \;\; x(0) = b \tag{C.4}$$

where a and b are constants, and x is a scalar. This equation is also easily integrated *analytically* to obtain

$$x(t) = \frac{1}{2}at^2 + b \tag{C.5}$$

which defines a parabola in the (t, x) plane, with slope at any given point given by at and intercept b. Notice how the right-hand side is again the slope of $x(t)$, which is changing as a function of t.

C.2.3 A more interesting, but still simple first-order ODE

Instead of a t-dependence, let the right-hand side depend on the dependent variable:

$$\dot{x} = ax, \quad \text{with initial conditions} \;\; x(0) = b \tag{C.6}$$

where a and b are constants, and x is a scalar. This equation is also easily integrated *analytically* by using separation of variables (exercise!) to obtain

$$x(t) = be^{at} \tag{C.7}$$

which defines an exponential curve in the (t, x) plane, with slope at any given point given by $abe^{at} = ax(t)$ and intercept b. Notice how the right-hand side is still the slope of $x(t)$, which is again changing as a function of t.

Remark 1 *All of the preceding examples are of the class known as linear first-order ordinary differential equations. The two examples where $f(x, t)$ does not have any t-dependence are also known as constant-coefficient or time-invariant differential equations. The example where $f(x, t)$ does have time-dependence is known as a time-varying differential equation.*

C.2.4 Some nonlinear first-order ODEs

If the right-hand side has a nonlinear dependence on the dependent variable, then the equation is a nonlinear differential equation. Here are some examples:

$$\dot{x} = ax^2 \tag{C.8}$$

$$\dot{x} = a \sin x \tag{C.9}$$
$$\dot{x} = ae^x \tag{C.10}$$
$$\dot{x} = a \log x \tag{C.11}$$

All of these examples can be integrated *analytically* to obtain the solution $x(t)$, because the equations are all *separable.* As an exercise, you should separate the equations and carry out the integration.

C.2.5 Second-order ODEs

Many problems of interest to engineers arise in the form of second-order ODEs, because many problems are developed by using Newton's second law,

$$f = ma \quad e.g. \quad f = m\ddot{x} \tag{C.12}$$

where f is the force acting on an object of mass m, and a is the acceleration of the object, which might be for example the second derivative of a position described by the cartesian coordinate, x.

A familiar example is the mass-spring-damper system, where a mass m has position described by a scalar variable x and is acted on by a spring force $f_s = -kx$, by a damper force $f_d = -c\dot{x}$, and by a control force f_c. Application of $f = ma$ to this problem leads to the second-order ODE

$$m\ddot{x} + c\dot{x} + kx = f_c \tag{C.13}$$

Assuming that the *parameters* m, c, and k are all constant, then this equation is a second-order, constant-coefficient (time-invariant), linear ODE. Furthermore, if the control force f_c is zero, then the equation is called a homogeneous ODE.

As with the simple first-order examples already presented, this equation can be solved analytically. You should review the analytical solution to this equation in your vibrations and ODE textbooks. One can also put this second-order ODE into the first-order form of a *system* of first-order ODEs as follows. First define two *states* x_1 and x_2 as the position and velocity; *i.e.*

$$x_1 = x \tag{C.14}$$
$$x_2 = \dot{x} \tag{C.15}$$

Differentiating these two states leads to

$$\dot{x}_1 = \dot{x} = x_2 \tag{C.16}$$
$$\dot{x}_2 = \ddot{x} = \frac{f_c}{m} - \frac{k}{m}x - \frac{c}{m}\dot{x} = u - \hat{k}x_1 - \hat{c}x_2 \tag{C.17}$$

where we have introduced three new symbols: $u = f_c/m$ is the acceleration due to the control force, and $\hat{k}$ and $\hat{c}$ are the spring and damping coefficients divided by the mass. These equations may be written in matrix form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\hat{k} & -\hat{c} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{C.18}$$

This form of the equation is so common throughout the dynamics and control literature that it has a standard notation. If we denote the 2×1 state vector by $\mathbf{x} = [x_1 \;\; x_2]^T$, the 2×2 matrix that appears in the equation by $\mathbf{A}$, and the 2×1 matrix that multiplies u by $\mathbf{B}$, then we can write the system as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \tag{C.19}$$

In this form of linear, constant-coefficient ODEs, the variable matrix $\mathbf{x}$ is known as the *state vector*, and the variable u is known as the *control*. The matrices $\mathbf{A}$ and $\mathbf{B}$ are the *plant* and *input* matrices, respectively.

