7

Estimation of Dynamic Systems: Applications

In theory, there is no difference between theory and practice. But, in practice, there is. van de Snepscheut, Jan

THE previous chapters provided the basic concepts for state estimation of dynamic systems. The foundations of these chapters still rely on the estimation results of Chapter 1 and the probability concepts introduced in Chapter 2. Applications of the fundamental concepts have been shown for various systems in Chapter 4. In this chapter these applications are extended to demonstrate the power of the sequential Kalman filter and batch estimation algorithms. As with Chapter 4, this chapter shows only the fundamental concepts of these applications, where the emphasis is upon the utility of the estimation methodologies. The interested reader is encouraged to pursue these applications in more depth by studying the references cited in this chapter.

7.1 GPS Position Estimation

In §4.1 nonlinear least squares has been used to determine the position of a vehicle using Global Positioning System (GPS) pseudorange measurements. An application of this concept has been demonstrated in example 4.1 using simulated GPS satellite position locations. In the example the GPS locations are shown in Earth-Centered-Earth-Fixed (ECEF), which provides an easy approach to convert the position of a vehicle into longitude and latitude. However, example 4.1 shows only a point-by-point solution approach (i.e., only one specific solution in time). The results of this chapter give a trajectory of solutions over an entire time interval. Hence, the evolution of the positions of the GPS satellites over time must be discussed here. Using these GPS positions a Kalman filter application will then be shown to determine filtered position and velocity estimates.

7.1.1 GPS Coordinate Transformations

In this section we first review the basic concepts of the Earth-Centered-Inertial (ECI) to ECEF transformation, which can be used to propagate the GPS satellite

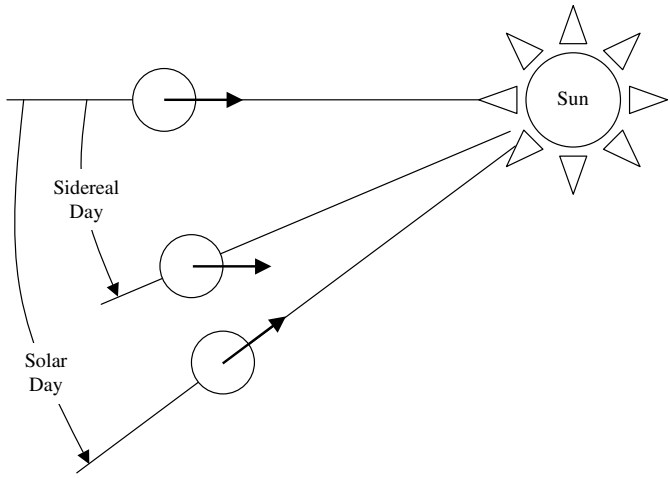

Figure 7.1: Solar and Sidereal Day

positions over time. Also, specific formulas for the conversion from ECEF coordinates to longitude and latitude are shown. The ECEF coordinate system, denoted by $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3\}$, is similar to the ECI coordinate system shown in Figure 4.7. In fact the $\hat{\mathbf{i}}_3$ axis is identical, so that $\hat{\mathbf{e}}_3 = \hat{\mathbf{i}}_3$. The only difference between these two coordinates systems is that while the ECI $\hat{\mathbf{i}}_1$ axis is inertially fixed, the ECEF $\hat{\mathbf{e}}_1$ axis points at the Earth's prime meridian, which rotates with the Earth. Also, $\hat{\mathbf{e}}_2$ rotates through the same angle as $\hat{\mathbf{e}}_1$. This rotation angle is denoted by θ in Figure 4.7. The transformation from ECI to ECEF follows

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} \tag{7.1}$$

where i_1, i_2, and i_3 are the components of the ECI position vector, and e_1, e_2, and e_3 are the components of the ECEF position vector.

In order to determine the ECEF position vector we must first determine the angle θ, which is related to time. Suppose that we know our current time in Universal Time (UT), which is defined by the Greenwich hour angle augmented by 12 hours of a fictitious Sun uniformly orbiting in the equatorial plane.[1] At first glance one might believe that θ equals zero at midnight UT, which corresponds directly to the solar day time. To see why this thinking is erroneous consider the exaggerated angular movement of a solar day shown in Figure 7.1. A solar day is the length of time that elapses between the Sun reaching its highest point in the sky two consecutive times. However, the ECI coordinate system is fixed relative to the stars, not the Sun. A *sidereal day* is the length of time that passes between a given fixed star in the sky crossing a given projected meridian. From Figure 7.1 a sidereal day is clearly shorter than a solar day. This difference is about 4 minutes.[2]

The Greenwich Mean Sidereal Time (GMST) is the mean sidereal time at zero longitude, which can be given by the angle θ. Several formulas for the conversion from UT to GMST are given in the open literature (e.g., see Ref. [3]). One of the most widely-used formulas is presented by Meeus.[4] First, given UT year y, month m, day d, hour h, minute min, and second s, compute the days past or before the year 2000 using

$$\boxed{\begin{aligned} d_{2000} =& 367y - \text{INT}\left\{\frac{7\{y + \text{INT}[(m+9)/12]\}}{4}\right\} \\ &+ \text{INT}\left\{\frac{275m}{9}\right\} + \frac{h + min/60 + s/3600}{24} + d - 730531.5 \end{aligned}} \tag{7.2}$$

where INT is the integer part of the fraction (e.g., INT(23.8) = 23). The angle θ in degrees is given by

$$\boxed{\theta = 280.46061837 + 360.98564736628 \times d_{2000}} \tag{7.3}$$

Another formula that takes into account the precession and nutation of the Earth that occurs as a result of the Moon's motion is given in Ref. [4], which is accurate to within 0.03 seconds up to the year 2050. However, eqn. (7.3) is accurate enough for simulation purposes.

The GPS satellite information is usually given by a GPS almanac, which provides orbital element information, including: eccentricity, inclination, semimajor axis, right ascension, argument of perigee, and mean anomaly.* These parameters can be converted to an initial ECI position and velocity using the method described in §3.8.2. The ECI position and velocity at any time can be computed using eqn. (3.202). It should be noted that GPS time is based on the atomic standard time and is continuous without the leap seconds of UT, due to the non-smooth rotation of the Earth. GPS epoch is midnight of January 6, 1980, and GPS time is conventionally represented in weeks and seconds from this epoch. The GPS week is represented by an integer from 0 to 1023. A rollover occurred on August 22, 1999, so that 1024 needs to be added for references past this date. For simulation purposes counting the days past GPS epoch to determine UT is adequate (ignoring leap seconds, but not leap days). More details on the conversion from GPS week to UT can be found in Refs. [1] and [5]. With the known UT reference the ECEF position vector can be determined by first computing θ using eqn. (7.3). Then, the ECEF position vector is computed using eqn. (7.1). Another, more direct method to determine the ECEF position vector involves using the ephemeris parameters, which are broadcasted by the satellites and are available from the receiver.[6]

As previously mentioned the ECEF position vector is useful since this gives a simple approach to determine the longitude and latitude of a user. The Earth's geoid can be approximated by an ellipsoid of revolution about its minor axis. A common

*The U.S. Coast Guard Navigation Center maintains a website that contains GPS almanacs, and as of this writing this website is given by http://www.navcen.uscg.gov/.

ellipsoid model is given by the World Geodetic System 1984 model (WGS-84), with semimajor axis $a = 6,378,137.0$ m and semiminor axis $b = 6,356,752.3142$ m. The eccentricity of this ellipsoid is given by $e = 0.0818$. The geodetic coordinates are given by the longitude ϕ, latitude λ, and height h. To determine the ECEF position vector, the length of the normal to the ellipsoid is first computed, given by[6]

$$\boxed{N = \frac{a}{\sqrt{1 - e^2 \sin^2 \lambda}}} \tag{7.4}$$

Then, given the observer geodetic quantities ϕ, λ, and h, the observer ECEF position coordinates are computed using

$$\boxed{\begin{aligned} x &= (N+h)\cos\lambda\cos\phi && (7.5a)\\ y &= (N+h)\cos\lambda\sin\phi && (7.5b)\\ z &= [N(1-e^2)+h]\sin\lambda && (7.5c) \end{aligned}}$$

The conversion from ECEF to geodetic coordinates is not that straightforward. A complicated closed-form solution is given in Ref. [6], but a good approximation up to low Earth orbit is given by[1]

$$\boxed{\begin{aligned} p &= \sqrt{x^2 + y^2} && (7.6a)\\ \psi &= \operatorname{atan}\left(\frac{z\,a}{p\,b}\right) && (7.6b)\\ \bar{e}^2 &= \frac{a^2 - b^2}{b^2} && (7.6c)\\ \lambda &= \operatorname{atan}\left(\frac{z + \bar{e}^2\, b \sin^3\psi}{p - e^2\, a \cos^3\psi}\right) && (7.6d)\\ \phi &= \operatorname{atan2}(y, x) && (7.6e)\\ h &= \frac{p}{\cos\lambda} - N && (7.6f) \end{aligned}}$$

where N is given by eqn. (7.4) and atan2 is a four quadrant inverse tangent function.

Another useful quantity is the availability of a particular GPS satellite at a given observer longitude and latitude. The solution to this problem involves computing the "up" vector shown in Figure 7.2, which is given by

$$\boxed{\mathbf{u} = \begin{bmatrix} \cos\lambda\cos\phi \\ \cos\lambda\sin\phi \\ \sin\lambda \end{bmatrix}} \tag{7.7}$$

The zenith angle, ξ, for the i^{th} GPS satellite is given by

$$\boxed{\cos\xi_i = \boldsymbol{\rho}_i^T \mathbf{u}} \tag{7.8}$$

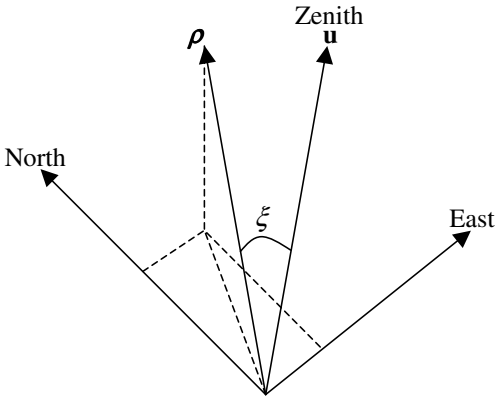

Figure 7.2: Definition of the Zenith Angle

From the assumed observer longitude and latitude of the user, its ECEF coordinates can be computed using eqn. (7.5), which is now defined by the vector $\mathbf{r} \equiv \begin{bmatrix} x & y & z \end{bmatrix}^T$. The ECEF coordinates of the i^{th} GPS satellite, denoted by $\mathbf{e}_i$, is computed using the ECI to ECEF coordinate transformation shown by eqn. (7.1). Next, the following vector is computed:

$$\boxed{\boldsymbol{\rho}_i = \frac{\mathbf{e}_i - \mathbf{r}}{||\mathbf{e}_i - \mathbf{r}||}} \tag{7.9}$$

Note that the pseudorange is given by $||\mathbf{e}_i - \mathbf{r}||$ plus the clock bias. Then, the following vertical (elevation) angle is computed:

$$\boxed{E_i = 90^\circ - \xi_i} \tag{7.10}$$

An adequate elevation cutoff for an observer on the Earth is given by 15 degrees.[1] Therefore, the i^{th} GPS satellite is available if $E_i > 15^\circ$ is satisfied.

7.1.2 Extended Kalman Filter Application to GPS

In this section an application of the EKF is shown for GPS position and velocity estimation. In this formulation we assume that an Inertial Navigation System (INS), including gyroscopes and/or accelerometers, is not present. More details on a combined GPS/INS integrated architecture can be found in Ref. [7]. In our formulation we assume a random walk process for the acceleration of the vehicle and derivative of the clock bias, so that the truth model is given by

$$\ddot{\mathbf{r}}(t) = \mathbf{w}_r(t) \tag{7.11a}$$

$$\dot{\tau}(t) = w_\tau(t) \tag{7.11b}$$

where $\mathbf{r} \equiv \begin{bmatrix} x & y & z \end{bmatrix}^T$ is a 3×1 vector of the vehicle position in ECEF coordinates, and τ is the GPS clock bias defined in §4.1. The state-space model of eqn. (7.11) is

given by

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & 0_{1\times3} & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0_{3\times4} \\ I_{4\times4} \end{bmatrix} \mathbf{w}(t) \tag{7.12}$$

where $\mathbf{x} \equiv \begin{bmatrix} \mathbf{r}^T & \dot{\mathbf{r}}^T & \tau \end{bmatrix}^T$ and the process noise is defined by $\mathbf{w} \equiv \begin{bmatrix} \mathbf{w}_r^T & w_\tau \end{bmatrix}^T$. The discrete-time version of the system shown in eqn. (7.12) is given by

$$\mathbf{x}_{k+1} = \begin{bmatrix} I_{3\times3} & \Delta t I_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & I_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & 0_{1\times3} & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \Delta t^2/2 I_{3\times3} & 0_{3\times1} \\ \Delta t I_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & \Delta t \end{bmatrix} \mathbf{w}_k \tag{7.13}$$

The i^{th} row of the output vector is the pseudorange, given by eqn. (4.2). Therefore, matrix $H(\hat{\mathbf{x}})$ used in the EKF (see Table 5.9) is given by

$$H(\hat{\mathbf{x}}) = \begin{bmatrix} \frac{\partial\hat{\rho}_1}{\partial\hat{x}} & \frac{\partial\hat{\rho}_1}{\partial\hat{y}} & \frac{\partial\hat{\rho}_1}{\partial\hat{z}} & 0 & 0 & 0 & 1 \\ \frac{\partial\hat{\rho}_2}{\partial\hat{x}} & \frac{\partial\hat{\rho}_2}{\partial\hat{y}} & \frac{\partial\hat{\rho}_2}{\partial\hat{z}} & 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial\hat{\rho}_n}{\partial\hat{x}} & \frac{\partial\hat{\rho}_n}{\partial\hat{y}} & \frac{\partial\hat{\rho}_n}{\partial\hat{z}} & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.14}$$

where the partials are given by eqn. (4.4).

The procedure to simulate the application of the EKF to GPS position determination is as follows. First, from the known orbital elements of the GPS satellites, determine the ECI position and velocity coordinates using eqns. (3.211) and (3.212). The GPS ECI vectors can then be propagated forward in time using eqn. (3.202). Then, using a chosen time reference, the GPS ECI position vectors are converted to ECEF coordinates using eqns. (7.1) and (7.3). Next, convert the chosen longitude, latitude, and height of the vehicle position on the Earth to ECEF coordinates using eqn. (7.5). Then, check the availability of each GPS satellite using the 15° cutoff in the elevation angles computed in eqn. (7.10). If the i^{th} satellite is available, then compute a pseudorange measurement using eqn. (4.1). Next, compute pseudorange measurements at different time steps using the aforementioned procedure with a new vehicle location due to its movement on the Earth. Finally, pick a value for the process noise covariance and run the EKF to determine the vehicle's position and velocity. Typically, $\mathbf{w}_r$ and w_τ are uncorrelated, and the variance of w_τ is very well known, which is derived from the clock stability. As a first pass consider the covariance of $\mathbf{w}_r$ to be given by a scalar times the identity matrix, where the scalar is used as a tuning parameter.

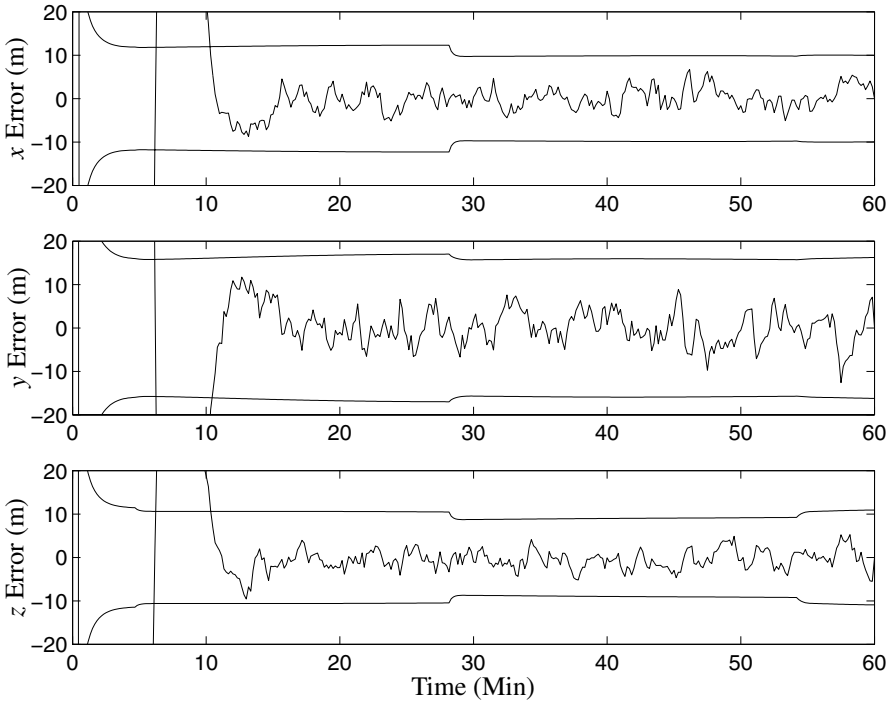

Figure 7.3: Vehicle Position Error Estimates

Example 7.1: In this example the extended Kalman filter is employed to determine the position and velocity of a vehicle on the Earth from GPS pseudorange measurements. The initial parameters of the ground vehicle are identical to example 4.1, where the vehicle is assumed to initially have coordinates of 38°N and 77°W (i.e., in Washington, DC). For simulation purposes the vehicle moves in the south direction at a rate of 100 km/hr. The variance of the continuous-time clock-bias process noise is assumed to be 200, which is used to generate the clock drift over time using eqn. (7.11b). The simulation runs over a 60-minute interval with a sampling interval of 10 seconds. The initial conditions for the EKF are given by $\hat{\mathbf{x}}_0 = \mathbf{0}$, which tests the convergence properties of the filter.

The continuous-time process noise covariance for the position error-process, $\mathbf{w}_r(t)$, shown in eqn. (7.11) is given by a scalar times a 3×3 identity matrix. The simulation uses only the true position movement of the vehicle (i.e., due south motion) to create the actual position. The scalar parameter in the process noise is only used in the EKF as a tuning factor to produce filtered estimates. After some trial and error a value of 1×10^{-5} for this tuning parameter is found to produce adequate results. Note that this value may be different for various vehicle motions. The adaptive methods of §5.7.4 may be employed to determine the optimal value if needed.

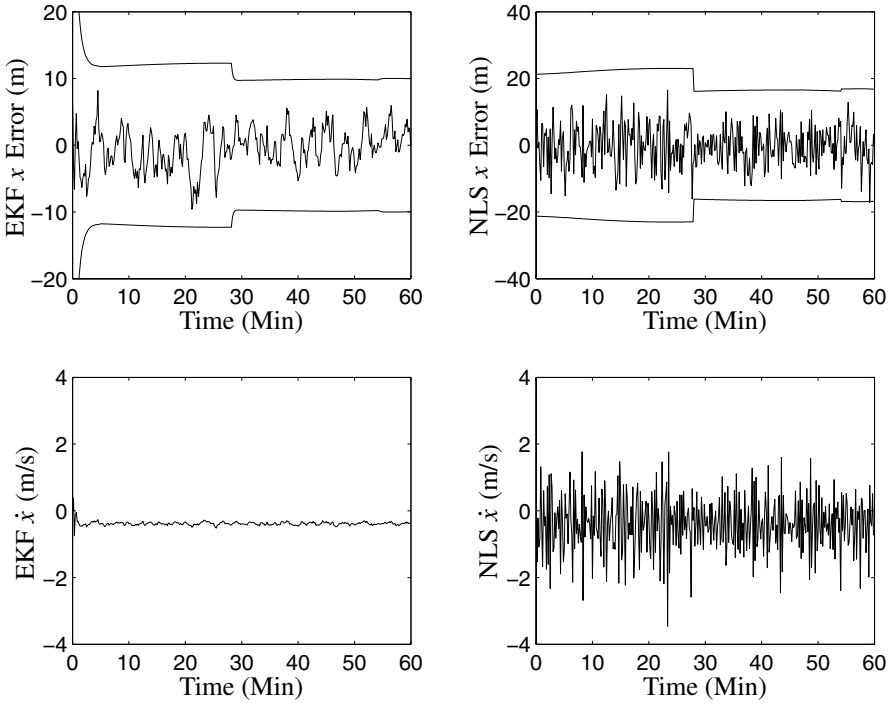

Figure 7.4: EKF and Nonlinear Least Squares Comparison

A plot of the EKF position error estimates with 3σ bounds is shown in Figure 7.3. The EKF converges in about 10 minutes, and the EKF accuracy in position is about 15 meters. Note that the 3σ bounds become smaller just before 30 minutes. This is due to the addition of another available GPS satellite in the vehicle's view (for this simulation the number of available satellites is 7 at the beginning of the simulation time). This result is intuitively correct, although the accuracy also depends on the GPS satellite geometry at the time of interest. A comparison of the EKF accuracy versus the nonlinear least squares deterministic solution of §4.1 is shown in Figure 7.4. For this simulation run the EKF estimates are initialized using the nonlinear least squares solution. The velocity estimates using the nonlinear least squares approach are determined by a finite difference of the determined position. The position estimates show a modest performance increase using the EKF; however, the true power of the EKF is shown in the velocity estimates, which are much smoother than the finite difference approach in nonlinear least squares. Similar results are obtained for the other axes. This example clearly shows how a Kalman filter can be employed to provide better performance capabilities using GPS pseudorange measurements to determine the position and velocity of a vehicle.

