

13

Data Fusion in Nonlinear Systems

- 13.1 Introduction
- 13.2 Estimation in Nonlinear Systems
 - Problem Statement • The Transformation of Uncertainty
- 13.3 The Unscented Transformation (UT)
 - The Basic Idea • An Example Set of Sigma Points • Properties of the Unscented Transformation
- 13.4 Uses of the Transformation
 - Polar to Cartesian Coordinates • A Discontinuous Transformation
- 13.5 The Unscented Filter (UF)
- 13.6 Case Study: Using the UF with Linearization Errors
- 13.7 Case Study: Using the UF with a High-Order Nonlinear System
- 13.8 Multilevel Sensor Fusion
- 13.9 Conclusions
- Acknowledgments
- References

Simon Julier

IDAK Industries

Jeffrey K. Uhlmann

University of Missouri

13.1 Introduction

The extended Kalman filter (EKF) has been one of the most widely used methods for tracking and estimation based on its apparent simplicity, optimality, tractability, and robustness. However, after more than 30 years of experience with it, the tracking and control community has concluded that the EKF is difficult to implement, difficult to tune, and only reliable for systems that are almost linear on the time scale of the update intervals. This chapter reviews the unscented transformation (UT), a mechanism for propagating mean and covariance information through nonlinear transformations, and describes its implications for data fusion. This method is more accurate, is easier to implement, and uses the same order of calculations as the EKF. Furthermore, the UT permits the use of Kalman-type filters in applications where, traditionally, their use was not possible. For example, the UT can be used to rigorously integrate artificial intelligence-based systems with Kalman-based systems.

Performing data fusion requires estimates of the state of a system to be converted to a common representation. The mean and covariance representation is the *lingua franca* of modern systems engineering. In particular, the covariance intersection (CI)¹ and Kalman filter (KF)² algorithms provide mechanisms for fusing state estimates defined in terms of means and covariances, where each mean vector defines the nominal state of the system and its associated error covariance matrix defines a lower bound on the squared error. However, most data fusion applications require the fusion of mean and covariance estimates defining the state of a system in different coordinate frames. For example, a tracking

system might maintain estimates in a global Cartesian coordinate frame, while observations of the tracked objects are generated in the local coordinate frames of various sensors. Therefore, a transformation must be applied to convert between the global coordinate frame and each local coordinate frame.

If the transformation between coordinate frames is linear, the linearity properties of the mean and covariance makes the application of the transformation trivial. Unfortunately, most tracking sensors take measurements in a local polar or spherical coordinate frame (i.e., they measure range and bearings) that is not linearly transformable to a Cartesian coordinate frame. Rarely are the natural coordinate frames of two sensors linearly related. This fact constitutes a fundamental problem that arises in virtually all practical data fusion systems.

The UT, a mechanism that addresses the difficulties associated with converting mean and covariance estimates from one coordinate frame to another, can be applied to obtain mean and covariance estimates from systems that do not inherently produce estimates in that form. For example, this chapter describes how the UT can allow high-level artificial intelligence (AI) and fuzzy control systems to be integrated seamlessly with low-level KF and CI systems.

The structure of this chapter is as follows: Section 13.2 describes the nonlinear transformation problem within the Kalman filter framework and analyzes the KF prediction problem in detail. The UT is introduced and its performance is analyzed in Section 13.3. Section 13.4 demonstrates the effectiveness of the UT with respect to a simple nonlinear transformation (polar to Cartesian coordinates with large bearing uncertainty) and a simple discontinuous system. Section 13.5 examines how the transformation can be embedded into a fully recursive estimator that incorporates process and observation noise. Section 13.6 discusses the use of the UT in a tracking example, and Section 13.7 describes its use with a complex process and observation model. Finally, Section 13.8 shows how the UT ties multiple levels of data fusion together into a single, consistent framework.

13.2 Estimation in Nonlinear Systems

13.2.1 Problem Statement

Minimum mean squared error (MMSE) estimators can be broadly classified into linear and nonlinear estimators. Of the linear estimators, by far the most widely used is the Kalman filter.^{2*} Many researchers have attempted to develop suitable nonlinear MMSE estimators. However, the optimal solution requires that a complete description of the conditional probability density be maintained,³ and this exact description requires a potentially unbounded number of parameters. As a consequence, many suboptimal approximations have been proposed in the literature. Traditional methods are reviewed by A. H. Jazwinski⁴ and P. S. Maybeck.⁵ Recent algorithms have been proposed by F. E. Daum,⁶ N. J. Gordon et al.,⁷ and M. A. Kouritzin.⁸ Despite the sophistication of these and other approaches, the extended Kalman filter (EKF) remains the most widely used estimator for nonlinear systems.^{9,10} The EKF applies the Kalman filter to nonlinear systems by simply linearizing all of the nonlinear models so that the traditional linear Kalman filter equations can be applied. However, in practice, the EKF has three well-known drawbacks:

1. Linearization can produce highly unstable filters if the assumption of local linearity is violated. Examples include estimating ballistic parameters of missiles¹¹⁻¹⁴ and some applications of computer vision.¹⁵ As demonstrated later in this chapter, some extremely common transformations that are used in target tracking systems are susceptible to these problems.

*Researchers often (and incorrectly) claim that the Kalman filter can be applied only if the following two conditions hold: (i) all probability distributions are Gaussian and (ii) the system equations are linear. The Kalman filter is, in fact, the minimum mean squared *linear* estimator that can be applied to *any* system with *any* distribution, provided the first two moments are known. However, it is only the globally optimal estimator under the special case that the distributions are all Gaussian.

2. Linearization can be applied only if the Jacobean matrix exists, and the Jacobian matrix exists only if the system is differentiable at the estimate. Although this constraint is satisfied by the dynamics of continuous physical systems, some systems do not satisfy this property. Examples include jump-linear systems, systems whose sensors are quantized, and expert systems that yield a finite set of discrete solutions.
3. Finally, the derivation of the Jacobian matrices is nontrivial in most applications and can often lead to significant implementation difficulties. In P. A. Dulimov,¹⁶ for example, the derivation of a Jacobian requires six pages of dense algebra. Arguably, this has become less of a problem, given the widespread use of symbolic packages such as Mathematica¹⁷ and Maple.¹⁸ Nonetheless, the computational expense of calculating a Jacobian can be extremely high if the expressions for the terms are nontrivial.

Appreciating how the UT addresses these three problems requires an understanding of some of the mechanics of the KF and EKF.

Let the state of the system at a time step k be the state vector $\mathbf{x}(k)$. The Kalman filter propagates the first two moments of the distribution of $\mathbf{x}(k)$ recursively and has a distinctive “predictor-corrector” structure. Let $\hat{\mathbf{x}}(i|j)$ be the estimate of $\mathbf{x}(i)$ using the observation information up to and including time j , $\mathbf{Z}^j = [\mathbf{z}(1), \dots, \mathbf{z}(j)]$. The covariance of this estimate is $\mathbf{P}(i|j)$. Given an estimate $\hat{\mathbf{x}}(k|k)$, the filter first predicts what the future state of the system will be using the process model. Ideally; the predicted quantities are given by the expectations

$$\hat{\mathbf{x}}(k+1|k) = E \left[\mathbf{f} \left[\mathbf{x}(k), \mathbf{u}(k), \mathbf{v}(k), k \right] \middle| \mathbf{Z}^k \right] \quad (13.1)$$

$$\mathbf{P}(k+1|k) = E \left[\left\{ \mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|k) \right\} \left\{ \mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|k) \right\}^T \middle| \mathbf{Z}^k \right] \quad (13.2)$$

When $\mathbf{f}[\cdot]$ and $\mathbf{h}[\cdot]$ are nonlinear, the precise values of these statistics can be calculated only if the distribution of $\mathbf{x}(k)$ is perfectly known. However, this distribution has no general form, and a potentially unbounded number of parameters are required. Therefore, in most practical algorithms these expected values must be approximated.

The estimate $\hat{\mathbf{x}}(k+1|k+1)$ is found by updating the prediction with the current sensor measurement. In the Kalman filter, a linear update rule is specified and the weights are chosen to minimize the mean squared error of the estimate.

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\mathbf{v}(k+1) \\ \mathbf{P}(k+1|k+1) &= \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{P}_{vv}(k+1|k)\mathbf{W}^T(k+1) \\ \mathbf{v}(k+1) &= \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1|k) \\ \mathbf{W}(k+1) &= \mathbf{P}_{xv}(k+1|k)\mathbf{P}_{vv}^{-1}(k+1|k) \end{aligned} \quad (13.3)$$

Note that these equations are only a function of the predicted values of the first two moments of $\mathbf{x}(k)$ and $\mathbf{z}(k)$. Therefore, the problem of applying the Kalman filter to a nonlinear system is the ability to predict the first two moments of $\mathbf{x}(k)$ and $\mathbf{z}(k)$.

