

9 SLAM – Simultaneous Localization and Mapping

Probleme

- Sowohl Positionen der Landmarken als auch Roboterweg sind unbekannt
- Die Zuordnung von Sensordaten zu Landmarken sind in der Regel unbekannt
- Roboter muss entscheiden, ob Sensordaten zu einer bereits beobachteten Landmarke zugeordnet werden können oder zu einer noch nichtgesehenen Landmarke
- Zuordnungsproblematik wird durch Fehler im Roboterweg verstärkt

SLAM – Simultaneous Localization and Mapping

$$p(x_t, m \mid z_{1:t}, u_{1:t})$$

Online SLAM Problem

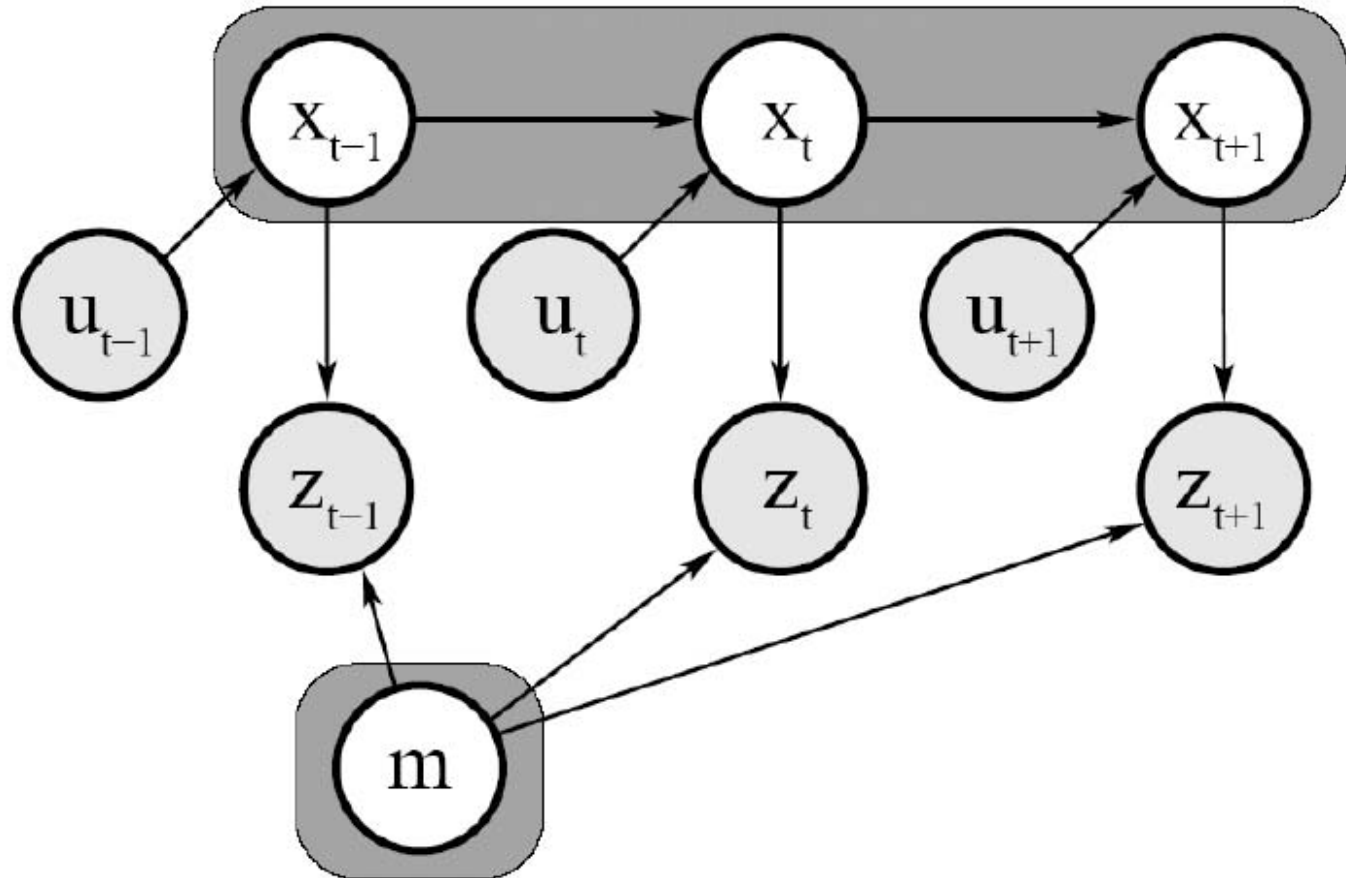
$$p(x_t, m, c_t \mid z_{1:t}, u_{1:t})$$

