The Unscented Kalman Filter (Ch. 14)

The EKF requires us to linearize the nonlinear models f and h so that we can transform covariance matrices through them

→ Linearized models are only effective if the higher-order derivatives are small enough

Thought experiment: What would break a linear KF more?

1. Poor estimate of P_0^+, Q_k, or R_k
2. Incorrect F_k or H_k model

→ Model errors typically a bigger issue than covariance errors

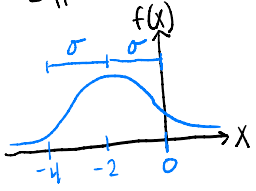
→ EKF linearization errors can cause performance/divergence

→ We want a new method for transforming mean/covariance through a nonlinear function by approximating the mean/covariance instead of the nonlinear function

→ This is the motivation for the unscented transform.

The Unscented Transform

Suppose we have an RV $X \sim N(-2, 4)$ w/ $Y = 4X + 10$ (all scalars)



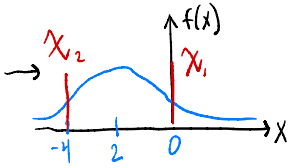
→ Y will be $\sim N(4 \cdot (-2) + 10, 4^2 \cdot 4) = N(2, 64)$

→ Easy enough since transformation is linear.

Different method: Pick two possible values of x called $\mathcal{X}_1$ and $\mathcal{X}_2$ with $\mathcal{X}_1 = \mu_x + \sigma_x = 0$ and $\mathcal{X}_2 = \mu_x - \sigma_x = -4$

Mean: $\bar{\mathcal{X}} = \frac{0-4}{2} = -2$ → True mean of X

Variance: $\sigma_{\mathcal{X}}^2 = \frac{(0+2)^2 + (-4+2)^2}{2} = \frac{4+4}{2} = 4$ → True variance of X



→ So, we have 2 discrete samples of X whose mean and variance match the true mean and variance of X

→ We can transform χ_1 & χ_2 through function to get χ_1^y and χ_2^y

→ $\chi_1^y = 4(0) + 10 = \underline{10}$ $\quad \chi_2^y = 4(-4) + 10 = \underline{-6}$

Mean value? → $\bar{\chi}^y = \frac{1}{N}\sum_{i=1}^{N}\chi_i^y = \frac{1}{2}\sum_{i=1}^{2}\chi_i^y = \frac{10-6}{2} = \underline{2}$ → Matches true mean of Y!

Variance value? → $\sigma_{\chi^y}^2 = \frac{1}{N}\sum_{i=1}^{N}(\chi_i^y - \bar{\chi}^y)^2 = \frac{1}{2}\sum_{i=1}^{N}(\chi_i^y - 2)^2$

$= \frac{(10-2)^2 + (-6-2)^2}{2} = \frac{64+64}{2} = \underline{64}$ → Matches true variance of Y!

→ Okay, great. What about a nonlinear transformation?

Let $X \sim N(-2, 4)$ and $Z = X^2$

→ $\mu_z = E\{z\} = E\{X^2\} = \sigma_x^2 + \mu_x^2 = 4 + (-2)^2 = \underline{8}$

→ $\sigma_z^2 = E\{z^2\} - \mu_z^2 = E\{X^4\} - \mu_z^2$ (will skip a few details)

$= \mu_x^4 + 6\mu_x^2\sigma_x^2 + 3\sigma_x^4 - \mu_z^2 = \underline{96}$

EKF uses linearization around μ_x:

→ $z = f(X) = X^2 \approx f(\mu_x) + \left.\frac{df}{dx}\right|_{X=\mu_x}(X - \mu_x) = f(\mu_x) + (2X)|_{X=\mu_x}(X - \mu_x)$

$= \mu_x^2 + 2\mu_x X - 2\mu_x^2 = \underline{-4X - 4}$

→ $\hat{\mu}_z = E\{-4X - 4\} = \underline{4}$ → Not the true mean!