Since the input must either be constant, a function of time, a function of the state, or a function of both the time and the state, then this system of equations may be written in the even more general form of

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \tag{C.20}$$

Note the similarity between this ODE and Eq. (C.1). The only difference is that we have introduced the convention of using a bold-face font to denote that a particular variable or parameter is a matrix.

Also note that the right-hand side is the time-rate of change of the state, which can be interpreted as the slope of the state as a function of time.

In general, most ordinary differential equations can be written in the form of Eq. (C.20). In the next subsection we give some higher-order examples.

C.2.6 Higher-order ODEs

The rotational equations of motion for a rotating rigid body may be written as the combination of Euler's equations and a set of kinematics equations. A typical choice for kinematics variables is the quaternion. These equations are:

$$\dot{\omega} = -\mathbf{I}^{-1}\left[\omega^{\times}\mathbf{I}\omega + \mathbf{g}\right] \tag{C.21}$$

$$\dot{\bar{\mathbf{q}}} = \mathbf{Q}(\bar{\mathbf{q}})\omega \tag{C.22}$$

Note that the right-hand side of the $\dot{\omega}$ equation depends on ω and the torque $\mathbf{g}$, which (like the control in Eq. (C.19)) may depend on the angular velocity ω, on the attitude

$\bar{\mathbf{q}}$, and on the time t. The right-hand side of the $\dot{\bar{\mathbf{q}}}$ equation depends on $\bar{\mathbf{q}}$ and ω. Thus the equations are coupled. The combination of ω and $\bar{\mathbf{q}}$ forms the state; *i.e.*

$$\mathbf{x} = \begin{bmatrix} \omega \\ \bar{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{C.23}$$

Thus the equations can be combined into a single state vector differential equation (or system of equations):

$$\begin{bmatrix} \dot{\omega} \\ \dot{\bar{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} -\mathbf{I}^{-1}\left[\omega^{\times}\mathbf{I}\omega + \mathbf{g}\right] \\ \mathbf{Q}(\bar{\mathbf{q}})\omega \end{bmatrix} \tag{C.24}$$

which may be written in the more concise form of

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \tag{C.25}$$

As noted above, most dynamics and control problems can be written in this form, where the right-hand side corresponds to the rate of change of the state and depends on both the state and time.

C.3 Analytical *vs.*Numerical Solutions

We use the term *analytical* to refer to a solution that can be written in closed form in terms of known function such as trigonometric functions, hyperbolic functions, elliptic functions, and so forth. Most dynamics and control problems do not admit analytical solutions, and so the solution $\mathbf{x}(t)$ corresponding to a given set of initial conditions must be found numerically using a numerical integration algorithm. A numerical integration algorithm can be tested by using it to integrate a system of equations that has a known analytical solution and comparing the results.

C.4 Numerical Integration Algorithms

There are many popular numerical integration algorithms appropriate for solving the initial value problems of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \quad \text{with initial conditions} \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{C.26}$$

The simplest algorithm of any practical use is known as Euler integration. A slightly more complicated algorithm with substantially better performance is the Runge-Kutta algorithm. Higher-order (more accurate, but more complicated) algorithms are widely available, and usually one just selects an existing implementation of a suitable algorithm and uses it for the problem of interest. In most cases, the algorithm needs to know the initial and final times (t_0 is usually 0, and t_f), the initial conditions ($\mathbf{x}_0$), and how to calculate the right-hand side or slope $\mathbf{f}(\mathbf{x}, t)$ for given values of the state and time.

