14.2 UNSCENTED TRANSFORMATIONS

The problem with nonlinear systems is that it is difficult to transform a probability density function through a general nonlinear function. In the previous section, we were able to obtain exact nonlinear transformations of the mean and covariance, but only for a simple two-dimensional transformation. The extended Kalman filter works on the principle that a linearized transformation of means and covariances is approximately equal to the true nonlinear transformation, but we saw in the previous section that the approximation could be unsatisfactory.

An unscented transformation is based on two fundamental principles. First, it is easy to perform a nonlinear transformation on a single point (rather than an entire pdf). Second, it is not too hard to find a set of individual points in state space whose sample pdf approximates the true pdf of a state vector.

Taking these two ideas together, suppose that we know the mean $\bar{x}$ and covariance P of a vector x. We then find a set of deterministic vectors called sigma points whose ensemble mean and covariance are equal to $\bar{x}$ and P. We next apply our known nonlinear function $y = h(x)$ to each deterministic vector to obtain transformed vectors. The ensemble mean and covariance of the transformed vectors will give a good estimate of the true mean and covariance of y. This is the key to the unscented transformation.

As an example, suppose that x is an $n \times 1$ vector that is transformed by a nonlinear function $y = h(x)$. Choose $2n$ sigma points $x^{(i)}$ as follows:

$$
\begin{aligned}
x^{(i)} &= \bar{x} + \tilde{x}^{(i)} \qquad i = 1, \cdots, 2n \\
\tilde{x}^{(i)} &= \left(\sqrt{nP}\right)_i^T \qquad i = 1, \cdots, n \\
\tilde{x}^{(n+i)} &= -\left(\sqrt{nP}\right)_i^T \qquad i = 1, \cdots, n
\end{aligned} \tag{14.29}
$$

where $\sqrt{nP}$ is the matrix square root of nP such that $(\sqrt{nP})^T\sqrt{nP} = nP$, and $(\sqrt{nP})_i$ is the ith row of $\sqrt{nP}$.[1] In the next couple of subsections, we will see how the ensemble mean of the above sigma points can be used to approximate the mean and covariance of a nonlinearly transformed vector.

14.2.1 Mean approximation

Suppose that we have a vector x with a known mean $\bar{x}$ and covariance P, a nonlinear function $y = h(x)$, and we want to approximate the mean of y. We propose transforming each individual sigma point of Equation (14.29) using the nonlinear function $h(\cdot)$, and then taking the weighted sum of the transformed sigma points to approximate the mean of y. The transformed sigma points are computed as follows:

$$
y^{(i)} = h\left(x^{(i)}\right) \qquad i = 1, \cdots, 2n \tag{14.30}
$$

[1] MATLAB's Cholesky factorization routine CHOL can be used to find a matrix square root. See Section 6.3.1, but note the slight difference between the matrix square root definition used in that section and here.

The true mean of y is denoted as $\bar{y}$. The approximated mean of y is denoted as $\bar{y}_u$ and is computed as follows:

$$\bar{y}_u = \sum_{i=1}^{2n} W^{(i)} y^{(i)} \tag{14.31}$$

The weighting coefficients $W^{(i)}$ are defined as follows:

$$W^{(i)} = \frac{1}{2n} \quad i = 1, \cdots, 2n \tag{14.32}$$

Equation (14.31) can therefore be written as

$$\bar{y}_u = \frac{1}{2n} \sum_{i=1}^{2n} y^{(i)} \tag{14.33}$$

Now let's compute the value of $\bar{y}_u$ to see how well it matches the true mean of y. To do this we first use Equation (1.89) to expand each $y^{(i)}$ in Equation (14.33) in a Taylor series around $\bar{x}$. This results in

$$\begin{aligned} \bar{y}_u &= \frac{1}{2n} \sum_{i=1}^{2n} \left(h(\bar{x}) + D_{\tilde{x}^{(i)}} h + \frac{1}{2!} D^2_{\tilde{x}^{(i)}} h + \cdots \right) \\ &= h(\bar{x}) + \frac{1}{2n} \sum_{i=1}^{2n} \left(D_{\tilde{x}^{(i)}} h + \frac{1}{2!} D^2_{\tilde{x}^{(i)}} h + \cdots \right) \end{aligned} \tag{14.34}$$