13.2.2 The Transformation of Uncertainty

The problem of predicting the future state or observation of the system can be expressed in the following form. Suppose that $\mathbf{x}$ is a random variable with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$. A second random variable, $\mathbf{y}$, is related to $\mathbf{x}$ through the nonlinear function

$$\mathbf{y} = \mathbf{f}[\mathbf{x}] \quad (13.4)$$

The mean $\bar{\mathbf{y}}$ and covariance $\mathbf{P}_{yy}$ of $\mathbf{y}$ must be calculated.

The statistics of $\mathbf{y}$ are calculated by (1) determining the density function of the transformed distribution and (2) evaluating the statistics from that distribution. In some special cases, exact, closed form solutions exist (e.g., when $\mathbf{f}[\cdot]$ is linear or is one of the forms identified in F. E. Daum⁶). However; as explained above, most data fusion problems do not possess closed-form solutions and some kind of an approximation must be used. A common approach is to develop a transformation procedure from the Taylor series expansion of Equation 13.4 about $\bar{\mathbf{x}}$. This series can be expressed as

$$\begin{aligned} \mathbf{f}[\mathbf{x}] &= \mathbf{f}[\bar{\mathbf{x}} + \delta\mathbf{x}] \\ &= \mathbf{f}[\bar{\mathbf{x}}] + \nabla\mathbf{f}\delta\mathbf{x} + \frac{1}{2}\nabla^2\mathbf{f}\delta\mathbf{x}^2 + \frac{1}{3!}\nabla^3\mathbf{f}\delta\mathbf{x}^3 + \frac{1}{4!}\nabla^4\mathbf{f}\delta\mathbf{x}^4 + \dots \end{aligned} \quad (13.5)$$

where $\delta\mathbf{x}$ is a zero mean Gaussian variable with covariance $\mathbf{P}_{xx}$ and $\nabla^n\mathbf{f}\delta\mathbf{x}^n$ is the appropriate n th order term in the multidimensional Taylor Series. The transformed mean and covariance are

$$\bar{\mathbf{y}} = \mathbf{f}[\bar{\mathbf{x}}] + \frac{1}{2}\nabla^2\mathbf{f}\mathbf{P}_{xx} + \frac{1}{2}\nabla^4\mathbf{f}E[\delta\mathbf{x}^4] + \dots \quad (13.6)$$

$$\begin{aligned} \mathbf{P}_{yy} &= \nabla\mathbf{f}\mathbf{P}_{xx}(\nabla\mathbf{f})^T + \frac{1}{2 \times 4!}\nabla^2\mathbf{f}\left(E[\delta\mathbf{x}^4] - E[\delta\mathbf{x}^2\mathbf{P}_{yy}] - E[\mathbf{P}_{yy}\delta\mathbf{x}^2] + \mathbf{P}_{yy}^2\right)(\nabla^2\mathbf{f})^T + \\ &\quad \frac{1}{3!}\nabla^3\mathbf{f}E[\delta\mathbf{x}^4](\nabla\mathbf{f})^T + \dots \end{aligned} \quad (13.7)$$

In other words, the n th order term in the series for $\bar{\mathbf{x}}$ is a function of the n th order moments of $\mathbf{x}$ multiplied by the n th order derivatives of $\mathbf{f}[\cdot]$ evaluated at $\mathbf{x} = \bar{\mathbf{x}}$. If the moments and derivatives can be evaluated correctly up to the n th order, the mean is correct up to the n th order as well. Similar comments hold for the covariance equation, although the structure of each term is more complicated. Since each term in the series is scaled by a progressively smaller and smaller term the lowest-order terms in the series are likely to have the greatest impact. Therefore, the prediction procedure should be concentrated on evaluating the lower order terms.

The EKF exploits linearization. Linearization assumes that the second- and higher-order terms of $\delta\mathbf{x}$ in Equation 13.5 can be neglected. Under this assumption,

$$\bar{\mathbf{y}} = \mathbf{f}[\bar{\mathbf{x}}] \quad (13.8)$$

$$\mathbf{P}_{yy} = \nabla\mathbf{f}\mathbf{P}_{xx}(\nabla\mathbf{f})^T \quad (13.9)$$

However, in many practical situations, linearization introduces significant biases or errors. These cases require more accurate prediction techniques.

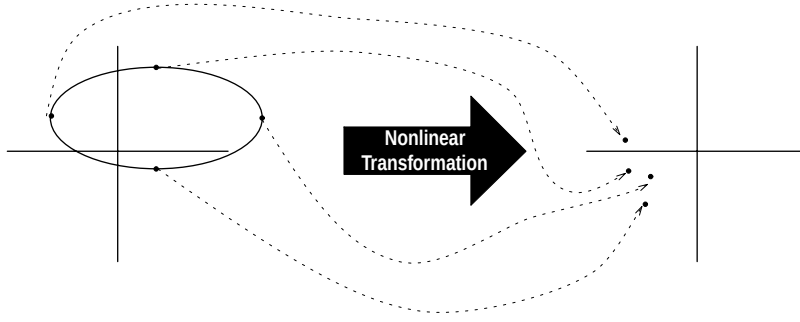


FIGURE 13.1 The principle of the unscented transformation.

13.3 The Unscented Transformation (UT)

13.3.1 The Basic Idea

The UT is a method for calculating the statistics of a random variable that undergoes a nonlinear transformation. This method is founded on the intuition that *it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation*.¹⁹ The approach is illustrated in Figure 13.1. A set of points (*sigma points*) is chosen with sample mean and sample covariance of the nonlinear function is $\bar{\mathbf{x}}$ and $\mathbf{P}_{xx}$. The nonlinear function is applied to each point, in turn, to yield a cloud of transformed points; $\bar{\mathbf{y}}$ and $\mathbf{P}_{yy}$ are the statistics of the transformed points.

Although this method bears a superficial resemblance to Monte Carlo-type methods, there is an extremely important and fundamental difference. The samples are not drawn at random; they are drawn according to a specific, deterministic algorithm. Since the problems of statistical convergence are not relevant, high-order information about the distribution can be captured using only a very small number of points. For an n -dimensional space, only $n + 1$ points are needed to capture any given mean and covariance. If the distribution is known to be symmetric, $2n$ points are sufficient to capture the fact that the third- and all higher-order odd moments are zero for any symmetric distribution.¹⁹

The set of sigma points, S , consists of l vectors and their appropriate weights, $S = \{i = 0, 0, \dots, l - 1 : \mathbf{X}_i, W_i\}$. The weights W_i can be positive or negative but must obey the normalization condition

$$\sum_{i=0}^{l-1} W_i = 1 \quad (13.10)$$

Given these points, $\bar{\mathbf{y}}$ and $\mathbf{P}_{yy}$ are calculated using the following procedure:

1. Instantiate each point through the function to yield the set of transformed sigma points,

$$\mathbf{y}_i = \mathbf{f}[\mathbf{X}_i]$$

2. The mean is given by the weighted average of the transformed points,

$$\bar{\mathbf{y}} = \sum_{i=0}^{l-1} W_i \mathbf{y}_i \quad (13.11)$$

3. The covariance is the weighted outer product of the transformed points,

$$\mathbf{P}_{yy} = \sum_{i=0}^{l-1} W_i \{y_i - \bar{y}\} \{y_i - \bar{y}\}^T \quad (13.12)$$

The crucial issue is to decide how many sigma points should be used, where they should be located, and what weights they should be assigned. The points should be chosen so that they capture the “most important” properties of $\mathbf{x}$. This can be formalized as follows. Let $P_x(\mathbf{x})$ be the density function of $\mathbf{x}$. The sigma points capture the necessary properties by obeying the condition

$$\mathbf{g}[S, p_x(\mathbf{x})] = 0$$

The decision as to which properties of $\mathbf{x}$ are to be captured precisely and which are to be approximated is determined by the demands of the particular application in question. Here, the *moments* of the distribution of the sigma points are matched with those of $\mathbf{x}$. This is motivated by the Taylor series expansion, given in Section 13.2.2, which shows that matching the moments of $\mathbf{x}$ up to the n th order means that Equations 13.11 and 13.12 capture $\bar{y}$ and $\mathbf{P}_{yy}$ up to the n th order as well.²⁰

Note that the UT is distinct from other efforts published in the literature. First, some authors have considered the related problem of assuming that the distribution takes on a particular parameterized form, rather than an entire, arbitrary distribution. Kushner, for example, describes an approach whereby a distribution is approximated at each time step by a Gaussian.²¹ However, the problem with this approach is that it does not address the fundamental problem of calculating the mean and covariance of the nonlinearly transformed distribution. Second, the UT bears some relationship to quadrature, which has been used to approximate the integrations implicit in statistical expectations. However, the UT avoids some of the difficulties associated with quadrature methods by approximating the unknown distribution. In fact, the UT is most closely related to perturbation analysis. In a 1989 article, Holtzmann introduced a noninfinitesimal perturbation for a scalar system.²² Holtzmann’s solution corresponds to that of the symmetric UT in the scalar case, but their respective generalizations (e.g., to higher dimensions) are not equivalent.