→ $\sigma_z^2 = E\{(-4X - 4 - 4)^2\} = 16\underbrace{E\{X^2\}}_{=\sigma_x^2 + \mu_x^2 = 8} + 64E\{X\} + 64 = 128 - 128 + 64 = \underline{64}$

→ Not the true variance

Can we get better results with $\mathcal{X}_1^z$ and $\mathcal{X}_2^z$?

→ $\mathcal{X}_1^z = 0^2 = \underline{0}$ $\quad \mathcal{X}_2^z = (-4)^2 = \underline{16}$

→ $\hat{\mu}_z = \frac{0+16}{2} = \underline{8}$ → Matches true mean!

→ $\hat{\sigma}_z^2 = \frac{(0-8)^2 + (16-8)^2}{2} = \frac{64+64}{2} = \underline{64}$ → Does not match true variance (but same result as linearization)

This is a 1-D example of the unscented transform. The two points $\mathcal{X}_1$ & $\mathcal{X}_2$ at $\pm\sigma_x$ are called sigma points.

→ The unscented transform (UT) picks a set of sigma points whose mean/covariance match the mean/covariance we are trying to approximate

→ We then evaluate $f(\bar{\mathcal{X}}_i)$ for all the sigma points, and return the mean/covariance of the transformed points

→ The UT approximates mean up to third-order Taylor series exactly, while linearization only approximates first-order mean → Better mean accuracy

→ The UT approximates covariance up to third-order Taylor series exactly. Same as linearization (proof in book), but usually better approximation of fourth-order terms as well, which linearization omits entirely → Better or equal covariance accuracy

There are actually many ways to define the UT for $N \times 1$ RV $\bar{x}$.

Most simple way:

1. We have known $\bar{x} \sim (\bar{\mu}_x, P_x)$ and $\bar{y} = f(\bar{x})$
2. We can compute $2N$ sigma points:

$$\bar{\mathcal{X}}_i = \bar{\mu}_x + \left(\sqrt{N P_x}\right)_i^T \quad i = 1, \ldots, N$$
$$\bar{\mathcal{X}}_i = \bar{\mu}_x - \left(\sqrt{N P_x}\right)_{i-N}^T \quad i = N+1, \ldots, 2N$$

} Multiple ways to select sigma points! Will revisit later

→ $\sqrt{N P_x}$ is the matrix square root w/ $\sqrt{B} = A$ so that $B = A^T A$

$(\sqrt{N P_x})_i$ is the i^{th} row of $\sqrt{N P_x}$

3. Transform the sigma points:

$$\bar{\mathcal{X}}_i^f = f(\bar{\mathcal{X}}_i)$$

4. Find mean/covariance approximation:

$$\hat{\mu}_y = \frac{1}{2N} \sum_{i=1}^{2N} \bar{\mathcal{X}}_i^f$$

$$\hat{P}_y = \frac{1}{2N} \sum_{i=1}^{2N} (\bar{\mathcal{X}}_i^f - \hat{\mu}_y)(\bar{\mathcal{X}}_i^f - \hat{\mu}_y)^T$$

A note on matrix square roots:

- Multiple ways to define $A = \sqrt{B}$ — usually either $B = A^T A$ or $B = AA$
- Also multiple possible values of A
- Common way to compute A where $A^T A = B$ is Cholesky factorization
  - → Produces upper triangular A
  - → Use `chol(B)` in MATLAB
- The `sqrtm(B)` function in MATLAB gives A where $AA = B$

→ If you use A with form $AA = B$, you may compute $\bar{\mathcal{X}}_i$ using the columns of $\sqrt{N P_x}$ rather than the rows

How do we integrate the UT into the KF?

