9 SLAM – Simultaneous Localization and Mapping

Einleitung

- eines der aktivsten Forschungsgebiete innerhalb der Robotik
- Roboterlokalisierung bei gegebener Karte (Kap. 8) und Karte aus Sensordaten bei bekannter Position (Kap. 7) sind relativ einfach
- beide Probleme aber gleichzeitig zu lösen ist schwierig
- aktuelle Algorithmen verwenden probabilistische Ansätze, um Unsicherheiten explizit zu modellieren
- deterministische Ansätze sind meist schlechter

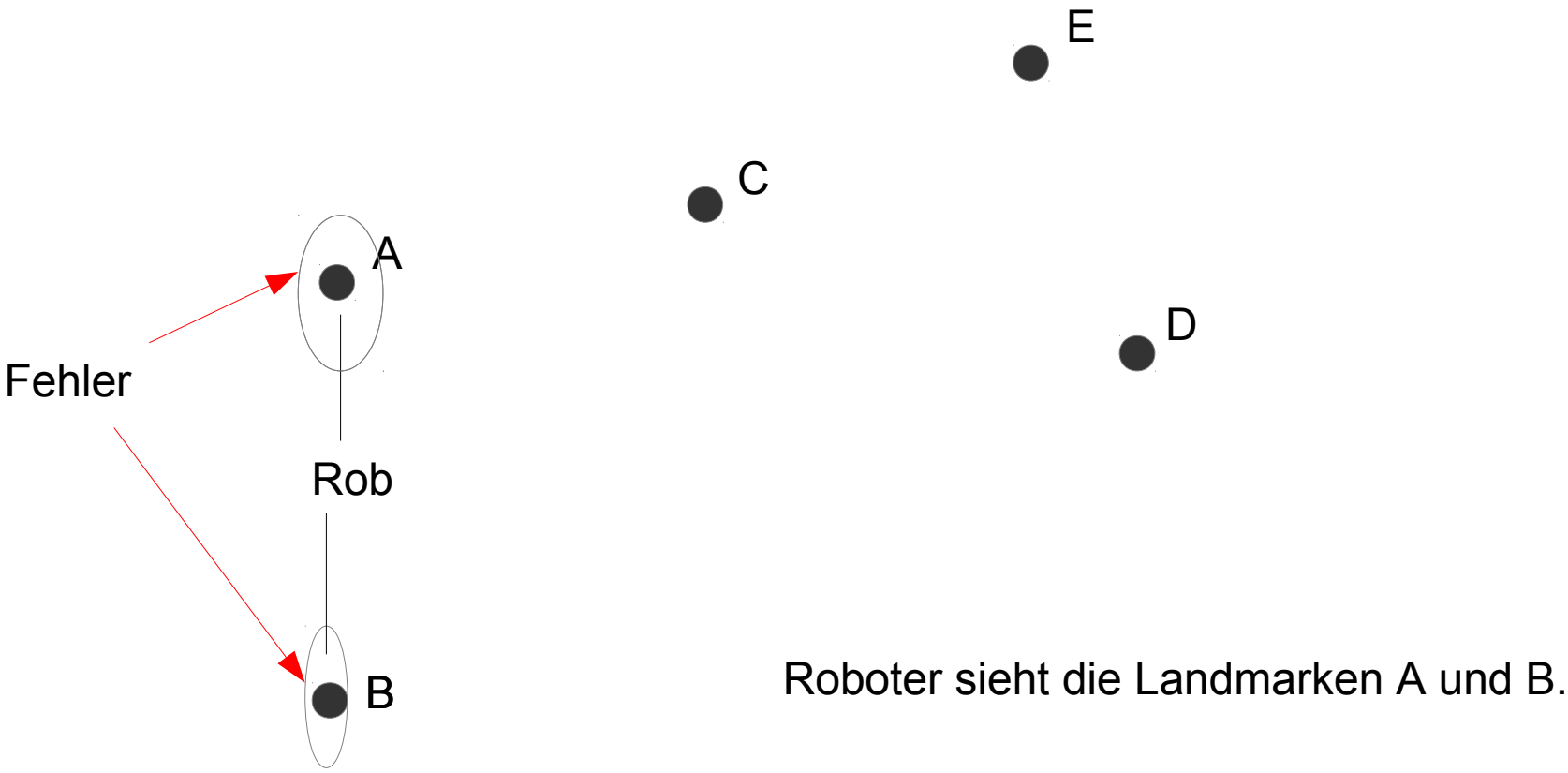
Einleitung

- Eingabe:
  - Sensordaten
  - Kontrollkommandos des Roboters (Fahren)
- Gesucht:
  - Schätzung der Karte
  - Pfad des Roboters (alle bisherigen Positionen)

Probleme

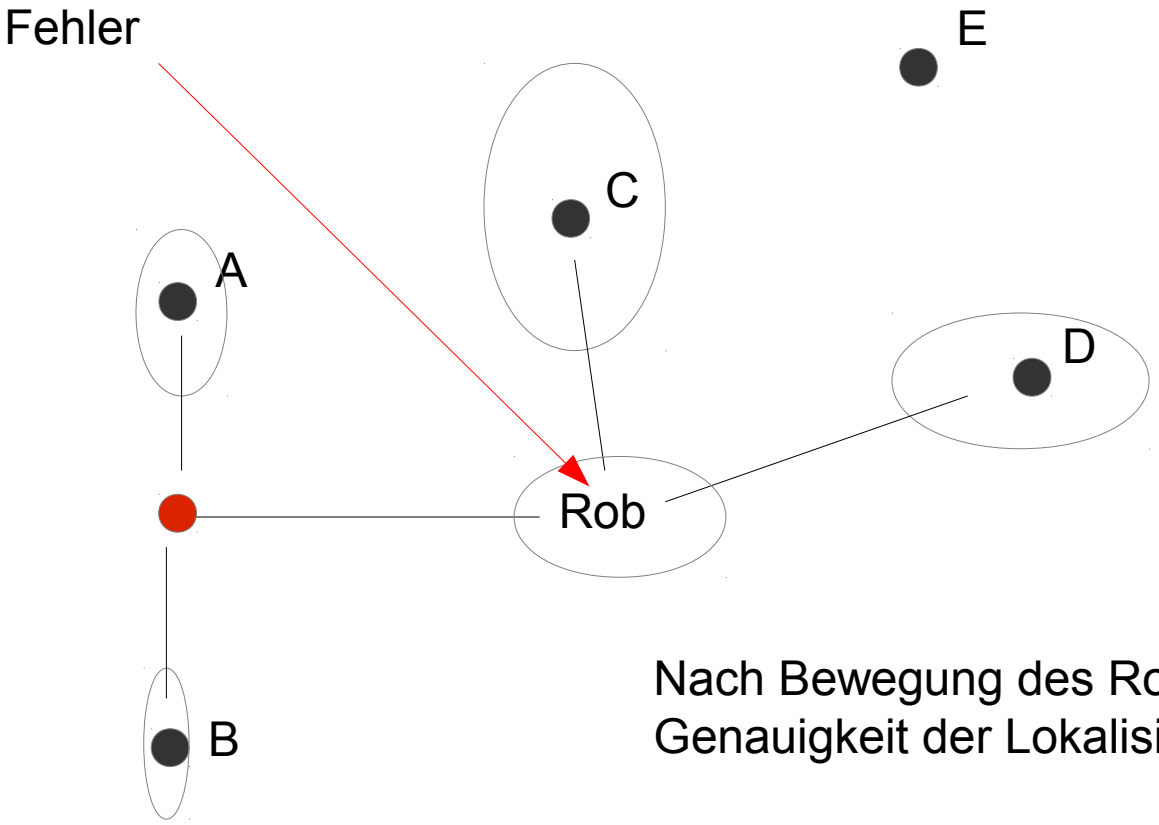
- Sowohl Positionen der Landmarken als auch Roboterweg sind unbekannt
- Die Zuordnung von Sensordaten zu Landmarken sind in der Regel unbekannt
- Roboter muss entscheiden, ob Sensordaten zu einer bereits beobachteten Landmarke zugeordnet werden können oder zu einer noch nicht gesehenen Landmarke
- Zuordnungsproblematik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

SLAM – Graph – 1



Roboter sieht die Landmarken A und B.