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

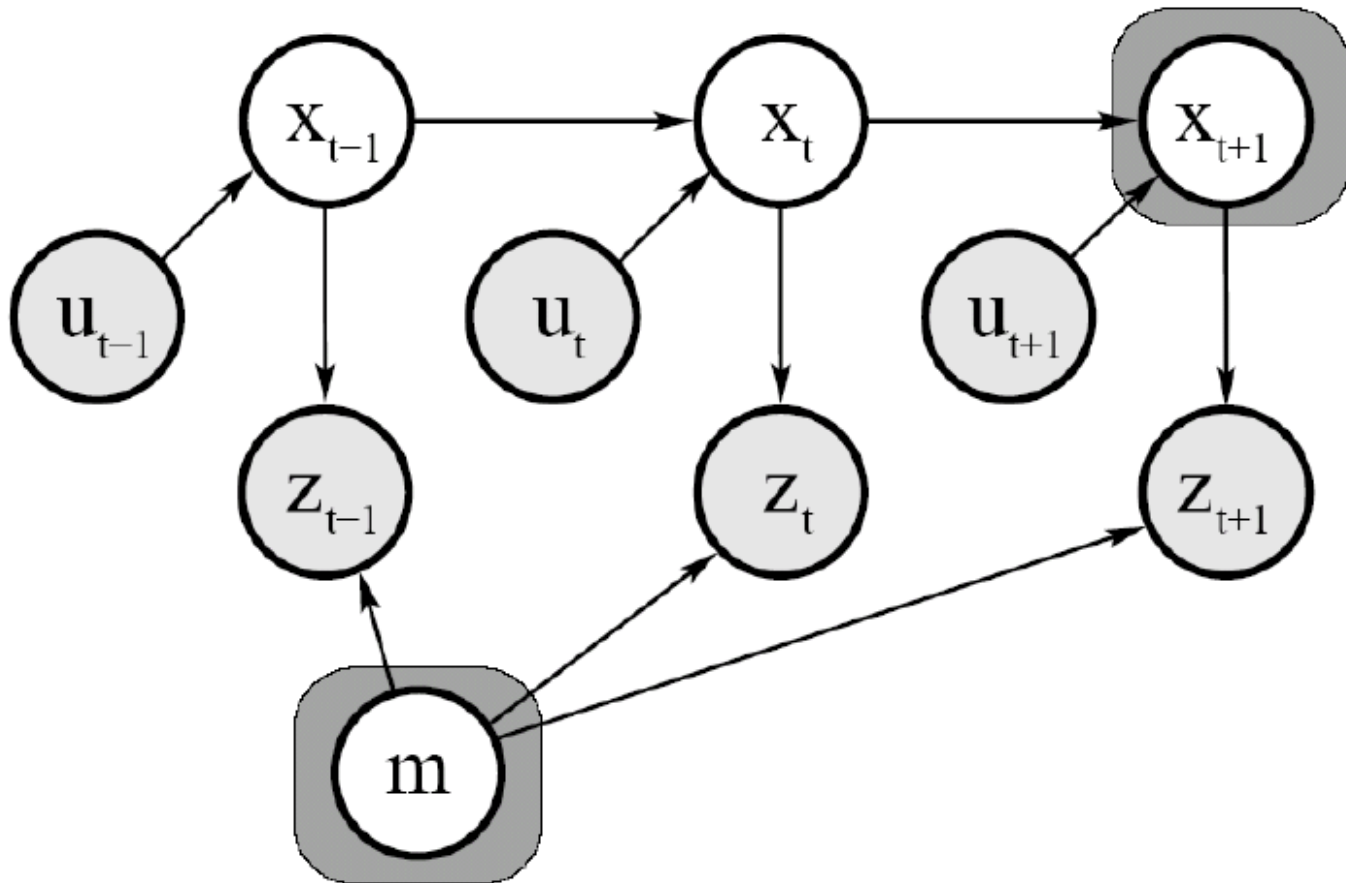
Full SLAM Problem

$$p(x_{1:t}, m, c_{1:t} \mid z_{1:t}, u_{1:t})$$

Full SLAM



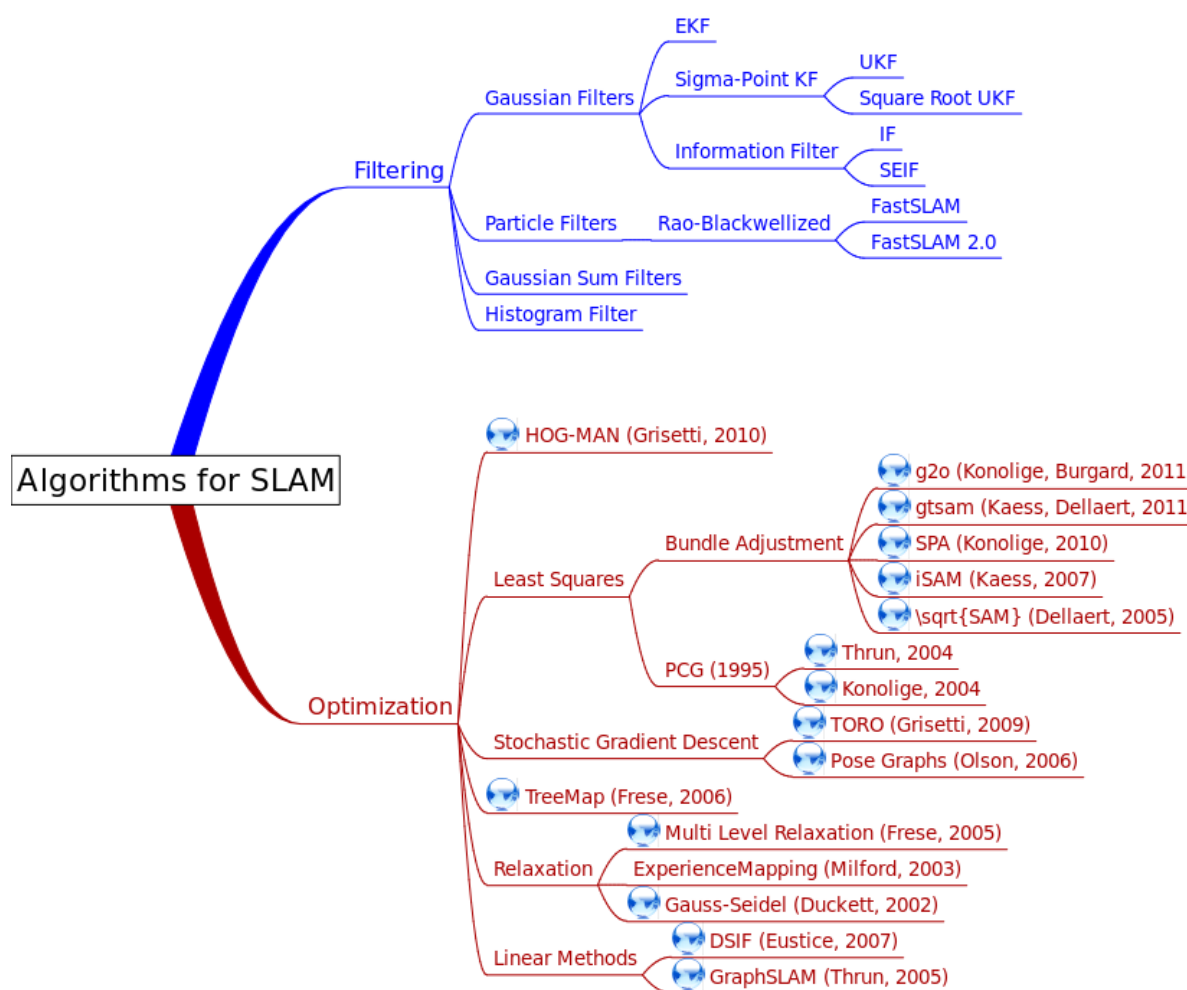
Online SLAM



Verfahren

- EKF SLAM
 - verwendet Extended Kalman Filter
 - Online
- Graph SLAM
- SEIF SLAM
 - sparse extended information filter
 - effizienter Online Algorithmus
- Fast SLAM
 - verwendet Partikel Filter

