MODULE 2 LESSON 6

AN ALTERNATIVE TO THE EKF: THE UNSCENTED

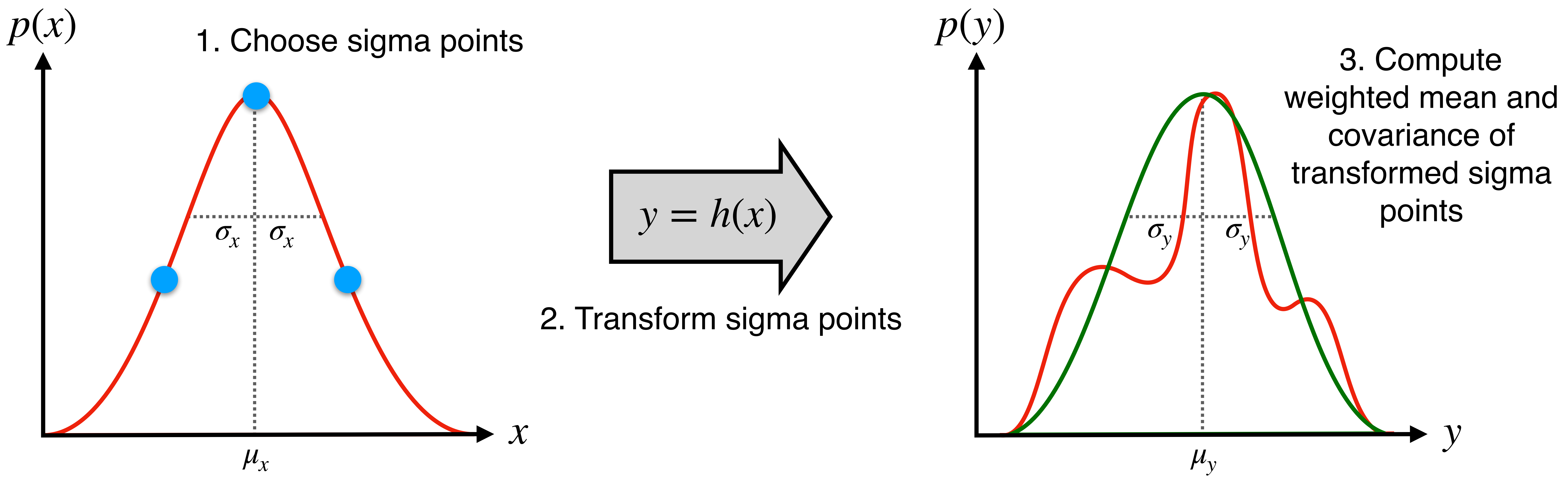
The Unscented Kalman Filter (UKF)

By the end of this video, you will be able to…

- Use the Unscented Transform to pass a probability distribution through a nonlinear function
- Describe how the Unscented Kalman Filter (UKF) uses the Unscented Transform in the prediction and correction steps
- Explain the advantages of the UKF over the EKF
- Apply the UKF to a simple nonlinear tracking problem

