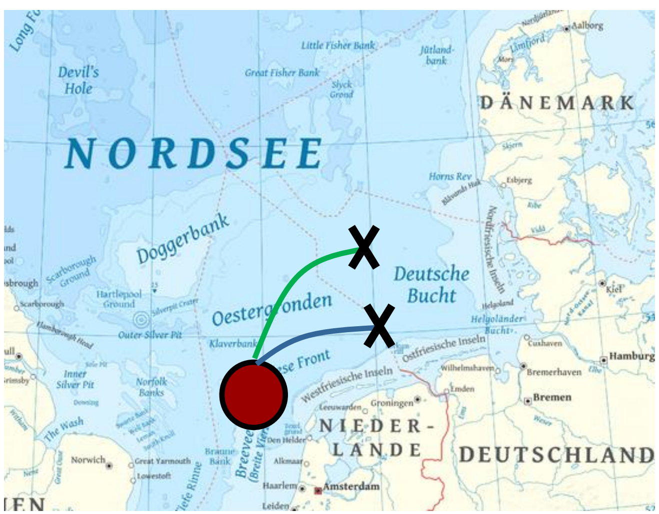
Extended Kalman Filter Localization

Nived Chebrolu

Slides adapted from Cyrill Stachniss at Uni. Bonn

Localization Example



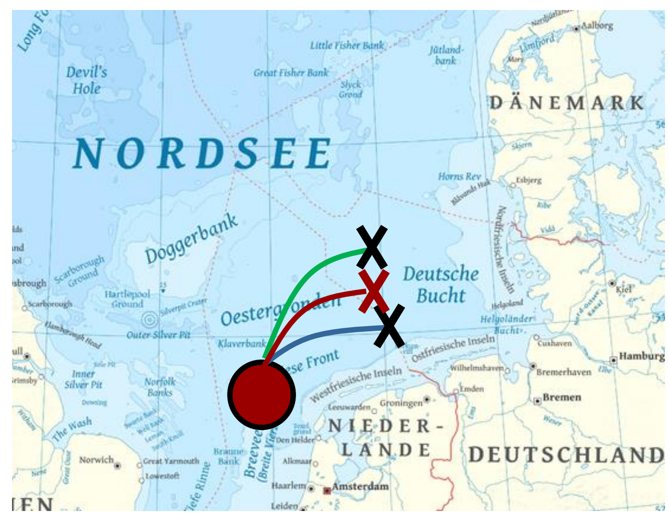


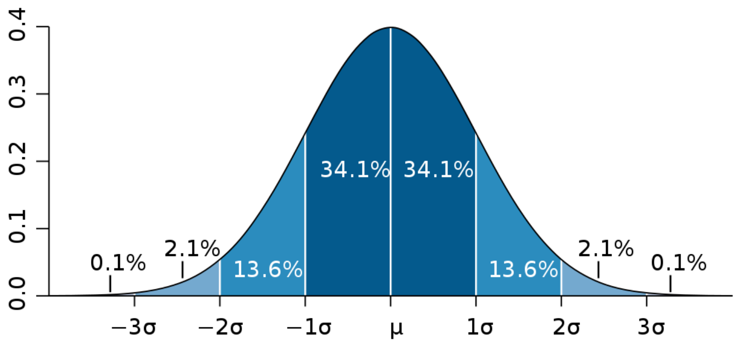
Image courtesy: Wikipedia

Kalman Filter

- It is a Bayes filter
- Everything is Gaussian

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

- Optimal solution for linear models and Gaussian distributions



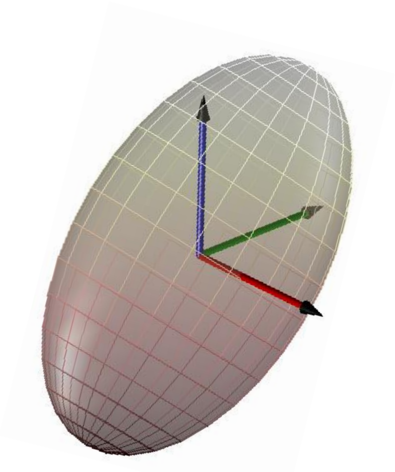
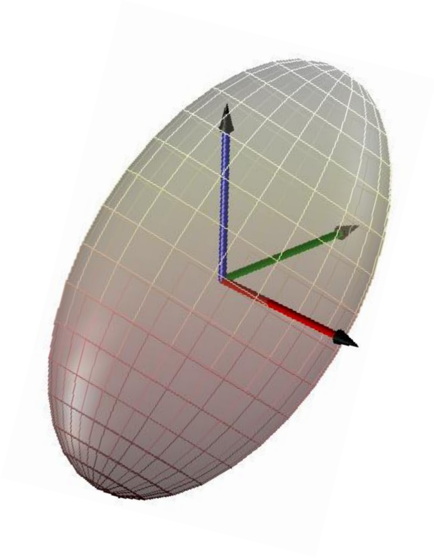


