Unscented Kalman Filter

Gaurav Pandey

EKF Localization

Prediction:

$$G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{pmatrix} \frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial \theta'}{\partial \mu_{t-1,x}} & \frac{\partial \theta'}{\partial \mu_{t-1,y}} & \frac{\partial \theta'}{\partial \mu_{t-1,\theta}} \end{pmatrix}$$

Jacobian of *g* w.r.t location

$$B_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t} = \begin{pmatrix} \frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial \omega_t} \\ \frac{\partial y'}{\partial v_t} & \frac{\partial y'}{\partial \omega_t} \\ \frac{\partial \theta'}{\partial v_t} & \frac{\partial \theta'}{\partial \omega_t} \end{pmatrix}$$

Jacobian of *g* w.r.t control

$$Q_t = \begin{pmatrix} (\alpha_1 |v_t| + \alpha_2 |\omega_t|)^2 & 0 \\ 0 & (\alpha_3 |v_t| + \alpha_4 |\omega_t|)^2 \end{pmatrix}$$

Motion noise

$$\bar{\mu}_t = g(u_t, \mu_{t-1})$$

Predicted mean

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + B_t Q_t B_t^T$$

Predicted covariance

EKF Localization

Correction:

$$\hat{z}_t = \begin{pmatrix} \sqrt{(m_x - \bar{\mu}_{t,x})^2 + (m_y - \bar{\mu}_{t,y})^2} \\ \operatorname{atan2}(m_y - \bar{\mu}_{t,y}, m_x - \bar{\mu}_{t,x}) - \bar{\mu}_{t,\theta} \end{pmatrix}$$

Predicted measurement mean

$$H_t = \frac{\partial h(\bar{\mu}_t, m)}{\partial x_t} = \begin{pmatrix} \frac{\partial r_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,y}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,\theta}} \\ \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,y}} & \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,\theta}} \end{pmatrix}$$

Jacobian of *h* w.r.t location

$$R_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{pmatrix}$$

$$S_t = H_t \bar{\Sigma}_t H_t^T + R_t$$

Innovation covariance

$$K_t = \bar{\Sigma}_t H_t^T S_t^{-1}$$

Kalman gain

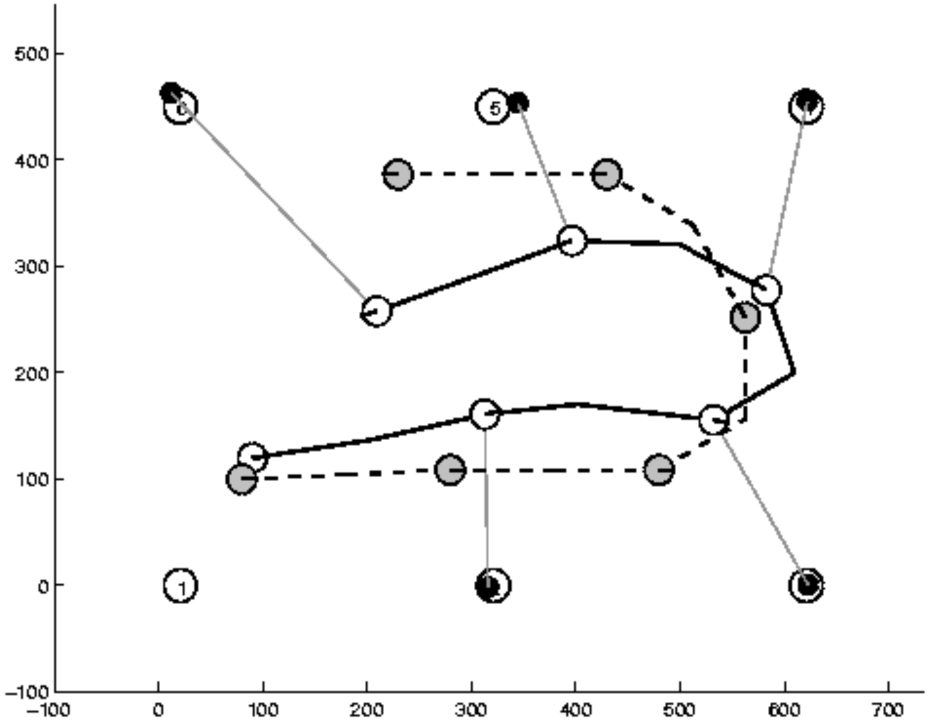
$$\mu_t = \bar{\mu}_t + K_t (z_t - \hat{z}_t)$$