C.4.1 Euler integration

Euler integration is based directly on the notion that the right-hand side defines the slope of the function $\mathbf{x}(t)$. Since we know the initial conditions, $\mathbf{x}_0$, the initial slope is simply $\mathbf{f}(\mathbf{x}_0, t_0)$, and one might reasonably suppose that the slope is nearly constant for a short "enough" timespan following the initial time. Thus, one chooses a small "enough" time step, Δt, and estimates the state at time $t_0 + \Delta t$ as

$$\mathbf{x}_1 = \mathbf{x}(t_0 + \Delta t) \approx \mathbf{x}_0 + \mathbf{f}(\mathbf{x}_0, t_0)\Delta t \tag{C.27}$$

Continuing in this manner, one obtains a sequence of approximations at increasing times as

$$\begin{aligned}
\mathbf{x}_1 &= \mathbf{x}_0 + \mathbf{f}(\mathbf{x}_0, t_0)\Delta t & \text{(C.28)}\\
\mathbf{x}_2 &= \mathbf{x}_1 + \mathbf{f}(\mathbf{x}_1, t_1)\Delta t & \text{(C.29)}\\
\mathbf{x}_3 &= \mathbf{x}_2 + \mathbf{f}(\mathbf{x}_2, t_2)\Delta t & \text{(C.30)}\\
&\cdot & \text{(C.31)}\\
&\cdot & \text{(C.32)}\\
&\cdot & \text{(C.33)}\\
\mathbf{x}_n &= \mathbf{x}_{n-1} + \mathbf{f}(\mathbf{x}_{n-1}, t_{n-1})\Delta t & \text{(C.34)}
\end{aligned}$$

where $t_j = t_{j-1} + \Delta t$. This method is easy to understand, and easy to implement; however it is not accurate enough for integrating over any reasonably long time intervals. It is accurate enough to use in short time integrations though.

C.4.2 Runge-Kutta integration

C.5 Numerical Integration Using Matlab

Matlab has several numerical integration algorithms implemented as `.m` files. The most commonly used of these algorithms is `ode45`, and some examples using this function are provided below. In most implementations, the use of the integration function requires the development of two Matlab `.m` files: a *driver* file and a *right-hand side* file. For the examples here, we use the filenames `driver.m` and `rhs.m`. The driver file is a script file, meaning that it contains a sequence of Matlab statements that are executed in order. The rhs file is a function file, meaning that it defines a function with arguments (generally t and $\mathbf{x}$) that returns the value of the right-hand side (or slope) at the given values of time and state.

The driver file is responsible for defining parameters and initial conditions, calling the integration function, and then completing any post-processing of the results, such as making plots of the state variables as functions of time.

First we present a simple implementation of the Runge-Kutta algorithm.

C.5.1 An implementation of Runge-Kutta

Here is a sample integration algorithm that works similarly to the `ode45` algorithm, but is simpler. It implements the standard Runge-Kutta algorithm. It is a Matlab function, and has 3 arguments: `frhs` is a string containing the name of the `.m` file that computes the right-hand side of the differential equations; `tspan` is a $1 \times N$ matrix of the times at which the algorithm will sample the right-hand side $(t_0, t_0 + \Delta t, ...t_f)$; and `x0` is the initial state vector.

```
% oderk.m
% C. Hall
% [t, x] = oderk(frhs,tspan,x0)
% implements runge-kutta 4th order algorithm
% frhs = 'filename' where filename.m is m-file
% tspan = 1xN matrix of t values
% x0 = nx1 matrix of initial conditions
% returns t=tspan, x=nxN matrix
%                  x(:,i) = x(t(i))
function [tspan,x] = oderk(frhs,tspan,x0)
    N=length(tspan);
    n=length(x0);
    x0=reshape(x0,n,1);
    x=[x0 zeros(n,N-1)];
    w=x0;
    for i=1:N-1
        h=tspan(i+1)-tspan(i);
        t=tspan(i);
        K1=h*feval(frhs,t,w);
        K2=h*feval(frhs,t+h/2,w+K1/2);
        K3=h*feval(frhs,t+h/2,w+K2/2);
        K4=h*feval(frhs,t+h,w+K3);
        w=w+(K1+2*K2+2*K3+K4)/6;
        x(:,i+1)=w;
    end
    x=x';
```

As should be evident from the function, it divides each time interval into 2 segments and computes the right-hand side (slope) at the initial, final, and midpoint times of each interval. You should make an effort to know how these algorithms work, but you can use them as *black boxes* just as you might use an eigenvalue solver. That is, you can think of `oderk` as a function that takes its three arguments and returns the state vector calculated at each of the times in the `tspan` argument.