#1: State Function

The state function is used/approximated in several places in the EKF:

$$\rightarrow \hat{x}_k^- = f(\hat{x}_{k-1}^+, \bar{u}_{k-1})$$

$$P_k^- = F_{k-1} P_{k-1}^- F_{k-1}^T + Q_{k-1}$$

Assume linear noise for now

We can compute the mean $\hat{x}_k^-$ and covariance P_k^- instead using the UT:

$$\rightarrow \hat{x}_k^- = UT_m\{\hat{x}_{k-1}^+, P_{k-1}^+, \bar{u}_{k-1}, f\}$$

$$P_k^- = UT_c\{\hat{x}_{k-1}^+, P_{k-1}^+, f\} + Q_{k-1}$$

$\rightarrow UT_m\{\cdot\}$ computes mean using UT

$\rightarrow UT_c\{\cdot\}$ computes covariance using UT

The measurement function is also used/approximated multiple places in the EKF:

$$\rightarrow K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

Assume linear measurement noise

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(\bar{z}_k - h_k(\hat{x}_k^-, 0))$$

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T$$

$\rightarrow$ It's much less obvious how to apply the UT here.

$\rightarrow$ We can rearrange/reinterpret components of these equations to fit in the UT!

#2: Innovation

Start with the innovation, which we will call $\tilde{z}_k$

$\rightarrow$ linear KF: $\tilde{z}_k = \bar{z}_k - H_k \hat{x}_k^-$

$\rightarrow$ EKF: $\tilde{z}_k = \bar{z}_k - h(\hat{x}_k^-, 0)$

Both cases: $\tilde{z}_k = \bar{z}_k - E\{\bar{z}_k\} = \bar{z}_k - \hat{z}_k$

$H_k \hat{x}_k^-$ & $h(\hat{x}_k^-, 0)$ are estimates of $\bar{z}_k$!

$\rightarrow \tilde{z}_k$ is an RV w/ zero-mean and covariance S_k

Assuming $\bar{z}_k$ is independent of $\hat{x}_k^-$, then

$$S_k = \underbrace{cov\{\bar{z}_k\}}_{=R_k} + \underbrace{cov\{H_k \hat{x}_k^-\}}_{=P_{z,k}=H_k P_k^- H_k^T}$$

$$\rightarrow S_k = R_k + H_k P_k^- H_k^T$$

$\rightarrow$ This component of the Kalman gain is actually the covariance of the innovation!

With nonlinear $h \rightarrow$ converts RV w/ mean $\hat{x}_k^-$ and P_k^- into an RV w/ mean $\hat{z}_k$ and covariance $P_{z,k}$

→ We can use the UT again!

$$\rightarrow \hat{z}_k = UT_m\{\hat{x}_k^-, P_k^-, h\}$$
$$P_{z,k} = UT_c\{\hat{x}_k^-, P_k^-, h\}$$

→ Now, $\tilde{z} = \bar{z}_k - \hat{z}_k$ and $S_k = P_{z,k} + R_k$

Our new KF equations become:

$$K_k = P_k^- H_k^T S_k^{-1}$$
$$\hat{x}_k^+ = \hat{x}_k^- + K_k(\bar{z}_k - \hat{z}_k)$$
$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T$$

Next: What is $P_k^- H_k^T$?

#3: Cross-Covariance

Linear KF: $\hat{z}_k = H_k \hat{x}_k^-$

Suppose we want to know cross-covariance of $\hat{x}_k^-$ and $\hat{z}_k$, $P_{xz,k}$:

$$\rightarrow P_{xz,k} = E\{(\bar{x}_k - \hat{x}_k^-)(\bar{z}_k - \hat{z}_k)^T\} = E\{(\bar{x}_k - \hat{x}_k^-)(H_k\bar{x}_k - H_k\hat{x}_k^-)^T\}$$

$$= E\{\bar{x}_k\bar{x}_k^T H_k^T\} - E\{\hat{x}_k^-\bar{x}_k^T H_k^T\} - E\{\bar{x}_k\hat{x}_k^{-T} H_k^T\} + E\{\hat{x}_k^-\hat{x}_k^{-T} H_k^T\}$$

$$= \left(E\{\bar{x}_k\bar{x}_k^T\} - \hat{x}_k^- \underbrace{E\{\bar{x}_k^T\}}_{=\hat{x}_k^{-T}} - \underbrace{E\{\bar{x}_k\}}_{=\hat{x}_k^-}\hat{x}_k^{-T} + \hat{x}_k^-\hat{x}_k^{-T}\right) H_k^T$$

$$= \underbrace{\left(E\{\bar{x}_k\bar{x}_k^T\} - \hat{x}_k^-\hat{x}_k^{-T}\right)}_{=P_k^-} H_k^T = \boxed{P_k^- H_k^T} \rightarrow P_k^- H_k^T = P_{xz,k} \text{ is the cross-covariance of } \hat{x}_k^- \text{ and } \hat{z}_k!$$