Now notice that for any integer $k \geq 0$ we have

$$\begin{aligned} \sum_{j=1}^{2n} D^{2k+1}_{\tilde{x}^{(j)}} h &= \sum_{j=1}^{2n} \left[\left(\sum_{i=1}^{n} \tilde{x}_i^{(j)} \frac{\partial}{\partial x_i} \right)^{2k+1} h(x) \Big|_{x=\bar{x}} \right] \\ &= \sum_{j=1}^{2n} \left[\sum_{i=1}^{n} \left(\tilde{x}_i^{(j)} \right)^{2k+1} \frac{\partial^{2k+1}}{\partial x_i^{2k+1}} h(x) \Big|_{x=\bar{x}} \right] \\ &= \sum_{i=1}^{n} \left[\sum_{j=1}^{2n} \left(\tilde{x}_i^{(j)} \right)^{2k+1} \frac{\partial^{2k+1}}{\partial x_i^{2k+1}} h(x) \Big|_{x=\bar{x}} \right] \\ &= 0 \end{aligned} \tag{14.35}$$

because from Equation (14.29) $\tilde{x}^{(j)} = -\tilde{x}^{(n+j)}$ $(j = 1, \cdots, n)$. Therefore, all of the odd terms in Equation (14.34) evaluate to zero and we have

$$\begin{aligned} \bar{y}_u &= h(\bar{x}) + \frac{1}{2n} \sum_{i=1}^{2n} \left(\frac{1}{2!} D^2_{\tilde{x}^{(i)}} h + \frac{1}{4!} D^4_{\tilde{x}^{(i)}} h + \cdots \right) \\ &= h(\bar{x}) + \frac{1}{2n} \sum_{i=1}^{2n} \frac{1}{2!} D^2_{\tilde{x}^{(i)}} h + \\ &\quad \frac{1}{2n} \sum_{i=1}^{2n} \left(\frac{1}{4!} D^4_{\tilde{x}^{(i)}} h + \frac{1}{6!} D^6_{\tilde{x}^{(i)}} h + \cdots \right) \end{aligned} \tag{14.36}$$

Now look at the second term on the right side of the above equation:

$$
\begin{aligned}
\frac{1}{2n}\sum_{i=1}^{2n}\frac{1}{2!}D_{\tilde{x}^{(i)}}^{2}h &= \frac{1}{2n}\sum_{k=1}^{2n}\frac{1}{2!}\left(\sum_{i=1}^{n}\tilde{x}_i^{(k)}\frac{\partial}{\partial x_i}\right)^2 h(x)\bigg|_{x=\bar{x}} \\
&= \frac{1}{4n}\sum_{k=1}^{2n}\sum_{i,j=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i \partial x_j}h(x)\bigg|_{x=\bar{x}} \\
&= \frac{1}{4n}\sum_{i,j=1}^{n}\sum_{k=1}^{2n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i \partial x_j}h(x)\bigg|_{x=\bar{x}} \\
&= \frac{1}{2n}\sum_{i,j=1}^{n}\sum_{k=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i \partial x_j}h(x)\bigg|_{x=\bar{x}}
\end{aligned}
\tag{14.37}
$$

where we have again used the fact from Equation (14.29) that $\tilde{x}^{(k)} = -\tilde{x}^{(k+n)}$ $(k = 1, \cdots, n)$. Substitute for $\tilde{x}_i^{(k)}$ and $\tilde{x}_j^{(k)}$ from Equation (14.29) in the above equation to obtain

$$
\begin{aligned}
\frac{1}{2n}\sum_{i,j=1}^{n}\sum_{k=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2 h(x)}{\partial x_i \partial x_j}\bigg|_{x=\bar{x}} &= \frac{1}{2n}\sum_{i,j=1}^{n}\sum_{k=1}^{n}\left(\sqrt{nP}\right)_{ki}\left(\sqrt{nP}\right)_{kj}\frac{\partial^2 h(x)}{\partial x_i \partial x_j}\bigg|_{x=\bar{x}} \\
&= \frac{1}{2n}\sum_{i,j=1}^{n} nP_{ij}\frac{\partial^2 h(x)}{\partial x_i \partial x_j}\bigg|_{x=\bar{x}} \\
&= \frac{1}{2}\sum_{i,j=1}^{n} P_{ij}\frac{\partial^2 h(x)}{\partial x_i \partial x_j}\bigg|_{x=\bar{x}}
\end{aligned}
\tag{14.38}
$$

Equation (14.36) can therefore be written as

$$
\begin{aligned}
\bar{y}_u &= h(\bar{x}) + \frac{1}{2}\sum_{i,j=1}^{n} P_{ij}\frac{\partial^2 h}{\partial x_i \partial x_j}\bigg|_{x=\bar{x}} + \\
&\quad \frac{1}{2n}\sum_{i=1}^{2n}\left(\frac{1}{4!}D_{\tilde{x}^{(i)}}^{4}h + \frac{1}{6!}D_{\tilde{x}^{(i)}}^{6}h + \cdots\right)
\end{aligned}
\tag{14.39}
$$

Now recall that the true mean of y is given by Equation (14.15) as

$$
\bar{y} = h(\bar{x}) + \frac{1}{2!}E\left[D_{\tilde{x}}^{2}h\right] + \frac{1}{4!}E\left[D_{\tilde{x}}^{4}h\right] + \cdots \tag{14.40}
$$

