Lecture 21: Unscented Kalman Filter

These lecture notes are based on the approach presented in [1, Ch. 14] and [2, Ch. 3.4]. Recall that the EKF linearized the transformation of a Gaussian random variable using a Taylor series expansion. However, if the nonlinearities are severe the EKF may give unreliable estimates. More accurate ways in which this linearization can be achieved are via:

- *Moment matching*: the linearization is calculated to preserve the true mean and the true covariance of the posterior
- *Unscented transform*: the linearization uses a weighted statistical linear regression using several *sigma points*

In this lecture, we will discuss the Unscented Kalman Filter (UKF) that uses the uscented transform to propagate the mean and covariance of a Gaussian random variable through a nonlinear function.

Unscented Transform

Consider the Gaussian random vector $\boldsymbol{x} \in \mathbb{R}^n$ with a known mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{P}$. That is, $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{P})$. The random vector is transformed through a nonlinear function

$$\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x})$$

where $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^m$. The *unscented transform* takes advantage of two fundamental principles: (i) it is easy to transform a single point $\boldsymbol{x}$ through $\boldsymbol{g}(\boldsymbol{x})$ and (ii) it's not difficult to find a set of particular points in the state-space $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(p)}$ whose sample pdf approximates the true pdf. Suppose we choose a particular set of p *sigma points* in the state-space

$$\mathcal{S} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(p)}\}$$

and associate with each sigma point a scalar weight $w^{(i)}$ for $i = 1, \dots, p$. After propagating each of the sigma points through our nonlinear function we obtain the transformed sigma points

$$\boldsymbol{z}^{(i)} = \boldsymbol{g}(\boldsymbol{x}^{(i)}) \tag{1}$$

for $i = 1, \dots, p$. This process is sketched below for carefully chosen sigma points from a particular level set of the Gaussian multivariate p.d.f.

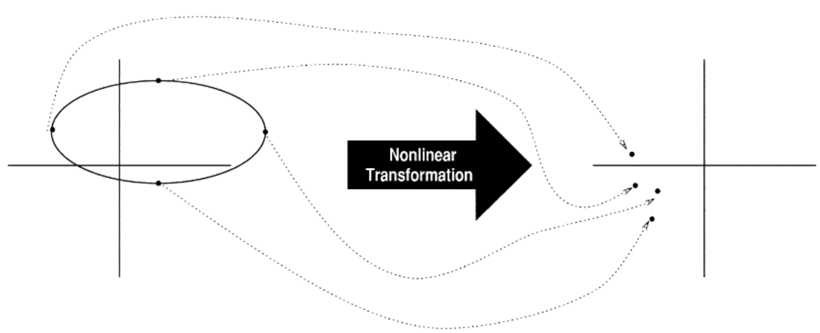


Image Source: [3]

The mean is then given as the weighted average of the points:

$$\bar{\boldsymbol{z}} = \sum_{i=1}^{p} w^{(i)} \boldsymbol{z}^{(i)}$$

where, to have an unbiased estimate, we require that the weights sum to one:

$$\sum_{i=1}^{p} w^{(i)} = 1$$

The covariance is the weighted outer product of the transformed points:

$$\boldsymbol{\Sigma}_z = \sum_{i=1}^{p} w^{(i)} \{\boldsymbol{z}^{(i)} - \bar{\boldsymbol{z}}\}\{\boldsymbol{z}^{(i)} - \bar{\boldsymbol{z}}\}^T$$

Any set of sigma points with uniform weights $w^{(i)} = 1/(2n)$ for all $i = 1, \ldots, 2n$, that correctly encode the initial (untransformed) mean and covariance, $\boldsymbol{\mu}$ and $\boldsymbol{P}$, can be chosen in the above approximation. One choice is to use $2n$ sigma points that are located on covariance contours:

$$\boldsymbol{x}^{(i)} = \bar{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^{(i)}, \qquad i = 1, \ldots, 2n \tag{2}$$

where

$$\tilde{\boldsymbol{x}}^{(i)} = (\sqrt{n\boldsymbol{P}})_i^T, \qquad i = 1, \ldots, n \tag{3}$$

$$\tilde{\boldsymbol{x}}^{(i+q)} = -(\sqrt{n\boldsymbol{P}})_i^T, \qquad i = 1, \ldots, n \tag{4}$$

and $\boldsymbol{L} = \sqrt{n\boldsymbol{P}}$ is the *matrix square root* of $n\boldsymbol{P}$ such that $n\boldsymbol{P} = \boldsymbol{L}\boldsymbol{L}^T$ and $(\sqrt{q\boldsymbol{P}})_i$ is the ith row of $\sqrt{n\boldsymbol{P}}$. Since $n\boldsymbol{P}$ is always positive semi-definite it can always be decomposed as $n\boldsymbol{P} = \boldsymbol{L}\boldsymbol{L}^T$.)