→ We can use UT to approximate $P_{xz,k}$ by computing cross-covariance of transformed sigma points from calculation of P_k^- and S_k!

$$\rightarrow P_{xz,k} = UT_{xc}\{P_k^-, S_k\}$$

(we will define this modified UT in the UKF algorithm)

So now we have in our KF:

$$K_k = P_{x2,k} S_k^{-1}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(\bar{z}_k - \hat{z}_k)$$

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T$$

How to deal with these?

#4: Posterior Covariance

We can rewrite P_k^+ in expanded form:

$$P_k^+ = P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + \underbrace{K_k H_k P_k^- H_k^T K_k^T + K_k R_k K_k^T}_{= K_k(H_k P_k^- H_k^T + R_k) K_k^T}$$

$$= P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + K_k \underbrace{(H_k P_k^- H_k^T + R_k)}_{= S_k} K_k^T$$

$$\rightarrow K_k = P_k^- H_k^T S_k^{-1} \rightarrow K_k S_k = P_k^- H_k^T \xrightarrow{\text{transpose}} S_k K_k^T = H_k P_k^-$$

$$\rightarrow P_k^+ = P_k^- - K_k S_k K_k^T - K_k S_k K_k^T + K_k S_k K_k^T$$

$$= \boxed{P_k^- - K_k S_k K_k^T}$$

→ We have S_k, so no UT required!

Finally:

$$K_k = P_{x2,k} S_k^{-1}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(\bar{z}_k - \hat{z}_k)$$

$$P_k^+ = P_k^- - K_k S_k K_k$$

We can use the UT to compute:
$\hat{x}_k^-$ and P_k^-
$\hat{z}_k$ and S_k
and $P_{x2,k}$

The Unscented Kalman Filter

1. Define the model:

$$\bar{x}_k = f_k(\bar{x}_{k-1}, \bar{u}_{k-1}) + \bar{w}_k \quad \text{w/} \quad \bar{w}_k \sim (0, Q_k)$$

$$\bar{z}_k = h_k(\bar{x}_k) + \bar{v}_k \quad \text{w/} \quad \bar{v}_k \sim (0, R_k)$$

2. Initialize the UKF:

$$\hat{x}_0^+ = E\{\bar{x}_0\}$$

$$P_0^+ = E\{(\bar{x}_0 - \hat{x}_0^+)(\bar{x}_0 + \hat{x}_0^+)\}$$

3. Time-Update/Prediction:

a.) Let $\bar{x}_{i,k-1}^+$ be the i^{th} sigma point representing a distribution w/ mean $\hat{x}_{k-1}^+$ and covariance P_{k-1}^+

Calculate $2N$ sigma points:

$$\bar{x}_{i,k-1}^+ = \hat{x}_{k-1}^+ + \left(\sqrt{N P_{k-1}^+}\right)_i^T \qquad i = 1, \ldots, N$$

$$\bar{x}_{i,k-1}^+ = \hat{x}_{k-1}^+ - \left(\sqrt{N P_{k-1}^+}\right)_{i-N}^T \qquad i = N+1, \ldots, 2N$$

b.) Let $\bar{x}_{i,k-1}^f$ be the transformed version of $\bar{x}_{i,k-1}^+$ through $f(\cdot)$.