Image courtesy: K. Arras

Kalman Filter Assumptions

- Gaussian distributions and noise
- Linear motion and observation model



$$x_t = A_t x_{t-1} + B_t u_t + \epsilon_t$$

$$z_t = C_t x_t + \delta_t$$

What if this is not the case?

Non-Linear Dynamic Systems

- Most realistic problems involve nonlinear functions

$$\cancel{x_t = A_t x_{t-1} + B_t u_t + \epsilon_t} \qquad \cancel{z_t = C_t x_t + \delta_t}$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$x_t = g(u_t, x_{t-1}) + \epsilon_t \qquad z_t = h(x_t) + \delta_t$$

Linearized Motion Model

- The linearized model leads to

$$p(x_t \mid u_t, x_{t-1}) \approx \det\left(2\pi R_t\right)^{-\frac{1}{2}}$$
$$\exp\Big(-\frac{1}{2}\left(x_t - g(u_t, \mu_{t-1}) - G_t\left(x_{t-1} - \mu_{t-1}\right)\right)^T$$
$$R_t^{-1}\left(x_t - \underbrace{g(u_t, \mu_{t-1}) - G_t\left(x_{t-1} - \mu_{t-1}\right)}_{\text{linearized model}}\right)\Big)$$

- R_t describes the noise of the motion

Linearized Observation Model

- The linearized model leads to

$$p(z_t \mid x_t) = \det\left(2\pi Q_t\right)^{-\frac{1}{2}}$$
$$\exp\left(-\frac{1}{2}\left(z_t - h(\bar{\mu}_t) - H_t\left(x_t - \bar{\mu}_t\right)\right)^T\right.$$
$$\left.Q_t^{-1}\left(z_t - \underbrace{h(\bar{\mu}_t) - H_t\left(x_t - \bar{\mu}_t\right)}_{\text{linearized model}}\right)\right)$$

- Q_t describes the measurement noise

Extended Kalman Filter Algorithm

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

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

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

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

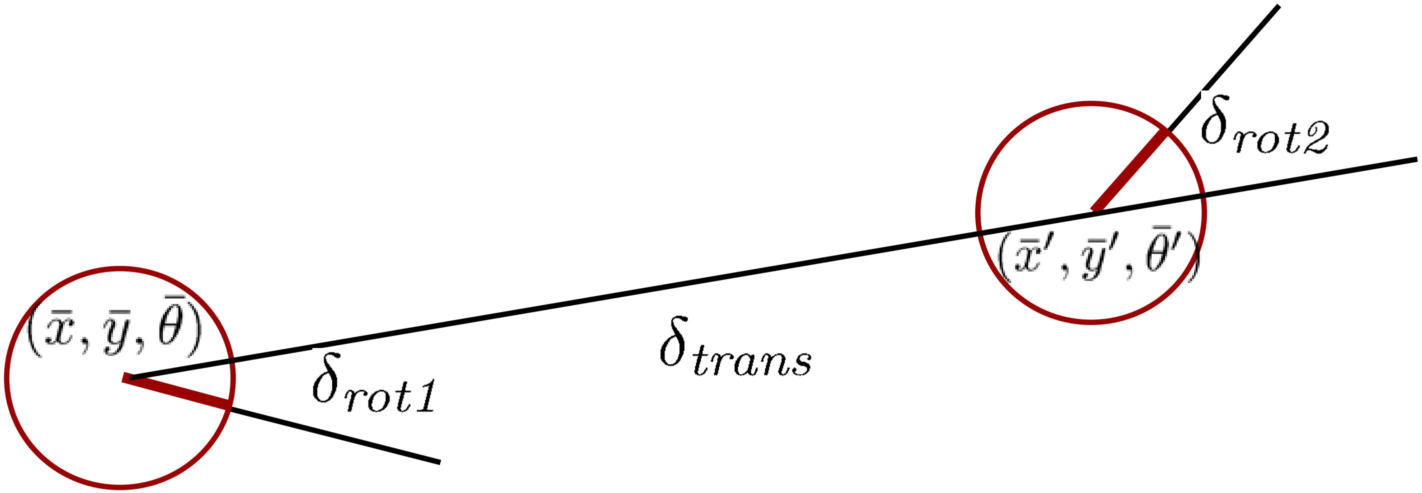
5: $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

