

CHAPTER 14

The unscented Kalman filter

We use the intuition that it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation.

—Simon Julier, Jeffrey Uhlmann, and Hugh Durrant-Whyte [Jul00]

As discussed earlier, the extended Kalman filter (EKF) is the most widely applied state estimation algorithm for nonlinear systems. However, the EKF can be difficult to tune and often gives unreliable estimates if the system nonlinearities are severe. This is because the EKF relies on linearization to propagate the mean and covariance of the state. This chapter discusses the unscented Kalman filter (UKF), an extension of the Kalman filter that reduces the linearization errors of the EKF. The use of the UKF can provide significant improvement over the EKF.

First, we will take a diversion from filtering in Section 14.1 to investigate how means and covariances propagate in nonlinear equations. In Section 14.2, we will present the unscented transformation, which is a way to approximate how the mean and covariance of a random variable change when the random variable undergoes a nonlinear transformation. In Section 14.3, we will use the previous results to derive the UKF and show that it has less linearization error than the EKF. In Section 14.4, we will present some modifications of the standard UKF which can be used to obtain more accurate or faster filtering results.

14.1 MEANS AND COVARIANCES OF NONLINEAR TRANSFORMATIONS

In this section, we will show how linearization approximations can result in errors in the transformation of means and covariances when a random variable is operated on by a nonlinear function. This section does not really have anything to do directly with state estimation, Kalman filtering, or the UKF. However, this section provides some background that will allow us to develop the UKF later in this chapter. This section will also give us a more complete background to understand the type of problems that can arise in the EKF (which relies on linearization).

Consider the nonlinear transformation

$$\begin{aligned} y_1 &= r \cos \theta \\ y_2 &= r \sin \theta \end{aligned} \quad (14.1)$$

This is a standard polar-to-rectangular coordinate transformation. For instance, we might have a sensor that measures range r and angle θ , and we want to convert the measured data to rectangular coordinates y_1 and y_2 . The coordinate transformation can be written more generally as

$$y = h(x) \quad (14.2)$$

where y is the two-element output of $h(x)$, and the two-element vector x is defined as

$$x = \begin{bmatrix} r \\ \theta \end{bmatrix} \quad (14.3)$$

Suppose that x_1 (which is the range r) is a random variable with a mean of 1 and a standard deviation of σ_r . Suppose that x_2 (which is the angle θ) is a random variable with a mean of $\pi/2$ and a standard deviation of σ_θ . In other words, the means of the components of x are given as $\bar{r} = 1$ and $\bar{\theta} = \pi/2$. In addition, we will assume that r and θ are independent, and that their probability density functions are symmetric around their means (for example, Gaussian or uniform).

14.1.1 The mean of a nonlinear transformation

An initial consideration of the above problem, along with Equation (14.1), would lead us to believe that y_1 has a mean of 0, and y_2 has a mean of 1. In addition, a linearization approach would lead us to the same conclusion. If we perform a first-order linearization of Equation (14.2) and take the expected value of both sides, we obtain

$$\begin{aligned} \bar{y} &= E[h(x)] \\ &\approx E \left[h(\bar{x}) + \left. \frac{\partial h}{\partial x} \right|_{\bar{x}} (x - \bar{x}) \right] \\ &= h(\bar{x}) + \left. \frac{\partial h}{\partial x} \right|_{\bar{x}} E(x - \bar{x}) \\ &= h(\bar{x}) \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \quad (14.4)$$

Our intuition, along with a first-order linearization analysis, both lead us to the same conclusion. However, let us pursue this problem with more rigor to check our previous analysis. We can write r and θ as

$$\begin{aligned} r &= \bar{r} + \tilde{r} \\ \theta &= \bar{\theta} + \tilde{\theta} \end{aligned} \quad (14.5)$$

where \tilde{r} and $\tilde{\theta}$ are simply the deviations of r and θ from their means. A rigorous analysis of the mean of y_1 can be performed as follows:

$$\begin{aligned} \bar{y}_1 &= E(r \cos \theta) \\ &= E[(\bar{r} + \tilde{r}) \cos(\bar{\theta} + \tilde{\theta})] \\ &= E[(\bar{r} + \tilde{r})(\cos \bar{\theta} \cos \tilde{\theta} - \sin \bar{\theta} \sin \tilde{\theta})] \end{aligned} \quad (14.6)$$

Carrying out the multiplication, remembering that \tilde{r} and $\tilde{\theta}$ are independent with symmetric pdfs, and taking the expected value, results in

$$\begin{aligned} \bar{y}_1 &= \bar{r} \cos \bar{\theta} \\ &= 0 \end{aligned} \quad (14.7)$$

Our intuition and our first-order approximation of \bar{y}_1 have been confirmed by rigorous analysis. Let us repeat the process for y_2 :

$$\begin{aligned} \bar{y}_2 &= E(r \sin \theta) \\ &= E[(\bar{r} + \tilde{r}) \sin(\bar{\theta} + \tilde{\theta})] \\ &= E[(\bar{r} + \tilde{r})(\sin \bar{\theta} \cos \tilde{\theta} + \cos \bar{\theta} \sin \tilde{\theta})] \end{aligned} \quad (14.8)$$

Carrying out the multiplication, remembering that \tilde{r} and $\tilde{\theta}$ are independent with symmetric pdfs, and taking the expected value, results in

$$\begin{aligned} \bar{y}_2 &= \bar{r} \sin \bar{\theta} E(\cos \tilde{\theta}) \\ &= E(\cos \tilde{\theta}) \end{aligned} \quad (14.9)$$