Verfahren – Niko Sünderhauf



9.1 Simple SLAM – Einfaches Modell

Simple SLAM

Roboterposition

$$(x_r, y_r)$$

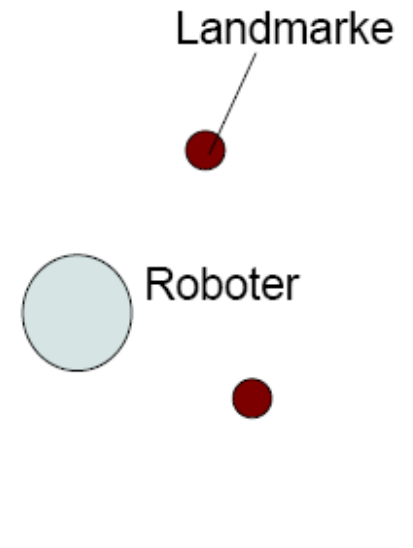
N Landmarken (Positionen):

$$(x_{l_1}, y_{l_1})$$

$$(x_{l_2}, y_{l_2})$$

...

$$(x_{l_N}, y_{l_N})$$



Simple SLAM - Zustand

$$\mathbf{x}_t = (x_r(t), y_r(t), x_{l_1}, y_{l_1}, \dots, x_{l_i}, y_{l_i}, \dots, x_{l_N}, y_{l_N})^T$$

Bewegung

Omnidirektionales Bewegungsmodell

(Antrieb in alle Richtungen)

(Roboterfront zeigt immer in gleiche Richtung)

$$\mathbf{u}_t = (v_x(t), v_y(t))^T$$

Simple SLAM – Bewegungsgleichung

$$\mathbf{x}_t = A_t \cdot \mathbf{x}_{t-1} + B_t \cdot \mathbf{u}_t + \boldsymbol{\varepsilon}_t$$

$$A_t = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} \quad B_t = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix}$$

Landmarken bewegen sich nicht

$$\boldsymbol{\varepsilon}_t = (\varepsilon_{rx}(t), \varepsilon_{ry}(t), 0, 0, \dots, 0)^T$$

$$(\varepsilon_{rx}(t), \varepsilon_{ry}(t))^T \rightarrow N(0, V_{rt})$$

$$\boldsymbol{\varepsilon}_t \rightarrow N(0, V_t)$$

$$V_t = \begin{pmatrix} V_{rt} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

Simple SLAM – Sensordaten

Landmarken seien eindeutig identifizierbar, für jede Landmarke wird ihre Position relativ zum Roboter gemessen

$$\mathbf{z}_t = \begin{pmatrix} z_1(t) \\ \vdots \\ z_i(t) \\ \vdots \\ z_N(t) \end{pmatrix} = \begin{pmatrix} x_{l_1} - x_r(t) \\ y_{l_1} - y_r(t) \\ \vdots \\ x_{l_N} - x_r(t) \\ y_{l_N} - y_r(t) \end{pmatrix}$$

Simple SLAM - Messgleichung

$$\mathbf{z}_t = H_t \cdot \mathbf{x}_t + \delta_t$$

$$H_t = \begin{pmatrix} H_1(t) \\ H_2(t) \\ \vdots \\ H_N(t) \end{pmatrix}$$

$$H_i = \begin{pmatrix} -1 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \end{pmatrix}$$

(2i+1)-te Spalte



Simple SLAM – Messgleichung

$$\delta_t \rightarrow N(0, W_t)$$

$$\delta_t = (\delta_1(t), \delta_2(t), \dots, \delta_N(t))^T$$

$$\mathbf{z}_t = H_t \cdot \mathbf{x}_t + \delta_t$$

$$\delta_i(t) \rightarrow N(0, W_{it})$$

$$W_t = \begin{pmatrix} W_{1t} & 0 & \dots & 0 \\ 0 & W_{2t} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & W_{Nt} \end{pmatrix}$$

Simple SLAM – Kalmanfilter

- Damit sind alle alle Systemparameter (lineares System!) festgelegt und der lineare Kalman-Filter-Algorithmus kann direkt verwendet werden.

Simple SLAM – Erweiterungen

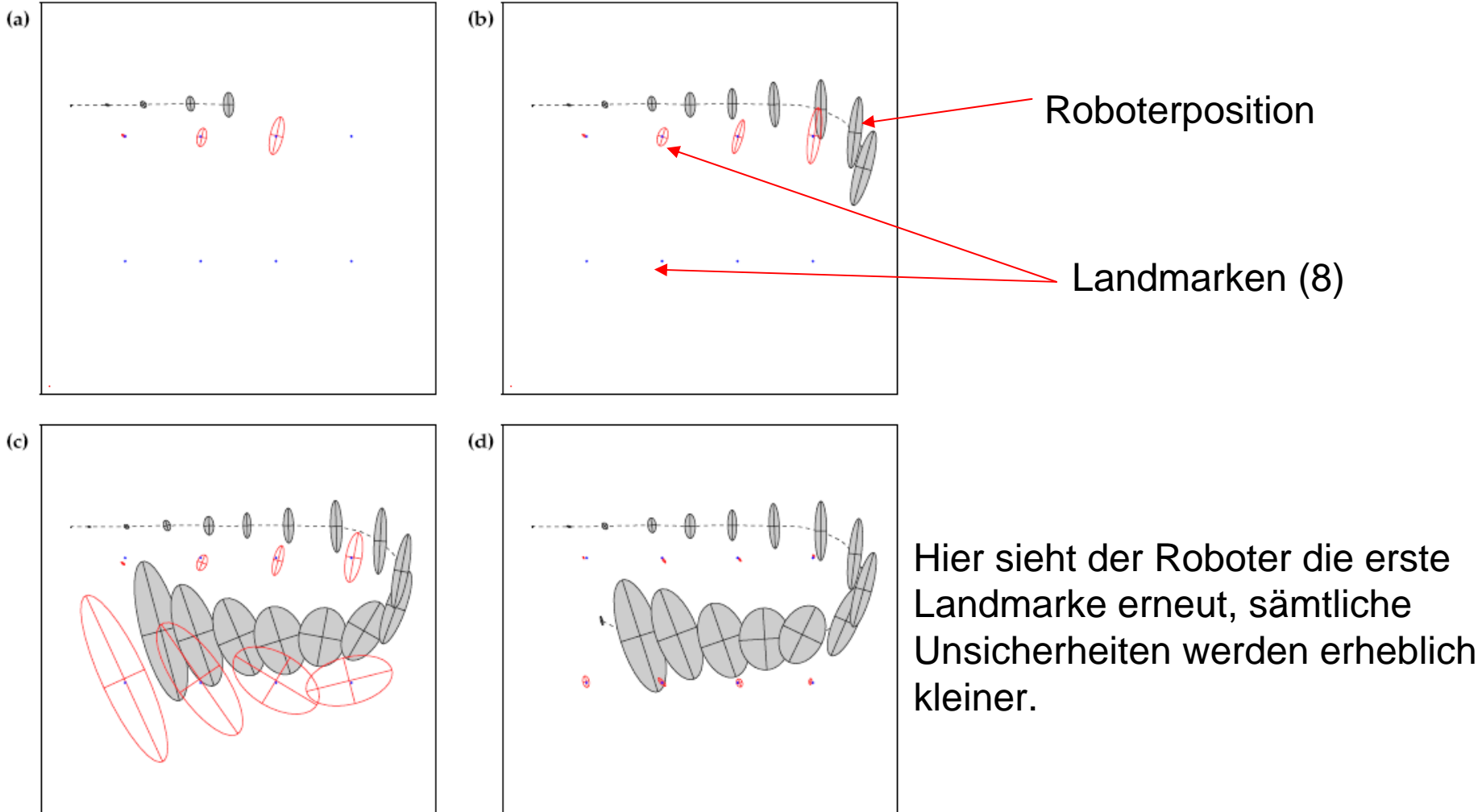
- Das Bewegungsmodell kann erweitert werden. Zur Roboterposition kommt dann noch die Orientierung dazu.
- Statt bei jeder Landmarke die relative Position zu messen, kann auch Abstand und Orientierung gemessen werden.
- Zusätzlich kann noch das Problem der Zuordnung der Messdaten zu den Landmarken-Nummern dazukommen.
- EKF SLAM