7.2 Attitude Estimation

In this section an extended Kalman filter is used to sequentially estimate the attitude and rate of a vehicle with attitude sensor measurements and three-axis strapdown gyroscopes. Several parameterizations can be used to represent the attitude, such as Euler angles,[8] quaternions,[9] modified Rodrigues parameters,[10] and even the rotation vector.[11] Quaternions are especially appealing since no singularities are present and the kinematics equation is bilinear. However, the quaternion must obey a normalization constraint, which can be violated by the linear measurement-updates associated with the standard EKF approach. The most common approach to overcome this shortfall involves using a multiplicative error quaternion, where after neglecting higher-order terms, the four-component quaternion can effectively be replaced by a three-component error vector.[9] Under ideal circumstances, such as small attitude errors, this approach works extremely well. Also, a useful variation to this filter is shown, which processes a single vector measurement at one time. This approach substantially reduces the computational burden.

7.2.1 Multiplicative Quaternion Formulation

The extended Kalman filter for attitude estimation begins with the quaternion kinematics model, shown previously in §3.7.1 as

$$\dot{\mathbf{q}} = \frac{1}{2}\Xi(\mathbf{q})\boldsymbol{\omega} = \frac{1}{2}\Omega(\boldsymbol{\omega})\mathbf{q} \tag{7.15}$$

The quaternion, $\mathbf{q} \equiv \begin{bmatrix}\boldsymbol{\varrho}^T & q_4\end{bmatrix}^T$, must obey a normalization constraint given by $\mathbf{q}^T\mathbf{q} = 1$. The most straightforward method for the filter design is to use eqn. (7.15) directly in the extended Kalman filter of Table 5.9; however, this "additive" approach can destroy normalization. This is clearly seen by example. Consider a true quaternion of $\mathbf{q} = [0 \;\; 0 \;\; \sqrt{0.001} \;\; \sqrt{0.999}]^T$, and assume that the estimated quaternion is given by $\hat{\mathbf{q}} = [0 \;\; 0 \;\; 0 \;\; 1]^T$. The additive error quaternion is given by the difference $\hat{\mathbf{q}} - \mathbf{q} = [0 \;\; 0 \;\; -\sqrt{0.001} \;\; 1 - \sqrt{0.999}]^T$, which clearly is not close to being a unit vector. This can cause significant difficulties during the filtering process. A more physical (true to nature) approach involves using a multiplicative error quaternion in the body frame, given by

$$\delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \tag{7.16}$$

with $\delta\mathbf{q} \equiv \begin{bmatrix}\delta\boldsymbol{\varrho}^T & \delta q_4\end{bmatrix}^T$. Also, the quaternion inverse is defined by eqn. (3.169). Taking the time derivative of eqn. (7.16) gives

$$\delta\dot{\mathbf{q}} = \dot{\mathbf{q}} \otimes \hat{\mathbf{q}}^{-1} + \mathbf{q} \otimes \dot{\hat{\mathbf{q}}}^{-1} \tag{7.17}$$

We now need to determine an expression for $\dot{\hat{\mathbf{q}}}^{-1}$. The estimated quaternion kinematics model follows

$$\boxed{\dot{\hat{\mathbf{q}}} = \frac{1}{2}\Xi(\hat{\mathbf{q}})\hat{\boldsymbol{\omega}} = \frac{1}{2}\Omega(\hat{\boldsymbol{\omega}})\hat{\mathbf{q}}} \tag{7.18}$$

Taking the time derivative of $\hat{\mathbf{q}} \otimes \hat{\mathbf{q}}^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$ gives

$$\dot{\hat{\mathbf{q}}} \otimes \hat{\mathbf{q}}^{-1} + \hat{\mathbf{q}} \otimes \dot{\hat{\mathbf{q}}}^{-1} = \mathbf{0} \tag{7.19}$$

Substituting eqn. (7.18) into eqn. (7.19) gives

$$\frac{1}{2}\Omega(\hat{\boldsymbol{\omega}})\hat{\mathbf{q}} \otimes \hat{\mathbf{q}}^{-1} + \hat{\mathbf{q}} \otimes \dot{\hat{\mathbf{q}}}^{-1} = \mathbf{0} \tag{7.20}$$

Since $\hat{\mathbf{q}} \otimes \hat{\mathbf{q}}^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$, and using the definition of $\Omega(\hat{\boldsymbol{\omega}})$ in eqn. (3.162), then eqn. (7.20) reduces down to

$$\frac{1}{2}\begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} + \hat{\mathbf{q}} \otimes \dot{\hat{\mathbf{q}}}^{-1} = \mathbf{0} \tag{7.21}$$

Solving eqn. (7.21) for $\dot{\hat{\mathbf{q}}}^{-1}$ yields

$$\dot{\hat{\mathbf{q}}}^{-1} = -\frac{1}{2}\hat{\mathbf{q}}^{-1} \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \tag{7.22}$$

Also, a useful identity is given by

$$\dot{\mathbf{q}} = \frac{1}{2}\Omega(\boldsymbol{\omega})\mathbf{q} = \frac{1}{2}\begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \otimes \mathbf{q} \tag{7.23}$$

This identity can easily be verified using the definitions of $\Omega(\boldsymbol{\omega})$ in eqn. (3.162) and quaternion multiplication in eqn. (3.168). Substituting eqns. (7.22) and (7.23) into eqn. (7.17), and using the definition of the error quaternion in eqn. (7.16) gives

$$\delta\dot{\mathbf{q}} = \frac{1}{2}\left\{ \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \otimes \delta\mathbf{q} - \delta\mathbf{q} \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \right\} \tag{7.24}$$

We now define the following error angular velocity: $\delta\boldsymbol{\omega} \equiv \boldsymbol{\omega} - \hat{\boldsymbol{\omega}}$. Substituting $\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega}$ into eqn. (7.24) leads to

$$\delta\dot{\mathbf{q}} = \frac{1}{2}\left\{ \begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \otimes \delta\mathbf{q} - \delta\mathbf{q} \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \right\} + \frac{1}{2}\begin{bmatrix} \delta\boldsymbol{\omega} \\ 0 \end{bmatrix} \otimes \delta\mathbf{q} \tag{7.25}$$

Next, consider the following helpful identities:

$$\begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \otimes \delta\mathbf{q} = \Omega(\hat{\boldsymbol{\omega}})\delta\mathbf{q} \tag{7.26a}$$

$$\delta\mathbf{q} \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} = \Gamma(\hat{\boldsymbol{\omega}})\delta\mathbf{q} \tag{7.26b}$$

where $\Gamma(\hat{\boldsymbol{\omega}})$ is given by eqn. (3.165). Substituting eqn. (7.26) into eqn. (7.25), and after some algebraic manipulations (which are left as an exercise for the reader), leads to

$$\delta\dot{\mathbf{q}} = -\begin{bmatrix}[\hat{\boldsymbol{\omega}}\times]\delta\boldsymbol{\varrho}\\ 0\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\delta\boldsymbol{\omega}\\ 0\end{bmatrix}\otimes\delta\mathbf{q} \tag{7.27}$$

where the cross product matrix $[\hat{\boldsymbol{\omega}}\times]$ is defined by eqn. (3.149). Note that eqn. (7.27) is an exact relationship since no linearizations have been performed yet. The nonlinear term is present only in the last term on the right-hand side of eqn. (7.27). Its first-order approximation is given by[9]

$$\frac{1}{2}\begin{bmatrix}\delta\boldsymbol{\omega}\\ 0\end{bmatrix}\otimes\delta\mathbf{q} \approx \frac{1}{2}\begin{bmatrix}\delta\boldsymbol{\omega}\\ 0\end{bmatrix} \tag{7.28}$$

Substituting eqn. (7.28) into eqn. (7.27) leads to the following linearized model:

$$\delta\dot{\boldsymbol{\varrho}} = -[\hat{\boldsymbol{\omega}}\times]\delta\boldsymbol{\varrho} + \frac{1}{2}\delta\boldsymbol{\omega} \tag{7.29a}$$

$$\delta\dot{q}_4 = 0 \tag{7.29b}$$

Note that the fourth error-quaternion component is constant. The first-order approximation, which assumes that the true quaternion is "close" to the estimated quaternion, gives $\delta q_4 \approx 1$. This allows us to reduce the order of the system in the EKF by one state. The linearization using eqn. (7.16) maintains quaternion normalization to within first-order if the estimated quaternion is "close" to the true quaternion, which is within the first-order approximation in the EKF.

A common sensor that measures the angular rate is a rate-integrating gyro. For this sensor, a widely used model is given by[12]

$$\boldsymbol{\omega} = \tilde{\boldsymbol{\omega}} - \boldsymbol{\beta} - \boldsymbol{\eta}_v \tag{7.30a}$$

$$\dot{\boldsymbol{\beta}} = \boldsymbol{\eta}_u \tag{7.30b}$$

where $\boldsymbol{\eta}_v$ and $\boldsymbol{\eta}_u$ are zero-mean Gaussian white-noise processes with covariances usually given by $\sigma_v^2 I_{3\times3}$ and $\sigma_u^2 I_{3\times3}$, respectively, $\boldsymbol{\beta}$ is a bias vector, and $\tilde{\boldsymbol{\omega}}$ is the measured observation. The estimated angular velocity is given by

$$\hat{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}} \tag{7.31}$$

Also, the estimated bias differential equation follows

$$\dot{\hat{\boldsymbol{\beta}}} = \mathbf{0} \tag{7.32}$$

Substituting eqns. (7.30a) and (7.31) into $\delta\boldsymbol{\omega} \equiv \boldsymbol{\omega} - \hat{\boldsymbol{\omega}}$ gives

$$\delta\boldsymbol{\omega} = -(\Delta\boldsymbol{\beta} + \boldsymbol{\eta}_v) \tag{7.33}$$

where $\Delta\boldsymbol{\beta} \equiv \boldsymbol{\beta} - \hat{\boldsymbol{\beta}}$. Substituting eqn. (7.33) into eqn. (7.29a) gives

$$\delta\dot{\boldsymbol{\varrho}} = -[\hat{\boldsymbol{\omega}}\times]\delta\boldsymbol{\varrho} - \frac{1}{2}(\Delta\boldsymbol{\beta} + \boldsymbol{\eta}_v) \tag{7.34}$$

A common simplification, which is discussed in §3.7.1, is given by the small angle approximation $\delta\boldsymbol{\varrho} \approx \delta\boldsymbol{\alpha}/2$, where $\delta\boldsymbol{\alpha}$ has components of roll, pitch, and yaw error-angles for any rotation sequence. Using this simplification in eqn. (7.34) gives

$$\delta\dot{\boldsymbol{\alpha}} = -[\hat{\boldsymbol{\omega}}\times]\delta\boldsymbol{\alpha} - (\Delta\boldsymbol{\beta} + \boldsymbol{\eta}_v) \tag{7.35}$$

This approach minimizes the use of factors of $1/2$ and 2 in the EKF, and also gives a direct physical meaning to the state error-covariance, which can be used to directly determine the 3σ bounds of the actual attitude errors. The EKF error model is now given by

$$\Delta\dot{\tilde{\mathbf{x}}}(t) = F(\hat{\mathbf{x}}(t),\, t)\,\Delta\tilde{\mathbf{x}}(t) + G(t)\,\mathbf{w}(t) \tag{7.36}$$

where $\Delta\tilde{\mathbf{x}}(t) \equiv \left[\delta\boldsymbol{\alpha}^T(t)\ \Delta\boldsymbol{\beta}^T(t)\right]^T$, $\mathbf{w}(t) \equiv \left[\boldsymbol{\eta}_v^T(t)\ \boldsymbol{\eta}_u^T(t)\right]^T$, and $F(\hat{\mathbf{x}}(t),\, t)$, $G(t)$, and $Q(t)$ are given by

$$F(\hat{\mathbf{x}}(t),\, t) = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}(t)\times] & -I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{7.37a}$$

$$G(t) = \begin{bmatrix} -I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{7.37b}$$

$$Q(t) = \begin{bmatrix} \sigma_v^2 I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \sigma_u^2 I_{3\times3} \end{bmatrix} \tag{7.37c}$$

Note that these matrices are 6×6 matrices now, since the order of the system has been reduced by one state.

Our next step involves the determination of the sensitivity matrix $H_k(\hat{\mathbf{x}}_k^-)$ used in the EKF. Discrete-time attitude observations for a single sensor are given by eqn. (4.11). Multiple, n, vector measurements can be concatenated to form

$$\tilde{\mathbf{y}}_k = \left.\begin{bmatrix} A(\mathbf{q})\mathbf{r}_1 \\ A(\mathbf{q})\mathbf{r}_2 \\ \vdots \\ A(\mathbf{q})\mathbf{r}_n \end{bmatrix}\right|_{t_k} + \left.\begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \\ \vdots \\ \boldsymbol{\nu}_n \end{bmatrix}\right|_{t_k} \equiv \mathbf{h}_k(\hat{\mathbf{x}}_k) + \mathbf{v}_k \tag{7.38a}$$

$$R = \mathrm{diag}\left[\sigma_1^2 I_{3\times3}\ \sigma_2^2 I_{3\times3}\ \ldots\ \sigma_n^2 I_{3\times3}\right] \tag{7.38b}$$

where diag denotes a diagonal matrix of appropriate dimension. The actual attitude matrix, $A(\mathbf{q})$, is related to the propagated attitude, $A(\delta\mathbf{q})$, through

$$A(\mathbf{q}) = A(\delta\mathbf{q})A(\hat{\mathbf{q}}^-) \tag{7.39}$$

The first-order approximation of the error-attitude matrix is given by (see §3.7.1)

$$A(\delta\mathbf{q}) \approx I_{3\times3} - [\delta\boldsymbol{\alpha}\times] \tag{7.40}$$

where $\boldsymbol{\delta\alpha}$ is again the small angle approximation. For a single sensor the true and estimated body vectors are given by

$$\mathbf{b} = A(\mathbf{q})\mathbf{r} \tag{7.41a}$$

$$\hat{\mathbf{b}}^- = A(\hat{\mathbf{q}}^-)\mathbf{r} \tag{7.41b}$$

Substituting eqns. (7.39) and (7.40) into eqn. (7.41) yields

$$\Delta\mathbf{b} = [A(\hat{\mathbf{q}}^-)\mathbf{r}\times]\boldsymbol{\delta\alpha} \tag{7.42}$$

where $\Delta\mathbf{b} \equiv \mathbf{b} - \hat{\mathbf{b}}^-$. The sensitivity matrix for all measurement sets is therefore given by

$$\boxed{H_k(\hat{\mathbf{x}}_k^-) = \left[\begin{matrix} [A(\hat{\mathbf{q}}^-)\mathbf{r}_1\times] & 0_{3\times 3} \\ [A(\hat{\mathbf{q}}^-)\mathbf{r}_2\times] & 0_{3\times 3} \\ \vdots & \vdots \\ [A(\hat{\mathbf{q}}^-)\mathbf{r}_n\times] & 0_{3\times 3} \end{matrix}\right]_{t_k}} \tag{7.43}$$

Note that the number of columns of $H_k(\hat{\mathbf{x}}_k^-)$ is six, which is the dimension of the reduced-order state.

The final part in the EKF involves the quaternion and bias updates. The error-state update follows

$$\boxed{\Delta\hat{\hat{\mathbf{x}}}_k^+ = K_k[\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)]} \tag{7.44}$$

where $\Delta\hat{\hat{\mathbf{x}}}_k^+ \equiv \left[\boldsymbol{\delta}\hat{\boldsymbol{\alpha}}_k^{+T} \;\; \Delta\hat{\boldsymbol{\beta}}_k^{+T}\right]^T$, $\tilde{\mathbf{y}}_k$ is the measurement output, and $\mathbf{h}_k(\hat{\mathbf{x}}_k^-)$ is the estimate output, given by

$$\boxed{\mathbf{h}_k(\hat{\mathbf{x}}_k^-) = \left[\begin{matrix} A(\hat{\mathbf{q}}^-)\mathbf{r}_1 \\ A(\hat{\mathbf{q}}^-)\mathbf{r}_2 \\ \vdots \\ A(\hat{\mathbf{q}}^-)\mathbf{r}_n \end{matrix}\right]_{t_k}} \tag{7.45}$$

The gyro bias update is simply given by

$$\boxed{\hat{\boldsymbol{\beta}}_k^+ = \hat{\boldsymbol{\beta}}_k^- + \Delta\hat{\boldsymbol{\beta}}_k^+} \tag{7.46}$$

The quaternion update is more complicated. As previously mentioned the fourth component of $\boldsymbol{\delta}\mathbf{q}$ is nearly one. Therefore, to within first-order the quaternion update is given by

$$\hat{\mathbf{q}}_k^+ = \begin{bmatrix} \frac{1}{2}\boldsymbol{\delta}\hat{\boldsymbol{\alpha}}_k^+ \\ 1 \end{bmatrix} \otimes \hat{\mathbf{q}}_k^- \tag{7.47}$$

Table 7.1: Extended Kalman Filter for Attitude Estimation

Initialize	$\hat{\mathbf{q}}(t_0) = \hat{\mathbf{q}}_0, \quad \boldsymbol{\beta}(t_0) = \boldsymbol{\beta}_0$ $P(t_0) = P_0$
Gain	$K_k = P_k^- H_k^T(\hat{\mathbf{x}}_k^-)[H_k(\hat{\mathbf{x}}_k^-) P_k^- H_k^T(\hat{\mathbf{x}}_k^-) + R]^{-1}$ $H_k(\hat{\mathbf{x}}_k^-) = \left.\begin{bmatrix} [A(\hat{\mathbf{q}}^-)\mathbf{r}_1\times] & 0_{3\times 3} \\ \vdots & \vdots \\ [A(\hat{\mathbf{q}}^-)\mathbf{r}_n\times] & 0_{3\times 3} \end{bmatrix}\right\|_{t_k}$
Update	$P_k^+ = [I - K_k H_k(\hat{\mathbf{x}}_k^-)] P_k^-$ $\Delta\hat{\tilde{\mathbf{x}}}_k^+ = K_k[\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)]$ $\Delta\hat{\tilde{\mathbf{x}}}_k^+ \equiv \begin{bmatrix} \delta\hat{\boldsymbol{\alpha}}_k^{+T} & \Delta\hat{\boldsymbol{\beta}}_k^{+T} \end{bmatrix}^T$ $\mathbf{h}_k(\hat{\mathbf{x}}_k^-) = \left.\begin{bmatrix} A(\hat{\mathbf{q}}^-)\mathbf{r}_1 \\ A(\hat{\mathbf{q}}^-)\mathbf{r}_2 \\ \vdots \\ A(\hat{\mathbf{q}}^-)\mathbf{r}_n \end{bmatrix}\right\|_{t_k}$ $\hat{\mathbf{q}}_k^+ = \hat{\mathbf{q}}_k^- + \frac{1}{2}\Xi(\hat{\mathbf{q}}_k^-)\delta\hat{\boldsymbol{\alpha}}_k^+, \quad$ re-normalize quaternion $\hat{\boldsymbol{\beta}}_k^+ = \hat{\boldsymbol{\beta}}_k^- + \Delta\hat{\boldsymbol{\beta}}_k^+$
Propagation	$\hat{\boldsymbol{\omega}}(t) = \tilde{\boldsymbol{\omega}}(t) - \hat{\boldsymbol{\beta}}(t)$ $\dot{\hat{\mathbf{q}}}(t) = \frac{1}{2}\Xi\left(\hat{\mathbf{q}}(t)\right)\hat{\boldsymbol{\omega}}(t)$ $\dot{P}(t) = F(\hat{\mathbf{x}}(t), t)\, P(t) + P(t)\, F^T(\hat{\mathbf{x}}(t), t) + G(t)\, Q(t)\, G^T(t)$ $F(\hat{\mathbf{x}}(t), t) = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}(t)\times] & -I_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{bmatrix}, \quad G(t) = \begin{bmatrix} -I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & I_{3\times 3} \end{bmatrix}$

Note that the small angle approximation has been used to define the vector part of the error-quaternion. Using the quaternion multiplication rule of eqn. (3.168) in eqn. (7.47) gives

$$\boxed{\hat{\mathbf{q}}_k^+ = \hat{\mathbf{q}}_k^- + \frac{1}{2}\Xi(\hat{\mathbf{q}}_k^-)\delta\hat{\boldsymbol{\alpha}}_k^+} \tag{7.48}$$

This updated quaternion is a unit vector to within first-order; however, a brute-force normalization should be performed to insure $\hat{\mathbf{q}}_k^{+T}\hat{\mathbf{q}}_k^+ = 1$.

