8

NONLINEAR APPROXIMATIONS

> I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail
>
> —Abraham H. Maslow
> The Psychology of Science: A Reconnaissance, Harper, 1966.

8.1 CHAPTER FOCUS

Although the Kalman filter (our hammer) was developed to nail the *linear* least-mean-squares estimation problem, it has since been applied with impunity—and considerable success—to a variety of important less-than-linear problems. As was mentioned in Chapter 3, the introduction of nonlinearity in the dynamic or measurement model corrupts the standard (linear) propagation of the mean and covariance by coupling *unknown* higher order moments into the mean and covariance.

8.1.1 Topics to Be Covered

Methods proposed for coping with such nonlinearities generally fall into four broad categories:

I. Extension to affine dynamic and measurement models, which are equivalent to the standard Kalman filter model with nonzero-mean dynamic disturbances and/or measurement noise. The solution is trivial and exact.

Kalman Filtering: Theory and Practice Using MATLAB®, Fourth Edition.
Mohinder S. Grewal and Angus P. Andrews.

II. Linear approximation of

A. The variation of measurement variables with changes in state variables, usually by partial derivatives which may be either

1. Analytical, by taking partial derivatives of differentiable nonlinear functions $z = h(x)$, or
2. Numerical, by evaluating $h(x)$ at the estimated state $h(\hat{x})$ and at "perturbations" $h(\hat{x} + \delta_j)$, and computing the ratios

$$\left.\frac{\partial z_i}{\partial x_j}\right|_{x=\hat{x}} \approx \frac{h_i(\hat{x} + \delta_j) - h_i(\hat{x})}{\{\delta_j\}_j},$$

where $h_i(\cdot)$ is the ith component of h and the components of δ_j are nonzero only in the jth row. Then the linearized approximation of the measurement sensitivity matrix H will have the numerical approximation in the ith row and jth column.

In either case, these approximations of H are only used for computing the Kalman gain. The expected measurement is always computed as

$$\hat{z} = h(\hat{x}).$$

B. Covariance dynamics, again by using either analytical or numerical partial differentiation for the purpose of propagating the covariance matrix of state estimation uncertainty by any of the four possible methods:

1. If the nonlinear dynamic model has the form

$$\dot{x} = f(x),$$

use the analytical partial derivative

$$F = \frac{\partial f}{\partial x}$$

to propagate P using

$$\dot{P} = FP + PF^{\mathrm{T}} + Q(t).$$

2. Use the value of F from (II.B.1) to compute

$$\Phi_{k-1} \approx \exp\left(\int_{t_{k-1}}^{t_k} F(s)\, ds\right).$$

3. Approximate Φ by propagating $\dot{x} = f(x)$ from t_{k-1} to t_k for $n + 1$ different initial values of x, including $\hat{x}_{(k-1)(+)}$ and perturbations δ_j along each of the n state vector components. Then

$$\{\Phi_{k-1}\}_{ij} \approx \frac{\{x_{kj} - \hat{x}_{k(-)}\}_i}{\delta_j},$$

where x_{kj} is the perturbed solution at time t_k for the trajectory with initial value $\hat{x}_{(k-1)(+)} + \delta_j$.

4. If the nonlinear model has the form

$$x_k = f_k(x_{k-1}),$$

then the linear approximation of the state-transition matrix is the Jacobian matrix

$$\Phi_{k-1} \approx \frac{\partial f_k}{\partial x_{k-1}}.$$

In all cases, these approximations are only for propagating the covariance matrix P, not the estimate $\hat{x}$. The full nonlinear model is always used for propagating $\hat{x}$.

III. Replacing the matrix Riccati differential or difference equation with statistical approximations using discrete samples. These "sample-and-propagate" methods are similar in some ways to (II.B.3), except that they make more intelligent use of perturbations, and can carry the covariance solutions all the way through the measurements. They perform better than linearization methods for some applications—but they, too, have their limitations.

IV. Deriving something roughly equivalent to the Kalman filter, except with some limited model nonlinearities. These have included

A. A derivation including terms out to second order by Bass *et al.* [1].

B. A derivation including terms out to third order by Wiberg and Campbell [2].

C. A derivation by Beneš [3] with limited nonlinearities, but with a closed-form solution.

The computational complexities of the second- and third-order derivations are generally too extreme to justify their limited advantages, and they are only mentioned here. The Beneš filter will be described here, although it has not performed well against filters from Categories II or III in some studies [29].

Beyond approaches to modifying the Kalman filter implementation to better cope with model nonlinearities, there has been a long history of nonlinear stochastic system modeling. Although it has perhaps not produced anything more "snappy" than the Kalman filter, it has had some notable successes in modeling the behavior of thermodynamic systems, economic systems, and market systems—among others. It has also produced some "grid-based" implementations that have been used for computing probability densities for surveillance (detection and tracking) problems.

The approximation approaches generally depend on the fact that the Kalman filter is optimal for its model of the problem. Although some might expect optimal solutions of this sort to be brittle in some way, it is generally not the case. Optimizing solutions with respect to quadratic criteria such as mean-squared error tends to make derivatives of their first-order performance with respect to model variations zero, so that minor modifications of the model can have acceptable influences on optimality.

This chapter is about some of the more successful approaches.

8.1.2 What Does "Nonlinear" Mean?

It has been said that *"Classification of mathematical problems as linear and nonlinear is like classification of the Universe as bananas and non-bananas,"* the point being that the vast majority of real-world mathematical problems are not linear in nature.

We then need to qualify what we mean by "nonlinear," which is perhaps too strong a term for the more successful nonlinear applications of Kalman filtering. There are just too many wild things parading as mathematical functions to include them all, and we have shown in Chapter 3 what happens to the propagation of the fundamental variables of Kalman filtering (means and covariances) when some rather bland nonlinearities are introduced.

Many of the so-called "nonlinear" problems on which these methods have been successful are perhaps better described as being "quasilinear," in that the nonlinear functions involved are locally dominated their first-order variations—at least within the variations expected by the probability distributions of the estimation errors and measurement errors. As a consequence, the errors introduced by linear approximation depend on the quality of the estimates and measurements, and the success of a "nonlinear" application of Kalman filtering depends only on the linearization error within ranges determined by estimation uncertainty.

8.2 THE AFFINE KALMAN FILTER

This is a trivial extension of the linear Kalman filter to what might be called an *affine Kalman filter.* As we have seen in Chapter 3, affine transformations are not exactly linear, but their effects on probability distributions are not much different from those of linear transformations—especially their effects on the means and covariances, which are not corrupted by other moments of the distribution. We mention it here because it is also equivalent to the standard linear Kalman filter model, except with nonzero-mean noise sources.

The Kalman filter with control inputs is essentially an affine Kalman filter.

8.2.1 Affine Model

Affine transformations are linear transformations with an added known and constant "bias,"

$$x_k = \Phi_{k-1} x_{k-1} + b_{k-1} + w_{k-1} \tag{8.1}$$

$$z_k = H_k x_k + d_k + v_k, \tag{8.2}$$

where b_{k-1} is a known n-vector, d_k is a known m-vector, and both $\{w_{k-1}\}$ and $\{v_k\}$ are zero-mean white noise sequences with known covariances Q_{k-1} and R_k, respectively.

8.2.2 Nonzero-Mean Noise Model

The affine model is indistinguishable from the nonzero-mean noise model in which

$$\underset{w}{\mathrm{E}}\langle w_{k-1}\rangle = b_{k-1} \tag{8.3}$$

$$\underset{w}{\mathrm{E}}\langle (w_{k-1} - b_{k-1})(w_{k-1} - b_{k-1})^{\mathrm{T}}\rangle = Q_{k-1} \tag{8.4}$$

$$\underset{v}{\mathrm{E}}\langle v_k\rangle = d_k \tag{8.5}$$

$$\underset{v}{\mathrm{E}}\langle (v_k - d_k)(v_k - d_k)^{\mathrm{T}}\rangle = R_k. \tag{8.6}$$

The extension of the Kalman filter to include nonzero-mean noise is therefore equivalent to the affine extension.

8.2.3 Affine Filter Implementation

The equivalent implementation of the affine Kalman filter is then

$$\hat{x}_{k(-)} = \Phi_{k-1}\hat{x}_{(k-1)(+)} + b_{k-1} \tag{8.7}$$

$$P_{k(-)} = \Phi_{k-1} P_{(k-1)(+)} \Phi_{k-1}^{\mathrm{T}} + Q_{k-1}$$

$$\hat{z}_k = H_k \hat{x}_{k(-)} + d_k \tag{8.8}$$

$$\overline{K}_k = P_{k(-)} H_k^{\mathrm{T}} [H_k P_{k(-)} H_k^{\mathrm{T}} + R_k]^{-1}$$

$$\hat{x}_{k(+)} = \hat{x}_{k(-)} + \overline{K}_k (z_k - \hat{z}_k) \tag{8.9}$$

$$P_{k(+)} = P_{k(-)} - \overline{K}_k H_k P_{k(-)},$$

which differs from the linear Kalman filter only in the numbered equations.

8.3 LINEAR APPROXIMATIONS OF NONLINEAR MODELS

8.3.1 Linearizing the Riccati Differential Equation

Given an estimation problem with possibly nonlinear state dynamic model and measurement model with zero-mean error sources

$$\dot{x} = f(x,t) + w(t) \tag{8.10}$$

$$Q(t) \overset{\text{def}}{=} \underset{w}{\mathrm{E}}\langle w(t) w^{\mathrm{T}} t\rangle$$

$$z_k = h_k(x(t_k)) + v_k \tag{8.11}$$

$$R_k \overset{\text{def}}{=} \underset{v}{\mathrm{E}}\langle v_k v_k^{\mathrm{T}}\rangle,$$

the state estimate $\hat{x}$ can be propagated between measurement epochs by integrating

$$\frac{d}{dt}\hat{x}(t) = f(\hat{x}(t), t) \tag{8.12}$$

and the predicted measurement can be computed as

$$\hat{z}_k = h_k(\hat{x}_{k(-)}). \tag{8.13}$$

However, there is no comparable general solution for nonlinear propagatiion of the covariance matrix P of estimation uncertainty—which is needed for computing the Kalman gain.

If the vector-valued functions $f(\cdot, t)$ and $h_k(\cdot)$ are sufficiently differentiable, then their Jacobian matrices

$$F(x,t) \overset{\text{def}}{=} \frac{\partial f(x,t)}{\partial x} \tag{8.14}$$

$$H_k(x) \overset{\text{def}}{=} \frac{\partial h_k(x)}{\partial x} \tag{8.15}$$

can be used as a linear approximation for propagating $\hat{x}$ and P as the solution to

$$\dot{P}(t) \approx F(\hat{x}(t), t)P(t) + P(t)F^{\mathrm{T}}(\hat{x}(t), t) + Q(t) \tag{8.16}$$

$$\overline{K}_k = P_{k(-)}H_k^{\mathrm{T}}(x)[H_k(x)P_{k(-)}H_k^{\mathrm{T}}(x) + R_k]^{-1} \tag{8.17}$$

$$\hat{x}_{k(+)} = \overline{K}_k[z_k - h(\hat{x}_{k(-)}, t_k)] \tag{8.18}$$

$$P_{k(+)} = P_{k(-)} - \overline{K}_k P_{k(-)} H_k(x). \tag{8.19}$$

This approach only works if the linear approximations are “close enough for all practical purposes”—a concept that will be made more rigorous in Section 8.3.4.

Equation 8.14 assumes that $f(x,t)$ is a differentiable function of x. If its derivative (Jacobean matrix) is a known matrix function of x and t, then that derivative function is $F(x,t)$.

If the function $f(x,t)$ of Equation 8.10 were linear, then it could be expressed as

$$f(x,t) = F(t)x \tag{8.20}$$

$$F(t) = \frac{\partial f(x,t)}{\partial x}. \tag{8.21}$$

Otherwise, the partial derivative with respect to x will still be a function of x. In that case, it is necessary to assume some value of x ($\hat{x}$ will do) in evaluating the partial derivative.

8.3.2 Approximating Φ with Numerical Partial Derivatives

Even if $f(x,t)$ is a differentiable function of x, it is sometimes more efficient to propagate the covariance matrix using a state-transition matrix Φ approximated by numerical partial differentiation. This approach was used in early analysis and implementation of space navigation and control, in which the dynamic equations are for free-fall under the gravitational influence of multiple massive bodies. The same algorithms developed to generate the trajectory of the estimate $\hat{x}$ could then be used to generate "nearby" trajectories defined by perturbations from the initial conditions of $\hat{x}$. If we let the perturbed initial conditions be

$$x_{(k-1)\,[\ell]} = \hat{x}_{k-1} + \delta_{\,[\ell]}, \tag{8.22}$$

where $\delta_{\,[\ell]}$ has zeros except in its ℓth component, then the respective trajectory solutions at time t_k can be used to approximate the state-transition matrix as

$$\Phi_k \approx \frac{\partial x_k}{\partial x_{k-1}} \tag{8.23}$$

$$\phi_{kij} \approx \frac{x_{(k-1)(-)\,[j]i} - \hat{x}_{k(-)i}}{x_{(k-1)(+)\,[j]\,j} - \hat{x}_{(k-1)(+)j}}. \tag{8.24}$$

Many of these perturbation techniques had already been developed before the Space Age, during which they were used for satellite tracking, satellite navigation, and targeting and guiding ballistic missiles. Major players in this effort included William H. Guier (1926–2011) and George C. Weiffenbach (1921–2003) at the Applied Physics Laboratory of Johns Hopkins University and J. Halcombe Laning (1920–2012) and Richard H. Battin at the MIT Instrumentation Laboratory.

8.3.3 Linearized and Extended Kalman Filters

8.3.3.1 Linearized Kalman Filtering The Apollo Project to send Americans to the surface of the moon and back was announced by President John Fitzgerald Kennedy before a joint session of Congress on May 25, 1961. At that time, Stanley F. Schmidt was already studying the associated navigation and guidance problem at the NASA Ames Research Center in Mountain View, California. It was already a standard practice to use numerical partial differentiation to linearize problems, and Schmidt had been the first to recognize the potential of Kalman filtering for this application. For that purpose, he had been directing feasibility studies using perturbations evaluated about a "nominal" earth-to-moon-and-return trajectory. That is, the partial derivative approximations

$$\Phi_k \approx \left. \frac{\partial x_k}{\partial x_{k-1}} \right|_{x=x_N} \tag{8.25}$$

$$H_k \approx \left. \frac{\partial h_k(x)}{\partial x} \right|_{x=x_N}, \tag{8.26}$$

where x_N denotes a nominal trajectory.

This approach is quite common in preliminary design studies, when the actual trajectories are not known precisely. With this technique, performance requirements for the Apollo navigation sensors (a "space sextant" mounted on an inertial platform) could be determined by varying the associated error covariances to determine the minimum sensor performance required for mission success.

These preliminary performance studies need not use the full Kalman filter, but only the covariance computations of the Riccati equation. Linearized Kalman filter models are used almost exclusively for solving Riccati equations, not for state estimation.

Covariance analysis using linearized models is also used for assessing the expected level of linearization errors (Section 8.3.4) from extended Kalman filtering.

8.3.3.2 Extended Kalman Filtering Stanley F. Schmidt was the first to propose that the linearization technique could be adapted for onboard navigation *by using partial derivatives evaluated at the current time and estimated state variable.* That is,

$$\Phi_k \approx \left. \frac{\partial x_k}{\partial x_{k-1}} \right|_{x=\hat{x}} \tag{8.27}$$

$$H_k(\hat{x}) \approx \left. \frac{\partial h_k(x)}{\partial x} \right|_{x=\hat{x}}, \tag{8.28}$$

where $\hat{x}$ denotes the estimated trajectory.

Schmidt called this the *extended Kalman filter* (EKF). Others at the time called it the *Kalman–Schmidt filter*.[1]

There are many ways in which the linear approximations of the linearized and EKFs can be implemented, however.

[1] A later innovation of Schmidt's would come to be called the *Schmidt–Kalman filter* (SKF).

Example 8.1 (Analytical Linear Approximation EKF for n−body Dynamics) For more than half a century, the navigation, tracking, and control of spacecraft has made use of linearized and/or extended Kalman filtering models for the gravitational dynamics of travel in "free space," where the gravitational acceleration of a body of relatively insignificant mass m located at X due to a body of significant mass M located at Y some distance R from X is given by Newton's third law as

$$\begin{aligned}\ddot{X} &= g(X)\\ &= \frac{GM(Y-X)}{R^3}\\ R &= [(Y_1-X_1)^2+(Y_2-X_2)^2+(Y_3-X_3)^2]^{1/2},\end{aligned}$$

Where G is the gravitational constant and X and Y are specified as inertial (i.e., nonrotating and nonaccelerating) coordinates.

The gravity gradient matrix for a single massive body is then expressible as an analytical partial derivative,

$$\frac{\partial\ddot{X}}{\partial X} = -\frac{GM}{R^3}I_3 - \frac{3GM}{R^5}(Y-X)(Y-X)^{\mathrm{T}}.$$

Because accelerations are additive, the net acceleration on the small body at X from n such massive bodies of masses M_i and located at Y_i is the sum,

$$\begin{aligned}\ddot{X} &= G\left[\sum_{i=1}^{N}\frac{M_i\left(Y_i-X\right)}{R_i^3}\right]\\ R_i &= |Y_i-X|,\end{aligned}\tag{8.29}$$

and the net gravity gradient

$$\frac{\partial\ddot{X}}{\partial X} = -G\left[\sum_{i=1}^{n}\frac{M_i}{R_i^3}\right]I_3 - 3G\left[\sum_{i=1}^{n}\frac{M_i}{R_i^5}\left(Y_i-X\right)(Y_i-X)^{\mathrm{T}}\right].\tag{8.30}$$

In this dynamic model, the vectors Y_i are known functions of time, and the influence of the mass m on their trajectories is considered to be insignificant.

One can then write a nonlinear state-space model for the dynamics of Equation 8.29 in terms of a 6-vector as

$$\begin{aligned}x(t) &\overset{\text{def}}{=} \begin{bmatrix}X(t)\\ \dot{X}(t)\end{bmatrix}\\ \dot{x}(t) &= f(x,t)\\ &= \begin{bmatrix}\dot{X}(t)\\ \ddot{X}(t)\end{bmatrix},\end{aligned}$$

and any initial value $x(t_0)$ can be propagated forward in time using Equations 8.29.