We cannot go any further unless we assume some distribution for $\tilde{\theta}$, so let us assume that $\tilde{\theta}$ is uniformly distributed between $\pm\theta_m$. In that case, we can compute

$$\begin{aligned} \bar{y}_2 &= E(\cos \tilde{\theta}) \\ &= \frac{\sin \theta_m}{\theta_m} \end{aligned} \quad (14.10)$$

We expected to get 1 for our answer in confirmation of Equation (14.4), but instead we got some number that is less than 1. [Note that $(\sin \theta_m)/\theta_m < 1$ for all $\theta_m > 0$, and $\lim_{\theta_m \rightarrow 0} (\sin \theta_m)/\theta_m = 1$.] The analysis reveals a problem with our initial intuition and the first-order linearization that we performed earlier. The mean of y_2 will indeed be less than 1. This can be seen by looking at a plot of 300 randomly generated r and θ values, where \tilde{r} is uniformly distributed between ± 0.01 , and $\tilde{\theta}$ is uniformly distributed between ± 0.35 radians. The small variance of \tilde{r} and the large

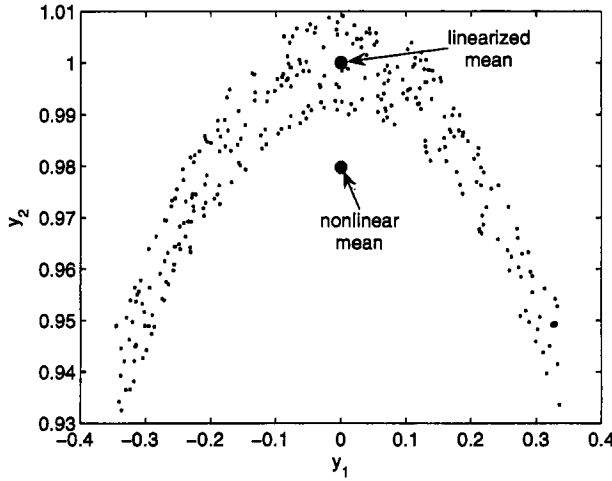


Figure 14.1 Linearized and nonlinear mean of 300 randomly generated points with \tilde{r} uniformly distributed between ± 0.01 and $\tilde{\theta}$ uniformly distributed between ± 0.35 radians.

variance of $\tilde{\theta}$ result in an arc-shaped distribution of points as seen in Figure 14.1. This arc-shaped distribution results in $\bar{y}_2 < 1$.

This is not a Kalman filtering example. But since the EKF uses first-order linearization to update the mean of the state, this example shows the kind of error that can creep into the EKF when it is applied to a nonlinear system.

For a more general analysis of the mean of a nonlinear transformation, recall from Equation (1.89) that $y = h(x)$ can be expanded in a Taylor series around \bar{x} as follows:

$$\begin{aligned} y &= h(x) \\ &= h(\bar{x}) + D_{\bar{x}}h + \frac{1}{2!}D_{\bar{x}}^2h + \frac{1}{3!}D_{\bar{x}}^3h + \dots \end{aligned} \quad (14.11)$$

where $\tilde{x} = x - \bar{x}$. The mean of y can therefore be expanded as

$$\begin{aligned} \bar{y} &= E \left[h(\bar{x}) + D_{\bar{x}}h + \frac{1}{2!}D_{\bar{x}}^2h + \frac{1}{3!}D_{\bar{x}}^3h + \dots \right] \\ &= h(\bar{x}) + E \left[D_{\bar{x}}h + \frac{1}{2!}D_{\bar{x}}^2h + \frac{1}{3!}D_{\bar{x}}^3h + \dots \right] \end{aligned} \quad (14.12)$$

By using $D_{\bar{x}}h$ from Equation (1.88) we can see that

$$\begin{aligned} E[D_{\bar{x}}h] &= E \left[\sum_{i=1}^n \tilde{x}_i \frac{\partial}{\partial x_i} h(x) \Big|_{x=\bar{x}} \right] \\ &= \sum_{i=1}^n E(\tilde{x}_i) \frac{\partial}{\partial x_i} h(x) \Big|_{x=\bar{x}} \\ &= 0 \end{aligned} \quad (14.13)$$

because $E(\tilde{x}_i) = 0$. Likewise, we can see that

$$\begin{aligned} E[D_{\tilde{x}}^3 h] &= E \left[\left(\sum_{i=1}^n \tilde{x}_i \frac{\partial}{\partial x_i} \right)^3 h(x) \Big|_{x=\tilde{x}} \right] \\ &= 0 \end{aligned} \quad (14.14)$$

This is because the sum in the above equation consists only of third-order moments [$E(\tilde{x}_1^3)$, $E(\tilde{x}_1^2 \tilde{x}_2)$, etc.]. These expected values will always be zero as shown at the end of Section 2.2. Similarly all of the odd terms in Equation (14.12) will be zero, which leads to the simplification

$$\bar{y} = h(\bar{x}) + \frac{1}{2!} E[D_{\tilde{x}}^2 h] + \frac{1}{4!} E[D_{\tilde{x}}^4 h] + \dots \quad (14.15)$$