The attitude estimation algorithm is summarized in Table 7.1. The filter is first

initialized with a known state (the bias initial condition is usually assumed zero) and error-covariance matrix. The first three diagonal elements of the error-covariance matrix correspond to attitude errors. Then, the Kalman gain is computed using the measurement-error covariance R and sensitivity matrix in eqn. (7.43). The state error-covariance follows the standard EKF update, while the error-state update is computed using eqn. (7.44). The bias and quaternion updates are now given by eqns. (7.46) and (7.48). Also, the updated quaternion is re-normalized by brute force. Finally, the estimated angular velocity is used to propagate the quaternion kinematics model in eqn. (7.18) and standard error-covariance in the EKF. Note that the gyro bias propagation is constant as shown by eqn. (7.32).

7.2.2 Discrete-Time Attitude Estimation

The propagation of the state and covariance can be accomplished by using numerical integration techniques. However, in general, the gyro observations are sampled at a high rate (usually higher than or at least at the same rate as the vector attitude observations). Therefore, a discrete propagation is usually sufficient. Discrete propagation of the quaternion model in eqn. (7.18) can be derived by using a power series approach:[13]

$$\begin{aligned}\exp\left[\frac{1}{2}\Omega(\hat{\omega})\,t\right] &= \sum_{j=0}^{\infty}\frac{\left[\frac{1}{2}\Omega(\hat{\omega})\,t\right]^{j}}{j!}\\ &= \sum_{k=0}^{\infty}\left\{\frac{\left[\frac{1}{2}\Omega(\hat{\omega})\,t\right]^{2k}}{(2k)!}+\frac{\left[\frac{1}{2}\Omega(\hat{\omega})\,t\right]^{2k+1}}{(2k+1)!}\right\}\end{aligned} \tag{7.49}$$

Next, consider the following identities:

$$\Omega^{2k}(\hat{\omega}) = (-1)^{k}||\hat{\omega}||^{2k} I_{4\times 4} \tag{7.50a}$$

$$\Omega^{2k+1}(\hat{\omega}) = (-1)^{k}||\hat{\omega}||^{2k}\Omega(\hat{\omega}) \tag{7.50b}$$

Substituting eqn. (7.50) into eqn. (7.49) gives

$$\begin{aligned}\exp\left[\frac{1}{2}\Omega(\hat{\omega})\,t\right] = I_{4\times 4}&\sum_{k=0}^{\infty}\frac{(-1)^{k}\left(\frac{1}{2}||\hat{\omega}||\,t\right)^{2k}}{(2k)!}\\ &+||\hat{\omega}||^{-1}\Omega(\hat{\omega})\sum_{k=0}^{\infty}\frac{(-1)^{k}\left(\frac{1}{2}||\hat{\omega}||\,t\right)^{2k+1}}{(2k+1)!}\end{aligned} \tag{7.51}$$

Recognizing that the first series in eqn. (7.51) is the cosine function and that the second series in eqn. (7.51) is the sine function yields

$$\exp\left[\frac{1}{2}\Omega(\hat{\omega})\,t\right] = I_{4\times4}\cos\left(\frac{1}{2}||\hat{\omega}||\,t\right) + \Omega(\hat{\omega})\frac{\sin\left(\frac{1}{2}||\hat{\omega}||\,t\right)}{||\hat{\omega}||} \tag{7.52}$$

Hence, given post-update estimates $\hat{\omega}_k^+$ and $\hat{\mathbf{q}}_k^+$, the propagated quaternion is found using

$$\boxed{\hat{\mathbf{q}}_{k+1}^- = \bar{\Omega}(\hat{\omega}_k^+)\hat{\mathbf{q}}_k^+} \tag{7.53}$$

with

$$\bar{\Omega}(\hat{\omega}_k^+) \equiv \begin{bmatrix} \cos\left(\frac{1}{2}||\hat{\omega}_k^+||\,\Delta t\right) I_{3\times3} - \left[\hat{\psi}_k^+\times\right] & \hat{\psi}_k^+ \\ -\hat{\psi}_k^{+T} & \cos\left(\frac{1}{2}||\hat{\omega}_k^+||\,\Delta t\right) \end{bmatrix} \tag{7.54}$$

where

$$\hat{\psi}_k^+ \equiv \frac{\sin\left(\frac{1}{2}||\hat{\omega}_k^+||\,\Delta t\right)\hat{\omega}_k^+}{||\hat{\omega}_k^+||} \tag{7.55}$$

and Δt is the sampling interval in the gyro. In the standard EKF formulation, given a post-update estimate $\hat{\beta}_k^+$, the post-update angular velocity and propagated gyro bias follow

$$\hat{\omega}_k^+ = \tilde{\omega}_k - \hat{\beta}_k^+ \tag{7.56a}$$

$$\hat{\beta}_{k+1}^- = \hat{\beta}_k^+ \tag{7.56b}$$

Note that the propagated gyro-bias estimate is equal to the previous update, which is due to the propagation model in eqn. (7.32).

The discrete propagation of the covariance equation is given by

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + G_k\, Q_k\, G_k^T \tag{7.57}$$

where G_k is given by

$$G_k = \begin{bmatrix} -I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{7.58}$$

The discrete error-state transition matrix can also be derived using a power series

approach (which is left as an exercise for the reader):

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \tag{7.59a}$$

$$\Phi_{11} = I_{3\times 3} - [\hat{\omega}\times]\frac{\sin(||\hat{\omega}||\,\Delta t)}{||\hat{\omega}||} + [\hat{\omega}\times]^2\frac{\left\{1-\cos(||\hat{\omega}||\,\Delta t)\right\}}{||\hat{\omega}||^2} \tag{7.59b}$$

$$\begin{aligned}\Phi_{12} = {} & [\hat{\omega}\times]\frac{\left\{1-\cos(||\hat{\omega}||\,\Delta t)\right\}}{||\hat{\omega}||^2} - I_{3\times 3}\Delta t \\ & - [\hat{\omega}\times]^2\frac{\left\{||\hat{\omega}||\,\Delta t - \sin(||\hat{\omega}||\,\Delta t)\right\}}{||\hat{\omega}||^3}\end{aligned} \tag{7.59c}$$

$$\Phi_{21} = 0_{3\times 3} \tag{7.59d}$$

$$\Phi_{22} = I_{3\times 3} \tag{7.59e}$$

The discrete process noise covariance has already been derived in example 5.3, which is given by

$$Q_k = \begin{bmatrix} \left(\sigma_v^2\Delta t + \frac{1}{3}\sigma_u^2\Delta t^3\right) I_{3\times 3} & -\left(\frac{1}{2}\sigma_u^2\Delta t^2\right) I_{3\times 3} \\ -\left(\frac{1}{2}\sigma_u^2\Delta t^2\right) I_{3\times 3} & \left(\sigma_u^2\Delta t\right) I_{3\times 3} \end{bmatrix} \tag{7.60}$$

Therefore, the continuous-time propagations of eqns. (7.18), (7.32), and covariance propagation can be replaced by their discrete-time equivalents of eqns. (7.53), (7.56b), and (7.57), respectively. These discrete-time forms make the EKF especially suitable for on-board implementation.

7.2.3 Murrell's Version

The only problem for the filter shown in Table 7.1 occurs in the gain calculation, which requires an inverse of a $3n \times 3n$ matrix. In order to overcome this difficulty a variation to this filter can be used, based on an algorithm by Murrell.[14] Even though the extended Kalman filter involves nonlinear models, a linear update is still performed. Therefore, linear tools such as the principle of superposition (see §3.1) can still be used. Murrell's filter uses this principle to process one 3×1 vector observation at a time. A flow diagram of Murrell's approach is given in Figure 7.5. The first step involves propagating the quaternion, gyro bias, and error-covariance to the current observation time. Then, the attitude matrix is computed. The propagated state vector is now initialized to zero. Next, the error-covariance and state quantities are updated using a single vector observation. This procedure is continued (replacing the propagated error-covariance and state vector with the updated values) until all vector observations are processed. Finally, the updated values are used to propagate the error-covariance and state quantities to the next observation time. Therefore, this approach reduces taking an inverse of a $3n \times 3n$ matrix to taking an inverse of a 3×3 matrix n times, which can significantly decrease the computational load.

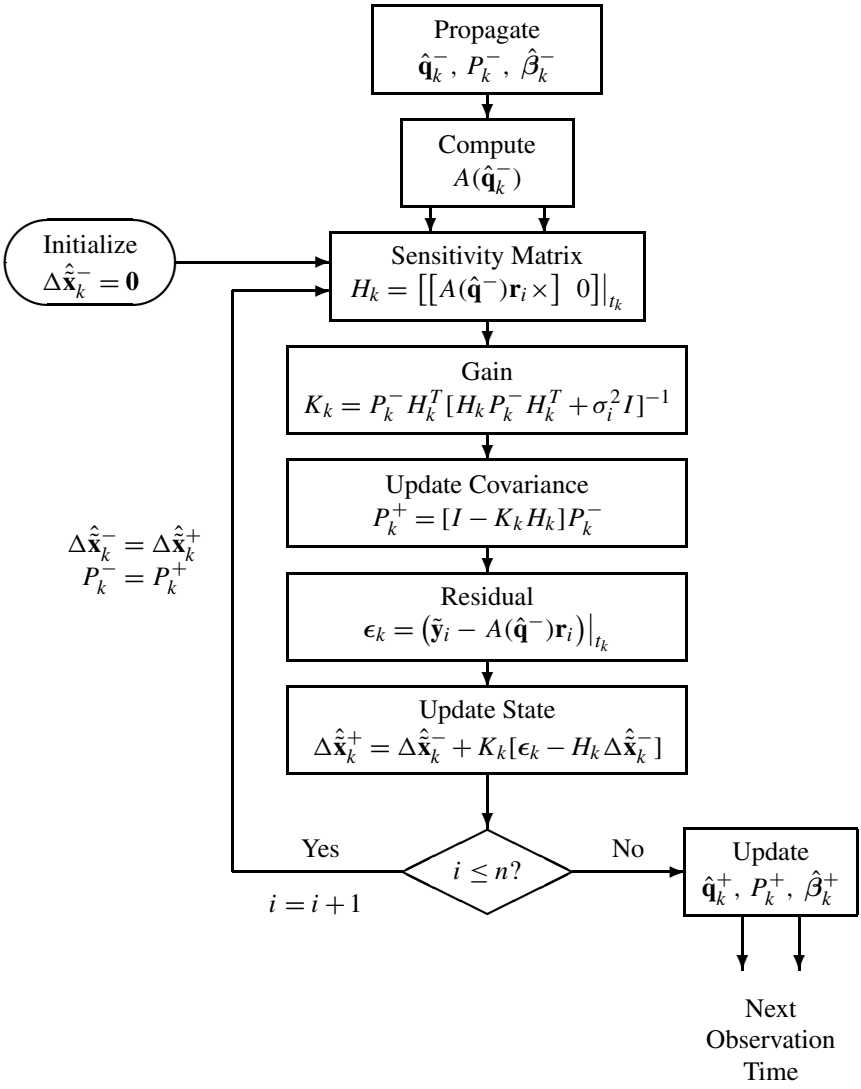

Figure 7.5: Computationally Efficient Attitude Estimation Algorithm

---

Example 7.2: In this example the extended Kalman filter algorithm shown in Table 7.1 is employed for attitude estimation using the simulation parameters shown by example 4.2. The attitude determination results of the deterministic approach (i.e., without using a filter) are shown in Figure 4.5. The goals of the EKF application involve the estimation of the gyro biases for all three axes and the filtering of the attitude star camera measurements. The standard deviation of the star camera

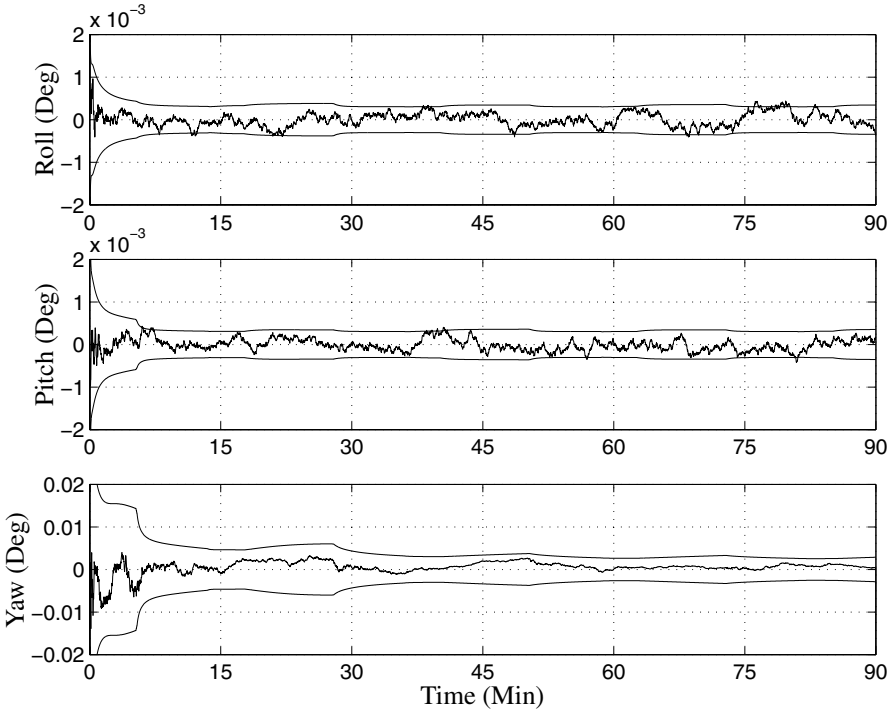

Figure 7.6: Attitude Errors and Boundaries

measurement error is the same as given in example 4.2. The noise parameters for the gyro measurements are given by $\sigma_u = \sqrt{10} \times 10^{-10}$ rad/sec$^{3/2}$ and $\sigma_v = \sqrt{10} \times 10^{-7}$ rad/sec$^{1/2}$. The initial bias for each axis is given by 0.1 deg/hr. Also, the gyro measurements are sampled at the same rate as the star camera measurements (i.e., at 1 Hz). We should note that in practice the gyros are sampled at a much higher frequency, which is usually required for jitter control. The initial covariance for the attitude error is set to 0.1^2 deg^2, and the initial covariance for the gyro drift is set to 0.2^2 (deg/hr)^2. Converting these quantities to radians gives the following initial attitude and gyro drift covariances for each axis: $P_0^a = 3.0462 \times 10^{-6}$ and $P_0^b = 9.4018 \times 10^{-13}$, so that the initial covariance is given by

$$P_0 = \text{diag}\left[P_0^a \;\; P_0^a \;\; P_0^a \;\; P_0^b \;\; P_0^b \;\; P_0^b\right]$$

The initial attitude condition for the EKF is given by the deterministic quaternion from example 4.2. The initial gyro bias conditions in the EKF are set to zero.

A plot of the attitude errors and associated 3σ boundaries is shown in Figure 7.6. Clearly, the computed 3σ boundaries do indeed bound the attitude errors. Comparing Figure 4.5 to Figure 7.6 shows a vast improvement (by an order of magnitude) in the attitude accuracy. This is due to the combination of the attitude measurements with

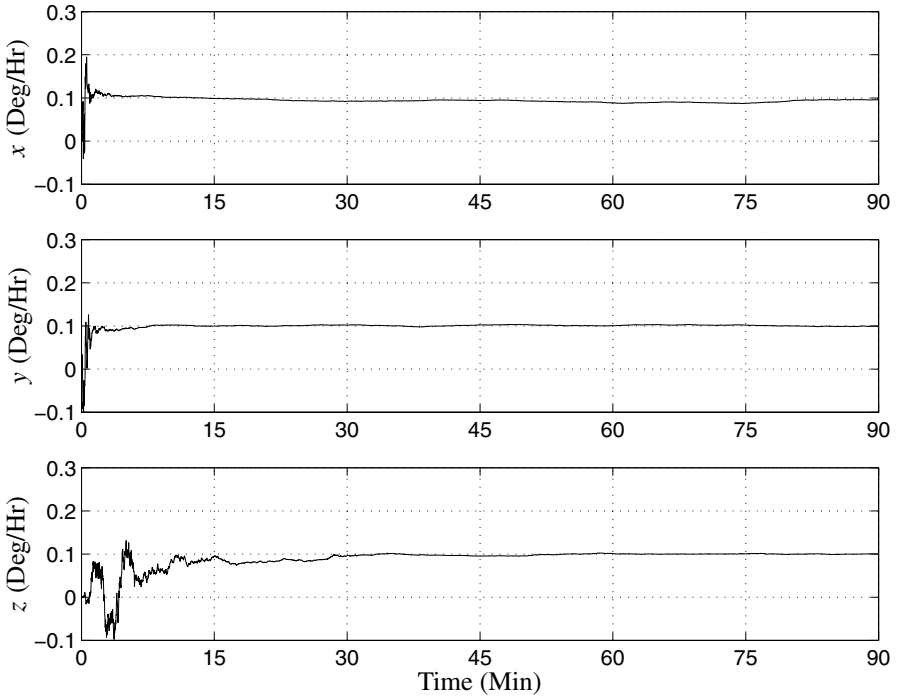

Figure 7.7: Gyro Drift Estimates

an accurate three-axis gyro. As with the deterministic solution the EKF results show that the yaw errors are much larger than the roll and pitch errors, which is intuitively correct. Also, the accuracy degrades as the number of available stars decreases, although this effect is not as pronounced with EKF results as with the deterministic results. This is due to the effect of filtering on the measurements. A plot of the gyro drift estimates is shown in Figure 7.7. The EKF is able to accurately estimate the initial bias errors. Also, the "drift" in this plot looks very steady, which is due to the fact that a high-grade three-axis gyro has been used in the simulation. A single axis analysis that can be used to access the performance of the EKF with various gyros will be shown in §7.2.4. This example clearly shows the power of the EKF for attitude estimation, which has been successfully applied to many spacecraft (e.g., see Ref. [15]). Another more robust approach to initial condition errors involves the application of the Unscented filter of §5.7.6, which may be found in Ref. [16].