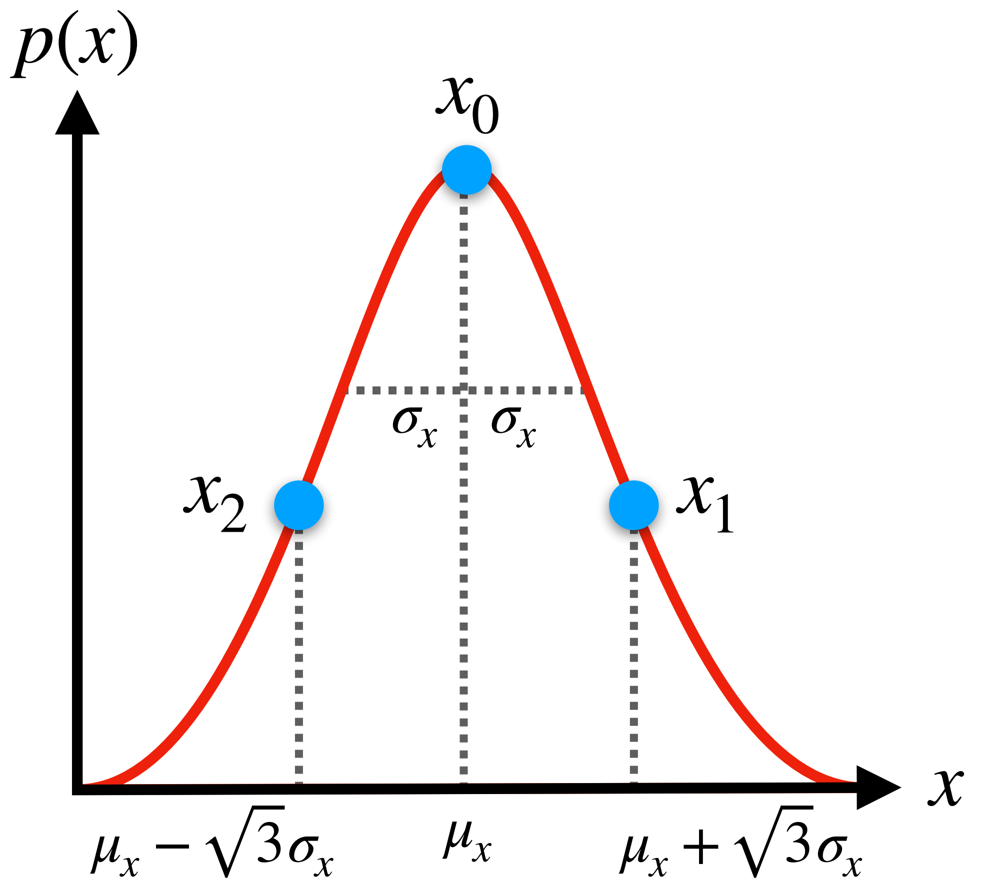
The Unscented Transform

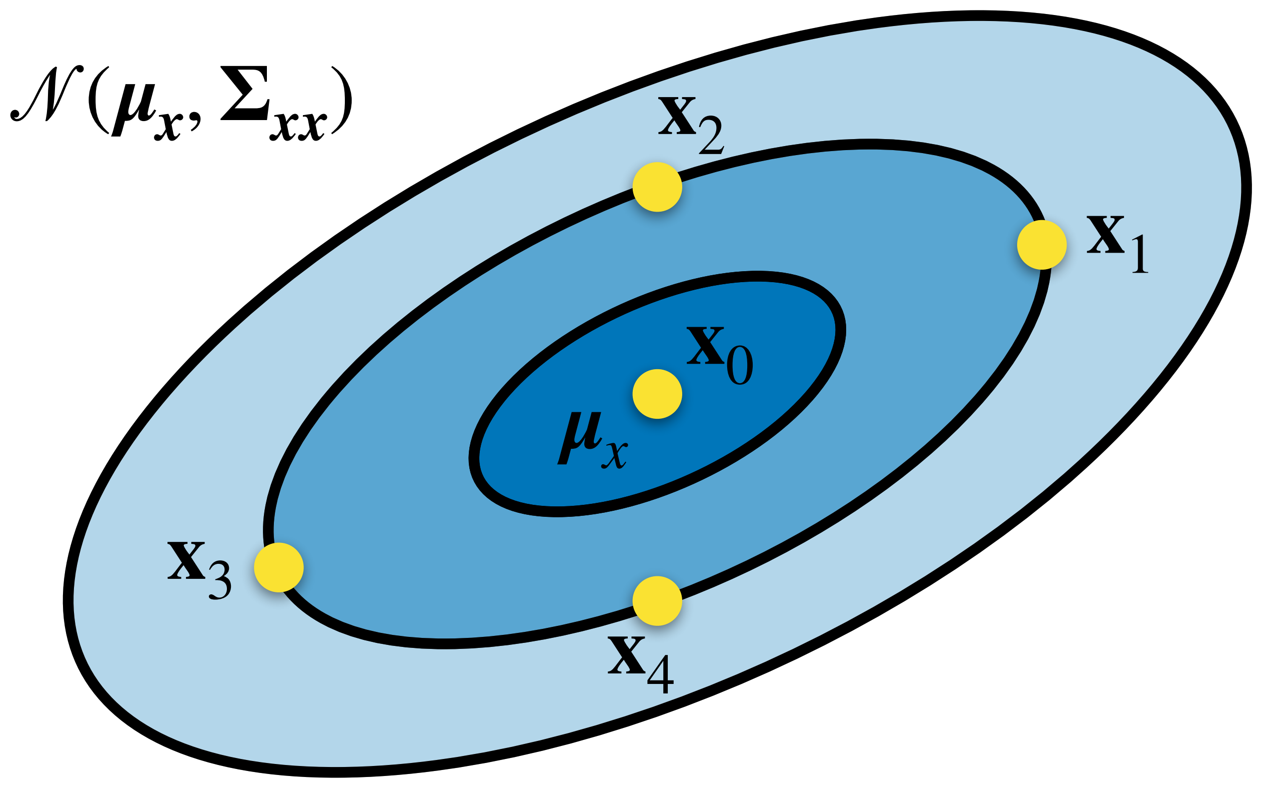
"It is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function" — S. Julier, J. Uhlmann, and H. Durrant-Whyte (2000)



The Unscented Transform | Choosing Sigma Points

For an N-dimensional PDF $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx})$, we need $2N + 1$ sigma points:





1. <u>Compute the Cholesky Decomposition of the covariance matrix</u>

   $\mathbf{L}\mathbf{L}^T = \boldsymbol{\Sigma}_{xx}$ ($\mathbf{L}$ lower triangular)

2. <u>Calculate the sigma points</u>

$$\mathbf{x}_0 = \boldsymbol{\mu}_x$$

$$\mathbf{x}_i = \boldsymbol{\mu}_x + \sqrt{N+\kappa}\ \mathrm{col}_i \mathbf{L} \qquad i = 1, \ \ldots, N$$

$$\mathbf{x}_{i+N} = \boldsymbol{\mu}_x - \sqrt{N+\kappa}\ \mathrm{col}_i \mathbf{L} \qquad i = 1, \ \ldots, N$$

$\kappa = 3 - N$
for Gaussian PDFs

The Unscented Transform | Transforming and Recombining

Next we pass each of our $2N + 1$ sigma points through the nonlinear function $\mathbf{h}(\mathbf{x})$