For all $i = 1, \ldots, 2N$, calculate:

$$\bar{x}_{i,k-1}^f = f(\bar{x}_{i,k-1}^+, \bar{u}_{k-1})$$

c.) Compute the prior state and covariance:

$$\hat{x}_k^- = \frac{1}{2N}\sum_{i=1}^{2N} \bar{x}_{i,k-1}^f$$

$$P_k^- = \frac{1}{2N}\sum_{i=1}^{2N} (\bar{x}_{i,k-1}^f - \hat{x}_k^-)(\bar{x}_{i,k-1}^f - \hat{x}_k^-)^T + Q_{k-1}$$

4. Measurement - Update:

a.) Let $\bar{x}_{i,k}^-$ be the i^{th} sigma point representing a distribution w/ mean $\hat{x}_k^-$ and covariance P_k^-

We can select $2N$ $\bar{x}_{i,k}^-$ two ways:

i.)
$$\bar{x}_{i,k}^- = \hat{x}_k^- + \left(\sqrt{NP_k^-}\right)_i^T \qquad i = 1, \dots, N$$

$$\bar{x}_{i,k}^- = \hat{x}_k^- - \left(\sqrt{NP_k^-}\right)_{i-N}^T \qquad i = N+1, \dots, 2N$$

Accurate but more expensive

ii.) Reuse: $\bar{x}_{i,k}^- = \bar{x}_{i,k-1}^f$

Less accurate but cheaper. In my experience, this way works just fine.

b.) Let $\bar{x}_{i,k}^h$ be the transformed version of $\bar{x}_{i,k}^-$ through $h(\cdot)$.

For all $i = 1, \dots, 2N$, calculate:

$$\bar{x}_{i,k}^h = h(\bar{x}_{i,k}^-)$$

These will be Mx1!

c.) Compute the predicted measurement $\hat{z}_k$ and the innovation covariance S_k:

$$\hat{z}_k = \frac{1}{2N} \sum_{i=1}^{2N} \bar{\mathcal{Y}}_{i,k}^{h}$$

$$S_k = \frac{1}{2N} \sum_{i=1}^{N} (\bar{\mathcal{Y}}_{i,k}^{h} - \hat{z}_k)(\bar{\mathcal{Y}}_{i,k}^{h} - \hat{z}_k)^T + R_k$$

d.) Compute the cross-covariance $P_{xz,k}$ between $\hat{x}_k^-$ and $\hat{z}_k$:

$$P_{xz,k} = \frac{1}{2N} \sum_{i=1}^{2N} (\bar{\mathcal{X}}_{i,k}^{-} - \hat{x}_k^-)(\bar{\mathcal{Y}}_{i,k}^{h} - \hat{z}_k)^T$$

e.) Compute the Kalman gain and the posterior state and covariance:

$$K_k = P_{xz,k} S_k^{-1}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(\bar{z}_k - \hat{z}_k)$$

$$P_k^+ = P_k^- - K_k S_k K_k^T$$

Summary: The UKF performs well with nonlinear systems and will virtually always perform better than the EKF with only a minor increase to computational complexity.

1. May still diverge if $\hat{x}_k^+$ ever drifts too far from true $\bar{x}_k$ since UT only a local approximation.
2. Several different UT forms depending on application.

Nonlinear Noise:

Suppose our model no longer has additive noise:

$$\bar{X}_k = f(\bar{X}_{k-1}, \bar{U}_{k-1}, \bar{W}_{k-1}) \quad \text{w/} \quad \bar{W}_k \sim (0, \tilde{Q}_k)$$

$$\bar{Z}_k = h(\bar{X}_{k-1}, \bar{V}_{k-1}) \quad \text{w/} \quad \bar{V}_k \sim (0, \tilde{R}_k)$$

We can include $\bar{W}_k$ and $\bar{V}_k$ in the UT by defining a new augmented state vector:

$$\bar{X}_k = \begin{bmatrix} \bar{X}_k \\ \bar{W}_k \\ \bar{V}_k \end{bmatrix}$$

→ The UT will generate noise realizations in the sigma points and allow tracking of the impact of nonlinear noise over time!

We run the UKF as normal, but set $Q_k = 0$ and $R_k = 0$.

