Robot Mapping

Unscented Kalman Filter

Cyrill Stachniss

UNI FREIBURG

Autonomous Intelligent Systems

KF, EKF and UKF

- Kalman filter requires linear models
- EKF linearizes via Taylor expansion

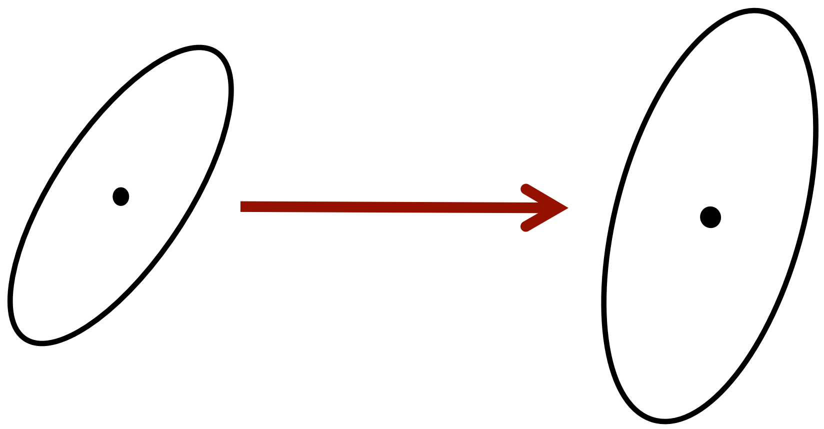
Is there a better way to linearize?

Unscented Transform

↓

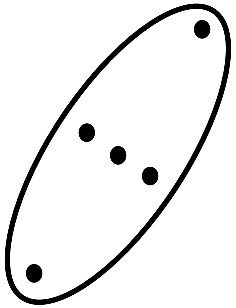
Unscented Kalman Filter (UKF)

Taylor Approximation (EKF)



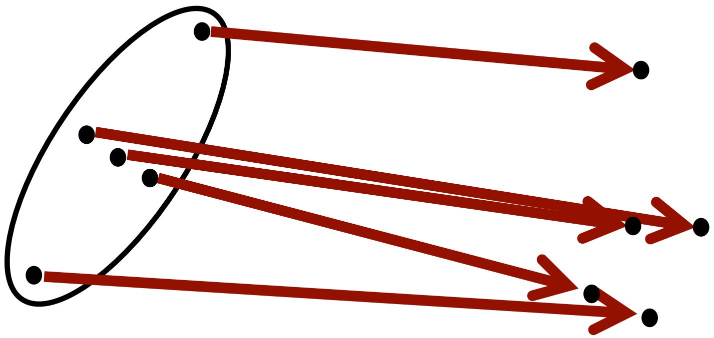
Linearization of the non-linear function through Taylor expansion

Unscented Transform



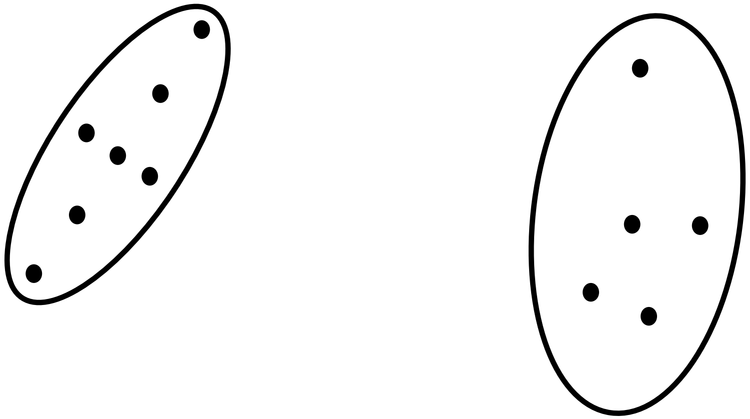
Compute a set of (so-called) sigma points

Unscented Transform



Transform each sigma point
through the non-linear function

Unscented Transform



Compute Gaussian from the transformed and weighted sigma points

Unscented Transform Overview

- Compute a set of sigma points
- Each sigma points has a weight
- Transform the point through the non-linear function
- Compute a Gaussian from weighted points

- Avoids to linearize **around the mean** as Taylor expansion (and EKF) does

Sigma Points

- How to choose the sigma points?
- How to set the weights?

Sigma Points Properties

- How to choose the sigma points?
- How to set the weights?
- Select $\mathcal{X}^{[i]}, w^{[i]}$ so that:

$$\sum_i w^{[i]} = 1$$

$$\mu = \sum_i w^{[i]} \mathcal{X}^{[i]}$$

$$\Sigma = \sum_i w^{[i]} (\mathcal{X}^{[i]} - \mu)(\mathcal{X}^{[i]} - \mu)^T$$

- There is no unique solution for $\mathcal{X}^{[i]}, w^{[i]}$

Sigma Points

- Choosing the sigma points

$$\mathcal{X}^{[0]} = \mu$$

First sigma point is the mean

Sigma Points

- Choosing the sigma points

$$
\begin{aligned}
\mathcal{X}^{[0]} &= \mu \\
\mathcal{X}^{[i]} &= \mu + \left(\sqrt{(n+\lambda)\,\Sigma}\right)_i \qquad \text{for } i = 1, \ldots, n \\
\mathcal{X}^{[i]} &= \mu - \left(\sqrt{(n+\lambda)\,\Sigma}\right)_{i-n} \qquad \text{for } i = n+1, \ldots, 2n
\end{aligned}
$$

matrix square root

column vector

dimensionality

scaling parameter

Matrix Square Root

- Defined as S with $\Sigma = SS$
- Computed via diagonalization

$$
\begin{aligned}
\Sigma &= VDV^{-1} \\
&= V \begin{pmatrix} d_{11} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & d_{nn} \end{pmatrix} V^{-1} \\
&= V \begin{pmatrix} \sqrt{d_{11}} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \sqrt{d_{nn}} \end{pmatrix} \begin{pmatrix} \sqrt{d_{11}} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \sqrt{d_{nn}} \end{pmatrix} V^{-1}
\end{aligned}
$$

Matrix Square Root

- Thus, we can define

$$S = V \underbrace{\begin{pmatrix} \sqrt{d_{11}} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \sqrt{d_{nn}} \end{pmatrix}}_{D^{1/2}} V^{-1}$$