Look at the second term on the right side of the above equation. It can be written as follows:

$$
\begin{aligned}
\frac{1}{2!}E\left[D_{\tilde{x}}^{2}h\right] &= \frac{1}{2!}E\left[\left(\sum_{i=1}^{n}\tilde{x}_i\frac{\partial}{\partial x_i}\right)^2 h(x)\bigg|_{x=\bar{x}}\right] \\
&= \frac{1}{2!}E\left[\sum_{i,j=1}^{n}\tilde{x}_i\tilde{x}_j\frac{\partial^2 h}{\partial x_i \partial x_j}\bigg|_{x=\bar{x}}\right]
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{2!}\sum_{i,j=1}^{n} E(\tilde{x}_i\tilde{x}_j)\,\frac{\partial^2 h}{\partial x_i \partial x_j}\bigg|_{x=\bar{x}} \\
&= \frac{1}{2!}\sum_{i,j=1}^{n} P_{ij}\,\frac{\partial^2 h}{\partial x_i \partial x_j}\bigg|_{x=\bar{x}} \qquad (14.41)
\end{aligned}
$$

We therefore see that $\bar{y}$ can be written from Equation (14.40) as

$$
\begin{aligned}
\bar{y} &= h(\bar{x}) + \frac{1}{2}\sum_{i,j=1}^{n} P_{ij}\,\frac{\partial^2 h}{\partial x_i \partial x_j}\bigg|_{x=\bar{x}} + \\
&\quad \frac{1}{4!}E\left[D_{\tilde{x}}^4 h\right] + \frac{1}{6!}E\left[D_{\tilde{x}}^6 h\right] + \cdots \qquad (14.42)
\end{aligned}
$$

Comparing this with Equation (14.39) we see that $\bar{y}_u$ (the approximated mean of y) matches the true mean of y correctly up to the third order, whereas linearization only matches the true mean of y up to the first order (see Section 14.1.1). If we compute $\bar{y}_u$ using Equations (14.29), (14.30), and (14.33), then the value of $\bar{y}_u$ will match the true mean of y up to the third order. The biggest difficulty with this algorithm is the matrix square root that is required in Equation (14.29). But the unscented transformation has the computational advantage that the linearization matrix H does not need to be computed. Of course, the greatest advantage of the unscented transformation (relative to linearization) is the increased accuracy of the mean transformation.

14.2.2 Covariance approximation

Now suppose that we want to approximate the covariance of the nonlinearly transformed vector x. That is, we have an n-element vector x with known mean $\bar{x}$ and covariance P, and we have a known nonlinear function $y = h(x)$. We want to estimate the covariance of y. We will denote the estimate as P_u, and we propose using the following equation:

$$
\begin{aligned}
P_u &= \sum_{i=1}^{2n} W^{(i)}(y^{(i)} - y_u)(y^{(i)} - y_u)^T \\
&= \frac{1}{2n}\sum_{i=1}^{2n}(y^{(i)} - y_u)(y^{(i)} - y_u)^T \qquad (14.43)
\end{aligned}
$$

where the $y^{(i)}$ vectors are the transformed sigma points that were computed in Equation (14.30), and the $W^{(i)}$ weighting coefficients are the same as those given in Equation (14.32). Expanding this approximation using Equations (1.89) and (14.36) gives the following:

$$
P_u = \frac{1}{2n}\sum_{i=1}^{2n}\left[h(x^{(i)}) - y_u\right]\left[h(x^{(i)}) - y_u\right]^T \qquad (14.44)
$$

$$\begin{aligned}
&= \frac{1}{2n}\sum_{i=1}^{2n}\Bigg[h(\bar{x}) + D_{\tilde{x}^{(i)}}h + \frac{1}{2}D^2_{\tilde{x}^{(i)}}h + \frac{1}{3!}D^3_{\tilde{x}^{(i)}}h + \cdots \\
&\quad -h(\bar{x}) - \frac{1}{2n}\sum_{j=1}^{2n}\left(\frac{1}{2}D^2_{\tilde{x}^{(j)}}h + \frac{1}{4!}D^4_{\tilde{x}^{(j)}}h + \cdots\right)\Bigg]\Big[\cdots\Big]^T
\end{aligned} \tag{14.45}$$