Aside: Square root of matrices A common algorithm for computing the square root of a matrix is Cholesky decomposition. This algorithm is often used to solve the equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ when $\boldsymbol{A}$ is a symmetric, positive-definite matrix. In that case the Cholsky decomposition solves for a lower-triangular matrix $\boldsymbol{L}$ where $\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{A}$. Because of this triangular structure it easy to solve the equiavalent problem $\boldsymbol{L}\boldsymbol{z} = \boldsymbol{b}$ for $\boldsymbol{z}$ using forward substitution where $\boldsymbol{z} = \boldsymbol{L}^T\boldsymbol{x}$. Then $\boldsymbol{z} = \boldsymbol{L}^T\boldsymbol{x}$ can be solved for the desired vector $\boldsymbol{x}$ using backwards substitution. The matrix square root can be found using Cholsky factorization implemented in MATLAB as `chol`.

Accuracy of the Unscented Transform

Suppose we have a random vector $\boldsymbol{x}$ (with a known mean $\bar{\boldsymbol{x}}$ and covariance $\boldsymbol{P}$) that is transformed by the nonlinear function $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x})$. We wish to approximate the true mean $\bar{\boldsymbol{z}} = E[\boldsymbol{z}]$ by taking the above sigma points $\boldsymbol{x}^{(i)}$ and transforming each one via $\boldsymbol{g}(\cdot)$ and then computing a weighted sum of the outputs. Denote the transformed sigma points as

$$\boldsymbol{z}^{(i)} = \boldsymbol{g}(\boldsymbol{x}^{(i)}) \qquad \text{for} \quad i = 1, 2, \cdots, 2n \tag{5}$$

The approximate mean is denoted $\bar{z}_u$ and is computed as follows:

$$\bar{z}_u = \sum_{i=1}^{2n} w^{(i)} z^{(i)} \tag{6}$$

where the weighs are all equal and sum to one:

$$w^{(i)} = \frac{1}{2n} \qquad \text{for} \quad i = 1, 2, \cdots, 2n \tag{7}$$

Thus, the

$$\bar{z}_u = \frac{1}{2n} \sum_{i=1}^{2n} z^{(i)} \tag{8}$$

and The covariance is the weighted outer product of the transformed points:

$$\Sigma_z = \frac{1}{2n} \sum_{i=1}^{p} \{z^{(i)} - \bar{z}\}\{z^{(i)} - \bar{z}\}^T$$

The unscented transform is able to more accurately approximate the mean and covariance of a nonlinear transformation than a linearization. In the Appendix to this lecture, we prove that the UT approximation is accurate to third order whereas our previously used linearization approach is accurate only to first order.

Example: The following example considers the transformation of two random variables: a noisy distance $r = \bar{r} + \tilde{r}$ and a noise angle $\theta = \bar{\theta} + \tilde{\theta}$ through the nonlinear polar transformation:

$$y_1 = r \cos\theta \tag{9}$$
$$y_2 = r \sin\theta \tag{10}$$

A comparison of the nonlinear, linearized, and unscented predictions of the mean and covariance are shown below.

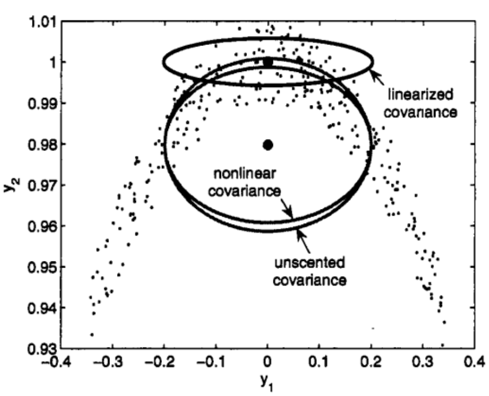


Figure 14.3 Results of Example 14.1. A comparison of the exact, linearized, and unscented mean and covariance of 300 randomly generated points with $\tilde{r}$ uniformly distributed between ± 0.01 and $\tilde{\theta}$ uniformly distributed between ± 0.35 radians.