No additive noise!

Initialize as:

$$\hat{X}_0^+ = \begin{bmatrix} E\{\bar{X}_0\} \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad P_0 = \begin{bmatrix} E\{(\bar{X}_0 - \bar{X}_0^+)(\bar{X}_0 - \bar{X}_0^+)^T\} & 0 & 0 \\ 0 & \tilde{Q}_k & 0 \\ 0 & 0 & \tilde{R}_k \end{bmatrix}$$

The Scaled UT

Problem: Look at the expression for sigma points:

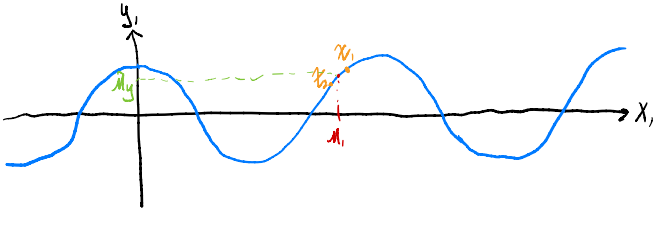
$$\chi_i = \bar{M}_x \pm \left(\sqrt{N P_x}\right)^T_{i \text{ or } i-L}$$

As N increases, the radius of the "hyperellipse" containing the sigma points also increases

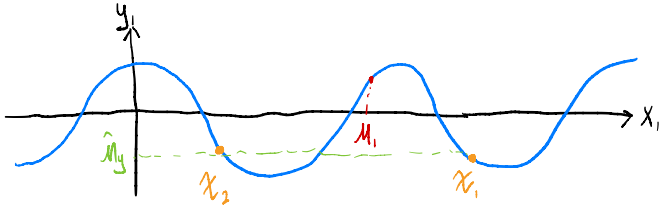
Example: Suppose that we have an N-dimensional vector space w/ $\bar{y} = f(\bar{x})$.

Suppose for y_1 and x_1, we have $y_1 = \cos(x_1)$ which is independent of all other dimensions

For N = 1, we select 2 sigma points around the mean of x_1 (M_1):



But for much larger N (say, N = 100), these points may be further from the mean:



Note: There would be 2N sigma points, not just 2!

→ For large N systems, sigma points may start to capture non-local behavior of the nonlinear function

Solution: The scaled UT was proposed to solve this problem. It has the added benefit of allowing us to represent some higher-order moments of the underlying distribution.

The scaled UT forces the sigma points to be closer to the mean based on some scaling factor α

→ By default, sigma points are spread from the mean w/ radius $\sqrt{N}$

→ In the scaled UT, the radius is $\alpha\sqrt{N}$

→ We can set α to anything $0 < \alpha \leq 1$, usually w/ $\alpha << 1$

→ Propagated sigma points must be weighted when forming mean/covariance to account for α

The Scaled Unscented Transform:

1. We have known $\bar{x} \sim (\bar{\mu}_x, P_x)$ and $\bar{y} = f(\bar{x})$
2. We define the following hyperparameters:
   - α: Controls spread of sigma points. Set within $0 < \alpha \leq 1$. A useful default value is $\alpha = 10^{-3}$.
   - κ: Secondary parameter also controlling sigma point spread. Usually, $\kappa = 0$.
   - β: Used to account for additional knowledge of the prior distribution of $\bar{x}$. If $\bar{x}$ is Gaussian, then optimal $\beta = 2$.
   - λ: We set a fixed $\boxed{\lambda = \alpha^2(N + \kappa) - N}$
3. We define $2N+1$ weights W_i^m and W_i^c for $i = 0, \ldots, 2N$.

   Technically, we can pick any real-valued W_i^m so long as $\boxed{\sum_i W_i^m = 1}$

   → W_i^m can be negative! In fact, usually W_0^m is negative, and with $|W_i^m|$ being very large!

Typically, we define W_i^m as follows:

$$W_0^m = \frac{\lambda}{N+\lambda} \quad \text{and} \quad W_i^m = \frac{1}{2(N+\lambda)} \quad i = 1, \dots, 2N$$

→ These weights are used in the mean calculation.