Updated mean

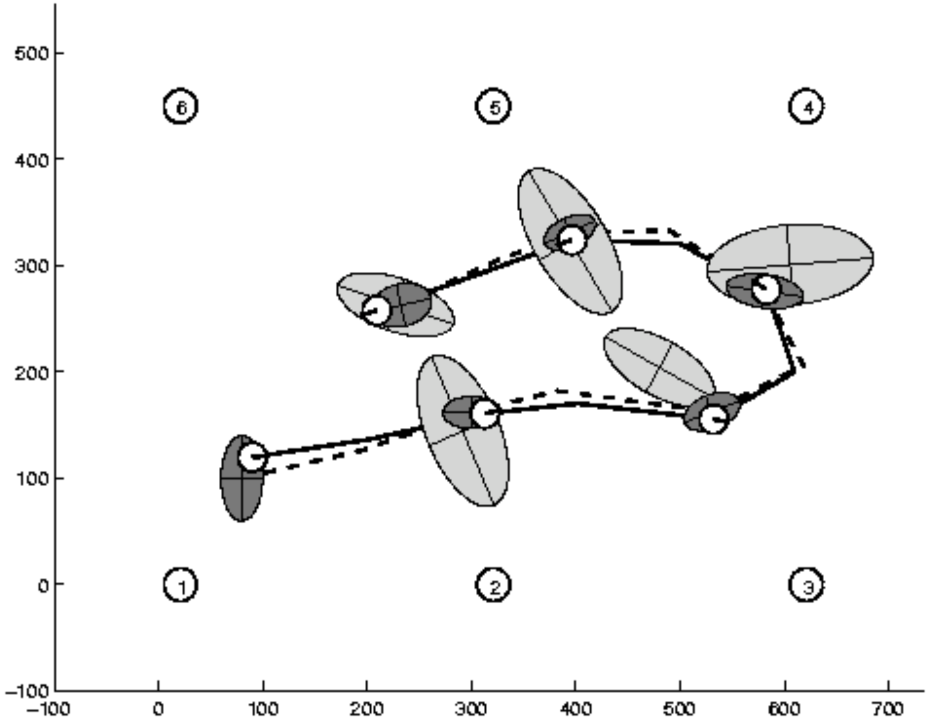
$$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$$