9.2 EKF SLAM

EKF – SLAM

- erster SLAM Algorithmus
- verwendet erweiterten Kalmanfilter
- Online SLAM
- bekannte Korrespondenzen zu den Landmarken
- unbekannte Korrespondenzen zu den Landmarken

Beispiel



EKF SLAM

zu berechnen:

$$p(y_t | z_{1:t}, u_{1:t})$$

$$y_t = \begin{pmatrix} x_t \\ m \end{pmatrix}$$

$$y_t = (x, y, \theta, m_{1,x}, m_{1,y}, s_1, \dots, m_{N,x}, m_{N,y}, s_N)^T$$

EKF SLAM

$$\mu_0 = (0, 0, \dots, 0)^T \quad 3N + 3 - \text{Vektor}$$

$$\Sigma_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \infty & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \infty \end{pmatrix}$$

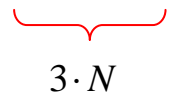
$(3N + 3) \times (3N + 3)$ - Matrix

EKF SLAM

$$\mathbf{y}_t = \mathbf{y}_{t-1} + F_x^T \cdot \begin{pmatrix} -\frac{v_t}{w_t} \cdot \sin \theta + \frac{v_t}{w_t} \cdot \sin(\theta + w_t \cdot \Delta t) \\ \frac{v_t}{w_t} \cdot \cos \theta - \frac{v_t}{w_t} \cdot \cos(\theta + w_t \cdot \Delta t) \\ w_t \cdot \Delta t \end{pmatrix} + N(0, F_x^T \cdot R_t \cdot F_x)$$

$$\mathbf{y}_t = g(\mathbf{u}_t, \mathbf{y}_{t-1}) + N(0, F_x^T \cdot R_t \cdot F_x)$$

$$F_x = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \end{pmatrix}$$


3 · N

EKF SLAM

$$\mathbf{y}_t = g(\mathbf{u}_t, \mathbf{y}_{t-1}) + N(0, F_x^T \cdot R_t \cdot F_x)$$

$$g(\mathbf{u}_t, \mathbf{y}_{t-1}) \approx g(\mathbf{u}_t, \mu_{t-1}) + G_t \cdot (\mathbf{y}_{t-1} - \mu_{t-1})$$

$$G_t = I + F_x^T \cdot g_t \cdot F_x$$

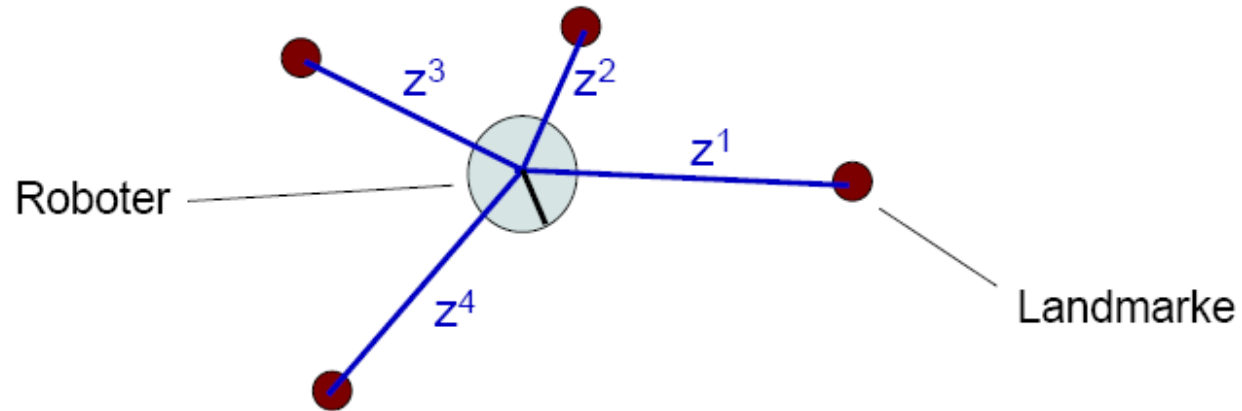
$$g_t = \begin{pmatrix} 0 & 0 & \frac{v_t}{w_t} (-\cos(\mu_{t-1,\theta}) + \cos(\mu_{t-1,\theta} + w_t \Delta t)) \\ 0 & 0 & \frac{v_t}{w_t} (-\sin(\mu_{t-1,\theta}) + \sin(\mu_{t-1,\theta} + w_t \Delta t)) \\ 0 & 0 & 0 \end{pmatrix}$$

EKF SLAM

$$\bar{\boldsymbol{\mu}}_t = g(\mathbf{u}_t, \boldsymbol{\mu}_{t-1}) = \boldsymbol{\mu}_{t-1} + F_x^T \cdot \begin{pmatrix} -\frac{v_t}{w_t} \cdot \sin \theta + \frac{v_t}{w_t} \cdot \sin(\theta + w_t \cdot \Delta t) \\ \frac{v_t}{w_t} \cdot \cos \theta - \frac{v_t}{w_t} \cdot \cos(\theta + w_t \cdot \Delta t) \\ w_t \cdot \Delta t \end{pmatrix} \quad \theta = \mu_{t-1, \theta}$$

$$\bar{\boldsymbol{\Sigma}}_t = G_t \cdot \boldsymbol{\Sigma}_{t-1} \cdot G_t^T + F_x^T \cdot R_t \cdot F_x$$