This shows why the mean calculation in Equation (14.4) was incorrect; that calculation was only correct up to the first order. If we approximate \bar{y} for our polar-to-rectangular transformation using terms up to the second order from Equation (14.15), we obtain

$$\begin{aligned} \bar{y} &\approx h(\bar{x}) + \frac{1}{2!} E[D_{\tilde{x}}^2 h] \\ &= h(\bar{x}) + \frac{1}{2} E \left[\left(\sum_{i=1}^2 \tilde{x}_i \frac{\partial}{\partial x_i} \right)^2 h(x) \Big|_{x=\tilde{x}} \right] \\ &= h(\bar{x}) + \frac{1}{2} \left(E(\tilde{x}_1^2) \frac{\partial^2 h(x)}{\partial x_1^2} \Big|_{x=\tilde{x}} + 2E(\tilde{x}_1 \tilde{x}_2) \frac{\partial^2 h(x)}{\partial x_1 \partial x_2} \Big|_{x=\tilde{x}} + E(\tilde{x}_2^2) \frac{\partial^2 h(x)}{\partial x_2^2} \Big|_{x=\tilde{x}} \right) \\ &= h(\bar{x}) + \frac{1}{2} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \sigma_\theta^2 \begin{bmatrix} -r \cos \theta \\ -r \sin \theta \end{bmatrix} \Big|_{x=\tilde{x}} \right) \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \frac{1}{2} \sigma_\theta^2 \begin{bmatrix} 0 \\ -1 \end{bmatrix} \end{aligned} \quad (14.16)$$

We therefore obtain

$$\begin{aligned} \bar{y}_1 &\approx 0 \\ \bar{y}_2 &\approx 1 - \frac{\sigma_\theta^2}{2} \\ &= 1 - \frac{E(\tilde{\theta}^2)}{2} \end{aligned} \quad (14.17)$$

Note that we found the exact value of \bar{y}_2 in Equation (14.9) to be equal to $E(\cos \tilde{\theta})$. The approximate expression found in Equation (14.17) is the first two nonzero terms of the Taylor series expansion of $E(\cos \tilde{\theta})$.

14.1.2 The covariance of a nonlinear transformation

Now we turn our attention to the covariance of a random variable that undergoes a nonlinear transformation. The covariance of y is given as

$$P_y = E[(y - \bar{y})(y - \bar{y})^T] \quad (14.18)$$

We can use Equations (14.11) and (14.15) to write $(y - \bar{y})$ as

$$\begin{aligned} y - \bar{y} &= \left[h(\bar{x}) + D_{\bar{x}}h + \frac{1}{2!}D_{\bar{x}}^2h + \cdots \right] - \\ &\quad \left[h(\bar{x}) + \frac{1}{2!}E(D_{\bar{x}}^2h) + \frac{1}{4!}E(D_{\bar{x}}^4h) + \cdots \right] \\ &= \left[D_{\bar{x}}h + \frac{1}{2!}D_{\bar{x}}^2h + \cdots \right] - \left[\frac{1}{2!}E(D_{\bar{x}}^2h) + \frac{1}{4!}E(D_{\bar{x}}^4h) + \cdots \right] \end{aligned} \quad (14.19)$$

We substitute this expression into Equation (14.18) and use the same type of reasoning as in the previous section to see that all of the odd-powered terms in the expected value evaluate to zero (assuming that \bar{x} is zero-mean with a symmetric pdf). This results in

$$\begin{aligned} P_y &= E[D_{\bar{x}}h(D_{\bar{x}}h)^T] + \\ &\quad E\left[\frac{D_{\bar{x}}h(D_{\bar{x}}^3h)^T}{3!} + \frac{D_{\bar{x}}^2h(D_{\bar{x}}^2h)^T}{2!2!} + \frac{D_{\bar{x}}^3h(D_{\bar{x}}h)^T}{3!}\right] + \\ &\quad E\left(\frac{D_{\bar{x}}^2h}{2!}\right)E\left(\frac{D_{\bar{x}}^2h}{2!}\right)^T + \cdots \end{aligned} \quad (14.20)$$

The first term on the right side of the above equation can be written as

$$\begin{aligned} E[D_{\bar{x}}h(D_{\bar{x}}h)^T] &= E\left[\left(\sum_{i=1}^n \tilde{x}_i \frac{\partial h}{\partial x_i} \Big|_{x=\bar{x}}\right) \left(\cdots\right)^T\right] \\ &= E\left[\sum_{i,j} \tilde{x}_i \frac{\partial h}{\partial x_i} \Big|_{x=\bar{x}} \frac{\partial h^T}{\partial x_j} \Big|_{x=\bar{x}} \tilde{x}_j\right] \\ &= \sum_{i,j} H_i E(\tilde{x}_i \tilde{x}_j) H_j^T \\ &= \sum_{i,j} H_i P_{ij} H_j^T \end{aligned} \quad (14.21)$$

where the partial derivative vector H_i and the expected value P_{ij} are defined by the above equation. Recall from Equation (1.16) that an equation in this form can be written as

$$\begin{aligned} E[D_{\bar{x}}h(D_{\bar{x}}h)^T] &= \frac{\partial h}{\partial x} \Big|_{x=\bar{x}} P \frac{\partial h^T}{\partial x} \Big|_{x=\bar{x}} \\ &= HPH^T \end{aligned} \quad (14.22)$$

where the partial derivative matrix H and the covariance matrix P are defined by the above equation. H_i in Equation (14.21) is the i th column of H , and P_{ij} in Equation (14.21) is the element in the i th row and j th column of $P = E(\bar{x}\bar{x}^T)$. We can use this in Equation (14.20) to write the covariance of a nonlinear transformation $y = h(x)$ as follows:

$$\begin{aligned}
P_y = & HPH^T + E \left[\frac{D_{\bar{x}}h(D_{\bar{x}}^3h)^T}{3!} + \frac{D_{\bar{x}}^2h(D_{\bar{x}}^2h)^T}{2!2!} + \frac{D_{\bar{x}}^3h(D_{\bar{x}}h)^T}{3!} \right] + \\
& E \left(\frac{D_{\bar{x}}^2h}{2!} \right) E \left(\frac{D_{\bar{x}}^2h}{2!} \right)^T + \dots
\end{aligned} \quad (14.23)$$

This is the complete Taylor series expansion for the covariance of a nonlinear transformation.