13.3.2 An Example Set of Sigma Points

A set of sigma points can be constructed using the constraints that they capture the first three moments of a symmetric distribution: $\mathbf{g}[S, p_x(\mathbf{x})] = [\mathbf{g}_1[S, p_x(\mathbf{x})] \ \mathbf{g}_2[S, p_x(\mathbf{x})] \ \mathbf{g}_3[S, p_x(\mathbf{x})]]^T$ where

$$\mathbf{g}_1[S, p_x(\mathbf{x})] = \sum_{i=0}^p W_i \mathbf{X}_i - \hat{\mathbf{x}} \quad (13.13)$$

$$\mathbf{g}_2[S, p_x(\mathbf{x})] = \sum_{i=0}^p W_i (\mathbf{X}_i - \bar{\mathbf{x}})^2 - \mathbf{P}_{xx} \quad (13.14)$$

$$\mathbf{g}_3[S, p_x(\mathbf{x})] = \sum_{i=0}^p W_i (\mathbf{X}_i - \bar{\mathbf{x}})^3 \quad (13.15)$$

The set is²³

$$\begin{aligned}
\mathbf{X}_0(k|k) &= \hat{\mathbf{x}}(k|k) \\
W_0 &= \kappa / (n + \kappa) \\
\mathbf{X}_i(k|k) &= \hat{\mathbf{x}}(k|k) + \left(\sqrt{(n + \kappa) \mathbf{P}(k|k)} \right)_i \\
W_i &= 1 / \left\{ 2(n + \kappa) \right\} \\
\mathbf{X}_{i+n}(k|k) &= \hat{\mathbf{x}}(k|k) - \left(\sqrt{(n + \kappa) \mathbf{P}(k|k)} \right)_i \\
W_{i+n} &= 1 / \left\{ 2(n + \kappa) \right\}
\end{aligned} \tag{13.16}$$

where κ is a real number, $\left(\sqrt{(n + \kappa) \mathbf{P}(k|k)} \right)_i$ is the i th row or column* of the matrix square root of $(n + \kappa) \mathbf{P}(k|k)$, and W_i is the weight associated with the i th point.

13.3.3 Properties of the Unscented Transform

Despite its apparent similarity to other efforts described in the data fusion literature, the UT has a number of features that make it well suited for the problem of data fusion in practical problems:

- The UT can predict with the same accuracy as the second-order Gauss filter, but without the need to calculate Jacobians or Hessians. The reason is that the mean and covariance of $\mathbf{x}$ are captured precisely up to the second order, and the calculated values of the mean and covariance of $\mathbf{y}$ also are correct to the second order. This indicates that the mean is calculated to a higher order of accuracy than the EKF, whereas the covariance is calculated to the same order of accuracy.
- The computational cost of the algorithm is the same order of magnitude as the EKF. The most expensive operations are calculating the matrix square root and determining the outer product of the sigma points to calculate the predicted covariance. However, both operations are $O(n^3)$, which is the same cost as evaluating the $n \times n$ matrix multiplies needed to calculate the predicted covariance.**
- The algorithm naturally lends itself to a “black box” filtering library. The UT calculates the mean and covariance using standard vector and matrix operations and does not exploit details about the specific structure of the model.
- The algorithm can be used with distributions that are not continuous. Sigma points can straddle a discontinuity. Although this does not precisely capture the effect of the discontinuity, its effect is to spread the sigma points out such that the mean and covariance reflect the presence of the discontinuity.
- The UT can be readily extended to capture more information about the distribution. Because the UT captures the properties of the distribution, a number of refinements can be applied to improve greatly the performance of the algorithm. If only the first two moments are required, then $n + 1$ sigma points are sufficient. If the distribution is assumed or is known to be symmetric, then $n + 2$

*If the matrix square root $\mathbf{A}$ of $\mathbf{P}$ is of the form $\mathbf{P} = \mathbf{A}\mathbf{A}^T$, then the sigma points are formed from the *rows* of $\mathbf{A}$. However, for a root of the form $\mathbf{P} = \mathbf{A}\mathbf{A}^T$, the *columns* of $\mathbf{A}$ are used.

**The matrix square root should be calculated using numerically efficient and stable methods such as the Cholesky decomposition.²⁴

sigma points are sufficient. Therefore, the total number of calculations required for calculating the new covariance is $O(n^3)$, which is the same order as that required by the EKF. The transform has also been demonstrated to propagate successfully the fourth-order moment (or kurtosis) of a Gaussian distribution²⁵ and that it can be used to propagate the third-order moments (or skew) of an arbitrary distribution.²⁶

13.4 Uses of the Transformation

This section demonstrates the effectiveness of the UT with respect to two nonlinear systems that represent important classes of problems encountered in the data fusion literature — coordinate conversions and discontinuous systems.

13.4.1 Polar to Cartesian Coordinates

One of the most important transformations in target tracking is the conversion from polar to Cartesian coordinates. This transformation is known to be highly susceptible to linearization errors. D. Lerro and Y. Bar-Shalom, for example, show that the linearized conversion can become inconsistent when the standard deviation in the bearing estimate is less than a degree.²⁷ This subsection illustrates the use of the UT on a coordinate conversion problem with extremely high angular uncertainty.

Suppose a mobile autonomous vehicle detects targets in its environment using a range-optimized sonar sensor. The sensor returns polar information (range, r , and bearing, θ), which is converted to estimate Cartesian coordinates. The transformation is

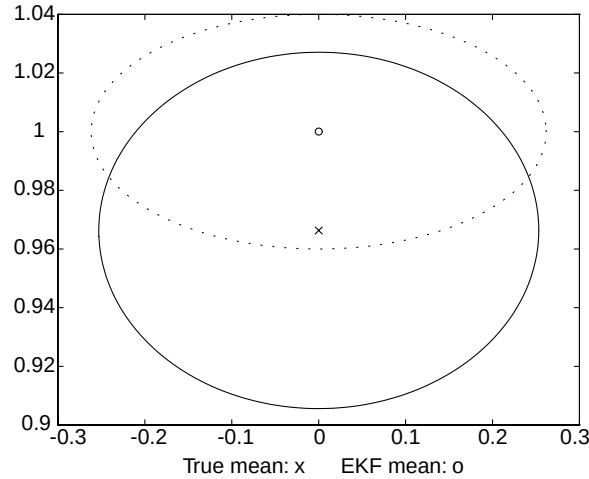
$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r \cos \theta \\ r \sin \theta \end{pmatrix} \text{ with } \nabla \mathbf{f} = \begin{bmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{bmatrix}$$

The real location of the target is $(0, 1)$. The difficulty with this transformation arises from the physical properties of the sonar. Fairly good range accuracy (with 2cm standard deviation) is traded off to give a very poor bearing measurement (standard deviation of 15°).²⁸ The large bearing uncertainty causes the assumption of local linearity to be violated.

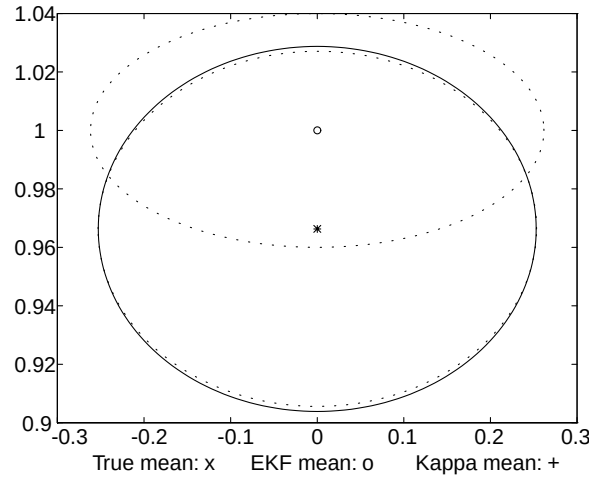
To appreciate the errors that can be caused by linearization, compare its values for the statistics of (x, y) with those of the true statistics calculated by Monte Carlo simulation. Due to the slow convergence of random sampling methods, an extremely large number of samples (3.5×10^6) were used to ensure that accurate estimates of the true statistics were obtained. The results are shown in [Figure 13.2\(a\)](#). This figure shows the mean and 1σ contours calculated by each method. The 1σ contour is the locus of points $\{\mathbf{y} : (\mathbf{y} - \bar{\mathbf{y}})^T \mathbf{P}_y^{-1} (\mathbf{y} - \bar{\mathbf{y}}) = 1\}$ and is a graphical representation of the size and orientation of $\mathbf{P}_{yy}$. The figure demonstrates that the linearized transformation is biased and inconsistent. This is most pronounced along the y -axis, where linearization estimates that the position is 1m, whereas in reality it is 96.7cm. In this example, linearization errors effectively introduce an error which is over 1.5 times the standard deviation of the range measurement. Since it is a bias that arises from the transformation process itself, the same error with the same sign will be committed each time a coordinate transformation takes place. Even if there were no bias, the transformation would still be inconsistent because its ellipse is not sufficiently extended along the y -axis.