7.2.4 Farrenkopf's Steady-State Analysis

The predicted performance of the attitude estimation can be found by checking the diagonal elements of the attitude error covariance. If a sensor is used to measure the integrated rates directly (i.e., assuming that the error angles can be decoupled) with standard deviation of the measurement error process given by σ_n, then a steady-state covariance given can be used. The model used for a single-axis analysis is shown in example 5.3, which is repeated here for completeness. The attitude rate $\dot{\theta}$ is assumed to be related to the gyro output $\tilde{\omega}$ by

$$\boxed{\dot{\theta} = \tilde{\omega} - \beta - \eta_v} \tag{7.61}$$

where β is the gyro drift, and η_v is a zero-mean Gaussian white-noise process with variance given by σ_v^2. The drift rate is modelled by a random walk process, given by

$$\boxed{\dot{\beta} = \eta_u} \tag{7.62}$$

where η_u is a zero-mean Gaussian white-noise process with variance given by σ_u^2. The state transition matrix and process noise covariance are shown in example 5.3. The discrete-time system used in the Kalman filter is given by

$$\mathbf{x}_{k+1} = \Phi\,\mathbf{x}_k + \Gamma\,\tilde{\omega}_k + \mathbf{w}_k \tag{7.63a}$$

$$\tilde{y}_k = H\,\mathbf{x}_k + v_k \tag{7.63b}$$

where $\mathbf{x} = \begin{bmatrix}\theta & \beta\end{bmatrix}^T$, $\Gamma = \begin{bmatrix}\Delta t & 0\end{bmatrix}^T$, $H = \begin{bmatrix}1 & 0\end{bmatrix}$, and $E\left\{\mathbf{w}_k \mathbf{w}_k^T\right\} = Q$. The matrices Q and Φ are given in example 5.3:

$$Q = \begin{bmatrix} \sigma_v^2 \Delta t + \frac{1}{3}\sigma_u^2 \Delta t^3 & -\frac{1}{2}\sigma_u^2 \Delta t^2 \\ -\frac{1}{2}\sigma_u^2 \Delta t^2 & \sigma_u^2 \Delta t \end{bmatrix} \tag{7.64a}$$

$$\Phi = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \tag{7.64b}$$

Using the model in eqn. (7.63) a solution to the resulting steady-state algebraic Riccati equation shown in Table 5.2 can be determined for the attitude and gyro drift estimate variances. Farrenkopf[12] obtained analytic solutions to the resulting Riccati equation. First, define the following the propagated and updated covariances:

$$P^- \equiv \begin{bmatrix} p_{\theta\theta}^- & p_{\theta\beta}^- \\ p_{\theta\beta}^- & p_{\beta\beta}^- \end{bmatrix}, \quad P^+ \equiv \begin{bmatrix} p_{\theta\theta}^+ & p_{\theta\beta}^+ \\ p_{\theta\beta}^+ & p_{\beta\beta}^+ \end{bmatrix} \tag{7.65}$$

Next, define the following variables:

$$\boxed{\begin{aligned} \xi &\equiv p_{\theta\beta}^- \Delta t \Big/ \sigma_n^2 && (7.66a)\\ S_u &\equiv \sigma_u \Delta t^{3/2} \Big/ \sigma_n && (7.66b)\\ S_v &\equiv \sigma_v \Delta t^{1/2} \Big/ \sigma_n && (7.66c) \end{aligned}}$$

Using the defined matrices in this section for Φ, Q, H, and $R = \sigma_n^2$, from the steady-state Riccati equation in Table 5.2 the following equation can be derived for ξ in terms of S_u and S_v (note, the procedure to determine this equation is outlined in §7.4.1):

$$\xi^4 + S_u^2\xi^3 + S_u^2\left[(S_u^2/6) - S_v^2 - 2\right]\xi^2 + S_u^4\xi + S_u^4 = 0 \tag{7.67}$$

This a quartic equation, but it can be simplified significantly since it is actually the product of two quadratic equations:

$$\xi^2 + \left[(S_u^2/2) \pm \vartheta\right]\xi + S_u^2 = 0 \tag{7.68}$$

where

$$\boxed{\vartheta = \left[S_u^2(4 + S_v^2) + S_u^4/12\right]^{1/2}} \tag{7.69}$$

The root of physical significance is the maximally negative root, assuming $+\vartheta$ in eqn. (7.68), so that

$$\boxed{\xi = -\frac{1}{2}\left[\left(\frac{S_u^2}{2} + \vartheta\right) + \sqrt{\left(\frac{S_u^2}{2} + \vartheta\right)^2 - 4S_u^2}\right]} \tag{7.70}$$

Then the solution for $p_{\theta\beta}^-$ is given using eqn. (7.66a). Once $p_{\theta\beta}^-$ is determined then the solutions for $p_{\theta\theta}^-$ and $p_{\beta\beta}^-$ are fairly straightforward (which are left as an exercise for the reader):

$$\boxed{p_{\theta\theta}^- = \sigma_n^2\left[\left(\frac{\xi}{S_u}\right)^2 - 1\right]} \tag{7.71a}$$

$$\boxed{p_{\beta\beta}^- = \left(\frac{\sigma_n}{\Delta t}\right)^2\left[S_u^2\left(\frac{1}{\xi} + \frac{1}{2}\right) - \xi\right]} \tag{7.71b}$$

The updated variances can be determined using the steady-state version of eqn. (5.44), which yields

$$\boxed{p_{\theta\theta}^+ = \sigma_n^2\left[1 - \left(\frac{S_u}{\xi}\right)^2\right]} \tag{7.72a}$$

$$\boxed{p_{\beta\beta}^+ = \left(\frac{\sigma_n}{\Delta t}\right)^2\left[S_u^2\left(\frac{1}{\xi} - \frac{1}{2}\right) - \xi\right]} \tag{7.72b}$$

Equations (7.71) and (7.72) can be used to determine 3σ bounds on the expected attitude and bias errors.

In the limiting case of very frequent updates, the pre-update and post-update attitude error standard deviations both approach the continuous-update limit, given by

$$\sqrt{p_{\theta\theta}^-} = \sqrt{p_{\theta\theta}^+} \equiv \sigma_c = \Delta t^{1/4}\sigma_n^{1/2}\left(\sigma_v^2 + 2\sigma_u\sigma_v\Delta t^{1/2}\right)^{1/4} \tag{7.73}$$

The even simpler limiting form when the contribution of σ_u to the attitude error is negligible is given by

$$\sigma_c = \Delta t^{1/4} \sigma_n^{1/2} \sigma_v^{1/2} \tag{7.74}$$

which indicates a one-half power dependence on both σ_n and σ_v, and a one-fourth power dependence on the update time Δt. This shows why it is extremely difficult to improve the attitude performance by simply increasing the update frequency. Farrenkopf's equations are useful for an initial estimate on attitude performance. Using the noise parameters from example 5.3 in eqn. (7.74) gives an approximate 3σ bound of 6.96 μrad for the attitude error, which is very close the actual solution of 7.18 μrad. Even though the observation model is not realistic, it can provide relative accuracies for various gyro parameters and sampling intervals. Converting 6.96 μrad to degrees gives 4×10^{-4} deg, which closely matches the roll and pitch errors of the results shown in Figure 7.6.

7.3 Orbit Estimation

In §4.3 a nonlinear least squares approach is shown to determine the initial state of an orbiting vehicle from range and line-of-sight (angle) observations. Another approach for orbit determination incorporates an *iterated Kalman filter*. This procedure uses the extended Kalman filter shown in Table 5.9 with $Q = 0$ to process the data forward with some initial condition guess, and then process the data backward to epoch. Initial conditions for the state are then given by previous pass results (e.g., the backward pass uses the final state from the forward pass for its initial condition). Also, the covariance must be reset after each forward or backward pass (this is required since no "new" information is given with each pass). The algorithm for orbit determination is essentially equivalent to the nonlinear fixed-point smoother in §6.1.3 with a covariance reset. The truth model used in the EKF is given by (see §3.8.2)

$$\boxed{\ddot{\mathbf{r}}(t) = -\frac{\mu}{||\mathbf{r}(t)||^3}\mathbf{r}(t) + \mathbf{w}(t)} \tag{7.75}$$

where $\mathbf{r}(t)$ is the orbital position and $\mathbf{w}(t)$ is the process noise, which is assumed to be zero. The discrete-time measurements include the azimuth, elevation, and range. The observation equations are given by eqn. (4.46). The goal of orbit determination is to determine initial conditions for the position and velocity of $\mathbf{x}_0 = \begin{bmatrix}\mathbf{r}_0^T & \dot{\mathbf{r}}_0^T\end{bmatrix}^T$ from the observations. The model equation is given by (7.75) with $\mathbf{x} = \begin{bmatrix}\mathbf{r}^T & \dot{\mathbf{r}}^T\end{bmatrix}^T$. Unlike the Gaussian Least Squares Differential Correction (GLSDC) shown in §4.3, the only analytical computations for the orbital EKF are the evaluations for the partial derivatives of eqns. (4.47) and (4.46) with respect to the state vector $\mathbf{x}$. These Jacobian, F, and sensitivity, H, matrix expressions are given by eqns. (4.50) and (4.60), respectively, which are evaluated at the current estimated state. Therefore,

Table 7.2: Extended Kalman Filter Iterations for Orbit Determination

Iteration	Position (km)			Velocity (km/sec)		
0	6,990	1	1	1	1	1
1	7,121	1,046	192	-0.07	5.70	1.67
2	7,000	1,000	200	4.00	7.00	2.00
3	7,000	1,000	200	4.00	7.00	2.00

the implementation of the EKF algorithm for orbit estimation at epoch is much more straightforward than the GLSDC.

Example 7.3: In this example the EKF algorithm is used to determine the orbit of a spacecraft from range, azimuth, and elevation measurements. The parameters used for the simulation are equivalent to the ones shown in example 4.3, but are repeated here for completeness. The true spacecraft position and velocity at epoch are given by

$$\mathbf{r}_0 = \begin{bmatrix} 7,000 & 1,000 & 200 \end{bmatrix}^T \text{ km}$$
$$\dot{\mathbf{r}}_0 = \begin{bmatrix} 4 & 7 & 2 \end{bmatrix}^T \text{ km/sec}$$

The latitude of the observer is given by $\lambda = 5°$, and the initial sidereal time is given by $\theta_0 = 10°$. Measurements are given at 10-second intervals over a 100-second simulation. The measurement errors are zero-mean Gaussian with a standard deviation of the range measurement error given by $\sigma_\rho = 1$ km, and a standard deviation of the angle measurements given by $\sigma_{az} = \sigma_{el} = 0.01°$.

A plot of a typical EKF iteration for the first position and velocity states is shown in Figure 7.8 (an iteration is one forward and one backward pass). The discontinuous jumps are due to the discrete-time measurement updates in the EKF. Note how these measurement updates help to reduce the error due to the propagation. Results for the EKF iterations are given in Table 7.2. Clearly, the EKF converges much faster than the least-square approach. This is due to the fact that the EKF uses a sequential process to update the estimates with each new measurement, while the GLSDC approach considers the entire batch of data to make a correction. The 3σ boundaries (determined using the diagonal elements of the estimate error-covariance) for position are $3\boldsymbol{\sigma}_{\mathbf{r}} = \begin{bmatrix} 1.26 & 0.25 & 0.51 \end{bmatrix}^T$ km, and for velocity are $3\boldsymbol{\sigma}_{\dot{\mathbf{r}}} = \begin{bmatrix} 0.020 & 0.008 & 0.006 \end{bmatrix}^T$ km/sec. The covariance results for the GLSDC in example 4.3 and EKF approaches are nearly identical, within the assumed applicability of linear error theory. The boundaries are useful to predict the performance of the algorithms.

The algorithm presented in this section uses a batch of data to determine the initial state of an orbit. The advantage of the Kalman filter approach is that the matrix $\Phi(t, t_0)$ used in the GLSDC is not required. The disadvantage of using a Kalman filter is that other quantities, such as biases, need to be appended into an augmented state vector. Another use of the Kalman filter involves the *navigation* problem that implements only a forward pass in the filter to determine the states in real time (typically with a nonzero value for Q), which can be used for control purposes. Modern-day navigation approaches predominately use GPS data to determine an orbit estimate, while differential GPS uses the on-board data with data collected from multiple ground stations. More details on orbit determination using GPS can be found in Ref. [17].

7.4 Target Tracking of Aircraft

One of the most useful early-day applications of the Kalman filter involves target tracking of aircraft from radar observations. Kalman filtering for target tracking has two main purposes. The first involves actual filtering of the radar measurements to obtain accurate range estimates. The second involves the estimation of velocity (and possibly acceleration). Velocity information is extremely important for air traffic control radar, which is used to avoid aircraft collisions when tracking multiple targets. Accurate velocity information can be used to predict ahead of time where multiple targets are expected in future radar scans in order to make a correct association of each target. A 3σ bound from the error covariance can be used to access the validity of the radar scan at future times.[18] This is used to ensure that the same target is actually tracked, thus avoiding incorrect target associations of multiple vehicles. In this section several tracking filters are introduced. The first two, called the α-β and α-β-γ filters, use kinematical models to derive the state estimate, which usually involves the aircraft's position and its derivatives. The third incorporates a dynamics-based model, which will be used to estimate the dynamical parameters of an aircraft from various observations, but can also be used to provide enhanced aircraft tracking capabilities.

7.4.1 The α-β Filter

One of the simplest target trackers is known as the α-β filter, which is used to estimate the position and velocity (usually range and range rate) of a vehicle. To derive this filter we begin with the following simple truth model in continuous-time:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t) \tag{7.76}$$

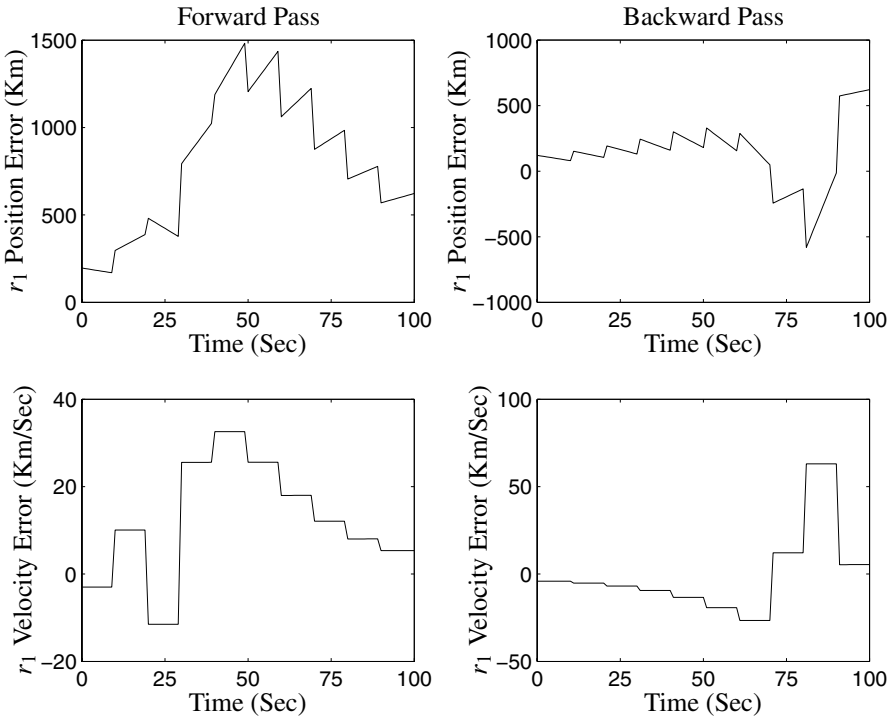

Figure 7.8: Extended Kalman Filter Iteration

where $w(t)$ is the process noise with variance q, and the states $\mathbf{x} \equiv \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ are position and velocity, denoted by r and $\dot{r}$, respectively. Note that the first state does not contain any process noise in this formulation. This is due to the fact that this state represents a kinematical relationship that is valid in theory and in the real-world, since velocity is always the derivative of position. Discrete-time measurements of position are assumed, so that

$$\tilde{y}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_k + v_k \equiv H \mathbf{x}_k + v_k \tag{7.77}$$

where v_k is the measurement noise, which is assumed to be modelled by a zero-mean Gaussian white-noise process with variance σ_n^2. The α-β filter uses a discrete-time model, which is easy to derive for the model in eqn. (7.76). The state transition matrix can be computed using eqn. (3.25). Since $F^2 = 0$ for the model in eqn. (7.76), then the discrete-time state matrix is given by

$$\Phi = I + \Delta t \, F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{7.78}$$

where Δt is the sampling interval.

Our next step in the derivation of the α-β filter involves the determination of the discrete-time process noise covariance. This can be accomplished using eqn. (5.135).

Performing a change of variables gives an equivalent integral for constant sampling with constant G and Q matrices:

$$\Upsilon Q \Upsilon^T = \int_0^{\Delta t} \Phi(\tau) G Q G^T \Phi^T(\tau)\, d\tau \tag{7.79}$$

where $G = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. Therefore, the discrete-time process noise covariance is given by

$$\Upsilon Q \Upsilon^T = q \int_0^{\Delta t} \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tau & 1 \end{bmatrix} d\tau \tag{7.80}$$

Evaluating the integral in eqn. (7.80) yields

$$\Upsilon Q \Upsilon^T = q \begin{bmatrix} \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t \end{bmatrix} \tag{7.81}$$

Notice, unlike the continuous-time process noise term given by $q\, G\, G^T$, the discrete-time process noise has nonzero values in all elements. This is due to the effect of sampling of a continuous-time process. However, if Δt is small, then eqn. (7.81) reduces down to eqn. (5.136).

Substituting the sensitivity and state matrices of eqns. (7.77) and (7.78) into the discrete-time Kalman update and propagation equations shown in Table 5.1 leads to

$$\hat{r}_k^+ = \hat{r}_k^- + \alpha\,[\tilde{y}_k - \hat{r}_k^-] \tag{7.82a}$$

$$\dot{\hat{r}}_k^+ = \dot{\hat{r}}_k^- + \frac{\beta}{\Delta t}\,[\tilde{y}_k - \hat{r}_k^-] \tag{7.82b}$$

$$\hat{r}_{k+1}^- = \hat{r}_k^+ + \dot{\hat{r}}_k^+ \Delta t \tag{7.82c}$$

$$\dot{\hat{r}}_{k+1}^- = \dot{\hat{r}}_k^+ \tag{7.82d}$$

where the gain matrix in Table 5.1 is given by $K_k = K \equiv \begin{bmatrix} \alpha & \beta/\Delta t \end{bmatrix}^T$. The gains α and β are often treated as tuning parameters to enhance the tracking performance. However, conventional wisdom tells us that tuning these gains individually is incorrect. To understand this concept we must remember that the model in eqn. (7.76) shows a kinematical relationship. If α and β are chosen separately, then this kinematical relationship can be lost. This means the velocity estimate may not truly be the derivative of the position estimate, even though we know that this relationship is exact. A more true-to-physics approach involves tuning the continuous-time process noise parameter q. From eqn. (7.81) changes in the velocity over the sampling interval are of the order $\sqrt{q\,\Delta t}$, which can be used as a guideline in the choice of q.[19] The complete solution involves the determination of the Kalman gain through the steady-state covariance solution shown by its equation in Table 5.2. Fortunately, the α-β filter is just a subset of the Farrenkopf steady-state analysis shown in §7.2.4.

First, define the following the propagated and updated covariances:

$$P^- \equiv \begin{bmatrix} p_{rr}^- & p_{r\dot{r}}^- \\ p_{r\dot{r}}^- & p_{\dot{r}\dot{r}}^- \end{bmatrix}, \quad P^+ \equiv \begin{bmatrix} p_{rr}^+ & p_{r\dot{r}}^+ \\ p_{r\dot{r}}^+ & p_{\dot{r}\dot{r}}^+ \end{bmatrix} \tag{7.83}$$

Also, define the following variable:

$$\boxed{S_q = q^{1/2}\,\Delta t^{3/2} \Big/ \sigma_n} \tag{7.84}$$

Now, determine the following parameter, ξ, which is related to $p_{r\dot{r}}^-$, using

$$\xi = \frac{1}{2}\left[\left(\frac{S_q^2}{2} + \vartheta\right) + \sqrt{\left(\frac{S_q^2}{2} + \vartheta\right)^2 - 4S_q^2}\right] \tag{7.85a}$$

$$\vartheta = \left[4S_q^2 + \frac{S_q^4}{12}\right]^{1/2} \tag{7.85b}$$

The pre-update variance parameters are then given by

$$p_{rr}^- = \sigma_n^2\left[\left(\frac{\xi}{S_q}\right)^2 - 1\right] \tag{7.86a}$$

$$p_{\dot{r}\dot{r}}^- = \left(\frac{\sigma_n}{\Delta t}\right)^2\left[S_q^2\left(\frac{1}{2} - \frac{1}{\xi}\right) + \xi\right] \tag{7.86b}$$

$$p_{r\dot{r}}^- = \frac{\sigma_n^2 \xi}{\Delta t} \tag{7.86c}$$

The Kalman gain and thus the parameters α and β can be determined by using the steady-state version of eqn. (5.42), which leads to

$$K \equiv \begin{bmatrix} \alpha \\ \beta/\Delta t \end{bmatrix} = \frac{1}{p_{rr}^- + \sigma_n^2}\begin{bmatrix} p_{rr}^- \\ p_{r\dot{r}}^- \end{bmatrix} \tag{7.87}$$

This clearly shows that α and β are closely related to one another.