Updated covariance

Estimation sequence with a good range sensor

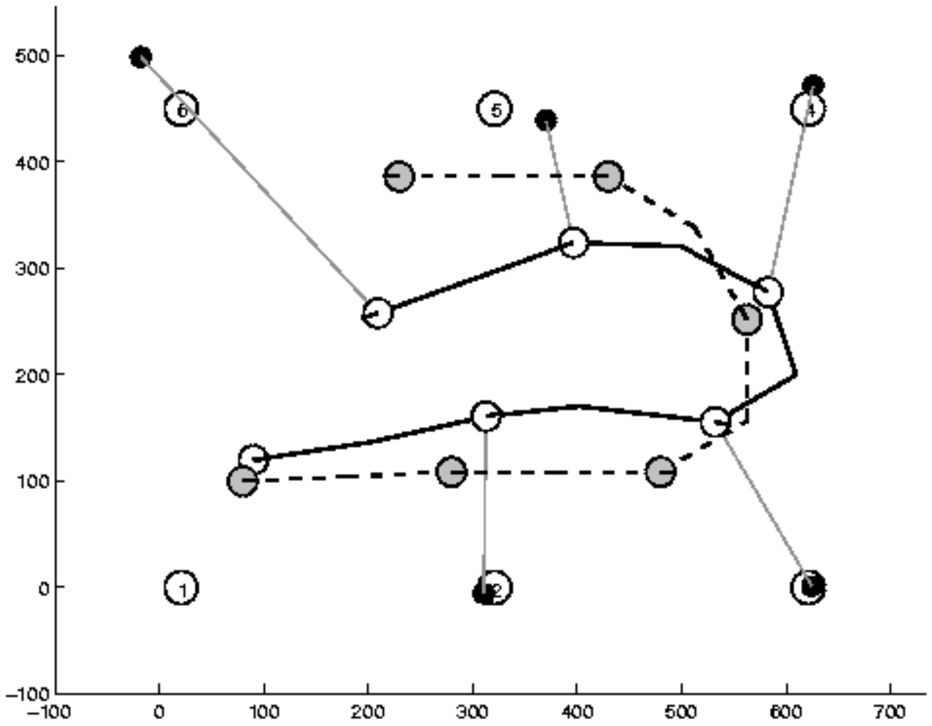


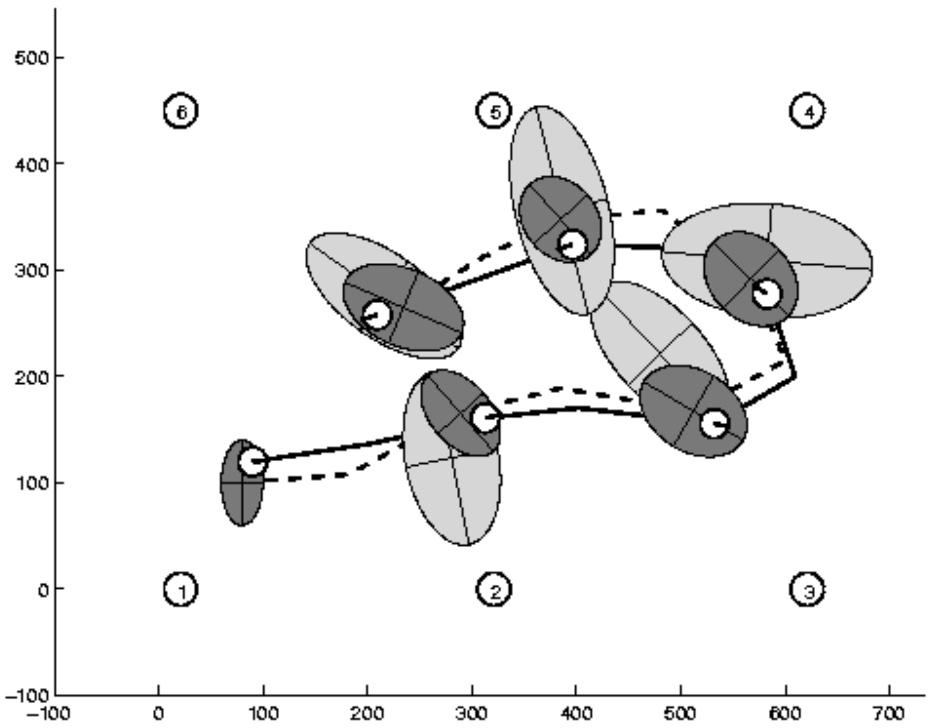
Dashed line is trajectory estimated from motion control. Solid line is true motion of the robot



Dashed line is corrected robot trajectory along with predicted and corrected uncertainty. Solid line is true motion of the robot

Estimation sequence with a poor range sensor





Dashed line is trajectory estimated from motion control. Solid line is true motion of the robot

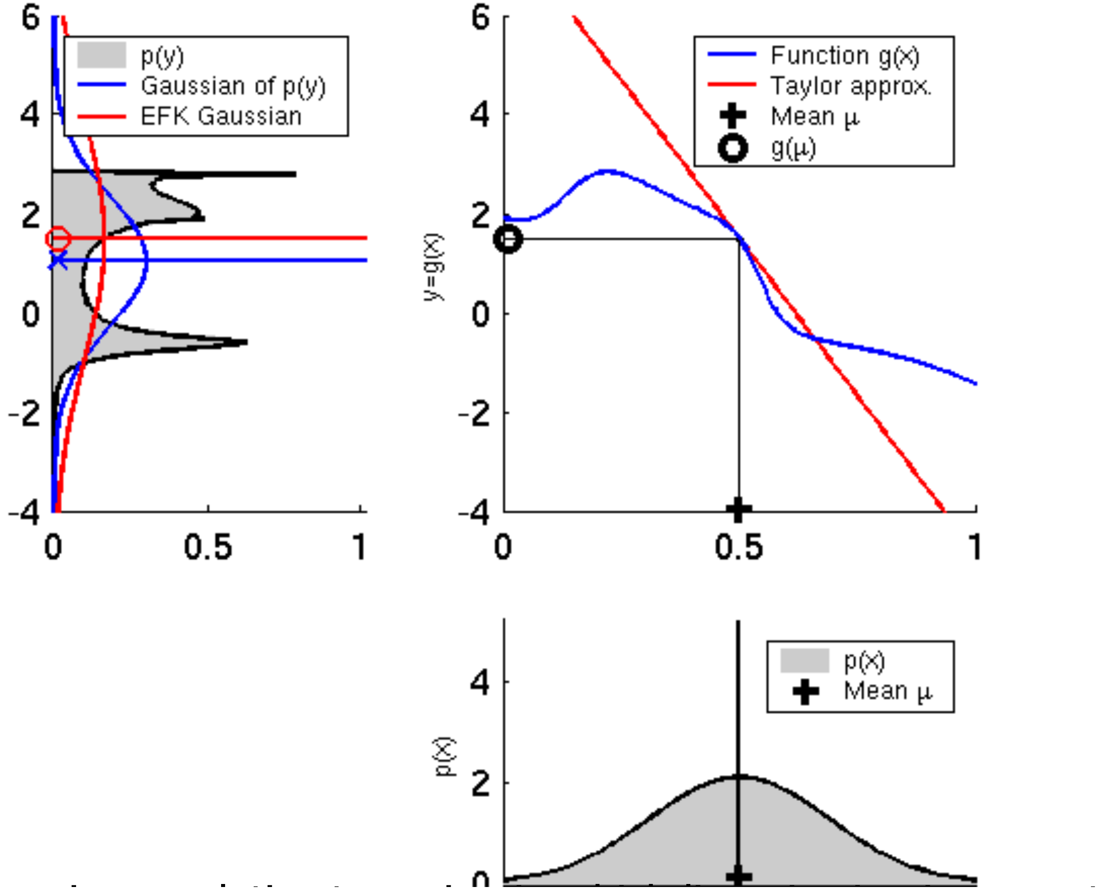
Dashed line is corrected robot trajectory along with predicted and corrected uncertainty. Solid line is true motion of the robot