In the analytical EKF approximation, propagation of the covariance matrix P of x can use a first-order dynamic coefficient matrix approximated by partial differentiation of f with respect to x:

$$\begin{aligned} F(t) &\overset{\text{def}}{=} \left.\frac{\partial f}{\partial x}\right|_{\hat{x}} \\ &= \begin{bmatrix} 0 & I_3 \\ \left.\dfrac{\partial \ddot{X}}{\partial X}\right|_{\hat{x}(t)} & 0 \end{bmatrix}, \end{aligned}$$

which can be evaluated using Equation 8.30.

The resulting formula can be used to propagate the covariance matrix P in a couple of ways:[2]

$$\begin{aligned} \dot{P}(t) &= F(t)P(t) + P(t)F^{\mathrm{T}}(t) + Q(t), \text{ or} \\ P_{k(-)} &= \Phi_{k-1}P_{k(-)}\Phi_{k-1}^{\mathrm{T}} + Q_k \\ \Phi_{k-1} &\overset{\text{def}}{=} \exp\left[\int_{t_{k-1}}^{t_k} F(s)\,ds\right]. \end{aligned}$$

The EKF process flow using the first of these methods is diagrammed in Figure 8.1, where analytical partial derivatives are also used for approximating the measurement sensitivity matrix H_k.

Note, however, that the predicted state vector $\hat{x}_{k(-)}$ and measurement vector $\hat{z}_k$ are computed without linear approximation. That is the characteristic of all EKF implementations.

Example 8.2 (Numerical Linear Approximation EKF for n-body Dynamics) There is yet a third way to implement an EKF for the same nonlinear dynamics problem as Example 8.1, this time using numerical partial derivatives to estimate the state-transition matrix Φ.

In this implementation, Equation 8.29 is not only used for propagating the estimate $\hat{x}_{(k-1)(+)}$ to obtain $\hat{x}_{k(-)}$, but also for propagating six perturbed initial values

$$\begin{aligned} \mathcal{S}_j(t_{k-1}) &= \hat{x}_{(k-1)(+)} + \delta_j \\ \{\delta_j\}_i &= \begin{cases} \varepsilon_j \neq 0, & i = j \\ 0, & i \neq j \end{cases}, \end{aligned}$$

where $\{\delta_j\}_i$ denotes the ith component of δ_j.

If the trajectory with initial value $\mathcal{S}_j(t_{k-1})$ at time t_{k-1} has solution

$$\mathcal{S}_j(t_k) = \hat{x}_{k(-)} + \Delta_j$$

[2] These models show dynamic disturbance noise covariance $Q(t)$ or Q_k, although dynamic disturbance noise is insignificant for space trajectories.

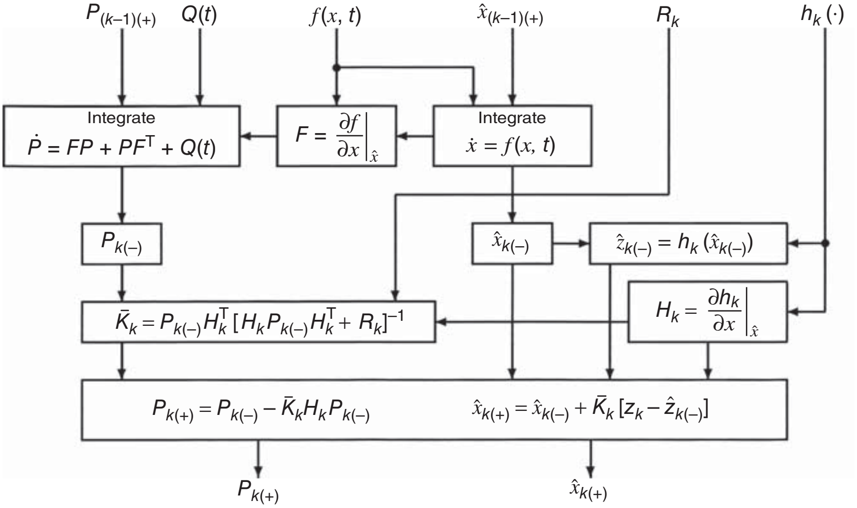
$P_{(k-1)(+)}$
$Q(t)$
$f(x, t)$
$\hat{x}_{(k-1)(+)}$
R_k
$h_k(\cdot)$
Integrate
$\dot{P} = FP + PF^T + Q(t)$
$F = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}}$
Integrate
$\dot{x} = f(x, t)$
$P_{k(-)}$
$\hat{x}_{k(-)}$
$\hat{z}_{k(-)} = h_k(\hat{x}_{k(-)})$
$\bar{K}_k = P_{k(-)}H_k^T[H_kP_{k(-)}H_k^T + R_k]^{-1}$
$H_k = \left.\frac{\partial h_k}{\partial x}\right|_{\hat{x}}$
$P_{k(+)} = P_{k(-)} - \bar{K}_kH_kP_{k(-)}$
$\hat{x}_{k(+)} = \hat{x}_{k(-)} + \bar{K}_k[z_k - \hat{z}_{k(-)}]$
$P_{k(+)}$
$\hat{x}_{k(+)}$

Figure 8.1 EKF process flow using analytical partial derivatives.

at time t_k, then the numerical partial derivative approximation of Φ_{k-1} will have the value

$$\phi_{ij} \approx \frac{\{\Delta_j\}_i}{\varepsilon_j}$$

in its ith row and jth column, where $\{\Delta_j\}_i$ is the ith component of the vector Δ_j.

The EKF process flow using this approach is diagrammed in Figure 8.2. This resembles, in some ways, that of the unscented Kalman filter (UKF) diagrammed in Figure 8.19.

Example 8.3 (Analytical Linear Approximation for the 2−body Problem) Not all linearized or EKF applications require integrating the differential equations of a dynamic model for propagating the trajectory solution forward in time. In those cases for which a closed-form analytical solution already exists, an approximation of the state-transition matrix Φ can be obtained by taking analytical partial derivatives of the closed-form "initial value" solution with respect to initial conditions.

In the case of the two-body problem, such as that for an orbit about the sun—neglecting the minor influences of the other planets—the solution is provided by Kepler's equations, which can be manipulated to yield the position and velocity trajectory as a function of time and initial conditions. The result is a set of formulas for $x(t_k)$ as a function of $x(t_{k-1})$, which is sufficient for propagating the estimate. It also provides an approximation for the linearized state-transition matrix Φ_{k-1} as the Jacobian matrix

$$\Phi_{k-1} \approx \left.\frac{\partial x(t_k)}{\partial x(t_{k-1})}\right|_{\hat{x}_{(k-1)(+)}}.$$

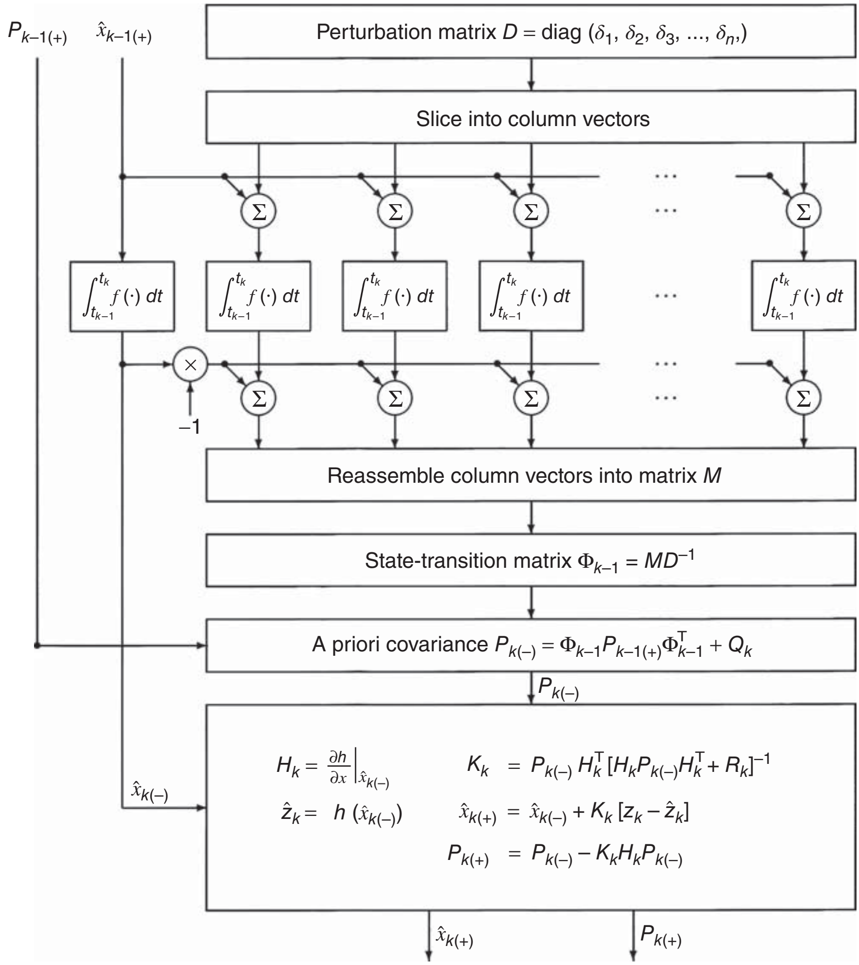
$P_{k-1(+)}$
$\hat{x}_{k-1(+)}$
Perturbation matrix $D = \text{diag}(\delta_1, \delta_2, \delta_3, \dots, \delta_n,)$
Slice into column vectors
Σ
$\int_{t_{k-1}}^{t_k} f(\cdot)\, dt$
$\times$
-1
Reassemble column vectors into matrix M
State-transition matrix $\Phi_{k-1} = MD^{-1}$
A priori covariance $P_{k(-)} = \Phi_{k-1}P_{k-1(+)}\Phi_{k-1}^{\mathrm{T}} + Q_k$
$P_{k(-)}$
$\hat{x}_{k(-)}$
$H_k = \frac{\partial h}{\partial x}\Big|_{\hat{x}_{k(-)}}$
$\hat{z}_k = h(\hat{x}_{k(-)})$
$K_k = P_{k(-)} H_k^{\mathrm{T}} [H_k P_{k(-)} H_k^{\mathrm{T}} + R_k]^{-1}$
$\hat{x}_{k(+)} = \hat{x}_{k(-)} + K_k [z_k - \hat{z}_k]$
$P_{k(+)} = P_{k(-)} - K_k H_k P_{k(-)}$
$\hat{x}_{k(+)}$
$P_{k(+)}$

Figure 8.2 EKF data flow using numerical differentiation of $\dot{x} = f(x, t)$.

For the case with $n = 2$ bodies, this provides a fourth alternative EKF solution for the same problem as Examples 8.1 and 8.2. However, this closed-form solution does not provide adequately accurate EKF estimates for high precision applications such as estimating the emphemerides of Global Navigation Satellite System (GNSS) satellites or ICBM guidance. These applications cannot ignore the nonspherical anomalies of Earth's gravitational field or the gravitational effects of the sun and moon.

Example 8.4 (Dynamic Model Parameter Estimation) Estimating the parameters of a Kalman filtering model is a notoriously nonlinear problem, but a fairly common one. This example uses the linear damped harmonic oscillator model from Example 5.7, and as a nonlinear model parameter estimation problem, assuming that

ζ (damping coefficient) is an unknown constant. Therefore, the damping coefficient can be modeled as a state variable and its value estimated using an EKF.

Let

$$x_3(t) = \zeta$$

and

$$\dot{x}_3(t) = 0.$$

Then the otherwise linear dynamic equation in continuous time becomes nonlinear:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} x_2 \\ -\omega^2 x_1 - 2x_2x_3 \quad \omega \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} w(t) + \begin{bmatrix} 0 \\ 12 \\ 0 \end{bmatrix}.$$

The observation equation is still linear, however:

$$z(t) = x_1(t) + v(t)$$

One hundred data points were simulated with random plant noise and measurement noise, $\zeta = 0.1, \omega = 10\,\text{rad/s}$, and initial conditions

$$\begin{bmatrix} x^1(0) \\ x^2(0) \\ x^3(0) \end{bmatrix} = \begin{bmatrix} 0\ \text{ft} \\ 0\ \text{ft/s} \\ 0 \end{bmatrix}, \qquad P(0) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix},$$

$$Q = 4.47(\text{ft/s})^2, \qquad R = 0.001(\text{ft}^2).$$

The discrete nonlinear plant and linear observation equations for this model are

$$\begin{aligned} x_k^1 &= x_{(k-1)}^1 + Tx_{(k-1)}^2 \\ x_k^2 &= -25Tx_{(k-1)}^1 + (1 - 10Tx_{(k-1)}^3)x_{(k-1)}^2 + 12T + Tw_{k-1} \\ x_k^3 &= x_{(k-1)}^3 \\ z_k &= x_k^1 + v_k. \end{aligned}$$

Figure 8.3 shows the estimated position, velocity, and damping factor states using an EKF. This is implemented in MATLAB®script `DampParamEst.m`, which runs successive calls with independent random samples. (Because this is a Monte Carlo simulation, successive calls will not produce the same results.)

Example 8.5 (Nonlinear Freeway Traffic Modeling with EKF) This is an application of extended Kalman filtering to estimating parameters of an already nonlinear dynamic model. The objective of this particular task was to estimate certain key parameters of a macroscopic freeway traffic model designed to simplify a driver-level "microscopic" model [4].

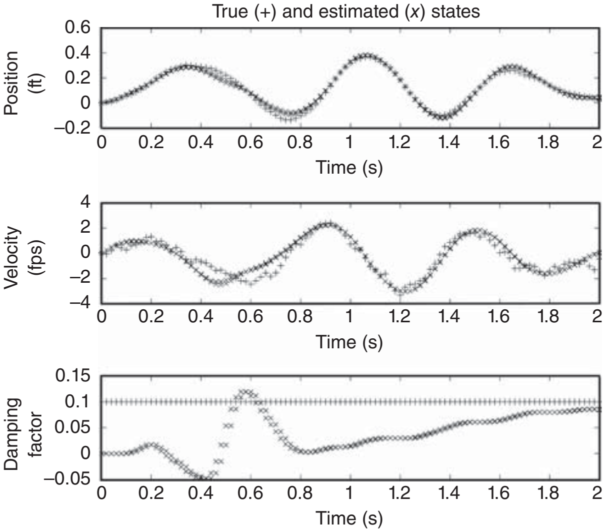

Figure 8.3 EKF estimation of oscillator position, velocity, and damping factor.

Macroscopic-Level Freeway Traffic Modeling: This particular application was designed to represent freeway traffic dynamics in terms of the speed and traffic densities within a 42-mile-long freeway circuit through Los Angeles and Long Beach, California. This was an early application of modern estimation and control methods to the problems of improving performance of urban infrastructure. Traffic on freeways had increased to the point that their carrying capacity was diminished, and something had to be done to prevent gridlock. Controlling the rate of access at on-ramps was the preferred method, but it had to be coupled with a reasonably accurate dynamic model for determining how access control influences traffic dynamics, and the model had to represent how real drivers respond to changing traffic conditions.

"Microscopic" vehicle-level simulations with driver response models developed for offline traffic simulations were good at representing actual freeway traffic dynamics, but were not fast enough for real-time controller-in-the-loop implementation. What was needed was a system-level "macroscopic" dynamic model that could be abstracted from vehicle-level dynamical behavior and used as part of real-time traffic control. The resulting model includes parameters that would need to be tuned to agree with results of the microscopic offline vehicle-level simulations.

For modeling purposes, a representative freeway circuit was divided into a sequence of contiguous segments, as illustrated in Figure 8.4.[3] Traffic dynamics along a stretch of freeway with N segments require N such state variables, x_j; $j = 1, 2, 3, \cdots, N$, whereas the microscopic model might have thousands of

[3] The segments shown in the figure are short, for illustrative purposes only. Segment lengths were more typically in the order of a mile or less.

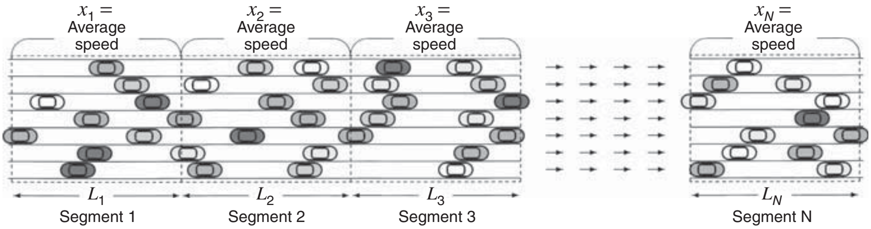

Figure 8.4 Freeway traffic model geometry (not to scale).

simulated vehicles. These segments may also have different lengths, modeled by the parameters L_j representing the lengths of the numbered segments. Generally, segment lengths are chosen so that driver responses are determined by conditions within a segment and in the segment ahead.

Nonlinear Dynamic Model: The core state variables are the average traffic speeds within the segments, as illustrated in the figure. The traffic dynamic model included driver responses to local conditions (speed, density, and their longitudinal gradients) within the various segments, including small random excursions due to lags in driver perceptions and actions.