Sensorbeobachtungen



EKF SLAM

Landmarke j wird zum ersten Mal gesehen:

$$\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \\ \bar{\mu}_{j,s} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \\ s_t^i \end{pmatrix} + \begin{pmatrix} r_t^i \cdot \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ r_t^i \cdot \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \\ 0 \end{pmatrix}$$

$$\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix} \quad q = \delta^T \cdot \delta$$

EKF SLAM

$$z_t^i = \begin{pmatrix} r_t^i \\ \phi_t^i \\ s_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta \\ m_{j,s} \end{pmatrix} + N(0, Q_t)$$

$$m = \{m_1, \dots, m_N\} \quad m_j = (m_{j,x}, m_{j,y}, m_{j,s})^T \quad j = c_t^i$$

$$z_t^i = h(\mathbf{y}_t, j) + N(0, Q_t)$$

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

$$h(\mathbf{y}_t, j) \approx h(\bar{\mu}_t, j) + H_t^i (\mathbf{y}_t - \bar{\mu}_t)$$

EKF SLAM

$$h(\mathbf{y}_t, j) \approx h(\bar{\boldsymbol{\mu}}_t, j) + H_t^i (\mathbf{y}_t - \bar{\boldsymbol{\mu}}_t)$$

$$H_t^i = h_t^i \cdot F_{x,j}$$

$$H_t^i = \frac{1}{q} \begin{pmatrix} \sqrt{q} \cdot \delta_x & -\sqrt{q} \cdot \delta_y & 0 & -\sqrt{q} \cdot \delta_x & \sqrt{q} \cdot \delta_y & 0 \\ \delta_y & \delta_x & -1 & -\delta_y & -\delta_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot F_{x,j}$$

EKF SLAM

$$H_t^i = h_t^i \cdot F_{x,j}$$

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{pmatrix}$$

$\underbrace{\hspace{10em}}_{3 \cdot j - 3} \qquad \underbrace{\hspace{10em}}_{3 \cdot N - 3 \cdot j}$

EKF SLAM

for all features z_t^i :

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

$$K_t^i = \bar{\Sigma}_t \cdot (H_t^i)^T \cdot (S_t^i)^{-1}$$

$$\bar{\mu}_t = \bar{\mu}_t + K_t^i \left(\begin{array}{c} \sqrt{q} \\ \text{atan } 2(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \\ \bar{\mu}_{j,s} \end{array} \right)$$

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

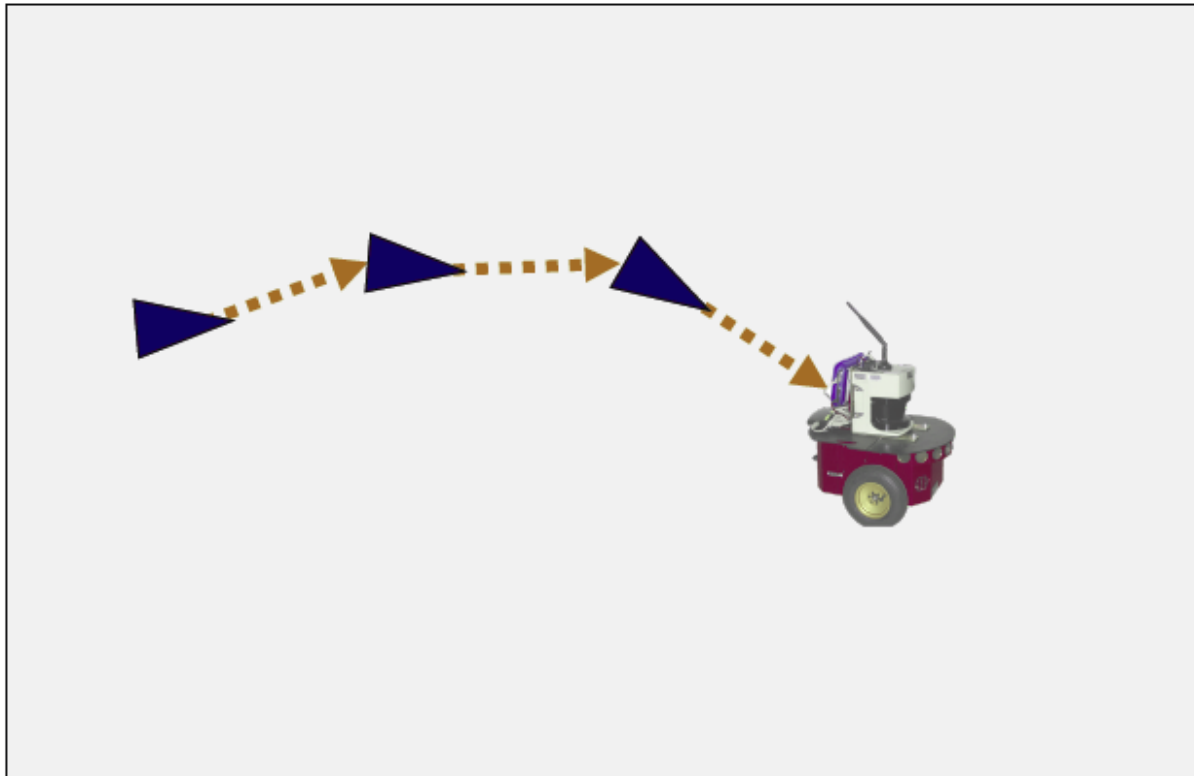
$$\mu_t = \bar{\mu}_t \quad \Sigma_t = \bar{\Sigma}_t$$