Multiplying this equation out gives

$$\begin{aligned}
P_u &= \frac{1}{2n}\sum_{i=1}^{2n}\Bigg\{(D_{\tilde{x}^{(i)}}h)(\cdots)^T + \underbrace{\left[\left(\frac{1}{2}D_{\tilde{x}^{(i)}}h\right)(D^2_{\tilde{x}^{(i)}}h)^T\right]}_{0} + \underbrace{\Big[\cdots\Big]^T}_{0} + \\
&\quad \frac{1}{4!}(D^2_{\tilde{x}^{(i)}}h)(\cdots)^T - \underbrace{\left[D_{\tilde{x}^{(i)}}h\left(\frac{1}{2n}\sum_j \frac{1}{2}D^2_{\tilde{x}^{(i)}}h\right)^T\right]}_{0} - \underbrace{\Big[\cdots\Big]^T}_{0} + \\
&\quad \frac{1}{4n^2}\left(\sum_j D^2_{\tilde{x}^{(j)}}h\right)\Big(\cdots\Big)^T - \left[\frac{1}{4n}D^2_{\tilde{x}^{(i)}}h\left(\sum_j D^2_{\tilde{x}^{(j)}}h\right)^T\right] - \Big[\cdots\Big]^T + \\
&\quad \left[D_{\tilde{x}^{(i)}}h\left(\frac{1}{3!}D^3_{\tilde{x}^{(j)}}h\right)^T\right] + \Big[\cdots\Big]^T + \cdots\Bigg\}
\end{aligned} \tag{14.46}$$

Some of the terms in the above equation are zero as noted above because $\tilde{x}^{(i)} = -\tilde{x}^{(i+n)}$ for $i = 1, \cdots, n$. So the covariance approximation can be written as

$$P_u = \frac{1}{2n}\sum_{i=1}^{2n}(D_{\tilde{x}^{(i)}}h)(\cdots)^T + \text{HOT} \tag{14.47}$$

where HOT means higher-order terms (i.e., terms to the fourth power and higher). Expanding this equation for P_u while neglecting the higher order terms gives

$$P_u = \frac{1}{2n}\sum_{i=1}^{2n}\sum_{j,k=1}^{n}\left(\tilde{x}_j^{(i)}\frac{\partial h(\bar{x})}{\partial x_j}\right)\left(\tilde{x}_k^{(i)}\frac{\partial h(\bar{x})}{\partial x_k}\right)^T \tag{14.48}$$

Now recall that $\tilde{x}_j^{(i)} = -\tilde{x}_j^{(i+n)}$ and $\tilde{x}_k^{(i)} = -\tilde{x}_k^{(i+n)}$ for $i = 1, \cdots, n$. Therefore, the covariance approximation becomes

$$\begin{aligned}
P_u &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j,k=1}^{n}\left(\tilde{x}_j^{(i)}\frac{\partial h(\bar{x})}{\partial x_j}\right)\left(\tilde{x}_k^{(i)}\frac{\partial h(\bar{x})}{\partial x_k}\right)^T \\
&= \sum_{j,k=1}^{n}P_{jk}\frac{\partial h(\bar{x})}{\partial x_j}\left(\frac{\partial h(\bar{x})}{\partial x_k}\right)^T \\
&= HPH^T
\end{aligned} \tag{14.49}$$

where the last equality comes from Equation (14.22). Comparing this equation for P_u with the true covariance of y from Equation (14.23), we see that Equation (14.43) approximates the true covariance of y up to the third order (i.e., only terms to the fourth and higher powers are incorrect). This is the same approximation order as the linearization method, as seen on page 439. However, we would intuitively expect the magnitude of the error of the unscented approximation in Equation (14.43) to be smaller than the linear approximation HPH^T, because the unscented approximation at least contains correctly signed terms to the fourth power and higher, whereas the linear approximation does not contain any terms other than HPH^T.

The unscented transformation can be summarized as follows.

The unscented transformation

1. We begin with an n-element vector x with known mean $\bar{x}$ and covariance P. Given a known nonlinear transformation $y = h(x)$, we want to estimate the mean and covariance of y, denoted as $\bar{y}_u$ and P_u.

2. Form $2n$ sigma point vectors $x^{(i)}$ as follows:

$$\begin{aligned} x^{(i)} &= \bar{x} + \tilde{x}^{(i)} \quad i = 1, \cdots, 2n \\ \tilde{x}^{(i)} &= \left(\sqrt{nP}\right)_i^T \quad i = 1, \cdots, n \\ \tilde{x}^{(n+i)} &= -\left(\sqrt{nP}\right)_i^T \quad i = 1, \cdots, n \end{aligned} \tag{14.50}$$

where $\sqrt{nP}$ is the matrix square root of nP such that $(\sqrt{nP})^T\sqrt{nP} = nP$, and $(\sqrt{nP})_i$ is the ith row of $\sqrt{nP}$.