Image Source: [1]

Algorithm: Unscented Kalman Filter

1. Ensure your system is in the form:

$$\boldsymbol{x}_k = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{w}_{k-1} \tag{11}$$
$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{12}$$
$$\boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{Q}) \tag{13}$$
$$\boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{R}) \tag{14}$$

where the system dynamics $\boldsymbol{f}_k$, measurement equation $\boldsymbol{h}_k$, and control input $\boldsymbol{u}_k$ is known for all $k = 0, 1, 2, \ldots$. The process and measurement noise covariances $\boldsymbol{Q}$ and $\boldsymbol{R}$ are also known where

$$E[\boldsymbol{w}_k \boldsymbol{w}_k^T] = \boldsymbol{Q} \tag{15}$$
$$E[\boldsymbol{v}_k \boldsymbol{v}_k^T] = \boldsymbol{R} \tag{16}$$

and it is assumed that $E[\boldsymbol{w}_k] = 0$ and $E[\boldsymbol{v}_k] = 0$ for all $k = 0, 1, 2, \ldots$ and $E[\boldsymbol{w}_k \boldsymbol{w}_q^T] = 0$ and $E[\boldsymbol{v}_k \boldsymbol{v}_q^T] = 0$ for all $q \neq k$.

2. Select an initial guess

$$\hat{\boldsymbol{x}}_{0|0} = E[\boldsymbol{x}_0] \tag{17}$$
$$\boldsymbol{P}_{0|0} = E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_{0|0})(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_{0|0})^{\mathrm{T}}] \tag{18}$$

for the filter and an initial covariance. In the absence of other information you may choose $\hat{\boldsymbol{x}}_0 = \boldsymbol{0}$ and $\boldsymbol{P}_0 = \boldsymbol{1}_{n \times n} \sigma_0^2$ where σ_0^2 is a large number (e.g., 10E6).

3. For $k = 1, 2, 3, \ldots$ perform the following:

   (a) Choose $2n$ sigma points $\boldsymbol{x}_k^{(i)}$ (where n is the size of the state vector) as specified by (2) with the change that the mean and covariance uses the posterior from the $(k-1)$th step):

$$\bar{\boldsymbol{x}}_{k-1|k-1}^{(i)} = \hat{\boldsymbol{x}}_{k-1|k-1} + \tilde{\boldsymbol{x}}^{(i)}, \qquad i = 1, \ldots, 2n \tag{19}$$

   where

$$\tilde{\boldsymbol{x}}^{(i)} = (\sqrt{n \boldsymbol{P}_{k-1}^{+}})_i^T, \qquad i = 1, \ldots, n \tag{20}$$
$$\tilde{\boldsymbol{x}}^{(i+n)} = -(\sqrt{n \boldsymbol{P}_{k-1}^{+}})_i^T, \qquad i = 1, \ldots, n \tag{21}$$

   (b) Propagate the sigma points through the nonlinear function describing the dynamics, similar to (1), to obtain the transformed state vectors (motion updated):

$$\begin{aligned}
\bar{\boldsymbol{x}}_{k|k-1}^{(1)} &= \boldsymbol{f}(\bar{\boldsymbol{x}}_{k-1|k-1}^{(1)}, \boldsymbol{u}_{k-1}) \\
\bar{\boldsymbol{x}}_{k|k-1}^{(2)} &= \boldsymbol{f}(\bar{\boldsymbol{x}}_{k-1|k-1}^{(2)}, \boldsymbol{u}_{k-1}) \\
&\vdots \\
\bar{\boldsymbol{x}}_{k|k-1}^{(2n)} &= \boldsymbol{f}(\bar{\boldsymbol{x}}_{k-1|k-1}^{(2n)}, \boldsymbol{u}_{k-1})
\end{aligned}$$

(c) Average the transformed state vectors to obtain the predicted state at time-step k:

$$\hat{\boldsymbol{x}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \bar{\boldsymbol{x}}_{k|k-1}^{(i)} \tag{22}$$

and, similarily, compute the *a priori* state error covariance matrix:

$$\hat{\boldsymbol{P}}_k^- = \frac{1}{2n}\sum_{i=1}^{2n} (\bar{\boldsymbol{x}}_k^{(i)} - \hat{\boldsymbol{x}}_{k|k-1})(\bar{\boldsymbol{x}}_k^{(i)} - \hat{\boldsymbol{x}}_{k|k-1})^{\mathrm{T}} + \boldsymbol{Q}_{k-1} \tag{23}$$

where $\boldsymbol{Q}_{k-1}$ accounts for the process noise.