In the EKF, we use only the first term of this expansion to approximate the covariance of the estimation error. For example, if the measurement $y = h(x) + v$ then we see from Equation (10.98) that the covariance of y is approximated as $P_y = HP_xH^T + R$, where H is the partial derivative of h with respect to x , and R is the covariance of v . Likewise, if the state propagates as $x_{k+1} = f(x_k) + w_k$ then we see from Equation (10.100) that the covariance of x is approximately updated as $P_k^- = FP_{k-1}^+F^T + Q$, where F is the partial derivative of $f(x)$ with respect to x , and Q is the covariance of w_k . However, these covariance approximations can result in significant errors if the underlying functions $h(x)$ and $f(x)$ are highly nonlinear.

For example, consider the nonlinear transformation introduced at the beginning of this section. A linear covariance approximation would indicate that $P_y \approx HP_xH^T$, where H and P_x are given as

$$\begin{aligned}
H &= \left. \frac{\partial h}{\partial x} \right|_{x=\bar{x}} \\
&= \begin{bmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{bmatrix}_{x=\bar{x}} \\
&= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \\
P_x &= E \left(\begin{bmatrix} r - \bar{r} \\ \theta - \bar{\theta} \end{bmatrix} \begin{bmatrix} \dots \end{bmatrix}^T \right) \\
&= \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}
\end{aligned} \quad (14.24)$$

This gives P_y as follows.

$$\begin{aligned}
P_y &\approx HP_xH^T \\
&= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}
\end{aligned} \quad (14.25)$$

This is an approximation of P_y . However, a more rigorous analysis of P_y can be conducted using Equations (14.1), (14.7), and (14.10):

$$\begin{aligned}
P_y &= E[(y - \bar{y})(y - \bar{y})^T] \\
&= E \left[\begin{pmatrix} r \cos \theta \\ r \sin \theta - (\sin \theta_m)/\theta_m \end{pmatrix} \begin{pmatrix} \dots \end{pmatrix}^T \right] \\
&= E \begin{bmatrix} r^2 \cos^2 \theta & r^2 \cos \theta \sin \theta - r \cos \theta (\sin \theta_m)/\theta_m \\ r^2 \cos \theta \sin \theta - r \cos \theta (\sin \theta_m)/\theta_m & (r \sin \theta - (\sin \theta_m)/\theta_m)^2 \end{bmatrix}
\end{aligned} \quad (14.26)$$

We again use our assumption that r and θ are independent, r is uniformly distributed with a mean of 1 and a standard deviation of σ_r , and $\theta = \pi/2 + \tilde{\theta}$, with $\tilde{\theta}$ uniformly distributed between $\pm\theta_m$. We can therefore compute

$$\begin{aligned}
 E(r^2) &= 1 + \sigma_r^2 \\
 E(\cos^2 \tilde{\theta}) &= \frac{1 - E(\cos 2\tilde{\theta})}{2} \\
 E(\cos 2\tilde{\theta}) &= \frac{\sin 2\theta_m}{2\theta_m} \\
 E(\sin \theta) &= E(\cos \tilde{\theta}) \\
 &= \frac{\sin \theta_m}{\theta_m}
 \end{aligned} \tag{14.27}$$

We can use these expressions in Equation (14.26) to compute

$$P_y = \begin{bmatrix} \frac{1}{2}(1 + \sigma_r^2)(1 - \sin 2\theta_m/2\theta_m) & 0 \\ 0 & \frac{1}{2}(1 + \sigma_r^2)(1 + \sin 2\theta_m/2\theta_m) - \sin^2 \theta_m/\theta_m^2 \end{bmatrix} \tag{14.28}$$

This matrix defines a two-dimensional ellipse, where $P_y(1,1)$ specifies the square of the y_1 axis length, and $P_y(2,2)$ specifies the square of the y_2 axis length. Figure 14.2 shows the linearized covariance defined by Equation (14.25), and the exact covariance defined by Equation (14.28). The linearized covariance is centered at the linearized mean, and the exact covariance is centered around at the exact mean. It can be seen that the linearized covariance is not a very good approximation to the exact covariance, at least not in the y_2 direction.

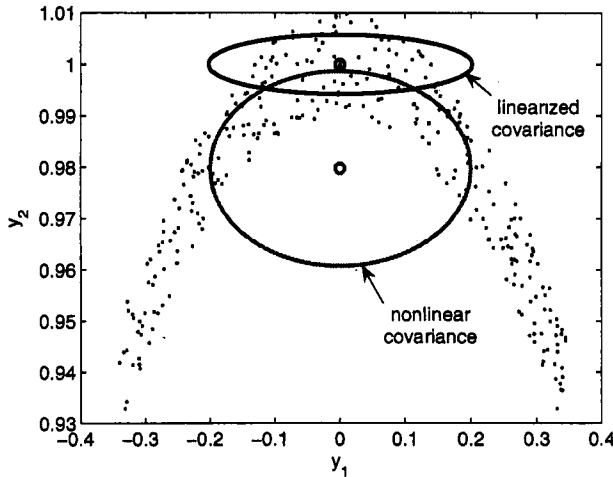


Figure 14.2 Linearized and nonlinear mean and covariance of 300 randomly generated points with \tilde{r} uniformly distributed between ± 0.01 and $\tilde{\theta}$ uniformly distributed between ± 0.35 radians.