9.3 Graph SLAM

Idee

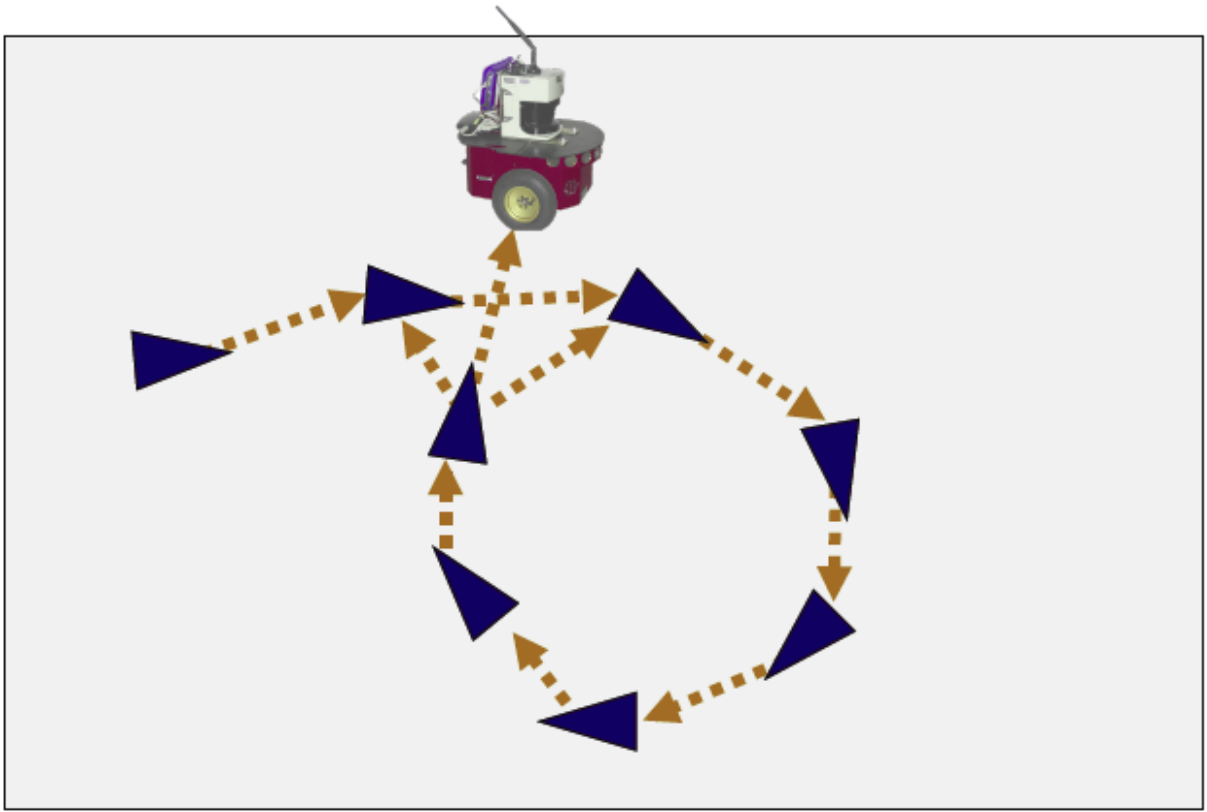
- Graph zur Repräsentation des Problems
- Knoten – Roboterposition
- Kante – Constraint zwischen 2 Knoten (Beobachtung)
- Konstruiere den Graphen und finde eine Knotenmenge, die einen Fehler über alle Constraints minimiert