4. Next, the measurement update equations are implemented.

(a) Pass each transformed sigma points through the measurement equation.

$$\begin{aligned}
\bar{\boldsymbol{z}}_{k|k+1}^{(1)} &= \boldsymbol{g}(\bar{\boldsymbol{x}}_{k|k-1}^{(1)}) \\
\bar{\boldsymbol{z}}_{k|k+1}^{(2)} &= \boldsymbol{g}(\bar{\boldsymbol{x}}_{k|k-1}^{(2)}) \\
&\vdots \\
\bar{\boldsymbol{z}}_{k|k+1}^{(2n)} &= \boldsymbol{g}(\bar{\boldsymbol{x}}_{k|k-1}^{(2n)})
\end{aligned}$$

Note: we could have generated new sigma points at this stage based on $\hat{\boldsymbol{x}}_{k|k-1}$ and $\hat{\boldsymbol{P}}_k^-$ to improve filter accuracy.

(b) Compute the mean measurement

$$\hat{\boldsymbol{z}}_k = \frac{1}{2n}\sum_{i=1}^{2n} \bar{\boldsymbol{z}}_{k|k-1}^{(i)} \tag{24}$$

and the covariance

$$\hat{\boldsymbol{P}}_y = \frac{1}{2n}\sum_{i=1}^{2n} (\bar{\boldsymbol{z}}_{k|k-1}^{(i)} - \hat{\boldsymbol{z}}_k)(\bar{\boldsymbol{z}}_{k|k-1}^{(i)} - \hat{\boldsymbol{z}}_k)^{\mathrm{T}} + \boldsymbol{R}_{k-1} \tag{25}$$

(c) Compute the cross-covariance between

$$\hat{\boldsymbol{P}}_{xy} = \frac{1}{2n}\sum_{i=1}^{2n} (\bar{\boldsymbol{x}}_{k|k-1}^{(i)} - \hat{\boldsymbol{x}}_{k|k-1})(\bar{\boldsymbol{z}}_{k|k-1}^{(i)} - \hat{\boldsymbol{z}}_{k|k-1})^{\mathrm{T}} \tag{26}$$

5. The measurement update of the state estimate can be performed using the standard DT-KF equations:

$$\boldsymbol{K}_k = \boldsymbol{P}_{xy}\boldsymbol{P}_y^{-1} \tag{27}$$

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_k) \tag{28}$$

$$\boldsymbol{P}_k^+ = \boldsymbol{P}_k^- - \boldsymbol{K}_k\boldsymbol{P}_y\boldsymbol{K}_k^T \tag{29}$$

Example

The following example is adapted from [1, p. 451]. Suppose we are trying to estimate the altitude x_1, velocity x_2, and constant ballistic coefficient x_3 of an object as it falls towards earth. A range measurement device is located at an altitude a and the horizontal range between measuring device and the object is M. The equations for this system are:

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

As usual, w_i is the noise that affects the ith 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. Suppose range measurements are obtained every T_s seconds. The constants are

$$\rho_0 = 2 \text{ lb} - \text{sec}^2/\text{ft}^4 \tag{30}$$
$$g = 32.2 \text{ ft}/\text{sec}^2 \tag{31}$$
$$k = 20,000 \text{ ft} \tag{32}$$
$$E[v_k^2] = 10,000 \text{ ft} \tag{33}$$
$$E[w_i^2(t)] = 0 \qquad \text{for all} \qquad i = 1, 2, 3 \tag{34}$$
$$M = 100,000 \text{ ft} \tag{35}$$
$$a = 100,000 \text{ ft} \tag{36}$$