Limitations of EKF

- If the non-linearity about the mean is small then the EKF approximation can be good.
- If the uncertainty in the state is more then the robot is affected more by the non-linearities in state transition.

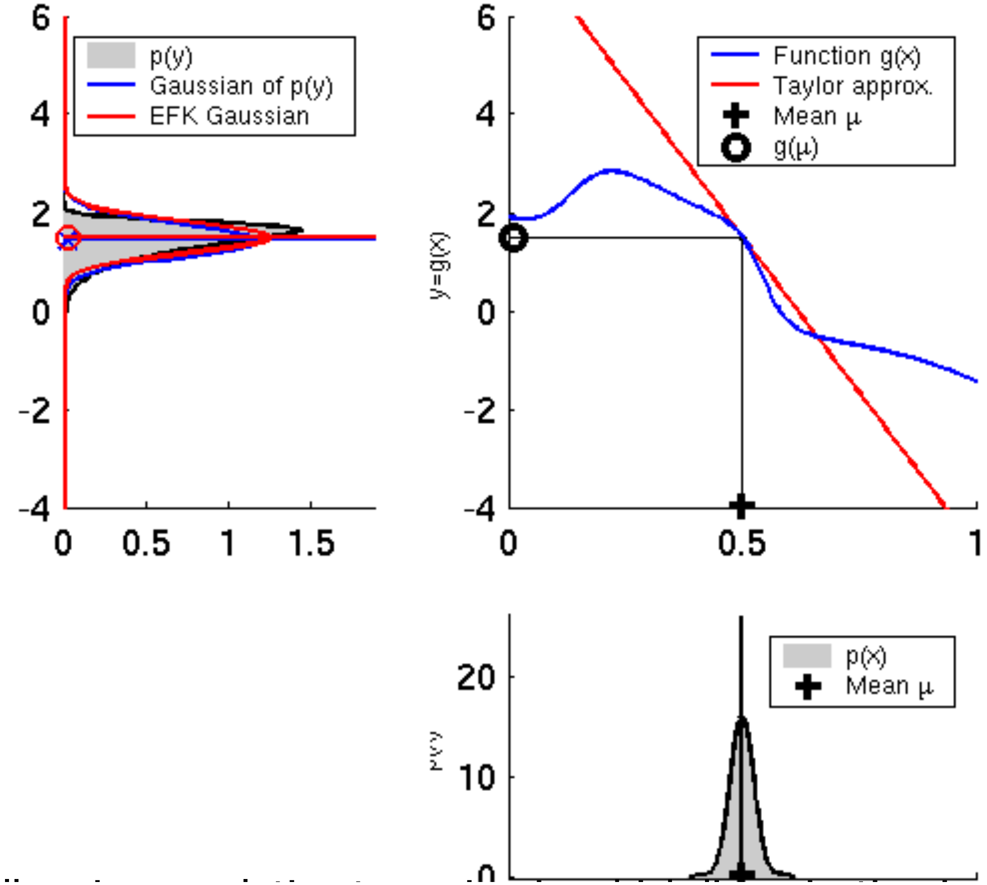
EKF Linearization

- Bel(x) has high variance



EKF Linearization

- Bel(x) has small variance



Unscented Kalman Filter

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

Is there a better way to linearize?

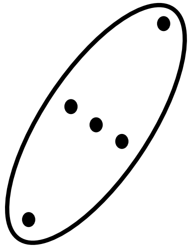
Unscented Transform

⬇

Unscented Kalman Filter (UKF)