This is not a Kalman filtering example. But since the EKF uses first-order linearization to update the covariance of the state, this example shows the kind of error that can creep into the EKF when it is applied to a nonlinear system.

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:

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

where \sqrt{nP} is the matrix square root of nP such that $(\sqrt{nP})^T \sqrt{nP} = nP$, and $(\sqrt{nP})_i$ is the i th 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:

$$y^{(i)} = h \left(x^{(i)} \right) \quad i = 1, \dots, 2n \quad (14.30)$$

¹MATLAB'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:

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

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

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

Equation (14.31) can therefore be written as

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned}
 \frac{1}{2n} \sum_{i,j=1}^n \sum_{k=1}^n \tilde{x}_i^{(k)} \tilde{x}_j^{(k)} \frac{\partial^2 h(x)}{\partial x_i \partial x_j} \Big|_{x=\tilde{x}} &= \frac{1}{2n} \sum_{i,j=1}^n \sum_{k=1}^n (\sqrt{nP})_{k_i} (\sqrt{nP})_{k_j} \frac{\partial^2 h(x)}{\partial x_i \partial x_j} \Big|_{x=\tilde{x}} \\
 &= \frac{1}{2n} \sum_{i,j=1}^n n P_{ij} \frac{\partial^2 h(x)}{\partial x_i \partial x_j} \Big|_{x=\tilde{x}} \\
 &= \frac{1}{2} \sum_{i,j=1}^n P_{ij} \frac{\partial^2 h(x)}{\partial x_i \partial x_j} \Big|_{x=\tilde{x}} \quad (14.38)
 \end{aligned}$$

Equation (14.36) can therefore be written as

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

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

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

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

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

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

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

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

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:

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

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:

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

$$\begin{aligned}
&= \frac{1}{2n} \sum_{i=1}^{2n} \left[h(\bar{x}) + D_{\bar{x}^{(i)}} h + \frac{1}{2} D_{\bar{x}^{(i)}}^2 h + \frac{1}{3!} D_{\bar{x}^{(i)}}^3 h + \dots \right. \\
&\quad \left. - h(\bar{x}) - \frac{1}{2n} \sum_{j=1}^{2n} \left(\frac{1}{2} D_{\bar{x}^{(j)}}^2 h + \frac{1}{4!} D_{\bar{x}^{(j)}}^4 h + \dots \right) \right] \left[\dots \right]^T \quad (14.45)
\end{aligned}$$

Multiplying this equation out gives

$$\begin{aligned}
P_u &= \frac{1}{2n} \sum_{i=1}^{2n} \left\{ (D_{\bar{x}^{(i)}} h) (\dots)^T + \underbrace{\left[\left(\frac{1}{2} D_{\bar{x}^{(i)}} h \right) (D_{\bar{x}^{(i)}}^2 h)^T \right]}_0 + \underbrace{\left[\dots \right]^T}_0 + \right. \\
&\quad \frac{1}{4!} (D_{\bar{x}^{(i)}}^2 h) (\dots)^T - \underbrace{\left[D_{\bar{x}^{(i)}} h \left(\frac{1}{2n} \sum_j \frac{1}{2} D_{\bar{x}^{(j)}}^2 h \right)^T \right]}_0 - \underbrace{\left[\dots \right]^T}_0 + \\
&\quad \frac{1}{4n^2} \left(\sum_j D_{\bar{x}^{(j)}}^2 h \right) (\dots)^T - \left[\frac{1}{4n} D_{\bar{x}^{(i)}}^2 h \left(\sum_j D_{\bar{x}^{(j)}}^2 h \right)^T \right] - \left[\dots \right]^T + \\
&\quad \left. \left[D_{\bar{x}^{(i)}} h \left(\frac{1}{3!} D_{\bar{x}^{(j)}}^3 h \right)^T \right] + \left[\dots \right]^T + \dots \right\} \quad (14.46)
\end{aligned}$$

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

$$P_u = \frac{1}{2n} \sum_{i=1}^{2n} (D_{\bar{x}^{(i)}} h) (\dots)^T + \text{HOT} \quad (14.47)$$

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

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

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

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

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:

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

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

3. Transform the sigma points as follows:

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

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

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

■ 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

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

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.

▽▽▽

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

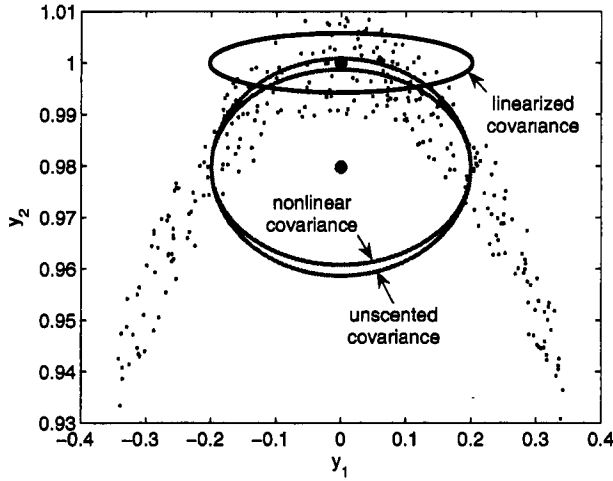


Figure 14.3 Results of Example 14.1. A comparison of the exact, linearized, and unscented mean and covariance of 300 randomly generated points with \tilde{r} uniformly distributed between ± 0.01 and $\tilde{\theta}$ uniformly distributed between ± 0.35 radians.

to propagate the mean and covariance of a system using a time-update and a measurement update. If the system is linear, then the mean and covariance can be exactly updated with the Kalman filter (Chapter 5). If the system is nonlinear, then the mean and covariance can be approximately updated with the extended Kalman filter (Section 13.2). However, the EKF is based on linearization, and the previous section showed that unscented transformations are more accurate than linearization for propagating means and covariances. Therefore, we simply replace the EKF equations with unscented transformations to obtain the UKF algorithm. The UKF algorithm can be summarized as follows.