- so that

$$SS = (VD^{1/2}V^{-1})(VD^{1/2}V^{-1}) = VDV^{-1} = \Sigma$$

Cholesky Matrix Square Root

- Alternative definition of the matrix square root

$$L \text{ with } \Sigma = LL^T$$

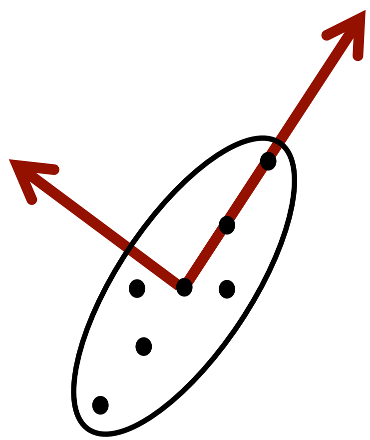
- Result of the Cholesky decomposition
- Numerically stable solution
- Often used in UKF implementations
- L and Σ have the same Eigenvectors

Sigma Points and Eigenvectors

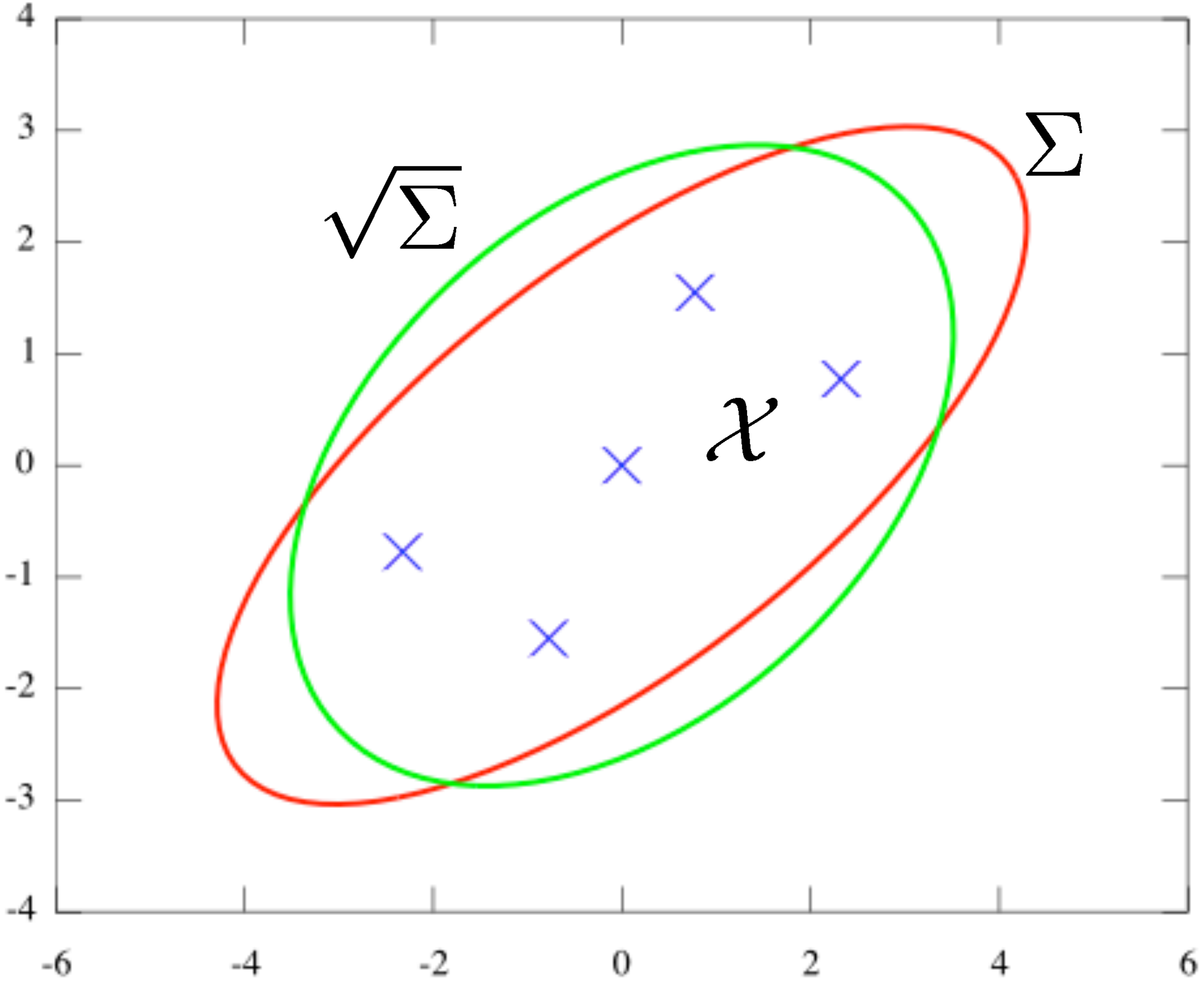
- Sigma point **can** but **do not have to** lie on the main axes of Σ

$$\mathcal{X}^{[i]} = \mu + \left(\sqrt{(n+\lambda)\,\Sigma}\right)_i \quad \text{for } i = 1, \ldots, n$$

$$\mathcal{X}^{[i]} = \mu - \left(\sqrt{(n+\lambda)\,\Sigma}\right)_{i-n} \quad \text{for } i = n+1, \ldots, 2n$$



Sigma Points Example



Sigma Point Weights

- Weight sigma points

for computing the mean

parameters

$$
\begin{aligned}
w_m^{[0]} &= \frac{\lambda}{n+\lambda} \\
w_c^{[0]} &= w_m^{[0]} + (1 - \alpha^2 + \beta) \\
w_m^{[i]} = w_c^{[i]} &= \frac{1}{2(n+\lambda)} \quad \text{for } i = 1, \ldots, 2n
\end{aligned}
$$