To determine the relationship between α and β, we first will determine the relationship between p_{rr}^- and $p_{r\dot{r}}^-$. Substituting $\xi = \Delta t p_{r\dot{r}}^- / \sigma_n^2$ into eqn. (7.86a) and solving the resulting equation for $p_{r\dot{r}}^-$ yields

$$p_{r\dot{r}}^- = \frac{\sigma_n S_q}{\Delta t}\sqrt{p_{rr}^- + \sigma_n^2} \tag{7.88}$$

Next, solving for p_{rr}^- from the definition of α in eqn. (7.87) gives

$$p_{rr}^- = \frac{\sigma_n^2 \alpha}{1 - \alpha} \tag{7.89}$$

Likewise, solving for $p_{r\dot{r}}^-$ from the definition of β in eqn. (7.87) gives

$$p_{r\dot{r}}^- = \frac{\beta\,(p_{rr}^- + \sigma_n^2)}{\Delta t} \tag{7.90}$$

Substituting eqn. (7.89) into eqn. (7.90) and simplifying gives

$$p_{r\dot{r}}^- = \frac{\sigma_n^2 \beta}{\Delta t\,(1-\alpha)} \tag{7.91}$$

Substituting eqns. (7.89) and (7.91) into eqn. (7.88), and after some moderate algebra (which is left as an exercise for the reader), yields

$$\boxed{\frac{\beta^2}{1-\alpha} = S_q^2} \tag{7.92}$$

The quantity S_q is known as the *tracking index*,[20] since it is proportional to the ratio of the process noise standard deviation and the measurement noise standard deviation. We should note that Kalata's index of Ref. [20] is slightly different, which is a function of Δt^2, not $\Delta t^{3/2}$ as shown by eqn. (7.84). This is due to the slightly different model chosen by Kalata, which is defined by

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix} w_k \tag{7.93}$$

This model assumes that the target undergoes a constant acceleration during the sampling interval and that the accelerations from period to period are independent.[19] This model may ignore the kinematical relationship shown by eqn. (7.76), and thus is not totally realistic.

A plot of α and β versus the tracking index S_q in eqn. (7.84) is shown in Figure 7.9. From this figure both α and β asymptotically approach limiting values. These limits will be assessed through a stability analysis. A simple closed-form solution for α and β can now be derived using eqns. (5.47) and (7.92). Using the steady-state version of eqn. (5.47) with $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$ and $R = \sigma_n^2$ yields the following simple form for the gain K:

$$K \equiv \begin{bmatrix} k_1 \\ \\ k_2 \end{bmatrix} = \sigma_n^{-2} \begin{bmatrix} p_{rr}^+ \\ \\ p_{r\dot{r}}^+ \end{bmatrix} \tag{7.94}$$

where $k_1 = \alpha$ and $k_2 = \beta/\Delta t$. The updated variances are given by eqn. (7.72) using the notation in this section:

$$p_{rr}^+ = \sigma_n^2 \left[1 - \left(\frac{S_q}{\xi} \right)^2 \right] \tag{7.95a}$$

$$p_{r\dot{r}}^+ = \left(\frac{\sigma_n}{\Delta t} \right)^2 \left[S_q^2 \left(\frac{1}{\xi} - \frac{1}{2} \right) - \xi \right] \tag{7.95b}$$

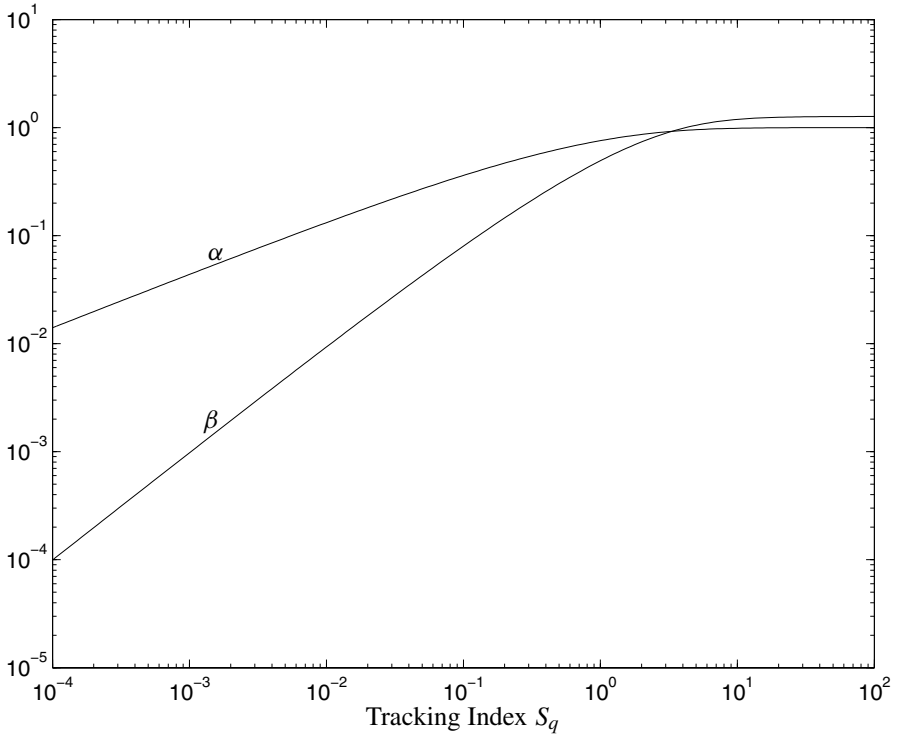

Figure 7.9: α-β Gains versus the Tracking Index

Therefore, from eqn. (7.94) α is simply given by

$$\boxed{\alpha = 1 - \left(\frac{S_q}{\xi}\right)^2} \tag{7.96}$$

Using eqn. (7.92) β is given by

$$\boxed{\beta = S_q\sqrt{1-\alpha}} \tag{7.97}$$

A direct relationship between α and β exists. This relationship is determined by first calculating the steady-state propagated and updated covariance in eqns. (5.35) and (5.44), respectively, with the definitions of Φ and $\Upsilon\, Q\, \Upsilon^T$ in eqns. (7.78) and (7.81), respectively. Substituting $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$ into eqn. (5.44) gives

$$\begin{bmatrix} p_{rr}^{+} & p_{r\dot{r}}^{+} \\ p_{r\dot{r}}^{+} & p_{\dot{r}\dot{r}}^{+} \end{bmatrix} = \begin{bmatrix} p_{rr}^{-}(1-k_1) & p_{r\dot{r}}^{-}(1-k_1) \\ p_{r\dot{r}}^{-} - k_2 p_{rr}^{-} & p_{\dot{r}\dot{r}}^{-} - k_2 p_{r\dot{r}}^{-} \end{bmatrix} \tag{7.98}$$

The matrix in eqn. (7.98) must be symmetric, which gives

$$k_1 = \left(\frac{p_{rr}^-}{p_{r\dot{r}}^-}\right) k_2 \tag{7.99}$$

Substituting eqns. (7.78) and (7.81) into eqn. (5.35) yields

$$\begin{bmatrix} p_{rr}^- & p_{r\dot{r}}^- \\ p_{r\dot{r}}^- & p_{\dot{r}\dot{r}}^- \end{bmatrix} = \begin{bmatrix} p_{rr}^+ + 2p_{r\dot{r}}^+ \Delta t + p_{\dot{r}\dot{r}}^+ \Delta t^2 & p_{r\dot{r}}^+ + p_{\dot{r}\dot{r}}^+ \Delta t \\ p_{r\dot{r}}^+ + p_{\dot{r}\dot{r}}^+ \Delta t & p_{\dot{r}\dot{r}}^+ \end{bmatrix} + q \begin{bmatrix} \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t \end{bmatrix} \tag{7.100}$$

From eqns. (7.98) and (7.100) the 2-2 element gives

$$k_2 = \frac{q\,\Delta t}{p_{r\dot{r}}^-} \tag{7.101}$$

Solving eqn. (7.99) for p_{rr}^- and using eqn. (7.101) gives

$$p_{rr}^- = \frac{k_1 q\,\Delta t}{k_2^2} \tag{7.102}$$

From eqns. (7.98) and (7.100) the 1-2 element gives

$$p_{\dot{r}\dot{r}}^- = p_{r\dot{r}}^- \left(\frac{k_1}{\Delta t} + k_2\right) - \frac{q\,\Delta t}{2} \tag{7.103}$$

From eqns. (7.98) and (7.100) the 1-1 element, with substituting of eqn. (7.103), yields

$$p_{rr}^- k_1 + p_{r\dot{r}}^- \Delta t\,(k_1 - 2) + \frac{q\,\Delta t^3}{6} = 0 \tag{7.104}$$

Solving eqn. (7.101) for $p_{r\dot{r}}^-$, and substituting the resulting equation and eqn. (7.102) into eqn. (7.104) yields

$$k_1^2 \Delta t + k_2 \Delta t^2 (k_1 - 2) + \frac{k_2^2 \Delta t^3}{6} = 0 \tag{7.105}$$

From the definitions of $k_1 \equiv \alpha$ and $k_2 \equiv \beta/\Delta t$, eqn. (7.105) reduces down to

$$\alpha^2 + \beta(\alpha - 2) + \frac{\beta^2}{6} = 0 \tag{7.106}$$

Hence, since β is always positive, which will be proven in the stability analysis, then α and β are related by

$$\boxed{\alpha = -\frac{1}{2}\beta + \frac{1}{2}\sqrt{\beta\,[(\beta/3) + 8]}} \tag{7.107}$$

This equation clearly shows the relationship between α and β, which can be written without S_q directly.

An interesting formula for β can also be derived using its relationship to $p_{r\dot{r}}^{+}$. Substituting eqn. (7.107) into eqn. (7.92) and squaring both sides of the resulting equation yields the following quartic equation:

$$\beta^4 + S_q^2\beta^3 + S_q^2\left[(S_q^2/6) - 2\right]\beta^2 + S_q^4\beta + S_q^4 = 0 \tag{7.108}$$

Note the similarity to eqn. (7.67)! In fact, the steps leading to eqn. (7.108) can be used to directly derive eqn. (7.67). The only solution that makes β valid in eqn. (7.92) is given by

$$\beta = \frac{1}{2}\left[\left(\frac{S_q^2}{2} + \vartheta\right) - \sqrt{\left(\frac{S_q^2}{2} + \vartheta\right)^2 - 4S_q^2}\right] \tag{7.109}$$

where

$$\vartheta = \left[4S_q^2 + S_q^4/12\right]^{1/2} \tag{7.110}$$

Also, from eqn. (7.92) α is given by

$$\alpha = \frac{S_q^2 - \beta^2}{S_q^2} \tag{7.111}$$

Both forms for α and β, eqns. (7.96) and (7.97), and eqns. (7.109) and (7.111), are acceptable.

The stability conditions for the α-β filter are now shown. From §5.3.2 the matrix $\Phi_k[I - K_k H_k]$ defines the stability of the Kalman filter. Since this matrix is now constant, its eigenvalues can be evaluated to develop a set of stability conditions for α and β. The eigenvalues of $\Phi_k[I - K_k H_k]$ are given by solving the following equation:

$$|zI - \Phi[I - K\,H]| = \det\begin{bmatrix} z + \alpha + \beta - 1 & -\Delta t \\ \beta/\Delta t & z - 1 \end{bmatrix} = 0 \tag{7.112}$$

Evaluating this determinant leads to the following characteristic equation:

$$z^2 + (\alpha + \beta - 2)z + (1 - \alpha) = 0 \tag{7.113}$$

As mentioned in §3.5 all eigenvalues must lie within the unit circle for a stable system. Even though the characteristic equation is second-order in nature, using the unit circle condition directly to prove stability is arduous. However, Jury's test[21] can be used to easily derive the stability conditions for α and β. Consider the following second-order polynomial:

$$z^2 + a_1 z + a_2 = 0 \tag{7.114}$$

where $a_1 \equiv \alpha + \beta - 2$ and $a_2 \equiv 1 - \alpha$. Jury's test for stability for this second-order equation involves satisfying the following three conditions:

$$a_2 < 1 \tag{7.115a}$$

$$a_2 > a_1 - 1 \tag{7.115b}$$

$$a_2 > -(a_1 + 1) \tag{7.115c}$$

From the definitions of a_1 and a_2, these conditions give $\alpha > 0$, $\beta > 0$, and $2\alpha + \beta < 4$. However, from eqn. (7.96), since $\alpha > 0$ and $(S_q/\xi)^2 > 0$ then the following conditions must be satisfied for stability:

$$0 < \alpha \leq 1 \tag{7.116a}$$

$$0 < \beta < 2 \tag{7.116b}$$

These conditions will always be met since §5.3.2 shows that the Kalman filter is stable as long as $q \geq 0$ and $\sigma_n^2 > 0$.

7.4.2 The α-β-γ Filter

In this section the α-β filter of §7.4.1 is expanded to include an acceleration state. This approach in theory provides better estimates since a higher-order filter is used, but the computational requirements will certainly be greater than the α-β filter. To derive this new filter we begin with the following simple truth model in continuous-time:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t) \tag{7.117}$$

where $w(t)$ is the process noise with variance q, and the states $\mathbf{x} \equiv \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$ are position velocity and acceleration denoted by r, $\dot{r}$, and $\ddot{r}$, respectively. Note that the first two states do not contain any process noise, since these are kinematical relationships. Discrete-time measurements of position are assumed, so that

$$\tilde{y}_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}_k + v_k \equiv H\mathbf{x}_k + v_k \tag{7.118}$$

where v_k is the measurement noise, which is assumed to be modelled by a zero-mean Gaussian white-noise process with variance σ_n^2. The state transition matrix for the discrete-time model can be computed using eqn. (3.25). Since $F^3 = 0$ for the model in eqn. (7.117), then the discrete-time state matrix is given by

$$\Phi = I + \Delta t\, F + \frac{\Delta t^2}{2} F^2 = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{7.119}$$

where Δt is the sampling interval. The discrete-time process noise can be computed using eqn. (7.79), which yields

$$\Upsilon Q \Upsilon^T = q \begin{bmatrix} \Delta t^5/20 & \Delta t^4/8 & \Delta t^3/6 \\ \Delta t^4/8 & \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^3/6 & \Delta t^2/2 & \Delta t \end{bmatrix} \tag{7.120}$$

Note that the lower left 2×2 sub-matrix of eqn. (7.120) is equivalent to the matrix in eqn. (7.81).

Substituting the sensitivity and state matrices of eqns. (7.118) and (7.119) into the discrete-time Kalman update and propagation equations shown in Table 5.1 leads to

$$\hat{r}_k^+ = \hat{r}_k^- + \alpha [\tilde{y}_k - \hat{r}_k^-] \tag{7.121a}$$

$$\dot{\hat{r}}_k^+ = \dot{\hat{r}}_k^- + \frac{\beta}{\Delta t} [\tilde{y}_k - \hat{r}_k^-] \tag{7.121b}$$

$$\ddot{\hat{r}}_k^+ = \ddot{\hat{r}}_k^- + \frac{\gamma}{2\Delta t^2} [\tilde{y}_k - \hat{r}_k^-] \tag{7.121c}$$

$$\hat{r}_{k+1}^- = \hat{r}_k^+ + \dot{\hat{r}}_k^+ \Delta t + \frac{1}{2} \ddot{\hat{r}}_k^+ \Delta t^2 \tag{7.121d}$$

$$\dot{\hat{r}}_{k+1}^- = \dot{\hat{r}}_k^+ + \ddot{\hat{r}}_k^+ \Delta t \tag{7.121e}$$

$$\ddot{\hat{r}}_{k+1}^- = \ddot{\hat{r}}_k^+ \tag{7.121f}$$

where the gain matrix in Table 5.1 is given by $K_k = K \equiv \left[\alpha \;\; \beta/\Delta t \;\; \gamma/(2\Delta t^2)\right]^T$.

As with the α-β filter, the gains of the α-β-γ filter are related to each other. The filter should be designed by tuning q only, where changes in the acceleration over the sampling interval are of the order $\sqrt{q\Delta t}$. However, unlike the α-β filter, a closed-form solution showing a direct relationship of q to the gains is not straightforward. The tracking index in eqn. (7.84) is still useful though. A plot of α, β, and γ versus the tracking index S_q is shown in Figure 7.10. From this figure α, β, and γ asymptotically approach limiting values. These limits will be assessed through a stability analysis, which has been presented in Ref. [22]. Consistent with the analysis shown in §7.4.1, the eigenvalues of $\Phi_k[I - K_k H_k]$ are given by solving the following equation:

$$|zI - \Phi[I - K\,H]| = \det \begin{bmatrix} z + \alpha + \beta + \frac{1}{4}\gamma - 1 & -\Delta t & -\frac{1}{2}\Delta t^2 \\ \frac{1}{2\Delta t}(2\beta + \gamma) & z - 1 & -\Delta t \\ \frac{1}{2\Delta t^2}\gamma & 0 & z - 1 \end{bmatrix} = 0 \tag{7.122}$$

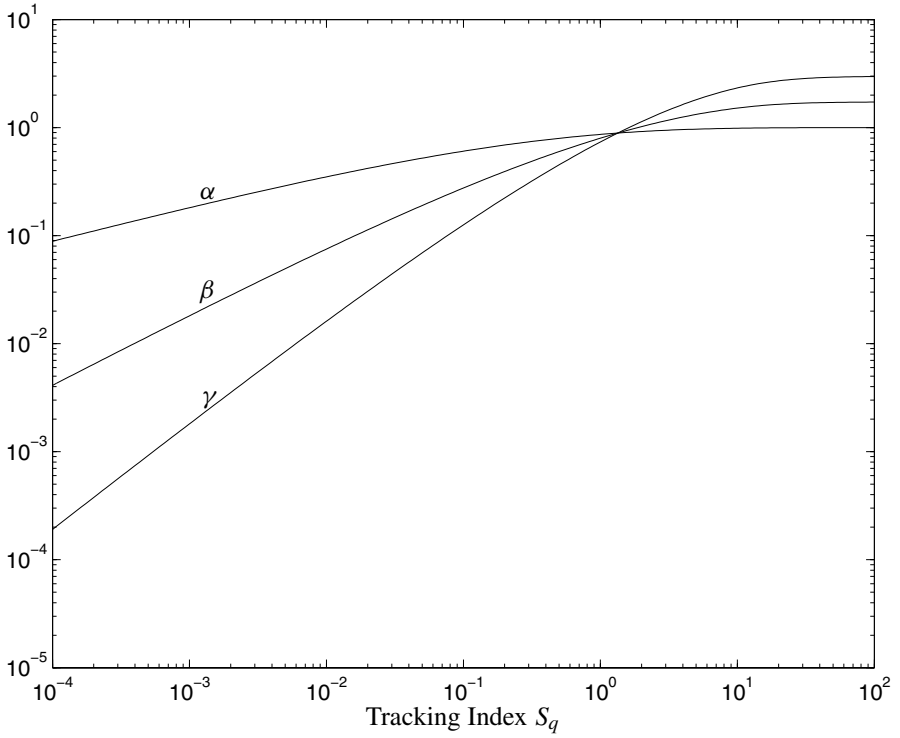

Figure 7.10: α-β-γ Gains versus the Tracking Index

Evaluating this determinant leads to the following characteristic equation:

$$z^3 + (\alpha + \beta + \frac{1}{4}\gamma - 3)z^2 + (3 - 2\alpha - \beta + \frac{1}{4}\gamma)z + (\alpha - 1) = 0 \tag{7.123}$$

Tenne and Singh[22] have evaluated the stability of this characteristic equation using Jury's test.[21] The conditions for stability are given by α and β greater than zero, and

$$2\alpha + \beta < 4 \tag{7.124a}$$

$$0 < \gamma < \frac{4\alpha\beta}{2-\alpha} \tag{7.124b}$$

From Figure 7.10 these conditions are clearly met for all positive values of q, as expected. Furthermore, if we assume $0 < \alpha \leq 1$, then the stability conditions in eqn. (7.124) reduce down to

$$0 < \alpha \leq 1 \tag{7.125a}$$

$$0 < \beta < 2 \tag{7.125b}$$

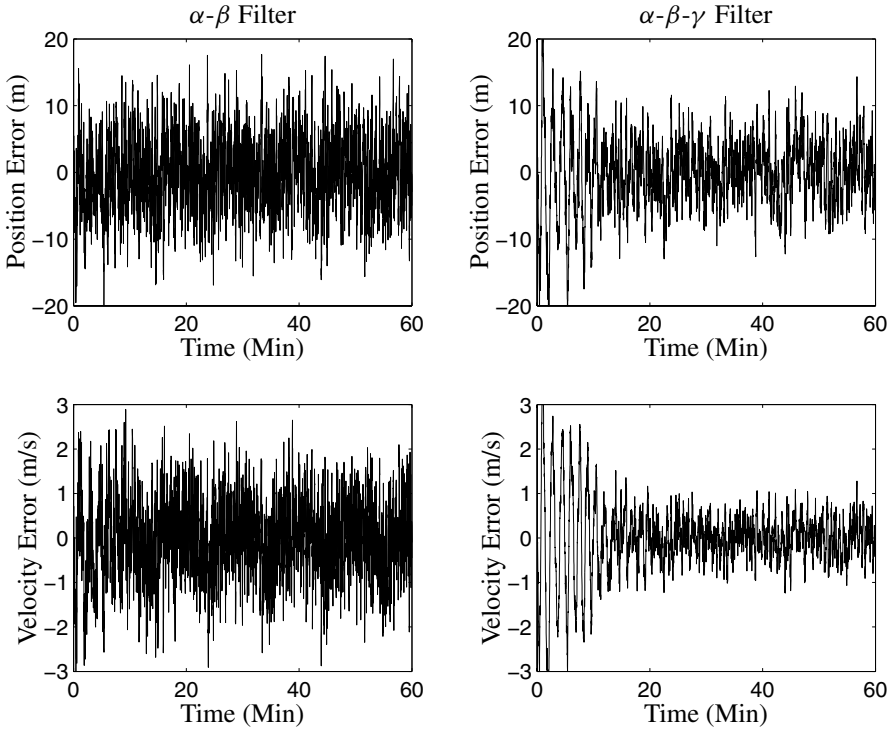

Figure 7.11: Position and Velocity Tracking Error Results Using Both Filters

$$0 < \gamma < \frac{4\alpha\beta}{2-\alpha} \tag{7.125c}$$

Reference [22] also derives metrics to gauge the transient response and steady-state tracking error, and also shows the relationships between the gain parameters for specific maneuvers. These relationships can be used to provide an initial estimate for α, β, and γ, although tuning q is preferred, which exploits the kinematically relationship in the assumed model.