The actual initial conditions of the system are

$$\boldsymbol{x}_0 = [300,000 \quad -20,000 \quad 0.001]^{\mathrm{T}} \tag{37}$$

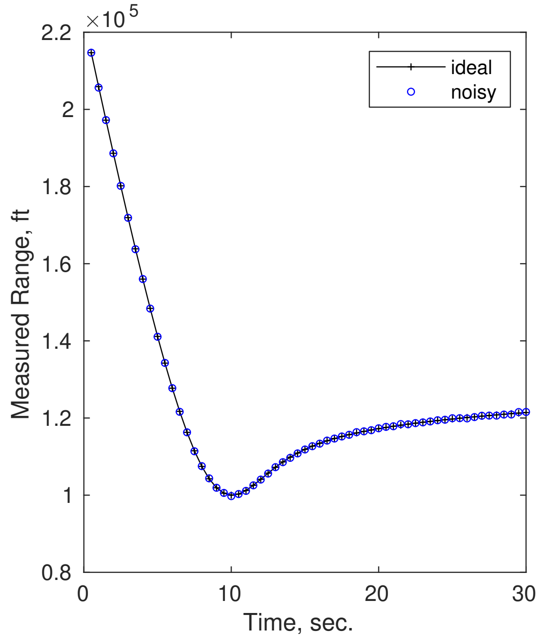
but the estimator is initialized with the guess

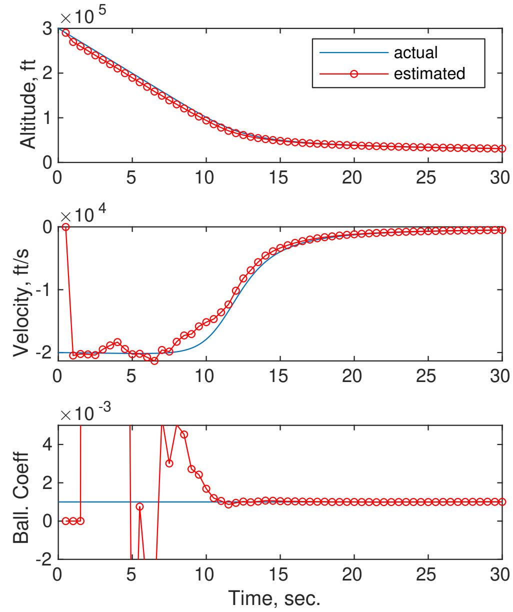
$$\hat{\boldsymbol{x}}_0^+ = [260,000 \quad 0 \quad 0]^{\mathrm{T}} \tag{38}$$

and covariance

$$\boldsymbol{P}_0^+ = \text{diag}([1,000,000 \quad 4,000,000 \quad 10]^{\mathrm{T}}) \tag{39}$$

The system is given in continuous-time but we can apply Euler integration to propagate it between time-steps so that it is in the required form. The figure below illustrates the input into the filter (range, blue data points on the left) and the resulting filter estimate (right). Notice that the amount of noise is relatively small compared to the scale of the measured range. The initial altitude guess was close to the actual value and quickly converges. Next, we see the velocity begins to track the actual values however only after 10 seconds once the ballistic coefficient is determined by the filter are all of the states being tracked accurately.


×10^5
ideal
noisy
Measured Range, ft
Time, sec.


×10^5
actual
estimated
Altitude, ft
×10^4
Velocity, ft/s
×10^-3
Ball. Coeff
Time, sec.

Appendix: Proof that UT approximates mean to third order.

The proof below is adapted from Dan Simon Sec. 14.2.1. First expand $\boldsymbol{z}(\boldsymbol{x})$ as a Taylor series around the mean $\bar{\boldsymbol{x}}$:

$$\boldsymbol{z} = \boldsymbol{g}(\bar{\boldsymbol{x}}) + D_{\tilde{\boldsymbol{x}}}\boldsymbol{g} + \frac{1}{2!}D^2_{\tilde{\boldsymbol{x}}}\boldsymbol{g} + \cdots \tag{40}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \bar{\boldsymbol{x}}$. Use Eq. 40 and Eq. 8 with $\boldsymbol{x} = \boldsymbol{x}^{(i)}$ to obtain:

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

Note the property of the sigma points: $\tilde{\boldsymbol{x}}^{(j)} = -\tilde{\boldsymbol{x}}^{(n+j)}$ for $j = 1, 2, \cdots, n$ implies that for all integers $k \geq 0$ (i.e. for all odd powers)

$$\begin{aligned}
\sum_{j=1}^{2n}\left(\tilde{\boldsymbol{x}}^{(j)}\right)^{2k+1} &= \sum_{j=1}^{n}\left(\tilde{\boldsymbol{x}}^{(j)}\right)^{2k+1} + \sum_{j=n+1}^{2n}\left(\tilde{\boldsymbol{x}}^{(j)}\right)^{2k+1} && (41)\\
&= \sum_{j=1}^{n}\left(\tilde{\boldsymbol{x}}^{(j)}\right)^{2k+1} + \sum_{j=1}^{n}\left(-\tilde{\boldsymbol{x}}^{(j)}\right)^{2k+1} && (42)\\
&= \boldsymbol{0} && (43)
\end{aligned}$$