Here is an example illustrating the integration of a single first-order equation, $\dot{x} = ax$, $x(0) = x_0$, from $t = 0$ to $t = 10$ with a Δt of 0.1 s. The driver file is presented first.

```
% driver.m
% this file sets the parameters and initial conditions,
% calls oderk, and makes a plot of x(t)
global a                        % make the parameter a global
a     = 0.1;                    % set the parameter a
x0    = 1.0;                    % initial condition of state
t     = 0:0.1:10;               % [0 0.1 0.2 ... 10]
fname = 'rhs';                  % the name of the rhs file (.m is assumed)
[t,x] = oderk(fname,t,x0);      % turn the work over to oderk
plot(t,x)                       % make the plot
xlabel('t');                    % add some labels
ylabel('x');
title('Sample Runge-Kutta Solution of First-Order ODE');
```

The right-hand side file is quite simple.

```
% rhs.m
% xdot = rhs(t,x)
function xdot = rhs(t,x)
global a
xdot = a*x;
% that's all!
```

The same integrator (`oderk`) can be used to integrate any system of ODEs. For example, suppose we have the system of three first-order equations:

$$\dot{x} = \sigma(y - x) \tag{C.35}$$

$$\dot{y} = -xz + rx - y \tag{C.36}$$

$$\dot{z} = xy - bz \tag{C.37}$$

These equations are called the Lorenz equations and are considered to be the first widely known system of equations admitting chaotic solutions*. The behavior of solutions to these equations is strongly dependent on the values of the three parameters b,

*For more information on chaotic systems, see Gleick's book *Chaos*, Moon's *Chaotic and Fractal Dynamics*, or Tabor's *Chaos and Integrability in Nonlinear Dynamics*.

σ, and r. An interesting chaotic attractor (also known as a *strange attractor*) occurs for the parameter values $b = 8/3$, $\sigma = 10$, $r = 28$.

Here are driver and rhs files to implement these equations and compute the strange attractor. Try it out.

```
% lorenzdriver.m
% this file sets the parameters and initial conditions,
% calls oderk, and makes plots
global b sig r              % make the parameters global
b     = 8/3;                % set the parameters
sig   = 10;
r     = 28;
x0    = [0; 1; 0];          % initial conditions of state vector
t     = 0:0.01:50;           % [0 0.01 0.02 ... 50]
fname = 'lorenzrhs';        % the name of the rhs file (.m is assumed)
[t,x] = oderk(fname,t,x0);  % turn the work over to oderk
plot3(x(:,1),x(:,2),x(:,3));% make a 3-d plot  (cool!)
xlabel('x');                % add some labels
ylabel('y');
zlabel('z');
title('Sample Runge-Kutta Solution of Lorenz System');
```

The right-hand side file is quite simple.

```
% lorenzrhs.m
% xdot = rhs(t,x)
function xdot = rhs(t,x)
global b sig r
xx = x(1);                   % the x component
y = x(2);                    % and y ...
z = x(3);
xdot = [ sig*(y-xx);  -xx*z+r*xx-y;  xx*y-b*z];
% that's all!
```

C.5.2 Using `ode45`

The calling format for using `ode45` is identical to that for calling `oderk`. However, there are numerous additional arguments that one can use in calling `ode45` that I have not included in my implementation of `oderk`. You can find out what the options are by typing `help ode45` at the Matlab prompt. Probably the most useful of these additional arguments are the `options` and `parameters` arguments. For example, instead of making the parameters in the Lorenz equations example `global` variables, we can pass them to the right-hand side file as parameters. And we can also set the desired precision we want from the integration. The following driver and rhs files implement these two options.

```
% lorenzdriverp.m
% this file sets the parameters and initial conditions,
% calls ode45, and makes plots

b       = 8/3;                % set the parameters
sig     = 10;
r       = 28;
x0      = [0; 1; 0];          % initial conditions of state vector
t       = 0:0.01:50;          % [0 0.01 0.02 ... 50]
fname   = 'lorenzrhs';        % the name of the rhs file (.m is assumed)
options = odeset('abstol',1e-9,'reltol',1e-9); % set tolerances
[t,x]   = ode45(fname,t,x0,options,b,sig,r);
plot3(x(:,1),x(:,2),x(:,3));  % make the plot
xlabel('x');                  % add some labels
ylabel('y');
zlabel('z');
title('Sample ode45 Solution of Lorenz System');

% lorenzrhsp.m
% xdot = rhs(t,x,b,sig,r)
function xdot = rhs(t,x,b,sig,r)
xx = x(1);                   % the x component
y  = x(2);                   % and y ...
z  = x(3);
xdot = [ sig*(y-xx);  -xx*z+r*xx-y;  xx*y-b*z];
% that's all!
```

C.6 References and further reading

Bibliography

C.7 Exercises

Get into MatLab by typing matlab at the DOS or Unix prompt, or by launching the MatLab application if you're using Windows or a Mac.