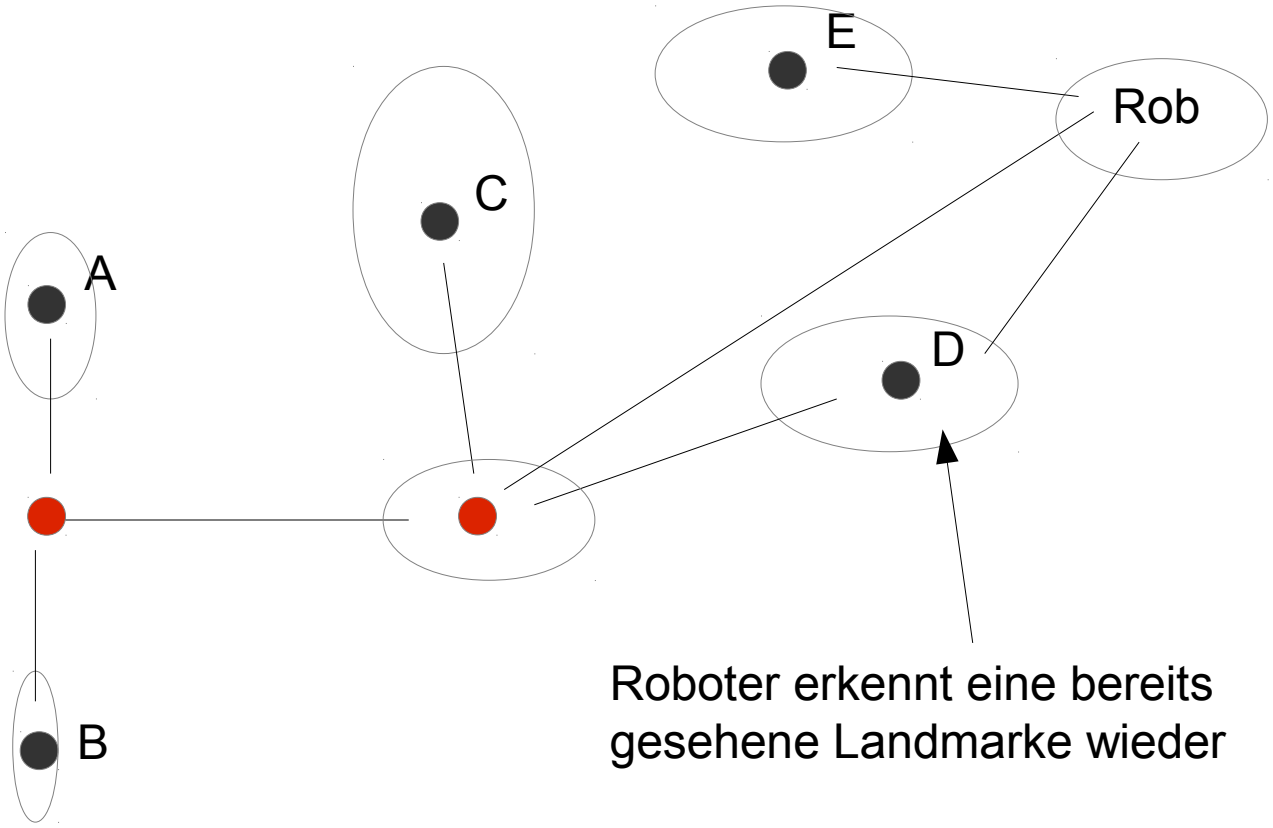
SLAM – Graph – 2



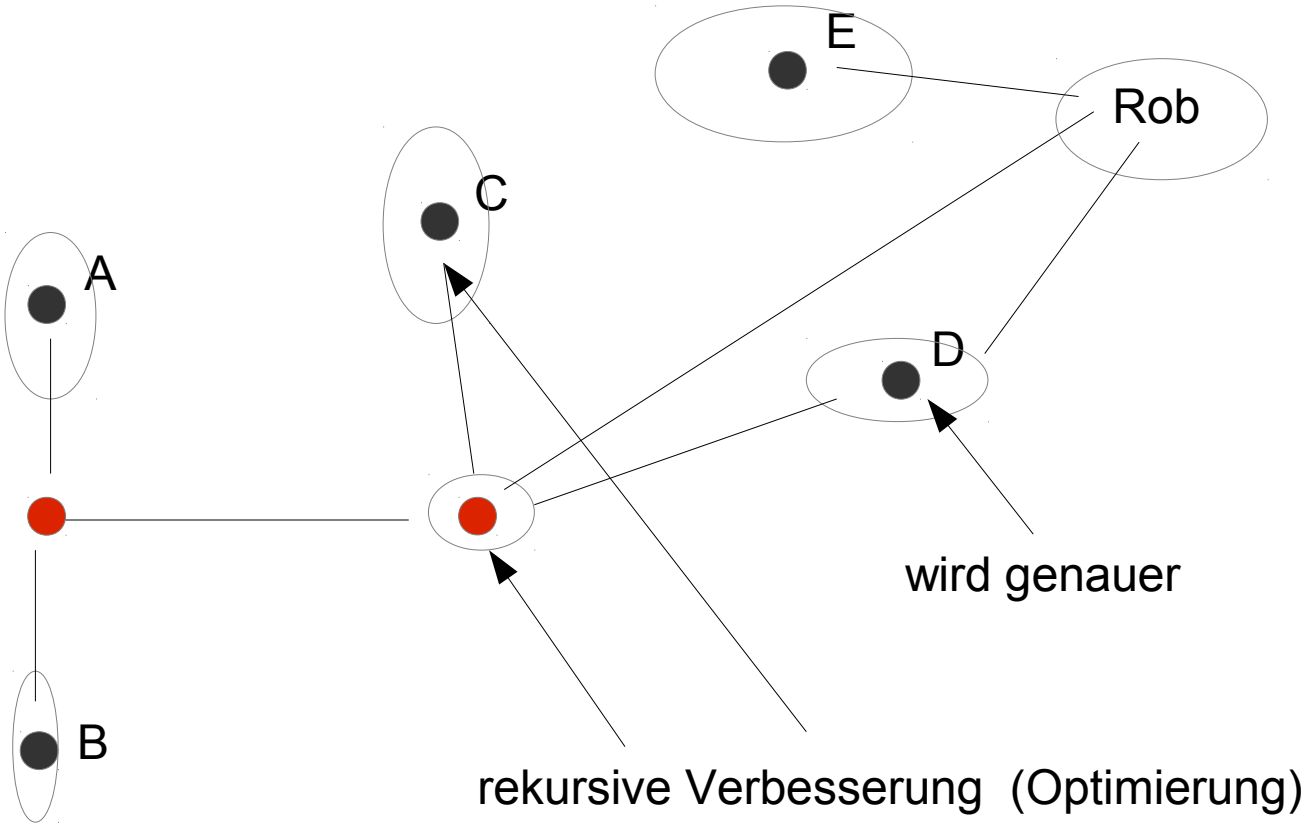
Nach Bewegung des Roboters nimmt die Genauigkeit der Lokalisierung ab

Die Unsicherheit der Positionsschätzungen der Landmarken C und D steigt

SLAM – Graph – 3


E
Rob
C
A
D
B
Roboter erkennt eine bereits
gesehene Landmarke wieder

SLAM – Graph – 4

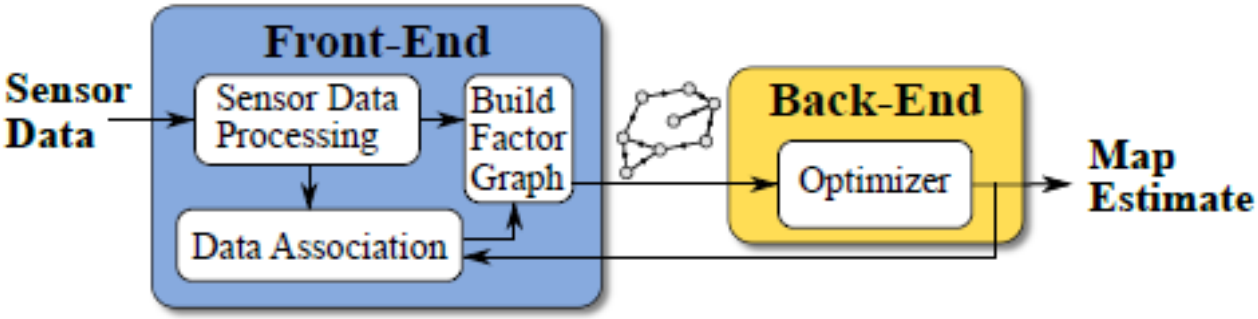

E
Rob
C
A
D
wird genauer
B
rekursive Verbesserung (Optimierung)

Problem

- Datenassoziation
- Sobald Landmarken nicht eindeutig unterscheidbar sind, muss die Frage beantwortet werden, in welchen Fällen zwei gleich aussehende Landmarken tatsächlich dieselbe Landmarke waren
- Macht der Roboter hier Fehler, kann das starke Auswirkungen auf die Lösung (Karte) haben

Lösung des SLAM – Problems

- Aufbau des Graphen
- Optimierung



SLAM – Simultaneous Localization and Mapping

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

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

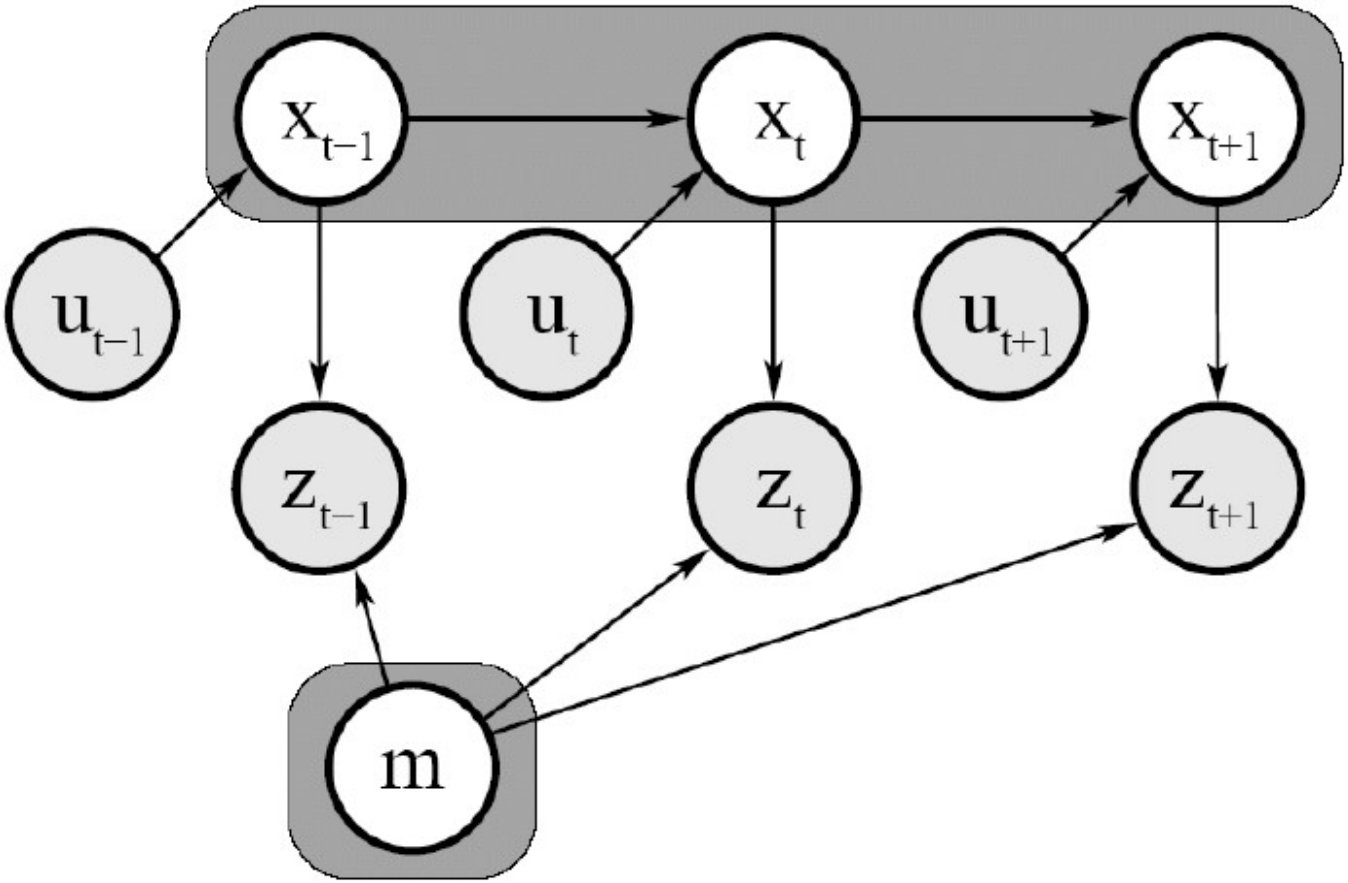
Online SLAM Problem
(inkrementelles SLAM)