$$\mathbf{y}_i = \mathbf{h}(\mathbf{x}_i) \qquad i = 0, \ldots, 2N$$

And finally compute the mean and covariance of the output PDF

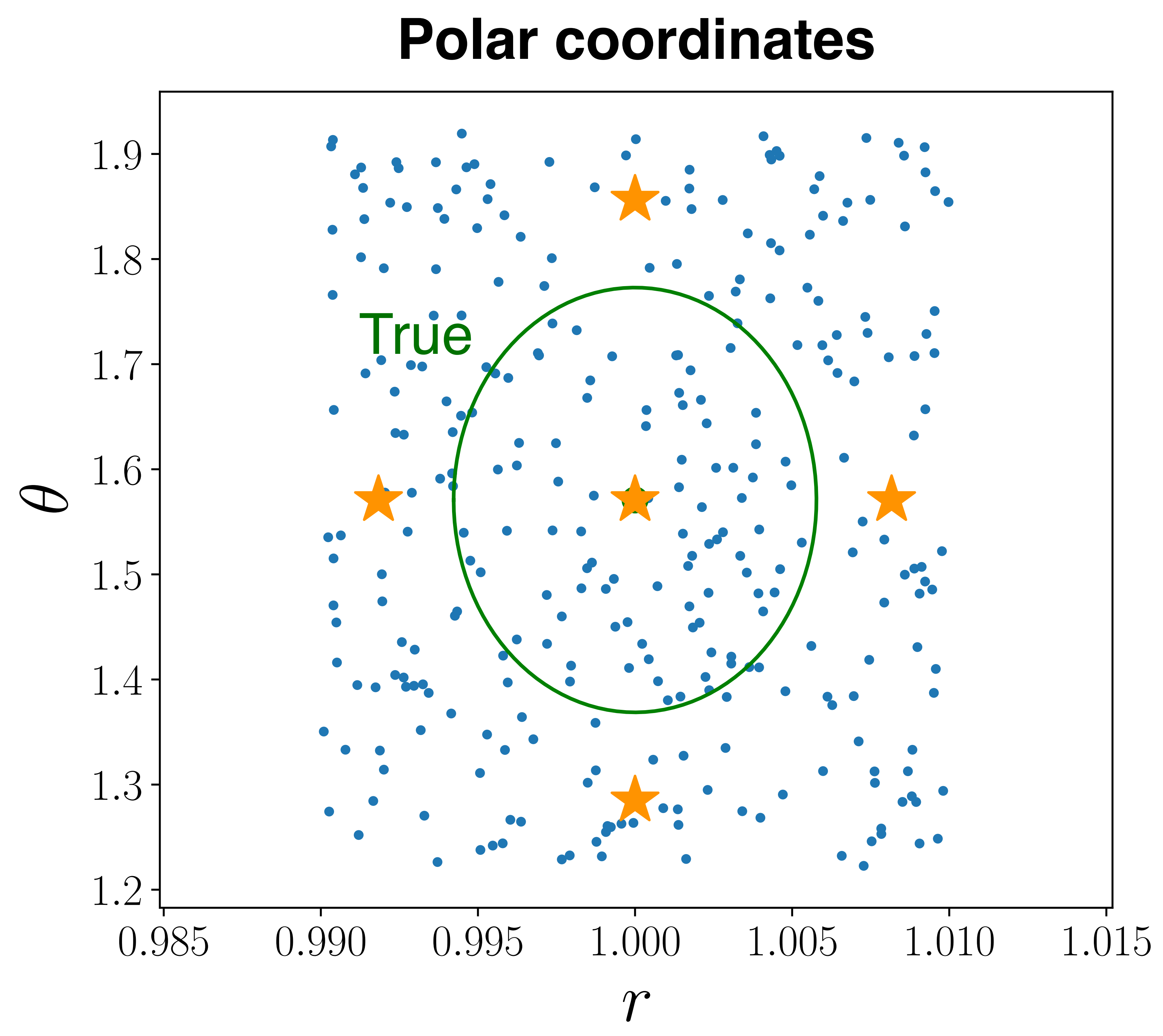
Mean: $$\boldsymbol{\mu}_y = \sum_{i=0}^{2N} \alpha_i \mathbf{y}_i$$

Covariance: $$\boldsymbol{\Sigma}_{yy} = \sum_{i=0}^{2N} \alpha_i \left(\mathbf{y}_i - \boldsymbol{\mu}_y\right)\left(\mathbf{y}_i - \boldsymbol{\mu}_y\right)^T$$

Weights: $$\alpha_i = \begin{cases} \dfrac{\kappa}{N+\kappa} & i = 0 \\ \dfrac{1}{2}\dfrac{1}{N+\kappa} & \text{otherwise} \end{cases}$$

The Unscented Transform vs. Linearization

Let's revisit our nonlinear transformation example from the previous video.