The unscented Kalman filter

1. We have an n -state discrete-time nonlinear system given by

$$\begin{aligned}
 x_{k+1} &= f(x_k, u_k, t_k) + w_k \\
 y_k &= h(x_k, t_k) + v_k \\
 w_k &\sim (0, Q_k) \\
 v_k &\sim (0, R_k)
 \end{aligned} \tag{14.56}$$

2. The UKF is initialized as follows.

$$\begin{aligned}
 \hat{x}_0^+ &= E(x_0) \\
 P_0^+ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]
 \end{aligned} \tag{14.57}$$

3. The following time update equations are used to propagate the state estimate and covariance from one measurement time to the next.

- (a) To propagate from time step $(k-1)$ to k , first choose sigma points $x_{k-1}^{(i)}$ as specified in Equation (14.29), with appropriate changes since the current best guess for the mean and covariance of x_k are \hat{x}_{k-1}^+ and P_{k-1}^+ :

$$\begin{aligned}\hat{x}_{k-1}^{(i)} &= \hat{x}_{k-1}^+ + \tilde{x}^{(i)} \quad i = 1, \dots, 2n \\ \tilde{x}^{(i)} &= \left(\sqrt{nP_{k-1}^+} \right)_i^T \quad i = 1, \dots, n \\ \tilde{x}^{(n+i)} &= - \left(\sqrt{nP_{k-1}^+} \right)_i^T \quad i = 1, \dots, n\end{aligned} \quad (14.58)$$

- (b) Use the known nonlinear system equation $f(\cdot)$ to transform the sigma points into $\hat{x}_k^{(i)}$ vectors as shown in Equation (14.30), with appropriate changes since our nonlinear transformation is $f(\cdot)$ rather than $h(\cdot)$:

$$\hat{x}_k^{(i)} = f(\hat{x}_{k-1}^{(i)}, u_k, t_k) \quad (14.59)$$

- (c) Combine the $\hat{x}_k^{(i)}$ vectors to obtain the *a priori* state estimate at time k . This is based on Equation (14.33):

$$\hat{x}_k^- = \frac{1}{2n} \sum_{i=1}^{2n} \hat{x}_k^{(i)} \quad (14.60)$$

- (d) Estimate the *a priori* error covariance as shown in Equation (14.43). However, we should add Q_{k-1} to the end of the equation to take the process noise into account:

$$P_k^- = \frac{1}{2n} \sum_{i=1}^{2n} \left(\hat{x}_k^{(i)} - \hat{x}_k^- \right) \left(\hat{x}_k^{(i)} - \hat{x}_k^- \right)^T + Q_{k-1} \quad (14.61)$$

4. Now that the time update equations are done, we implement the measurement-update equations.

- (a) Choose sigma points $x_k^{(i)}$ as specified in Equation (14.29), with appropriate changes since the current best guess for the mean and covariance of x_k are \hat{x}_k^- and P_k^- :

$$\begin{aligned}\hat{x}_k^{(i)} &= \hat{x}_k^- + \tilde{x}^{(i)} \quad i = 1, \dots, 2n \\ \tilde{x}^{(i)} &= \left(\sqrt{nP_k^-} \right)_i^T \quad i = 1, \dots, n \\ \tilde{x}^{(n+i)} &= - \left(\sqrt{nP_k^-} \right)_i^T \quad i = 1, \dots, n\end{aligned} \quad (14.62)$$

This step can be omitted if desired. That is, instead of generating new sigma points we can reuse the sigma points that were obtained from the time update. This will save computational effort if we are willing to sacrifice performance.

- (b) Use the known nonlinear measurement equation $h(\cdot)$ to transform the sigma points into $\hat{y}_k^{(i)}$ vectors (predicted measurements) as shown in Equation (14.30):

$$\hat{y}_k^{(i)} = h(\hat{x}_k^{(i)}, t_k) \quad (14.63)$$

- (c) Combine the $\hat{y}_k^{(i)}$ vectors to obtain the predicted measurement at time k . This is based on Equation (14.33):

$$\hat{y}_k = \frac{1}{2n} \sum_{i=1}^{2n} \hat{y}_k^{(i)} \quad (14.64)$$

- (d) Estimate the covariance of the predicted measurement as shown in Equation (14.43). However, we should add R_k to the end of the equation to take the measurement noise into account:

$$P_y = \frac{1}{2n} \sum_{i=1}^{2n} \left(\hat{y}_k^{(i)} - \hat{y}_k \right) \left(\hat{y}_k^{(i)} - \hat{y}_k \right)^T + R_k \quad (14.65)$$

- (e) Estimate the cross covariance between \hat{x}_k^- and \hat{y}_k based on Equation (14.43):

$$P_{xy} = \frac{1}{2n} \sum_{i=1}^{2n} \left(\hat{x}_k^{(i)} - \hat{x}_k^- \right) \left(\hat{y}_k^{(i)} - \hat{y}_k \right)^T \quad (14.66)$$