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

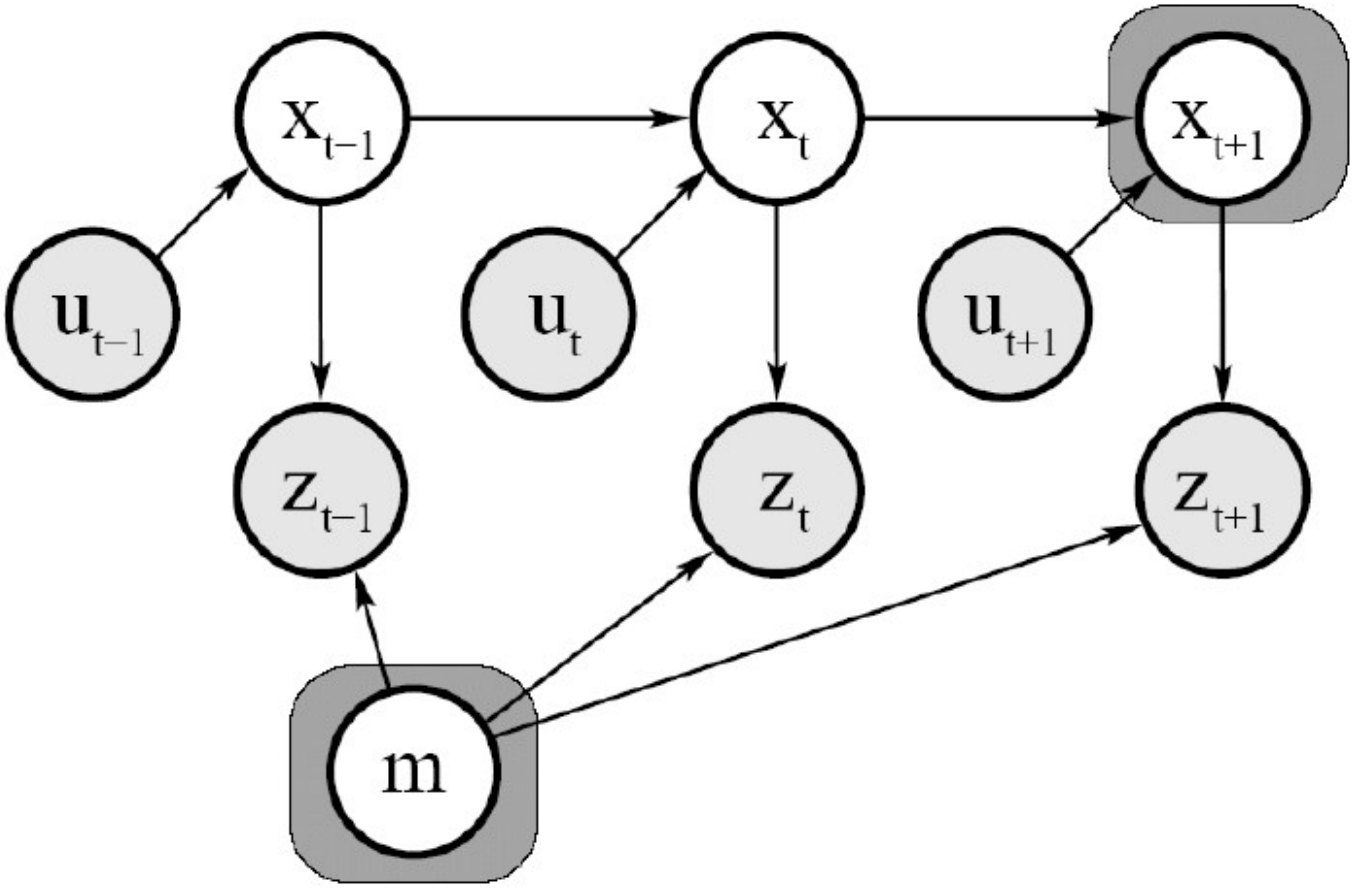
$$p(x_{1:t}, m, c_{1:t} | z_{1:t}, u_{1:t})$$

Full SLAM Problem
(vollständiges SLAM)

Full SLAM



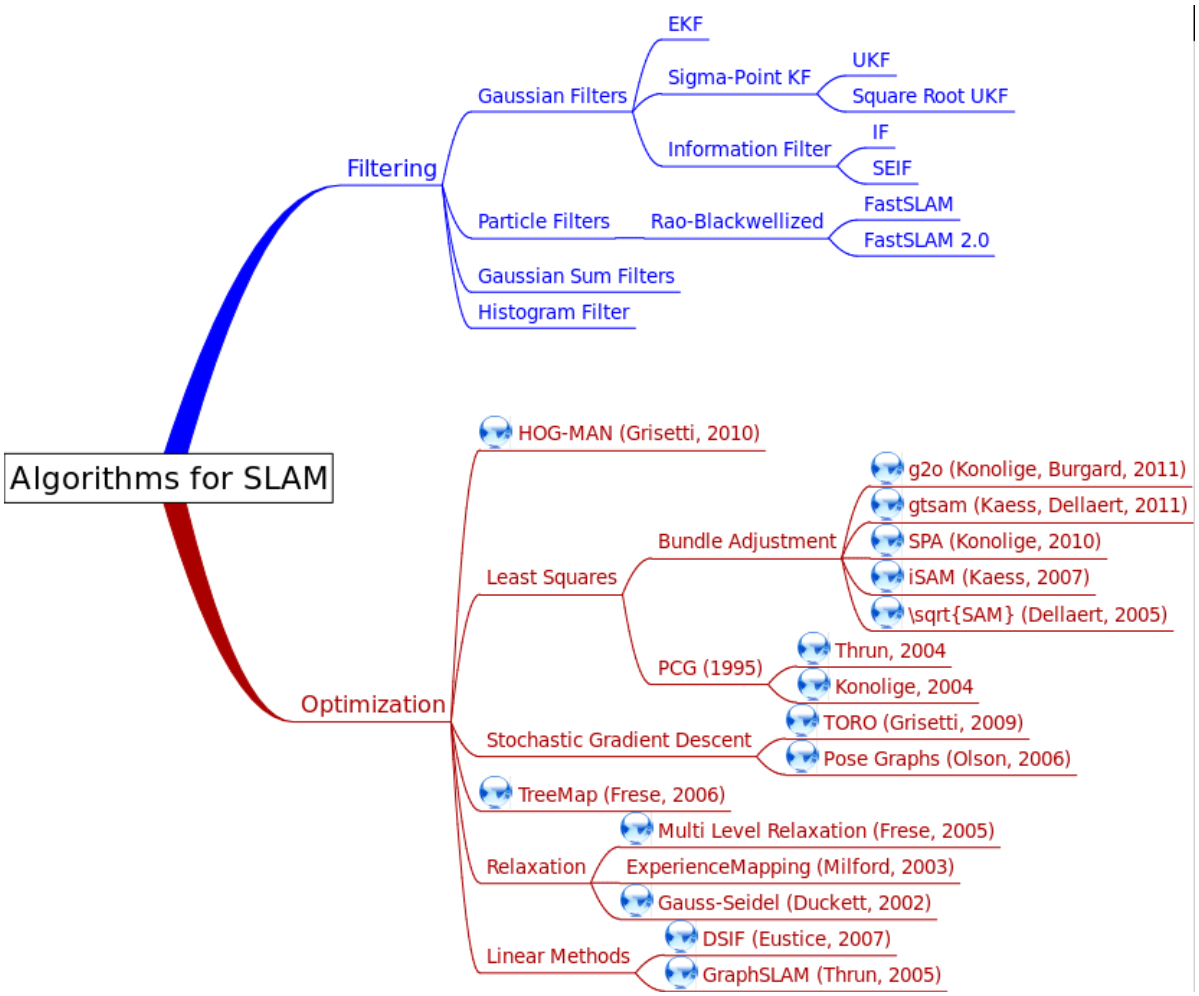
Online SLAM



Verfahren

- EKF SLAM
  - verwendet Extended Kalman Filter
  - Online
- Fast SLAM
  - verwendet Partikel Filter
- Graph SLAM
  - vollständiges SLAM

Verfahren – Niko Sünderhauf



9.1 Simple SLAM – Einfaches Modell

Simple SLAM

Roboterposition (x_r, y_r)

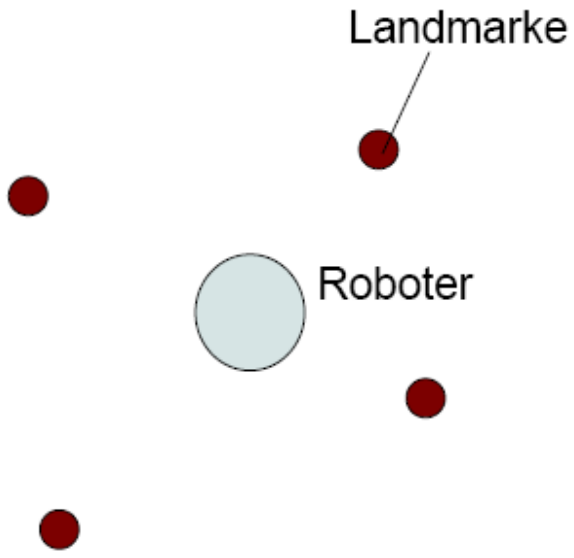
N Landmarken (Positionen):

(x_{l_1}, y_{l_1})

(x_{l_2}, y_{l_2})

...