Slide Credit: Cyrill Stachniss

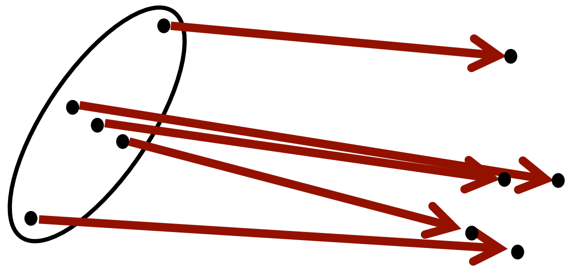
Unscented Transform



Compute a set of (so-called) sigma points

Slide Credit: Cyrill Stachniss

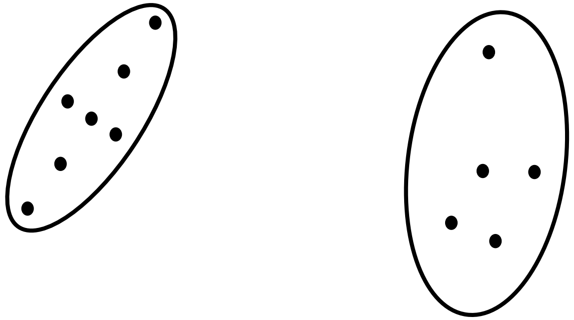
Unscented Transform



Transform each sigma point through the non-linear function

Slide Credit: Cyrill Stachniss

Unscented Transform



Compute Gaussian from the
transformed and weighted points

Slide Credit: Cyrill Stachniss

Unscented Transform

- 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 the EKF does**

Slide Credit: Cyrill Stachniss

Sigma Points

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

$$\begin{aligned} \sum_i w^{[i]} &= 1 \\ \mu &= \sum_i w^{[i]} \mathcal{X}^{[i]} \\ \Sigma &= \sum_i w^{[i]} (\mathcal{X}^{[i]} - \mu)(\mathcal{X}^{[i]} - \mu)^T \end{aligned}$$

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

Slide Credit: Cyrill Stachniss

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 vectors

dimensionality

scaling parameter

Slide Credit: Cyrill Stachniss

Matrix Square Root

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

$$
\begin{aligned}
\Sigma &= VDV^{-1} \\
&= V \begin{pmatrix} d_{11} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & d_{nn} \end{pmatrix} V^{-1} \\
&= V \begin{pmatrix} \sqrt{d_{11}} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \sqrt{d_{nn}} \end{pmatrix}^2 V^{-1}
\end{aligned}
$$

Slide Credit: Cyrill Stachniss

Matrix Square Root

- Thus, we can define

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

- so that

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

Slide Credit: Cyrill Stachniss

Sigma Points Weights

- Weight sigma points

for computing the mean

parameters

$$w_m^{[0]} = \frac{\lambda}{n+\lambda}$$

$$w_c^{[0]} = w_m^{[0]} + (1 - \alpha^2 + \beta)$$

$$w_m^{[i]} = w_c^{[i]} = \frac{1}{2(n+\lambda)} \quad \text{for } i = 1, \ldots, 2n$$

for computing the covariance

Slide Credit: Cyrill Stachniss

Recover Gaussian

- Compute Gaussian from weighted and transformed points

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

Slide Credit: Cyrill Stachniss

Recall EKF

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

$$\bar{\mu}_t = g(u_t, \mu_{t-1})$$

$$\bar{\Sigma}_t = G_t \; \Sigma_{t-1} \; G_t^T + R_t$$

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

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

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

$$return \; \mu_t, \Sigma_t$$

Slide Credit: Cyrill Stachniss

EKF to UKF

Unscented

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

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

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

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

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

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

7: $return\ \mu_t, \Sigma_t$

Slide Credit: Cyrill Stachniss

UKF Prediction

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

2: $\mathcal{X}_{t-1} = (\mu_{t-1} \qquad \mu_{t-1} + \gamma\sqrt{\Sigma_{t-1}} \qquad \mu_{t-1} - \gamma\sqrt{\Sigma_{t-1}})$

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

Slide Credit: Cyrill Stachniss

EKF to UKF

Unscented

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

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

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

use sigma point propagation for the expected observation and Kalman gain

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

6: $\Sigma_t = \bar{\Sigma}_t - K_t\, S_t\, K_t^T$

7: $return\ \mu_t, \Sigma_t$

Slide Credit: Cyrill Stachniss

UKF Correction

6: $\bar{\mathcal{X}}_t = (\bar{\mu}_t \qquad \bar{\mu}_t + \gamma\sqrt{\bar{\Sigma}_t} \qquad \bar{\mu}_t - \gamma\sqrt{\bar{\Sigma}_t})$

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)

Slide Credit: Cyrill Stachniss

UKF Correction

6: $\bar{\mathcal{X}}_t = (\bar{\mu}_t \qquad \bar{\mu}_t + \gamma\sqrt{\bar{\Sigma}_t} \qquad \bar{\mu}_t - \gamma\sqrt{\bar{\Sigma}_t})$

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: $return \; \mu_t, \Sigma_t$