1. Compute the following quantities:

$\sin^2(\pi/4) + \cos^2(\pi/4)$	by typing `sin(pi/4)^2+cos(pi/4)^2`
$e^{i\pi} + 1$	by typing `exp(sqrt(-1)*pi)+1`
$\ln(e^3), log_{10}(e^3), log_{10}(10^6)$	
$(1 + 3i)/(1 - 3i)$	can you check this result by hand?
$x = 32\pi, y = \cosh^2 x - \sinh^2 x$	
`exp(pi/2*i)-exp(pi/2i)`	*why isn't this zero?*

2. Create some vectors and matrices:

```
u = [1 2 3 4 5 6 7 8 9]                  (a row vector)
v = 1:9                                  (a row vector)
w = rand(1,9)                            (a random row vector)
wp = w'                                  (a column vector)
M = [sin(u)' cos(v)' tan(w') atan(wp)]   (a 9-by-4 matrix)
M4 = M(1:4,1:4)                          (1st 4 rows and columns of M)
```

Can you multiply **M** and either of the vectors? Remember that to multiply two matrices **A** and **B**, where **A** has m rows and n columns, and **B** has p rows and q columns, then $n = p$ is required.

3. Make some simple plots

$y = \sin x, x \in [0, 2\pi]$ `x = 0:2*pi; y = sin(x); plot(x,y)`

This might look a bit funny, so try making the "step" smaller

`x = 0:0.1: 2*pi; y = sin(x); plot(x,y)`

You can make it even smoother by using

`x = 0:0.01: 2*pi; y = sin(x); plot(x,y)`

Add some labels `xlabel('This is the x label')`

`ylabel('This is the y label')`

And a title `title('This is the title')`

And a legend `legend('This is the legend')`

Now, make some plots of your own.

4. Make a more complicated plot

$y = e^{-0.4x} \sin x, x \in [0, 2\pi]$ `x = linspace(0,2*pi,100);`

`y = exp(-0.4*x).*sin(x); plot(x,y)`

Label the axes, create a nifty title and legend. Type `help plot` and use the information you get to make another plot of just the unconnected data points. Type `help linspace` to find out what that function does.

5. Make some `log` and `semilog` plots of $y = \ln x$, $y = \log_{10} x$, $y = x^2$, and $y = x^3$. The commands to use are `semilogx`, `semilogy`, and `loglog`. Use help to find out how to use them. To create a vector of x^3, type `y = x.^3`.

6. Make a polar plot of the spiral $r = e^{0.2\theta}, \theta \in [0, 5\pi]$. Use the command `polar(theta,r)`.

7. Make a 3-dimensional plot of the helix $x = \sin t$, $y = \cos t$, $z = t$, for $t \in [0, 6\pi]$. Create the vectors t, x, y, and z, then use the command `plot3(x,y,z)`. Add some labels, and check out the `print` command.

8. Create a menu that gives a choice between two options. Use the command `choice = menu('MenuName', 'ChoiceA', 'ChoiceB')`. If you click `ChoiceA`, then the variable choice will be assigned the value 1, and if you click `ChoiceB`, the variable choice will be assigned the value 2. Menus are most useful when used in `MatLab ".m"` files.