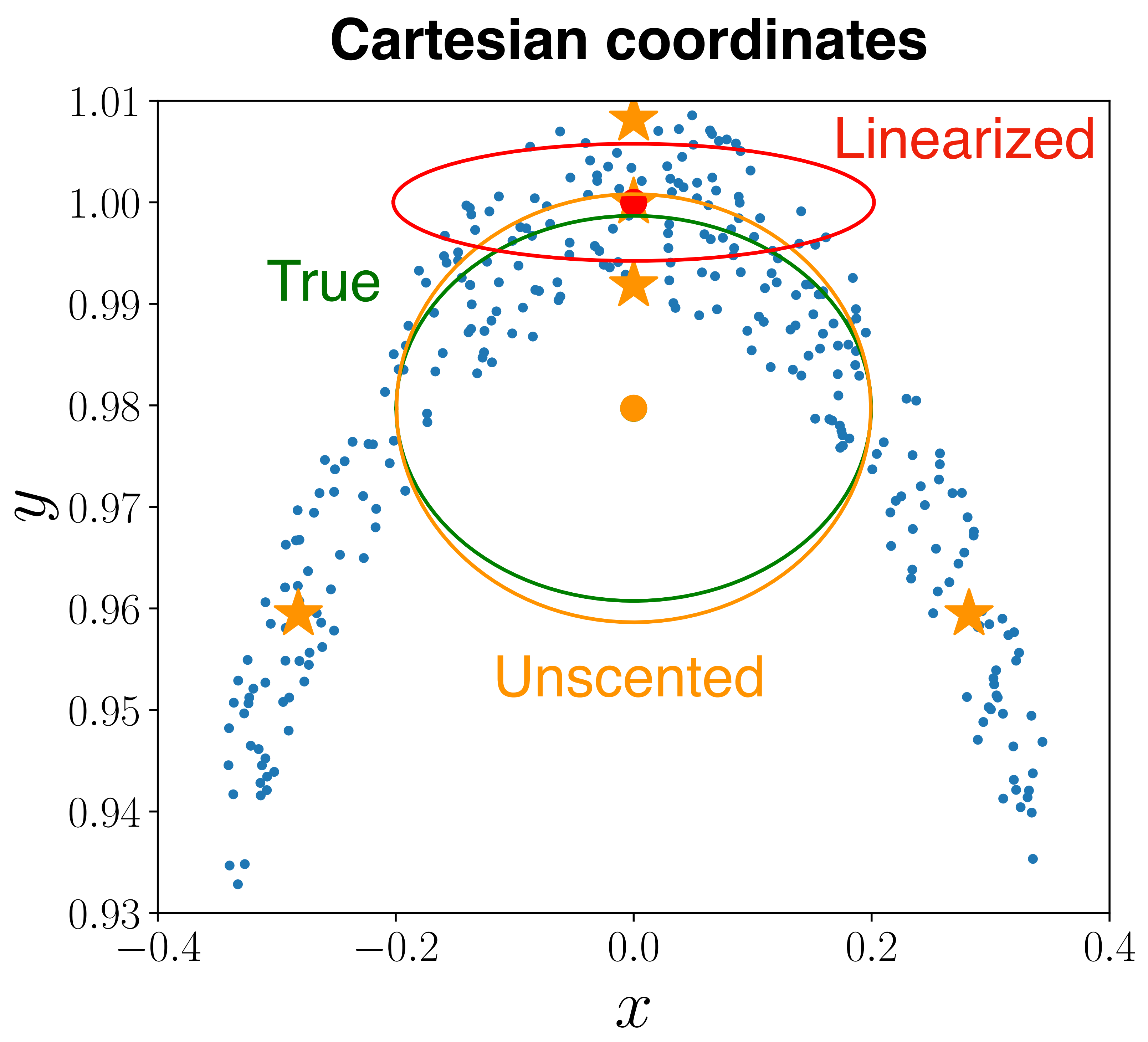


Nonlinear transformation

$$x = r\cos\theta$$
$$y = r\sin\theta$$

Unscented transform

★ sigma points



The Unscented Transform gives a much better approximation for similar work!

The Unscented Kalman Filter (UKF)

We can easily use the Unscented Transform in our Kalman Filtering framework with nonlinear models:

Nonlinear motion model

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})$$

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$$

Nonlinear measurement model

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k)$$

$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$$

Instead of approximating the system equations by linearizing, we will calculate sigma points and use the Unscented Transform to approximate the PDFs directly!

UKF | Prediction step

To propagate the state from time ($k - 1$) to time k, apply the Unscented Transform using the current best guess for the mean and covariance

1. Compute sigma points

$$\begin{aligned}
\hat{\mathbf{L}}_{k-1}\hat{\mathbf{L}}_{k-1}^T &= \hat{\mathbf{P}}_{k-1} \\
\hat{\mathbf{x}}_{k-1}^{(0)} &= \hat{\mathbf{x}}_{k-1} \\
\hat{\mathbf{x}}_{k-1}^{(i)} &= \hat{\mathbf{x}}_{k-1} + \sqrt{N+\kappa}\ \mathrm{col}_i \hat{\mathbf{L}}_{k-1} \qquad i = 1 \ldots N \\
\hat{\mathbf{x}}_{k-1}^{(i+N)} &= \hat{\mathbf{x}}_{k-1} - \sqrt{N+\kappa}\ \mathrm{col}_i \hat{\mathbf{L}}_{k-1} \qquad i = 1 \ldots N
\end{aligned}$$

2. Propagate sigma points

$$\check{\mathbf{x}}_k^{(i)} = \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{k-1}, \mathbf{0}) \qquad i = 0 \ldots 2N$$

3. Compute predicted mean and covariance

$$\alpha^{(i)} = \begin{cases} \dfrac{\kappa}{N+\kappa} & i = 0 \\ \dfrac{1}{2}\dfrac{1}{N+\kappa} & \text{otherwise} \end{cases}$$

$$\check{\mathbf{x}}_k = \sum_{i=0}^{2N} \alpha^{(i)} \check{\mathbf{x}}_k^{(i)}$$

$$\check{\mathbf{P}}_k = \sum_{i=0}^{2N} \alpha^{(i)} \left(\check{\mathbf{x}}_k^{(i)} - \check{\mathbf{x}}_k\right)\left(\check{\mathbf{x}}_k^{(i)} - \check{\mathbf{x}}_k\right)^T + \mathbf{Q}_{k-1}$$

Additive process noise!

UKF | Correction step

To correct the state estimate using measurements at time *k*, use the nonlinear measurement model and the sigma points from the prediction step to predict the measurements

1. Predict measurements from (redrawn) propagated sigma points

$$\hat{\mathbf{y}}_k^{(i)} = \mathbf{h}_k(\check{\mathbf{x}}_k^{(i)}, \mathbf{0}) \qquad i = 0, \ldots, 2N$$

With process noise considered!

2. Estimate mean and covariance of predicted measurements

$$\hat{\mathbf{y}}_k = \sum_{i=0}^{2N} \alpha^{(i)} \hat{\mathbf{y}}_k^{(i)}$$

$$\mathbf{P}_y = \sum_{i=0}^{2N} \alpha^{(i)} \left(\hat{\mathbf{y}}_k^{(i)} - \hat{\mathbf{y}}_k\right) \left(\hat{\mathbf{y}}_k^{(i)} - \hat{\mathbf{y}}_k\right)^T + \mathbf{R}_k$$

Additive measurement noise!