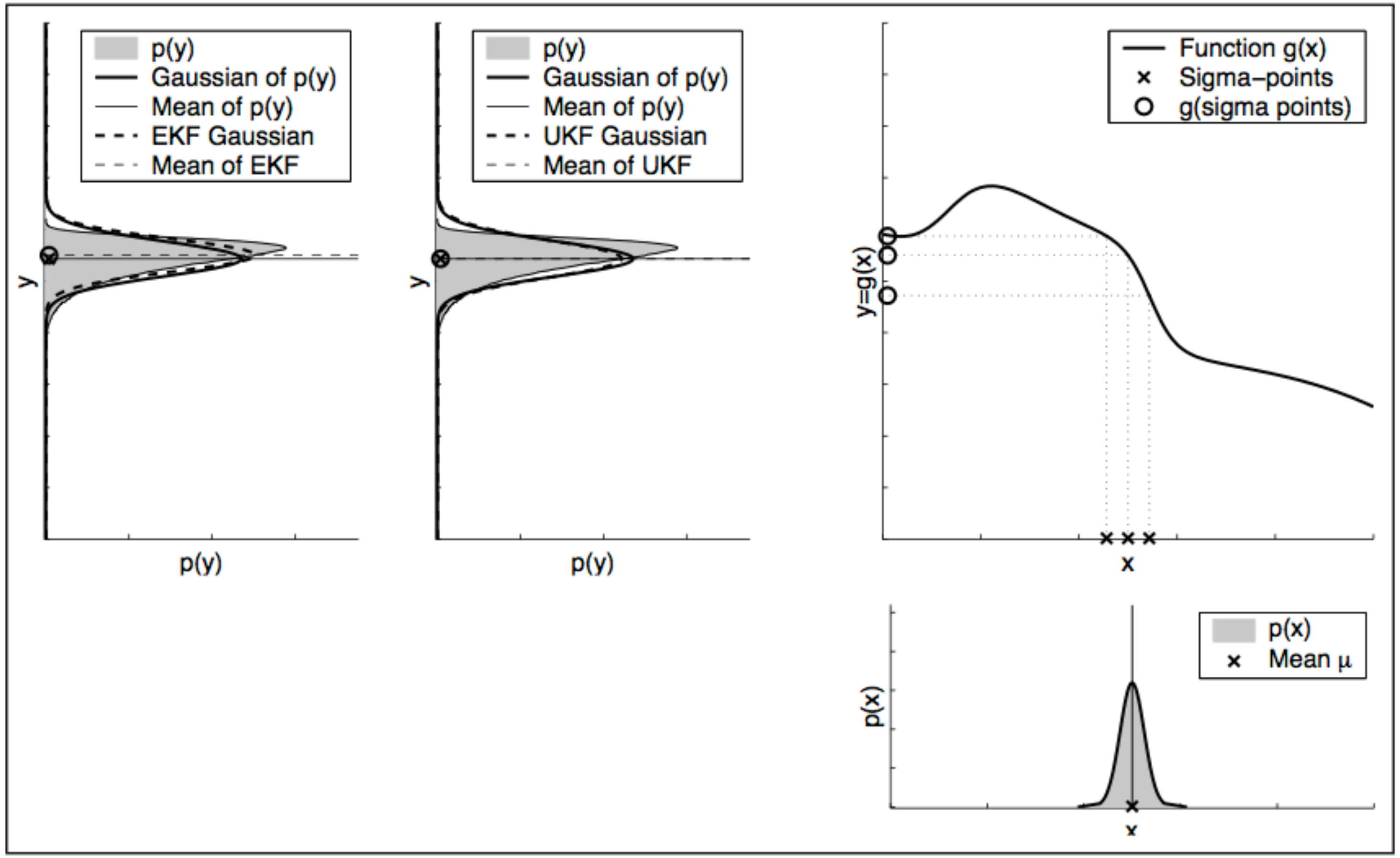
Slide Credit: Cyrill Stachniss

Computation of Posterior Covariance

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

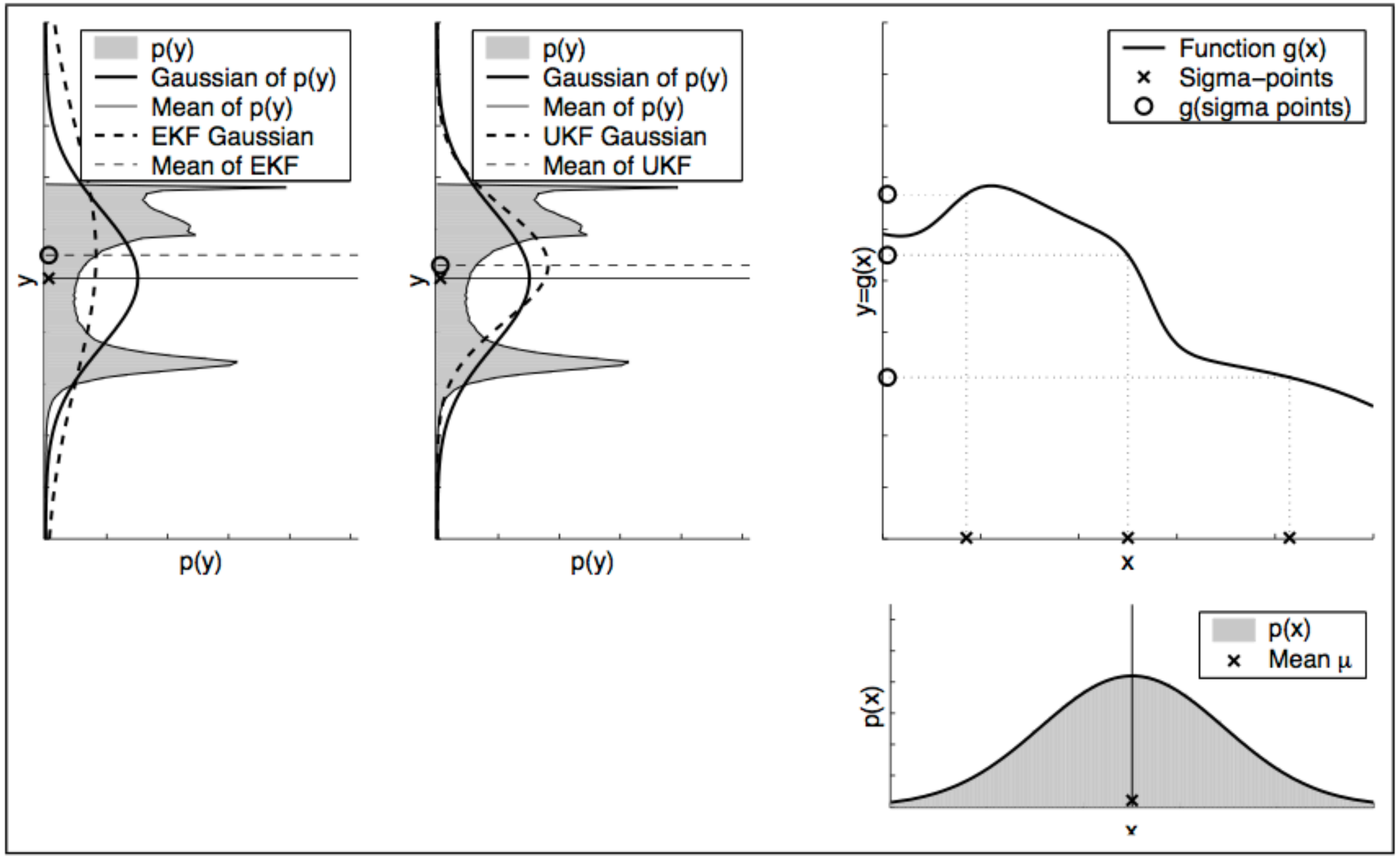
Slide Credit: Cyrill Stachniss

EKF vs UKF : Small Variance