9. Create some more matrices

```
I  = eye(5)                a 5-by-5 identity matrix
d  = diag(I)               a 5-by-1 matrix of the diagonal elements
A  = rand(5,4)             a 5-by-4 matrix with random elements
d1 = diag(A,1)             the "super-diagonal" elements
d2 = diag(A,-1)            the "sub-diagonal" elements
u  = linspace(pi,8*pi,100) a linearly spaced vector of length 100,
                           starting at pi and going to 8*pi
v  = logspace(pi,8*pi,100) a logarithmically spaced vector of
                           length 100, from pi to 8*pi
```

10. Do some logical operations

```
x = 1; y = 2;
if ( x > y )
    disp('x > y')
else
    disp('y >= x')
end
```

Some vector-based examples:

```
x = 1:5;                two vectors
y = [-1 2 -2 3 4];
m = x > y               0 for each false, 1 for each true
n = x > 4               ditto
p = m & n               logical AND
q = m | n               logical OR
```

11. Writing M-files

For serious MatLab use, you need to write M-files, i.e., files with a .m extension, containing MatLab statements. To create and edit files, you need to use a text editor. Once you've written an M-file, you will want to run it. For example, to execute the MatLab code in myprob.m, you type myprob at the ¿¿ prompt. You can also write your own MatLab functions, which also live in M-files. The main difference between a function M-file and an ordinary M-file, is that a function M-file must begin with the statement `function result = fname(arguments)` where `result` is the variable(s) for the function to return, `fname` is the name of the function, and `arguments` is a list of comma-separated variables that the function is to use. If you put comment statements before the function statement, they will be printed if you type `help fname`. Here are some examples:

```
% this function returns sin(x)*sin(x)
% put this function in a file named
%   sin2.m
% to call this function use the statement
%   y = sin2(x)
%
function y = sin2(x)
    s = sin(x);
    y = s*s;
```

```
% This function has as its argument a number s.
% It first takes the absolute value of s,
% then creates a t vector with 101 elements ranging
% from 0 to s, and evenly spaced. It returns t and sin(t)
% Put this function in a file named
%   myf.m
% to call this function use the statement
%   [myx,myt] = myf(mys)
%
function [x,t] = myf(s)
    s = abs(s);
    t = 0:s/100:s;
    x = sin(t);
```

```
% This is a more advanced function using the feval feature.
% This function finds a root of a mathematical function
```

```
% using the Bisection Method.
% Put this function in a file named
%   bisect.m
% to call this function use the statement
%   [root, steps] = bisect('function',guess,tolerance)
%
function [root,steps] = bisect(f,x,tol)
    if nargin < 3, tol = eps; end
    trace = (nargout==2);
    if x ~= 0, dx = x/20; else, dx=1/20; end
    a = x-dx;  fa = feval(f,a);
    b = x+dx;  fb = feval(f,b);

    % find change of sign
    while (fa > 0) == (fb > 0)
        dx = 2.0*dx;
        a = x-dx; fa = feval(f,a);
        if (fa > 0) ~= (fb > 0), break, end
        b = x + dx; fb = feval(f,b);
    end
    if trace, steps = [a fa; b fb]; end

    % Main Loop
    while abs(b-a) > 2.0*tol*max(abs(b),1.0)
        c = a + 0.5*(b-a);  fc = feval(f,c);
        if trace, steps = [steps; [c fc]]; end
        if (fb > 0) == (fc > 0)
            b = c; fb = fc;
        else
            a = c; fa = fc;
        end
    end
```

C.8 Problems

1. Develop a simple "simulator" that implements the rotational equations of motion for a rigid body in a central gravitational field. Use quaternions for the kinematics equations. Your simulator should allow you to specify the initial conditions on ω and $\bar{\mathbf{q}}$, the principal moments of inertia, and starting and stopping times. You should be able to plot ω and $\bar{\mathbf{q}}$ as functions of time.

   Using your simulator, make two plots for a satellite in a circular, equatorial orbit with $a = 7000$ km: ω^{bo} *vs.*t and $\bar{\mathbf{q}}^{bo}$ *vs.*t for $t \in [0, 1000]$. Use principal moments of inertia $15, 20, 10$, and initial conditions

$$\omega^{bo} = (0.1, 0.2, 0.3)\text{rad/s and } \bar{\mathbf{q}}^{bo} = (0.0413, 0.0100, 0.0264, 0.9988)$$

You must turn in the following:

(a) A report describing the problem, the equations, your approach, your analysis, your simulator, and your results.

(b) The two requested plots.

(c) Similar plots for two different sets of initial conditions to be chosen by you. Give me the initial conditions so I can check them.

(d) A copy of the code that implements your simulator, with enough comments that I can tell how it works.