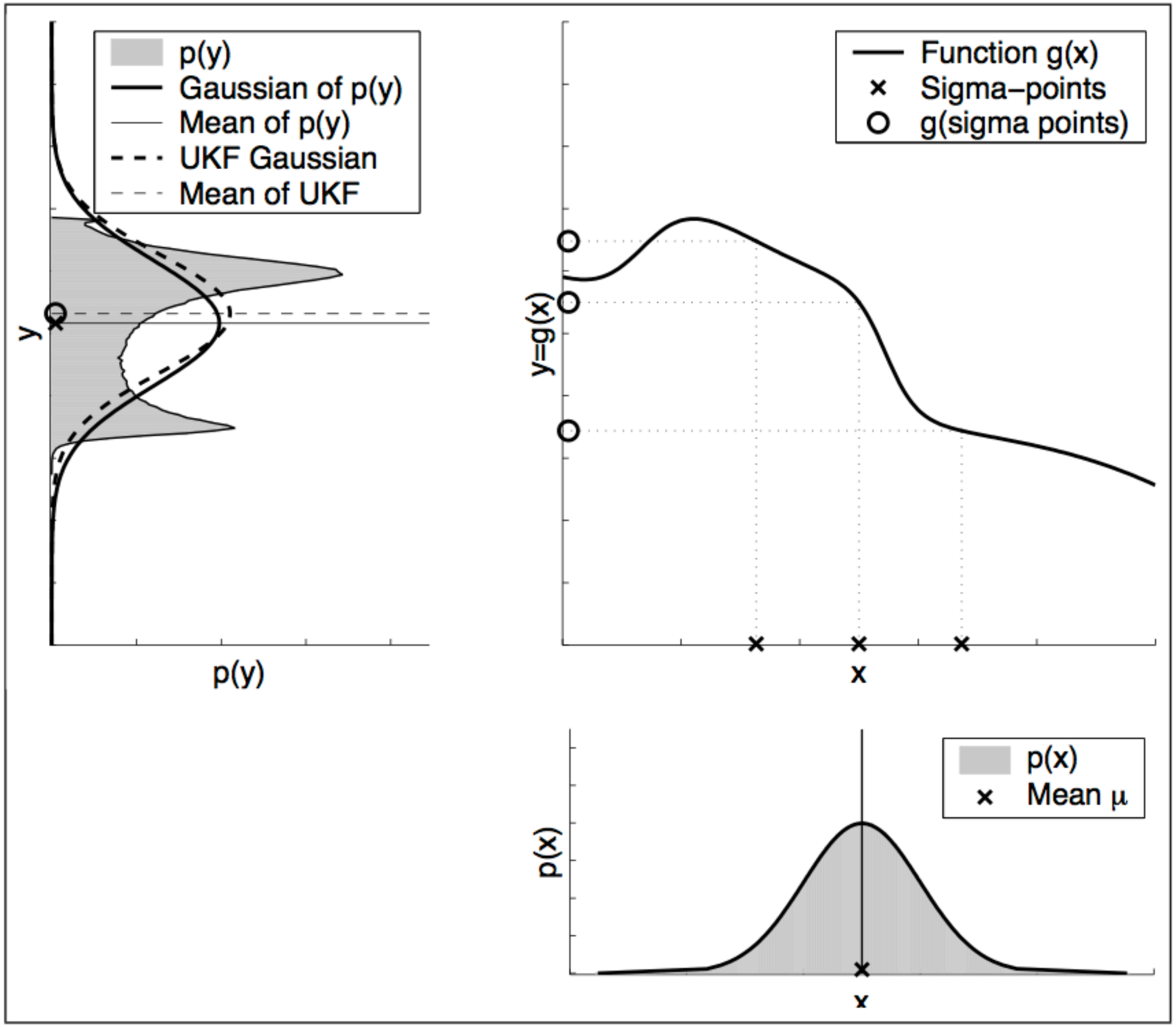
for computing the covariance

Recover the Gaussian

- Compute Gaussian from weighted and transformed points

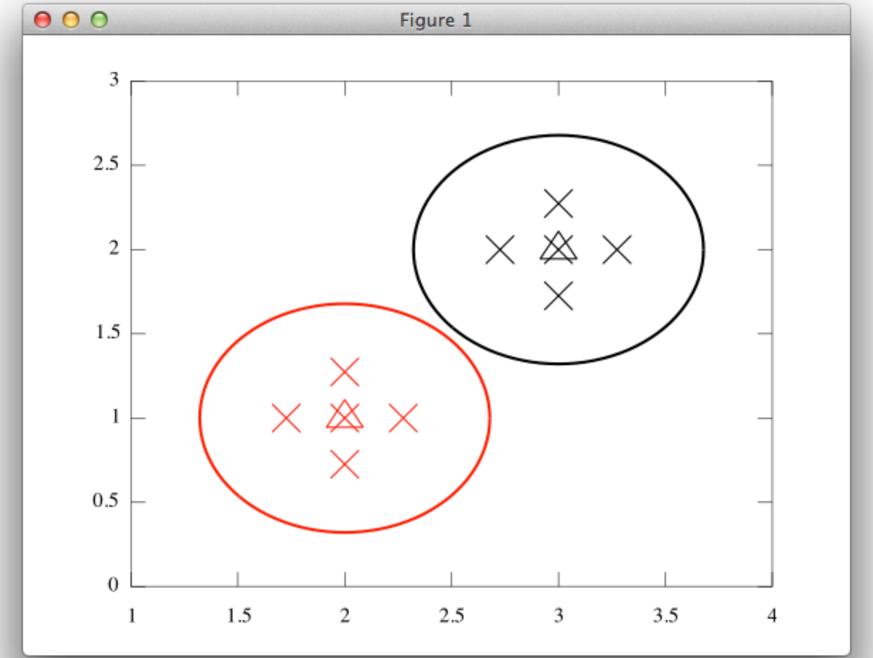
$$\begin{aligned} \mu' &= \sum_{i=0}^{2n} w_m^{[i]} \, g(\mathcal{X}^{[i]}) \\ \Sigma' &= \sum_{i=0}^{2n} w_c^{[i]} \, (g(\mathcal{X}^{[i]}) - \mu')(g(\mathcal{X}^{[i]}) - \mu')^T \end{aligned}$$

Example

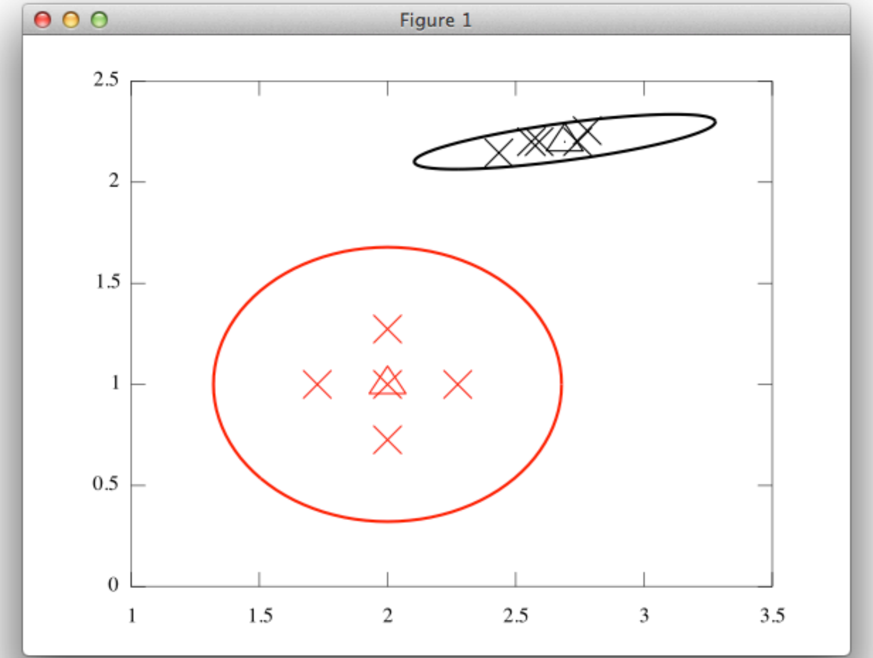


p(y)
Gaussian of p(y)
Mean of p(y)
UKF Gaussian
Mean of UKF
y
p(y)
Function g(x)
Sigma–points
g(sigma points)
y=g(x)
x
p(x)
Mean μ
p(x)
x