We also define weights W_i^c as follows:

$$W_0^c = W_0^m + 1 - \alpha^2 + \beta \quad \text{and} \quad W_i^c = W_i^m \quad i = 1, \dots, 2N$$

→ These weights are used in the covariance calculation.

4. We compute $2N+1$ sigma points:

$$\bar{\mathcal{X}}_0 = \bar{\mu}_x$$

$$\bar{\mathcal{X}}_i = \bar{\mu}_x + \left(\sqrt{(N+\lambda)P_x}\right)_i^T \quad i = 1, \dots, N$$

$$\bar{\mathcal{X}}_i = \bar{\mu}_x - \left(\sqrt{(N+\lambda)P_x}\right)_{i-N}^T \quad i = N+1, \dots, 2N$$

5. Transform the sigma points:

$$\bar{\mathcal{Y}}_i^f = f(\bar{\mathcal{X}}_i)$$

6. Find mean/covariance:

$$\hat{\mu}_y = \sum_{i=0}^{2N} W_i^m \bar{\mathcal{Y}}_i^f$$

$$\hat{P}_y = \sum_{i=0}^{2N} W_i^c \left(\bar{\mathcal{Y}}_i^f - \hat{\mu}_y\right)\left(\bar{\mathcal{Y}}_i^f - \hat{\mu}_y\right)^T$$

Notice: No $\frac{1}{2N+1}$ in front of the sum! This is included implicitly in W_i^m / W_i^c.

In the UKF, we can use the scaled UT in exactly the same way as the original UT to compute $\hat{x}_k^-$, P_k^-, $\hat{z}_k$, S_k, and $P_{x_k z_k}$ with appropriate modifications to their respective equations.

Remark: If $\alpha = 1$ and $\kappa = 0$, the scaled UT is identical to the original UT!

The Scaled Unscented Kalman Filter

1. Define the model:

$$\bar{x}_k = f_k(\bar{x}_{k-1}, \bar{u}_{k-1}) + \bar{w}_k \quad \text{w/} \quad \bar{w}_k \sim (0, Q_k)$$

$$\bar{z}_k = h_k(\bar{x}_k) + \bar{v}_k \quad \text{w/} \quad \bar{v}_k \sim (0, R_k)$$

2. Initialize the UKF:

$$\hat{x}_0^+ = E\{\bar{x}_0\}$$

$$P_0^+ = E\{(\bar{x}_0 - \hat{x}_0^+)(\bar{x}_0 + \hat{x}_0^+)\}$$

3. Define hyperparameters:

$\alpha \rightarrow 0 < \alpha \leq 1$ (default: $\alpha = 10^{-3}$)

$\kappa \rightarrow \kappa \geq 0$ (default: $\kappa = 0$)

$\beta \rightarrow$ If Gaussian, $\beta = 2$

$\lambda \rightarrow \lambda = \alpha^2(N+\kappa) - N$

Set $W_0^m = \frac{\lambda}{N+\lambda}$ and $W_i^m = \frac{1}{2(N+\lambda)}$ $i = 1, \ldots, 2N$

Set $W_0^c = W_0^m + 1 - \alpha^2 + \beta$ and $W_i^c = W_i^m$ $i = 1, \ldots, 2N$

4. Time-Update/Prediction:

a.) Let $\bar{\chi}_{i,k-1}^+$ be the i^{th} sigma point representing a distribution w/ mean $\hat{x}_{k-1}^+$ and covariance P_{k-1}^+

Calculate $2N+1$ sigma points:

$$\bar{\chi}_{i,k-1}^+ = \hat{x}_{k-1}^+ + \left(\sqrt{(N+\lambda)P_{k-1}^+}\right)_i^T \quad i = 1, \ldots, N$$

$$\bar{\chi}_{i,k-1}^+ = \hat{x}_{k-1}^+ - \left(\sqrt{(N+\lambda)P_{k-1}^+}\right)_{i-N}^T \quad i = N+1, \ldots, 2N$$

and $\bar{\chi}_{0,k-1}^+ = \hat{x}_{k-1}^+$

b.) Let $\bar{\mathcal{X}}^{f}_{i,k-1}$ be the transformed version of $\bar{\mathcal{X}}^{+}_{i,k-1}$ through $f(\cdot)$.