3. Transform the sigma points as follows:

$$y^{(i)} = h(x^{(i)}) \quad i = 1, \cdots, 2n \tag{14.51}$$

4. Approximate the mean and covariance of y as follows:

$$\begin{aligned} \bar{y}_u &= \frac{1}{2n}\sum_{i=1}^{2n} y^{(i)} \\ P_u &= \frac{1}{2n}\sum_{i=1}^{2n}\left(y^{(i)} - y_u\right)\left(y^{(i)} - y_u\right)^T \end{aligned} \tag{14.52}$$

■ **EXAMPLE 14.1**

To illustrate the unscented transformation, consider the nonlinear transformation shown in Equation (14.1). Since there are two independent variables (r and θ), we have $n = 2$. The covariance of P is given as $P = \text{diag}(\sigma_r^2, \sigma_\theta^2)$. Equation (14.32) shows that $W^{(i)} = 1/4$ for $i = 1, 2, 3, 4$. Equation (14.29) shows that the sigma points are determined as

$$\begin{aligned} x^{(1)} &= \bar{x} + \left(\sqrt{nP}\right)_1^T \\ &= \begin{bmatrix} 1 + \sigma_r\sqrt{2} \\ \pi/2 \end{bmatrix} \end{aligned}$$

$$
\begin{aligned}
x^{(2)} &= \bar{x} + \left(\sqrt{nP}\right)_2^T \\
&= \begin{bmatrix} 1 \\ \pi/2 + \sigma_\theta\sqrt{2} \end{bmatrix} \\
x^{(3)} &= \bar{x} - \left(\sqrt{nP}\right)_1^T \\
&= \begin{bmatrix} 1 - \sigma_r\sqrt{2} \\ \pi/2 \end{bmatrix} \\
x^{(4)} &= \bar{x} - \left(\sqrt{nP}\right)_2^T \\
&= \begin{bmatrix} 1 \\ \pi/2 - \sigma_\theta\sqrt{2} \end{bmatrix}
\end{aligned} \tag{14.53}
$$

Computing the nonlinearly transformed sigma points $y^{(i)} = h(x^{(i)})$ gives

$$
\begin{aligned}
y^{(1)} &= \begin{bmatrix} x_1^{(1)} \cos x_2^{(1)} \\ x_1^{(1)} \sin x_2^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 + \sigma_r\sqrt{2} \end{bmatrix} \\
y^{(2)} &= \begin{bmatrix} x_1^{(2)} \cos x_2^{(2)} \\ x_1^{(2)} \sin x_2^{(2)} \end{bmatrix} = \begin{bmatrix} \cos(\pi/2 + \sigma_\theta\sqrt{2}) \\ \sin(\pi/2 + \sigma_\theta\sqrt{2}) \end{bmatrix} \\
y^{(3)} &= \begin{bmatrix} x_1^{(3)} \cos x_2^{(3)} \\ x_1^{(3)} \sin x_2^{(3)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 - \sigma_r\sqrt{2} \end{bmatrix} \\
y^{(4)} &= \begin{bmatrix} x_1^{(4)} \cos x_2^{(4)} \\ x_1^{(4)} \sin x_2^{(4)} \end{bmatrix} = \begin{bmatrix} \cos(\pi/2 - \sigma_\theta\sqrt{2}) \\ \sin(\pi/2 - \sigma_\theta\sqrt{2}) \end{bmatrix}
\end{aligned} \tag{14.54}
$$

Now we can compute the unscented approximation of the mean and covariance of $y = h(x)$ as

$$
\begin{aligned}
\bar{y}_u &= \sum_{i=1}^{4} W^{(i)} y^{(i)} \\
P_u &= \sum_{i=1}^{4} W^{(i)} \left(y^{(i)} - y_u\right) \left(y^{(i)} - y_u\right)^T
\end{aligned} \tag{14.55}
$$

The results of these transformations are shown in Figure 14.3. This shows the improved accuracy of mean and covariance estimation when unscented transformations are used instead of linear approximations. The true mean and the approximate unscented mean are so close that they are plotted on top of each other. The true mean and the approximate unscented mean are both equal to $(0, 0.9797)$ to four significant digits.

$\nabla\nabla\nabla$

14.3 UNSCENTED KALMAN FILTERING

The unscented transformation developed in the previous section can be generalized to give the unscented Kalman filter. After all, the Kalman filter algorithm attempts

We use rectangular integration with a step size of 1 msec to simulate the system, the extended Kalman filter, and the unscented Kalman filter for 30 seconds. Figure 14.4 shows the altitude and velocity of the falling body. For the first few seconds, the velocity is constant. But then the air density increases and drag slows the falling object. Toward the end of the simulation, the object has reached a constant terminal velocity as the acceleration due to gravity is canceled by drag.

Figure 14.5 shows typical EKF and UKF estimation-error magnitudes for this system. It is seen that the altitude and velocity estimates both spike around 10 seconds, at which point the altitude of the measuring device and the falling body are about the same, so the measurement gives less information about the body's altitude and velocity. It is seen from the figure that the UKF consistently gives estimates that are one or two orders of magnitude better than the EKF.