Examples



$$g((x,y)^T) = \begin{pmatrix} x+1 \\ y+1 \end{pmatrix}^T$$



$$g((x,y)^T) = \begin{pmatrix} 1+x+\sin(2x)+\cos(y) \\ 2+0.2y \end{pmatrix}^T$$

Unscented Transform Summary

- Sigma points

$$\begin{aligned}
\mathcal{X}^{[0]} &= \mu \\
\mathcal{X}^{[i]} &= \mu + \left(\sqrt{(n+\lambda)\,\Sigma}\right)_i \qquad \text{for } i = 1, \ldots, n \\
\mathcal{X}^{[i]} &= \mu - \left(\sqrt{(n+\lambda)\,\Sigma}\right)_{i-n} \qquad \text{for } i = n+1, \ldots, 2n
\end{aligned}$$

- Weights

$$\begin{aligned}
w_m^{[0]} &= \frac{\lambda}{n+\lambda} \\
w_c^{[0]} &= w_m^{[0]} + (1 - \alpha^2 + \beta) \\
w_m^{[i]} = w_c^{[i]} &= \frac{1}{2(n+\lambda)} \qquad \text{for } i = 1, \ldots, 2n
\end{aligned}$$

UT Parameters

- Free parameters as there is no unique solution
- Scaled Unscented Transform suggests

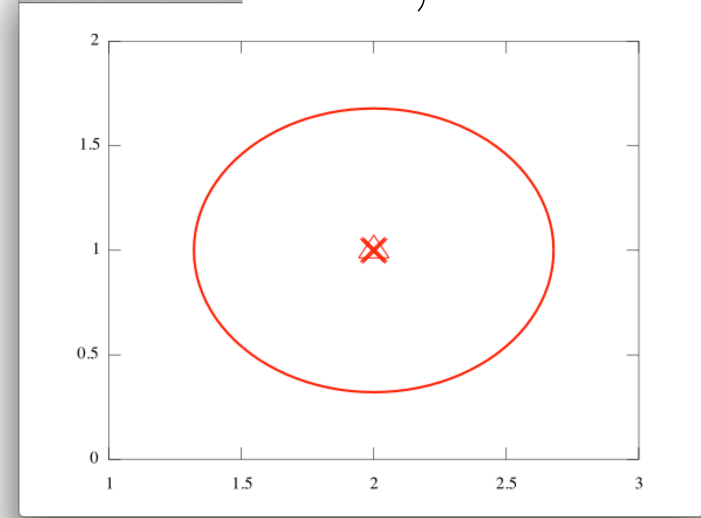
$$\begin{aligned} \kappa &\geq 0 \\ \alpha &\in (0, 1] \\ \lambda &= \alpha^2(n + \kappa) - n \\ \beta &= 2 \end{aligned}$$