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

7: $return \ \mu_t, \Sigma_t$

EKF Localization For Feature-based Map

Odometry As Controls



$$u = (\delta_{rot1}, \delta_{trans}, \delta_{rot2})$$

Motion Model

- Odometry motion model

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \underbrace{\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \delta_{\text{trans}} \cos(\theta + \delta_{rot_1}) \\ \delta_{\text{trans}} \sin(\theta + \delta_{rot_1}) \\ \delta_{rot_1} + \delta_{rot_2} \end{pmatrix}}_{g(u_t, x_{t-1})} + \mathcal{N}(0, R_t)$$

- Linearization

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1})$$

Jacobian For Motion Model

- Compute Jacobian

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

$$G_t = \begin{pmatrix} 1 & 0 & -\delta_{\text{trans}} \sin(\theta + \delta_{rot_1}) \\ 0 & 1 & \delta_{\text{trans}} \cos(\theta + \delta_{rot_1}) \\ 0 & 0 & 1 \end{pmatrix}$$

Observation Model

- Range-Bearing Model

$$z_t^i = \begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix} = \underbrace{\begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \operatorname{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta \end{pmatrix}}_{h(x_t, m)} + \mathcal{N}(0, Q_t)$$

- Linearization

$$h(x_t, m) \approx h(\bar{\mu}_t) + H_t^i (x_t - \bar{\mu}_t)$$

Jacobian For Observation Model

- Compute Jacobian

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

$$H_t^i = \begin{pmatrix} -\frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q}} & -\frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{q}} & 0 \\ \frac{m_{j,y} - \bar{\mu}_{t,y}}{q} & -\frac{m_{j,x} - \bar{\mu}_{t,x}}{q} & -1 \end{pmatrix}$$

$$\text{where} \quad q = (m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2$$

Extended Kalman Filter Algorithm

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

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

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

Prediction

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: $return \; \mu_t, \Sigma_t$

Correction

EKF Localization Algorithm

1. **Algorithm EKF localization** $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, c_t, m)$:

2. $\theta = \mu_{t-1,\theta}$

3. $G_t = \begin{pmatrix} 1 & 0 & -\delta_{\text{trans}} \sin(\theta + \delta_{\text{rot}_1}) \\ 0 & 1 & \delta_{\text{trans}} \cos(\theta + \delta_{\text{rot}_1}) \\ 0 & 0 & 1 \end{pmatrix}$