3. Compute cross-covariance and Kalman gain

$$\mathbf{P}_{xy} = \sum_{i=0}^{2N} \alpha^{(i)} \left(\check{\mathbf{x}}_k^{(i)} - \check{\mathbf{x}}_k\right) \left(\hat{\mathbf{y}}_k^{(i)} - \hat{\mathbf{y}}_k\right)^T$$

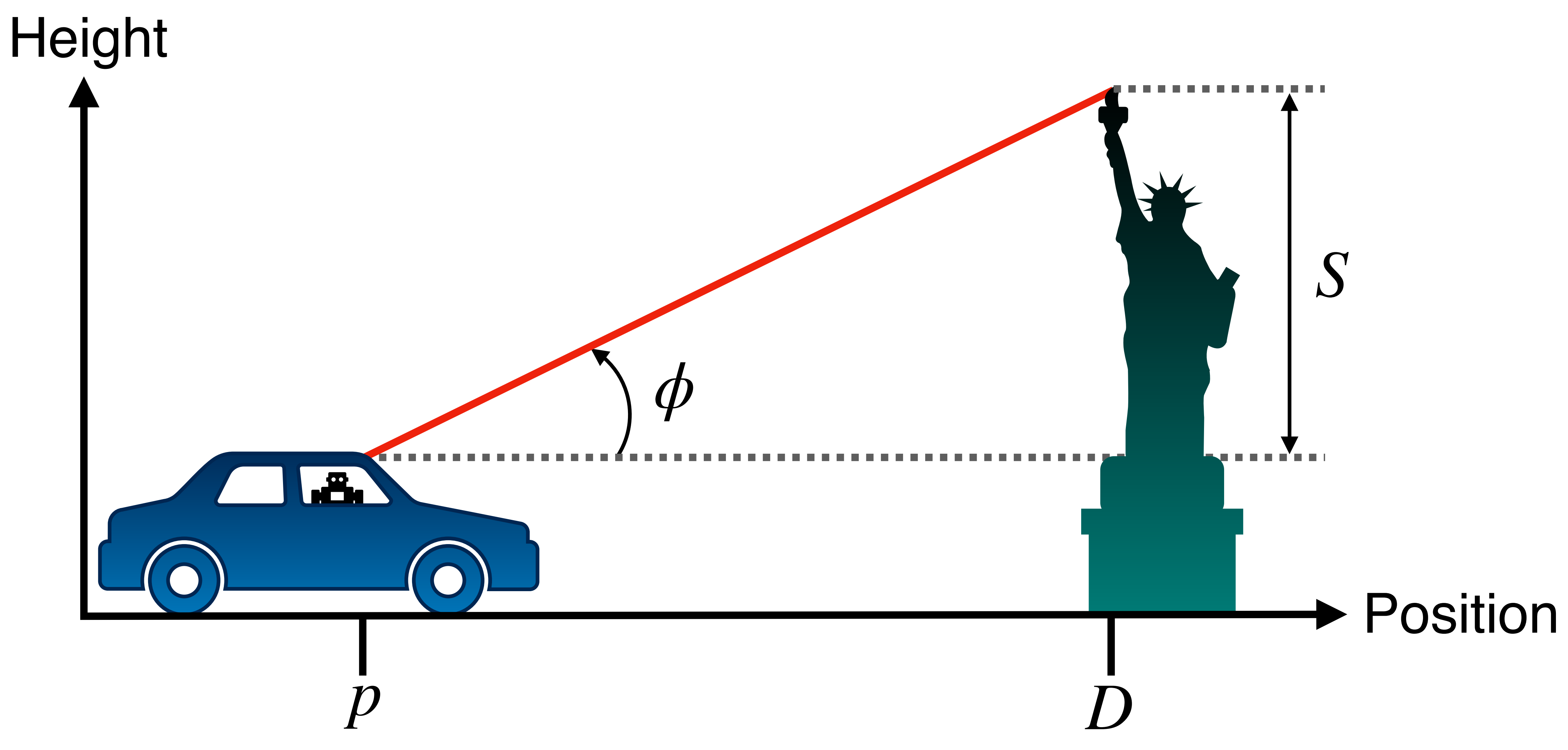
$$\mathbf{K}_k = \mathbf{P}_{xy} \mathbf{P}_y^{-1}$$

4. Compute corrected mean and covariance

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k \left(\mathbf{y}_k - \hat{\mathbf{y}}_k\right)$$

$$\hat{\mathbf{P}}_k = \check{\mathbf{P}}_k - \mathbf{K}_k \mathbf{P}_y \mathbf{K}_k^T$$

UKF | Short example



$$\mathbf{x} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix} \qquad \mathbf{u} = \ddot{p}$$