Graph – SLAM



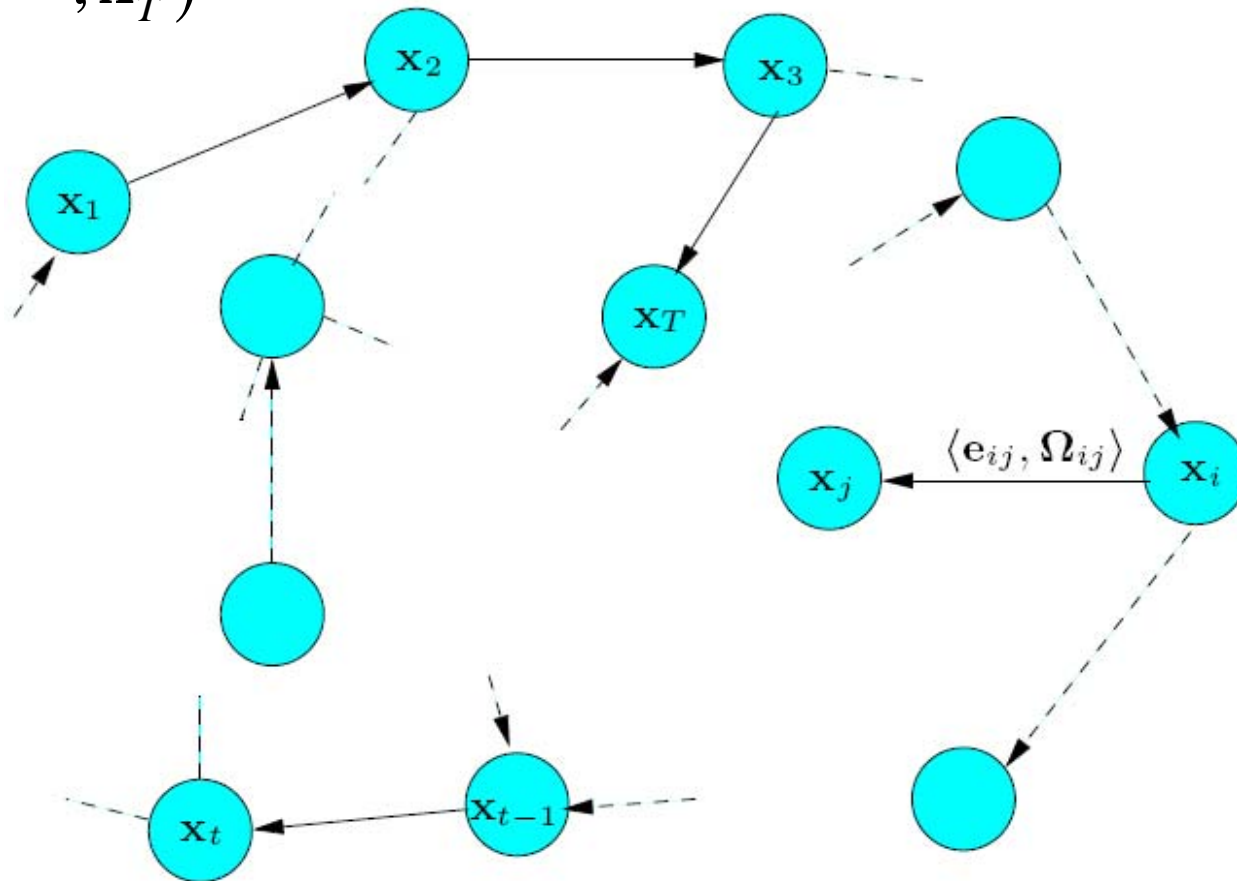
 Robot pose

 Constraint

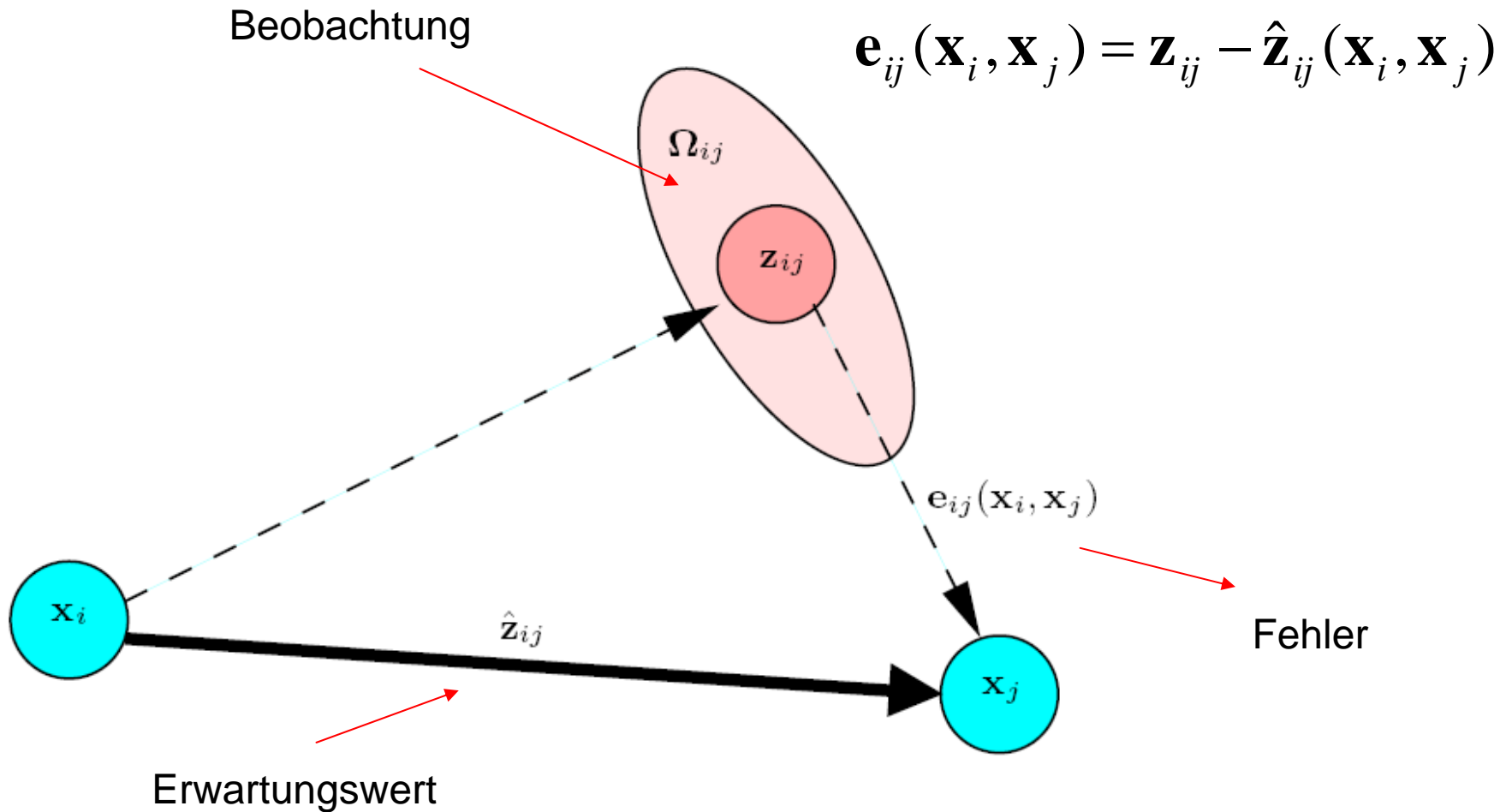


Graph

$$\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_T)^T$$



Graph



Ziel

$$\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_T)^T$$

$$\mathbf{x}^* = \operatorname{argmin} \mathbf{F}(\mathbf{x})$$

$$\mathbf{F}(\mathbf{x}) = \sum_{(i,j)} \mathbf{e}_{ij}^T \cdot \Omega_{ij} \cdot \mathbf{e}_{ij}$$

Lösung: Gauss – Newton oder Levenberg – Marquardt

Lösung – Gauss Newton

$$\begin{aligned} \mathbf{e}_{ij}(\check{\mathbf{x}}_i + \Delta \mathbf{x}_i, \check{\mathbf{x}}_j + \Delta \mathbf{x}_j) &= \mathbf{e}_{ij}(\check{\mathbf{x}} + \Delta \mathbf{x}) \\ &\simeq \mathbf{e}_{ij} + \mathbf{J}_{ij} \Delta \mathbf{x}. \end{aligned}$$

Initiallösung

Jacobi – Matrix

$$\begin{aligned} \mathbf{F}_{ij}(\check{\mathbf{x}} + \Delta \mathbf{x}) &= \mathbf{e}_{ij}(\check{\mathbf{x}} + \Delta \mathbf{x})^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}(\check{\mathbf{x}} + \Delta \mathbf{x}) \\ &\simeq (\mathbf{e}_{ij} + \mathbf{J}_{ij} \Delta \mathbf{x})^T \boldsymbol{\Omega}_{ij} (\mathbf{e}_{ij} + \mathbf{J}_{ij} \Delta \mathbf{x}) \\ &= \underbrace{\mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}}_{c_{ij}} + 2 \underbrace{\mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{b}_{ij}} \Delta \mathbf{x} + \Delta \mathbf{x}^T \underbrace{\mathbf{J}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{H}_{ij}} \Delta \mathbf{x} \\ &= c_{ij} + 2\mathbf{b}_{ij} \Delta \mathbf{x} + \Delta \mathbf{x}^T \mathbf{H}_{ij} \Delta \mathbf{x} \end{aligned}$$

Lösung – Gauss Newton

$$\begin{aligned} F(\check{\mathbf{x}} + \Delta \mathbf{x}) &= \sum_{\langle i,j \rangle \in \mathcal{C}} F_{ij}(\check{\mathbf{x}} + \Delta \mathbf{x}) \\ &\simeq \sum_{\langle i,j \rangle \in \mathcal{C}} c_{ij} + 2\mathbf{b}_{ij} \Delta \mathbf{x} + \Delta \mathbf{x}^T \mathbf{H}_{ij} \Delta \mathbf{x} \\ &= \mathbf{c} + 2\mathbf{b}^T \Delta \mathbf{x} + \Delta \mathbf{x}^T \mathbf{H} \Delta \mathbf{x}. \quad \longrightarrow \quad \text{min} \end{aligned}$$