- (f) The measurement update of the state estimate can be performed using the normal Kalman filter equations as shown in Equation (10.100):

$$\begin{aligned} K_k &= P_{xy} P_y^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k (y_k - \hat{y}_k) \\ P_k^+ &= P_k^- - K_k P_y K_k^T \end{aligned} \quad (14.67)$$

The algorithm above assumes that the process and measurement equations are linear with respect to the noise, as shown in Equation (14.56). In general, the process and measurement equations may have noise that enters the process and measurement equations nonlinearly. That is,

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k, t_k) \\ y_k &= h(x_k, v_k, t_k) \end{aligned} \quad (14.68)$$

In this case, the UKF algorithm presented above is not rigorous because it treats the noise as additive, as seen in Equations (14.61) and (14.65). To handle this situation, we can augment the noise onto the state vector as shown in [Jul04, Wan01]:

$$x_k^{(a)} = \begin{bmatrix} x_k \\ w_k \\ v_k \end{bmatrix} \quad (14.69)$$

Then we can use the UKF to estimate the augmented state $x_k^{(a)}$. The UKF is initialized as

$$\begin{aligned}\hat{x}_0^{a+} &= \begin{bmatrix} E(x_0) \\ 0 \\ 0 \end{bmatrix} \\ P_0^{a+} &= \begin{bmatrix} E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] & 0 & 0 \\ 0 & Q_0 & 0 \\ 0 & 0 & R_0 \end{bmatrix}\end{aligned}\quad (14.70)$$

Then we use the UKF algorithm presented above, except that we are estimating the augmented mean and covariance, so we remove Q_{k-1} and R_k from Equations (14.61) and (14.65).

■ EXAMPLE 14.2

Suppose we are trying to estimate the altitude x_1 , velocity x_2 , and constant ballistic coefficient x_3 of a body as it falls toward earth. A range measuring device is located at an altitude a and the horizontal range between the measuring device and the body is M . This system is the same as the one in Example 13.3. The equations for this system are

$$\begin{aligned}\dot{x}_1 &= x_2 + w_1 \\ \dot{x}_2 &= \rho_0 \exp(-x_1/k) x_2^2 x_3 / 2 - g + w_2 \\ \dot{x}_3 &= w_3 \\ y(t_k) &= \sqrt{M^2 + (x_1(t_k) - a)^2} + v_k\end{aligned}\quad (14.71)$$

As usual, w_i is the noise that affects the i th process equation, and v is the measurement noise. ρ_0 is the air density at sea level, k is a constant that defines the relationship between air density and altitude, and g is the acceleration due to gravity. We will use the continuous-time system equations to simulate the system, and suppose that we obtain range measurements every 0.5 seconds. The constants that we will use are given as

$$\begin{aligned}\rho_0 &= 2 \text{ lb-sec}^2/\text{ft}^4 \\ g &= 32.2 \text{ ft/sec}^2 \\ k &= 20,000 \text{ ft} \\ E[v_k^2] &= 10,000 \text{ ft}^2 \\ E[w_i^2(t)] &= 0 \quad i = 1, 2, 3 \\ M &= 100,000 \text{ ft} \\ a &= 100,000 \text{ ft}\end{aligned}\quad (14.72)$$

The initial conditions of the system and the estimator are given as

$$\begin{aligned}x_0 &= [300,000 \quad -20,000 \quad 0.001]^T \\ \hat{x}_0^+ &= x_0 \\ P_0^+ &= \begin{bmatrix} 1,000,000 & 0 & 0 \\ 0 & 4,000,000 & 0 \\ 0 & 0 & 10 \end{bmatrix}\end{aligned}\quad (14.73)$$

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.

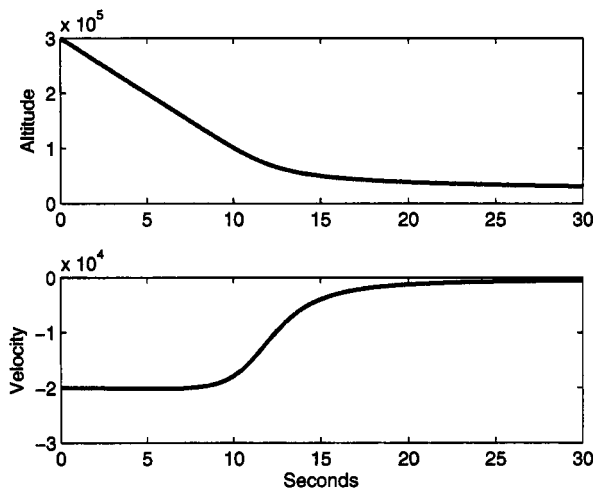


Figure 14.4 Altitude and velocity of a falling body for Example 14.2.

▽▽▽

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

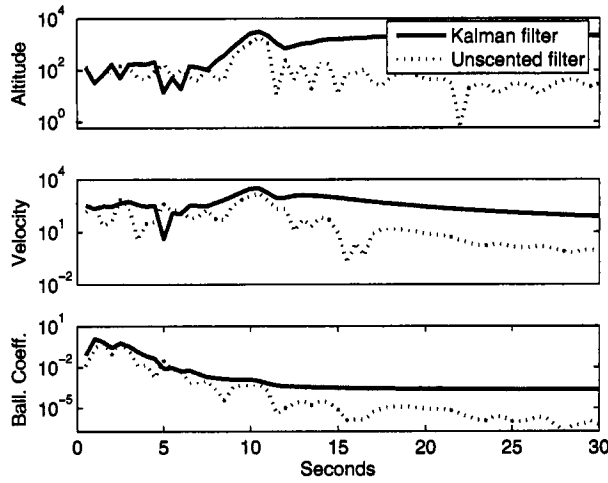


Figure 14.5 Kalman filter and unscented filter estimation-error magnitudes of the altitude, velocity, and ballistic coefficient of a falling body for Example 14.2.