Recall also the definition of our operator $D^k_{\tilde{\boldsymbol{x}}}\boldsymbol{f}$ used in the Taylor series: it is a sum of kth order partial derivatives for each element of $\boldsymbol{x}$ multiplied by the peturbation terms $\tilde{x}_i = x_i - \bar{x}_i$ and the function $\boldsymbol{f}(x)$ evaluated at the mean:

$$D^k_{\tilde{\boldsymbol{x}}}\boldsymbol{f} = \left(\sum_{i=1}^{n}\tilde{x}_i\frac{\partial}{\partial x_i}\right)^k \boldsymbol{f}(\boldsymbol{x})\Big|_{\bar{\boldsymbol{x}}} \tag{44}$$

Then it follows that for all integers $k \geq 0$:

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

In the last step we swapped the order of i and j summation and then used (43). Then we can simplify the expression for $\bar{\boldsymbol{z}}_u$ by eliminating all odd powers:

$$\begin{aligned}
\bar{\boldsymbol{z}}_u &= \boldsymbol{g}(\bar{\boldsymbol{x}}) + \frac{1}{2n}\sum_{i=1}^{2n}\left(\frac{1}{2!}D^2_{\tilde{\boldsymbol{x}}^{(i)}}\boldsymbol{g} + \frac{1}{4!}D^4_{\tilde{\boldsymbol{x}}^{(i)}}\boldsymbol{g} + \cdots\right) \\
&= \boldsymbol{g}(\bar{\boldsymbol{x}}) + \frac{1}{2n}\sum_{i=1}^{2n}\frac{1}{2!}D^2_{\tilde{\boldsymbol{x}}^{(i)}}\boldsymbol{g} + \frac{1}{2n}\sum_{i=1}^{2n}\left(\frac{1}{4!}D^4_{\tilde{\boldsymbol{x}}^{(i)}}\boldsymbol{g} + \frac{1}{6!}D^6_{\tilde{\boldsymbol{x}}^{(i)}}\boldsymbol{g} + \cdots\right)
\end{aligned}$$

Fact: A helpful fact for our derivation is that:

$$\begin{aligned}\left(\sum_{i=1}^{n} x_i\right)^2 &= (x_1 + x_2 + \cdots + x_n)(x_1 + x_2 + \cdots + x_n) \\ &= (x_1^2 + x_1x_2 + x_1x_n) + (\cdots) + (x_nx_1 + x_nx_2 + \cdots + x_n^2) \\ &= \sum_{i=1}^{n} x_i \sum_{j=1}^{n} x_j \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j\end{aligned}$$

Now consider the second term with the fact above:

$$\frac{1}{2n}\sum_{i=1}^{2n}\frac{1}{2!}D^2_{\tilde{\boldsymbol{x}}^{(i)}}\boldsymbol{g} = \frac{1}{2n}\sum_{k=1}^{2n}\frac{1}{2!}\left(\sum_{i=1}^{n}\tilde{x}_i^{(k)}\frac{\partial}{\partial x_i}\right)^2 \boldsymbol{g}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \tag{45}$$

$$= \frac{1}{2n}\sum_{k=1}^{2n}\frac{1}{2!}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i \partial x_j}\right)\boldsymbol{g}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \tag{46}$$

$$= \frac{1}{4n}\sum_{k=1}^{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i \partial x_j}\boldsymbol{g}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \tag{47}$$

Recall that $\tilde{\boldsymbol{x}}^k = -\tilde{\boldsymbol{x}}^{(n+k)}$ for $k = 1, 2, \cdots, n$. This implies that

$$\sum_{k=1}^{2n}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\right] = \sum_{k=1}^{2n}\left[\sum_{i=1}^{n}(\tilde{x}_i^{(k)})^2\right] \tag{48}$$

$$= \sum_{k=1}^{n}\left[\sum_{i=1}^{n}(\tilde{x}_i^{(k)})^2\right] + \sum_{k=1+n}^{2n}\left[\sum_{i=1}^{n}(\tilde{x}_i^{(k)})^2\right] \tag{49}$$

$$= \sum_{k=1}^{n}\left[\sum_{i=1}^{n}(\tilde{x}_i^{(k)})^2\right] + \sum_{k=1}^{n}\left[\sum_{i=1}^{n}(-\tilde{x}_i^{(k)})^2\right] \tag{50}$$

$$= 2\sum_{k=1}^{n}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_i^{(k)}\right] \tag{51}$$