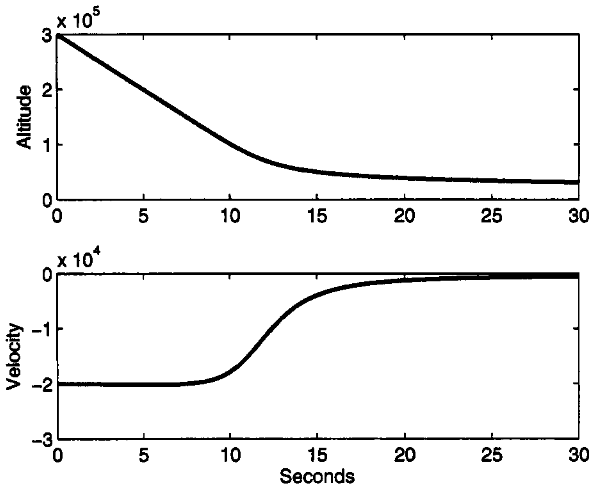

Figure 14.4 Altitude and velocity of a falling body for Example 14.2.

$\nabla\nabla\nabla$

14.4 OTHER UNSCENTED TRANSFORMATIONS

The unscented transformation discussed in the previous section is not the only one that exists. In this section, we discuss several other possible transformations. These other transformations can be used if we have some information about the statistics of the noise, or if we are interested in computational savings.

14.4.1 General unscented transformations

We have seen that an accurate mean and covariance approximation for a nonlinear transformation $y = h(x)$ can be obtained by choosing $2n$ sigma points (where n is the dimension of x) as given in Equation (14.29), and approximating the mean and

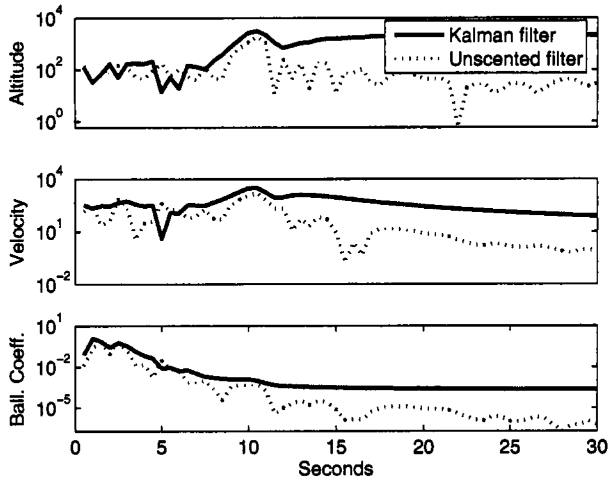

Figure 14.5 Kalman filter and unscented filter estimation-error magnitudes of the altitude, velocity, and ballistic coefficient of a falling body for Example 14.2.

covariance as given in Equations (14.33) and (14.43). However, it can be shown that the same order of mean and covariance estimation accuracy can be obtained by choosing $(2n+1)$ sigma points $x^{(i)}$ as follows:

$$\begin{aligned}
x^{(0)} &= \bar{x} \\
x^{(i)} &= \bar{x} + \tilde{x}^{(i)} \qquad i = 1, \cdots, 2n \\
\tilde{x}^{(i)} &= \left(\sqrt{(n+\kappa)P}\right)_i^T \qquad i = 1, \cdots, n \\
\tilde{x}^{(n+i)} &= -\left(\sqrt{(n+\kappa)P}\right)_i^T \qquad i = 1, \cdots, n
\end{aligned} \tag{14.74}$$

The $(2n+1)$ weighting coefficients are given as

$$\begin{aligned}
W^{(0)} &= \frac{\kappa}{n+\kappa} \\
W^{(i)} &= \frac{1}{2(n+\kappa)} \qquad i = 1, \cdots, 2n
\end{aligned} \tag{14.75}$$

The unscented mean and covariance approximations are computed as

$$\begin{aligned}
y^{(i)} &= h\left(x^{(i)}\right) \\
\bar{y}_u &= \sum_{i=0}^{2n} W^{(i)} y^{(i)} \\
P_u &= \sum_{i=0}^{2n} W^{(i)} \left(y^{(i)} - y_u\right)\left(y^{(i)} - y_u\right)^T
\end{aligned} \tag{14.76}$$

It can be seen that if $\kappa = 0$ then these definitions reduce to the quantities given in Section 14.2. Any κ value can be used [as long as $(n+\kappa) \neq 0$] and will give a mean and covariance estimation accuracy with the same order of accuracy as derived in Section 14.2. However, κ can be used to reduce the higher-order errors of the mean and covariance approximation. For example, if x is Gaussian then $\kappa = 3 - n$ will minimize some of the errors in the fourth-order terms in the mean and covariance approximation [Jul00, Jul04].

