Teil 5:
Lokalisierung mit einem Kalman-Filter

- Idee
- Eindimensionaler Kalman-Filter
- Mehrdimensionaler Kalman-Filter
- Erweiterter Kalman-Filter (EKF) für nicht-lineare Systeme
- Anwendungen

Idee des Kalman-Filters

Alle Größen sind normalverteilt:

- Die Zustände x_t (z.B. Position) sind normalverteilt.
- Zum Steuerbefehl u_t kommt ein normalverteiltes Rauschen $N(0,\sigma^2)$ dazu
- Die Sensordaten z_t sind ebenfalls normalverteilt

Ablaufs der Kalman-Filterung:

In einer Schleife werden ständig die folgenden beiden Schritte durchgeführt:

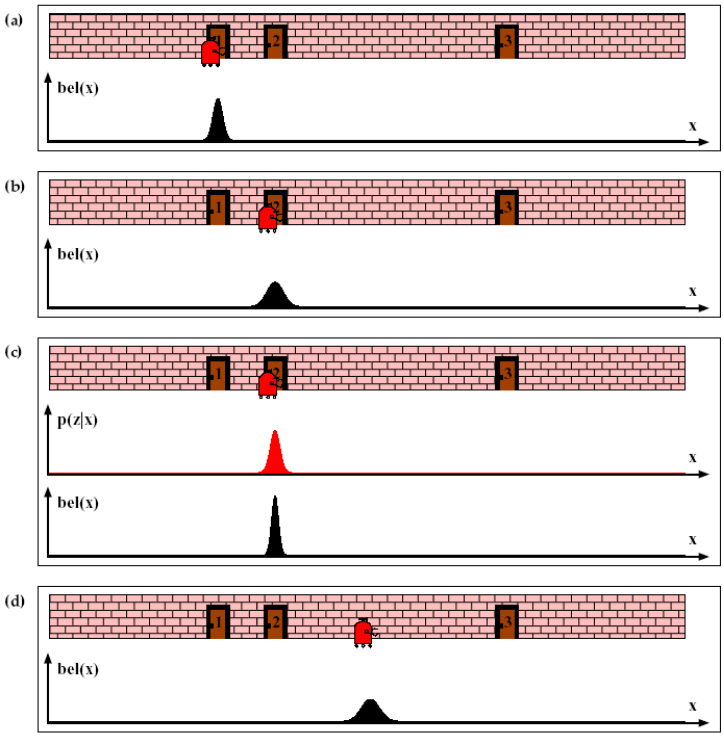
- Vorhersage-Schritt (prediction step):

  Mit Hilfe eines Systemmodells (z.B. Bewegungsmodell wie in Kap. 4) wird aus dem alten Zustand x_{t-1} und einem Steuerbefehl u_t der neue Zustand $\overline{x_t}$ geschätzt.

- Korrektur-Schritt (correction step; measurement update):

  Der Messwert z_t wird mit einer aus $\overline{x_t}$ geschätzten Messung verglichen. Daraus wird ein Korrekturwert berechnet.
  Die neue Schätzung für x_t ergibt sich dann aus $\overline{x_t}$ + Korrekturwert.

Illustrierung des Kalman-Filters



(a) Zustand (hier Position) ist normalverteilt.

(b) Vorhersageschritt: es wird aufgrund eines Steuerbefehls die neue Position geschätzt. Gewissheit der Position nimmt ab!

(c) Korrektur-Schritt: aus der aktuellen Sensormessung wird eine Korrektur der Positionsschätzung vorgenommen. Gewissheit der Position nimmt zu.