In practice, this inconsistency can be resolved by introducing additional stabilizing noise that increases the size of the transformed covariance. This is one possible explanation of why EKFs are difficult to tune — sufficient noise must be introduced to offset the defects of linearization. However, introducing stabilizing noise is an undesirable solution because the estimate remains biased and there is no general guarantee that the transformed estimate remains consistent or efficient.

The performance benefits of using the UT can be seen in [Figure 13.2\(b\)](#), which shows the means and 1σ contours determined by the different methods. The mismatch between the UT mean and the true mean is extremely small (approximately 6×10^{-4}). The transformation is consistent, ensuring that the



(A)



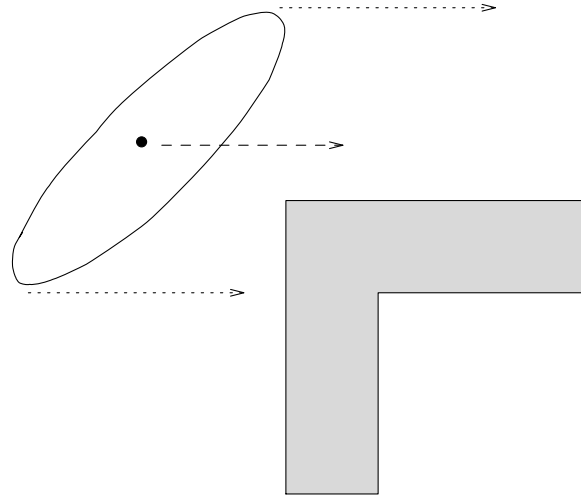
(B)

FIGURE 13.2 The mean and standard deviation ellipses for the true statistics, those calculated through linearization and those calculated by the unscented transformation. (A) Results from linearization. The true mean is at $\times$ and the uncertainty ellipse is solid. Linearization calculates the mean at $\circ$ and the uncertainty ellipse is dashed. (B) Results from the UT. The true mean is at $\times$ and the uncertainty ellipse is dotted. The UT mean is at $+$ (overlapping the position of the true mean) and is the solid ellipse. The linearized mean is at $\circ$ and its ellipse is also dotted.

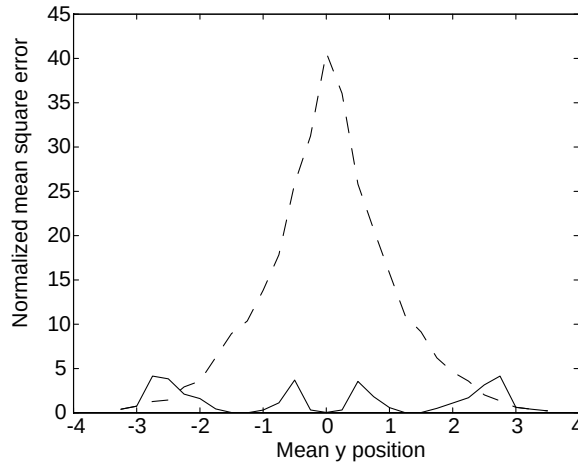
filter does not diverge. As a result, there is no need to introduce artificial noise terms that would degrade performance even when the angular uncertainty is extremely high.

13.4.2 A Discontinuous Transformation

Consider the behavior of a two-dimensional particle whose state consists of its position $\mathbf{x}(k) = [x(k), y(k)]^T$. The projectile is initially released at time 1 and travels at a constant and known speed, v_x , in the x direction. The objective is to estimate the mean position and covariance of the position at time 2, $[x(2), y(2)]^T$, where $\Delta T \triangleq t_2 - t_1$. The problem is made difficult by the fact that the path of the projectile is obstructed by a wall that lies in the “bottom right quarter-plane” ($x \geq 0, y \leq 0$). If the projectile hits the wall, a perfectly elastic collision will occur, and the projectile will be reflected back at the same velocity



(A)



(B)

FIGURE 13.3 A discontinuous system example: a particle can either strike a wall and rebound, or continue to move in a straight line. The experimental results show the effect of using different start values for y .

as it traveled forward. This situation is illustrated in [Figure 13.3\(A\)](#), which also shows the covariance ellipse of the initial distribution.

The process model for this system is

$$x(2) = \begin{cases} x(1) + \Delta T v_x & y(1) \geq 0 \\ x(1) - \Delta T v_x & y(1) < 0 \end{cases} \quad (13.17)$$

$$y(2) = y(1) \quad (13.18)$$

At time 1, the particle starts in the left half-plane ($x \leq 0$) with position $[x(1), y(1)]^T$. The error in this estimate is Gaussian, has zero mean, and has covariance $\mathbf{P}(1|1)$. Linearized about this start condition, the system appears to be a simple constant velocity linear model.

1. The set of sigma points are created by applying a sigma point selection algorithm (e.g., Equation 13.16) to the augmented system given by Equation 13.21.
2. The transformed set is given by instantiating each point through the process model,

$$\mathbf{X}_i(k+1|k) = \mathbf{f}\left[\mathbf{X}_i(k|k), \mathbf{u}(k), k\right]$$

3. The predicted mean is computed as

$$\hat{\mathbf{x}}(k+1|k) = \sum_{i=0}^{2n^a} W_i \mathbf{X}_i(k+1|k)$$

4. And the predicted covariance is computed as

$$\mathbf{P}(k+1|k) = \sum_{i=0}^{2n^a} W_i \left\{ \mathbf{X}_i(k+1|k) - \hat{\mathbf{x}}(k+1|k) \right\} \left\{ \mathbf{X}_i(k+1|k) - \hat{\mathbf{x}}(k+1|k) \right\}^T$$

5. Instantiate each of the prediction points through the observation model,

$$\mathbf{Z}_i(k+1|k) = \mathbf{h}\left[\mathbf{X}_i(k+1|k), \mathbf{u}(k), k\right]$$

6. The predicted observation is calculated by

$$\hat{\mathbf{z}}(k+1|k) = \sum_{i=1}^{2n^a} W_i \mathbf{Z}_i(k+1|k)$$

7. The innovation covariance is

$$\mathbf{P}_{vv}(k+1|k) = \sum_{i=0}^{2n^a} W_i \left\{ \mathbf{Z}_i(k|k-1) - \hat{\mathbf{z}}(k+1|k) \right\} \left\{ \mathbf{Z}_i(k|k-1) - \hat{\mathbf{z}}(k+1|k) \right\}^T$$

8. The cross-correlation matrix is determined by

$$\mathbf{P}_{xz}(k+1|k) = \sum_{i=0}^{2n^a} W_i \left\{ \mathbf{X}_i(k|k-1) - \hat{\mathbf{x}}(k+1|k) \right\} \left\{ \mathbf{Z}_i(k|k-1) - \hat{\mathbf{z}}(k+1|k) \right\}^T$$

9. Finally, the update can be performed using the normal Kalman filter equations:

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1) \nu(k+1) \\ \mathbf{P}(k+1|k+1) &= \mathbf{P}(k+1|k) - \mathbf{W}(k+1) \mathbf{P}_{vv}(k+1|k) \mathbf{W}^T(k+1) \\ \nu(k+1) &= \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1|k) \\ \mathbf{W}(k+1) &= \mathbf{P}_{xz}(k+1|k) \mathbf{P}_{vv}^{-1}(k+1|k) \end{aligned}$$

FIGURE 13.4 A general formulation of the Kalman filter using the unscented transformation. As explained in the text, there is a significant scope for optimizing this algorithm.

The true conditional mean and covariance was determined using Monte Carlo simulation for different choices of the initial mean of y . The mean squared error calculated by the EKF and by the UT for different values is shown in Figure 13.3(B). The UT estimates the mean very closely, suffering only small spikes as the translated sigma points successively pass the wall. Further analysis shows that the covariance for the filter is only slightly larger than the true covariance, but conservative enough to account for the deviation of its estimated mean from the true mean. The EKF, however, bases its entire estimate of the

conditional mean on the projection of the prior mean; therefore, its estimates bear no resemblance to the true mean, except when most of the distribution either hits or misses the wall and the effect of the discontinuity is minimized.