S and *D* are known in advance

Motion/Process model

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})$$

$$= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$

Landmark measurement model

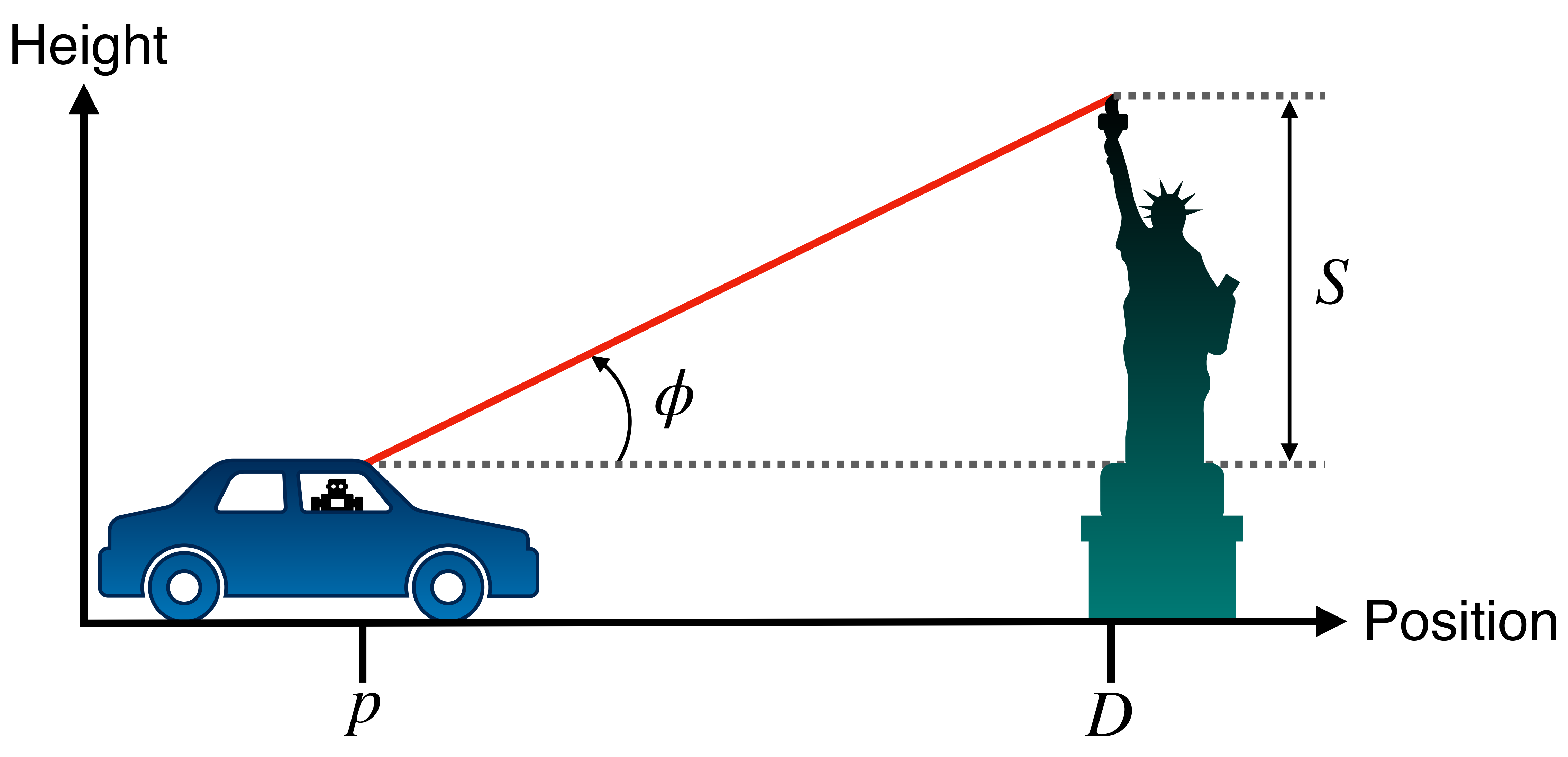
$$y_k = \phi_k = h(p_k, v_k)$$

$$= \tan^{-1}\left(\frac{S}{D - p_k}\right) + v_k$$

Noise densities

$$v_k \sim \mathcal{N}(0,\ 0.01) \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, (0.1)\mathbf{1}_{2\times 2})$$

UKF | Short example



Data

$$\hat{\mathbf{x}}_0 \sim \mathcal{N}\left(\begin{bmatrix}0\\5\end{bmatrix}, \begin{bmatrix}0.01 & 0\\0 & 1\end{bmatrix}\right)$$

$$\Delta t = 0.5\text{s}$$

$u_0 = -2\ [m/s^2]$ $\qquad$ $y_1 = 30\ [\text{deg}]$

$S = 20\ [m]$ $\qquad$ $D = 40\ [m]$

Using the Unscented Kalman Filter equations, what is our updated position?

$$\hat{p}_1$$

UKF | Short Example Solution

Prediction

$$N = 2, \quad \kappa = 3 - N = 1$$

$$\hat{\mathbf{L}}_0 \hat{\mathbf{L}}_0^T = \hat{\mathbf{P}}_0$$

$$\begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix}^T = \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\hat{\mathbf{x}}_0^{(0)} = \hat{\mathbf{x}}_0$$

$$\hat{\mathbf{x}}_0^{(i)} = \hat{\mathbf{x}}_0 + \sqrt{N+\kappa}\ \mathrm{col}_i \hat{\mathbf{L}}_0 \qquad i = 1\ \ldots, N$$

$$\hat{\mathbf{x}}_0^{(i+N)} = \hat{\mathbf{x}}_0 - \sqrt{N+\kappa}\ \mathrm{col}_i \hat{\mathbf{L}}_0 \qquad i = 1,\ \ldots, N$$

$$\hat{\mathbf{x}}_0^{(0)} = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$$

$$\hat{\mathbf{x}}_0^{(1)} = \begin{bmatrix} 0 \\ 5 \end{bmatrix} + \sqrt{3} \begin{bmatrix} 0.1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 5 \end{bmatrix}$$

$$\hat{\mathbf{x}}_0^{(2)} = \begin{bmatrix} 0 \\ 5 \end{bmatrix} + \sqrt{3} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 6.7 \end{bmatrix}$$

$$\hat{\mathbf{x}}_0^{(3)} = \begin{bmatrix} 0 \\ 5 \end{bmatrix} - \sqrt{3} \begin{bmatrix} 0.1 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.2 \\ 5 \end{bmatrix}$$

$$\hat{\mathbf{x}}_0^{(4)} = \begin{bmatrix} 0 \\ 5 \end{bmatrix} - \sqrt{3} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3.3 \end{bmatrix}$$

UKF | Short Example Solution

Prediction

$$\check{\mathbf{x}}_1^{(i)} = \mathbf{f}_0(\hat{\mathbf{x}}_0^{(i)}, \mathbf{u}_0, \mathbf{0}) \qquad i = 0, \ldots, 2N$$

$$\check{\mathbf{x}}_1^{(0)} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} (-2) = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}$$

$$\check{\mathbf{x}}_1^{(1)} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.2 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} (-2) = \begin{bmatrix} 2.7 \\ 4 \end{bmatrix}$$

$$\check{\mathbf{x}}_1^{(2)} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 6.7 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} (-2) = \begin{bmatrix} 3.4 \\ 5.7 \end{bmatrix}$$

$$\check{\mathbf{x}}_1^{(3)} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -0.2 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} (-2) = \begin{bmatrix} 2.3 \\ 4 \end{bmatrix}$$

$$\check{\mathbf{x}}_1^{(4)} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 3.3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} (-2) = \begin{bmatrix} 1.6 \\ 2.3 \end{bmatrix}$$