Influence how far the sigma points are away from the mean

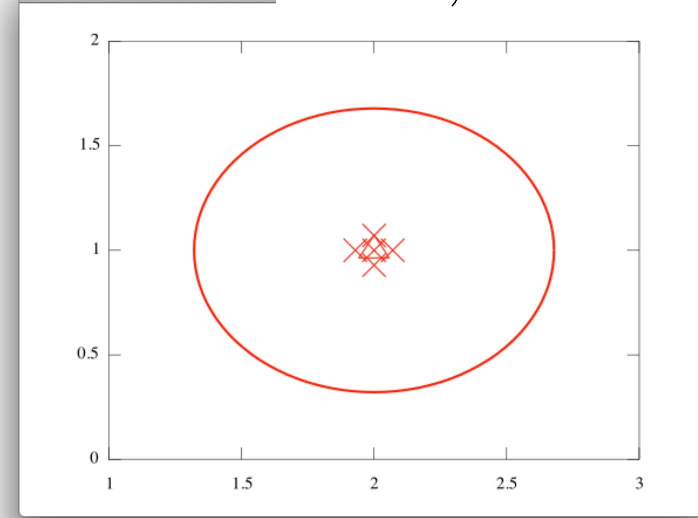
Optimal choice for Gaussians

Examples

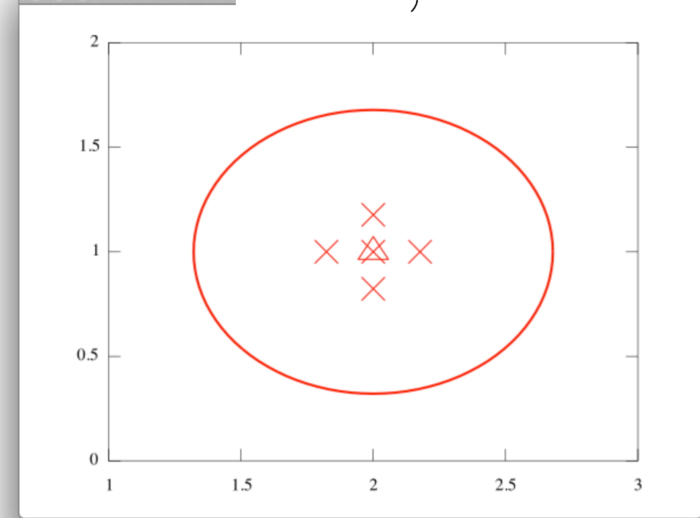
$\kappa = 3, \alpha = 0.01$



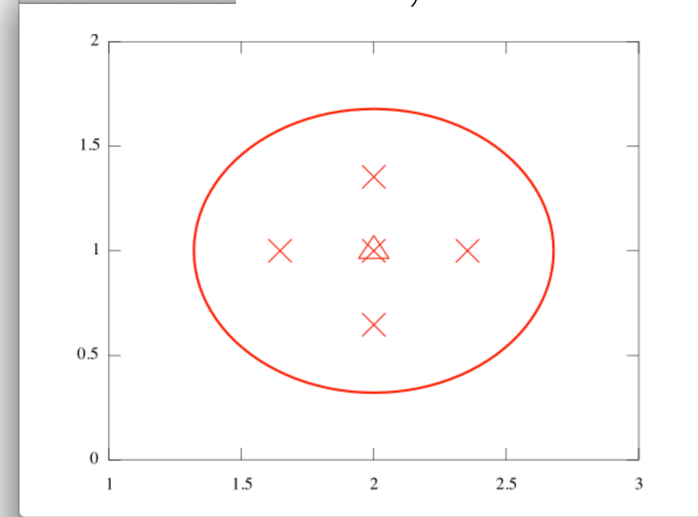
$\kappa = 3, \alpha = 0.1$



$\kappa = 3, \alpha = 0.25$

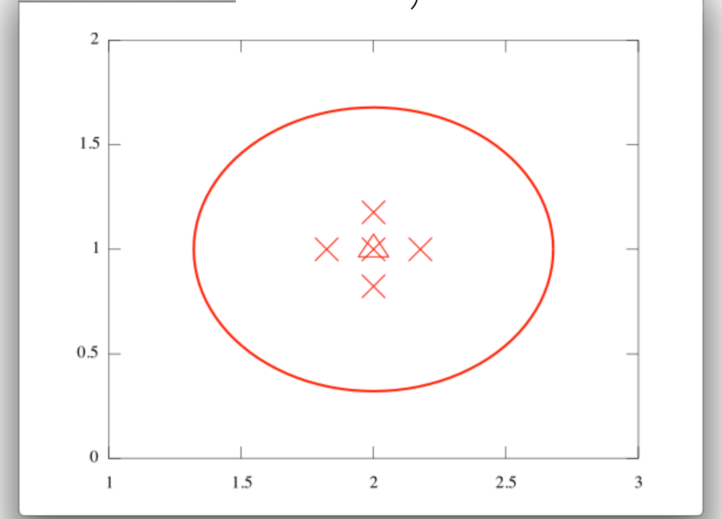


$\kappa = 3, \alpha = 0.75$

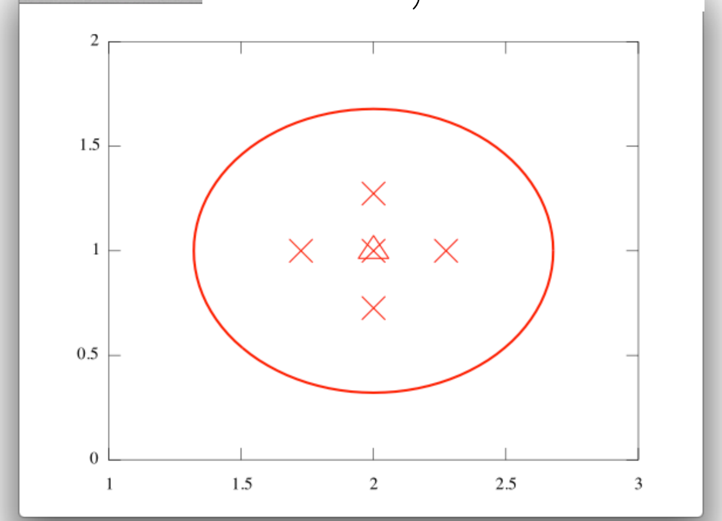


Examples

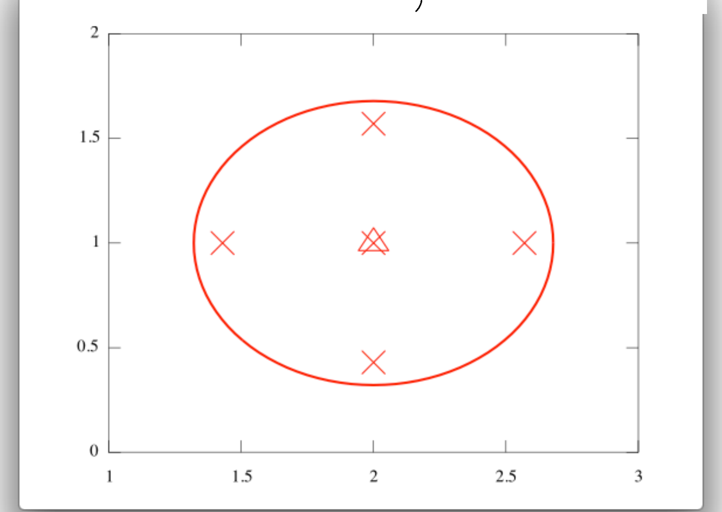
$\kappa = 3, \alpha = 0.25$



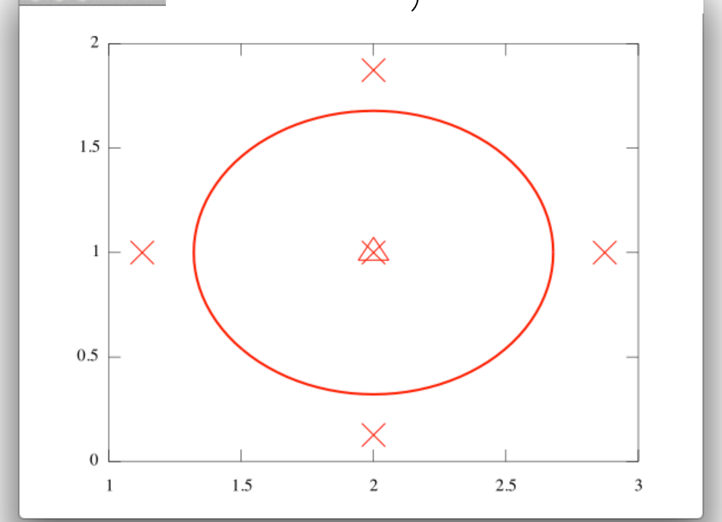
$\kappa = 10, \alpha = 0.25$



$\kappa = 50, \alpha = 0.25$



$\kappa = 120, \alpha = 0.25$



EKF Algorithm

$$\textbf{Extended\_Kalman\_filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\textbf{:}$$

$$\bar{\mu}_t = g(u_t, \mu_{t-1})$$
$$\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$$

$$K_t = \bar{\Sigma}_t \, H_t^T (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$$
$$\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$$
$$return \ \mu_t, \Sigma_t$$

EKF to UKF – Prediction

Unscented

~~Extended~~_Kalman_filter$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$**:**

$\bar{\mu}_t =$ replace this by sigma point

$\bar{\Sigma}_t =$ propagation of the motion

$K_t = \bar{\Sigma}_t\, H_t^T (H_t\, \bar{\Sigma}_t\, H_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

$\Sigma_t = (I - K_t\, H_t)\, \bar{\Sigma}_t$

return μ_t, Σ_t

UKF Algorithm – Prediction

1: **Unscented_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$**:**

2: $\mathcal{X}_{t-1} = (\mu_{t-1} \quad \mu_{t-1} + \sqrt{(n+\lambda)\Sigma_{t-1}} \quad \mu_{t-1} - \sqrt{(n+\lambda)\Sigma_{t-1}})$

3: $\bar{\mathcal{X}}_t^* = g(u_t, \mathcal{X}_{t-1})$

4: $\bar{\mu}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{X}}_t^{*[i]}$

5: $\bar{\Sigma}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)(\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)^T + R_t$

