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

Copyright Chris Hall March 18, 2003

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*

Copyright Chris Hall March 18, 2003

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.

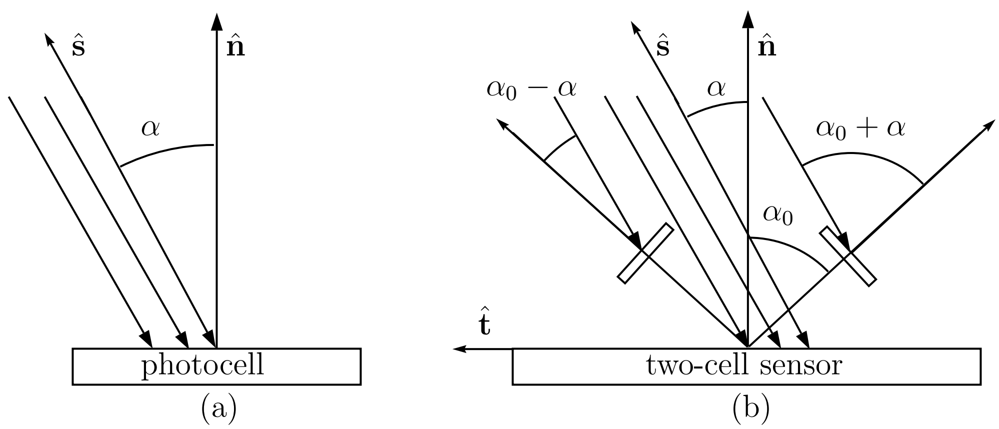

Figure 4.1: Photocells for Sun Sensors. (a) Single photocell. (b) Pair of photocells for measurement of α in $\hat{\mathbf{n}} - \hat{\mathbf{t}}$ plane.

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

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

Copyright Chris Hall March 18, 2003

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

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

Copyright Chris Hall March 18, 2003

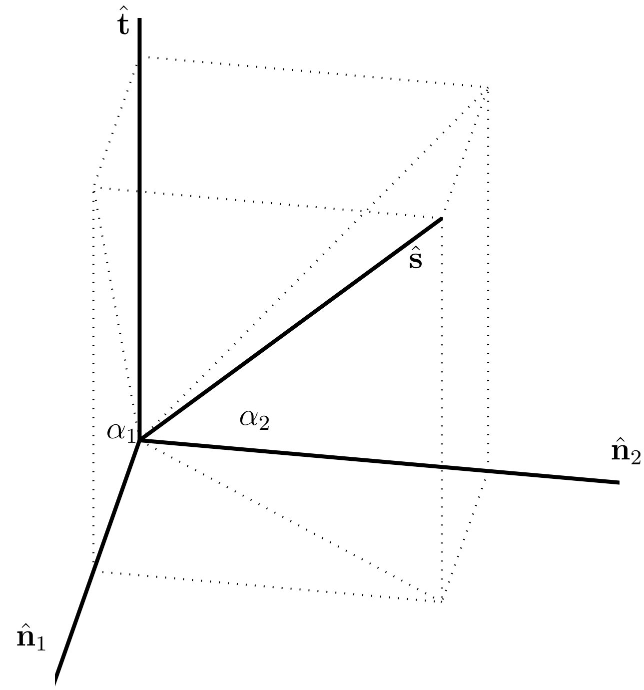

Figure 4.2: Geometry for a four-photocell sun sensor

we obtain the sun vector components by normalizing this vector:

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

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* $\mathbf{s}_b = [-0.7789 \ \ 0.5920 \ \ 0.2071]^{\mathsf{T}}$.

Copyright Chris Hall March 18, 2003

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, $\mathbf{s}_i$.*

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

Copyright Chris Hall March 18, 2003

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

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 $\mathcal{F}_i$.

Copyright Chris Hall March 18, 2003

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

Copyright Chris Hall March 18, 2003

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,

$$\begin{aligned} \hat{\mathbf{t}}_1 &= \hat{\mathbf{s}} && (4.18) \\ \mathbf{t}_{1b} &= \mathbf{s}_b && (4.19) \\ \mathbf{t}_{1i} &= \mathbf{s}_i && (4.20) \end{aligned}$$

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,

$$\begin{aligned} \hat{\mathbf{t}}_2 &= \hat{\mathbf{s}} \times \hat{\mathbf{m}} && (4.21) \\ \mathbf{t}_{2b} &= \frac{\mathbf{s}_b^\times \mathbf{m}_b}{|\mathbf{s}_b^\times \mathbf{m}_b|} && (4.22) \\ \mathbf{t}_{2i} &= \frac{\mathbf{s}_i^\times \mathbf{m}_i}{|\mathbf{s}_i^\times \mathbf{m}_i|} && (4.23) \end{aligned}$$

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:

$$\begin{aligned} \hat{\mathbf{t}}_3 &= \hat{\mathbf{t}}_1 \times \hat{\mathbf{t}}_2 && (4.24) \\ \mathbf{t}_{3b} &= \mathbf{t}_{1b}^\times \mathbf{t}_{2b} && (4.25) \\ \mathbf{t}_{3i} &= \mathbf{t}_{1i}^\times \mathbf{t}_{2i} && (4.26) \end{aligned}$$

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.

Copyright Chris Hall March 18, 2003

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

Copyright Chris Hall March 18, 2003

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.

Copyright Chris Hall March 18, 2003

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

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

Copyright Chris Hall March 18, 2003

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

$$\mathbf{F} = \frac{\partial J}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial J}{\partial x_1} & \frac{\partial J}{\partial x_2} & \frac{\partial J}{\partial x_3} \end{bmatrix}^\mathsf{T} \tag{4.54}$$

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

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

Copyright Chris Hall March 18, 2003

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,

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

Copyright Chris Hall March 18, 2003

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

$$\mathbf{B} = \sum_{k=1}^{N} w_k \left(\mathbf{v}_{kb}\mathbf{v}_{ki}{}^{\mathsf{T}}\right) \tag{4.66}$$

$$\mathbf{S} = \mathbf{B} + \mathbf{B}^{\mathsf{T}} \tag{4.67}$$

$$\mathbf{Z} = \begin{bmatrix} B_{23} - B_{32} & B_{31} - B_{13} & B_{12} - B_{21} \end{bmatrix}^{\mathsf{T}} \tag{4.68}$$

$$\sigma = \mathsf{tr}\,[\mathbf{B}] \tag{4.69}$$

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.*

Copyright Chris Hall March 18, 2003

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 $\pm 5°$. *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

Copyright Chris Hall March 18, 2003

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 **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

Copyright Chris Hall March 18, 2003

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

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

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:

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

Copyright Chris Hall March 18, 2003

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.

Copyright Chris Hall March 18, 2003

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.

Copyright Chris Hall March 18, 2003

[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

Copyright Chris Hall March 18, 2003

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.

Copyright Chris Hall March 18, 2003

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?

Copyright Chris Hall March 18, 2003