(d) Vorhersageschritt: …

Kalman-Filter im Eindimensionalen (1)

Lineare Systemgleichungen:

- Änderung des Zustands und Abhängigkeit der Messwerte vom Zustand werden durch lineare Gleichungen beschrieben:

$x_t = ax_{t-1} + bu_t + \varepsilon$ Systemgleichung

$z_t = cx_t + \delta$ Messgleichung

Zustand x_t, Steuerbefehl u_t und Sensorwert z_t sind eindimensional.
ε und δ ist Gausssches Rauschen (d.h. normalverteilt mit Mittelwert 0)

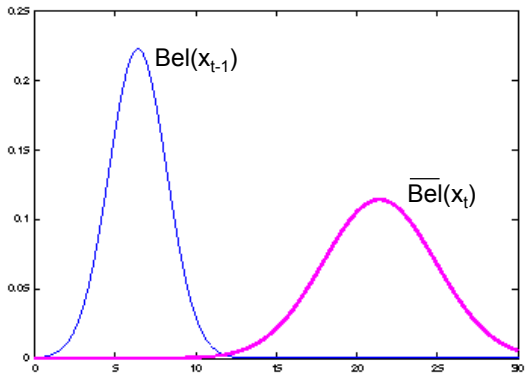
Kalman-Filter im Eindimensionalen (2)

Vohersageschritt:

- x_{t-1} ist normalverteilt
- Mit Hilfe der Systemgleichung und einem Steuerbefehl u_t wird eine Schätzung $\overline{Bel}(x_t)$ berechnet:

  $x_t = ax_{t-1} + bu_t + \varepsilon$

- x_t ist ebenfalls normalverteilt
  (lineare Transformation und Addition mit normalverteiltem Rauschen)



$$Bel(x_{t-1}) \sim N(\mu_{t-1}, \sigma_{t-1}^2)$$

$$\overline{Bel}(x_t) \sim N(\overline{\mu}_t, \overline{\sigma}_t^2) \quad \text{mit}$$

$$\overline{\mu}_t = a\mu_{t-1} + bu_t$$

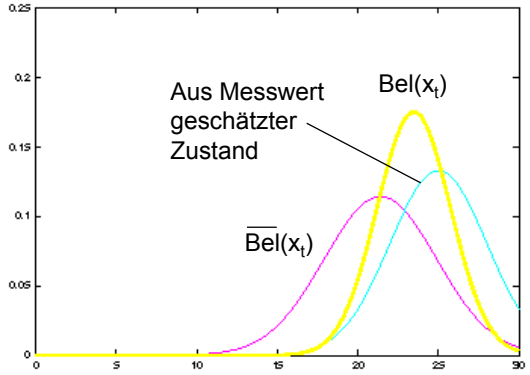
$$\overline{\sigma}_t^2 = a^2\sigma_t^2 + \sigma_\varepsilon^2$$

Kalman-Filter im Eindimensionalen (3)

Korrektur-Schritt:

- Aus dem vorhergesagtem Zustand $\overline{\mu}_t$ und der Messgleichung lässt sich ein Messwert $c\overline{\mu}_t$ schätzen.
- Aus der Differenz des tatsächlichen Messwertes z_t und des geschätzten Messwerts $c\overline{\mu}_t$ lässt sich Zustand $\overline{\mu}_t$ korrigieren:

$$\mu_t = \overline{\mu}_t + k_t(z_t - c\overline{\mu}_t) \quad \text{mit} \quad k_t = \frac{c\overline{\sigma}_t^2}{c^2\overline{\sigma}_t^2 + \sigma_\delta^2}$$



$$\overline{Bel}(x_{t-1}) \sim N(\overline{\mu}_t, \overline{\sigma}_t^2)$$

$$Bel(x_t) \sim N(\mu_t, \sigma_t^2) \quad \text{mit}$$

$$\mu_t = \overline{\mu}_t + k_t(z_t - c\overline{\mu}_t)$$

$$\sigma_t^2 = (1 - ck_t)\overline{\sigma}_t^2$$

$$k_t = \frac{c\overline{\sigma}_t^2}{c^2\overline{\sigma}_t^2 + \sigma_\delta^2}$$

Kalman-Filter im Eindimensionalen (4)

Anmerkung: Korrektur-Schritt ist gewichtete Mittelwertsbildung:

- Aus Meßgleichung folgt

  $$x_t = c^{-1} z_t - c^{-1}\delta$$

  Damit ist $c^{-1} z_t$ ein Schätzwert für den Zustand x_t mit Varianz $c^{-2}\,\sigma_\delta^2$.