For all $i = 0, \dots, 2N$, calculate:

$$\bar{\mathcal{X}}^{f}_{i,k-1} = f\left(\bar{\mathcal{X}}^{+}_{i,k-1}, \bar{u}_{k-1}\right)$$

c.) Compute the prior state and covariance:

$$\hat{x}_k^- = \sum_{i=0}^{2N} W_i^m \bar{\mathcal{X}}^{f}_{i,k-1}$$

$$P_k^- = \sum_{i=0}^{2N} W_i^c \left(\bar{\mathcal{X}}^{f}_{i,k-1} - \hat{x}_k^-\right)\left(\bar{\mathcal{X}}^{f}_{i,k-1} - \hat{x}_k^-\right)^T + Q_{k-1}$$

4. Measurement - Update:

a.) Let $\bar{\mathcal{X}}^{-}_{i,k}$ be the i^{th} sigma point representing a distribution w/ mean $\hat{x}_k^-$ and covariance P_k^-

We can select $2N+1$ $\bar{\mathcal{X}}^{-}_{i,k}$ two ways:

i.)
$$\bar{\mathcal{X}}^{-}_{i,k} = \hat{x}_k^-$$

$$\bar{\mathcal{X}}^{-}_{i,k} = \hat{x}_k^- + \left(\sqrt{(N+\lambda) P_k^-}\right)_i^T \quad i = 1, \dots, N$$

$$\bar{\mathcal{X}}^{-}_{i,k} = \hat{x}_k^- - \left(\sqrt{(N+\lambda) P_k^-}\right)_{i-N}^T \quad i = N+1, \dots, 2N$$

Accurate but more expensive

ii.) Reuse: $\bar{\mathcal{X}}^{-}_{i,k} = \bar{\mathcal{X}}^{f}_{i,k-1}$

Less accurate but cheaper. In my experience, this way works just fine.

b.) Let $\bar{\mathcal{X}}_{i,k}^{h}$ be the transformed version of $\bar{\mathcal{X}}_{i,k}^{-}$ through $h(\cdot)$.

For all $i = 0, \ldots, 2N$ calculate:

$$\bar{\mathcal{X}}_{i,k}^{h} = h\left(\bar{\mathcal{X}}_{i,k}^{-}\right)$$

These will be Mx1!

c.) Compute the predicted measurement $\hat{z}_k$ and the innovation covariance S_k:

$$\hat{z}_k = \sum_{i=0}^{2N} W_i^m \bar{\mathcal{X}}_{i,k}^{h}$$

$$S_k = \sum_{i=0}^{N} W_i^c \left(\bar{\mathcal{X}}_{i,k}^{h} - \hat{z}_k\right)\left(\bar{\mathcal{X}}_{i,k}^{h} - \hat{z}_k\right)^T + R_k$$

d.) Compute the cross-covariance $P_{xz,k}$ between $\hat{x}_k^-$ and $\hat{z}_k$:

$$P_{xz,k} = \sum_{i=0}^{2N} W_i^c \left(\bar{\mathcal{X}}_{i,k}^{-} - \hat{x}_k^-\right)\left(\bar{\mathcal{X}}_{i,k}^{h} - \hat{z}_k\right)^T$$

e.) Compute the Kalman gain and the posterior state and covariance:

$$K_k = P_{xz,k} S_k^{-1}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k\left(\bar{z}_k - \hat{z}_k\right)$$

$$P_k^+ = P_k^- - K_k S_k K_k^T$$

Summary: Scaled UKF is useful for large N, but has no drawbacks other than requiring selection of α, K, and β. In my opinion, this is the form of the UKF that should be implemented almost always.