The resulting discrete-time traffic dynamic model has the abstract form

$$x_k = \phi(x_{k-1}, p_1, \dots, p_4) + w_{k-1}, \quad w_{k-1} \in \mathcal{N}(0, Q),$$

where the p_i are four unknown parameters and w_{k-1} is a zero-mean Gaussian white noise sequence with known covariance Q. The jth component of the dynamic function ϕ is

$$\begin{aligned} &\phi_j(x_{k-1}, p_1, \dots, p_4) \\ &\quad \overset{\text{def}}{=} x_{(k-1),\,j} + \Delta t \left\{ -x_{(k-1),j} \frac{x_{(k-1),j} - x_{(k-1)(j-1)}}{L_j} \text{(speed gradient)} \right. \\ &\qquad\qquad - p_1 [x_{(k-1),\,j} - \underbrace{(p_2 + p_3 \rho_{(k-1),j})}_{\text{(equilibrium speed)}}] \\ &\qquad\qquad \left. + p_4 \left[\frac{(\rho_{(k-1)(j+1)} - \rho_{(k-1),j})}{\rho_{(k-1),j} L_j} \right] \right\} \text{ (density gradient)}, \end{aligned}$$

where the top line of the model equation includes effects due to longitudinal speed gradients, the second line (the one with p_1) includes an observed effect of density on speed, and the last line includes effects due to longitudinal density gradients. The various variables and parameters of the model are

$\Delta t \overset{\text{def}}{=}$ discrete time interval,

$L_j \overset{\text{def}}{=}$ jth freeway segment length (miles),

$\rho_{(k-1),j} \overset{\text{def}}{=}$ traffic density in jth segment (vehicles per mile),

$p_1 \overset{\text{def}}{=}$ reciprocal of driver reaction time constant (1/s), an unknown parameter,

$p_2 \overset{\text{def}}{=}$ equilibrium traffic speed at zero density, an unknown parameter,
(should be somewhere near the speed limit),

$p_3 \overset{\text{def}}{=}$ equilibrium speed sensitivity to density, an unknown parameter,

$p_4 \overset{\text{def}}{=}$ acceleration response to density gradient, an unknown parameter.

Measurement Model: Observations include the average speeds of vehicles within each segment, as simulated by the microscopic model:

$$z_k = x_k + v_k, v_k \in \mathcal{N}(0, R),$$

where v_k is a zero-mean Gaussian white noise sequence with known covariance R.

Performance Metrics: The primary measures of performance for control purposes are the throughputs of the segments, defined by the numbers of vehicles per unit time leaving the various segments between sample times t_{k-1} and t_k. The objective of this initial study was to develop reliable estimation methods for the parameters and to test the accuracy of a model with the resulting parameters in predicting traffic conditions across traffic segments.

Least-Squares Parameter Estimation: The parameters p_2 and p_3 form a linear model of mean speed as a function of traffic density. Their values could be estimated by straightforward least-squares curve filtering to speed and density data from the microscopic model simulations, as shown in Figure 8.5. This is a plot of simulated vehicle speeds at different traffic densities, which shows a definite trend of mean speed as a function of traffic density.

Parameter Estimation with Extended Kalman Filtering: The remaining unknown parameters are p_1 and p_4, which are appended as augmented state variables:

$$x_{(N+1)} \overset{\text{def}}{=} p_1$$

$$x_{(N+2)} \overset{\text{def}}{=} p_4,$$

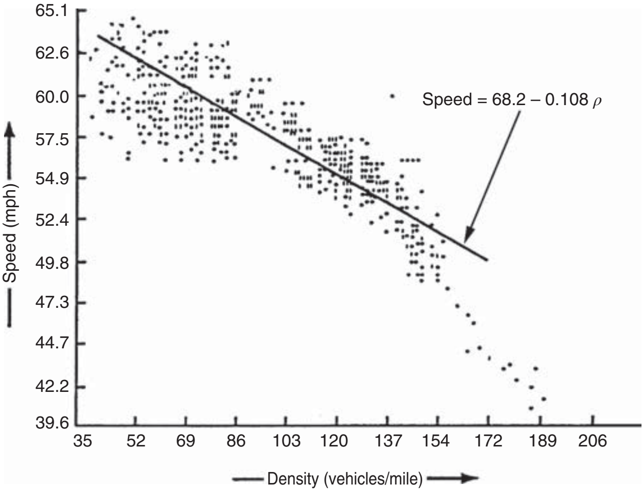

Figure 8.5 Speed–density relationship.

so that the augmented state vector $x^\star$ becomes

$$x^\star = \begin{bmatrix} x \\ x_{N+1} \\ x_{N+2} \end{bmatrix}, \quad x \overset{\text{def}}{=} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix},$$

EKF Dynamic Model: The augmented dynamic system model is then linearized for EKF implementation as

$$x_k^\star = \Phi_{k-1} x_{k-1}^\star + \begin{bmatrix} w_{k-1} \\ 0 \\ 0 \end{bmatrix}$$

$$\Phi_{k-1} \overset{\text{def}}{=} \left. \frac{\partial \phi}{\partial x^\star} \right|_{x^\star = \hat{x}_{k-1}^\star}$$

$$= \left[\begin{array}{c|c} \left. \frac{\partial \phi}{\partial x} \right|_{x = \hat{x}_{k-1}} & \left. \frac{\partial \phi}{\partial x_{N+1}, x_{N+2}} \right|_{x^\star = \hat{x}_{k-1}^\star} \\ \hline 0 & I_2 \end{array} \right].$$

The augmented observation equation then becomes

$$z_k = H_k^\star x_k^\star + v_k$$

$$H_k^\star = \left[I_{N\times N} \mid 0_{N\times 2} \right].$$

Implementation: A four-lane, 6000-ft-long section of freeway with no on- or off-ramps and no accidents was simulated on a mainframe computer using a microscopic model. Eight files of data sets (high flow cases), each containing 20 min of data, along with segment mean speeds and densities at 1.5-s intervals were collected.

To demonstrate the application and performance of the methodology of identifying parameters, results from a digital simulation are shown. It has been observed that speed bears a fairly consistent relationship to density. The "equilibrium" speed–density relationship is obtained from a least-squares straight-line fit to the above data, as shown in Figure 8.5. The parameters p_2 and p_3 were estimated by this procedure.

As a rule, the choice of time-step size tends to be driven by the bandwidth of the dynamic model, and the choice of segment length is chosen to isolate dynamic interactions to adjacent segments only. For this application, a time-step size $\Delta t =$ 4.5 s and the segment length $L_j = 0.5$ miles were chosen for stability considerations.

Parameter Estimation Results: An EKF was used to estimate p_1 and p_4 from the above data. For purposes of numerical computation, it is convenient to define dimensionless variables through the use of nominal values. The initial values used in the parameter identification algorithm are as follows:[4]

$$\text{Nominal segment mean speed} = 40 \text{ mph},$$

$$\text{Initial value of estimated driver reaction time } 1/p_1 = 30 \text{ s},$$

$$\text{Initial value of estimated density gradient sensitivity factor } p_1 p_4 = 4.0 \text{ mi}^2/\text{h}.$$

Figures 8.6 and 8.7 show convergence of the estimates of p_1 and p_4 during several minutes of estimation.

Performance Results: To test the resultant model as a predictor of future traffic conditions, the estimated values of p_1, p_2, p_3, and p_4 were then used in the model equation while segment traffic density $\rho_{k,j}$ and throughputs were computed from the resulting traffic flows. The model was used to predict density and speed of the middle segment by using the available data from the adjoining segments ($x_{k(j-1)}$, ρ_{kj}, $\rho_{k(j+1)}$, and throughputs). This model was found to be particularly effective in predicting speed of traffic flow and density in one segment of the freeway over 15-min intervals. The time interval of 4.5 s was adequate for traffic responsive control. The single-segment density prediction results from the model and actual density are shown in Figure 8.8. The single-segment speed prediction results from the model and actual segment speed are shown in Figure 8.9. The results show that the final model with the parameter values estimated by the above procedures predicted traffic conditions (density and speed) satisfactorily.

[4] The nominal speed is about where USDOT (United States Department of Transportation) formulas predict maximum throughput, which is in the order of 2000 vehicles per lane per hour. However, early studies by US traffic engineers in China found values closer to 30,000 vehicles per lane per hour in some cities. The difference was that those "vehicles" were bicycles and their speeds were around 12 mph.

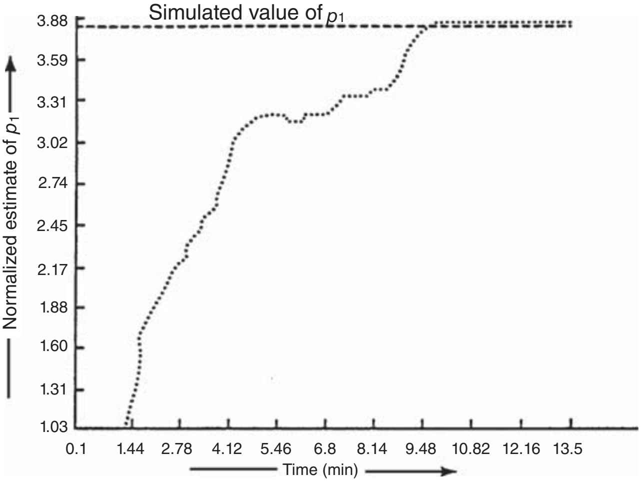

Figure 8.6 Estimation of reaction time.

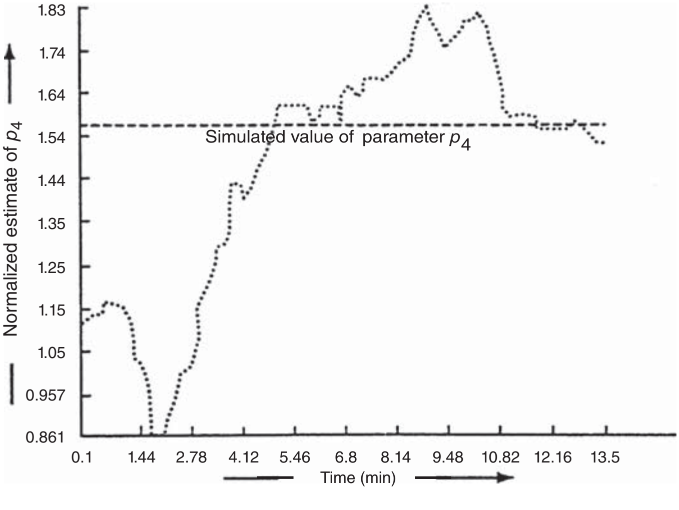

Figure 8.7 Estimation of sensitivity factor.

8.3.3.3 The Iterated Extended Kalman Filter (IEKF) Iteration is a long-established approach to solving many nonlinear problems, such as using Newton's method for solving nonlinear zero-crossing or minimization problems. There have been several iterated implementation techniques used in parts of nonlinear Kalman filtering, including temporal updates and observational updates in the iterated extended Kalman filter (IEKF). We will focus here on the observational update of the covaraince matrix P, because that is where IEKF has been compared favorably with other filtering methods.

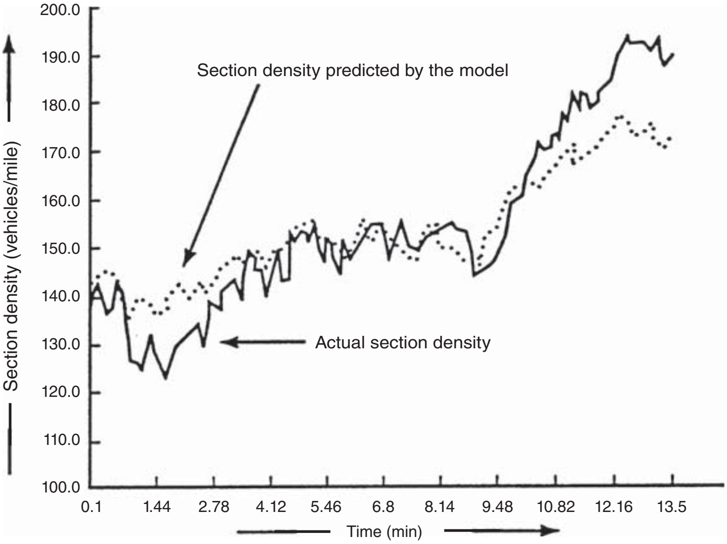

Figure 8.8 Single-segment density prediction.

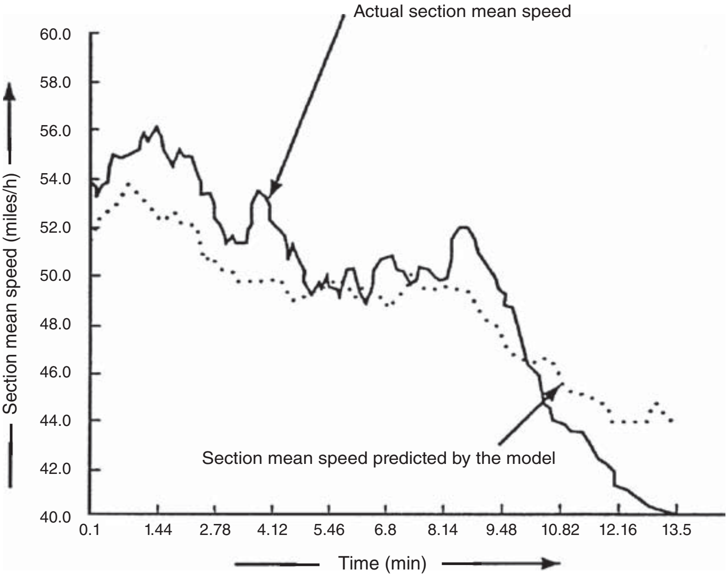

Figure 8.9 Single-segment speed prediction.

Iterated Measurement Update A performance comparison of nonlinear filtering methods by Lefebvre et al. [5] showed the IEKF to outperform the EKF, central difference filter (CDF), divided difference filter (DDF), and UKF for implementing the measurement update of the covariance matrix. Other studies have ranked sampling-based methods (e.g., UKF) higher overall but did not evaluate the observational update independently.

In the EKF, the measurement function $h(t)$ is linearized by partial differentiation,

$$H_k \approx \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_k(-)} \tag{8.31}$$

evaluated at the a priori estimate. The IEKF uses successive evaluations of the partial derivatives at better estimates of $\hat{x}$, starting with the a posteriori estimate from the EKF,

$$\hat{x}_k^{[0]} \overset{\text{def}}{=} \hat{x}_{k(+)}, \tag{8.32}$$

which is the zeroth iteration. The final a posteriori covariance of estimation uncertainty is only calculated after the last iteration, using the final iterated value of H.

Iteration proceeds for i=1, 2, 3, …

$$H_k^{[i]} = \left.\frac{\partial h}{\partial x}\right|_{x=x_k^{[i-1]}} \tag{8.33}$$

$$\overline{K}_k^{[i]} = P_{k(-)}H_k^{\mathrm{T}\,[i]}[H_k^{[i]}P_{k(-)}H_k^{[i]\mathrm{T}} + R_k]^{-1} \tag{8.34}$$

$$\tilde{z}_k^{[i]} = z_k - h_k(\hat{x}_k^{[i-1]}) \tag{8.35}$$

$$\hat{x}_k^{[i]} = \hat{x}_{k(-)} + \overline{K}_k^{[i]}\{\tilde{z}_k^{[i]} - H_k^{[i]}[\hat{x}_{k(-)} - \hat{x}_k^{[i-1]}]\}, \tag{8.36}$$

until some stopping condition is reached. That stopping condition is often that some vector norm of the difference between successive iterated estimates falls below some predetermined threshold ϵ_{limit}. For example, any of the conditions

$$|\hat{x}_k^{[i]} - \hat{x}_k^{[i-1]}|_2 < \epsilon_{\text{limit}} \tag{8.37}$$

$$|\hat{x}_k^{[i]} - \hat{x}_k^{[i-1]}|_\infty < \epsilon_{\text{limit}} \tag{8.38}$$

$$(\hat{x}_k^{[i]} - \hat{x}_k^{[i-1]})^{\mathrm{T}}P_k^{-1}(-)(\hat{x}_k^{[i]} - \hat{x}_k^{[i-1]}) < \epsilon_{\text{limit}} \tag{8.39}$$

might be used as a condition to cease iterating, where the threshold ϵ_{limit} has been chosen based on the analysis of real application data.

The final estimate is $\hat{x}_k^{[i]}$, and the associated a posteriori covariance matrix of estimation uncertainty is the standard formula:

$$P_{k(+)} = P_{k(-)} - \overline{K}_k^{[i]}H_k^{[i]}P_{k(-)}, \tag{8.40}$$

except that the most recently interated value of $H_k^{[i]}$ is used. In this way, the iteration to obtain a refined estimate of H does not corrupt the statistical recordkeeping of the Riccati equation.

Example 8.6 (Nonlinear Bearing-only Measurement Model) As an example of a nonlinear measurement function, consider the sensor geometry illustrated in Figure 8.10, where the angular measurement

$$z_k = \theta_k + v_k \tag{8.41}$$

$$= \arctan\left(\frac{x_k}{d}\right) + v_k \tag{8.42}$$

$$= h(x_k) + v_k \tag{8.43}$$

is the angle θ (plus zero-mean white noise $\{v_k\}$), a nonlinear function of the state variable x_k.

If the partial derivative is used as an approximation for the measurement sensitivity matrix,

$$H \approx \left.\frac{\partial h}{\partial x}\right|_x \tag{8.44}$$

$$= \frac{d}{d^2 + x^2}, \tag{8.45}$$

a function of the value of x at which it is evaluated.

Figure 8.11 is a plot of 100 Monte Carlo implementations of the IEKF with five iterations on this problem with the offset distance $d = 1$. The problem parameters used are

$d = 1$

$R = 10^{-6}$ (1 mrad RMS sensor noise)

$P = 1$ (mean-squared estimation uncertainty)

$x = 1/2$ (true value of state variable)

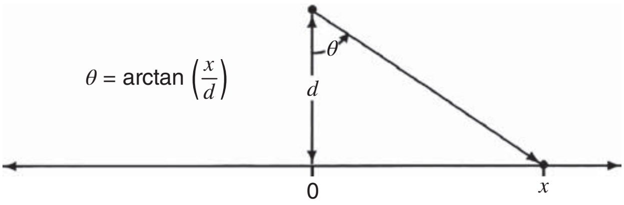

Figure 8.10 Nonlinear measurement model.