- Damit läßt sich eine neue Zustandsschätzung als gewichteter Mittelwert aus μ_t und $c^{-1} z_t$ bilden:

$$\mu_t = \frac{\frac{1}{\bar{\sigma}_t^2}\bar{\mu}_t + \frac{1}{c^{-2}\sigma_\delta^2} c^{-1} z_t}{\frac{1}{\bar{\sigma}_t^2} + \frac{1}{c^{-2}\sigma_\delta^2}}$$

$$= \frac{\sigma_\delta^2 \bar{\mu}_t + c^2 \bar{\sigma}_t^2 c^{-1} z_t}{c^2 \bar{\sigma}_t^2 + \sigma_\delta^2} \qquad \text{mit } \bar{\sigma}_t^2 \sigma_\delta^2 \text{ erweitert}$$

$$= \frac{c^2 \bar{\sigma}_t^2 \bar{\mu}_t + \sigma_\delta^2 \bar{\mu}_t + c^2 \bar{\sigma}_t^2 c^{-1} z_t - c\bar{\sigma}_t^2 c\bar{\mu}_t}{c^2 \bar{\sigma}_t^2 + \sigma_\delta^2} \qquad c^2 \bar{\sigma}_t^2 \bar{\mu}_t - c\bar{\sigma}_t^2 c\bar{\mu}_t = 0 \text{ im Zähler dazugeschreiben}$$

$$= \frac{(c^2 \bar{\sigma}_t^2 + \sigma_\delta^2)\bar{\mu}_t}{c^2 \bar{\sigma}_t^2 + \sigma_\delta^2} + \frac{c^2 \bar{\sigma}_t^2 c^{-1} z_t - c\bar{\sigma}_t^2 c\bar{\mu}_t}{c^2 \bar{\sigma}_t^2 + \sigma_\delta^2}$$

$$= \bar{\mu}_t + k_t (z_t - c\bar{\mu}_t) \qquad \text{mit} \quad k_t = \frac{c\bar{\sigma}_t^2}{c^2 \bar{\sigma}_t^2 + \sigma_\delta^2}$$

Kalman-Filter im Eindimensionalen (5)

Sytem- und Messgleichung:

$$x_t = a_t x_{t-1} + b_t u_t + \varepsilon_t$$

$$z_t = c_t x_t + \delta_t$$

$$\varepsilon_t \sim N(0, \sigma_{\varepsilon,t}^2) \quad \text{und} \quad \delta_t \sim N(0, \sigma_{\delta,t}^2)$$

Algorithmus KalmanFilter(μ_{t-1}, σ_{t-1}, u_t, z_t):

1. Vorhersage:
2. $\bar{\mu}_t = a_t \mu_{t-1} + b_t u_t$
3. $\bar{\sigma}_t^2 = a_t^2 \sigma_{t-1}^2 + \sigma_{\varepsilon,t}^2$
4. Korrektur:
5. $k_t = \bar{\sigma}_t^2 c (c^2 \bar{\sigma}_t^2 + \sigma_{\delta,t}^2)^{-1}$
6. $\mu_t = \bar{\mu}_t + k_t (z_t - c_t \bar{\mu}_t)$
7. $\sigma_t^2 = (1 - k_t c_t)\bar{\sigma}_t^2$
8. return μ_t, σ_t

Kalman-Filter im Mehrdimensionalen

Sytem- und Messgleichung:

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \qquad x_t \in \Re^n, u_t \in \Re^m$$

$$z_t = C_t x_t + \delta_t \qquad z_t \in \Re^k$$

$$\varepsilon_t \sim N(0, R_t) \quad \text{und} \quad \delta_t \sim N(0, Q_t)$$

Algorithmus KalmanFilter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

1. Vorhersage:
2. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
3. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$
4. Korrektur:
5. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
6. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
7. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$
8. return μ_t, Σ_t

Unsicherheit in den Steuerdaten

- Wie man beim Odometriemodell gesehen hat, können Steuerdaten gemessene Daten sein.
- Es ist daher mit einem Unterschied zwischen den tatsächlichen Steuerdaten und den bekannten (gemessenen) Steuerdaten zu rechnen.