From the definition of the vector $\tilde{\boldsymbol{x}}^{(k)}$ as a row of the matrix $(\sqrt{n\boldsymbol{P}})$, it follows that the i-th scalar element $\tilde{x}_i^{(k)}$ is an element of the matrix $(\sqrt{n\boldsymbol{P}})$ at the k-th row and i-th coloumn, we denote this as $(\sqrt{n\boldsymbol{P}})_{ki}$.

> **Aside: Index notation for matrix multiplication and transpose**
> Suppose $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times p}$ are two matrices that multiply to give $\boldsymbol{C} = \boldsymbol{AB}$. In index notation, an element of $\boldsymbol{C} \in \mathbb{R}^{m\times p}$ is given by $\boldsymbol{C}_{ij} = \sum_{k=1}^{n} \boldsymbol{A}_{ik}\boldsymbol{B}_{kj}$ for $i \in \{1, 2, \cdots, m\}$ and $j \in \{1, 2, \cdots, p\}$. Also if $\boldsymbol{D} = \boldsymbol{C}^T$ then $\boldsymbol{D}_{ij} = \boldsymbol{C}_{ji}$.

Then, using the property in the aside above, the summation

$$\sum_i^n \sum_j^n \tilde{x}_i^{(k)} \tilde{x}_j^{(k)} = \sum_i^n (\sqrt{n\boldsymbol{P}})_{ki} \sum_j^n (\sqrt{n\boldsymbol{P}})_{kj} = n\boldsymbol{P}_{ij} \tag{52}$$

gives an element $n\boldsymbol{P}_{ij}$ of the matrix $n\boldsymbol{P} = (\sqrt{n\boldsymbol{P}})^T(\sqrt{n\boldsymbol{P}})$. Now returning to (47) and using (51) and (52):

$$\frac{1}{2n}\sum_{i=1}^{2n} \frac{1}{2!} D^2_{\tilde{\boldsymbol{x}}^{(i)}} \boldsymbol{g} = \frac{1}{4n} \sum_{k=1}^{2n} \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{x}_i^{(k)} \tilde{x}_j^{(k)} \frac{\partial^2}{\partial x_i \partial x_j} \boldsymbol{g}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \tag{53}$$

$$= \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \tilde{x}_i^{(k)} \tilde{x}_j^{(k)} \frac{\partial^2}{\partial x_i \partial x_j} \boldsymbol{g}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \tag{54}$$

$$= \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{n} \left[\sum_{k=1}^{n} \tilde{x}_i^{(k)} \tilde{x}_j^{(k)}\right] \frac{\partial^2}{\partial x_i \partial x_j} \boldsymbol{g}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \tag{55}$$

$$= \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{n} n\boldsymbol{P}_{ij} \frac{\partial^2}{\partial x_i \partial x_j} \boldsymbol{g}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \tag{56}$$

$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \boldsymbol{P}_{ij} \frac{\partial^2}{\partial x_i \partial x_j} \boldsymbol{g}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \tag{57}$$

Recall from our lecture on transformations of random variables that if the p.d.f. of the random vector is symmetric then the *exact* value of the mean is a Taylor series with only even terms:

$$\bar{\boldsymbol{z}} = \boldsymbol{g}(\bar{\boldsymbol{x}}) + \frac{1}{2!}E[D^2_{\tilde{\boldsymbol{x}}}\boldsymbol{g}] + \frac{1}{4!}E[D^4_{\tilde{\boldsymbol{x}}}\boldsymbol{g}] + \cdots \tag{58}$$

Our derivation above was for the *approximate* value of the mean given the sigma points. If we can show that (57) from our approximation matches the 2nd term of (58) then we can claim the unscented transform is accurate up to third order (i.e., the first discrepencay will occur in the fourth-order term).

A similar procedure that compares the true to approximate covariance shows that the UT is accurate to third order.

References

[1] Dan Simon. *Optimal State Estimation: Kalman, H infinity, and Nonlinear Approaches*. John Wiley & Sons, 2006.

[2] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[3] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.