EKF to UKF – Correction

Unscented

~~Extended~~_Kalman_filter$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

$\bar{\mu}_t =$ replace this by sigma point
$\bar{\Sigma}_t =$ propagation of the motion

use sigma point propagation for the expected observation and Kalman gain

$\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$
$\Sigma_t = \bar{\Sigma}_t - K_t\, S_t\, K_t^T$
$return\ \mu_t, \Sigma_t$

UKF Algorithm – Correction (1)

6: $\bar{\mathcal{X}}_t = (\bar{\mu}_t \quad \bar{\mu}_t + \sqrt{(n+\lambda)\bar{\Sigma}_t} \quad \bar{\mu}_t - \sqrt{(n+\lambda)\bar{\Sigma}_t})$

7: $\bar{\mathcal{Z}}_t = h(\bar{\mathcal{X}}_t)$

8: $\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$

9: $S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + Q_t$

10: $\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T$

11: $K_t = \bar{\Sigma}_t^{x,z} \; S_t^{-1}$

UKF Algorithm – Correction (1)

6: $\bar{\mathcal{X}}_t = (\bar{\mu}_t \quad \bar{\mu}_t + \sqrt{(n+\lambda)\bar{\Sigma}_t} \quad \bar{\mu}_t - \sqrt{(n+\lambda)\bar{\Sigma}_t})$

7: $\bar{\mathcal{Z}}_t = h(\bar{\mathcal{X}}_t)$

8: $\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$

9: $S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + Q_t$

10: $\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T$

11: $K_t = \bar{\Sigma}_t^{x,z} \; S_t^{-1}$

$$K_t = \overbrace{\bar{\Sigma}_t \; H_t^T}^{\bar{\Sigma}_t^{x,z}} \overbrace{(H_t \; \bar{\Sigma}_t \; H_t^T + Q_t)}^{S_t}{}^{-1}$$

(from EKF)

UKF Algorithm – Correction (2)

$$\bar{\mathcal{X}}_t = (\bar{\mu}_t \quad \bar{\mu}_t + \sqrt{(n+\lambda)\bar{\Sigma}_t} \quad \bar{\mu}_t - \sqrt{(n+\lambda)\bar{\Sigma}_t})$$

$$\bar{\mathcal{Z}}_t = h(\bar{\mathcal{X}}_t)$$

$$\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$$

$$S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + Q_t$$

$$\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T$$

$$K_t = \bar{\Sigma}_t^{x,z} \; S_t^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t (z_t - \hat{z}_t)$$

$$\Sigma_t = \bar{\Sigma}_t - K_t \; S_t \; K_t^T$$

$$\text{return } \mu_t, \Sigma_t$$

UKF Algorithm – Correction (2)

6: $\bar{\mathcal{X}}_t = (\bar{\mu}_t \quad \bar{\mu}_t + \sqrt{(n+\lambda)\bar{\Sigma}_t} \quad \bar{\mu}_t - \sqrt{(n+\lambda)\bar{\Sigma}_t})$

7: $\bar{\mathcal{Z}}_t = h(\bar{\mathcal{X}}_t)$

8: $\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$

9: $S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + Q_t$

10: $\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T$

11: $K_t = \bar{\Sigma}_t^{x,z} \; S_t^{-1}$

12: $\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$

13: $\Sigma_t = \bar{\Sigma}_t - K_t \; S_t \; K_t^T$

14: return μ_t, Σ_t

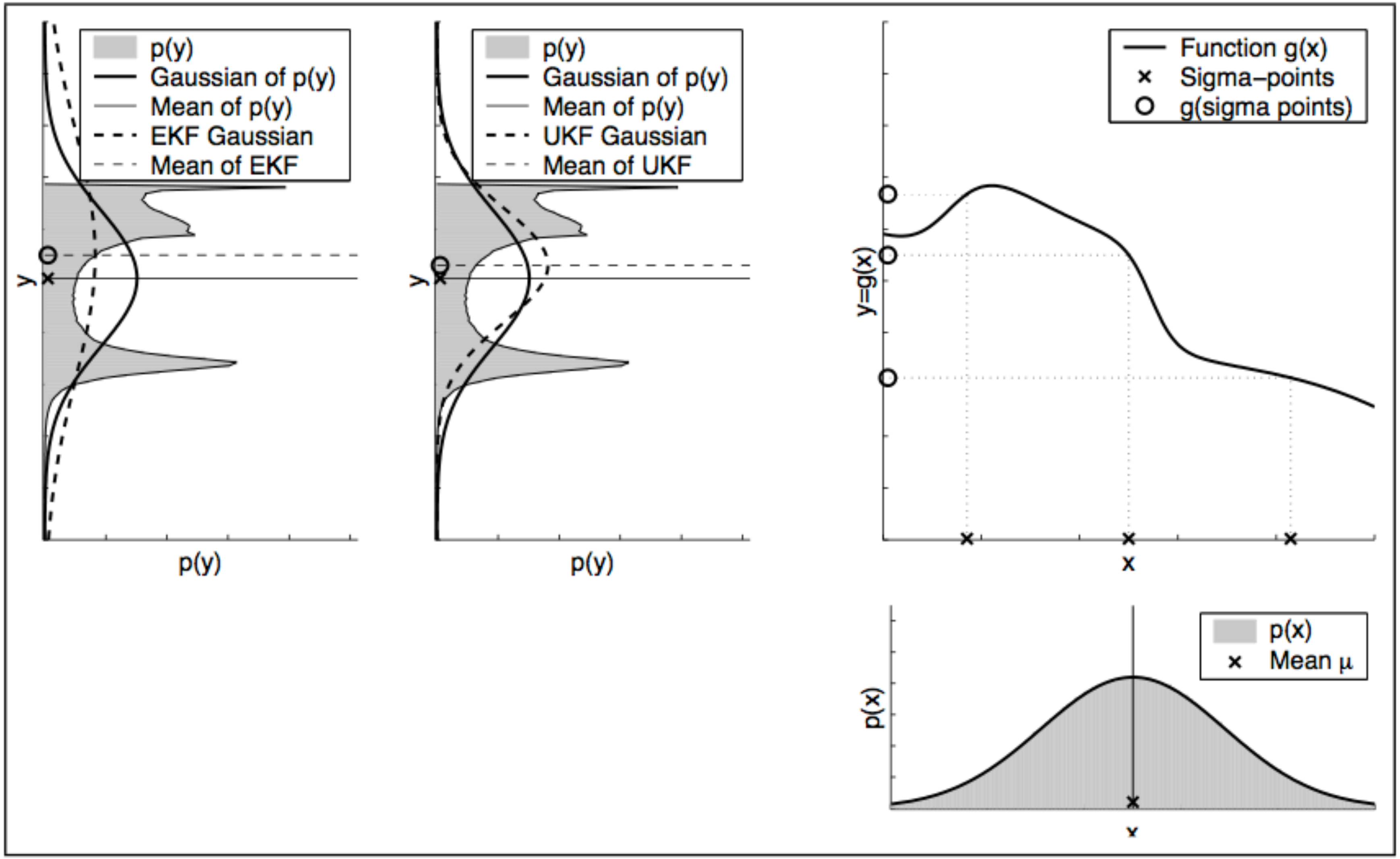
$$
\begin{aligned}
\Sigma_t &= (I - K_t H_t)\bar{\Sigma}_t \\
&= \bar{\Sigma}_t - K_t H_t \bar{\Sigma}_t \\
&= \bar{\Sigma}_t - K_t \left(\Sigma^{x,z}\right)^T \\
&= \bar{\Sigma}_t - K_t \left(\Sigma^{x,z} s_t^{-1} S_t\right)^T \\
&= \bar{\Sigma}_t - K_t \left(K_t S_t\right)^T \\
&= \bar{\Sigma}_t - K_t S_t^T K_t^T \\
&= \bar{\Sigma}_t - K_t S_t K_t^T
\end{aligned}
$$