$$\alpha^{(i)} = \begin{cases} \dfrac{\kappa}{N+\kappa} = \dfrac{1}{2+1} = \dfrac{1}{3} & i = 0 \\ \dfrac{1}{2}\dfrac{1}{N+\kappa} = \dfrac{1}{2}\dfrac{1}{2+1} = \dfrac{1}{6} & \text{otherwise} \end{cases}$$

$$\check{\mathbf{x}}_k = \sum_{i=0}^{2N} \alpha^{(i)} \check{\mathbf{x}}_k^{(i)}$$

$$= \frac{1}{3}\begin{bmatrix} 2.5 \\ 4 \end{bmatrix} + \frac{1}{6}\begin{bmatrix} 2.7 \\ 4 \end{bmatrix} + \frac{1}{6}\begin{bmatrix} 3.4 \\ 5.7 \end{bmatrix} + \frac{1}{6}\begin{bmatrix} 2.3 \\ 4 \end{bmatrix} + \frac{1}{6}\begin{bmatrix} 1.6 \\ 2.3 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}$$

UKF | Short Example Solution

Prediction

$$\check{\mathbf{P}}_k = \sum_{i=0}^{2N} \alpha^{(i)} \left(\check{\mathbf{x}}_k^{(i)} - \check{\mathbf{x}}_k\right)\left(\check{\mathbf{x}}_k^{(i)} - \check{\mathbf{x}}_k\right)^T + \mathbf{Q}_{k-1}$$

$$= \frac{1}{3}\left(\begin{bmatrix}2.5\\4\end{bmatrix} - \begin{bmatrix}2.5\\4\end{bmatrix}\right)\left(\begin{bmatrix}2.5\\4\end{bmatrix} - \begin{bmatrix}2.5\\4\end{bmatrix}\right)^T +$$

$$\frac{1}{6}\left(\begin{bmatrix}2.7\\4\end{bmatrix} - \begin{bmatrix}2.5\\4\end{bmatrix}\right)\left(\begin{bmatrix}2.7\\4\end{bmatrix} - \begin{bmatrix}2.5\\4\end{bmatrix}\right)^T + \frac{1}{6}\left(\begin{bmatrix}3.4\\5.7\end{bmatrix} - \begin{bmatrix}2.5\\4\end{bmatrix}\right)\left(\begin{bmatrix}3.4\\5.7\end{bmatrix} - \begin{bmatrix}2.5\\4\end{bmatrix}\right)^T +$$

$$\frac{1}{6}\left(\begin{bmatrix}2.3\\4\end{bmatrix} - \begin{bmatrix}2.5\\4\end{bmatrix}\right)\left(\begin{bmatrix}2.3\\4\end{bmatrix} - \begin{bmatrix}2.5\\4\end{bmatrix}\right)^T + \frac{1}{6}\left(\begin{bmatrix}1.6\\2.3\end{bmatrix} - \begin{bmatrix}2.5\\4\end{bmatrix}\right)\left(\begin{bmatrix}1.6\\2.3\end{bmatrix} - \begin{bmatrix}2.5\\4\end{bmatrix}\right)^T + \begin{bmatrix}0.1 & 0\\0 & 0.1\end{bmatrix}$$

$$= \begin{bmatrix}0.36 & 0.5\\0.5 & 1.1\end{bmatrix}$$

This is the same result as in the linear Kalman Filter example because the motion model is already linear!

UKF | Short Example Solution

Correction

$$\check{\mathbf{L}}_k \check{\mathbf{L}}_k^T = \check{\mathbf{P}}_k$$

$$\begin{bmatrix} 0.51 & 0 \\ 0.98 & 0.20 \end{bmatrix} \begin{bmatrix} 0.51 & 0 \\ 0.98 & 0.20 \end{bmatrix}^T = \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$$

$$\check{\mathbf{x}}_1^{(0)} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}$$

Redrawn sigma points!

$$\check{\mathbf{x}}_1^{(1)} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} + \sqrt{3} \begin{bmatrix} 0.51 \\ 0.98 \end{bmatrix} = \begin{bmatrix} 3.54 \\ 5.44 \end{bmatrix}$$

$$\check{\mathbf{x}}_1^{(2)} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} + \sqrt{3} \begin{bmatrix} 0 \\ 0.20 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 5.10 \end{bmatrix}$$

$$\check{\mathbf{x}}_1^{(3)} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} - \sqrt{3} \begin{bmatrix} 0.51 \\ 0.98 \end{bmatrix} = \begin{bmatrix} 1.46 \\ 2.56 \end{bmatrix}$$

$$\check{\mathbf{x}}_1^{(4)} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} - \sqrt{3} \begin{bmatrix} 0 \\ 0.20 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 2.90 \end{bmatrix}$$

$$\hat{y}_1^{(i)} = h_1(\check{\mathbf{x}}_1^{(i)}, 0) \qquad i = 0, \ldots, 2N$$

$$\hat{y}_1^{(0)} = \tan^{-1}\left(\frac{20}{40 - 2.5}\right) = 28.1$$

$$\hat{y}_1^{(1)} = \tan^{-1}\left(\frac{20}{40 - 2.7}\right) = 28.7$$

$$\hat{y}_1^{(2)} = \tan^{-1}\left(\frac{20}{40 - 2.5}\right) = 28.1$$

$$\hat{y}_1^{(3)} = \tan^{-1}\left(\frac{20}{40 - 2.3}\right) = 27.4$$

$$\hat{y}_1^{(4)} = \tan^{-1}\left(\frac{20}{40 - 2.5}\right) = 28.1$$

UKF | Short Example Solution

Correction

$$P_y = \sum_{i=0}^{2N} \alpha^{(i)} \left(\hat{y}_1^{(i)} - \hat{y}_k\right) \left(\hat{y}_1^{(i)} - \hat{y}_k\right)^T + R_k$$