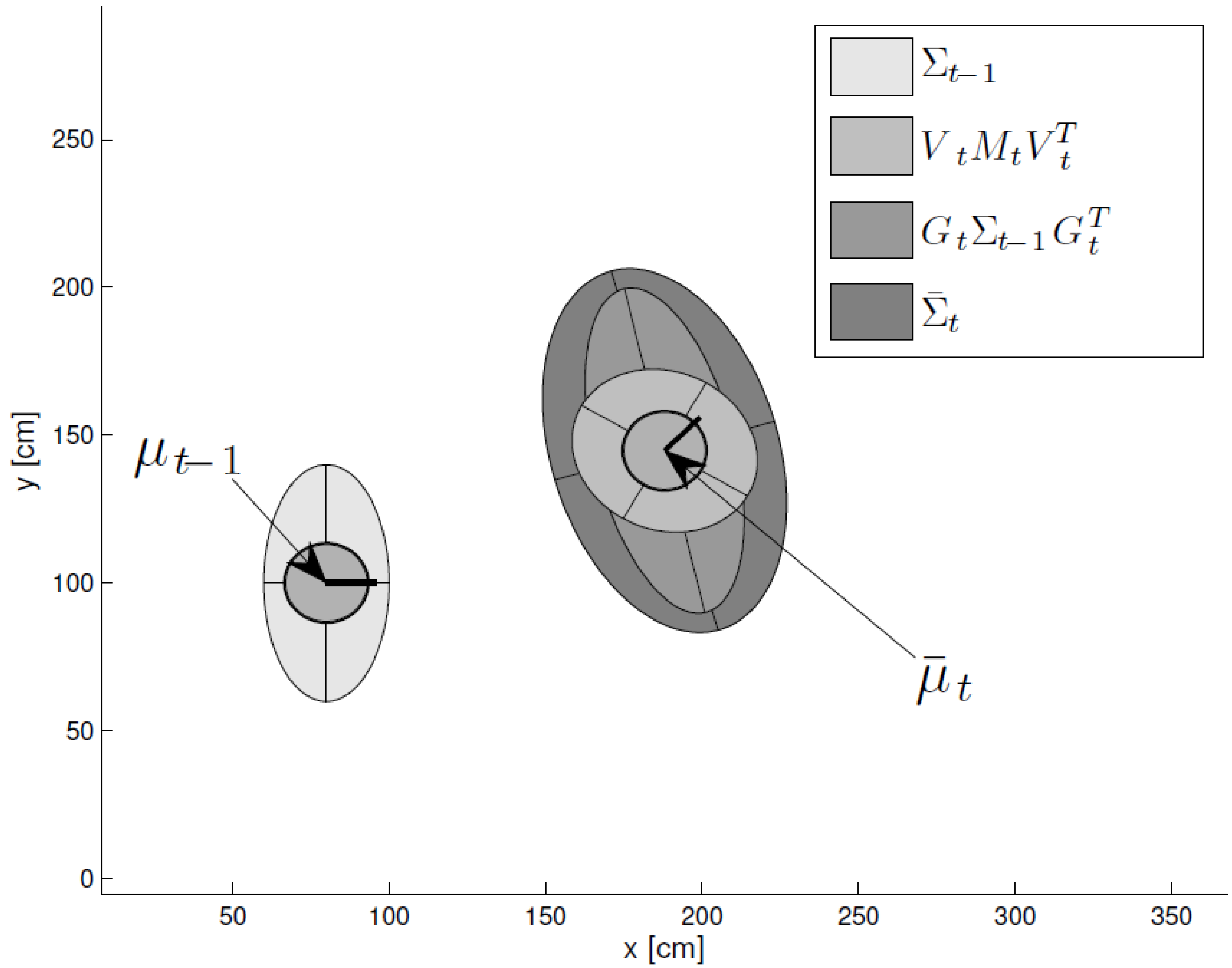
4. $V_t = \begin{pmatrix} -\delta_{\text{trans}} \sin(\theta + \delta_{\text{rot}_1}) & \cos(\theta + \delta_{\text{rot}_1}) & 0 \\ \delta_{\text{trans}} \cos(\theta + \delta_{\text{rot}_1}) & \sin(\theta + \delta_{\text{rot}_1}) & 0 \\ 1 & 0 & 1 \end{pmatrix}$

5. $M_t = \begin{pmatrix} \alpha_1 \delta_{\text{rot}_1}^2 + \alpha_2 \delta_{\text{trans}}^2 & 0 & 0 \\ 0 & \alpha_3 \delta_{\text{trans}}^2 + \alpha_4 (\delta_{\text{rot}_1}^2 + \delta_{\text{rot}_2}^2) & 0 \\ 0 & 0 & \alpha_1 \delta_{\text{rot}_2}^2 + \alpha_2 \delta_{\text{trans}}^2 \end{pmatrix}$

6. $\bar{\mu}_t = \mu_{t-1} + \begin{pmatrix} \delta_{\text{trans}} \cos(\theta + \delta_{\text{rot}_1}) \\ \delta_{\text{trans}} \sin(\theta + \delta_{\text{rot}_1}) \\ \delta_{\text{rot}_1} + \delta_{\text{rot}_2} \end{pmatrix}$ ⟵ **Prediction**