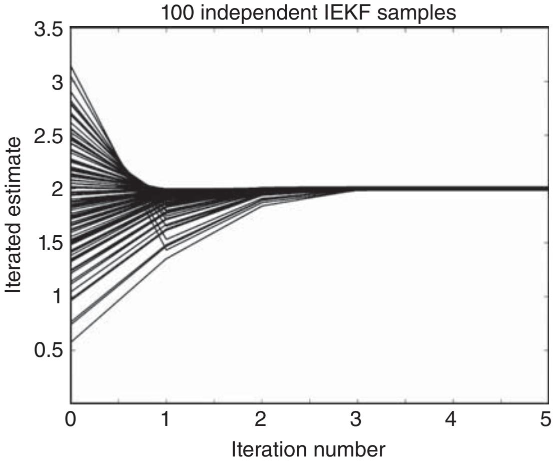

Figure 8.11 100 Monte Carlo simulations of IEKF convergence.

$$z = \arctan(x) + v, v \in \mathcal{N}(0, \sqrt{R}) \text{ (sample measurement)}$$

$$\hat{x} = x + w, w \in \mathcal{N}(0, \sqrt{P})\text{(sample estimate)}$$

The plot in Figure 8.11 shows how the 100 randomly sampled initial estimates converge in the IEKF implementation of Equations 8.33 through 8.36 in five iterations. This result was generated by the MATLAB m-file `ExamIEKF.m`. Each time it is run, it will generate 100 independent Monte Carlo simulations with random estimation errors and random measurement noise. The results shown in Figure 8.11 all converge, which is not that usual. You can demonstrate this by running `ExamIEKF.m` many times, in which case some outcomes will demonstrate instances of nonconvergence. This is due to a number of factors, including the unusually large initial covariance of estimation uncertainty and the relatively high degree of nonlinearity of the arctangent function over that range of initial value samples. You can change the value of *P* in `ExamIEKF.m` to see whether using smaller values of *P* improves the robustness of the implementation against this sort of divergence.

8.3.4 Bounding RMS Linearization Errors

The relative magnitudes of filtering errors due to linear approximation can be analyzed in a number of ways, including

1. Monte Carlo analysis, by which the means and covariances obtained by the EKF can be compared to the results of Monte Carlo simulation. This approach was used as part of the vetting process for Kalman filtering before it was accepted for use on the Apollo navigation problem. It is necessary to assume that all the probability distributions involved are completely known, however.

2. Propagation of multiples of the "one-sigma" points of the equiprobability ellipse through the nonlinearity and comparing the results to the linearized approximation.

In the second of these, linearization errors in the computation of means and covariances can be assessed by comparing the $N\sigma$ points of a the linearized approximation with values obtained by nonlinear simulation for $N \geq 3$. The "sigma points" in this case are the products of the eigenvectors of the covariance matrix of state uncertainty, multiplied by the square roots of the associated eigenvalues. These can be computed using singular value decomposition (SVD, `svd` in MATLAB) of the initial covariance matrix to obtain its eigenvalue–eigenvector decomposition as

$$P = \sum_i \sigma_i^2 e_i e_i^{\mathrm{T}}, \tag{8.46}$$

where the σ_i^2 are the singular values of P, the e_i are the associated eigenvectors, and the vectors

$$\pm N\ \sigma_i e_i$$

are multiples of the "sigma points" of the distribution.

The essential idea is that, within reasonably expected variations of the state vector from its estimated value (as determined by the covariance of state estimation uncertainty), the mean-squared errors due to linearization should be dominated by the model uncertainties. For measurement nonlinearities, the model uncertainties are characterized by the measurement noise covariance R. For dynamic nonlinearities, the model uncertainties are characterized by the dynamic disturbance noise covariance Q.

The range of perturbations under which these conditions need to be met is generally determined by the magnitude of uncertainty in the estimate, which can be estimated using the covariances from linearized Kalman filtering (Section 8.3.3.1). The ranges can be specified in units of the standard deviations of uncertainty.

The resulting statistical conditions for linearization can be stated in the following manner.

1. For the temporal state-transition function $\phi(x)$: for $N \approx 3$ or a bit more, for $N\sigma$ perturbations of x from $\hat{x}$, the linear approximation error is insignificant compared to Q. That is, for $N \geq 3$, for all perturbations δx of $\hat{x}$ such that

$$(\delta x)^{\mathrm{T}} P^{-1} (\delta x) \leq nN^2, \tag{8.47}$$

$$\varepsilon = \underbrace{\phi(\hat{x} + \delta x) - \left[\phi(\hat{x}) + \left.\frac{\partial \phi}{\partial x}\right|_{\hat{x}} \delta x\right]}_{\text{Approximation error}}, \tag{8.48}$$

$$\varepsilon^{\mathrm{T}} Q^{-1} \varepsilon \ll n, \tag{8.49}$$

where n is the dimension of the state vector. That is, for the expected range of variation of estimation errors, the nonlinear approximation errors are dominated by modeled dynamic uncertainty.

2. For the measurement/sensor transformation $h(x)$: for $N\sigma \geq 3\sigma$ perturbations of $\hat{x}$, the linear approximation error is insignificant compared to R. That is, for some $N \geq 3$, for all perturbations δx of $\hat{x}$ such that

$$(\delta x)^{\mathrm{T}} P^{-1} (\delta x) \leq nN^2, \tag{8.50}$$

$$\varepsilon_h = \underbrace{h(\hat{x} + \delta x) - \left[h(\hat{x}) + \left. \frac{\partial h}{\partial x} \right|_{\hat{x}} \delta x \right]}_{\text{Approximation error}}, \tag{8.51}$$

$$\varepsilon_h^{\mathrm{T}} R^{-1} \varepsilon_h \ll \ell, \tag{8.52}$$

where ℓ s the dimension of the measurement vector. The value of estimation uncertainty covariance P used in Equation 8.50 would ordinarily be the a priori value, calculated before the measurement is used. If the measurement update uses what is called the *iterated extended Kalman filter* (IEKF), however, the a posteriori value can be used.

Verifying these conditions requires simulating a nominal trajectory for $x(t)$, implementing the Riccati equation to compute the covariance P, sampling the estimate $\hat{x}$ to satisfy the test conditions, and evaluating the test conditions to verify that the problem is adequately quasilinear. The parameter N is essentially a measure of confidence that the linear approximation will be insignificant in the actual application.

8.3.4.1 Sampling for Testing As a minimum, the perturbations δx can be calculated along the principal axes on the $N\sigma$ equiprobability hyperellipse of the Gaussian distribution of estimation uncertainty. Figure 8.12 is an illustration of such an equiprobability ellipsoid in three dimensions, with arrows drawn along the principal axes of the ellipsoid. These axes and their associated 1σ values can be calculated in MATLAB by using the SVD[5] of a covariance matrix P:

$$P = U \Lambda U^{\mathrm{T}} \tag{8.53}$$

$$= \sum_{i=1}^{n} u_i \sigma_i^2 u_i^{\mathrm{T}} \tag{8.54}$$

$$\delta x_i = N \sigma_i u_i, \; 1 \leq i \leq n \tag{8.55}$$

$$\delta x_{2n+i} = -N \sigma_i u_i, \; 1 \leq i \leq n, \tag{8.56}$$

[5] Implemented by the MATLAB function `svd`.

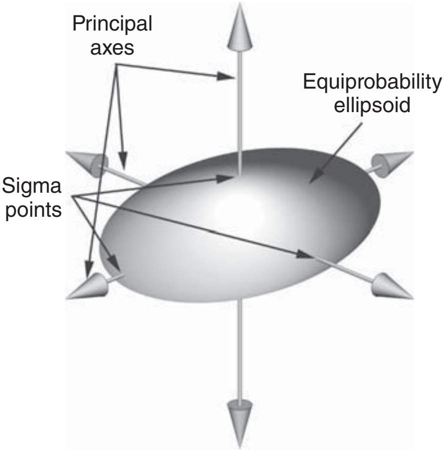

Figure 8.12 Principal axes of equiprobability ellipsoid.

where the vectors u_i are the columns of the orthogonal matrix U in the SVD of P. The principal standard deviations σ_i are the square roots of the diagonal elements of the matrix Λ in the SVD.

The procedure outlined by Equations 8.55 and 8.56 will yield $2n$ perturbation samples, where n is the dimension of x.

Conditions 8.47 and 8.50 depend on estimation uncertainty. The bigger the uncertainties in $\hat{x}$ are, the larger the perturbations must be for satisfying the quasilinearity constraints.

This approach is illustrated in the following example.

Example 8.7 (Satellite Pseudorange Measurements) GNSS use satellites in orbit around the earth as radio navigation beacons. Receivers on or near the surface of the earth use their internal clocks to measure the time delay Δt between when the signal was broadcast from each satellite in view and when it was received. The product of c, the speed of propagation, times the time delay,

$$\rho = c\ \Delta t$$

is called the *pseudorange* to the satellite from the receiver antenna. The essential geometric model for a single satellite is illustrated in Figure 8.13.

The GNSS navigation solution for the location of the receiver antenna requires several such pseudoranges to satellites in different directions. Each satellite broadcasts its precise transmission time and location to the receiver, and the receiver processor is tasked to estimate the location of its antenna from these pseudoranges and satellite positions. This is usually done by extended Kalman filtering, using the derivatives of psuedoranges with respect to receiver location to approximate the measurement sensitivity matrix.

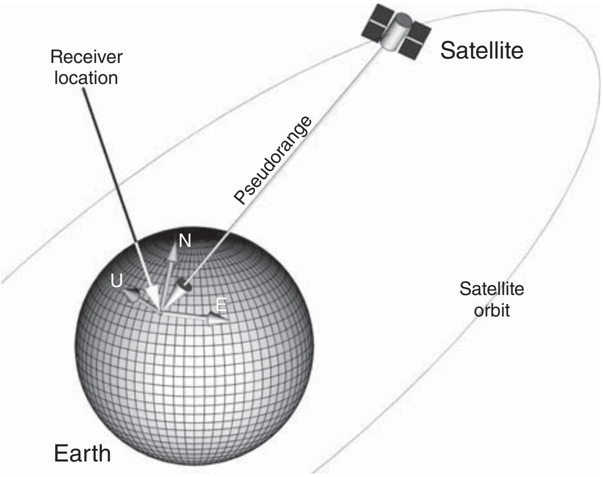

Figure 8.13 GNSS satellite pseudorange ρ.

One can use Equation 8.51 to determine whether the pseudorange measurement is sufficiently linear, given the uncertainty in the receiver antenna position, the nonlinear pseudorange measurement model, and the uncertainty in the pseudorange measurement.

For simplicity, we assume that the root-mean-square (RMS) position uncertainty is uniform in all directions, and

$$P = \begin{bmatrix} \sigma_{\text{pos}}^2 & 0 & 0 \\ 0 & \sigma_{\text{pos}}^2 & 0 \\ 0 & 0 & \sigma_{\text{pos}}^2 \end{bmatrix} \text{(receiver position covariance)},$$

$$R = \sigma_\rho^2 \text{ (pseudorange noise variance)},$$

$$\sigma_\rho = 10 \text{ m (RMS pseudorange noise)},$$

$$R_{\text{sat}} = 2.66 \times 10^7 \text{m, satellite orbit radius},$$

$$R_{\text{rec}} = 6.378 \times 10^6 \text{m, receiver radius from earth center},$$

$$\lambda = 0^\circ,\ 30^\circ,\ 60^\circ,\ \text{and } 90^\circ \text{ elevation of satellite above the horizon},$$

$$\alpha = \text{azimuth direction to the satellite},$$

$$\rho_0 = \sqrt{R_{\text{sat}}^2 - R_{\text{rec}}^2 \cos{(\lambda)}^2} - R_{\text{rec}} \sin(\lambda) \text{ (nominal pseudorange)},$$

$$X_{\text{sat}} = \rho_0 \begin{bmatrix} -\cos{(\lambda)}\cos(\alpha) \\ -\cos(\lambda)\sin(\alpha) \\ \sin(\lambda) \end{bmatrix} \text{(satellite position in east-north-up coordinates)},$$

$$\begin{aligned}
\hat{x} &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{(estimated receiver position in east-north-up coordinates),} \\
X_{\text{rec}} &= \delta x \\
&= \begin{bmatrix} \delta x_E \\ \delta x_N \\ \delta x_U \end{bmatrix} \text{(true receiver position in east-north-up coordinates),} \\
\rho &= |X_{\text{rec}} - X_{\text{sat}}| \text{(true pseudorange),} \\
&= h(\hat{x} + \delta x),
\end{aligned}$$

where the last equation is the nonlinear formula for pseudorange as a function of perturbations δx of receiver antenna position in east-north-up coordinates.

The linear approximation for the sensitivity of pseudorange to antenna position is

$$\begin{aligned}
H &\approx \left. \frac{\partial h}{\partial \delta x} \right|_{\delta x = 0} \\
&= \left[\cos(\lambda)\cos(\alpha) \quad \cos(\lambda)\sin(\alpha) \quad -\sin(\lambda) \right].
\end{aligned}$$

In this case, the nonlinear approximation error defined by Equation 8.51 will be a function of the satellite elevation angle α and the perturbation δx.

The RMS nonlinearity error for the six perturbations

$$\delta x = \begin{bmatrix} \pm 3\sigma_{\text{pos}} \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \pm 3\sigma_{\text{pos}} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ \pm 3\sigma_{\text{pos}} \end{bmatrix}$$

is plotted in Figure 8.14 as a function of σ_{pos} for elevation angles $\alpha = 0°,\ 30°,\ 60°$ and 90° above the horizon. The four barely distinguishable solid diagonal lines in the plot are for these four different satellite elevation angles, which have little influence on nonlinearity errors. The dashed horizontal line represents the RMS pseudorange noise, indicating that nonlinear approximation errors are dominated by pseudorange noise for RMS position uncertainties less than ≈ 7 km.

Typical RMS position errors in GNSS navigation are in the order of 1–100 m, which is well within the quasilinear range. This indicates that extended Kalman filtering with analytical partial derivative approximations is certainly justified for GNSS navigation applications. Furthermore, because it uses simple analytical partial derivative formulas, it will more efficient than the UKF.

8.3.5 Multilocal Linearization for Detection

This is a general method for overcoming initial nonlinearities in applications for which initial estimation uncertainties are significantly greater than those after a few observations have been processed and for which linear approximation errors tend

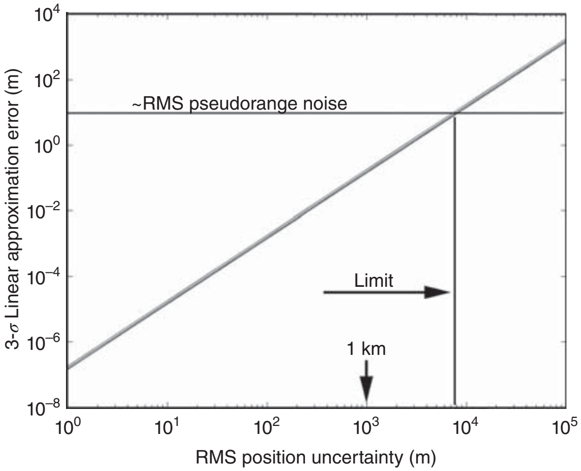

Figure 8.14 Linearization error analysis of pseudorange measurements.

to dominate [6]. It provides relief for that brief period when linearization errors are a significant problem, eliminating the extra computational burden when they are not.

8.3.5.1 Detection versus Tracking The Kalman filter is essentially a *tracking algorithm*. Given an initial estimate of the state of a system and the accompanying covariance of estimation uncertainty, it uses the sequence of measurements to track the state of the system thereafter. Nonlinear approximations to Kalman filtering generally depend on the initial uncertainties being sufficiently small that nonlinear approximation errors are not likely to be statistically significant.

Detection is the problem of obtaining that initial estimate to start tracking. It can lead to implementation problems from two sources:

1. In applications where there is inadequate information to start with, the effective covariance matrix would be infinite. If arbitrarily large values are used, instead, this can easily lead to numerical stability problems in the Kalman filter. This is a fundamental problem in Kalman filtering which is best addressed by using an alternative detection algorithm to initialize the estimate and its covariance or by switching to information-based estimation (which can cope with zero initial information) until sufficient estimation accuracy is attained.
2. If nonlinear approximation methods are being used, then nonlinear approximation errors are more likely to be unacceptable with large initial uncertainties. This problem has been addressed by using a multitude of potential initial guesses $\hat{x}_0$ and something called *Schweppe Gaussian likelihood ratio detection* [7, 8] for selecting the most likely of those initial guesses.

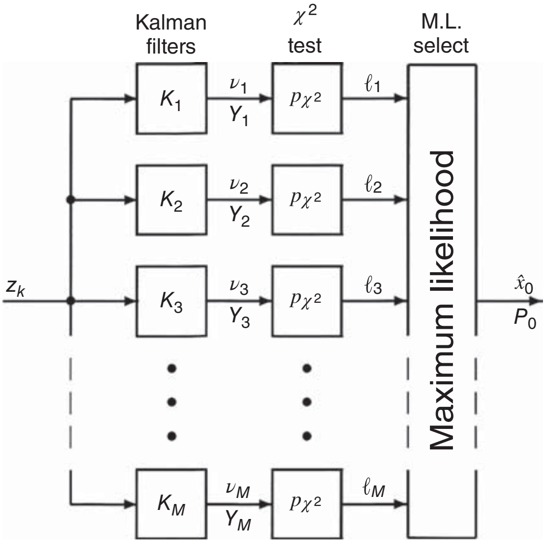

Figure 8.15 Multihypothesis detection/initialization.

Statistical Detection and Selection Schweppe likelihood ratio detection is derived in Section 3.6.5 as a method for deciding which of the two contending hypotheses is more likely, based on the analysis of the filter innovations.

For the nonlinear initialization problem, the approach can be generalized to decide which of a multitude of locally linearized initial guesses of the initial value of a Kalman filter state variable is the most likely, and that multitude of guesses is used for the purpose of reducing initial errors due to nonlinearities. The general approach is shown schematically in Figure 8.15, in this case using a multihypothesis chi-squared test to select the most likely initial guess.

If a Kalman filter is properly modeled, the innovations will be statistically independent, so that the probability of a sequence of innovations is just the product of the individual probabilities, and the logarithm of this product will be the sum of the logarithms of the individual probabilities.

In practice, this is often implemented by summing the logarithms of the *likelihoods* and selecting the largest, or comparing the relative closeness of the mean of the Kalman filter innovations to its expected value ℓ (the dimension of the innovations vector), or the closeness of the normalized mean $\overline{\nu}/\ell$ to 1, or by employing any combination of these measures.