14.4.2 The simplex unscented transformation

If computational effort is a primary consideration, then a minimum number of sigma points can be chosen to give the order of estimation accuracy derived in the previous section. It can be shown [Jul02a, Jul04] that if x has n elements then the minimum number of sigma points that gives the order of estimation accuracy of the previous section is equal to $(n + 1)$. These sigma points are called simplex sigma points. The following algorithm results in $(n + 2)$ sigma points, but the number can be reduced to $(n + 1)$ by choosing one of the weights to be zero. The simplex sigma-point algorithm can be summarized as follows.

The simplex sigma-point algorithm

1. Choose the weight $W^{(0)} \in [0, 1)$. The choice of $W^{(0)}$ affects only the fourth and higher order moments of the set of sigma points [Jul00, Jul02a].

2. Choose the rest of the weights as follows:

$$W^{(i)} = \begin{cases} 2^{-n}(1 - W^{(0)}) & i = 1, 2 \\ 2^{i-2}W^{(1)} & i = 3, \cdots, n+1 \end{cases} \tag{14.77}$$

3. Initialize the following one-element vectors:

$$\begin{aligned} \sigma_0^{(1)} &= 0 \\ \sigma_1^{(1)} &= \frac{-1}{\sqrt{2W^{(1)}}} \\ \sigma_2^{(1)} &= \frac{1}{\sqrt{2W^{(1)}}} \end{aligned} \tag{14.78}$$

4. Recursively expand the σ vectors by performing the following steps for $j = 2, \cdots, n$:

$$\sigma_i^{(j)} = \begin{cases} \begin{bmatrix} \sigma_0^{(j-1)} \\ 0 \end{bmatrix} & i = 0 \\ \begin{bmatrix} \sigma_i^{(j-1)} \\ \frac{-1}{\sqrt{2W^{(j+1)}}} \end{bmatrix} & i = 1, \cdots, j \\ \begin{bmatrix} 0_{j-1} \\ \frac{j}{\sqrt{2W^{(j+1)}}} \end{bmatrix} & i = j+1 \end{cases} \tag{14.79}$$

where 0_j is the column vector containing j zeros.

5. After the above recursion is complete we have the n-element vectors $\sigma_i^{(n)}$ ($i = 0, \cdots, n+1$). We modify the unscented transformation of Equation (14.29) and obtain the sigma points for the unscented transformation as follows:

$$x^{(i)} = \bar{x} + \sqrt{P}\sigma_i^{(n)} \qquad (i = 0, \cdots, n+1) \tag{14.80}$$

We actually have $(n+2)$ sigma points instead of the $(n+1)$ sigma points as we claimed, but if we choose $W^{(0)} = 0$ then the $x^{(0)}$ sigma point can be ignored in the ensuing unscented transformation. The unscented Kalman filter algorithm in Section 14.3 is then modified in the obvious way based on this minimal set of sigma points.

The problem with the simplex UKF is that the ratio of $W^{(n)}$ to $W^{(1)}$ is equal to 2^{n-2}, where n is the dimension of the state vector x. As the dimension of the state increases, this ratio increases and can quickly cause numerical problems. The only reason for using the simplex UKF is the computational savings, and computational savings is an issue only for problems of high dimension (in general). This makes the simplex UKF of limited utility and leads to the spherical unscented transformation in the following section.

14.4.3 The spherical unscented transformation

The unscented transformation discussed in Section 14.2 is numerically stable. However, it requires $2n$ sigma points and may be too computationally expensive for some applications. The simplex unscented transformation discussed in Section 14.4.2 is the cheapest computational unscented transformation but loses numerical stability for problems with a moderately large number of dimensions. The spherical unscented transformation was developed with the goal of rearranging the sigma points of the simplex algorithm in order to obtain better numerical stability [Jul03, Jul04]. The spherical sigma points are chosen with the following algorithm.

The spherical sigma-point algorithm

1. Choose the weight $W^{(0)} \in [0, 1)$. The choice of $W^{(0)}$ affects only the fourth- and higher-order moments of the set of sigma points [Jul00, Jul02a].

2. Choose the rest of the weights as follows:

$$W^{(i)} = \frac{1 - W^{(0)}}{n+1} \qquad i = 1, \cdots, n+1 \tag{14.81}$$

Note that (in contrast to the simplex unscented transformation) all of the weights are identical except for $W^{(0)}$.

3. Initialize the following one-element vectors:

$$\begin{aligned} \sigma_0^{(1)} &= 0 \\ \sigma_1^{(1)} &= \frac{-1}{\sqrt{2W^{(1)}}} \\ \sigma_2^{(1)} &= \frac{1}{\sqrt{2W^{(1)}}} \end{aligned} \tag{14.82}$$

4. Recursively expand the σ vectors by performing the following steps for $j = 2, \cdots, n$:

$$\sigma_i^{(j)} = \begin{cases} \begin{bmatrix} \sigma_0^{(j-1)} \\ 0 \end{bmatrix} & i = 0 \\ \begin{bmatrix} \sigma_i^{(j-1)} \\ \frac{-1}{\sqrt{j(j+1)W^{(1)}}} \end{bmatrix} & i = 1, \cdots, j \\ \begin{bmatrix} 0_{j-1} \\ \frac{j}{\sqrt{j(j+1)W^{(1)}}} \end{bmatrix} & i = j+1 \end{cases} \tag{14.83}$$

where 0_j is the column vector containing j zeros.