$$= \frac{1}{3}(28.1 - 28.1)^2 + \frac{1}{6}(28.7 - 28.1)^2 + \frac{1}{6}(28.1 - 28.1)^2 + \frac{1}{6}(27.4 - 28.1)^2 + \frac{1}{6}(28.1 - 28.1)^2 + 0.01$$

$$= 0.16$$

$$\hat{y}_1 = \sum_{i=0}^{2N} \alpha^{(i)} \hat{y}_1^{(i)}$$

$$= \frac{1}{3}(28.1) + \frac{1}{6}(28.7) + \frac{1}{6}(28.1) + \frac{1}{6}(27.4) + \frac{1}{6}(28.1)$$

$$= 28.1$$

UKF | Short Example Solution

Correction

$$\mathbf{P}_{xy} = \sum_{i=0}^{2N} \alpha^{(i)} \left(\check{\mathbf{x}}_k^{(i)} - \check{\mathbf{x}}_k\right) \left(\hat{y}_1^{(i)} - \hat{y}_k\right)^T$$

$$= \frac{1}{3}\left(\begin{bmatrix} 2.5 \\ 4 \end{bmatrix} - \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}\right)(28.1 - 28.1) + \frac{1}{6}\left(\begin{bmatrix} 3.54 \\ 5.44 \end{bmatrix} - \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}\right)(28.7 - 28.1) + \frac{1}{6}\left(\begin{bmatrix} 2.5 \\ 5.10 \end{bmatrix} - \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}\right)(28.1 - 28.1)$$

$$+ \frac{1}{6}\left(\begin{bmatrix} 1.46 \\ 2.56 \end{bmatrix} - \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}\right)(27.4 - 28.1) + \frac{1}{6}\left(\begin{bmatrix} 2.5 \\ 2.90 \end{bmatrix} - \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}\right)(28.1 - 28.1)$$

$$= \begin{bmatrix} 0.23 \\ 0.32 \end{bmatrix}$$

$$\mathbf{K}_1 = \mathbf{P}_{xy} P_y^{-1} = \begin{bmatrix} 0.23 \\ 0.32 \end{bmatrix} \frac{1}{0.16} = \begin{bmatrix} 1.47 \\ 2.05 \end{bmatrix}$$

$$\hat{\mathbf{x}}_1 = \check{\mathbf{x}}_1 + \mathbf{K}_1 \left(y_1 - \hat{y}_1\right) = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} + \begin{bmatrix} 1.47 \\ 2.05 \end{bmatrix}(30.0 - 28.1) = \begin{bmatrix} 5.33 \\ 7.93 \end{bmatrix} = \begin{bmatrix} \hat{p}_1 \\ \hat{\dot{p}}_1 \end{bmatrix}$$

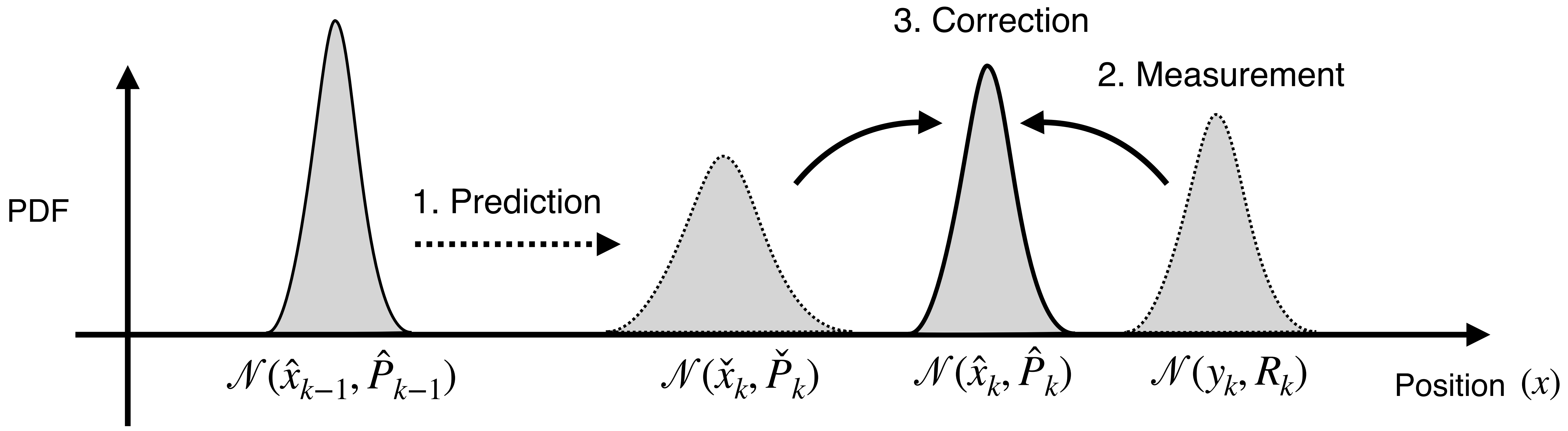
Summary | Unscented Kalman Filter (UKF)

- The UKF uses the unscented transform to adapt the Kalman filter to nonlinear systems.

- The unscented transform works by passing a small set of carefully chosen samples through a nonlinear system, and computing the mean and covariance of the outputs.

- The unscented transform does a better job of approximating the output distribution than analytical local linearization, for similar computational cost.

Summary | Kalman Filtering



- The Kalman filter (KF) is a form of recursive least-squares estimation that allows us to combine information frsom a motion model and sensor measurements.

- The KF uses the motion model to make *predictions* of the state, and uses the measurements to make *corrections* to the predictions.

- The KF is the Best Linear Unbiased Estimator (BLUE).

Summary | Nonlinear Kalman Filtering

	EKF	ES-EKF	UKF
Operating Principle	Linearization (Full State)	Linearization (Error State)	Unscented Transform
Accuracy	Good	Better	Best
Jacobians	Required	Required	Not required
Speed	Slightly faster	Slightly faster	Slightly slower