$$u_t \sim N(\mu_{u,t}, \sum_{u,t})$$

- Damit ist:

$$B_t u_t \sim N(B_t \mu_{u,t}, B_t \sum_{u,t} B_t^T)$$

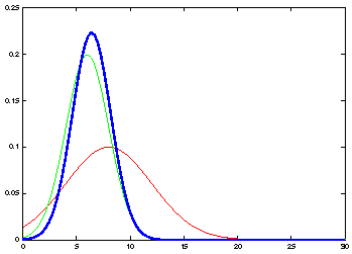
- Damit ist der Vorhersageschritt wie folgt anzupassen:

1. Vorhersage:
2. $\bar{\mu}_t = A_t \mu_{t-1} + B_t \mu_{u,t}$
3. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + B_t \Sigma_{u,t} B_t^T + R_t$

- Man könnte natürlich auch $B_t \sum_{u,t} B_t^T + R_t$ zu einer neuen Kovarianz R_t' zusammenfassen. Dann würde sich an der Formulierung des Algorithmus nichts ändern.

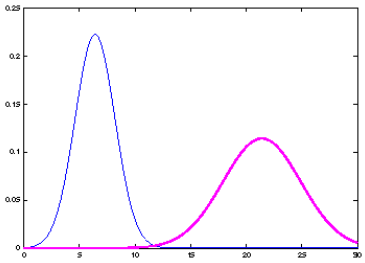
Kalman-Filter – Zusammenfassung (1)

Vorhersageschritt



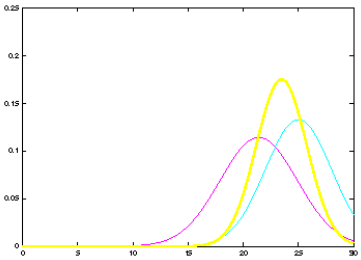
$$\overline{bel}(x_t) = \begin{cases} \overline{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \overline{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \overline{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$



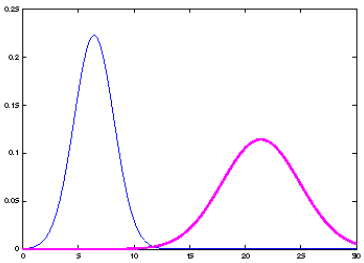
Kalman-Filter – Zusammenfassung (2)

Korrrekturschrittschritt



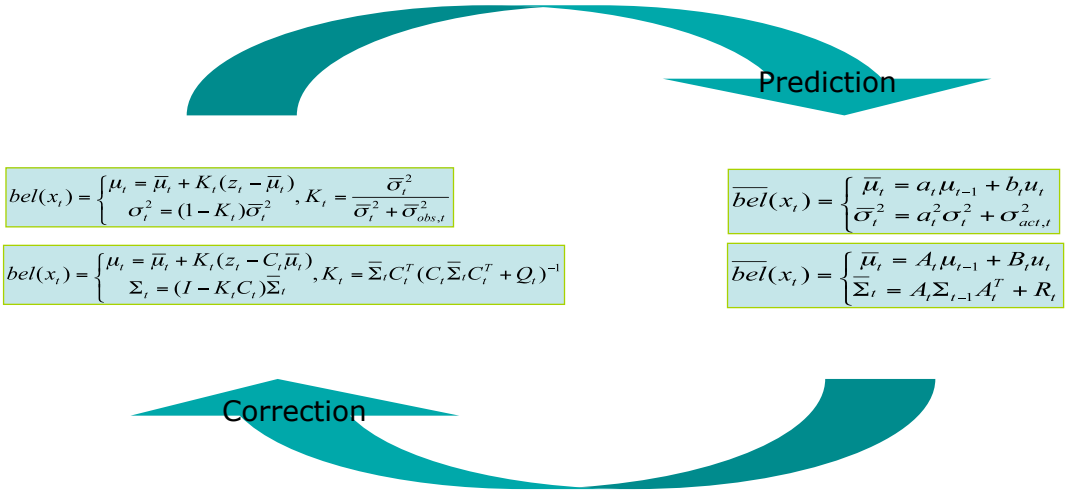
$$bel(x_t) = \begin{cases} \mu_t = \overline{\mu}_t + K_t(z_t - \overline{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\overline{\sigma}_t^2 \end{cases}, K_t = \frac{\overline{\sigma}_t^2}{\overline{\sigma}_t^2 + \overline{\sigma}_{obs,t}^2}$$

$$bel(x_t) = \begin{cases} \mu_t = \overline{\mu}_t + K_t(z_t - C_t\overline{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\overline{\Sigma}_t \end{cases}, K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$$