13.5 The Unscented Filter

The UT can be used as the cornerstone of a recursive Kalman-type of estimator. The transformation processes that occur in a Kalman filter (Equation 13.3) consist of the following steps:

1. Predict the new state of the system, $\hat{\mathbf{x}}(k+1|k)$, and its associated covariance, $\mathbf{P}(k+1|k)$. This prediction must take into account the effects of process noise.
2. Predict the expected observation, $\hat{\mathbf{z}}(k+1|k)$, and the innovation covariance, $\mathbf{P}_{vv}(k+1|k)$. This prediction should include the effects of observation noise.
3. Finally, predict the cross-correlation matrix, $\mathbf{P}_{xz}(k+1|k)$.

These steps can be easily accommodated by slightly restructuring the state vector and process and observation models. The most general formulation augments the state vector with the process and noise terms to give an $n^a = n + q + r$ dimensional vector,

$$\mathbf{x}^a(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{v}(k) \\ \mathbf{w}(k) \end{bmatrix} \quad (13.19)$$

The process and observation models are rewritten as a function of $\mathbf{x}^a(k)$,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}^a[\mathbf{x}^a(k), \mathbf{u}(k), k] \\ \mathbf{z}(k+1) &= \mathbf{h}^a[\mathbf{x}^a(k+1), \mathbf{u}(k), k] \end{aligned} \quad (13.20)$$

and the UT uses sigma points that are drawn from

$$\hat{\mathbf{x}}^a(k|k) = \begin{bmatrix} \hat{\mathbf{x}}(k|k) \\ 0_{q \times 1} \\ 0_{m \times 1} \end{bmatrix} \text{ and } \mathbf{P}^a(k|k) = \begin{bmatrix} \mathbf{P}(k|k) & 0 & 0 \\ 0 & \mathbf{Q}(k) & 0 \\ 0 & 0 & \mathbf{R}(k) \end{bmatrix} \quad (13.21)$$

The matrices on the leading diagonal are the covariances and the off-diagonal sub-blocks are the correlations between the state errors and the process noises.* Although this method requires the use of additional sigma points, it incorporates the noises into the predicted state with the same level of accuracy as the propagated estimation errors. In other words, the estimate is correct to the second order and no Jacobians, Hessians, or other numerical approximations must be calculated.

*If correlations exist between the noise terms, Equation 13.21 can be generalized to draw the sigma points from the covariance matrix

$$\mathbf{P}^a(k|k) = \begin{bmatrix} \mathbf{P}(k|k) & \mathbf{P}_{xv}(k|k) & \mathbf{P}_{xw}(k|k) \\ \mathbf{P}_{vx}(k|k) & \mathbf{Q}(k) & \mathbf{P}_{vw}(k|k) \\ \mathbf{P}_{wx}(k|k) & \mathbf{P}_{wv}(k|k) & \mathbf{R}(k) \end{bmatrix}$$

Such correlation structures commonly arise in algorithms such as the Schmidt-Kalman filter.²⁹

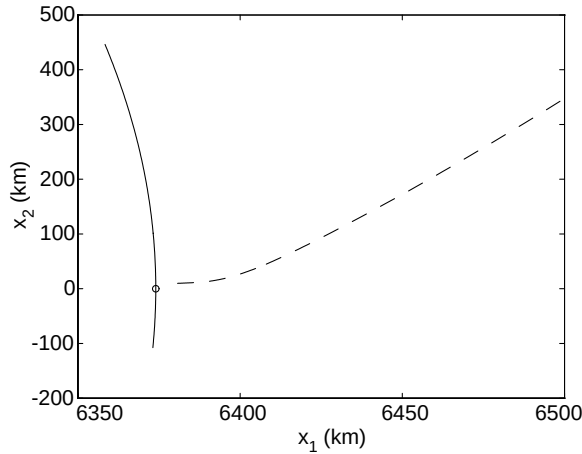


FIGURE 13.5 The reentry problem. The dashed line is the sample vehicle trajectory and the solid line is a portion of the Earth's surface. The position of the radar is marked by a C.

The full unscented filter is summarized in Figure 13.4. However, recall that this is the most general form of the UF and many optimizations can be made. For example, if the process model is linear, but the observation model is not, the normal linear Kalman filter prediction equations can be used to calculate $\hat{\mathbf{x}}(k+1|k)$ and $\mathbf{P}(k+1|k)$. The sigma points would be drawn from the prediction distribution and would only be used to calculate $\hat{\mathbf{z}}(k+1|k)$, $\mathbf{P}_{xy}(k+1|k)$, and $\mathbf{P}_{yy}(k+1|k)$.

The following two sections describe the application of the unscented filter to two case studies. The first demonstrates the accuracy of the recursive filter, and the second considers the problem of an extremely involved process model.

13.6 Case Study: Using the UF with Linearization Errors

This section considers the problem that is illustrated in Figure 13.5: a vehicle entering the atmosphere at high altitude and at very high speed. The position of the body is to be tracked by a radar which accurately measures range and bearing. This type of problem has been identified by a number of authors¹¹⁻¹⁴ as being particularly stressful for filters and trackers, based on the strongly nonlinear nature of three types of forces that act on the vehicle. The most dominant is aerodynamic drag, which is a function of vehicle speed and has a substantial nonlinear variation in altitude. The second type of force is gravity, which accelerates the vehicle toward the center of the earth. The final type of force is random buffeting. The effect of these forces gives a trajectory of the form shown in Figure 13.5. Initially the trajectory is almost ballistic; however, as the density of the atmosphere increases, drag effects become important and the vehicle rapidly decelerates until its motion is almost vertical. The tracking problem is made more difficult by the fact that the drag properties of the vehicle could be only very crudely known.

In summary, the tracking system should be able to track an object which experiences a set of complicated, highly nonlinear forces. These depend on the current position and velocity of the vehicle, as well as on certain characteristics that are not precisely known. The filter's state space consists of the position of the body (x_1 and x_2), its velocity (x_3 and x_4), and a parameter of its aerodynamic properties (x_5). The vehicle state dynamics are

$$\begin{aligned}\dot{x}_1(k) &= x_3(k) \\ \dot{x}_2(k) &= x_4(k) \\ \dot{x}_5(k) &= D(k)x_3(k) + G(k)x_1(k) + v_1(k)\end{aligned}\tag{13.22}$$

$$\begin{aligned}\dot{x}_4(k) &= D(k)x_4(k) + G(k)x_2(k) + v_2(k) \\ \dot{x}_5(k) &= v_3(k)\end{aligned}$$

where $D(k)$ is the drag-related force term, $G(k)$ is the gravity-related force term, and $v_1(k)$, $v_2(k)$, and $v_3(k)$ are the process noise terms. Defining $R(k) = \sqrt{x_1^2(k) + x_2^2(k)}$ as the distance from the center of the Earth and $V(k) = \sqrt{x_3^2(k) + x_4^2(k)}$ as absolute vehicle speed, the drag and gravitational terms are

$$\begin{aligned}D(k) &= -\beta(k) \exp\left\{\left[\frac{R_0 - R(k)}{H_0}\right]\right\} V(k), \quad G(k) = -\frac{Gm_0}{r^3(k)} \\ \text{and } \beta(k) &= \beta_0 \exp x_5(k)\end{aligned}$$

For this example, the parameter values are $\beta_0 = -0.59783$, $H_0 = 13.406$, $Gm_0 = 3.9860 \times 10^5$, and $R_0 = 6374$, and they reflect typical environmental and vehicle characteristics.¹³ The parameterization of the ballistic coefficient, $\beta(k)$, reflects the uncertainty in vehicle characteristics.¹² β_0 is the ballistic coefficient of a “typical vehicle,” and it is scaled by $\exp x_5(k)$ to ensure that its value is always positive. This is vital for filter stability.

The motion of the vehicle is measured by a radar located at (x_r, y_r) . It can measure range r and bearing θ at a frequency of 10Hz, where

$$\begin{aligned}r_r(k) &= \sqrt{(x_1(k) - x_r)^2 + (x_2(k) - y_r)^2} + w_1(k) \\ \theta(k) &= \tan^{-1}\left(\frac{x_2(k) - y_r}{x_1(k) - x_r}\right) + w_2(k)\end{aligned}$$

$w_1(k)$ and $w_2(k)$ are zero mean uncorrelated noise processes with variances of 1m and 17mrad, respectively.³⁰ The high update rate and extreme accuracy of the sensor results in a large quantity of extremely high quality data for the filter.