Slide Credit: Cyrill Stachniss

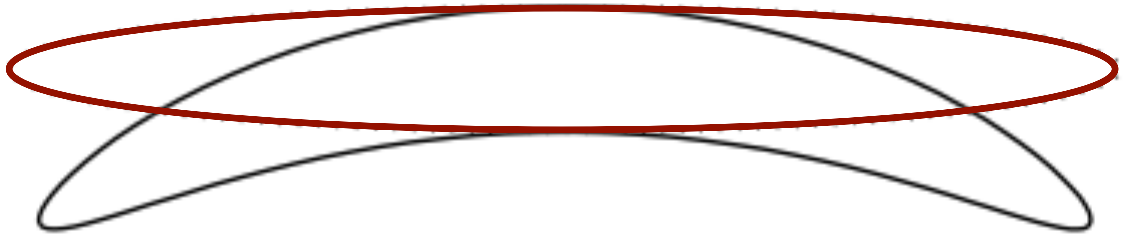
EKF vs UKF : Large Variance



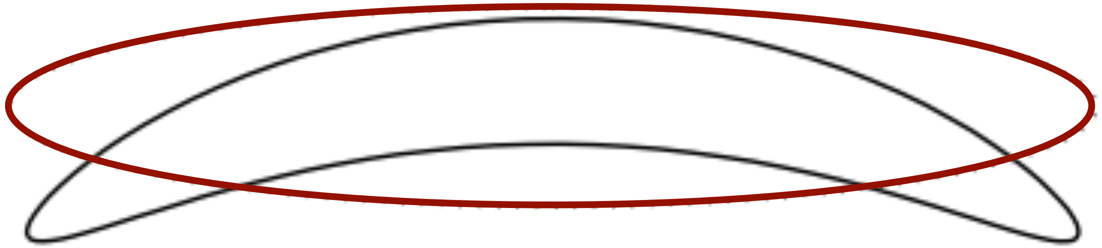
Slide Credit: Cyrill Stachniss

EKF vs UKF for Non-Linear Functions

EKF approximation

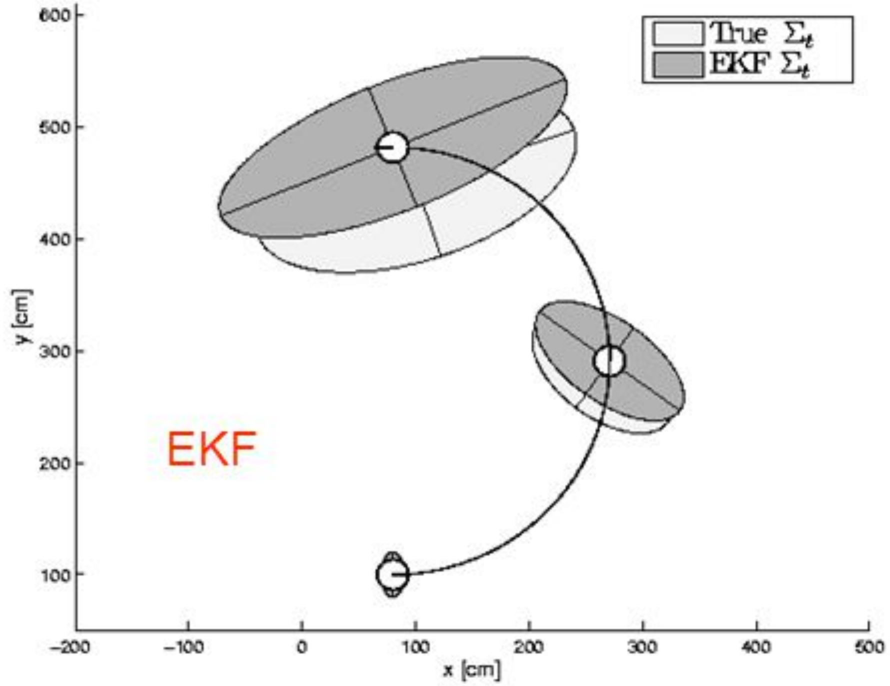


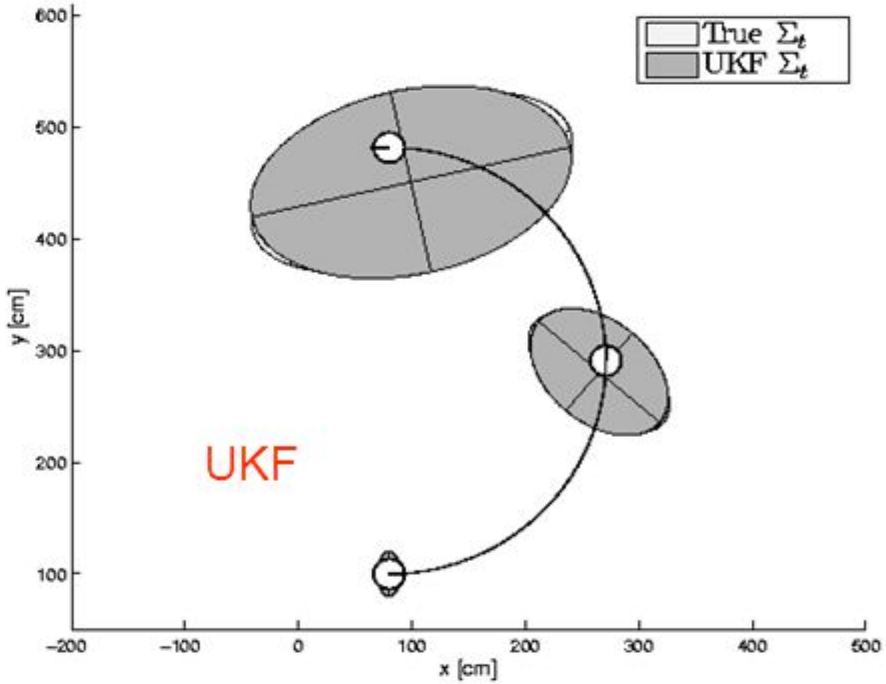
UKF approximation



Slide Credit: Cyrill Stachniss

EKF vs UKF (Prediction Quality)





Slide Credit: Cyrill Stachniss

UKF vs EKF (Summary)

- 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
- Both assumes Gaussian distributions

Slide Credit: Cyrill Stachniss