7. $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T$

Prediction Step



Courtesy: Thrun, Burgard, Fox

EKF Localization Algorithm

$$Q_t = \begin{pmatrix} {\sigma_r}^2 & 0 \\ 0 & {\sigma_\phi}^2 \end{pmatrix}$$

$\textbf{for}$ all observed features $z_t^i = (r_t^i, \phi_t^i)^T$ $\textbf{do}$

$$j = c_t^i$$

$$q = (m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2$$

$$\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(m_{j,y} - \bar{\mu}_{t,y}, m_{j,x} - \bar{\mu}_{t,x}) - \bar{\mu}_{t,\theta} \end{pmatrix}$$

$$H_t^i = \begin{pmatrix} -\frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q}} & -\frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{q}} & 0 \\ \frac{m_{j,y} - \bar{\mu}_{t,y}}{q} & -\frac{m_{j,x} - \bar{\mu}_{t,x}}{q} & -1 \end{pmatrix}$$

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

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

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

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

$\textbf{end for}$

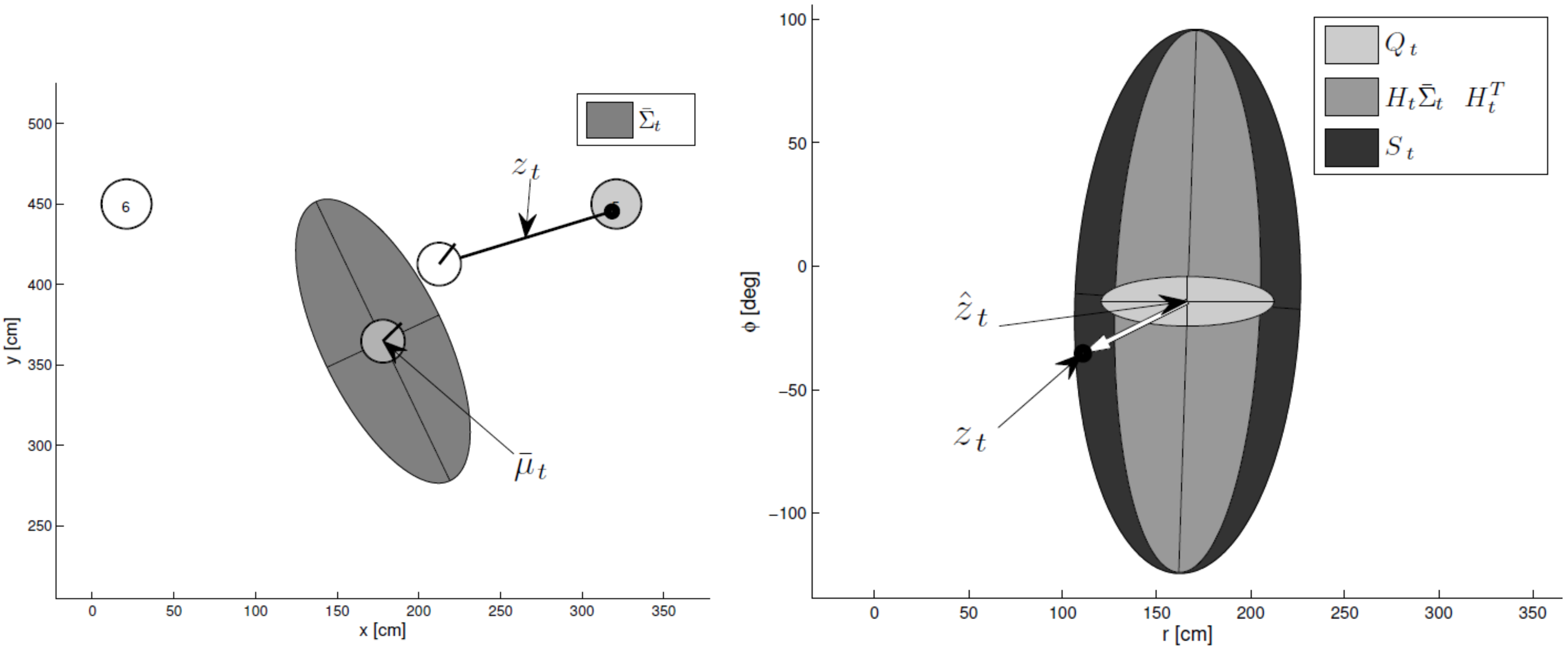
$$\mu_t = \bar{\mu}_t$$

$$\Sigma_t = \bar{\Sigma}_t$$

$\textbf{return}$ μ_t, Σ_t

Expected Measurement

Expected Measurement



Courtesy: Thrun, Burgard, Fox

EKF Localization Algorithm

$Q_t = \begin{pmatrix} {\sigma_r}^2 & 0 \\ 0 & {\sigma_\phi}^2 \end{pmatrix}$

for all observed features $z_t^i = (r_t^i, \phi_t^i)^T$ **do**

$\quad j = c_t^i$

$\quad q = (m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2$

$\quad \hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(m_{j,y} - \bar{\mu}_{t,y}, m_{j,x} - \bar{\mu}_{t,x}) - \bar{\mu}_{t,\theta} \end{pmatrix}$