The true initial conditions for the vehicle are

$$\mathbf{x}(0) = \begin{bmatrix} 6500.4 \\ 349.14 \\ -1.8093 \\ -6.7967 \\ 0.6932 \end{bmatrix} \quad \text{and} \quad \mathbf{P}(0) = \begin{bmatrix} 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In other words, the vehicle’s coefficient is twice the nominal coefficient.

The vehicle is buffeted by random accelerations,

$$\mathbf{Q}(k) = \begin{bmatrix} 2.4064 \times 10^{-5} & 0 & 0 \\ 0 & 2.4064 \times 10^{-5} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The initial conditions assumed by the filter are

$$\hat{\mathbf{x}}(0|0) = \begin{pmatrix} 6500.4 \\ 349.14 \\ -1.8093 \\ -6.7967 \\ 0 \end{pmatrix} \text{ and } \mathbf{P}(0|0) = \begin{bmatrix} 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The filter uses the nominal initial condition and, to offset for the uncertainty, the variance on the initial estimate is 1.

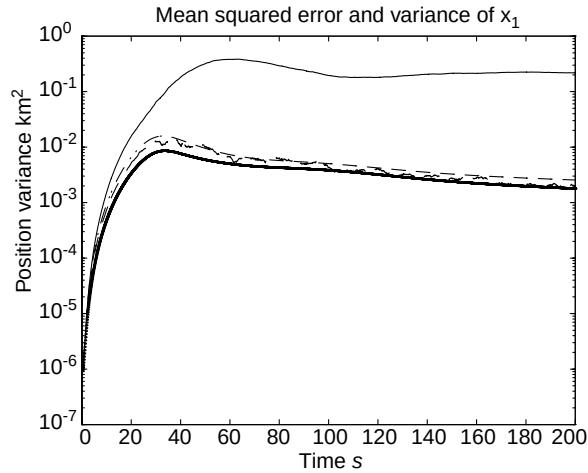
Both filters were implemented in discrete time and observations were taken at a frequency of 10Hz. However, as a result of the intense nonlinearities of the vehicle dynamics equations, the Euler approximation of Equation 13.22 was valid only for small time steps. The integration step was set to be 50ms, which meant that two predictions were made per update. For the unscented filter, each sigma point was applied through the dynamics equations twice. For the EKF, an initial prediction step and relinearization had to be performed before the second step.

The performance of each filter is shown in Figure 13.6. This figure plots the estimated mean squared estimation error (the diagonal elements of $\mathbf{P}(k|k)$) against actual mean squared estimation error (which is evaluated using 100 Monte Carlo simulations). Only x_1 , x_3 , and x_5 are shown. The results for x_2 are similar to x_1 , and the x_4 and x_3 results are the same. In all cases, the unscented filter estimates its mean squared error very accurately, maximizing the confidence of the filter estimates. The EKF, however, is highly inconsistent; the peak mean squared error in x_1 is 0.4km^2 , whereas its estimated covariance is over 100 times smaller. Similarly, the peak mean squared velocity error is $3.4 \times 10^{-4}\text{km}^2\text{s}^{-2}$, which is more than five times the true mean squared error. Finally, x_5 is highly biased, and this bias decreases only slowly over time. This poor performance shows that, even with regular and high quality updates, linearization errors inherent in the EKF can be sufficient to cause inconsistent estimates.

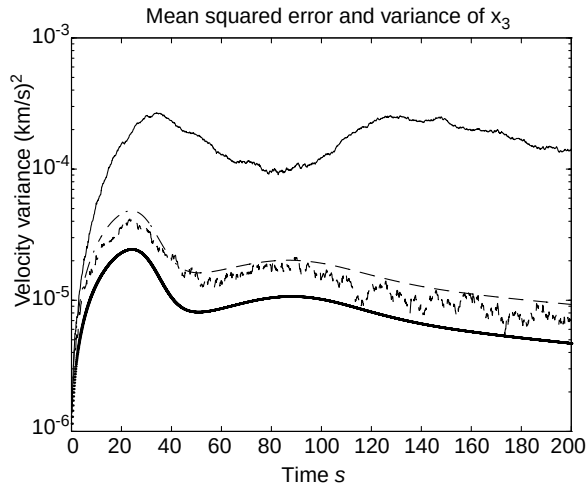
13.7 Case Study: Using the UF with a High-Order Nonlinear System

In many tracking applications, obtaining large quantities of accurate sensor data is difficult. For example, an air traffic control system might measure the location of an aircraft only once every few seconds. When information is scarce, the accuracy of the process model becomes extremely important for two reasons. First, if the control system must obtain an estimate of the target state more frequently than the tracker updates, a predicted tracker position must be used. Different models can have a significant impact on the quality of that prediction.³¹ Second, to optimize the performance of the tracker, the limited data from the sensors must be exploited to the greatest degree possible. Within the Kalman filter framework, this can only be achieved by developing the most accurate process model that is practical. However, such models can be high order and nonlinear. The UF greatly simplifies the development and refinement of such models.

This section demonstrates the ease with which the UF can be applied to a prototype Kalman filter-based localization system for a conventional road vehicle. The road vehicle, shown in Figure 13.7, undertakes maneuvers at speeds in excess of 45mph (15ms^{-1}). The position of the vehicle is to be estimated with submeter accuracy. This problem is made difficult by the paucity of sensor information. Only the following sources are available: an inertial navigation system (which is polled only at 10Hz), a set of encoders (also polled at 10Hz), and a bearing-only sensor (rotation rate of 4Hz) which measures bearing to a set of beacons. Because of the low quality of sensor information, the vehicle localization system can meet the performance requirements only through the use of an accurate process model. The model that was developed is nonlinear and incorporates kinematics, dynamics and slip due to tire deformation. It also contains a large number of process noise terms. This model is extremely cumbersome to work with, but the UF obviates the need to calculate Jacobians, greatly simplifying its use.



(A)

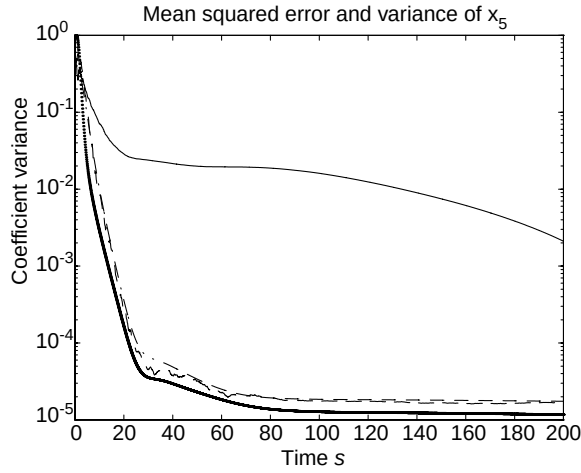


(B)

FIGURE 13.6 The mean squared errors and estimated covariances calculated by an EKF and an unscented filter. (A) Results for x_1 . (B) Results for x_3 . (C) Results for x_5 . In all of the graphs, the solid line is the mean squared error calculated by the EKF, and the dotted line is its estimated covariance. The dashed line is the unscented mean squared error and the dot-dashed line is its estimated covariance. In all diagrams, the EKF estimate is inconsistent but the UT estimate is not.

The model of vehicle motion is developed from the two-dimensional “fundamental bicycle” which is shown in [Figure 13.7](#).³²⁻³⁴ This approximation, which is conventional for vehicle ride and handling analysis, assumes that the vehicle consists of front and rear virtual wheels.* The vehicle body is the line FGR with the front axle affixed at F and the rear axle fixed at R . The center of mass of the vehicle is located at G , a distance a behind the front axle and b in front of the rear axle. The length of the wheel base is $B = a + b$. The wheels can slip laterally with slip angles α_f and α_r , respectively. The control inputs are the steer angle, δ , and angular speed, ω , of the front virtual wheel.

*Each virtual wheel lumps together the kinematic and dynamic properties of the pairs of wheels at the front and rear axles.

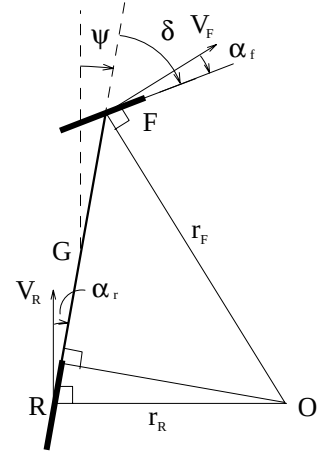


(C)

FIGURE 13.6 (continued).



(A)



(B)

FIGURE 13.7 The actual experimental vehicle and the “fundamental bicycle model” representation used in the design of the vehicle process model. (A) The host vehicle at the test site with sensor suite. (B) Vehicle kinematics.