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:

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

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

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

The unscented mean and covariance approximations are computed as

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

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:

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

3. Initialize the following one-element vectors:

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

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

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

where 0_j is the column vector containing j zeros.

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

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

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:

$$W^{(i)} = \frac{1 - W^{(0)}}{n+1} \quad i = 1, \dots, n+1 \quad (14.81)$$

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:

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

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

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

where 0_j is the column vector containing j zeros.

5. After the above recursion is complete, we have the n -element vectors $\sigma_i^{(n)}$ ($i = 0, \dots, n+1$). 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:

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

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

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

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

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

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

▽▽▽

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.

²It 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.

PROBLEMS

Written exercises

14.1 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = x^2$. What is \bar{y} ? What is the first-order approximation to \bar{y} ? What is the second-order approximation to \bar{y} ?

14.2 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is \bar{y} ? What is the first-order approximation to \bar{y} ? What is the second-order approximation to \bar{y} ? What is the third-order approximation to \bar{y} ? What is the fourth-order approximation to \bar{y} ?

14.3 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is the variance of y ? What is the first-order approximation to the variance of y ? What is the fourth-order approximation to the variance of y ?

14.4 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is \bar{y} ? What is the unscented approximation to \bar{y} ?

14.5 Consider the matrix

$$P = \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix}$$

Find an upper triangular matrix S (using only paper and pencil) such that $S^T S = P$. Find a lower triangular matrix S such that $S^T S = P$. (Note the difference between your solution to this problem and the solution to Problem 6.7.)

14.6 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is the variance of y ? What is unscented approximation to the variance of y ?

14.7 Show that for a system with an identity transition matrix, the UKF algorithm gives $\hat{x}_k^- = \hat{x}_{k-1}^+$.

14.8 Show that for a system with $y_k = x_k$, the UKF gain K_k is positive definite.

14.9 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is \bar{y} ? Use the generalized unscented transformation to approximate \bar{y} with $\kappa = 0$, $\kappa = 1$, and $\kappa = 2$.

14.10 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is the variance of y ? Use the generalized unscented transformation to approximate the variance of y with $\kappa = 0$, $\kappa = 1$, and $\kappa = 2$.

14.11 Consider the simplex sigma-point algorithm. Prove that $\sum_i W^{(i)} \sigma_i^{(j)} = 0$ (i.e., the weighted sample mean of the $\sigma_i^{(j)}$ vectors is zero).

14.12 Prove that the sum of the weights in the simplex sigma-point algorithm is equal to 1.

14.13 Consider the simplex sigma-point algorithm. Prove that the $\sum_i W^{(i)} x^{(i)} = \bar{x}$ (i.e., the weighted sample mean of the sigma points is equal to \bar{x}). (Hint: Use the results of Problems 14.11 and 14.12.)

Computer exercises

14.14 Design an unscented Kalman filter for the system described in Problem 13.21. Simulate the system and the filter for 60 s. Plot the estimation error for the four states. What is the experimental standard deviation of the estimation error for each of the four states? Based on the steady-state covariance matrix of the filter, what is the theoretical standard deviation of the estimation error for each of the four states? How does this compare with the extended Kalman filter results of Problem 13.21?

14.15 An inverted pendulum on a cart can be modeled as follows [Bay99, Che99].

$$\ddot{\theta} = \frac{mgl \sin \theta (M + m) - ml \cos \theta (u + ml\dot{\theta}^2 \sin \theta - B\dot{d})}{(J + ml^2)(M + m) - m^2 l^2 \cos^2 \theta}$$

$$\ddot{d} = \frac{u - ml\ddot{\theta} \cos \theta + ml\dot{\theta}^2 \sin \theta - B\dot{d}}{M + m}$$

The quantities in the system model are as follows:

- $\theta(0)$ = initial angle (0.1 rad)
- $d(0)$ = initial cart displacement (0 rad)
- m = pendulum mass (0.2 kg)
- M = cart mass (1 kg)
- g = acceleration due to gravity (9.81 m/s²)
- B = coefficient of friction between cart and ground [0.1 N/(m/s)]
- l = pendulum length (1 m)
- r = pendulum mass radius (0.02 m)
- u = external force applied to cart
- J = pendulum moment of inertia
- = $mr^2/2$

where we have assumed that the pendulum mass is concentrated in a cylinder at the end of the pendulum. Define the state of the system as $x = [d \ \dot{d} \ \theta \ \dot{\theta}]^T$. The horizontal displacement d is measured every 5 ms with a standard deviation of 0.1 m. The continuous-time process noise is $Q_c = \text{diag}(0, 0.0004, 0, 0.04)$. The system can be linearized (so that an EKF can be used to estimate the state) by assuming that θ is small, so $\cos \theta \approx 1$, $\sin \theta \approx \theta$, and $\dot{\theta}^2 \approx 0$. Suppose that the feedback control signal is given as $u = 40\theta$ and the initial state is perfectly known. Write an EKF and a UKF to estimate the state, where the control is assumed by the filters to be $\hat{u} = 40\hat{\theta}$. Plot the true states and estimated states for a 2 second simulation. Which filter appears to perform better?