$\quad H_t^i = \begin{pmatrix} -\frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q}} & -\frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{q}} & 0 \\ \frac{m_{j,y} - \bar{\mu}_{t,y}}{q} & -\frac{m_{j,x} - \bar{\mu}_{t,x}}{q} & -1 \end{pmatrix}$

$\quad S_t^i = H_t^i \bar{\Sigma}_t {H_t^i}^T + Q_t$

$\quad K_t^i = \bar{\Sigma}_t [H_t^i]^T [S_t^i]^{-1}$

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

$\quad \bar{\Sigma}_t = (I - K_t^i H_t^i) \bar{\Sigma}_t$

end for

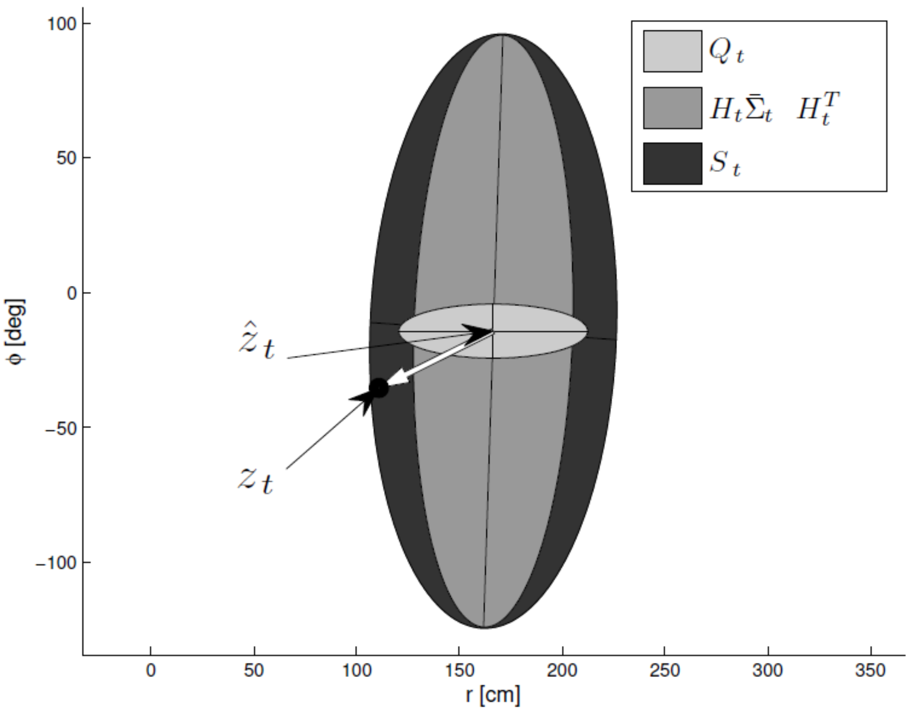
$\mu_t = \bar{\mu}_t$

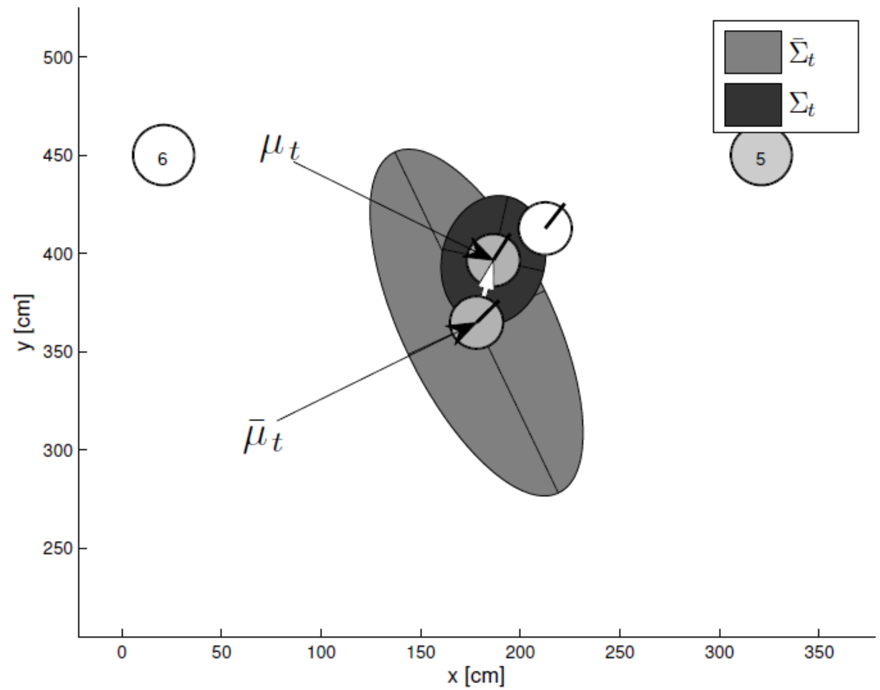
$\Sigma_t = \bar{\Sigma}_t$