Example 7.4: A simulation involving tracking the vertical position of a 747 aircraft using both the α-β and α-β-γ filters is shown. The longitudinal equations of motion are shown in example 4.4. Using the aircraft flight parameters shown in example 4.4 the equations of motion are integrated over a 60-minute simulation. The thrust is set equal to the computed drag, and the elevator is set to 1 degree down from the trim value for the entire simulation interval. The vertical position, z, has standard deviation of 10 m for the measurement error. Measurements are sampled at 1 sec intervals.

Since we know the truth, then the variance parameter q in both the α-β and α-β-γ filters is tuned to ensure the best possible performance. This parameter is varied

until transients begin to appear in the position errors. For the α-β filter the optimal parameter is given by $q = 0.5$. From eqns. (7.96) and (7.97) this value of q gives $\alpha = 0.31344$ and $\beta = 0.05859$. For the α-β-γ filter the optimal parameter is given by $q = 0.0001$. Note this value is much smaller than the value used in the α-β filter. This is due to the fact that q now affects changes in acceleration, which is smaller in magnitude than changes in velocity. Solving the steady-state discrete-time covariance equation in Table 5.2 using the method outlined in §5.3.4 gives $\alpha = 0.18127$, $\beta = 0.01811$, and $\gamma = 0.00181$. A plot of the tracking error results for vertical position and velocity using both filters is shown in Figure 7.11. The 3σ bounds computed from the steady-state error-covariance are 20.27 m (position) and 5.14 m/s (velocity) for the α-β filter, and 14.12 m (position), 1.70 m/s (velocity), and 0.136 m/s^2 (acceleration) for the α-β-γ filter. Clearly, the α-β-γ filter outperforms the α-β filter, but comes at a higher computational cost.

More details on α-β-γ filtering can be found in the references cited in §7.4.1 and §7.4.2. The α-β and α-β-γ filters described here have been widely used in a number of applications, which is mainly due to the simplicity of the filtering mechanisms. For aircraft applications a filter with a more rigorous flight dynamics-based model can significantly improve the tracking accuracy, as shown in Ref. [23]. Also, a simple dynamics-based filter for application to automatic landings on an aircraft carrier is shown in Ref. [24], which gives superior results to the standard α-β-γ filter for control purposes. The reader is highly encouraged to pursue actual applications in the references cited here and in the open literature.

7.4.3 Aircraft Parameter Estimation

In §4.4 parameter identification using a batch set of flight measurement data has been shown. In this section parameter estimation is considered using the extended Kalman filter. This allows for the implementation of real-time estimation, which can be used to update an aircraft model for adaptive control purposes. In this section the focus is only on the longitudinal equations of motions, but this formulation can easily be extended to the general case involving coupled motion. The EKF approach for aircraft parameter estimation involves appending the state vector to include the unknown parameters. The derivative of these parameters is zero, which can easily be put into a state-space form. In this section we present this approach to estimate C_{D_0}, C_{L_0} and C_{m_0} using measurements of angle of attack, velocity, angular rate, and pitch angle. The longitudinal equations of motion are shown in example 4.4. The state vector, $\mathbf{x}$, consists of v_1, v_2, ω_2, θ, C_{D_0}, C_{L_0}, and C_{m_0}. Note that the horizontal and vertical positions, x and z, are not required in this formulation. See §3.9 for a full description of the equations of motion for an aircraft.

Several partial derivatives are required in the EKF. These may be computed numerically using the method described in example 4.4, but we instead choose to derive analytical expressions here. The partial derivatives of α with respect to v_1 and v_3 are

given by

$$\frac{\partial \alpha}{\partial v_1} = -\frac{v_3}{v_1^2 + v_3^2} \tag{7.126a}$$

$$\frac{\partial \alpha}{\partial v_3} = \frac{v_1}{v_1^2 + v_3^2} \tag{7.126b}$$

where

$$\alpha = \tan^{-1} \frac{v_3}{v_1} \tag{7.127}$$

The partial derivatives of the drag force, D, with respect to v_1 and v_3 are given by

$$\frac{\partial D}{\partial v_1} = C_D \rho v_1 S - \frac{\rho C_{D_\alpha} v_3}{2(1+\alpha^2) v_1^2} ||\mathbf{v}||^2 S \tag{7.128a}$$

$$\frac{\partial D}{\partial v_3} = C_D \rho v_3 S + \frac{\rho C_{D_\alpha}}{2(1+\alpha^2) v_1} ||\mathbf{v}||^2 S \tag{7.128b}$$

where $||\mathbf{v}||^2 = v_1^2 + v_3^2$ and

$$C_D = C_{D_0} + C_{D_\alpha} \alpha + C_{D_{\delta_E}} \delta_E \tag{7.129}$$

The partial derivatives of the lift force, L, with respect to v_1 and v_3 are given by

$$\frac{\partial L}{\partial v_1} = C_L \rho v_1 S - \frac{\rho C_{L_\alpha} v_3}{2(1+\alpha^2) v_1^2} ||\mathbf{v}||^2 S \tag{7.130a}$$

$$\frac{\partial L}{\partial v_3} = C_L \rho v_3 S + \frac{\rho C_{L_\alpha}}{2(1+\alpha^2) v_1} ||\mathbf{v}||^2 S \tag{7.130b}$$

where

$$C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_{\delta_E}} \delta_E \tag{7.131}$$

These partial derivatives will be used in the derivation of the matrix $F(\hat{\mathbf{x}}(t), t)$ for the EKF shown in Table 5.9.

The partial derivative components of $\dot{v}_1$ with respect to the state vector, which give the first row of $F(\mathbf{x}(t), t)$, are given by

$$\frac{\partial \dot{v}_1}{\partial v_1} = \frac{1}{m} \left\{ \left[\frac{\partial L}{\partial v_1} + D \frac{\partial \alpha}{\partial v_1} \right] \sin\alpha + \left[L \frac{\partial \alpha}{\partial v_1} - \frac{\partial D}{\partial v_1} \right] \cos\alpha \right\} \tag{7.132a}$$

$$\frac{\partial \dot{v}_1}{\partial v_3} = \frac{1}{m} \left\{ \left[\frac{\partial L}{\partial v_3} + D \frac{\partial \alpha}{\partial v_3} \right] \sin\alpha + \left[L \frac{\partial \alpha}{\partial v_3} - \frac{\partial D}{\partial v_3} \right] \cos\alpha \right\} - \omega_2 \tag{7.132b}$$

$$\frac{\partial \dot{v}_1}{\partial \omega_2} = -v_3 \tag{7.132c}$$

$$\frac{\partial \dot{v}_1}{\partial \theta} = -g \cos\theta \tag{7.132d}$$

$$\frac{\partial \dot{v}_1}{\partial C_{D_0}} = -\frac{1}{2m} \rho ||\mathbf{v}||^2 S \cos\alpha \tag{7.132e}$$

$$\frac{\partial \dot{v}_1}{\partial C_{L_0}} = \frac{1}{2m}\rho\,||\mathbf{v}||^2 S\sin\alpha \tag{7.132f}$$

$$\frac{\partial \dot{v}_1}{\partial C_{m_0}} = 0 \tag{7.132g}$$

The partial derivative components of $\dot{v}_3$ with respect to the state vector, which give the second row of $F(\mathbf{x}(t), t)$, are given by

$$\frac{\partial \dot{v}_3}{\partial v_1} = \frac{1}{m}\left\{\left[-D\frac{\partial \alpha}{\partial v_1} - \frac{\partial L}{\partial v_1}\right]\cos\alpha + \left[L\frac{\partial \alpha}{\partial v_1} - \frac{\partial D}{\partial v_1}\right]\sin\alpha\right\} + \omega_2 \tag{7.133a}$$

$$\frac{\partial \dot{v}_3}{\partial v_3} = \frac{1}{m}\left\{\left[-D\frac{\partial \alpha}{\partial v_3} - \frac{\partial L}{\partial v_3}\right]\cos\alpha + \left[L\frac{\partial \alpha}{\partial v_3} - \frac{\partial D}{\partial v_3}\right]\sin\alpha\right\} \tag{7.133b}$$

$$\frac{\partial \dot{v}_3}{\partial \omega_2} = v_1 \tag{7.133c}$$

$$\frac{\partial \dot{v}_3}{\partial \theta} = -g\,\sin\theta \tag{7.133d}$$

$$\frac{\partial \dot{v}_3}{\partial C_{D_0}} = -\frac{1}{2m}\rho\,||\mathbf{v}||^2 S\sin\alpha \tag{7.133e}$$

$$\frac{\partial \dot{v}_3}{\partial C_{L_0}} = -\frac{1}{2m}\rho\,||\mathbf{v}||^2 S\cos\alpha \tag{7.133f}$$

$$\frac{\partial \dot{v}_3}{\partial C_{m_0}} = 0 \tag{7.133g}$$

The partial derivative components of $\dot{\omega}_2$ with respect to the state vector, which give the third row of $F(\mathbf{x}(t), t)$, are given by

$$\frac{\partial \dot{\omega}_2}{\partial v_1} = \frac{\rho S\bar{c}}{J_{22}}\left[\left(C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\delta_E}}\delta_E\right)v_1 - \frac{C_{m_\alpha} v_3}{2(1+\alpha^2)v_1^2}||\mathbf{v}||^2\right] \tag{7.134a}$$

$$\frac{\partial \dot{\omega}_2}{\partial v_3} = \frac{\rho S\bar{c}}{J_{22}}\left[\left(C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\delta_E}}\delta_E\right)v_3 + \frac{C_{m_\alpha}}{2(1+\alpha^2)v_1}||\mathbf{v}||^2\right] \tag{7.134b}$$

$$\frac{\partial \dot{\omega}_2}{\partial \omega_2} = \frac{1}{4J_{22}}\rho S\bar{c}^2 C_{m_q} \tag{7.134c}$$

$$\frac{\partial \dot{\omega}_2}{\partial \theta} = 0 \tag{7.134d}$$

$$\frac{\partial \dot{\omega}_2}{\partial C_{D_0}} = 0 \tag{7.134e}$$

$$\frac{\partial \dot{\omega}_2}{\partial C_{L_0}} = 0 \tag{7.134f}$$

$$\frac{\partial \dot{\omega}_2}{\partial C_{m_0}} = \frac{1}{2J_{22}}\rho\,||\mathbf{v}||^2 S\bar{c} \tag{7.134g}$$

The 4-3 element of $F(\mathbf{x}(t), t)$ is given by 1, which is derived from the kinematical equation $\dot{\theta} = \omega_2$. All other entries of $F(\mathbf{x}(t), t)$ are zero since C_{D_0}, C_{L_0}, and C_{m_0}

are constants. The output vector is given

$$\mathbf{y} = \begin{bmatrix} \alpha \\ ||\mathbf{v}|| \\ \omega_2 \\ \theta \end{bmatrix} \tag{7.135}$$

The matrix sensitivity matrix H is given by

$$H(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial \alpha}{\partial v_1} & \dfrac{\partial \alpha}{\partial v_3} & 0\ 0\ 0\ 0\ 0 \\ \dfrac{\partial ||\mathbf{v}||}{\partial v_1} & \dfrac{\partial ||\mathbf{v}||}{\partial v_3} & 0\ 0\ 0\ 0\ 0 \\ 0 & 0 & 1\ 0\ 0\ 0\ 0 \\ 0 & 0 & 0\ 1\ 0\ 0\ 0 \end{bmatrix} \tag{7.136}$$

where

$$\frac{\partial ||\mathbf{v}||}{\partial v_1} = \frac{v_1}{||\mathbf{v}||} \tag{7.137a}$$

$$\frac{\partial ||\mathbf{v}||}{\partial v_3} = \frac{v_3}{||\mathbf{v}||} \tag{7.137b}$$

The continuous-discrete extended Kalman filter in Table 5.9 can now be implemented with $F(\hat{\mathbf{x}}(t), t)$ and $H_k(\hat{\mathbf{x}}_k)$ evaluated at the current state estimates.

Example 7.5: To illustrate the power of using the extended Kalman filter for real-time parameter applications, we show an example of identifying the longitudinal parameters of a simulated 747 aircraft. The longitudinal equations of motion are shown in example 4.4. Using the aircraft flight parameters shown in example 4.4 the equations of motion are integrated over a 30-second simulation. The thrust is set equal to the computed drag, and the elevator is set to 1 degree down from the trim value for the first 10 seconds and then returned to the trimmed value thereafter. Measurements of angle of attack, α, velocity, $||\mathbf{v}||$, angular velocity, ω_2, and pitch angle, θ, are assumed with standard deviations of the measurement errors given by $\sigma_\alpha = 0.5$ degrees, $\sigma_{||\mathbf{v}||} = 1$ m/s, $\sigma_{\omega_2} = 0.01$ deg/sec, and $\sigma_\theta = 0.1$ degrees, respectively. Since real-time estimates are required the measurements are sampled at 0.1-second intervals. The continuous-time model and error-covariance are integrated using a time step of 0.01 seconds, which is needed to ensure adequate performance in the EKF propagation.

The initial conditions for $\hat{v}_1$, $\hat{v}_2$, $\hat{\omega}_2$, and $\hat{\theta}$ are set to their true values. The initial conditions for the parameters to be estimated are given by $C_{D_0} = 0.01$, $C_{L_0} = 0.1$, and $C_{m_0} = 0.01$. The initial error-covariance is given by

$$P_0 = \text{diag}\left[1 \times 10^{-5}\ 1 \times 10^{-5}\ 1 \times 10^{-5}\ 1 \times 10^{-6}\ 1\ 1\ 1\right]$$

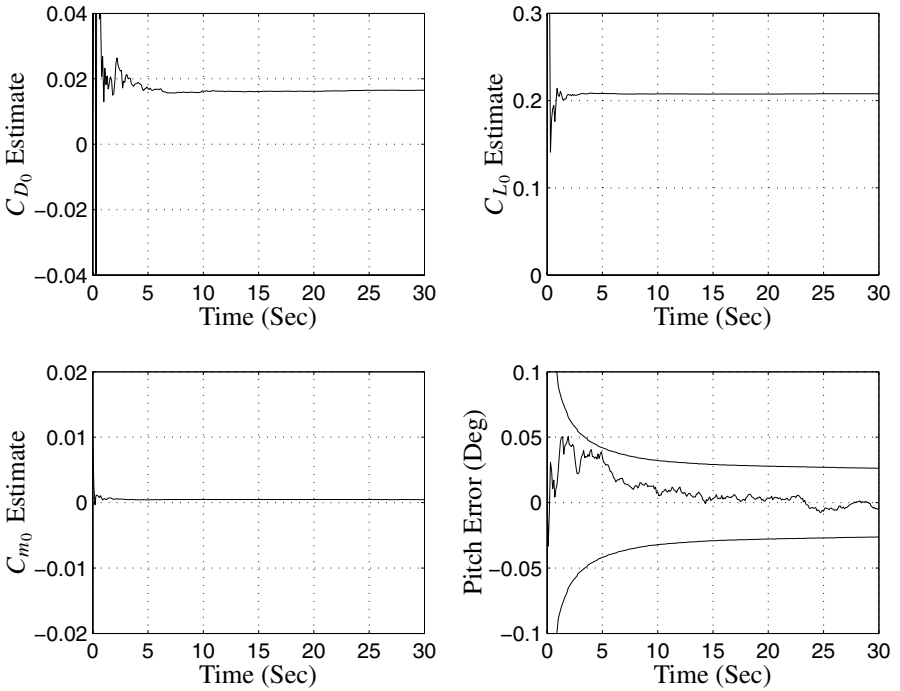

Figure 7.12: Parameter Estimates and Pitch Angle Error

A plot of the parameter estimates is shown in Figure 7.12. The final values at the end of the simulation run are given by $C_{D_0} = 0.0164$, $C_{L_0} = 0.2082$, and $C_{m_0} = 0.0003$, which are close to the batch solutions shown in example 4.4. A plot of the pitch angle errors and associated 3σ bounds is also shown in Figure 7.12. The errors are within their respective 3σ bounds, which indicates that the EKF is performing in an optimal manner. This example clearly shows the usefulness of the extended Kalman filter for real-time parameter estimations. The example shown herein can also be implemented as a real-time dynamics-based filter, without updating the aircraft parameters in the model.[23]

7.5 Smoothing with the Eigensystem Realization Algorithm

The Eigensystem Realization Algorithm (ERA) of §4.5 is fairly accurate for measurements that contain small measurement noise levels. However, significant errors can be produced with high measurement noise, which will be shown in example 7.6. This problem can be overcome by using frequency domain-based filtering methods, which use frequency-response function averaging. But this requires more data sets and computational effort. The approach presented in this section involves first smoothing the measurements using the discrete-time fixed-interval smoothing algorithm of §6.1.1. Since the ERA approach is in essence a batch least squares estimator, it seems natural to use a batch-type estimator to smooth the effects of the large measurement errors. This approach can be shown to be superior over standard band-pass or low-pass filtering of the data.[25]

The theoretical developments of the combined smoother/ERA approach begins with the state-space form of the vibratory system shown in §3.10:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix}\mathbf{u} + \begin{bmatrix} 0 \\ I \end{bmatrix}\mathbf{w} \tag{7.138a}$$

$$\equiv F\mathbf{x} + B\mathbf{u} + G\mathbf{w}$$

$$\tilde{\mathbf{y}}_k = H\mathbf{x}_k + \mathbf{v}_k \tag{7.138b}$$

where $\mathbf{x}$ now denotes a $2n$ vector of n position states and n velocity states. In this model the process noise is only added to the velocity states since, as discussed in §7.4.1, the first n states of eqn. (7.138a) represent a kinematical relationship. Typically, an *a priori* model of a particular vibratory system is predetermined using a finite element analysis, which was later demonstrated to be a Rayleigh-Ritz method.[26] Exploitation of the second-order block-structure of the model in eqn. (7.138) allows one to use a reduced-order Kalman filter and smoother form.[27, 28] However, since a steady-state gain in the forward-time Kalman filter and backward-time smoother will be used here, which can be determined off-line, we choose to retain the full-order form.