8.3.5.2 Partitioning the Detection Domain The "detection domain" is the region in which the initial value of the state vector may lie. If it can be partitioned into a reasonable number of smaller domains to suppress linearization errors, then an independent Kalman filter can be initialized with a value in each of those subdomains. This depends on the detection domain being finite (or "compact" in topological terms), but there are problem attributes which lend themselves to this.

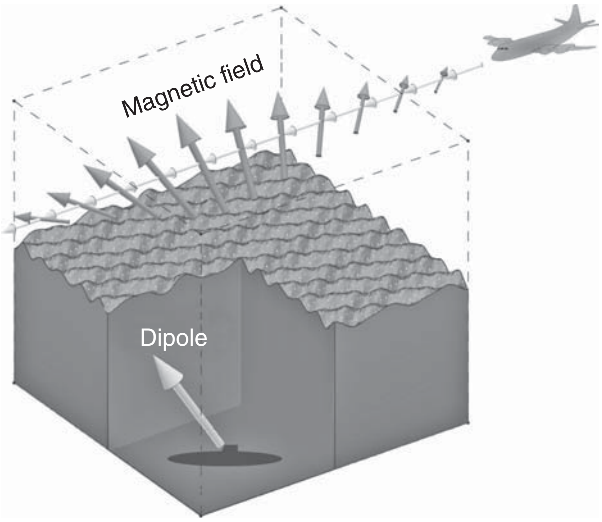

Figure 8.16 Airborne magnetic detection of dipoles.

Sensor Range Limitations In applications using sensors with limited range, the detection region could be within the range domain at which any signal might be discernible above the "noise floor." Radar, lidar, capacitance sensors, and magnetic sensors, for example, generally have output signals that tend to fall off as some inverse power of range.

The magnetic field sensor modeled in Reference 7, used for detecting and locating magnetic dipoles (Figure 8.16), is decidedly nonlinear and also has a limited detection range. In that case, a small number of subdomains can be chosen such that the nonlinearity errors are manageable within each subdomain, and a separate Kalman filter can be used within each subdomain.

The problem then becomes one of monitoring the performance of each Kalman filter and selecting one that estimates a reasonable magnetic moment with acceptable uncertainties.[6]

Subdividing Compact Manifolds In other applications, the full domain may be bounded naturally. Attitude, for example, can be represented by three angular variables and is commonly represented by unit quaternions, which parametrize the surface of the 3-sphere in 4-space in much the same way as unit vectors in 3-space parametrize the 2-sphere. Figure 8.17 represents a covering of the 2-sphere with 1-spheres (circles), in much the same way as the surface of the 3-sphere of the attitude domain might be covered by 2-spheres (which we are unable to depict in a two-dimensional drawing).

Attitude estimation is a notoriously nonlinear estimation problem [9], and being able to partition the search space into smaller domains can keep the influence of

[6] In practice, of course, the implementation does tend to be a bit more complicated.

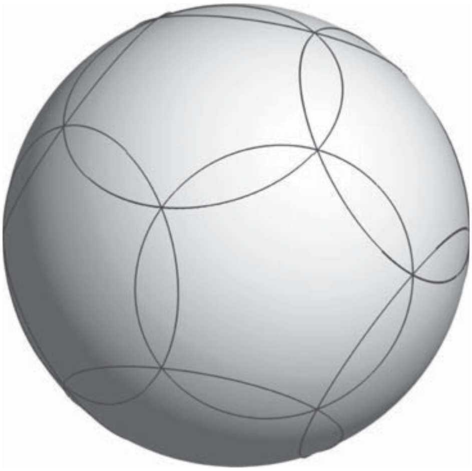

Figure 8.17 Covering the surface of the 2-sphere with 1-spheres.

nonlinear approximation errors down to levels that can be tolerated with an EKF or UKF, for example.

8.4 SAMPLE-AND-PROPAGATE METHODS

These methods for nonlinear propagation of probability densities have been applied to nonlinear extensions of the Kalman filter for propagating means and covariance matrices, including effects of nonlinear dynamics and nonlinear measurements. These are essentially perturbation methods, because they use samples as initial values for trajectories that are perturbations from the trajectory of the mean.

The EKF may also use perturbations for nonlinear propagation of means and expected measurements, but it only uses the perturbed trajectories to compute numerical partial derivatives. The sample-and-propagate methods use perturbed trajectories to approximate nonlinear transformations of the means and covariances.

Sampling essentially transforms a probability density distribution on a continuous domain into one on a discrete domain, by using the probability densities at a discrete set of points to represent the continuous distribution. In so doing, what may not have been computable with the continuous function becomes computable with the sampled set of numbers.

8.4.1 Assessing Performance

The problem of nonlinear propagation of just the means and covariances of a distribution—by any method—is ill-posed, in that there is no unique solution. The problem is that nonlinearities couple higher order moments into the means and covariances, and these higher order moments are not tracked. There is an unlimited number of distributions with the same mean and covariance, all with different higher order moments. Knowing the nonlinear transformation is of no use if one does not know the higher order moments.

Properly designed sample-and-propagate methods for estimating the mean and covariance will always produce the exact solution when the transformation of the variate is linear. Beyond that, there is no exact solution—and just assessing performance can be complicated. It requires specifying all the probability distributions involved, including that of the initial state vector and all the dynamic and measurement error sources involved, and using that information in propagating the full state vector probability through the filtering history. If all the probability distributions involved can be specified and simulated, this might be achieved through Monte Carlo analysis of many filter history instances for a specific nonlinear model. The result would be an assessment of performance on a specific nonlinear model, with specific probability distributions—which can be very useful for "tuning" any parameters of the sample-and-propagate implementation to improve performance on a specified application. Still, one has to be specific about these attributes of the application in specifying the performance of a particular sample-and-propagate method.

8.4.2 Monte Carlo Analysis

This started as a general-purpose method for estimating the evolution over time of a probability distribution. It involves selecting a sufficient number of representative random samples and using simulation to propagate these samples according to the dynamic model for the system. It represents probability density as sample density. The resulting distribution of propagated samples can then provide an estimate of the propagated probability distribution.

It has also been used for assessing probability distributions of estimation error by running many instances of the estimator history with sample input histories.

8.4.2.1 Origins The essential idea behind Monte Carlo analysis is the way experienced gamblers learn their craft: by calculating probabilities based on observed outcomes. The formalization of this as a mathematical process can be traced to the seventeenth century Bernoulli brothers, but its very name and its computer implementation can be traced to experiences of mathematician Stanislaw Ulam (1909–1984) when he was recovering from a bout with encephalitis in 1946 [10]. He was playing a card game called *Canfield solitaire* to while away the hours of recovery and began to wonder about the probability of being dealt a winning hand. It soon became apparent that the problem was not solvable with pencil and paper,[7] but might be approximated by random sampling and simulation. When he returned to work at the Los Alamos National Laboratory, he had both the means for such an approach

[7] The number of distinct shufflings of a 52-card deck is $52! \approx 8 \times 10^{67}$. The odds of winning are not good. The game had been introduced by Richard A. Canfield (1855–1914), an American gambling parlor owner who was able to profit from offering a \$500 prize for winning, after paying \$50 up-front for a deck to try it on.

(the ENIAC computer) and the motivation (solving the neutron diffusion problem for nuclear devices). Ulam worked with John von Neumann (1903–1957) and Nicholas Metropolis[8] (1915–1999), among others, [38] in reducing it to practice.

As proposed by Ulam, the idea is to randomly select representative points from an initial probability distribution and let them do their own thing, then calculate the resulting distribution statistics of interest. Although the *law of large numbers* may guarantee eventual conversion of such methods, convergence is not necessarily fast enough for all practical purposes. Over the years there have been several modifications of the procedures to speed up the process, and there is a vast literature on the design and use of such methods in statistical testing.

8.4.2.2 Random Sampling Random sampling depends on knowing the probability distribution involved and having a method for generating random samples from that distribution. The most common methods are for Gaussian distributions or uniform distributions, but random number generators for uniform scalar distributions can be used to generate random samples for any distribution for which the cumulative probability distribution $P(\cdot)$ and its inverse are known. Given a value $x_{[i]}$ from a uniform distribution between 0 (zero) and 1, the corresponding sample from the distribution represented by $P(\cdot)$ is $y_{[i]} = P^{-1}(x_{[i]})$.

8.4.2.3 Democratic Sampling This uses the inverse cumulative probability function to generate a set of N representative samples, each representing (in some way) one Nth of the population.

Example 8.8 (Arctangent Transformation of Gaussian Variate) The arctangent is an analytic function with power series expansion

$$\arctan(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} x^{2k+1},$$

convergent everywhere on the real line. It is called an *odd function* in that its power series expansion has only odd nonzero coefficients. It maps the infinite domain of the zero-mean univariate Gaussian distribution ($-\infty < x < +\infty$) into a finite range ($-\pi/2 \leq \arctan(x) \leq +\pi/2$).

The objective is to test how different nonlinear approximations perform in updating the means and covariances after the variate is transformed as

$$y(x) = \arctan(a\, x),$$

where a is an arbitrary positive scaling parameter.

[8] Metropolis' contributions to the method may have included naming the approach after the gambling casino at Monaco, where Ulam's uncle allegedly "played" [11].

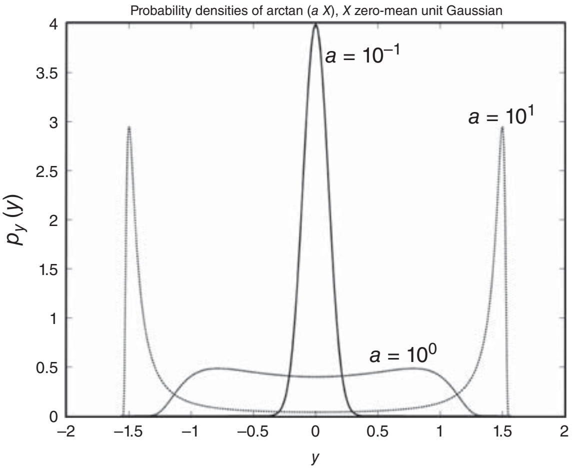

Figure 8.18 Probability densities of $y = \arctan\,(ax), x \in \mathcal{N}(0, 1)$.

If X is zero-mean unit-normal (Gaussian), the transformed variate $Y \overset{\text{def}}{=} \arctan\,(a\,X)$ will have probability density function

$$p_y(y) = \frac{1}{a\sqrt{2\pi}}\{1 + [\tan(y)]^2\}\exp\left(-\frac{1}{2}\left[\frac{\tan(y)}{a}\right]^2\right). \qquad (8.57)$$

This probability density function is nonzero for

$$-\frac{\pi}{2} < y < \frac{\pi}{2}$$

and zero elsewhere. It has the shapes shown in Figure 8.18 for various values of the positive parameter a. These resemble a Gaussian distribution when $a \ll 1$, but are decidedly non-Gaussian for $a \gg 1$.

8.4.2.4 Applications to Kalman Filtering When these methods are used in Kalman filtering for nonlinear propagation of means and covariances of distributions, the statistical parameters required for computing the Kalman gain, the fundamental problem is that one does not know the initial probability distribution—except for its mean and covariance.

This has led to the development of some specialized sample-and-propagate methods for approximation under this constraint.

8.4.3 Particle Filters

Particle filters propagate representative samples of the estimated distribution, which can be viewed as "particles" entrained in the dynamic flow of the distribution [39]. It is generally an improvement over random sampling, because it samples according to what statistics are to be estimated and the character of the nonlinearities involved. In the case of propagating the covariance matrix of a distribution, the selection of samples is biased to make the estimate converge faster than by using random sampling. It has also been called a *sequential Monte Carlo* method [37], because the sampling may change as the solution progresses. "Importance sampling" uses criteria that indicate the relative importance of sampled values for a specific application. Some particle filtering also makes use of particle weights that can be "tuned" to improve performance for a specific application—something that has been incorporated into the UKF, as well.

8.4.3.1 General Nonlinear Model An abstract general-purpose nonlinear model for the estimation problem can be represented in the form

$$x_k = f_{k-1}(x_{k-1}) + w_{k-1} \text{ (nonlinear state dynamics)} \tag{8.58}$$

$$z_k = h_k(x_k) + v_k \text{(nonlinear measurement)}, \tag{8.59}$$

where the functions $f_{k-1}(\cdot)$ and $h_k(\cdot)$ are known. Numerical evaluation of the function $f_{k-1}(x_{k-1})$ may be a simple function call, or it may require integrating a trajectory with x_{k-1} as its initial value.

8.4.3.2 Particle Sampling In practice, the ith sample (or "particle") $\mathcal{S}_{(k-1)\,[i]}$ (sample for x_{k-1}) is a known perturbation $\delta_{[i]}$ of the mean $\hat{x}_{k-1}$:

$$\mathcal{S}_{(k-1)\,[i]} = \hat{x}_{(k-1)(+)} + \delta_{[i]}, \tag{8.60}$$

where the perturbations are chosen to best represent the nonlinear transformation of the mean $\hat{x}$ and covariance P from the $(k-1)$th discrete-time epoch to the kth epoch. The predicted mean $\hat{x}_{k(-)}$ and covariance $P_{k(-)}$ are then estimated from the distribution of the transformed samples

$$\mathcal{S}_{k\,[i]} = f_{k-1}(\mathcal{S}_{(k-1)\,[i]}). \tag{8.61}$$

8.4.3.3 Solution These transformed samples can then be transformed through $h_k(\cdot)$ to obtain the nonlinear approximation to the covariance of sensor output variation due to state uncertainty (HPH^{T}) and the *cross-covariance* between sensor output variation and state vector estimation error (PH^{T}) used in computing the Kalman gain.

8.4.4 Sigma-Point Filters

8.4.4.1 Sampling These methods generally focus on perturbations located at critical points on the surface of the one-sigma equiprobability ellipse of an equivalent Gaussian distribution with the same mean and covariance as the estimate.

These use specific statistical parameters of the a posteriori estimate $\hat{x}_{(k-1)(+)}$ and its covariance matrix $P_{(k-1)(+)}$ to select the samples at time t_{k-1}. In the most straightforward case, sample values are determined from the estimated mean $\hat{x}$ and the eigenvalue–eigenvector decomposition of the covariance matrix P as

$$P = \sum_i \lambda_i e_i e_i^{\mathrm{T}}, \tag{8.62}$$

where the λ_i are positive numbers and the e_i are mutually orthogonal unit vectors such that

$$Pe_i = \lambda_i e_i. \tag{8.63}$$

The “positive” sigma-point sample values are then

$$\mathcal{S}_i = \hat{x} + \sqrt{\lambda_i} e_i. \tag{8.64}$$

There are n of these for an $n \times n$ covariance, but the sampling may use both positive and negative perturbations, numbering $2n$ in all. Counting the “zero perturbation” (i.e., $\hat{x}$), the approach requires either $n + 1$ or $2n + 1$ applications of the dynamic model function $f_{k-1}(\mathcal{S}_i)$.

For Gaussian distributions, the e_i correspond to the principal axes of the equiprobability ellipse. They can be calculated by using the *singular value decomposition* (calculated by the MATLAB function `svd`):

$$P = UD_\lambda U^{\mathrm{T}} \tag{8.65}$$

$$U = [e_1|e_2|e_2|\cdots|e_n] \tag{8.66}$$

$$D_\lambda = \mathrm{diag}(\lambda_1,\ \lambda_2,\ \lambda_3,\ \dots,\ \lambda_n) \tag{8.67}$$

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n \geq 0,$$

where U is an orthogonal matrix. The only thing unique about this method of decomposition is that the nonnegative singular values are generated in descending order.

The offsets of the sigma points from the mean in this case can be calculated as

$$\delta_{[i]} = \sqrt{\lambda_i} e_i, \tag{8.68}$$

in which case the matrix

$$\Gamma = [\delta_{[1]}|\delta_{[2]}|\delta_{[3]}|\cdots|\delta_{[n]}] \tag{8.69}$$

is a modified Cholesky factor of P. That is,

$$\Gamma\Gamma^{\mathrm{T}} = P. \tag{8.70}$$

The perturbed nonlinear trajectories in this case would be those with initial values

$$\mathcal{S}_{(k-1)\,[i]} = \hat{x}_{(k-1)(+)} + \delta_{\,[i]}. \tag{8.71}$$

Figure 8.12 shows an example of such sigma points on an equiprobability surface of a Gaussian distribution in three dimensions. In nonlinear transformations of the state vector, however, error distributions do not necessarily remain Gaussian. Sigma-point filtering may also be defined in terms of sample points that are not necessarily related to the eigenvectors of the covariance matrix. They may, for example, be defined by the columns of any generalized Cholesky factor Γ of P, which would be any solution Γ of $\Gamma\Gamma^{\mathrm{T}} = P$.

8.4.4.2 Propagation Sigma-point filtering has the potential for using different propagation methods for different classes of sigma points, depending on the respective magnitudes of the associated standard deviations. This allows for additional samples to be used for propagating the larger perturbations. However, the propagation error will generally depend on the magnitude of the higher order variations, as well as the magnitude of perturbation.

8.4.4.3 Symmetric Square Roots of Covariance Matrices The diagonal matrix D_λ of Equation 8.67 has nonnegative values and a matrix "square root"

$$\sqrt{D_\lambda} = \operatorname{diag}(\sqrt{\lambda_1},\ \sqrt{\lambda_2},\ \sqrt{\lambda_3},\ \ldots,\ \sqrt{\lambda_n}) \tag{8.72}$$

such that for the symmetric matrix

$$S \overset{\text{def}}{=} U\sqrt{D_\lambda}U^{\mathrm{T}}, \tag{8.73}$$

$$S^2 = S \times S \tag{8.74}$$

$$= P. \tag{8.75}$$

That is, S is a true matrix square root of P. Because it is symmetric, it is also a *generalized Cholesky factor*, and a symmetric one, to boot. However, it turns out that sigma-point-like filters are not overly sensitive to the form or the "square-root" matrix used.

8.5 UNSCENTED KALMAN FILTERS (UKF)

We say "filters" because there is a family of filters using different data structures, depending on the relative need for speed and the degree of nonlinearity, and different sample weightings that can be "tuned" for improved performance on a given application.