return μ_t, Σ_t

Correction

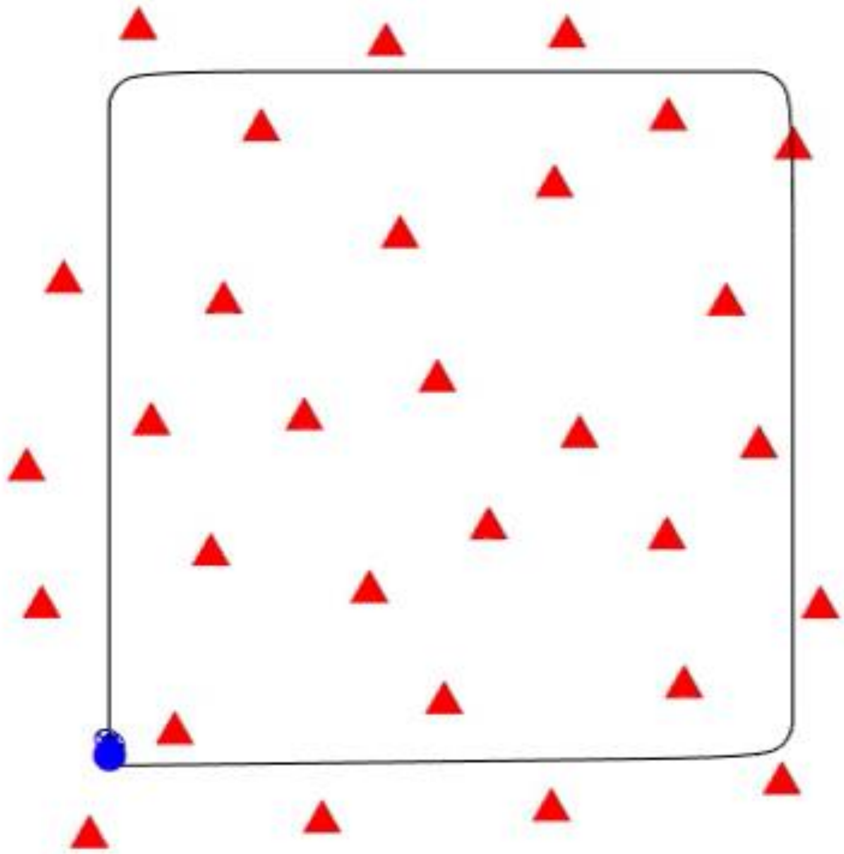
Correction Step





Courtesy: Thrun, Burgard, Fox

2D EKF Localization Example



Extended Kalman Filter Summary

- Extension of the Kalman filter
- One way to handle the non-linearities
- Performs local linearizations
- Works well in practice for moderate non-linearities
- Large uncertainty leads to increased approximation error

Literature

Kalman Filter and EKF

- Thrun et al.: "Probabilistic Robotics", Chapter 3
- Thrun et al.: "Probabilistic Robotics", Chapter 7
- Schön and Lindsten: "Manipulating the Multivariate Gaussian Density"