(x_{l_N}, y_{l_N})



Simple SLAM - Zustand

$$x_t = \left(x_r(t), y_r(t), x_{l_1}, y_{l_1}, \cdots, x_{l_i}, y_{l_i}, \cdots x_{l_N}, y_{l_N} \right)^T$$

Bewegung

Omnidirektionales Bewegungsmodell
(Antrieb in alle Richtungen)
(Roboterfront zeigt immer in gleiche Richtung)

$$u_t = \left(v_x(t), v_y(t) \right)^T$$

Simple SLAM – Bewegungsgleichung

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

$$A_t = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

$$B_t = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix}$$

Landmarken bewegen sich nicht

$$\varepsilon_t = \left(\varepsilon_{rx}(t), \varepsilon_{ry}(t), 0, 0, \cdots, 0\right)^T$$

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

$$\varepsilon_t \rightarrow N\left(0, V_t\right)$$

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

Simple SLAM – Sensordaten

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

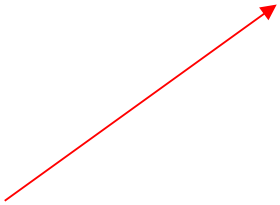
$$z_t = \begin{pmatrix} z_1(t) \\ \vdots \\ z_i(t) \\ \vdots \\ z_N(t) \end{pmatrix} = \begin{pmatrix} x_{l_1} - x_r(t) \\ y_{l_1} - y_r(t) \\ \vdots \\ x_{l_N} - x_r(t) \\ y_{l_N} - y_r(t) \end{pmatrix}$$

Simple SLAM - Messgleichung

$$z_t = H_t \cdot x_t + \delta_t$$

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

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



(2i+1)-te Spalte

Simple SLAM – Messgleichung

$$z_t = H_t \cdot x_t + \delta_t$$

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

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

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

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

Simple SLAM – Kalmanfilter

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

Simple SLAM – Erweiterungen

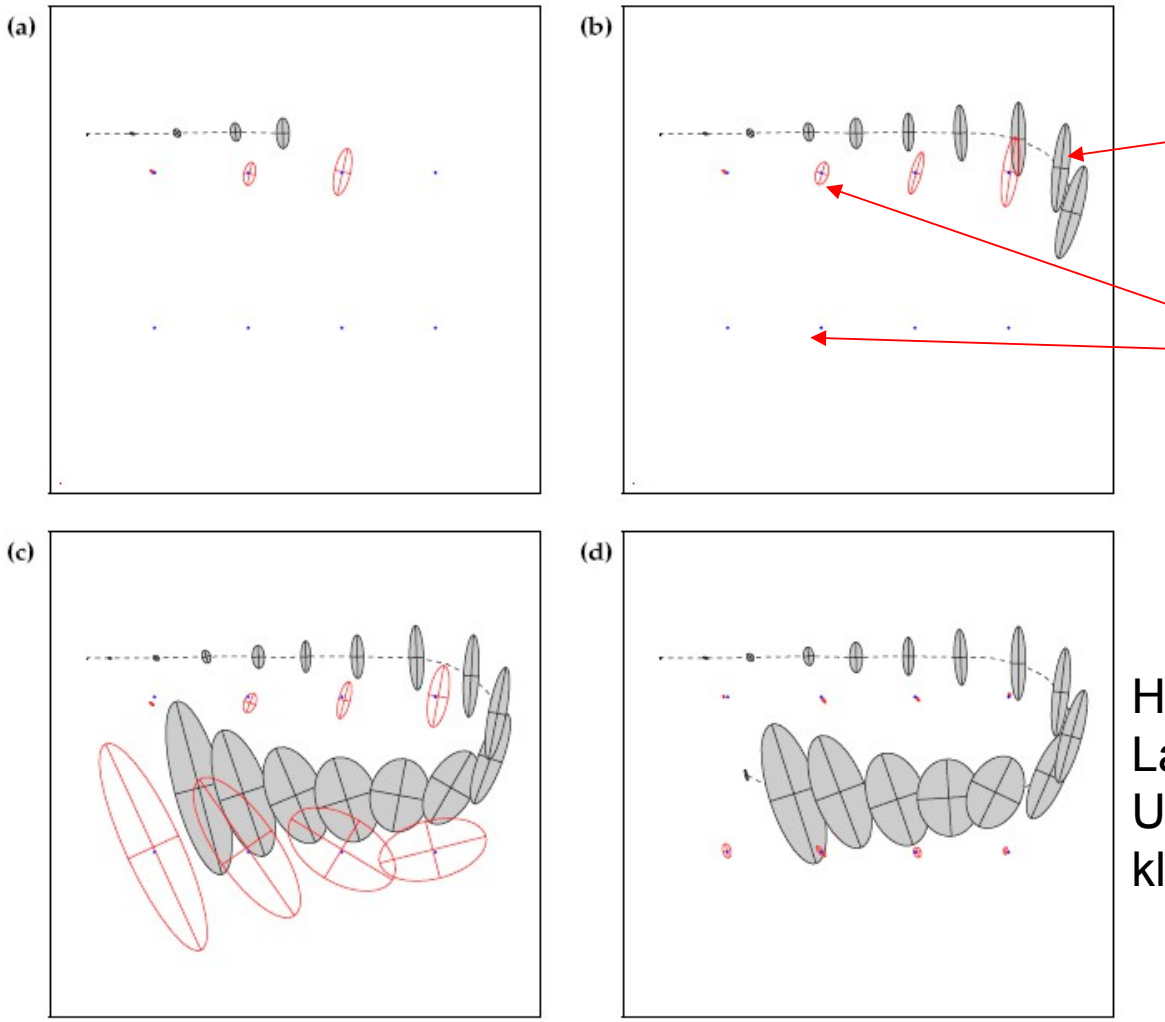
- Das Bewegungsmodel kann erweitert werden. Zur Roboterposition kommt dann noch die Orientierung dazu.
- Statt bei jeder Landmarke die relative Position zu messen, kann auch Abstand und Orientierung gemessen werden.
- Zusätzlich kann noch das Problem der Zuordnung der Messdaten zu den Landmarken-Nummern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



Hier sieht der Roboter die erste Landmarke erneut, sämtliche Unsicherheiten werden erheblich kleiner.

EKF SLAM

zu berechnen:

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

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

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

EKF SLAM

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

$$\Sigma_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \infty & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \infty \end{pmatrix} \quad (3N+3) \times (3N+3) \text{ - Matrix}$$

EKF SLAM

$$y_t = y_{t-1} + F_x^T \cdot \begin{pmatrix} -\frac{v_t}{w_t}\cdot\sin\theta + \frac{v_t}{w_t}\cdot\sin(\theta + w_t\cdot\Delta t) \\ \frac{v_t}{w_t}\cdot\cos\theta - \frac{v_t}{w_t}\cdot\cos(\theta + w_t\cdot\Delta t) \\ w_t\cdot\Delta t \end{pmatrix} + N\left(0, F_x^T\cdot R_t\cdot F_x\right)$$

$$y_t = g(u_t, y_{t-1}) + N\left(0, F_x^T\cdot R_t\cdot F_x\right)$$

$$F_x = \begin{pmatrix} 1 & 0 & 0 & \underbrace{0 & \cdots & 0}_{3\cdot N} \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}$$

EKF SLAM

$$y_t = g(u_t, y_{t-1}) + N(0, F_x^T \cdot R_t \cdot F_x)$$

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

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

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

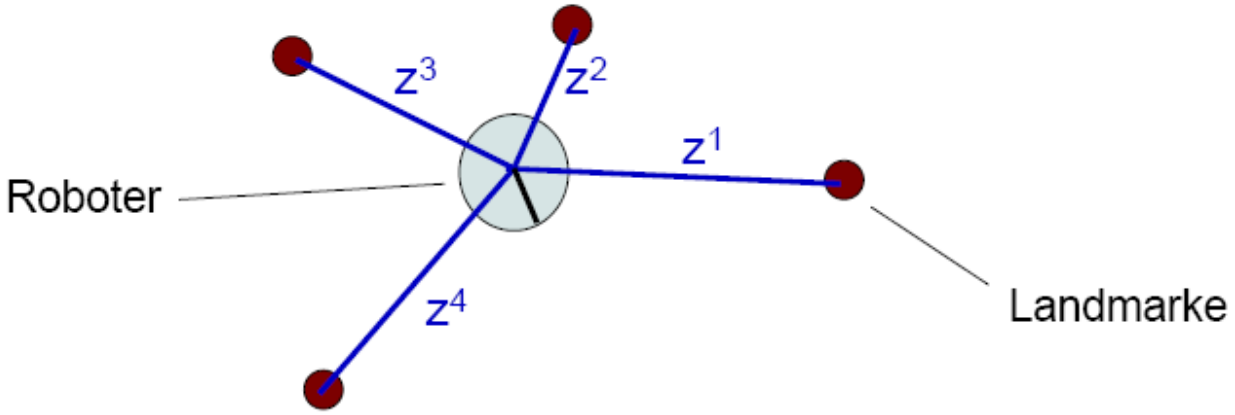
EKF SLAM

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

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

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

Sensorbeobachtungen



EKF SLAM

Landmarke j wird zum ersten Mal gesehen:

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

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

$$q = \delta^T \cdot \delta$$

EKF SLAM

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

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

$$z_t^i = h(y_t, j) + N(0, Q_t)$$

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

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

EKF SLAM

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

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

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

EKF SLAM

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

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & \underbrace{0 & \cdots & 0}_{3\cdot j-3} & 0 & 0 & 0 & \underbrace{0 & \cdots & 0}_{3\cdot N-3\cdot j} \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}$$

EKF SLAM

for all features z_t^i:

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

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

$$\bar{\mu}_t = \bar{\mu}_t + K_t^i \left(z_t^i - \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{pmatrix}\right)$$

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

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

9.3 Graph SLAM

9.3.1 Pose Graph SLAM Problem

Idee

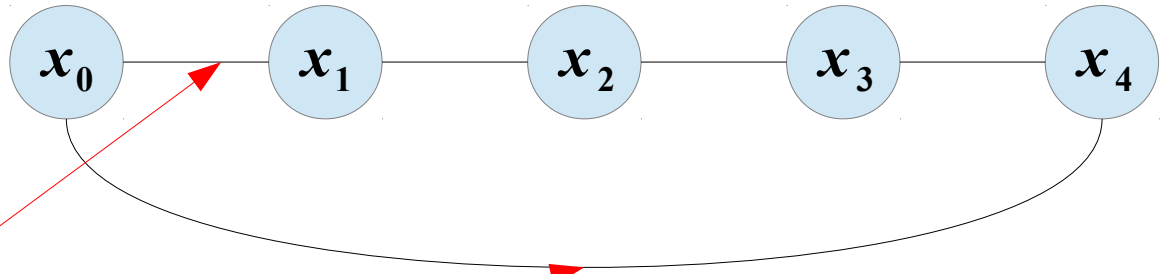
- Karte – alle Roboterpositionen, keine Landmarken
- Graph (Pose Graph) zur Repräsentation des Problems
- Knoten – Roboterpositionen
- Kante – Constraint zwischen 2 Knoten (Odometrie Messung)
- Konstruiere den Graphen und finde eine Knotenmenge, die einen Fehler über alle Constraints minimiert

Karte und Pose Graph

2D Welt: $\boldsymbol{x} = \left(x, y, \theta \right)^T$

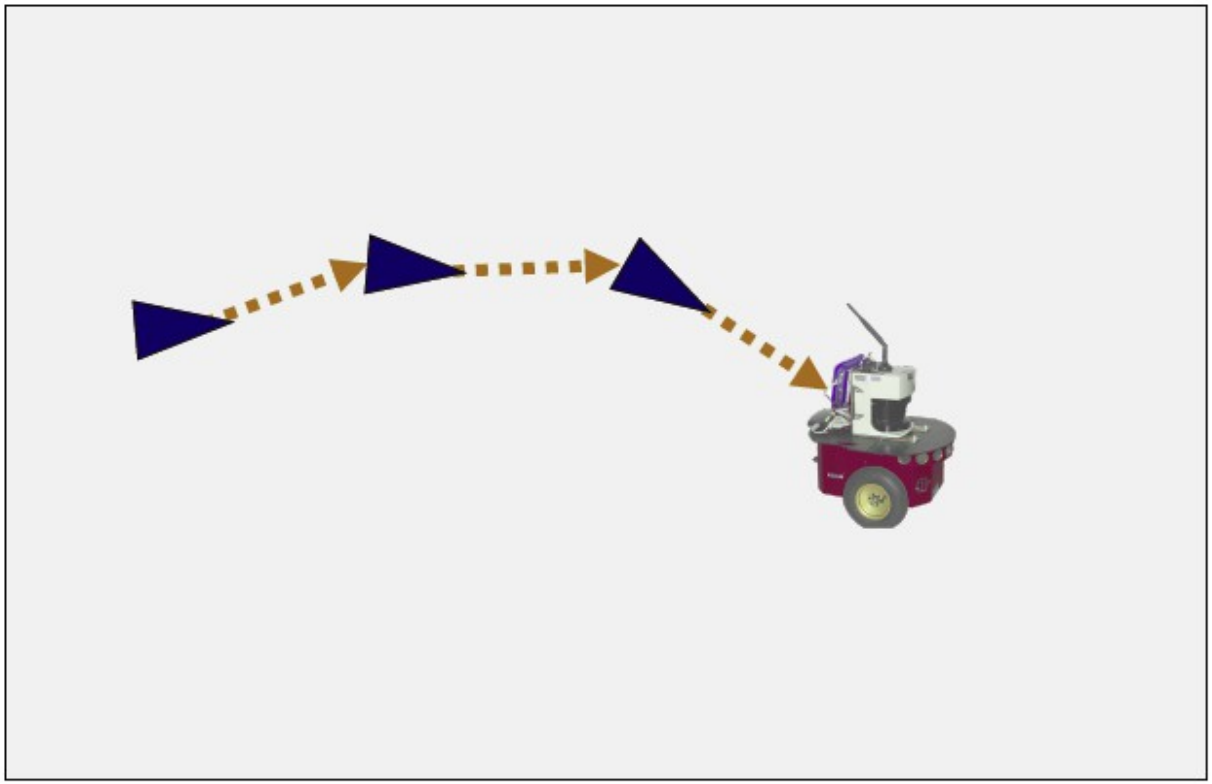
Roboterpositionen: $\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots$

Pose Graph:



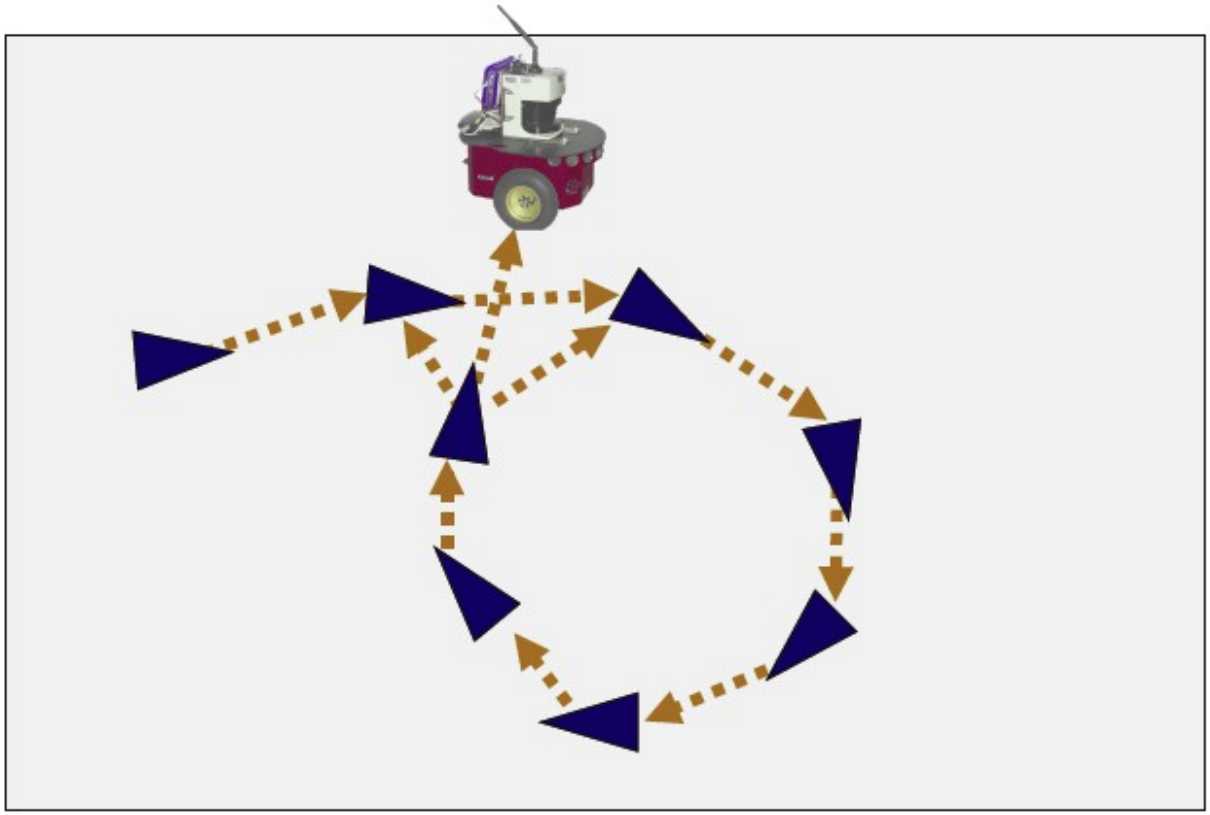
Odometrie – Constraints

Weg des Roboters

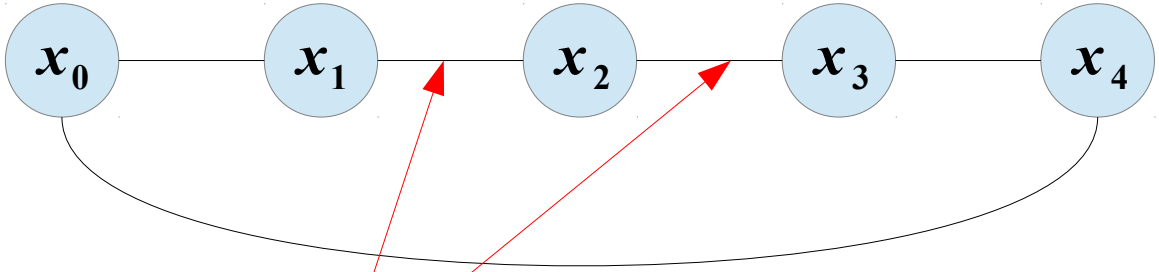


Robot pose Constraint

Roboter sieht Positionen erneut



Kanten – Constraints



Bewegungsmodell

Odometrie: $$x_{i+1} = g(x_i, u_i) + w_i$$

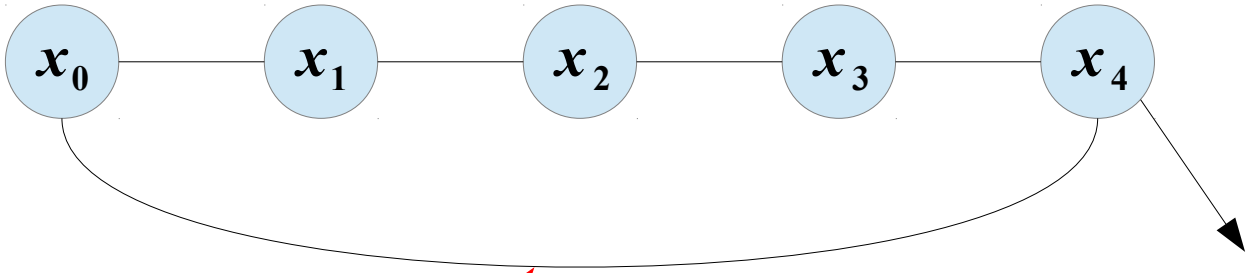
Normalverteilung

$$w_i \sim N(0, \Sigma_i)$$

$$x_{i+1} \sim N(g(x_i, u_i), \Sigma_i)$$

Kanten – Constraints

Loop closure constraints:



Der Roboter sieht eine Position erneut.

$$x_j \sim N\left(g\left(x_i, u_{i,j}\right), \Lambda_{ij}\right)$$

Graph SLAM – Optimierung

Gegeben: $\boldsymbol{u}_i, \boldsymbol{u}_{ij} \in U$

Gesucht: X^* $\boldsymbol{x}_i \in X$

$$X^* = \underset{X}{argmax}\, P(X|U)$$

Graph SLAM – Optimierung

$$X^* = \underset{X}{argmax}\, P(X|U)$$

$$P(X|U) = \alpha \prod_i P(\boldsymbol{x}_{i+1}|\boldsymbol{x}_i, \boldsymbol{u}_i) \prod_{i,j} P(\boldsymbol{x}_j|\boldsymbol{x}_i, \boldsymbol{u}_{i,j})$$

Odometry Constraints

Loop Closure Constraints

Graph SLAM – Optimierung

$$P(X|U) = \alpha \prod_i P(\boldsymbol{x}_{i+1}|\boldsymbol{x}_i, \boldsymbol{u}_i) \prod_{i,j} P(\boldsymbol{x}_j|\boldsymbol{x}_i, \boldsymbol{u}_{i,j})$$

Squared Mahalanobis distance: $\|\boldsymbol{a}-\boldsymbol{b}\|_\Sigma^2 = (\boldsymbol{a}-\boldsymbol{b})^T \Sigma^{-1} (\boldsymbol{a}-\boldsymbol{b})$

$$P(\boldsymbol{x}_{i+1}|\boldsymbol{x}_i, \boldsymbol{u}_i) = \eta e^{-\frac{1}{2}\|g(\boldsymbol{x}_i, \boldsymbol{u}_i)-\boldsymbol{x}_{i+1}\|_{\Sigma_i}^2}$$

$$P(\boldsymbol{x}_j|\boldsymbol{x}_i, \boldsymbol{u}_{ij}) = \eta e^{-\frac{1}{2}\|g(\boldsymbol{x}_i, \boldsymbol{u}_{ij})-\boldsymbol{x}_j\|_{\Lambda_{i,j}}^2}$$

Graph SLAM – Optimierung

$$P(X|U) = \eta \prod_i e^{-\frac{1}{2}\|g(\boldsymbol{x_i}, \boldsymbol{u_i}) - \boldsymbol{x_{i+1}}\|^2_{\Sigma_i}} \prod_{i,j} e^{-\frac{1}{2}\|g(\boldsymbol{x_i}, \boldsymbol{u_{i,j}}) - \boldsymbol{x_j}\|^2_{\Lambda_{ij}}}$$

$$-\log(P(X|U)) = \eta\left(\sum_i \|g(\boldsymbol{x_i}, \boldsymbol{u_i}) - \boldsymbol{x_{i+1}}\|^2_{\Sigma_i} + \sum_{i,j} \|g(\boldsymbol{x_i}, \boldsymbol{u_{i,j}}) - \boldsymbol{x_j}\|^2_{\Lambda_{ij}}\right)$$

$$X^* = \underset{X}{argmax}\, P(X|U)$$

$$X^* = \underset{X}{argmin} -\log P(X|U)$$

Graph SLAM – Optimierung

$$X^* = \underset{X}{argmin} - \log P(X \mid U)$$

$$X^* = \underset{X}{argmin} \left(\sum_i \| g(\boldsymbol{x}_i, \boldsymbol{u}_i) - \boldsymbol{x}_{i+1} \|^2_{\Sigma_i} + \sum_{i,j} \| g(\boldsymbol{x}_i, \boldsymbol{u}_{i,j}) - \boldsymbol{x}_j \|^2_{\Lambda_{ij}} \right)$$

(weighted) least squares optimization problem

Least Squares Optimization Problem

$$f_i : R^m \rightarrow R \qquad \boldsymbol{x} \in R^m \qquad \boldsymbol{x} = \left(x_{1,} \cdots, x_m \right)^T$$

$$F(\boldsymbol{x}) = \frac{1}{2} \sum_{i=1}^{n} f_i(\boldsymbol{x})^2 \qquad F(\boldsymbol{x}) = \frac{1}{2} f(\boldsymbol{x})^T \cdot f(\boldsymbol{x})$$

$$f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_n(\boldsymbol{x}))^T$$

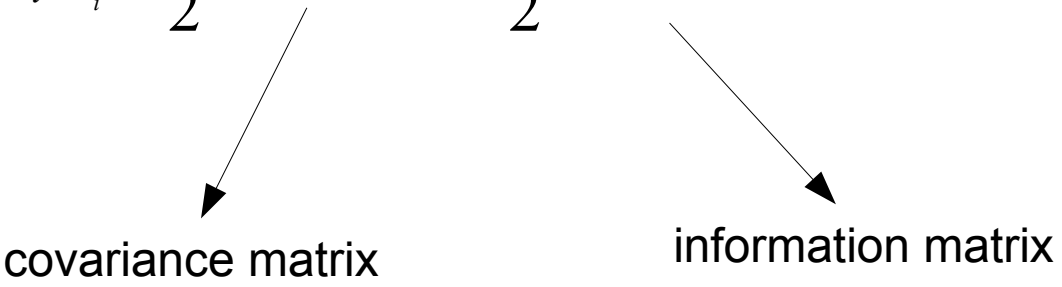
Optimierungsproblem:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{argmin}\, F(\boldsymbol{x}) = \underset{\boldsymbol{x}}{argmin}\, \frac{1}{2} f(\boldsymbol{x})^T \cdot f(\boldsymbol{x})$$

Weighted Least Squares Optimization Problem

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{argmin}\, F(\boldsymbol{x}) = \underset{\boldsymbol{x}}{argmin}\, \frac{1}{2} f(\boldsymbol{x})^T \Omega f(\boldsymbol{x})$$

$$F = \frac{1}{2}\sum_i \|f_i\|^2_{\Sigma_i} = \frac{1}{2} f^T \Sigma^{-1} f = \frac{1}{2} f^T \Omega f$$



Lösung Gauss – Newton

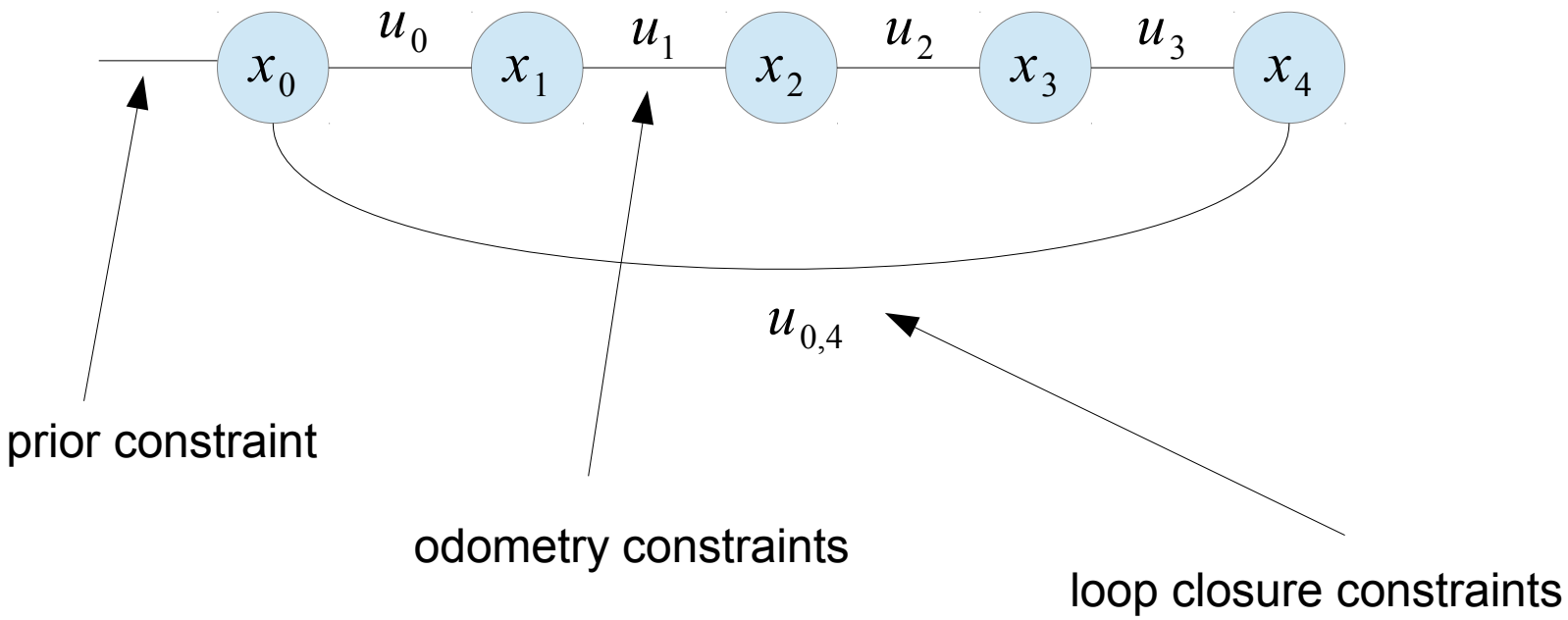
$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{argmin}\, F(\boldsymbol{x}) = \underset{\boldsymbol{x}}{argmin}\, \frac{1}{2} f(\boldsymbol{x})^T \Omega f(\boldsymbol{x})$$