We say “unscented” because that is what Jeffrey Uhlmann named them. (See

```
http://www.ieeeghn.org/wiki6/index.php/First-Hand:The_Unscented_
   Transform
```

for an interview with Jeffrey Uhlmann on the subject.)

Uhlmann started development of the UKF in the 1990s by analyzing the performance of different sigma-point sampling and weighting strategies on a nonlinear estimation problem in robotics. His analysis showed no significant variation in performance related to the particular “square-root” factorization of the initial covariance matrix and established some weighting strategies that can be “tuned” to improve performance for different applications. The results of this and collaborative studies was a suite of nonlinear Kalman filter extensions introduced by Simon J. Julier, Jeffrey K. Uhlmann, Hugh F. Durrant-Whyte, and others [12–24], all based on a core methodology for approximating nonlinear transformations of the mean and covariance.

These filter implementations use sample-and-propagate methods for nonlinear propagation of state variables and nonlinear measurements, much as the EKF uses perturbations for numerical approximations of partial derivatives. The UKF makes more intelligent use of the perturbations, so it has about the same computational requirements as the EKF, but improved performance in many applications. These attributes favor the UKF for consideration in many nonlinear filtering applications.

The various implementations of the UKF are a subclass of sigma-point filters and equivalent to linear regression Kalman filters [25] (LRKF) in some cases [26].

8.5.1 The Unscented Transform (UT)

This is the core method for propagating means and covariances, including the cross-covariance between the state and measurement. It replaces the Riccati equation in Kalman filtering and changes the way the Kalman gain is computed.

8.5.1.1 Covariance Factoring The unscented transform (UT) implements mean and covariance propagation using the column vectors of a Cholesky decomposition of the covariance matrix. Like many of the other sample-and-propagate filters, it uses weightings of the columns of the Cholesky factors to allow for some fine-tuning to improve performance on specific applications. These weightings are constrained to be exact when the problem is linear but may otherwise be varied to adapt to problem-specific nonlinearities.

The Cholesky decomposition algorithm is implemented in the built-in MATLAB function `chol`, which yields a lower triangular solution $C_{(k-1)(+)}$ of

$$P_{(k-1)(+)} = C_{(k-1)(+)} C^{\mathrm{T}}_{(k-1)(+)} \tag{8.76}$$

with nonnegative diagonal elements.

8.5.1.2 Sampling Strategies Major sampling and weighting strategies used in the UT are summarized in Table 8.1.

Numbers of Samples The greatest number of sampling perturbations used in the UT are the zero perturbation (i.e., the mean itself) and the signed columns $\pm c_{[i]}$ of $C_{(k-1)(+)}$. That is, the initial conditions of the simulated trajectories will be

$$\mathcal{S}_{(k-1)(+)\,[i]} = \begin{cases} \hat{x}_{(k-1)(+)}, & i = 0 \\ \hat{x}_{(k-1)(+)} + c_{[i]}, & 1 \leq i \leq n \\ \hat{x}_{(k-1)(+)} - c_{[i-n]}, & n+1 \leq i \leq 2n \end{cases} \tag{8.77}$$

and the terminal values will be

$$\mathcal{S}_{k(-)\,[i]} = f_{k-1}(\mathcal{S}_{(k-1)(+)\,[i]}), 0 \leq i \leq 2n. \tag{8.78}$$

However, the subset $\mathcal{S}_{(k-1)(+)\,[i]}$ for $0 \leq i \leq n$ may also be used.

TABLE 8.1 Unscented Transform Sample Weights

Sampling Strategy	Sample Size*	Sample Values†	Sample Weights‡	Index Values
Simplex	$n+2$	$\mathcal{S}_0 = \hat{x}$	$0 \leq \mathcal{W}_0 \leq 1$ (can be varied)	
		$\mathcal{S}_{[i]} = \mathcal{S}_0 + \gamma_{[i]}$	$\mathcal{W}_1 = 2^{-n}(1 - \mathcal{W}_0)$	
		$\Gamma = C_{xx}\Psi$	$\mathcal{W}_2 = \mathcal{W}_1$	
		$\Psi = [\psi_1 \ \psi_2 \ \cdots \ \psi_{n+1}]$	$\mathcal{W}_i = 2^{i-1}\mathcal{W}_1$	$3 \leq i \leq n+1$
		Algorithms for $\psi_{[i]}$ & $\mathcal{W}_i$ are in MATLAB function `UTsimplex`.		
Symmetric	$2n$	$\mathcal{S}_{[i]} = \hat{x} + \sqrt{n}\, c_{[i]}$	$\mathcal{W}_i = 1/(2n)$	$1 \leq i \leq n$
		$\mathcal{S}_{i+n} = \hat{x} - \sqrt{n}\, c_{[i]}$	$\mathcal{W}_{i+n} = 1/(2n)$	$1 \leq i \leq n$
	$2n+1$	$\mathcal{S}_0 = \hat{x}$	$\mathcal{W}_0 = \kappa/\sqrt{n+\kappa}$	
		$\mathcal{S}_{[i]} = \hat{x} + \sqrt{n+\kappa}\, c_{[i]}$	$\mathcal{W}_i = 1/[2(n+\kappa)]$	$1 \leq i \leq n$
		$\mathcal{S}_{i+n} = \hat{x} - \sqrt{n+\kappa}\, c_{[i]}$	$\mathcal{W}_{i+n} = 1/[2(n+\kappa)]$	$1 \leq i \leq n$
Scaled	$2n+1$	$\mathcal{S}_0 = \hat{x}$	$\mathcal{W}_0^{[\hat{y}]} = \lambda/(n+\lambda)$	
		$\lambda = \alpha^2(n+\kappa) - n$	$\mathcal{W}_0^{[P_{yy}]} = \mathcal{W}_0^{[\hat{y}]} + (1 - \alpha^2 + \beta)$	
		$\mathcal{S}_{[i]} = \hat{x} + \sqrt{n+\lambda}\, c_{[i]}$	$\mathcal{W}_i = 1/[2(n+\kappa)]$	$1 \leq i \leq n$
		$\mathcal{S}_{n+i} = \hat{x} - \sqrt{n+\lambda}\, c_{[i]}$	$\mathcal{W}_{n+i} = 1/[2(n+\kappa)]$	$1 \leq i \leq n$

$*$ n = dimension of state space.

† $c_{[i]}$ = ith column of a Cholesky factor C_{xx} of P_{xx}. $\gamma_{[i]}$ is the ith column of Γ.

‡ α, β, κ, and λ are "tuning" parameters.

Even smaller sample sizes were later obtained by using n-simplices.[9] However, the resulting matrix of simplex vectors is multiplied by a Cholesky factor of P to obtain the sampled values to be propagated for estimating the mean and covariance. This could be viewed as a form of "preweighting" of the Cholesky factor samples—before propagation.

8.5.1.3 Weighting Strategies Major strategies for computing weighted averages of propagated statistics are summarized in Table 8.1—and there are more.

The means, covariances, and cross-covariances of state and measurement uncertainty, given the perturbed-and-simulated values $\mathcal{S}_{k(-)i}$, will be weighted averages

$$\hat{x}_{k(-)} = \sum_i \mathcal{W}_{\hat{x},i} \mathcal{S}_{k(-)i} \tag{8.79}$$

$$= \Phi_{k-1} \hat{x}_{(k-1)(+)} \text{ when } f_{k-1}(x) = \Phi_{k-1} x \tag{8.80}$$

$$P_{k(-)} = \sum_i \mathcal{W}_{Pxx,i} (\mathcal{S}_{k(-)i} - \hat{x}_{k(-)})(\mathcal{S}_{k(-)i} - \hat{x}_{k(-)})^{\mathrm{T}} + Q_{k-1} \tag{8.81}$$

$$= \Phi_{k-1} P_{(k-1)(+)} \Phi_{k-1}^{\mathrm{T}} + Q_{k-1} \text{ when } f_{k-1}(x) = \Phi_{k-1} x \tag{8.82}$$

$$\hat{z}_k = \sum_i \mathcal{W}_{\hat{z},i} h_k(\mathcal{S}_{k(-)i}) \tag{8.83}$$

$$= H_k \hat{x}_{k(-)} \text{ when } f_{k-1}(x) = \Phi_{k-1} x \text{ and } h_k(x) - H_k x \tag{8.84}$$

$$P_{zz,k} = \sum_i \mathcal{W}_{Pzz,i} [h_k(\mathcal{S}_{k(-)i}) - \hat{z}_k][h_k(\mathcal{S}_{k(-)i}) - \hat{z}_k]^{\mathrm{T}} \tag{8.85}$$

$$= H_k P_{k(-)} H_k^{\mathrm{T}} \text{ when } h_k(x) = H_k x \tag{8.86}$$

$$P_{xz,k} = \sum_i \mathcal{W}_{Pxz,i} (\mathcal{S}_{k(-)i} - \hat{x}_{k(-)})[h_k(\mathcal{S}_{k(-)i}) - \hat{z}_k]^{\mathrm{T}} \tag{8.87}$$

$$= P_{k(-)} H_k^{\mathrm{T}} \text{ when } f_{k-1}(x) = \Phi_{k-1} x \text{ and } h_k(x) = H_k x, \tag{8.88}$$

where the weightings $\mathcal{W}_{\cdot,i}$ are constrained to obtain the same result as the Kalman filter when $f_{k-1}(\cdot)$ and $h_k(\cdot)$ are linear. Those constraints do not uniquely determine the weightings, however.

Also, if the variations in the functions f_{k-1} and h_k with respect to x at $\hat{x}$ are antisymmetric at $\hat{x}$, that is, for $i > n$

$$(\mathcal{S}_{k(-)i} - \hat{x}_{k(-)}) = -(\mathcal{S}_{k(-)(i-n)} - \hat{x}_{k(-)}) \tag{8.89}$$

$$[h_k(\mathcal{S}_{k(-)i}) - \hat{z}_k] = -[h_k(\mathcal{S}_{k(-)(i-n)}) - \hat{z}_k], \tag{8.90}$$

[9] An n-simplex sampling is a set of $n + 1$ equidistant points in n-dimensional real space. They can be defined as the vertices of the set of points in $(n + 1)$-dimensional space whose coordinates are nonnegative and add up to 1. For example, the 1-simplex is the two ends of a line segment, the 2-simplex is the three vertices of a triangle, and the 3-simplex is the four corners of a tetrahedron.

then the samples with $i > n$ are redundant. That is, only $n + 1$ samples of the state vector are required.

In fact, even if $h_k(\cdot)$ is not antisymmetric but $f_{k-1}(\cdot)$ is, only $n + 1$ samples of the state vector are required. The other samples for $h_k(\cdot)$ can be generated by using the antisymmetry of $f_{k-1}(\cdot)$ in Equation 8.89.

8.5.1.4 Weighting Conformity with the Linear Case The weights of the UT must yield the linear result when core function is linear.

For example, for means, for any $0 < \alpha \leq 1$, the weighting strategy

$$\mathcal{W}_{\hat{x},i} = \begin{cases} \alpha, & i = 0 \\ \dfrac{1-\alpha}{2n}, & i > 0 \end{cases} \tag{8.91}$$

is equivalent to the Kalman filter values when the problem is linear. That is,

$$\hat{x}_{k(-)} = \sum_{i=1}^{2n} \underbrace{\mathcal{W}_{\hat{x},i} H_k \{\hat{x}_{(k-1)(+)} + c_i\}}_{\mathcal{W}_{\hat{x},i} s_{k(-)i}} \tag{8.92}$$

$$= \left\{ \alpha + 2n \times \frac{(1-\alpha)}{2n} \right\} H_k \hat{x}_{(k-1)(+)} + \frac{(1-\alpha)}{2n} H_k \sum_{i=1}^{n} (c_i - c_i) \tag{8.93}$$

$$= H_k \hat{x}_{(k-1)(+)}, \tag{8.94}$$

the linear Kalman filter formula. The same weighting formula works for $\hat{z}_{k(-)}$. For $\alpha = 1$, the results are the EKF values.

The same sort of test can also be applied to the covariance weights.

8.5.2 UKF Implementations

Example 8.9 (Simplex Unscented Transform with Alternative Choleksy Factors) This is a demonstration of the UT "simplex" sampling strategy of Table 8.1, using the nonlinear measurement problem with initial mean and covariance

$$\hat{x} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{8.95}$$

$$P_{xx} = \begin{bmatrix} 84 & -64 & -32 & 16 \\ -64 & 84 & 16 & -32 \\ -32 & 16 & 84 & -64 \\ 16 & -32 & -64 & 84 \end{bmatrix}, \tag{8.96}$$

respectively.

All the UTs use Choleksy factors of P_{xx} in sample selection, without constraining which form is to be used. The effect of the choice of Cholesky factor is demonstrated here by using three alternative Cholesky factors of P_{xx}:

$$C_{\mathrm{UT}} = \begin{bmatrix} 3.6956 & -7.4939 & -3.3371 & 1.7457 \\ 0 & 8.3556 & -1.4118 & -3.4915 \\ 0 & 0 & 5.9362 & -6.9830 \\ 0 & 0 & 0 & 9.1652 \end{bmatrix} \text{(upper triangular)} \tag{8.97}$$

$$C_{\mathrm{LT}} = \begin{bmatrix} 9.1652 & 0 & 0 & 0 \\ -6.9830 & 5.9362 & 0 & 0 \\ -3.4915 & -1.4118 & 8.3556 & 0 \\ 1.7457 & -3.3371 & -7.4939 & 3.6956 \end{bmatrix} \text{(lower triangular)} \tag{8.98}$$

$$C_{\mathrm{EV}} = \begin{bmatrix} -7 & -5 & 3 & 1 \\ 7 & 5 & 3 & 1 \\ 7 & -5 & -3 & 1 \\ -7 & 5 & -3 & 1 \end{bmatrix} \text{(eigenvalue–eigenvector)}, \tag{8.99}$$

and for the second-order nonlinear measurement function

$$z = h(x) \tag{8.100}$$

$$= \begin{bmatrix} x_2 x_3 \\ x_3 x_1 \\ x_1 x_2 \end{bmatrix}. \tag{8.101}$$

For this value of the nonlinear function h and mean $\hat{x} = 0$, the value of $\hat{z}$ for an initial Gaussian distribution can be derived in closed form, as

$$\hat{z} = \mathrm{E}_x\langle h(x)\rangle \tag{8.102}$$

$$= \frac{1}{(2\pi)^2 \det P_{xx}^{-1}} \int dx_4 \int dx_3 \int dx_2 \int dx_1\ h(x)\ \exp\ (-xP_{xx}^{-1}x^{\mathrm{T}}/2) \tag{8.103}$$

$$= \begin{bmatrix} P_{2,3}\left(1 - P_{2,3}^2/P_{2,2}/P_{3,3}\right)^{-3/2} \\ P_{3,1}(1 - P_{3,1}^2/P_{3,3}/P_{1,1})^{-3/2} \\ P_{1,2}(1 - P_{1,2}^2/P_{1,1}/P_{2,2})^{-3/2} \end{bmatrix}. \tag{8.104}$$

That is, the true mean for an initial Gaussian distribution with zero mean depends on the covariance. This dependence is not represented in extended Kalman filtering.

The results, shown in Table 8.2, were generated by the MATLAB m-file `UTsimplexDemo.m`, using the MATLAB function `UTsimplex`, both of which are on the Wiley web site.

The EKF approximation values are also shown in Table 8.2, along with the exact Gaussian solution of Equation 8.104. So that noise covariance does not mask nonlinear error approximations, we also assume that the sensor noise covariance $R = 0$.

TABLE 8.2 Results from Example 8.9: Simplex unscented transform with Alternative Choleksy Factors

Nonlinear Measurement Problem

$$\hat{x} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; P_{xx} = \begin{bmatrix} 84 & -64 & -32 & 16 \\ -64 & 84 & 16 & -32 \\ -32 & 16 & 84 & -64 \\ 16 & -32 & -64 & 84 \end{bmatrix}, \; z = h(x) = \begin{bmatrix} x_2 x_3 \\ x_3 x_1 \\ x_1 x_2 \end{bmatrix}, \; R = 0$$

Solution by Simplex Unscented Transform with $W_0 = 1/2$

Cholesky Factor	Upper Triangular	Lower Triangular	Eigen vector
Sample P_{zz}	$\begin{bmatrix} 4991. & -4247. & -1231. \\ -4247. & 5313. & 4374. \\ -1231. & 4374. & 14750. \end{bmatrix}$	$\begin{bmatrix} 25775. & -17738. & -29636. \\ -17738. & 13961. & 23304. \\ -29636. & 23304. & 40512. \end{bmatrix}$	$\begin{bmatrix} 4465. & -7093. & -9973. \\ -7093. & 15373. & 6985. \\ -9973. & 6985. & 44381. \end{bmatrix}$
Sample $\hat{z}$	$\begin{bmatrix} -8.38 \\ -19.81 \\ -57.90 \end{bmatrix}$	$\begin{bmatrix} 16 \\ -32 \\ -64 \end{bmatrix}$	$\begin{bmatrix} 15 \\ -33 \\ -55 \end{bmatrix}$

Exact Gaussian Solution

$$\hat{z} = \begin{bmatrix} 16.91 \\ -40.49 \\ -235.55 \end{bmatrix}$$

Solution by Extended Kalman Filter Approximation

$$\hat{z} \approx h(\hat{x}) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \; H \approx \left. \frac{\partial h}{\partial x} \right|_{x=\hat{x}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; P_{zz} \approx HP_{xx}H^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Results generated by the MATLAB m-file `UTsimplexDemo.m`, using the MATLAB function `UTsimplex`.

Example 8.9 demonstrates a number of interesting features of the simplex UT for this particular nonlinear problem:

1. The quality of the approximations of $\hat{z}$ and P_{zz} are not what we are used to in extended Kalman filtering of truly quasilinear problems. The initial application of extended Kalman filtering on the Apollo moon missions was able to achieve trajectory estimation accuracies in the order of tens of kilometers at distances in the order of hundreds of thousands of kilometers. However, the degraded performance in this example has more to do with the level of nonlinearity in the application than with qualities of the estimation algorithm. The performance of the EKF approximation in this application is very much worse than that of the UT.
2. The sensitivity of the simplex UT to the selection of a Cholesky factor of P_{xx} is troublesome. The simplex sampling strategy is primarily designed to be more efficient than the symmetric sampling strategies—with the potential risk of being less robust against the potential diversity of Cholesky factors.
3. The simplex strategy—although it is relatively efficient—is not reliably accurate. We must keep in mind, however, that the propagation of means and covariances through nonlinear functions is not uniquely determinable unless the complete initial probability distribution is known. For this reason, all applications of sampling methods to truly nonlinear filtering problems need to be verified through simulation and testing before being trusted in practice.