The filter estimates the position of F , (X_F, Y_F) the orientation of FGR , ψ , and the effective radius of the front wheel, R , (defined as the ratio of vehicle velocity to the rotation rate of the front virtual wheel). The speed of the front wheel is V_F and the path curvature is ρ_F . From the kinematics, the velocity of F is

$$\dot{X}_F = V_F \cos[\psi + \delta - \alpha_f] \quad \rho_F = \left(\sin[\delta - \alpha_f] + \cos[\delta - \alpha_f] \tan \alpha_r \right) / B$$

$$\dot{Y}_F = V_F \sin[\psi + \delta - \alpha_f] \quad \text{where} \quad V_F = R\omega \cos \alpha_f$$

$$\dot{\psi} = \rho_F V_F - \dot{\delta} - \dot{\alpha}_f$$

$$\dot{R} = 0$$

The slip angle (α_f) plays an extremely important role in determining the path taken by the vehicle and a model for determining the slip angle is highly desirable. The slip angle is derived from the properties of the tires. Specifically, tires behave as if they are linear torsional springs. The slip angle on each wheel is proportional to the force that acts on the wheel³³

$$\alpha_f = F_{yf} / C_{\alpha_f} \quad \alpha_r = F_{yr} / C_{\alpha_r} \quad (13.23)$$

C_{α_f} and C_{α_r} are the front and rear wheel lateral stiffness coefficients (which are imprecisely known). The front and rear lateral forces, F_{yf} and F_{yr} , are calculated under the assumption that the vehicle has reached a steady state; at any instant in time the forces are such that the vehicle moves along an arc with constant radius and constant angular speed.³⁵ Resolving moments parallel and perpendicular to OG , and taking moments about G , the following simultaneous nonlinear equations must be solved:

$$\begin{aligned} \rho_G m V_G^2 &= F_{xf} \sin[\delta - \beta] + F_{yf} \cos[\delta - \beta] + F_{yr} \cos \beta, & \beta &= \tan^{-1} \left(\left\{ b \tan[\delta - \alpha_f] - \alpha \tan \alpha_r \right\} / B \right) \\ 0 &= F_{xf} \cos[\delta - \beta] - F_{yf} \sin[\delta - \beta] + F_{yr} \sin \beta, & \rho_G &= \cos \beta \left\{ \tan[\delta - \alpha_f] + \tan \alpha_r \right\} / B \\ 0 &= F_{xf} \sin \delta + a F_{yf} \cos \delta - F_{yr}, & V_G &= V_F \cos[\delta - \alpha_f] \sec \beta \end{aligned}$$

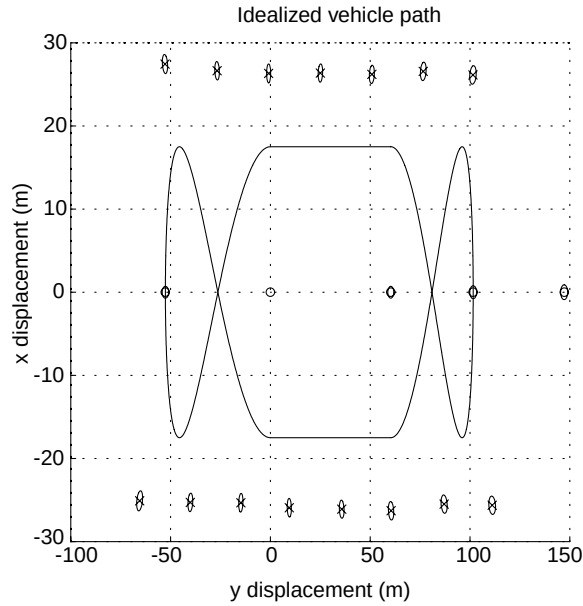
m is the mass of the vehicle, V_G and ρ_G are the speed and path curvature of G , and β is the attitude angle (illustrated in Figure 13.7). These equations are solved using a conventional numerical solver²⁴ to give the tire forces. Through Equation 13.23, these determine the slip angles and, hence, the path of the vehicle. Since C_{α_f} and C_{α_r} must account for modeling errors (such as the inaccuracies of a linear force-slip angle relationship), these were treated as states and their values were estimated.

As this section has shown, a comprehensive vehicle model is extremely complicated. The state space consists of six highly interconnected states. The model is made even more complicated by the fact that it possesses twelve process noise terms. Therefore, 18 terms must be propagated through the nonlinear process model. The observation models are also very complex. (The derivation and debugging of such Jacobians proved to be extremely difficult.) However, the UF greatly simplified the implementation, tuning, and testing of the filter. An example of the performance of the final navigation system is shown in Figure 13.8. Figure 13.8(a) shows a “figure of eight” route that was planned for the vehicle. This path is highly dynamic (with continuous and rapid changes in both vehicle speed and steer angle) and contains a number of well-defined landmarks (which were used to validate the algorithm). There is extremely good agreement between the estimated and the actual paths, and the covariance estimate ($0.25m^2$ in position) exceeds the performance requirements.

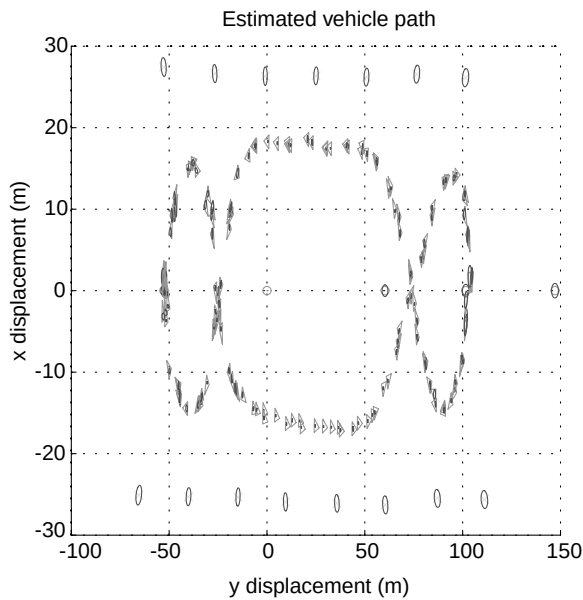
13.8 Multilevel Sensor Fusion

This section discusses how the UT can be used in systems that do not inherently use a mean and covariance description to describe their state. Because the UT can be applied to such systems, it can be used as a consistent framework for multilevel data fusion. The problem of data fusion has been decomposed into a set of hierarchical domains.³⁶ The lowest levels, Level 0 and Level 1 (object refinement), are concerned with quantitative data fusion problems such as the calculation of a target track. Level 2 (situation refinement) and Level 3 (threat refinement) apply various high-level data fusion and pattern recognition algorithms to attempt to glean strategic and tactical information from these tracks.

The difficulty lies in the fundamental differences in the representation and use of information. On the one hand, the low-level tracking filter provides only mean and covariance information. It does not specify an exact kinematic state from which an expert system could attempt to infer a tactical state. On the other hand, an expert system may be able to predict accurately the behavior of a pilot under a range of situations.



(A)



(B)

FIGURE 13.8 The positions of the beacons can be seen in (A) and (B) as the row of ellipses at the top and bottom of the figures.

However, the system does not define a rigorous low-level framework for fusing its predictions with raw sensor information to obtain high-precision estimates suitable for reliable tracking. The practical solution to this problem has been to take the output of standard control and estimation routines, discretize them into a more symbolic form (e.g., “slow” or “fast”), and process them with an expert/fuzzy rule base. The

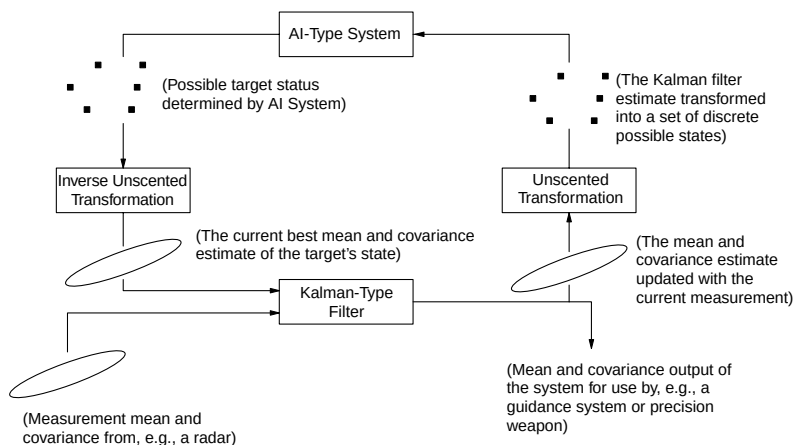


FIGURE 13.9 A possible framework for multilevel information fusion using the unscented transformation.

results of such processing are then converted into forms that can be processed by conventional process technology.