Iteration: $\boldsymbol{x}_0$ $\boldsymbol{x} \rightarrow \boldsymbol{x} + \Delta \boldsymbol{x}$

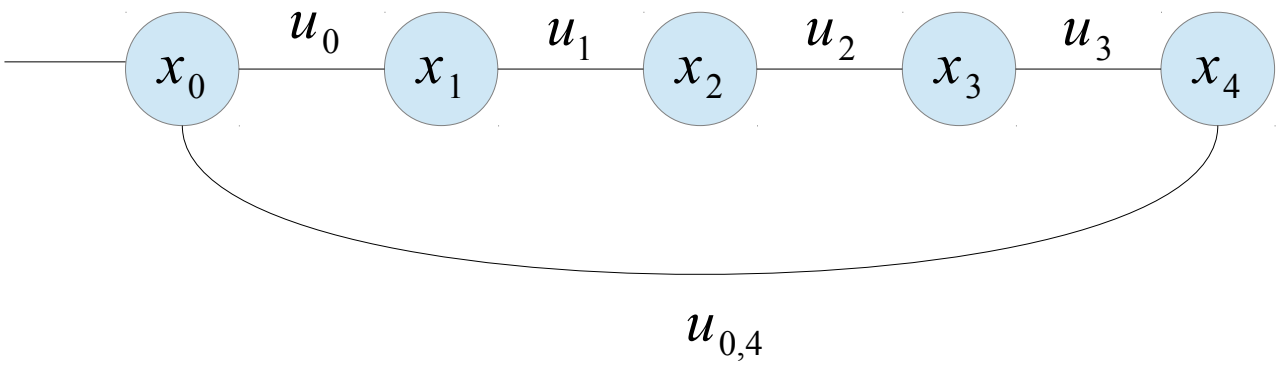
$$J = \frac{\partial f}{\partial \boldsymbol{x}}$$

$$J^T \Omega J \Delta \boldsymbol{x} = -J^T \Omega^T f(\boldsymbol{x})$$

Beispiel

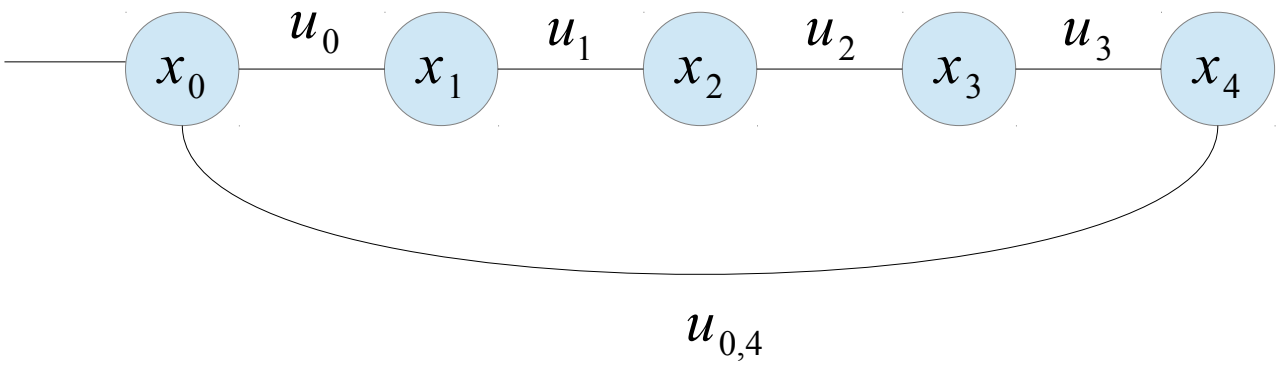

x_0
u_0
x_1
u_1
x_2
u_2
x_3
u_3
x_4
$u_{0,4}$
prior constraint
odometry constraints
loop closure constraints

Beispiel



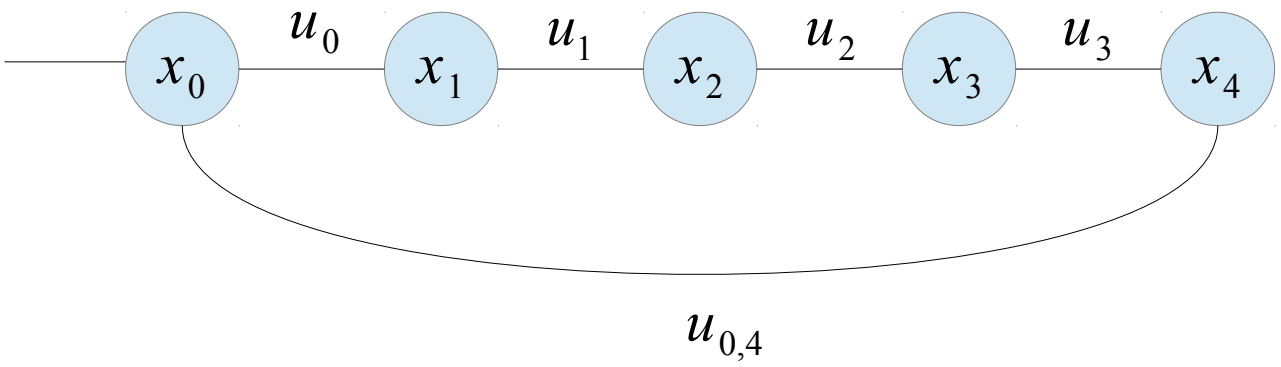
	exakter Wert	Messung
u_0	1	1.1
u_1	1	1.0
u_2	1	1.1
u_3	-3	-2.7
$u_{0,4}$	0	0

Beispiel



	exakter Wert	Messung
x_0	0	0.0
x_1	1	1.1
x_2	2	2.1
x_3	3	3.2
x_4	0	0.5

Beispiel



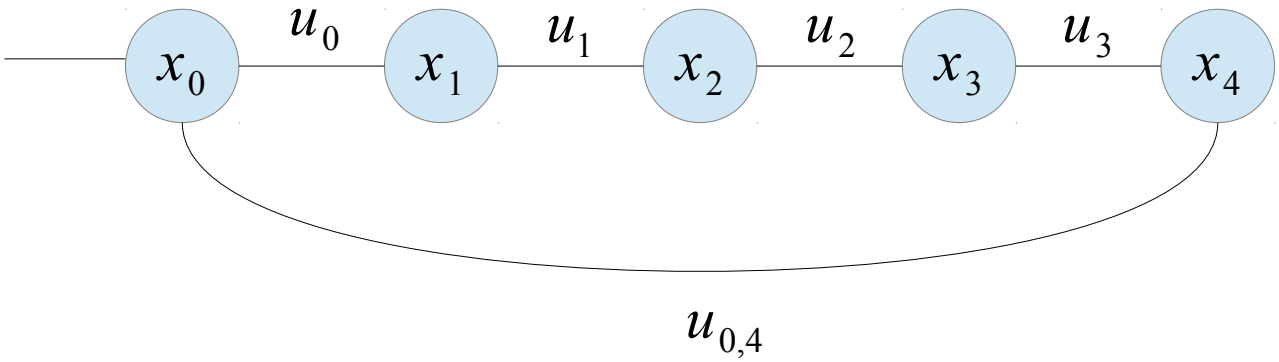
Constraints: $\|g(\boldsymbol{x_i}, \boldsymbol{u_i}) - \boldsymbol{x_{i+1}}\|^2_{\Sigma_i} = \|x_i + \boldsymbol{u_i} - x_{i+1}\|^2_{\Sigma_i}$

$i = 0,1,2,3$

$\|g(\boldsymbol{x_0}, \boldsymbol{u_{0,4}}) - \boldsymbol{x_4}\|^2_{\Lambda_{0,4}} = \|x_0 + \boldsymbol{u_{0,4}} - x_4\|^2_{\Lambda_{0,4}}$

$\|x_0 - 0\|^2_{\Pi}$ prior constraint

Beispiel

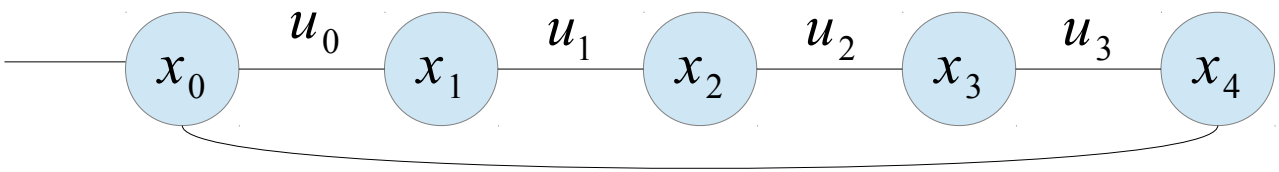


$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{argmin}\, F(\boldsymbol{x}) = \underset{\boldsymbol{x}}{argmin}\, \frac{1}{2} f(\boldsymbol{x})^T \Omega f(\boldsymbol{x})$$

$$f(\boldsymbol{x}) = \begin{pmatrix} x_0 + u_0 - x_1 \\ x_1 + u_1 - x_2 \\ x_2 + u_2 - x_3 \\ x_3 + u_3 - x_4 \\ x_0 + u_{0,4} - x_4 \\ x_0 - 0 \end{pmatrix}$$

$$\Omega = \begin{pmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.001 \end{pmatrix}^{-1}$$