(see next slide)

From EKF to UKF – Computing the Covariance

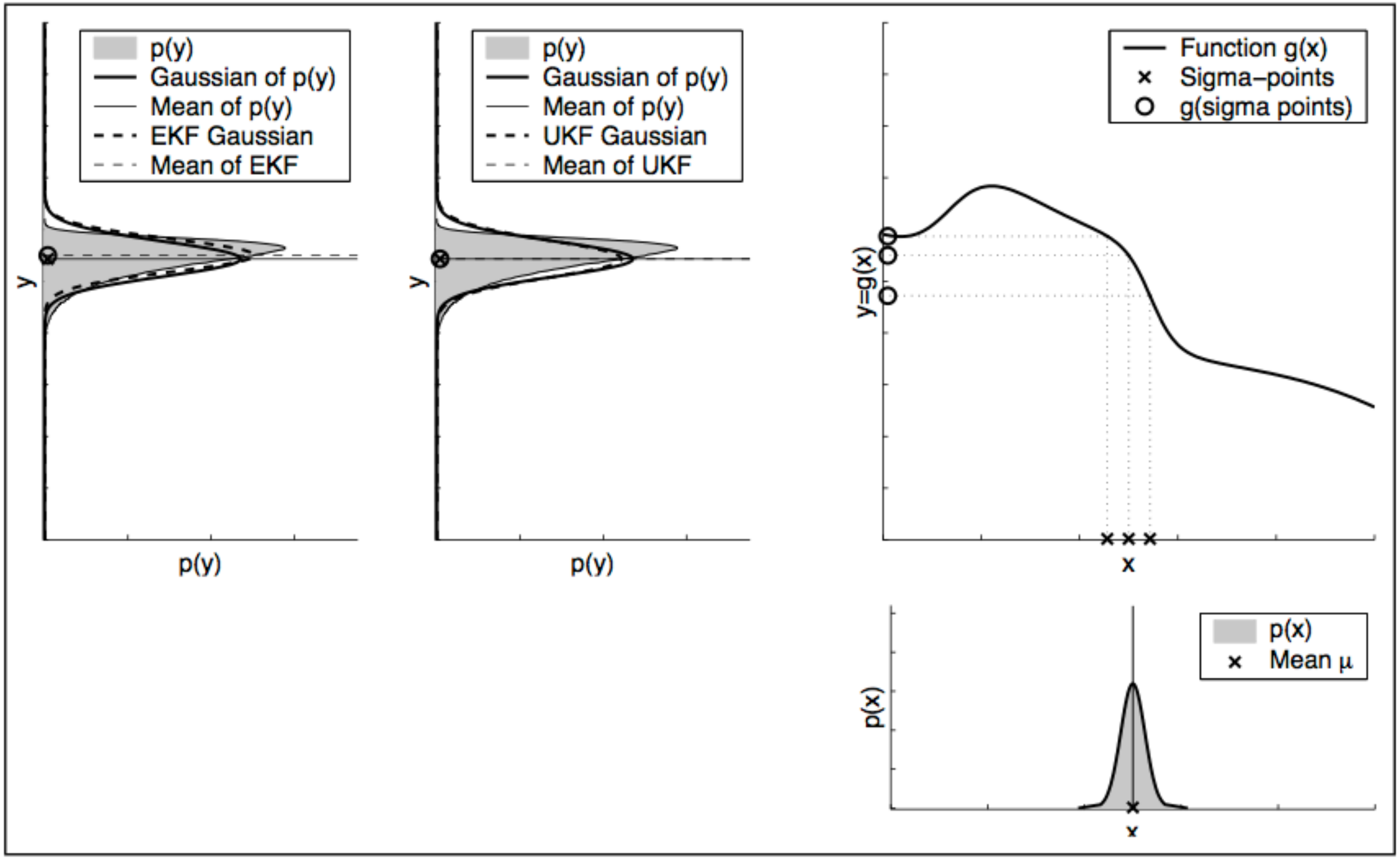
$$\begin{aligned}
\Sigma_t &= (I - K_t H_t)\bar{\Sigma}_t \\
&= \bar{\Sigma}_t - K_t H_t \bar{\Sigma}_t \\
&= \bar{\Sigma}_t - K_t \left(\bar{\Sigma}^{x,z}\right)^T \\
&= \bar{\Sigma}_t - K_t \left(\bar{\Sigma}^{x,z} S_t^{-1} S_t\right)^T \\
&= \bar{\Sigma}_t - K_t \left(K_t S_t\right)^T \\
&= \bar{\Sigma}_t - K_t S_t^T K_t^T \\
&= \bar{\Sigma}_t - K_t S_t K_t^T
\end{aligned}$$

UKF vs. EKF



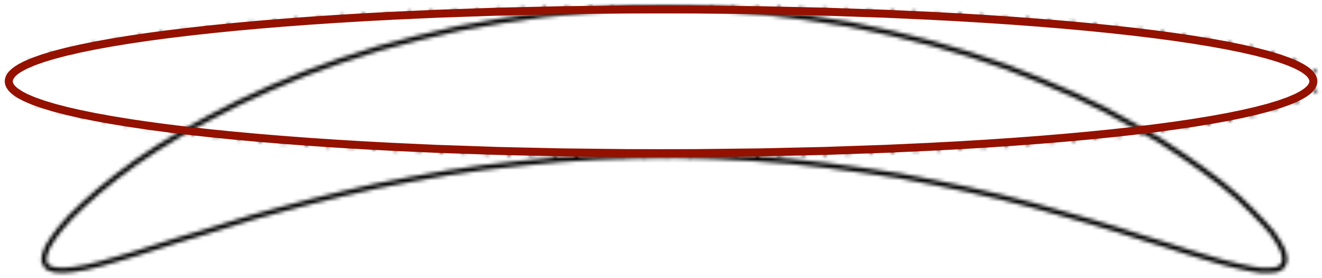
p(y)
Gaussian of p(y)
Mean of p(y)
EKF Gaussian
Mean of EKF
y
p(y)
p(y)
Gaussian of p(y)
Mean of p(y)
UKF Gaussian
Mean of UKF
y
p(y)
Function g(x)
Sigma–points
g(sigma points)
y=g(x)
x
p(x)
Mean μ
p(x)
x