The first step in the Rauch, Tung, and Striebel (RTS) smoother involves executing the Kalman filter forward in time. A method to determine the process noise covariance involves an off-line computation to satisfy the autocorrelation test in eqns. (5.250) and (5.251). Since the state matrices are constant and the measurements are assumed to be sampled frequently, then the steady-state discrete-time Kalman filter shown in Table 5.2 can be used. The discrete-time state matrices, Φ and Γ, can be numerically determined using eqns. (3.112) and (3.113). An analytical solution for the discrete-time process noise covariance is difficult to determine for high-order models. Therefore, eqn. (5.140) will be used to determine this covariance matrix. The steady-state error-covariance matrix computed from the discrete-time algebraic Riccati equation in Table 5.2 is now denoted by P_f^- to reflect the fact that this matrix is the propagated steady-state solution of the forward Kalman filter. The RTS

smoother steady-state gain in Table 6.2 is given by

$$\boxed{\mathcal{K} = P_f^+ \Phi^T (P_f^-)^{-1}} \tag{7.139}$$

where P_f^+ is given in Table 6.2 as well:

$$P_f^+ = [I - K_f H] P_f^- \tag{7.140a}$$

$$K_f = P_f^- H^T [H P_f^- H^T + R]^{-1} \tag{7.140b}$$

where R is the covariance of $\mathbf{v}_k$, shown in eqn. (7.138b). From Table 6.2 the steady-state RTS smoother covariance, denoted by P, can be computed by solving the following discrete-time Lyapunov equation:

$$\boxed{P = \mathcal{K} P \mathcal{K}^T + [P_f^+ - \mathcal{K} P_f^- \mathcal{K}^T]} \tag{7.141}$$

This covariance can be used to determine the performance characteristics of the RTS smoothing algorithm.

The procedure to determine the state-space system matrices is as follows. First, determine an initial model of the system at hand. If one is not given, then the ERA algorithm can be employed to determine this model from the noisy measurement sets. Next, implement the discrete-time Kalman filter to determine filtered state estimates. Then, use the discrete-time RTS smoother to determine smoothed output estimates. Finally, use the ERA algorithm with the smoothed output estimates to determine the system matrices. The Modal Amplitude Coherence (MAC) in eqn. (4.91) can be used to compare the performance of the combined smoother/ERA approach with the ERA approach alone. If the smoother is working properly, then an identified mode should have a higher MAC value than the mode identified by ERA alone.

Example 7.6: In this example we will use the ERA to identify the mass, stiffness, and damping matrices of a 4-mode system from simulated high-noise mass-position measurements. The description of the model and the assumed mass, stiffness, and damping matrices are shown in example 4.5. With the exact solution known, Gaussian white-noise of approximately 5% the size of the signal amplitude is added to simulate the output measurements. A 50-second simulation is performed, with measurements sampled every 0.1 seconds. Using all available measurements, the Hankel matrix in the ERA was chosen to be a 400×1600 dimension matrix. After computing the discrete-time state matrices using eqn. (4.85), a conversion to continuous-time state matrices is performed, and the mass, stiffness, and damping matrices are computed using eqn. (4.99). The results of this computation are

$$M = \begin{bmatrix} -0.7376 & 2.0831 & -1.5368 & 0.8198 \\ 2.3310 & -1.7600 & 1.8760 & -0.8917 \\ -1.5544 & 1.9296 & -0.4804 & 0.8381 \\ 0.7807 & -0.8992 & 0.6590 & 0.6519 \end{bmatrix}$$

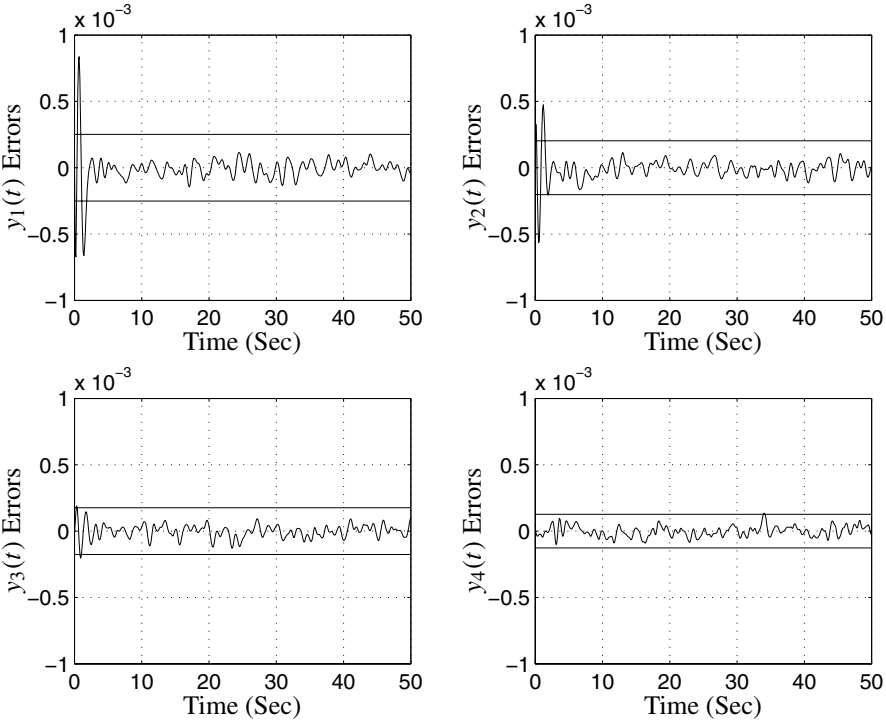

Figure 7.13: Position Errors with 3σ Bounds

$$K = \begin{bmatrix} 6.2382 & -0.2996 & -3.6281 & 1.8429 \\ 1.3916 & 2.4294 & -0.1185 & -1.9367 \\ -9.3119 & 5.9243 & 3.3579 & -2.6469 \\ 6.7156 & -7.4445 & -1.0620 & 9.0596 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.3355 & 1.6663 & -2.9785 & 2.0538 \\ 2.9750 & -2.9882 & 2.4879 & -1.4765 \\ -6.9475 & 6.7105 & -1.6973 & -0.4730 \\ 5.5428 & -5.6182 & 0.5978 & 2.8823 \end{bmatrix}$$

These realized matrices are not close to the true matrices, shown in example 4.5, which is due to the large measurement errors used in the current simulation. Note that some of the diagonal elements are not even positive!

The RTS smoother is implemented to provided smoothed estimates, which are used in the ERA. For the RTS state model we assume that the mass matrix is given by the true mass matrix, but the stiffness matrix is given by 0.9 times the true stiffness matrix. Also, the damping matrix is given by the true *stiffness* matrix divided by 10, which introduces a large error in the state model. This error approach in the assumed model provides a typical scenario where the mass and stiffness matrices are well known, but the damping matrix is not well known. The continuous-time

covariance is determined by trial and error. A value of $1 \times 10^{-6} I_{4\times 4}$ is found to produce accurate results, which can be verified by the 3σ bounds computed from the diagonal elements of eqn. (7.141). A plot of the position errors with 3σ bounds for an impulse input to the first mass is shown in Figure 7.13. The initial transients are due to the fact that a steady-state gain is used in the RTS smoother. Clearly, the RTS smoother is performing in an optimal fashion. Using the smoothed estimates in the ERA the mass, stiffness, and damping matrices are now computed to be

$$M = \begin{bmatrix} 1.0170 & 0.0023 & 0.0043 & 0.0093 \\ -0.0050 & 1.0093 & -0.0071 & 0.0005 \\ 0.0123 & -0.0027 & 1.0031 & 0.0084 \\ 0.0173 & 0.0145 & -0.0141 & 1.0203 \end{bmatrix}$$

$$K = \begin{bmatrix} 9.4631 & -4.4972 & -0.1975 & -0.0529 \\ -4.4832 & 9.2467 & -4.4816 & -0.1554 \\ -0.0814 & -4.4870 & 9.1694 & -4.3763 \\ 0.0065 & -0.0894 & -4.5658 & 9.5069 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.2389 & -0.4058 & -0.1324 & -0.0259 \\ -0.4370 & 1.1933 & -0.4592 & -0.1316 \\ -0.1377 & -0.4456 & 1.1852 & -0.4273 \\ -0.0232 & -0.1346 & -0.3893 & 1.2430 \end{bmatrix}$$

These matrices are now much closer to the true values than the ones computed using the ERA with the raw measurements. A better comparison involves looking at the identified natural frequencies and damping ratios, which are given by

True		ERA		RTS/ERA	
ω_n	ζ	ω_n	ζ	ω_n	ζ
1.3820	0.1382	1.3814	0.1376	1.3786	0.1354
2.6287	0.2629	2.6563	0.2658	2.6016	0.2155
3.6180	0.3618	0.2545, 1.5778	1.0000	3.4694	0.2231
4.2533	0.4253	3.5146, 4.6940	1.0000	4.0181	0.2184

The modes with a damping ratio of 1 correspond to real-valued modes (i.e., with no complex parts). The MAC factors are given by

ERA		RTS/ERA	
ω_n	MAC	ω_n	MAC
1.3814	1.0000	1.3786	1.0000
2.6563	0.9957	2.6016	0.9990
0.2545, 1.5778	0.7148, 0.7658	3.4694	0.9976
3.5146, 4.6940	0.7841, 0.7398	4.0181	0.9983

Clearly, using the ERA with the raw measurements did not properly identify the high frequency modes, which is due to the fact that the noisy levels make these modes nearly unobservable. The combined RTS/ERA approach did manage to provide a significant improvement in the obtained results. The results are reinforced by the MAC factors, where the higher modes have a MAC close to one using the combined RTS/ERA approach.

7.6 Summary

In this chapter several applications of the linear and extended Kalman filter have been presented for Global Positioning System navigation, spacecraft attitude estimation and gyro bias determination from various sensor devices, orbit determination from ground-based sensors, aircraft tracking from radar measurements and parameter identification using on-board measurements, and robust modal identification of vibratory systems using the RTS smoother to provide optimal estimates. As with Chapter 4, we anticipate that most readers will profit greatly by a careful study of these applications in this chapter. Once again, the constraints imposed by the length of this text did not, however, permit an entirely self-contained and satisfactory development of the concepts introduced in the applications of this chapter. It will likely prove useful for the interested reader to pursue these important subjects in the cited literature. For example, the integration of GPS and Inertial Navigation Systems represents an extremely useful tool in modern-day navigation. However, due to constraints imposed by the length of this text, a full treatise is not possible here. Several texts dedicated just to this subject have been written, (e.g., see Refs. [6], [7], and [29]), which we highly recommend to the interested reader.

A summary of the key formulas presented in this chapter is given below.

- GPS Coordinate Transformations

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix}$$

$$d_{2000} = 367y - \text{INT}\left\{\frac{7\{y + \text{INT}[(m+9)/12]\}}{4}\right\} + \text{INT}\left\{\frac{275m}{9}\right\} + \frac{h + min/60 + s/3600}{24} + d - 730531.5$$

$$\theta = 280.46061837 + 360.98564736628 \times d_{2000}$$

- Geodetic to ECEF Conversion

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \lambda}}$$

$$x = (N + h) \cos\lambda \cos\phi$$
$$y = (N + h) \cos\lambda \sin\phi$$
$$z = [N(1 - e^2) + h] \sin\lambda$$

- ECEF to Geodetic Conversion

$$p = \sqrt{x^2 + y^2}$$
$$\psi = \text{atan}\left(\frac{z\,a}{p\,b}\right)$$
$$\bar{e}^2 = \frac{a^2 - b^2}{b^2}$$
$$\lambda = \text{atan}\left(\frac{z + \bar{e}^2 b \sin^3\psi}{p - e^2 a \cos^3\psi}\right)$$
$$\phi = \text{atan2}(y, x)$$
$$h = \frac{p}{\cos\lambda} - N$$

- GPS Satellite Elevation

$$\mathbf{u} = \begin{bmatrix} \cos\lambda \cos\phi \\ \cos\lambda \sin\phi \\ \sin\lambda \end{bmatrix}$$
$$\cos\xi_i = \boldsymbol{\rho}_i^T \mathbf{u}$$
$$\boldsymbol{\rho}_i = \frac{\mathbf{e}_i - \mathbf{r}}{||\mathbf{e}_i - \mathbf{r}||}$$
$$E_i = 90^\circ - \xi_i$$

- Attitude Estimation

$$\dot{\hat{\mathbf{q}}}(t) = \frac{1}{2} \Xi\left(\hat{\mathbf{q}}(t)\right) \hat{\boldsymbol{\omega}}(t)$$

$$\Delta\tilde{\mathbf{x}}(t) \equiv \begin{bmatrix} \boldsymbol{\delta\alpha} \\ \Delta\boldsymbol{\beta} \end{bmatrix}$$

$$F(\hat{\mathbf{x}}(t), t) = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}(t)\times] & -I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}$$
$$G(t) = \begin{bmatrix} -I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix}$$
$$Q(t) = \begin{bmatrix} \sigma_v^2 I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \sigma_u^2 I_{3\times3} \end{bmatrix}$$

$$H_k(\hat{\mathbf{x}}_k^-) = \left.\begin{bmatrix} [A(\hat{\mathbf{q}}^-)\mathbf{r}_1\times] & 0_{3\times 3} \\ [A(\hat{\mathbf{q}}^-)\mathbf{r}_2\times] & 0_{3\times 3} \\ \vdots & \vdots \\ [A(\hat{\mathbf{q}}^-)\mathbf{r}_n\times] & 0_{3\times 3} \end{bmatrix}\right|_{t_k} \tag{7.142}$$

$$\mathbf{h}_k(\hat{\mathbf{x}}_k^-) = \left.\begin{bmatrix} A(\hat{\mathbf{q}}^-)\mathbf{r}_1 \\ A(\hat{\mathbf{q}}^-)\mathbf{r}_2 \\ \vdots \\ A(\hat{\mathbf{q}}^-)\mathbf{r}_n \end{bmatrix}\right|_{t_k} \tag{7.143}$$

$$\Delta\hat{\tilde{\mathbf{x}}}_k^+ = K_k[\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)]$$

$$\hat{\mathbf{q}}_k^+ = \hat{\mathbf{q}}_k^- + \frac{1}{2}\Xi(\hat{\mathbf{q}}_k^-)\delta\hat{\boldsymbol{\alpha}}_k^+$$
$$\hat{\boldsymbol{\beta}}_k^+ = \hat{\boldsymbol{\beta}}_k^- + \Delta\hat{\boldsymbol{\beta}}_k^+$$

- Discrete-Time Quaternion Propagation

$$\hat{\mathbf{q}}_{k+1}^- = \bar{\Omega}(\hat{\boldsymbol{\omega}}_k^+)\hat{\mathbf{q}}_k^+$$

$$\bar{\Omega}(\hat{\boldsymbol{\omega}}_k^+) \equiv \begin{bmatrix} \cos\left(\frac{1}{2}||\hat{\boldsymbol{\omega}}_k^+||\,\Delta t\right) I_{3\times 3} - \left[\hat{\boldsymbol{\psi}}_k^+\times\right] & \hat{\boldsymbol{\psi}}_k^+ \\ -\hat{\boldsymbol{\psi}}_k^{+T} & \cos\left(\frac{1}{2}||\hat{\boldsymbol{\omega}}_k^+||\,\Delta t\right) \end{bmatrix}$$

$$\hat{\boldsymbol{\psi}}_k^+ \equiv \frac{\sin\left(\frac{1}{2}||\hat{\boldsymbol{\omega}}_k^+||\,\Delta t\right)\hat{\boldsymbol{\omega}}_k^+}{||\hat{\boldsymbol{\omega}}_k^+|}$$

- Farrenkopf's Steady-State Analysis

$$\dot{\theta} = \tilde{\omega} - \beta - \eta_v$$
$$\dot{\beta} = \eta_u$$

$$S_u \equiv \sigma_u \Delta t^{3/2} \Big/ \sigma_n$$
$$S_v \equiv \sigma_v \Delta t^{1/2} \Big/ \sigma_n$$

$$\vartheta = \left[S_u^2(4 + S_v^2) + S_u^4/12\right]^{1/2}$$
$$\xi = -\frac{1}{2}\left[\left(\frac{S_u^2}{2} + \vartheta\right) + \sqrt{\left(\frac{S_u^2}{2} + \vartheta\right)^2 - 4S_u^2}\,\right]$$

$$p_{\theta\theta}^{-} = \sigma_n^2 \left[\left(\frac{\xi}{S_u} \right)^2 - 1 \right]$$

$$p_{\beta\beta}^{-} = \left(\frac{\sigma_n}{\Delta t} \right)^2 \left[S_u^2 \left(\frac{1}{\xi} + \frac{1}{2} \right) - \xi \right]$$

$$p_{\theta\theta}^{+} = \sigma_n^2 \left[1 - \left(\frac{S_u}{\xi} \right)^2 \right]$$

$$p_{\beta\beta}^{+} = \left(\frac{\sigma_n}{\Delta t} \right)^2 \left[S_u^2 \left(\frac{1}{\xi} - \frac{1}{2} \right) - \xi \right]$$

- Orbit Estimation

$$\ddot{\mathbf{r}}(t) = -\frac{\mu}{||\mathbf{r}(t)||^3} \mathbf{r}(t) + \mathbf{w}(t)$$

- The α-β Filter

$$\hat{r}_k^{+} = \hat{r}_k^{-} + \alpha \left[\tilde{y}_k - \hat{r}_k^{-} \right]$$

$$\dot{\hat{r}}_k^{+} = \dot{\hat{r}}_k^{-} + \frac{\beta}{\Delta t} \left[\tilde{y}_k - \hat{r}_k^{-} \right]$$

$$\hat{r}_{k+1}^{-} = \hat{r}_k^{+} + \dot{\hat{r}}_k^{+} \Delta t$$

$$\dot{\hat{r}}_{k+1}^{-} = \dot{\hat{r}}_k^{+}$$

$$S_q = q^{1/2} \Delta t^{3/2} \Big/ \sigma_n$$

$$\xi = \frac{1}{2} \left[\left(\frac{S_q^2}{2} + \vartheta \right) + \sqrt{\left(\frac{S_q^2}{2} + \vartheta \right)^2 - 4 S_q^2} \right]$$

$$\vartheta = \left[4 S_q^2 + \frac{S_q^4}{12} \right]^{1/2}$$

$$p_{rr}^{-} = \sigma_n^2 \left[\left(\frac{\xi}{S_q} \right)^2 - 1 \right]$$

$$p_{\dot{r}\dot{r}}^{-} = \left(\frac{\sigma_n}{\Delta t} \right)^2 \left[S_q^2 \left(\frac{1}{2} - \frac{1}{\xi} \right) + \xi \right]$$

$$p_{r\dot{r}}^{-} = \frac{\sigma_n^2 \xi}{\Delta t}$$

$$\frac{\beta^2}{1-\alpha} = S_q^2$$

$$\alpha = 1 - \left(\frac{S_q}{\xi}\right)^2$$

$$\beta = S_q\sqrt{1-\alpha}$$

$$\alpha = -\frac{1}{2}\beta + \frac{1}{2}\sqrt{\beta\left[(\beta/3)+8\right]}$$

- The α-β-γ Filter

$$\hat{r}_k^+ = \hat{r}_k^- + \alpha\left[\tilde{y}_k - \hat{r}_k^-\right]$$

$$\dot{\hat{r}}_k^+ = \dot{\hat{r}}_k^- + \frac{\beta}{\Delta t}\left[\tilde{y}_k - \hat{r}_k^-\right]$$

$$\ddot{\hat{r}}_k^+ = \ddot{\hat{r}}_k^- + \frac{\gamma}{2\Delta t^2}\left[\tilde{y}_k - \hat{r}_k^-\right]$$

$$\hat{r}_{k+1}^- = \hat{r}_k^+ + \dot{\hat{r}}_k^+\Delta t + \frac{1}{2}\ddot{\hat{r}}_k^+\Delta t^2$$

$$\dot{\hat{r}}_{k+1}^- = \dot{\hat{r}}_k^+ + \ddot{\hat{r}}_k^+\Delta t$$

$$\ddot{\hat{r}}_{k+1}^- = \ddot{\hat{r}}_k^+$$

- Smoothing with the Eigensystem Realization Algorithm

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix}\mathbf{u} + \begin{bmatrix} 0 \\ I \end{bmatrix}\mathbf{w}$$

$$\equiv F\mathbf{x} + B\mathbf{u} + G\mathbf{w}$$

$$\tilde{\mathbf{y}}_k = H\mathbf{x}_k + \mathbf{v}_k$$

$$\mathcal{K} = P_f^+\Phi^T(P_f^-)^{-1}$$

$$P = \mathcal{K}P\mathcal{K}^T + \left[P_f^+ - \mathcal{K}P_f^-\mathcal{K}^T\right]$$

Exercises

7.1 Write a general computer subroutine that converts a user's position from a known height, longitude, and latitude on the Earth to ECEF position coordinates. Also, write a general computer subroutine that converts ECEF position coordinates to height, longitude, and latitude.