Example 8.10 (Symmetric Unscented Transform with Alternative Choleksy Factors) Results from using the symmetric UT sampling strategy with the same nonlinear measurement problem as Example 8.9 are shown in Table 8.3. These were generated by the MATLAB m-file `UTscaledDemo.m`, using the MATLAB function `UTscaled`, both of which are on the Wiley web site, and should be compared to the results in Table 8.2 for the simplex UT.

These results are not that different from those using the simplex UT and exhibit similar sensitivities to the choice of Cholesky factor. They are significantly better than the EKF results, however.

This example shows the importance of vetting any nonlinear filtering method on the intended application before proceeding with it.

8.5.2.1 Data Flow versus EKF A data flow diagram of the UKF with $2n + 1$ samples is shown in Figure 8.19. Note the similarity to the comparable EKF data flow diagram in Figure 8.2. With comparable computational effort, the UKF is able to use a more robust strategy for propagating covariances.

Example 8.11 (UKF versus EKF on Artangent Nonlinearity) This example compares the performance of the UKF with that of the EKF for the nonlinear transformation used in Example 8.8, for which we already have a methodology for comparing estimation errors. The values of the mean $\mathrm{E}_x\langle y(x)\rangle$ and variance

TABLE 8.3 Results from Example 8.10: Symmetric Unscented Transform with Alternative Choleksy Factors

Nonlinear Measurement Problem

$$\hat{x} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; P_{xx} = \begin{bmatrix} 84 & -64 & -32 & 16 \\ -64 & 84 & 16 & -32 \\ -32 & 16 & 84 & -64 \\ 16 & -32 & -64 & 84 \end{bmatrix}, \; z = h(x) = \begin{bmatrix} x_2 x_3 \\ x_3 x_1 \\ x_1 x_2 \end{bmatrix}, \; R = 0$$

Solution by Scaled Unscented Transform with $\alpha = 1,\ \beta = 2,\ \kappa = 2$

Cholesky Factor	Upper Triangular	Lower Triangular	Eigen vector
Sample P_{zz}	$\begin{bmatrix} 141 & 332 & -604 \\ 332 & 785 & -1427 \\ -604 & -1427 & 8476 \end{bmatrix}$	$\begin{bmatrix} 1738 & -1829 & -3657 \\ -1829 & 2048 & 4096 \\ -3657 & 4096 & 8192 \end{bmatrix}$	$\begin{bmatrix} 4465. & -7093. & -9973. \\ -7093. & 15373. & 6985. \\ -9973. & 6985. & 44381. \end{bmatrix}$
Sample $\hat{z}$	$\begin{bmatrix} -8.38 \\ -19.81 \\ -57.90 \end{bmatrix}$	$\begin{bmatrix} 16 \\ -32 \\ -64 \end{bmatrix}$	$\begin{bmatrix} 15 \\ -33 \\ -65 \end{bmatrix}$

Exact Gaussian Solution

$$\hat{z} = \begin{bmatrix} 16.91 \\ -40.49 \\ -235.55 \end{bmatrix}$$

Solution by Extended Kalman Filter Approximation

$$\hat{z} \approx h(\hat{x}) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \; H \approx \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; P_{zz} \approx H P_{xx} H^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Results generated by the MATLAB m-file `UTscaledDemo.m`, using the MATLAB function `UTscaled`.

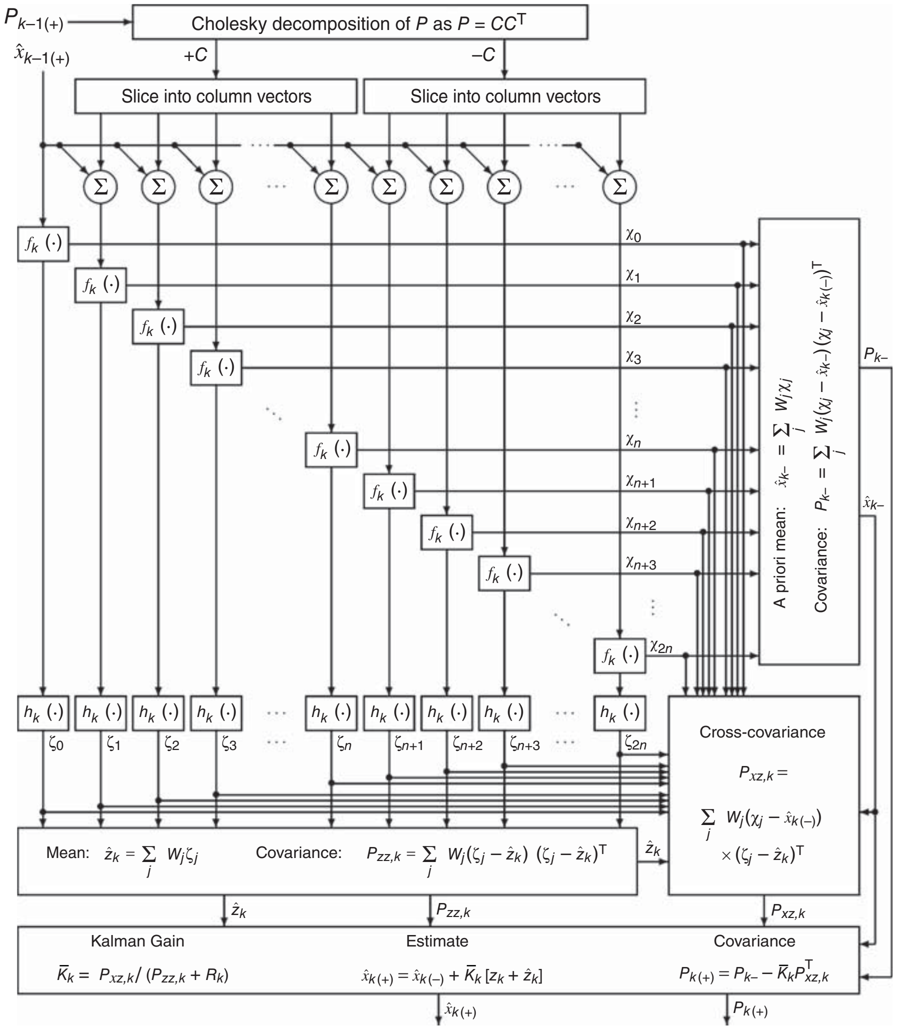
$P_{k-1(+)}$
$\hat{x}_{k-1(+)}$
Cholesky decomposition of P as $P = CC^{\mathrm{T}}$
$+C$
$-C$
Slice into column vectors
Slice into column vectors
Σ
$f_k(\cdot)$
χ_0
χ_1
χ_2
χ_3
χ_n
χ_{n+1}
χ_{n+2}
χ_{n+3}
χ_{2n}
A priori mean: $\hat{x}_{k-} = \sum_j W_j \chi_j$
Covariance: $P_{k-} = \sum_j W_j(\chi_j - \hat{x}_{k-})(\chi_j - \hat{x}_{k(-)})^{\mathrm{T}}$
P_{k-}
$\hat{x}_{k-}$
$h_k(\cdot)$
ζ_0
ζ_1
ζ_2
ζ_3
ζ_n
ζ_{n+1}
ζ_{n+2}
ζ_{n+3}
ζ_{2n}
Cross-covariance
$P_{xz,k} = \sum_j W_j(\chi_j - \hat{x}_{k(-)}) \times (\zeta_j - \hat{z}_k)^{\mathrm{T}}$
Mean: $\hat{z}_k = \sum_j W_j \zeta_j$
Covariance: $P_{zz,k} = \sum_j W_j(\zeta_j - \hat{z}_k)\ (\zeta_j - \hat{z}_k)^{\mathrm{T}}$
$\hat{z}_k$
$\hat{z}_k$
$P_{zz,k}$
$P_{xz,k}$
Kalman Gain
$\overline{K}_k = P_{xz,k}/\,(P_{zz,k} + R_k)$
Estimate
$\hat{x}_{k(+)} = \hat{x}_{k(-)} + \overline{K}_k\,[z_k + \hat{z}_k]$
Covariance
$P_{k(+)} = P_{k-} - \overline{K}_k P^{\mathrm{T}}_{xz,k}$
$\hat{x}_{k(+)}$
$P_{k(+)}$

Figure 8.19 Data flow for symmetric UKF with weighting.

$\mathrm{E}_x\langle[y(x) - E_\chi\langle y(\chi)\rangle]^2\rangle$ from the EKF and UKF are compared to the value obtained by numerical integration of the expected values involved, assuming the initial probability is Gaussian with mean and variance

$$\hat{x}_{k-1\ (+)} = 0$$

$$P_{k-1\ (+)} = 1.$$

EKF Implementation: The nonlinear temporal state transition

$$x_k = \arctan(a\, x_{k-1})$$

would be modeled in an EKF as

$$\begin{aligned}
\hat{x}_{k\,(-)} &= \arctan(a\,\hat{x}_{k-1\,(+)}) \\
&= 0 \\
P_{k\,(-)} &= \left[\left. \frac{\partial \arctan\,(ax)}{\partial x} \right|_{x=\hat{x}_{k-1\,(+)}} \right]^2 P_{k-1\,(+)} \\
&= a^2.
\end{aligned}$$

UKF Implementation: The equivalent update for the UKF implementation would be

$$\begin{aligned}
\hat{x}_{k\,(-)} &= \arctan(a\,\hat{x}_{k-1\,(+)}) \\
&= 0 \\
P_{k\,(-)} &= [\arctan(a\sqrt{P_{k-1\,(+)}})]^2 \\
&= [\arctan(a)]^2.
\end{aligned}$$

Numerical Integration: The mean of the transformed variate will be zero because the arctangent is an odd function. Both the EKF and UKF transform the mean correctly. As a check of how they do on variances, numerical integration of the variance uses a "democratic sampling" of the univariate distribution, in which the real line is divided into N conterminous segments with each segment representing one Nth of the cumulative Gaussian probability. Each segment is represented by its median, which is the mid-point of its cumulative probability. Then the expected value of any function $f(x)$ on the real line can be approximated as[10]

$$\mathop{\mathrm{E}}_{x}\langle f(x)\rangle = \int_{-\infty}^{+\infty} f(x)\,p(x)\,dx \tag{8.105}$$

$$\approx \frac{1}{N}\sum_{n=1}^{N} f(x_n), \tag{8.106}$$

where the x_n are the medians of the segments and the function

$$f(x) = \arctan^2(ax), \tag{8.107}$$

with scaling parameter values $a > 0$ varied to show how scaling influences the variance of the transformed distribution.

[10] The summation formula can be implemented as a dot-product in MATLAB.

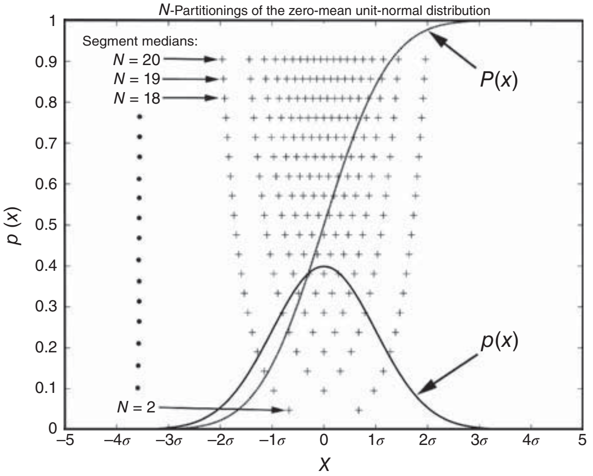

Figure 8.20 Equiprobability partitionings of a Gaussian distribution.

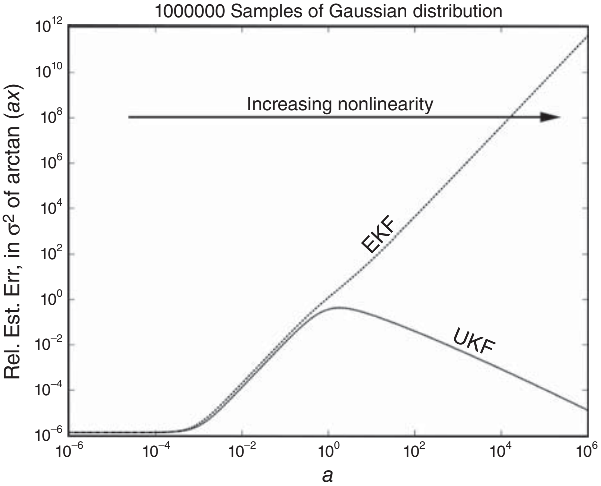

Figure 8.21 Relative error in variance estimates.

The medians of the segments are plotted as "+" symbols in Figure 8.20 for $2 \leq N \leq 20$. A value of $N = 10^6$ was used in the numerical integrations, in which case the sample values of x range from about $-4.9\ \sigma$ to $+4.9\ \sigma$.

Results Results are shown in Figure 8.21, in which values of the numerically integrated variances are compared to the EKF and UKF approximations as a ranges over 12 orders of magnitude. On the plots, the UKF values are denoted by "UKF" and the EKF values by "EKF."

This is a plot of the relative error in the EKF and UKF estimates of variance, in which the relative error in estimating the transformed variance for the EKF approach grows without bound as a increases. The worst relative error of the UKF approximation, on the other hand, gets progressively better when $a \gg 1$.

The result clearly shows—for this particular example—the superiority of the unscented transformation of the variance estimate over that of the EKF. However, the peak relative error in the UKF solution for this problem is still unacceptably high—around 43%.

A tutorial-level introduction to the UKF (with MATLAB implementations) by Yi Cao can be found at

```
http://www.mathworks.com/matlabcentral/fileexchange/18217-learning-
the-unscented-kalman-filter/content/ukf.m.
```

8.5.3 Unscented *sigmaRho* Filtering

The *sigmaRho* filter (described in Section 7.6) is well suited for the UT. It uses a factorization of the Kalman filter covariance matrix P as

$$P = D_\sigma C_\rho D_\sigma \tag{8.108}$$

$$D_\sigma = \mathrm{diag}[\sigma_1,\ \sigma_2,\ \sigma_3,\ \ldots\ ,\ \sigma_n] \tag{8.109}$$

$$= \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_n \end{bmatrix} \tag{8.110}$$

$$C_\rho = \begin{bmatrix} 1 & \rho_{12} & \rho_{13} & \cdots & \rho_{1n} \\ \rho_{21} & 1 & \rho_{23} & \cdots & \rho_{2n} \\ \rho_{31} & \rho_{32} & 1 & \cdots & \rho_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \rho_{n3} & \cdots & 1 \end{bmatrix}, \tag{8.111}$$

where the σ_i are the standard deviations of the error distribution and the ρ_{ij} are the correlation coefficients.

Therefore, if A_ρ is any solution of

$$A_\rho A_\rho^{\mathrm{T}} = C_\rho, \tag{8.112}$$

then $D_\sigma A_\rho$ is a "square root" (generalized Cholesky factor[11]) of P. The columns of $D_\sigma A_\rho$ can then be put through any nonlinear dynamic function f and measurement function h to propagate the covariance matrix and compute the Kalman gain.

[11] Cholesky factors are defined by the Cholesky factoring algorithm, the output of which is a lower triangular matrix L. A "generalized" Cholesky factor has the form $C = LU$, where U is orthogonal.

8.6 TRULY NONLINEAR ESTIMATION

The early history of development of stochastic systems theory has been described in Section 4.2.1. Up to the time the Kalman filter was published, major contributors included Max Planck (1858–1947), Paul Langevin (1872–1946), Marian Smoluchowski (1872–1917), Albert Einstein (1879–1955), Adriaan Fokker (1887–1972), Andrey Kolmogorov (1903–1987), and Ruslan Stratonovich (1930–1997). The Kalman filter was the first exact and practical solution to estimation problem, but it assumed linear dynamic and measurement models.

Few practical nonlinear solutions have emerged since that time, but application areas have included military surveillance, economic system modeling, and market modeling—disciplines that have not been overly biased toward open publication. The following are two results that have appeared in the open literature.

8.6.1 The Beneš Filter

This is perhaps the most successful of the higher order derivations, in that it has an exact solution. It is one of the many discoveries of Czech-American mathematician Václav Beneš[12] [3], this one a finite-dimensional representation of a class of filters using linear measurements with additive noise and nonlinear stochastic dynamics satisfying certain regularity conditions.

It is equivalent to the Kalman filter when the dynamic model is linear.

The Beneš filter model has the form

$$\frac{d}{dt}x(t) = f(x) + Gw(t) \text{ (nonlinear time-invariant)} \tag{8.113}$$

$$z_k = H_k x_k + v_k \text{ (linear discrete)} \tag{8.114}$$

$$\underset{w}{\mathrm{E}}\langle w(t)w^{\mathrm{T}}(t)\rangle = Q \text{ (time-invariant and nonsingular)} \tag{8.115}$$

$$v_k \in \mathcal{N}(0, R), \tag{8.116}$$

plus additional constraints on the unconditioned probability density function of the state vector.

Derivations of the Beneš filter implementation formulas can be found in References 3 and 27, and comparative studies can be found in References 27-29. These tend to show that computational requirements for real-time implementation of the Beneš filter can easily be overwhelming.

We mention the Beneš filter here because it provides an example of a finite-dimensional nonlinear stochastic estimator, although it did not compare well with the EKF in some studies [29]. It has probably been used more in financial market modeling than as a nonlinear extension of Kalman filtering. Its asymptotic performance has been analyzed by Ocone [30]. Farina et al. [29] have compared its

[12] The Czech "š" in proper names is not uncommonly transliterated as "s," although it sounds more like the English "sh."

performance to that of the EKF, linearized filters, and particle filters on a specific problem with known solution. Interested readers are referred to the expositions by Beneš [3,35], Daum [31], Bain and Crisan [27], and the references therein.