One approach for resolving this problem, illustrated in Figure 13.9, is to combine the different data fusion algorithms together into a single, composite data fusion algorithm that takes noise-corrupted raw sensor data and provides the inferred high-level state. From the perspective of the track estimator, the higher level fusion rules are considered to be arbitrary, nonlinear transformations. From the perspective of the higher level data fusion algorithms, the UT converts the output from the low-level tracker into a set of vectors. Each vector is treated as a possible kinematic state, which is processed by the higher-level fusion algorithms. In other words, the low-level tracking algorithms do not need to understand the concept of higher-level constructs, such as maneuvers, whereas the higher level algorithms do not need to understand or produce probabilistic information.

Consider the problem of tracking an aircraft. The aircraft model consists of two components — a kinematic model, which describes the trajectory of the aircraft for a given set of pilot inputs, and an expert system, which attempts to infer current pilot intentions and predict future pilot inputs. The location of the aircraft is measured using a tracking system, such as a radar.

Some sigma points might imply that the aircraft is making a rapid acceleration, some might indicate a moderate acceleration, and yet others might imply that there is no discernible acceleration. Each of the state vectors produced from the UT can be processed individually by the expert system to predict various possible future states of the aircraft. For some of the state vectors, the expert system will signal air evasive maneuvers and predict the future position of the aircraft accordingly. Other vectors, however, will not signal a change of tactical state and the expert system will predict that the aircraft will maintain its current speed and bearing. The second step of the UT consists of computing the mean and covariance of the set of predicted state vectors from the expert system. This mean and covariance gives the predicted state of the aircraft in a form that can then be fed back to the low-level filter. The important observation to be made is that this mean and covariance reflect the probability that the aircraft will maneuver *even though the expert system did not produce any probabilistic information and the low-level filter knows nothing about maneuvers*.

13.9 Conclusions

This chapter has described some of the important issues arising from the occurrence of nonlinear transformations in practical data fusion applications. Linearization is the most widely used approach for dealing with nonlinearities, but linearized approximations have been shown to yield relatively poor results. In response to this and other deficiencies of linearization, a new technique based on the UT has

been developed for directly applying nonlinear transformations to discrete point distributions having specified statistics (such as mean and covariance). An analysis of the new approach reveals that

- The UT is demonstrably superior to linearization in terms of expected error for all absolutely continuous nonlinear transformations. The UT can be applied with non-differentiable functions in which linearization is not possible.
- The UT avoids the derivation of Jacobian (and Hessian) matrices for linearizing nonlinear kinematic and observation models. This makes it conducive to the creation of efficient, general purpose “black box” code libraries.
- Empirical results for several nonlinear transformations that typify those arising in practical applications clearly demonstrate that linearization yields very poor approximations compared to those of the UT.

Beyond analytic claims of unproved accuracy, the UT offers a black box solution to a wide variety of problems arising in both low- and high-level data fusion applications. In particular, it offers a mechanism for seamlessly integrating the benefits of high-level methodologies — such as artificial intelligence, fuzzy logic and neural networks — with the low-level workhorses of modern engineering practice — such as covariance intersection and the Kalman filter.

Acknowledgments

The authors gratefully acknowledge support from IDAK Industries and the University of Oxford.

References

1. Julier, S.J. and Uhlmann, J.K., A non-divergent estimation algorithm in the presence of unknown correlations, *American Control Conf.*, 4, 2369–2373, 1997.
2. Kalman, R.E., A new approach to linear filtering and prediction problems, *Trans. ASME, J. Basic Engineering*, 82, 34–45, 1960.
3. Kushner, H.J., Dynamical equations for optimum non-linear filtering, *J. Differential Equations*, 3, 179–190, 1967.
4. Jazwinski, A.H., *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970.
5. Maybeck, P.S., *Stochastic Models, Estimation, and Control*, Vol. 2, Academic Press, New York, 1982.
6. Daum, F.E., New exact nonlinear filters, *Bayesian Analysis of Time Series and Dynamic Models*, J.C. Spall, ed., Marcel Dekker, Inc., New York, 199–226, 1988.
7. Gordon, N.J., Salmond, D.J., and Smith, A.F.M., Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEEE Proc.-F*, 140(2), 107–113, 1993.
8. Kouritzin, M.A., On exact filters for continuous signals with discrete observations. Technical report, University of Minnesota, Institute for Mathematics and its Applications, May 1996.
9. Uhlmann, J.K., Algorithms for multiple target tracking, *American Scientist*, 80(2), 128–141, 1992.
10. Sorenson, H.W., Ed. *Kalman Filtering: Theory and Application*, IEEE Press, 1985.
11. Athans, M., Wishner, R.P., and Bertolini, A., Suboptimal state estimation for continuous time nonlinear systems from discrete noisy measurements, *IEEE Trans. Automatic Control*, TAC-13(6), 504–518, 1968.
12. Costa, P.J., Adaptive model architecture and extended Kalman-Bucy filters, *IEEE Trans. Aerospace and Electronic Systems*, AES-30(2), 525–533, 1994.
13. Austin, J.W. and Leondes, C.T., Statistically Linearized Estimation of Reentry Trajectories, *IEEE Trans. Aerospace and Electronic Systems*, AES-17(1), 54–61, 1981.
14. Mehra, R. K. A Comparison of Several Nonlinear Filters for Reentry Vehicle Tracking, *IEEE Trans. Automatic Control*, AC-16(4), 307–319, 1971.
15. Viéville, T. and Sander, P., Using pseudo Kalman filters in the presence of constraints application to sensing behaviours, Technical report, INRIA, April 1992.

16. Dulimov, P.A., Estimation of Ground Parameters for the Control of a Wheeled Vehicle, Master's thesis, The University of Sydney, 1997.
17. Wolfram, S., *The Mathematica Book*, Wolfram Research, 4th edition, 1999.
18. Redfern, D., *Maple V Handbook — Release 4*, 1996.
19. Uhlmann, J.K., Simultaneous map building and localization for real-time applications, technical report, Transfer thesis. University of Oxford, 1994.
20. Julier, S.J. and Uhlmann, J.K. A general method for approximating nonlinear transformations of probability distributions, <http://www.robots.ox.ac.uk/~siju>, August 1994.
21. Kushner, H.J., Approximations to optimal nonlinear filters, *IEEE Trans. Automatic Control*, AC-12(5), 546–556, 1967.
22. Holtzmann, J., On using perturbation analysis to do sensitivity analysis: derivatives vs. differences, *IEEE Conf. Decision and Control*, 2018–2023, 1989.
23. Julier, S.J., Uhlmann, J.K., and Durrant-Whyte, H.F., A new approach for the nonlinear transformation of means and covariances in linear filters, *IEEE Trans. Automatic Control*, 477–482, March 2000.
24. Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery B.P., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 2nd edition, 1992.
25. Julier, S.J. and Uhlmann, J.K., A consistent, debiased method for converting between polar and Cartesian coordinate systems, *Proc. AeroSense: Acquisition, Tracking and Pointing XI*, SPIE, 3086, 110–121, 1997.
26. Julier, S.J., A skewed approach to filtering, *Proc. of AeroSense: The 12th International Symp. Aerospace/Defense Sensing, Simulation, and Controls*, 3373, 54–65. SPIE, 1998. Signal and Data Processing of Small Targets.
27. Lerro, D. and Bar-Shalom, Y.K., Tracking with Debiased Consistent Converted Measurements vs. EKF. *IEEE Trans. Aerospace and Electronics Systems*, AES-29(3), 1015–1022, 1993.
28. Leonard, J., *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic Press, Boston, 1991.
29. Schmidt, S.F., Applications of state space methods to navigation problems, *Advanced Control Systems*, C.T. Leondes, Ed., Academic Press, New York, Vol. 3, 293–340, 1966.
30. Chang, C.B., Whiting, R.H., and Athans, M. On the state and parameter estimation for maneuvering reentry vehicles, *IEEE Trans. Automatic Control*, AC-22, 99–105, 1977.
31. Julier, S.J., *Comprehensive Process Models for High-Speed Navigation*, Ph.D. thesis, University of Oxford, 1997.
32. Ellis, J.R., *Vehicle Handling Dynamics*, Mechanical Engineering Publications, London, 1994.
33. Wong, J.Y., *Theory of Ground Vehicles*, John Wiley & Sons, New York, 2nd edition, 1993.
34. Dixon, J.C., *Tyres, Suspension and Handling*, Cambridge University Press, Cambridge, U.K., 1991.
35. Julier, S.J. and Durrant-Whyte, H.F., Navigation and parameter estimation of high speed road vehicles, *Robotics and Automation Conf.*, 101–105, 1995.
36. Klein, L., *Sensor and Data Fusion Concepts and Applications*, SPIE, 2nd edition, 1965.