$$\mathbf{H} \Delta \mathbf{x}^* = -\mathbf{b}$$

neue Lösung



$$\mathbf{x}^* = \check{\mathbf{x}} + \Delta \mathbf{x}^*$$

Beispiel – 2D

$$\mathbf{x}_i^T = (\mathbf{t}_i^T, \theta_i)$$

$$\mathbf{z}_{ij}^T = (\mathbf{t}_{ij}^T, \theta_{ij})$$

2D – Vektor

Rotationswinkel

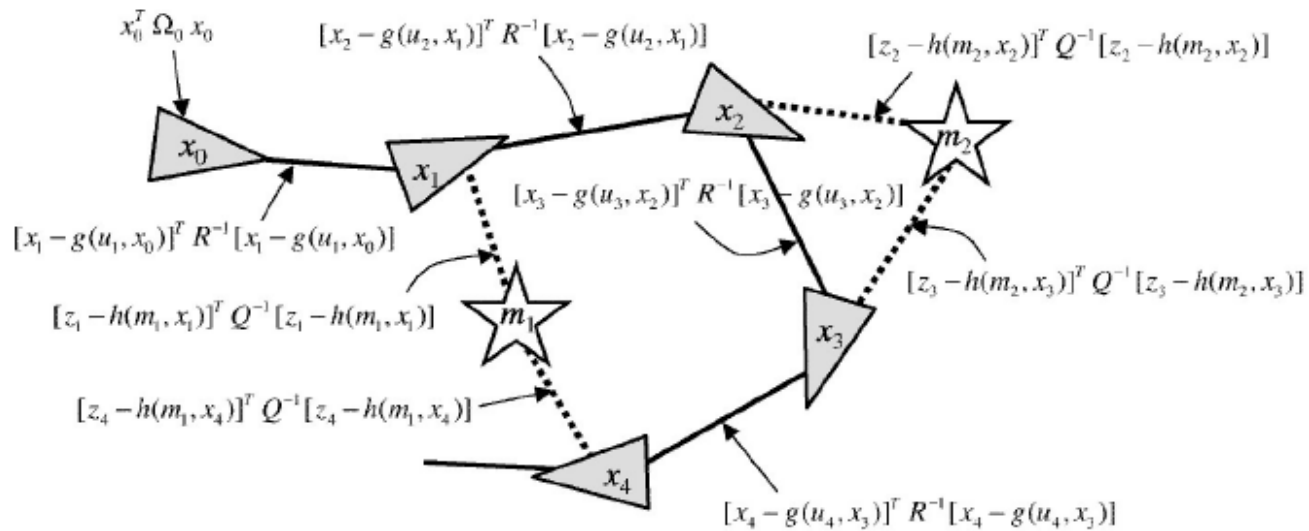
$$\mathbf{e}_{ij}(\mathbf{x}) = \begin{pmatrix} \mathbf{R}_{ij}^T (\mathbf{R}_i^T (\mathbf{t}_j - \mathbf{t}_i) - \mathbf{t}_{ij}) \\ \theta_j - \theta_i - \theta_{ij} \end{pmatrix}$$

$$\mathbf{R}_i = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{pmatrix}$$

Graph – SLAM mit Landmarken

- Full SLAM
- Knoten sind Roboterpositionen und Landmarken
- Kanten beschreiben Bewegungen und Sensorbeobachtungen
- Kanten sind nichtlineare Constraints zugeordnet

Graph – SLAM mit Landmarken



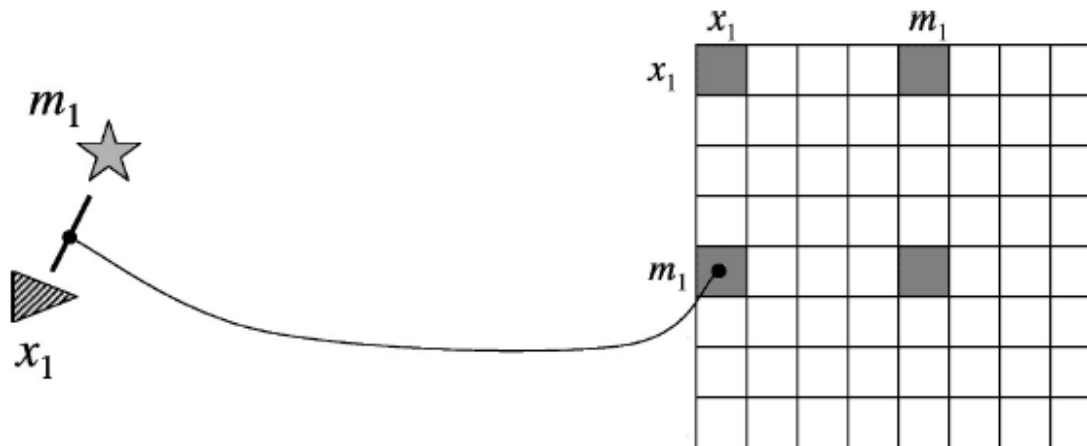
Sum of all constraints:

$$J_{\text{GraphSLAM}} = x_0^T \Omega_0 x_0 + \sum_t [x_t - g(u_t, x_{t-1})]^T R^{-1} [x_t - g(u_t, x_{t-1})] + \sum_t [z_t - h(m_c, x_t)]^T Q^{-1} [z_t - h(m_c, x_t)]$$

wird minimiert

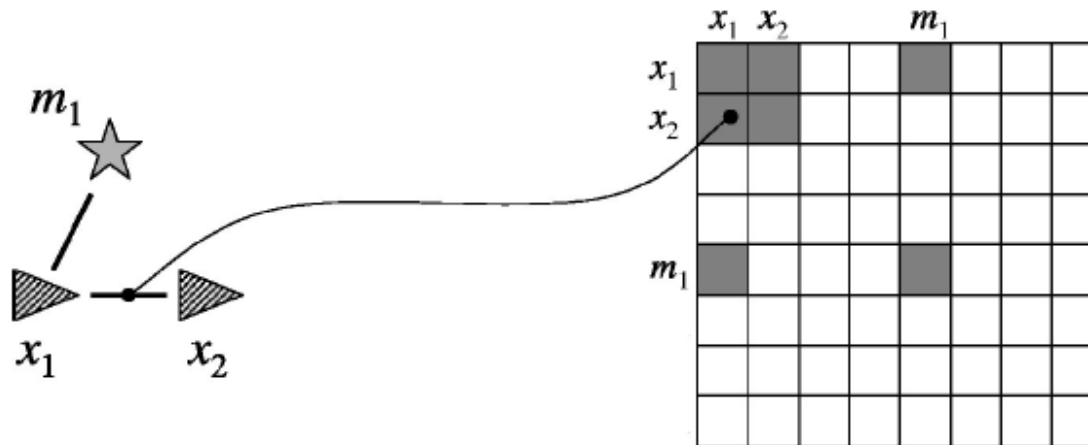
Konstruktion

(a) Observation is landmark m_1



Konstruktion

(b) Robot motion from x_1 to x_2



Konstruktion

(c) Several steps later