8.6.2 Surveillance Solution of Richardson and Marsh

This could be viewed as a nonlinear alternative to multilocal linearization for solving the detection problem. It is what would be called a *grid-based* method for numerical solution of the nonlinear detection and estimation problem. It solves an approximation of the conditioned Fokker–Planck equation on a regular grid in state space.

The following discussion is about an example of an extension of the open-source theory applied to integrated detection and tracking, starting with what is called *the conditioned Fokker–Planck equation* [32], a stochastic differential equation modeling the evolution over time of a state probability density function, given measurements related to the system state.

This particular application is for *surveillance problems*, which include the detection, identification, and tracking of objects within a certain region of state space, where the initial distribution of potential objects to be detected and tracked is according to some random "point process model."

A *point process* is a type of random process for modeling events or objects that are distributed over time and/or space, such as the arrivals[13] of messages at a communications switching center, or the locations of stars in the sky. It is also a model for the initial states of systems in many estimation problems, such as the locations in time and space of aircraft or spacecraft under surveillance by a radar installation, or the locations of submarines under sonar surveillance in the ocean. Results for these applications tend to be classified, so that there is scant information about capabilities in the open literature. The one statistical surveillance method mentioned here is in the open literature [33], but it is not known how this might compare with classified methods.

A unified approach of this sort, combining detection and tracking into one optimal estimation method, was derived by the late John M. Richardson (1918–1996) and specialized to several applications [33]. The detection and tracking problem for a *single object* is represented by the conditioned Fokker–Planck equation. Richardson derived from this one-object model an infinite hierarchy of partial differential equations representing *object densities* and truncated this hierarchy with a simple closure assumption about the relationships between moments. Using object densities in place of single-object probabilities makes the approach applicable to the multiobject surveillance problem. The result is a single partial differential equation approximating the evolution over time of the density of objects, based on the observations. The resulting simplified model was programmed into a numerical solution by Richardson and Marsh [33] and applied to a few simple test problems. Results show some promise as a unified solution to the difficult problem of detecting dynamic objects whose initial states are represented by a point process.

[13] In these applications, a point process is also called an *arrival process*.

This would be classified as a "grid-based" approach, in that its output is a set of probability densities evaluated on a grid covering the searchable surveillance space.

8.7 SUMMARY

8.7.1 Main Points

1. As a rule, nonlinear extensions of the Kalman filter are designed to yield the same result as the Kalman filter when the model functions are linear and to degrade gradually when the model becomes increasing less linear.
2. Affine (linear plus offset) filter models are equivalent to models with nonzero-mean noise sources, both can be accommodated with very minor changes in the Kalman filter equations and both result in exact solutions.
3. The EKF has enjoyed a long career with "ever-so-slightly" nonlinear applications and with considerable success. For many problems with acceptable linear approximation using analytic partial derivatives, it is still the most efficient implementation.
4. The UKF has about the same computation requirements as the EKF using numerical partial derivatives and provides an additional margin of robustness against nonlinear effects.
5. The Beneš filter is the only nonlinear extension of the Kalman filter (besides the affine Kalman filter) that is not an approximation, but its performance compared to the EKF or UKF on some applications has not been promising.
6. Any assessment of the relative efficacies of the various nonlinear approximations will be highly dependent on the particulars of the application(s) assumed. Such assessments are best used for picking which approximation to choose for your particular application.

8.7.2 Limitations of Nonlinear Approximation

Truly nonlinear estimation, in general, has reasonably good models from the standpoint of representing the evolution over time of probability distributions. However, their computational complexity tends to rule them out for the sort of real-time applications for which Kalman filtering has been so useful [37].

Except for the affine and Beneš filters, all existing nonlinear extensions of the Kalman filter are approximations of one sort or another.

The mean and covariance matrix are sufficient statistics for minimum mean-squared error estimation only if the measurements and dynamics are linear (or affine) functions of the state vector.

Otherwise, the mean and covariance matrix will be corrupted by the higher order moments of the distribution which are not propagated in the linearized, extended, or sample-and-propagate filters. Approximation methods attempt to minimize this corruption, but they cannot guarantee to reduce it to acceptable levels for all nonlinear models. The remarkable thing is that sample-and-propagate methods have performed as well as they have, given the inherent insufficiency of information for propagating means and covariances.

The effects of linear approximation errors can be approximated for the EKF. Comparable assessments for sample-and-propagate approximations have been done assuming Gaussian distributions. However, nonlinear transformations do not preserve Gaussianity.

PROBLEMS

8.1 A scalar stochastic sequence x_k is given by

$$\begin{gathered} x_k = -0.1x_{k-1} + \cos x_{k-1} + w_{k-1}, \quad z_k = x_k^2 + v_k, \\ \mathrm{E}\langle w_k \rangle = 0 = \mathrm{E}\langle v_k \rangle, \quad \operatorname{cov} w_k = \Delta(k_2 - k_1), \quad \operatorname{cov} v_k = 0.5\Delta(k_2 - k_1), \\ \mathrm{E}\langle x_0 \rangle = 0, \quad P_0 = 1, \quad x_k^{\mathrm{nom}} = 1. \end{gathered}$$

Determine the linearized and extended Kalman estimator equations.

8.2 A scalar stochastic process $x(t)$ is given by

$$\begin{aligned} \dot{x}(t) &= -0.5x^2(t) + w(t), \\ z(t) &= x^3(t) + v(t), \\ \mathrm{E}\langle w(t) \rangle &= \mathrm{E}\langle v(t) \rangle = 0 \\ \operatorname{cov} w_t &= \delta(t_1 - t_2), \quad \operatorname{cov} v(t) = 0.5\delta(t_1 - t_2), \\ \mathrm{E}\langle x_0 \rangle &= 0, \quad P_0 = 1, \quad x^{\mathrm{nom}} = 1. \end{aligned}$$

Determine the linearized and extended Kalman estimator equations.

8.3 **(a)** Verify the results of Example 8.6 (IEKF).

(b) Estimate the states from a noisy data.

(c) Compare the results of linearized Kalman filter and EKF.
Assume that the plant noise is normally distributed with mean 0 and covariance 0.2 and measurement noise is normally distributed with mean 0 and covariance 0.001.

8.4 Derive the linearized and EKF equations for the following equations:

$$x_k = f_{k-1}(x_{k-1}) + Gw_{k-1}, \quad z_k = h_k(x_k) + v_k.$$

8.5 Given the following plant and measurement model for a scalar dynamic system:

$$\begin{aligned} \dot{x}(t) &= ax(t) + w(t), \quad z(t) = x(t) + v(t), \\ w(t) &\sim \mathcal{N}(0,1), \quad v(t) \sim \mathcal{N}(0,2) \\ \mathrm{E}\langle x(0)\rangle &= 1 \\ \mathrm{E}\langle w(t)v(t)\rangle &= 0 \\ P(0) &= 2, \end{aligned}$$

assume an unknown constant parameter a and derive an estimator for a, given $z(t)$.

8.6 Let **r** represent the position vector to a magnet with dipole moment vector **m**. The magnetic field vector **H** at the origin of the coordinate system in which **r** is measured is given by the formula

$$\mathbf{B} = \frac{\mu_0}{4\pi|\mathbf{r}|^5}\{3\mathbf{r}\mathbf{r}^{\mathrm{T}} - |\mathbf{r}|^2\mathbf{I}\}\mathbf{m} \tag{8.117}$$

in SI units.

(a) Derive the measurement sensitivity matrix for **H** as the measurement and **m** as the state vector.

(b) Derive the sensitivity matrix for **r** as the state vector.

(c) If **r** is known but **m** is to be estimated from measurements of **B**, is the estimation problem linear?

8.7 Generate the error covariance results for the plant and measurement models given in Example 8.4 with the given values of process and measurement noise covariance and initial state estimation error covariance.

8.8 This problem is taken from Reference 34. The equations of motion for the space vehicle are given as

$$\ddot{r} - r\dot{\theta}^2 + \frac{k}{r^2} = w_r(t), \quad r\ddot{\theta} + 2\dot{r}\dot{\theta} = w_\theta(t),$$

where r is range, θ is bearing angle, k is a constant, and $w_r(t)$ and $w_\theta(t)$ are small random forcing functions in the r and θ directions.

The observation equation is given by

$$z(t) = \begin{bmatrix} \sin^{-1}\frac{R_e}{r} \\ \alpha_0 - \theta \end{bmatrix},$$

where R_e is the earth radius and α_0 is a constant.

Linearize these equations about $r_{\mathrm{nom}} = R_0$ and $\theta_{\mathrm{nom}} = w_0 t$.

8.9 Let the 3 × 3 covariance matrix[14]

$$P = \begin{bmatrix} 11 & 7 & -9 \\ 7 & 11 & -9 \\ -9 & -9 & 27 \end{bmatrix}. \tag{8.118}$$

(a) Use the built-in MATLAB function `chol` to compute the lower-triangular Cholesky factor of P. Note that the MATLAB command `C = chol(P)` returns an upper triangular matrix C such that $C^{\mathrm{T}}C = P$, in which case C^{T} is a lower triangular Cholesky factor.

(b) Use the MATLAB function `utchol` on the Wiley web site to compute the upper triangular Choleksy factor of P.

(c) Use the MATLAB function `svd` to compute its symmetric square root, which is also a Cholesky factor. (Note that `[U,Lambda,V] = svd(P)` returns V such that $P = U\Lambda V^{\mathrm{T}}$, where $V = U$.) The square root of P is then $S_P = U\Lambda^{1/2}U^{\mathrm{T}}$, where $S_P = C$ is also a symmetric Cholesky factor of P. Multiply your result by itself to verify your answer.

(d) What are the maximum and minimum eigenvalues of P?

(e) What is the condition number of P?

8.10 For each of the Cholesky factors in the previous problem, compute the corresponding $2n + 1 = 7$ sigma points for the UT with $\kappa = 0$. Assume $\hat{x} = 0$, and do not forget to multiply your Cholesky factors by $\sqrt{n} = \sqrt{3}$ to obtain the properly scaled sigma points.

8.11 Using the three sets of sigma points from the previous problem, write a MATLAB program to compute the resulting means and variances obtained by transforming the sigma points through the nonlinear function

$$\vec{f}\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} x_2 x_3 \\ x_3 x_1 \\ x_1 x_2 \end{bmatrix}.$$

8.12 Compare the resulting means from the previous problem to the exact mean

$$\mathrm{E}\left\langle \begin{bmatrix} x_2 x_3 \\ x_3 x_1 \\ x_1 x_2 \end{bmatrix} \right\rangle = \begin{bmatrix} p_{2,3}\left(1 - p_{2,3}^2/p_{2,2}/p_{3,3}\right)^{-3/2} \\ p_{3,1}(1 - p_{3,1}^2/p_{3,3}/p_{1,1})^{-3/2} \\ p_{1,2}(1 - p_{1,2}^2/p_{1,1}/p_{2,2})^{-3/2} \end{bmatrix}$$

for Gaussian initial distributions with zero mean and covariance P. Use the value of P from Equation 8.118 in this calculation.

[14] The value of P in Equation 8.118 was used to generate the equiprobability ellipsoid shown in Figure 8.12.

8.13 Rework Example 8.4 using the symmetric UKF implementation with $2n+1$ samples. Use `utchol` to compute upper triangular Cholesky factors and the MATLAB function `ode45` (fourth- to fifth order Runge–Kutta integration) to propagate the samples forward in time. Use the EKF approximation to compute Q_k from $\dot{Q} = FQF^{\mathrm{T}} + Q_t$ with $Q(t_{k-1}) = 0$ and $F = \frac{\partial \dot{x}}{\partial x}$.

8.14 Is a symmetric square root of a symmetric positive-definite matrix also a Cholesky factor? Explain your answer.

REFERENCES

[1] R. W. Bass, V. D. Norum, and L. Schwartz, "Optimal multichannel nonlinear filtering," *Journal of Mathematical Analysis and Applications*, Vol. 16, pp. 152–164, 1966.

[2] D. M. Wiberg and L. A. Campbell, "A discrete-time convergent approximation of the optimal recursive parameter estimator", in *Proceedings of the IFAC Identification and System Parameter Identification Symposium*, Vol. 1, International Federation on Automatic Control, Laxenburg, Austria, pp. 140–144, 1991.

[3] V. E. Beneš, "Exact finite-dimensional filters with certain diffusion non-linear drift," *Stochastics*, Vol. 5, pp. 65–92, 1981.

[4] M. S. Grewal and H. J. Payne, "Identification of parameters in a freeway traffic model," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-6, pp. 176–185, 1976.

[5] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Kalman Filters for nonlinear systems: a comparison of performance," *International Journal of Control*, Vol. 77, No. 7, pp. 639–653, 2004.

[6] A. P. Andrews, U.S. Patent No. 5381095, "Method of estimating location and orientation of magnetic dipoles using extended Kalman filtering and Schweppe likelihood ratio detection," 1995.

[7] A. P. Andrews, "The accuracy of navigation using magnetic dipole beacons," *Navigation, The Journal of the Institute of Navigation*, Vol. 38, pp. 369–397, 1991–1992.

[8] F. C. Schweppe, "Evaluation of likelihood functions for Gaussian signals," *IEEE Transactions on Information Theory*, Vol. IT-11, pp. 61–70, 1965.

[9] F. L. Markley. "Attitude error representations for Kalman Filtering," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, pp. 311–317, 2003.

[10] S. M. Ulam, *Adventures of a Mathematician*, University of California Press, Berkeley, CA, 1991.

[11] N. Metropolis, "The beginning of the Monte Carlo method," *Los Alamos Science*, No. 15, Special Issue: Stanislaw Ulam (1909–1984), pp. 125–130, 1987.

[12] S. J. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Technical Report, Robotics Research Group, Department of Engineering Science, University of Oxford, 1994.

[13] S. J. Julier, J. K. Uhlmann, and H. F. Durrenat-Whyte, "A new approach to filtering nonlinear systems," in *Proceedings of the 1995 American Control Conference*, Seattle, WA, pp. 1628–1632, 1995.

[14] S. J. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Technical Report, Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford, 1996.

[15] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management*, SPIE, Orlando, FL, pp. 182–193, 1997.

[16] S. J. Julier, J. K. Uhlmann and H. F. Durrant-Whyte, "A new approach for the nonlinear transformation of means and covariances in linear filters," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, pp. 477–482, 2000.

[17] S. J. Julier and J. K. Uhlmann, "Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations," *Proceedings of the IEEE American Control Conference*, Anchorage, Alaska, Vol. 2, pp. 887–892, 2002.

[18] S. J. Julier and J. K. Uhlmann, "Comment on 'a new method for the nonlinear transformation of means and covariances in filters and estimators' [author's reply]," *IEEE Transactions on Automatic Control*, Vol. 47, No. 8, pp. 1408–1409, 2002.

[19] S. J. Julier, "The spherical simplex unscented transformation," in *Proceedings of the 2003 American Control Conference*, Denver, Colorado, Vol. 3, pp. 2430–2434, 2003.

[20] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, pp. 401–422, 2004.

[21] S. J. Julier and J. K. Uhlmann, "Corrections to Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, No. 12, p. 1958, 2004.

[22] J. J. LaViola, "A comparison of unscented and extended Kalman filtering for estimating quaternion motion," in *Proceedings of the 2003 American Control Conference*, Denver, Colorado, Vol. 3, pp. 2435–2440, June2003.

[23] R. Van der Merwe and E. A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, Vol. 6, pp. 3461–3464, 2001.

[24] J. R. Van Zandt, "A more robust unscented transform," *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, San Diego, CA, 2001.

[25] T. Lefebvre, H. Bruyninckx, and J. De Schutter, *"The linear regression Kalman filter,"* in *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*, *Springer Tracts in Advanced Robotics*, Springer-Verlag, Berlin, Vol. 19, pp. 205–210, 2005.

[26] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Comment on A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, Vol. 47, No. 8, pp. 1406–1408, 2002.

[27] A. Bain and D. Crisan, *Fundamentals of Stochastic Filtering*, *Stochastic Modeling and Applied Probability Series*, No. 60, Springer, New York, 2009.

[28] L. V. Bagaschi, "A comparative study of nonlinear tracking algorithms," Doctorate dissertation, Swiss Federation of Technology (ETH), Zurich, 1991.

[29] A. Farina, D. Benvenuti, and B. Ristic, "A comparative study of the Benes filtering problem," *Signal Processing*, Vol. 82, No. 2, pp. 133–147, 2002.

[30] D. L. Ocone, "Asymptotic stability of Beneš filters," *Stochastic Analysis and Applications*, Vol. 17, No. 6, pp. 1053–1074, 1999.

[31] F. E. Daum, "Exact finite-dimensional nonlinear filters," *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 7, pp. 616–622, 1986.

[32] H. Risken and T. Frank, *The Fokker–Planck equation: methods of solution and applications*, 2nd ed., Springer, Berlin, 1989.

[33] J. M. Richardson and K. A. Marsh, *"Point process theory and the surveillance of many objects,"* in *Maximum Entropy and Bayesian Methods*, Seattle, Kluwer Academic Publishers, Norwell, MA, pp. 213–220, 1991.

[34] H. W. Sorenson, *"Kalman filtering techniques,"* in *Advances in Control Systems* Vol. 3 (C. T. Leondes, ed.), Academic Press, New York, pp. 219–292, 1966.

[35] V. Beneš and R. J. Elliott, "Finite-dimensional solutions of a modified Zakai equation," *Mathematics of Control, Signals, and Systems*, 9, pp. 341–351, 1996.

[36] A. Budhirajaa, L. Chenb, and C. Leea, "A survey of numerical methods for nonlinear filtering problems," *Physica D: Nonlinear Phenomena*, Vol. 230, No. 1 and 2, pp. 27–36, 2007.

[37] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, New York, 2001.

[38] R. Eckhardt, "Stan Ulam, John von Neumann, and the Monte Carlo Method," *Los Alamos Science*, No. 15, Special Issue: Stanislaw Ulam (1909–1984), pp. 131–141, 1987.

[39] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech, Boston, MA, 2004.