5. After the above recursion is complete, we have the n-element vectors $\sigma_i^{(n)}$ ($i = 0, \cdots, n+1$). As with the simplex sigma points, we actually have $(n+2)$ sigma points above, but if we choose $W^{(0)} = 0$ then the $x^{(0)}$ sigma point can be ignored in the ensuing unscented transformation. We modify the unscented transformation of Equation (14.29) and obtain the sigma points for the unscented transformation as follows:

$$x^{(i)} = \bar{x} + \sqrt{P}\sigma_i^{(n)} \qquad (i = 0, \cdots, n+1) \tag{14.84}$$

The unscented Kalman filter algorithm in Section 14.3 is then modified in the obvious way based on this set of sigma points.

The ratio of the largest element of $\sigma_i^{(n)}$ to the smallest element is

$$\frac{n}{\sqrt{n(n+1)W^{(1)}}} \Bigg/ \frac{1}{\sqrt{n(n+1)W^{(1)}}} = n \tag{14.85}$$

so numerical problems should not be an issue for the spherical unscented transformation.

■ EXAMPLE 14.3

Here we consider the falling-body system described in Example 14.2. The initial conditions of the system and the estimator are given as

$$\begin{aligned} x_0 &= \begin{bmatrix} 300,000 & -20,000 & 1/1000 \end{bmatrix}^T \\ \hat{x}_0^+ &= \begin{bmatrix} 303,000 & -20,200 & 1/1010 \end{bmatrix}^T \\ P_0^+ &= \begin{bmatrix} 30,000 & 0 & 0 \\ 0 & 2,000 & 0 \\ 0 & 0 & 1/10,000 \end{bmatrix} \end{aligned} \tag{14.86}$$

We ran 100 Monte Carlo simulations, each with a 60 s simulation time. The average RMS estimation errors of the EKF, standard UKF (six sigma points), simplex UKF (four sigma points since we chose $W^{(0)} = 0$), and spherical UKF (four sigma points since we chose $W^{(0)} = 0$) are given in Table 14.1. The simplex UKF performs best for altitude estimation, with the standard UKF not far behind. The standard UKF performs best for velocity estimation, and the spherical UKF performs best for ballistic coefficient estimation. The EKF is generally the worst performing of the four state estimators.

Table 14.1 Example 14.3 estimation errors for the extended Kalman filter, the standard unscented Kalman filter with $2n$ sigma points, and the spherical unscented Kalman filter with $(n+1)$ sigma points. The standard UKF generally performs best. The spherical UKF performance and computational effort lie between those of the EKF and the standard UKF.

	Altitude	Velocity	Ballistic Coefficient Reciprocal
EKF	615	173	11.6
UKF	460	112	7.5
Simplex UKF	449	266	80.8
Spherical UKF	578	142	0.4

$\nabla\nabla\nabla$

14.5 SUMMARY

The unscented filter can give greatly improved estimation performance (compared with the extended Kalman filter) for nonlinear systems. In addition, the EKF requires the computation of Jacobians (partial derivative matrices), and the UKF does not use Jacobians. For systems with analytic process and measurement equations (such as Example 14.2), it is easy to compute Jacobians. But some systems are not given in analytical form and it is numerically difficult to compute Jacobians.

The UKF was first published in 1995 [Jul95] and since then has been expounded upon in many publications.[2] Although the UKF is a relatively recent development, it is rapidly finding applications in such areas as aircraft engine health estimation [Dew03], aircraft model estimation [Cam01], neural network training [Wan01], financial forecasting [Wan01], and motor state estimation [Aki03]. In addition, just as in the Kalman filter, the UKF can be implemented in a square root form to effectively increase numerical precision [Van01, Wan01]. Note that a filter based on polynomial approximations of nonlinear functions is presented in [Nor00], and it seems that the UKF is a special case of this filter.

There is a lot of room for development in the area of unscented filtering. A glance through this book's table of contents shows many specialized topics that have been applied to Kalman and H_∞ filtering, revealing a rich source of research topics for unscented filtering. These include UKF stability properties, constrained unscented filtering, unscented smoothing, reduced-order unscented filtering, robust unscented filtering, unscented filtering with delayed measurements, hybrid unscented/H_∞ filtering, and others.

[2]It is interesting to note that the first journal publication of the UKF was submitted for publication in 1994, but did not appear in print until 2000 [Jul00]. Alternative technologies that are highly different than existing approaches tend to meet with resistance, but persistence (if accompanied by technical rigor) can break down barriers.