UKF vs. EKF (Small Covariance)

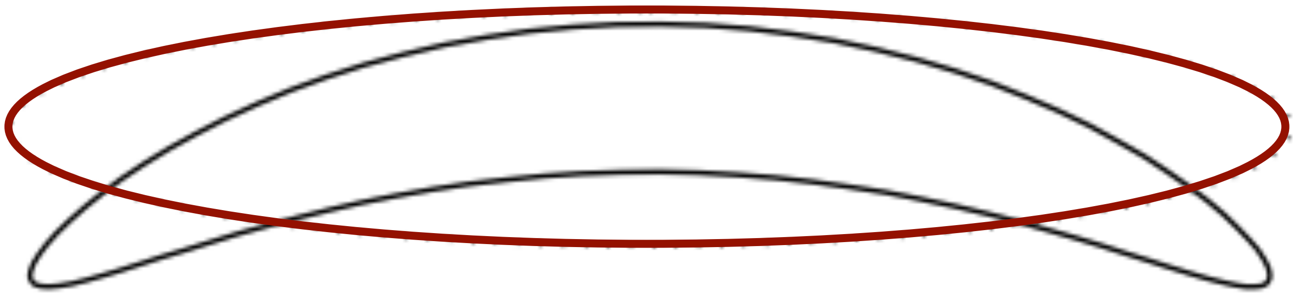


UKF vs. EKF – Banana Shape

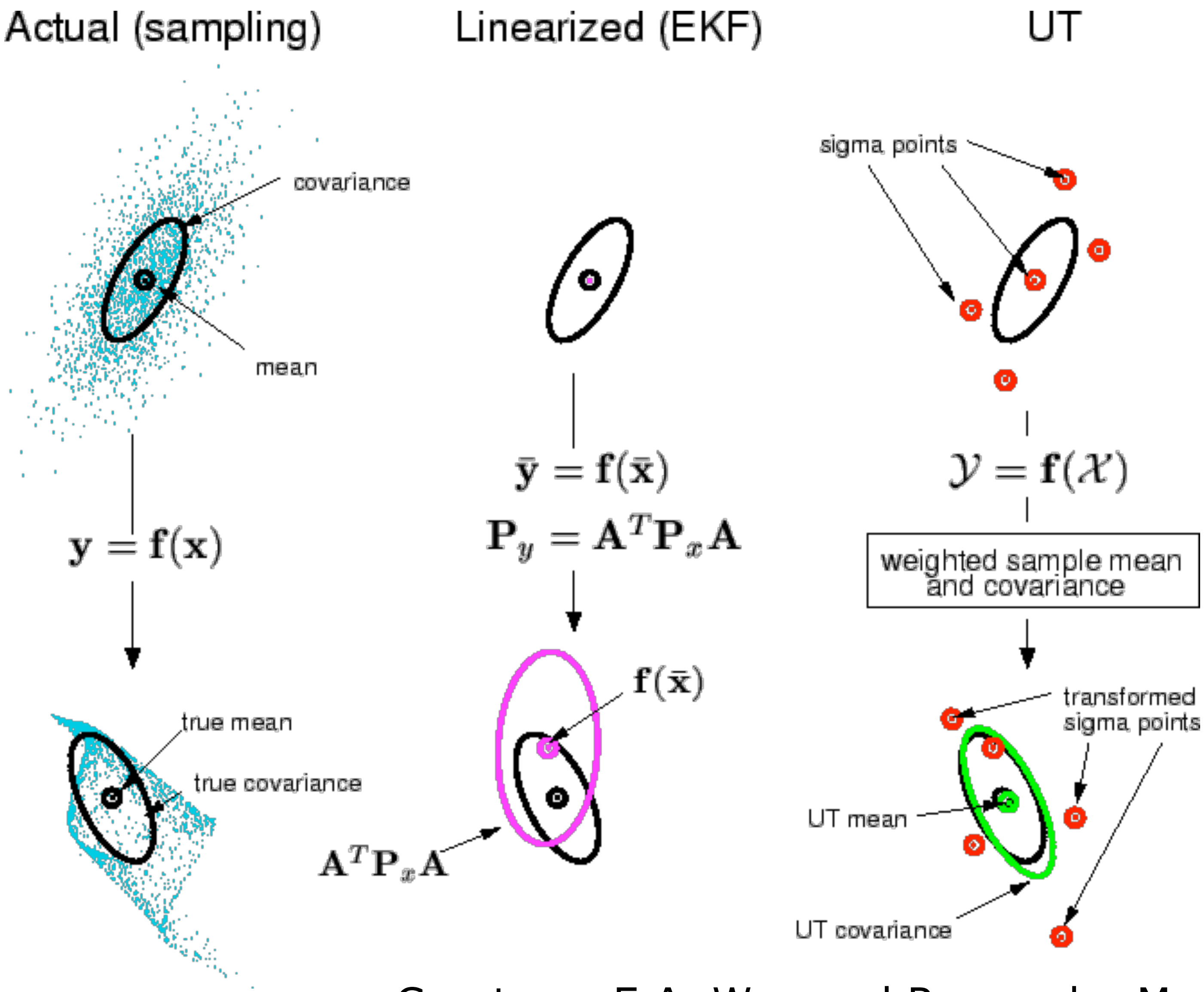
EKF approximation



UKF approximation



UKF vs. EKF



Courtesy: E.A. Wan and R. van der Merwe

UT/UKF Summary

- Unscented transforms as an alternative to linearization
- UT is a better approximation than Taylor expansion
- UT uses sigma point propagation
- Free parameters in UT
- UKF uses the UT in the prediction and correction step

UKF vs. EKF

- Same results as EKF for linear models
- Better approximation than EKF for non-linear models
- Differences often "somewhat small"
- No Jacobians needed for the UKF
- Same complexity class
- Slightly slower than the EKF
- Still restricted to Gaussian distributions

Literature

Unscented Transform and UKF

- Thrun et al.: "Probabilistic Robotics", Chapter 3.4
- "A New Extension of the Kalman Filter to Nonlinear Systems" by Julier and Uhlmann, 1995
- "Dynamische Zustandsschätzung" by Fränken, 2006, pages 31-34