7.2 Find a website that contains the orbital elements for every GPS satellite (as of this writing this website is given by http://www.navcen.uscg.gov/). Pick some user location of longitude and latitude on the Earth's surface. Convert the GPS orbital elements for each satellite into ECEF coordinates using the

methods of §3.8.2 and §7.1.1 Assuming that the current time is given by the time of applicability for the GPS satellite, use eqn. (7.10) to determine which GPS satellites are available at that time.

7.3 Using the initial ECI coordinates of each GPS satellite determined from the orbital elements obtained in exercise 7.2, propagate the position of each GPS satellite using eqn. (3.198). Also, at each time-step convert the ECI position vector for each GPS satellite into ECEF coordinates using eqn. (7.1).

7.4 Reproduce the results of the Kalman filter application to GPS in example 7.1. Adjust the process noise variance parameter and discuss its importance on the filtering performance. Try various trajectory motions and speed of the vehicles (e.g., circular motion instead of due southerly motion as shown in the example). Also, artificially reduce down the available number of GPS satellites to three for some time period. How does your EKF design work during this period?

7.5 A simple approach for GPS position and velocity estimation involves using a linear Kalman filter with position "measurements" given from the nonlinear least squares estimates of §4.1. The Kalman filter model now involves only the first six states of the model shown in eqn. (7.11a), since the clock bias is estimated in the least squares solution. The output is now linear in this formulation with $H = \begin{bmatrix} I & 0 \end{bmatrix}$. The measurement covariance in the Kalman filter is given by the first three rows and three columns of error-covariance, P, in eqn. (4.5). Using the simulation parameters shown in example 7.1, test the performance of this simple linear Kalman filter. Compare your results to the results obtained by the full EKF formulation. Discuss the disadvantages of the linear Kalman filter (if any) over the EKF formulation.

7.6 Starting with eqn. (7.25) prove that eqn. (7.27) is indeed correct.

7.7 Show that the second-order errors in eqn. (7.47) are small only if $\delta\hat{\boldsymbol{\alpha}}_k^+$ is small.

7.8 Show that following estimated error angle, defined in §7.2.1, propagation equation is valid up to second order:

$$\delta\dot{\boldsymbol{\alpha}} = -[\hat{\boldsymbol{\omega}}\times]\delta\boldsymbol{\alpha} + \delta\boldsymbol{\omega} - \frac{1}{2}\delta\boldsymbol{\omega}\times\delta\boldsymbol{\alpha}$$

7.9 Reproduce the results of example 7.2. Use the discrete-time propagation for the quaternion in eqn. (7.53) and covariance in eqn. (7.57). Try various values for σ_u and σ_v to generate synthetic gyro measurements, and discuss the performance of the extended Kalman filter under these variations. What parameter, σ_u or σ_v, seems to have the largest effect on the filter's performance?

7.10 Using the same procedure used to derive the eqn. (7.52), fully derive the state transition matrix in eqn. (7.59).

7.11 Fully derive the expressions shown in eqns. (7.71) and (7.72).

7.12 Use Murrell's version shown in Figure 7.5 on the simulated measurements developed in exercise 7.9. Discuss the performance in terms of accuracy and computational savings of Murrell's approach over the standard extended Kalman filter.

7.13 Write a general computer subroutine that solves Farrenkopf's equations in §7.2.4. Discuss how Farrenkopf's equations can be used to provide an initial hardware design from a spacecraft's attitude knowledge requirements. Also, use eqns. (7.71) and (7.72) to assess the expected extended Kalman filter performance for variations in σ_u and σ_v as discussed in exercise 7.9.

7.14 ♣ The extended Kalman filter for attitude estimation in Table 7.1 uses vector observations as measurements. Modify this algorithm to handle the case of quaternion measurements directly (hint: define an error quaternion between the measured quaternion and estimated quaternion).

7.15 Consider the problem of GPS spacecraft attitude estimation using phase difference measurements, as discussed in exercise 4.14. Pick a known position of a low-Earth orbiting spacecraft and simulate the availability of the GPS satellites at that position. Assume that a suitable elevation angle cut-off for the GPS availability in low-Earth orbit is 0 degrees. Generate an Earth-pointing motion in the spacecraft with a true attitude motion given by a constant angular velocity about the y-axis with $\boldsymbol{\omega} = \begin{bmatrix} 0 & -0.0011 & 0 \end{bmatrix}^T$ rad /sec. Assume that the inertia matrix of the spacecraft is given by

$$J = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 120 & 0 \\ 0 & 0 & 90 \end{bmatrix} \quad \text{Nms}$$

Using the dynamics model in eqn. (3.184b) an "open-loop" control input is given by

$$\mathbf{L} = -[\boldsymbol{\omega}\times]J\boldsymbol{\omega}$$

Pick a set of three baseline vectors and generate synthetic phase measurements using a standard deviation of $\sigma = 0.001$ for each measurements. Rederive the extended Kalman filter for attitude estimation, shown in §7.2.1, using the dynamics-based model instead of gyros. Use this filter to estimate the attitude of the vehicle from the GPS measurements and known control-torque input. Simulate process noise errors by varying the true value of J slightly, and tune the process noise covariance until reasonable results are obtained.

7.16 Consider the problem of determining the position and orientation of a vehicle using line-of-sight measurements from a vision-based beacon system based on Position Sensing Diode (PSD) technology, as shown in exercise 4.3. Develop an extended Kalman filter for this problem using the following

state model:

$$\dot{\mathbf{q}} = \frac{1}{2}\Xi(\mathbf{q})\boldsymbol{\omega}$$
$$\dot{\boldsymbol{\omega}} = \mathbf{w}_\omega$$
$$\dot{\mathbf{p}} = \mathbf{v}$$
$$\dot{\mathbf{v}} = \mathbf{w}_v$$

where $\mathbf{q}$ is the quaternion, ω is the angular velocity, $\mathbf{p} = \begin{bmatrix} X_c & Y_c & Z_c \end{bmatrix}^T$ is the position vector of the unknown object, and $\mathbf{v}$ is the velocity vector. The variables $\mathbf{w}_\omega$ and $\mathbf{w}_v$ are process noise vectors. Use the multiplicative error-quaternion approach of §7.2.1 to develop a 12^{th}-order reduced state vector. Use the simulation parameters discussed in exercise 4.3 to test the performance of your EKF algorithm. Tune your filter design by varying the process noise covariance associated with the vectors $\mathbf{w}_\omega$ and $\mathbf{w}_v$. Once the filter is properly tuned, reduce the number of beacons seen by the sensor to 2 beacons. For example, from time period 300 to 500 seconds use measurements from only the first two beacons to update the state in the EKF. Assess and discuss the performance of the estimated quantities during this period.

7.17 Consider the problem of estimating the state (position, $\mathbf{r}$, and velocity, $\dot{\mathbf{r}}$) and drag parameter of a vehicle at launch, as shown in exercise 4.17. Develop a 7-state extended Kalman filter for this problem using the following state model:

$$\ddot{x} = -p\,\dot{x}\,V + w_x$$
$$\ddot{y} = -p\,\dot{y}\,V + w_y$$
$$\ddot{z} = -g - p\,\dot{z}\,V + w_z$$
$$\dot{p} = w_p$$

where w_x, w_y, w_z, and w_p are process noise terms. Use the simulation parameters discussed in exercise 4.17 to test the performance of your EKF algorithm. Tune your filter design by varying the process noise covariance parameters associated with w_x, w_y, w_z, and w_p. Also, use a fully discrete-time version of your filter (i.e., use a discrete-time propagation of the state model and error-covariance). Also, re-derive your algorithm using the following simplified model in the EKF:

$$\ddot{x} = w_x$$
$$\ddot{y} = w_y$$
$$\ddot{z} = -g + w_z$$

Can you achieve reasonable results using this approximate model that ignores the effect of drag on the system?

7.18 Reformulate the parameter identification problem of the coupled weakly nonlinear oscillators shown in exercise 4.20 using the Kalman filter approach discussed in §7.3. Compare the performance of the EKF versus the nonlinear least squares approach developed for exercise 4.20.

7.19 Reproduce the results of example 7.3. Compare your results to the Gaussian Least Squares Differential Correction (GLSDC) of §4.3 for various initial conditions errors. Does the EKF approach always converge in less iterations than the GLSDC?

7.20 ♣ Instead of the extended Kalman filter formulation for orbit estimation shown in §7.3, use the Unscented filter (UF) of §5.7.6 to perform the iterations. Can you achieve better performance capabilities using the UF over the EKF for various initial conditions?

7.21 The orbit navigation problem involves estimating the position and velocity of the spacecraft in real time using an extended Kalman filter. Program a navigation filter where the true orbit trajectory is determined with a nonzero process noise in eqn. (7.75). Use GPS pseudorange measurements sampled at 1-second intervals from the to-be-determined spacecraft to the GPS satellites (assume that the spacecraft is in low-Earth orbit). Assume that a suitable elevation angle cutoff for the GPS availability in low-Earth orbit is 0 degrees. Discuss the performance of the navigation filter as the measurement sampling interval increases.

7.22 Fully derive the relationship shown in eqn. (7.92).

7.23 Using the model in eqn. (7.93) derive analytical expressions for the tracking index and error-covariance matrix. Also, derive a similar expression as shown in eqn. (7.107) for the relationship between α and β. How does this model simplify the analysis?

7.24 Assume that no process noise is given in the model described in eqn. (7.76). Therefore, the discrete-time model is simply given by

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_k \tag{7.144}$$

$$\tilde{y}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_k + v_k \tag{7.145}$$

Assuming that no *a priori* information exists, so that $P_0 = \infty$, which corresponds to maximum likelihood estimation, show that the filter gains are given by the following expressions:

$$\alpha_k = \frac{2(2k-1)}{k(k+1)} \tag{7.146}$$

$$\beta_k = \frac{6}{k(k+1)} \tag{7.147}$$

Discuss the significance of these gains as k increases.

7.25 Prove that the only solution that makes β valid in eqn. (7.92) is given by eqn. (7.109).

7.26 ♣ Analytically prove the stability bounds for α, β, and γ shown in eqn. (7.124) are correct.

7.27 Reproduce the results of example 7.4. Try various values for the process noise parameter in each filter, and discuss the robustness of the estimated results to variations in this parameter. Also, perform an assessment on the computation complexity (e.g., the number of Floating Point Operations) of the α-β-γ filter versus the α-β filter.

7.28 From the simulation performed in exercise 7.27, suppose we ignore the relationship between α and β in the α-β filter. Try tuning them separately. We know that this approach ignores the kinematical relationship inherent in the assumed model, but can you achieve better results than the results shown in example 7.4? Also, try varying α, β, and γ independently in the α-β-γ filter.

7.29 Suppose that an acceleration measurement is also available for the system described in example 7.4. Use an acceleration measurement with a standard deviation of 0.1 m/sec^2 in an acceleration-based Kalman filter. The state model is still given by eqn. (7.117), but the observation vector is now given by

$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k \equiv H\mathbf{x}_k$$

Derive a linear Kalman filter with this new observation model. Using the same value for q as in example 7.4, compare the performance of the α-β-γ filter versus this new filter. Also, try increasing the standard deviation of the acceleration measurement error and re-evaluate the performance of the new filter. At what value of this standard deviation does the acceleration measurement become practically useless?

7.30 Consider the nonlinear equations of motion for a highly maneuverable aircraft, as shown in exercise 4.22. Using a known "rich" input for δ_E, create synthetic measurements of the angle of attack α and pitch angle θ with zero initial conditions, as discussed in exercise 4.22. Use the extended Kalman filter to perform two tasks:

(A) Filter the measurements in the system by varying some of the coefficients in the assumed EKF model, using process noise to compensate for this error.

(B) Perform real-time estimation of some of the parametric values associated with the dynamic model. For example, try to estimate the true value (-4.208) associated with α in the differential equation for the pitch angle. Use the methods of §7.4.3 to develop your estimation algorithm. Try estimating other parameters as well.

7.31 Reproduce the results of example 7.5. How sensitive is this filter to variations in the initial state conditions and the initial error-covariance? Try estimating other parameters such as C_{D_α}, C_{L_α}, and C_{m_α}. Derive analytical expressions for the partial derivatives for these new parameters. Compare your EKF results to the results obtained in the nonlinear least squares approach, as shown in example 4.4.

7.32 Implement a nonlinear RTS smoother, shown in Table 6.5, to the simulation

performed in exercise 7.31. Discuss the performance enhancement capabilities of the smoother over the EKF.

7.33 Suppose that the model shown in §7.4.3 is used strictly to filter the noisy measurement and for real-time navigation purposes. Use only an 8-state EKF design with states given by v_1, v_3, ω_2, θ, x, and z. The position components follow:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} v_1 \\ v_3 \end{bmatrix}$$

The measurement model is now given by

$$\tilde{\mathbf{y}} = \begin{bmatrix} \alpha \\ ||\mathbf{v}|| \\ \omega_2 \\ \theta \\ ||\mathbf{r}|| \end{bmatrix} + \mathbf{v}$$

where $\mathbf{r} = \begin{bmatrix} x & z \end{bmatrix}^T$. Assume that the standard deviation of the measurement error associated with $||\mathbf{r}||$ is given by 10 m. Design an extended Kalman filter to track the position of the aircraft using the simulation parameters shown in example 7.5. Vary some of the coefficients in the assumed EKF dynamics model, and use process noise to compensate for this error. Also, implement an α-β-γ filter with measurements of $||\mathbf{r}||$ only. How do the results using a full dynamics-based model in an EKF compare to the results obtained by the simple α-β-γ filter?

7.34 ♣ Instead of the extended Kalman filter formulation for aircraft parameter estimation shown in §7.4.3, use the Unscented filter (UF) of §5.7.6 to perform the parameter estimation. Can you achieve better performance capabilities using the UF over the EKF for various initial condition and error-covariance errors?

7.35 Reproduce the results of the combined RTS/ERA results shown in example 7.6. Try various noise levels in the synthetic measurements and assess the value of using an RTS smoother as a "pre-filter" to the ERA.

7.36 Using the same simulation parameters shown in example 7.6, implement only the forward-time Kalman filter estimates in the ERA to realize the state model. How do the Kalman filter estimates combined with the ERA compare with the results obtained by the combined RTS/ERA approach? Try various noise levels in the synthetic measurements.

7.37 Instead of using the ERA approach to realize a state model, suppose we use the ARMA model instead, shown in exercise 1.13. Choose some simple second-order model with a significantly "rich" input and use a sequential version of the ARMA model to estimate the parameters of your chosen model. Add a significant amount of noise to the y_k and check the performance of your sequential estimator. Implement a simple linear Kalman filter with some assumed model to pre-filter the measurements before they are

used in the sequential ARMA estimator. Finally, ignore the ARMA model estimator approach altogether and use the Kalman filter to directly estimate the coefficients by appending the state vector. Discuss the accuracy and computational requirements of each approach for various noise levels in the synthetic measurements.

References

[1] Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J., *GPS: Theory and Practice*, Springer Wien, New York, NY, 5th ed., 2001.

[2] Bate, R.R., Mueller, D.D., and White, J.E., *Fundamentals of Astrodynamics*, Dover Publications, New York, NY, 1971.

[3] Wertz, J.R., "Space-Based Orbit, Attitude and Timing Systems," *Mission Geometry: Orbit and Constellation Design and Management*, chap. 4, Microcosm Press, El Segundo, CA and Kluwer Academic Publishers, The Netherlands, 2001.

[4] Meeus, J., *Astronomical Algorithms*, Willman-Bell, Inc., Richmond, VA, 2nd ed., 1999.

[5] Spilker, J.J., "GPS Navigation Data," *Global Positioning System: Theory and Applications*, edited by B. Parkinson and J. Spilker, Vol. 64 of *Progress in Astronautics and Aeronautics*, chap. 4, American Institute of Aeronautics and Astronautics, Washington, DC, 1996.

[6] Farrell, J. and Barth, M., *The Global Positioning System & Inertial Navigation*, McGraw-Hill, New York, NY, 1998.

[7] Grewal, M.S., Weill, L.R., and Andrews, A.P., *Global Positioning Systems, Inertial Navigation, and Integration*, John Wiley & Sons, New York, NY, 2001.

[8] Farrell, J.L., "Attitude Determination by Kalman Filter," *Automatica*, Vol. 6, No. 5, 1970, pp. 419–430.

[9] Lefferts, E.J., Markley, F.L., and Shuster, M.D., "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, Sept.-Oct. 1982, pp. 417–429.

[10] Crassidis, J.L. and Markley, F.L., "Attitude Estimation Using Modified Rodrigues Parameters," *Proceedings of the Flight Mechanics/Estimation Theory Symposium*, NASA-Goddard Space Flight Center, Greenbelt, MD, May 1996, pp. 71–83.

[11] Pittelkau, M.E., "Spacecraft Attitude Determination Using the Bortz Equation," *AAS/AIAA Astrodynamics Specialist Conference*, Quebec City, Quebec, Aug. 2001, AAS 01-310.

[12] Farrenkopf, R.L., "Analytic Steady-State Accuracy Solutions for Two Common Spacecraft Attitude Estimators," *Journal of Guidance and Control*, Vol. 1, No. 4, July-Aug. 1978, pp. 282–284.

[13] Markley, F.L., "Matrix and Vector Algebra," *Spacecraft Attitude Determination and Control*, edited by J.R. Wertz, appendix C, Kluwer Academic Publishers, The Netherlands, 1978.

[14] Murrell, J.W., "Precision Attitude Determination for Multimission Spacecraft," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Palo Alto, CA, Aug. 1978, pp. 70–87.

[15] Andrews, S. and Bilanow, S., "Recent Flight Results of the TRMM Kalman Filter," *AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, Aug. 2002, AIAA-2002-5047.

[16] Crassidis, J.L. and Markley, F.L., "Unscented Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 4, July-Aug. 2003, pp. 536–542.

[17] Yunck, T.P., "Orbit Determination," *Global Positioning System: Theory and Applications*, edited by B. Parkinson and J. Spilker, Vol. 164 of *Progress in Astronautics and Aeronautics*, chap. 21, American Institute of Aeronautics and Astronautics, Washington, DC, 1996.

[18] Brookner, E., *Tracking and Kalman Filtering Made Easy*, John Wiley & Sons, New York, NY, 1998.

[19] Bar-Shalom, Y. and Fortmann, T.E., *Tracking and Data Association*, Academic Press, Boston, MA, 1988.

[20] Kalata, P.R., "The Tracking Index: A Generalized Parameter for α-β and α-β-γ Target Trackers," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-20, No. 2, March 1984, pp. 174–182.

[21] Åström, K.J. and Wittenmark, B., *Computer-Controlled Systems*, Prentice Hall, Upper Saddle River, NJ, 3rd ed., 1997.

[22] Tenne, D. and Singh, T., "Characterizing Performance of α-β-γ Filters," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-38, No. 3, July 2002, pp. 1072–1087.

[23] Mook, D.J. and Shyu, I.M., "Nonlinear Aircraft Tracking Filter Utilizing Control Variable Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 1, Jan.-Feb. 1992, pp. 228–237.

[24] Crassidis, J.L., Mook, D.J., and McGrath, J.M., "Automatic Carrier Landing System Utilizing Aircraft Sensors," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 5, Sept.-Oct. 1993, pp. 914–921.

[25] Roemer, M.J. and Mook, D.J., "Enhanced Realization/Identification of Physical Modes," *Journal of Aerospace Engineering*, Vol. 3, No. 2, April 1990, pp. 128–139.

[26] Meirovitch, L., *Principles and Techniques of Vibrations*, Prentice Hall, Upper Saddle River, NJ, 1997.

[27] Hashemipour, H.R. and Laub, A.J., "Kalman Filtering for Second-Order Models," *Journal of Guidance, Control, and Dynamics*, Vol. 11, No. 2, March-April 1988, pp. 181–186.

[28] Crassidis, J.L. and Mook, D.J., "Integrated Estimation/Identification Using Second-Order Dynamic Models," *Journal of Vibration and Acoustics*, Vol. 119, No. 1, Jan. 1997, pp. 1–8.

[29] Rogers, R.M., *Applied Mathematics in Integrated Navigation Systems*, American Institute of Aeronautics and Astronautics, Inc., Reston, VA, 2000.