Beispiel



$u_{0,4}$

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{argmin}\, F(\boldsymbol{x}) = \underset{\boldsymbol{x}}{argmin}\, \frac{1}{2} f(\boldsymbol{x})^T \Omega f(\boldsymbol{x})$$

$$f(\boldsymbol{x}) = \begin{pmatrix} x_0 + u_0 - x_1 \\ x_1 + u_1 - x_2 \\ x_2 + u_2 - x_3 \\ x_3 + u_3 - x_4 \\ x_0 + u_{0,4} - x_4 \\ x_0 - 0 \end{pmatrix}$$

$$\Omega = \begin{pmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 \end{pmatrix}$$

Beispiel

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{argmin}\, F(\boldsymbol{x}) = \underset{\boldsymbol{x}}{argmin}\, \frac{1}{2} f(\boldsymbol{x})^T \Omega f(\boldsymbol{x})$$

$$f(\boldsymbol{x}) = \begin{pmatrix} x_0 + u_0 - x_1 \\ x_1 + u_1 - x_2 \\ x_2 + u_2 - x_3 \\ x_3 + u_3 - x_4 \\ x_0 + u_{0,4} - x_4 \\ x_0 - 0 \end{pmatrix} \qquad \Omega = \begin{pmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 \end{pmatrix}$$

$$J = \frac{\partial f}{\partial \boldsymbol{x}} = \begin{pmatrix} \frac{\partial f}{\partial x_0} & \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \frac{\partial f}{\partial x_3} & \frac{\partial f}{\partial x_4} \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Beispiel – Iteration

$$J^T \Omega J \Delta \boldsymbol{x} = -J^T \Omega^T f(\boldsymbol{x})$$

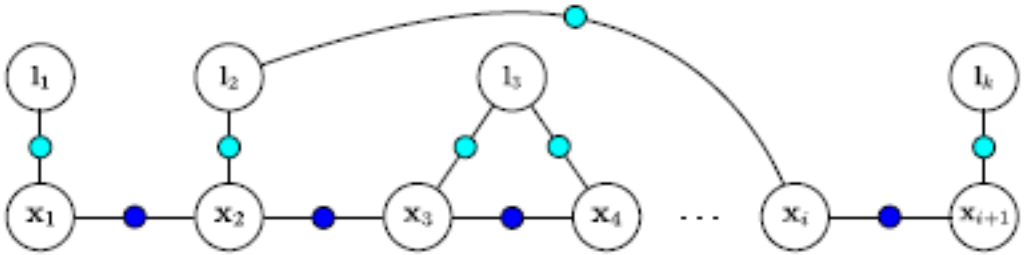
$$\boldsymbol{x} \rightarrow \boldsymbol{x} + \Delta \boldsymbol{x}$$

$$\boldsymbol{x_0} = \begin{pmatrix} 0 & 1.1 & 2.1 & 3.2 & 0.5 \end{pmatrix}^T$$

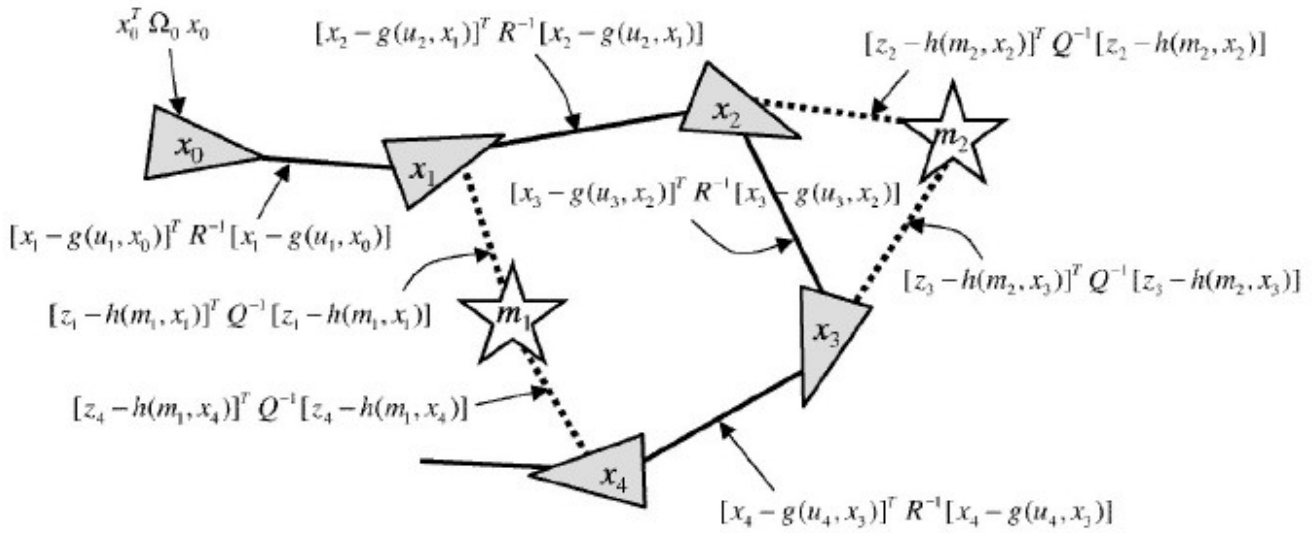
9.3.2 Graph SLAM mit Landmarken

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

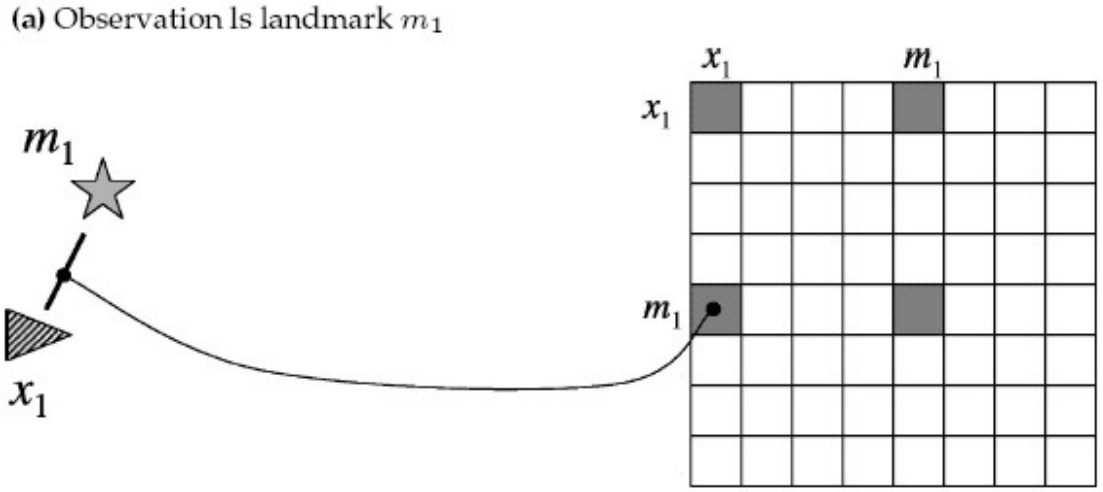


Sum of all constraints:

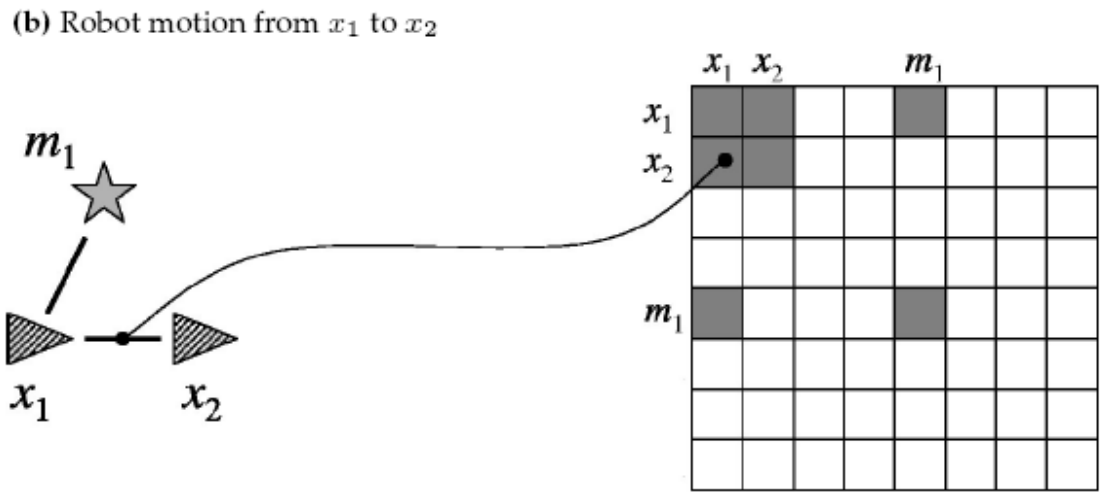
$$J_{\text{GraphSLAM}} = x_0^T \Omega_0 x_0 + \sum_t [x_t - g(u_t, x_{t-1})]^T R^{-1} [x_t - g(u_t, x_{t-1})] + \sum_t [z_t - h(m_{c_t}, x_t)]^T Q^{-1} [z_t - h(m_{c_t}, x_t)]$$

wird minimiert

Konstruktion



Konstruktion



Konstruktion