Kalman-Filter – Zusammenfassung (3)

Vorhersage-Korrrektur-Zyklus



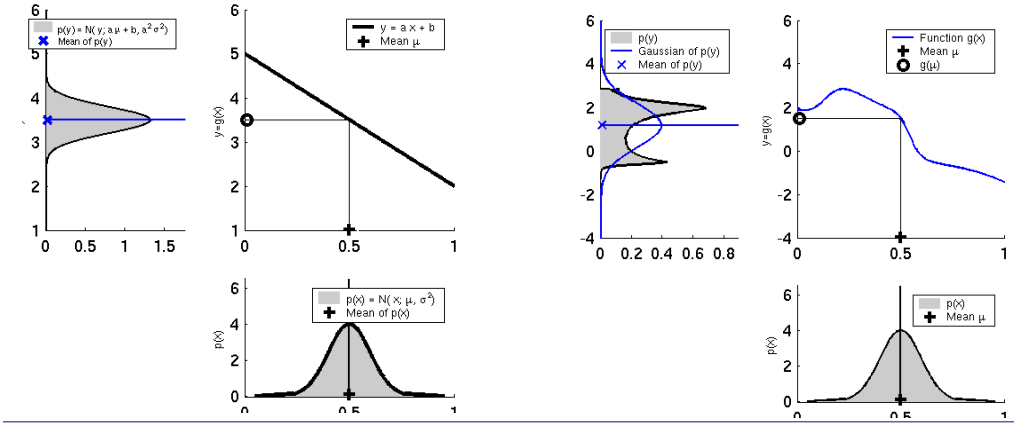
Erweiterter Kalman-Filter - EKF

Nichtlineare Systeme

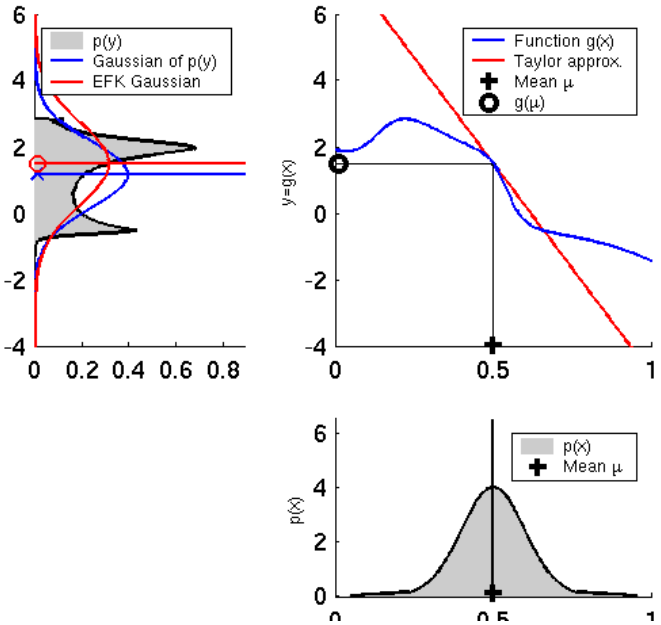
In vielen praktischen Situationen sind System- und Messverhalten nichtlinear.

$$\begin{aligned} x_t &= g(x_{t-1}, u_t) + \varepsilon_t && x_t \in \Re^n, u_t \in \Re^m \\ z_t &= h(x_t) + \delta_t && z_t \in \Re^k \\ \varepsilon_t &\sim N(0, R_t) \quad \text{und} && \delta_t \sim N(0, Q_t) \end{aligned}$$

Unterschieds zwischen linearem und nichtlinearen Verhalten



EKF-Linearisierung (1)



Linearisierung durch Taylor-Entwicklung:

$$g(x) \approx g'(\mu)(x-\mu) + g(\mu)$$

EKF-Linearisierung (2)

Ausgangsproblem: Nichtlineares System

Zusätzlich wollen wir annehmen, dass die Steuerdaten u_t einer Normalverteilung unterliegen.

$$\begin{aligned} x_t &= g(x_{t-1}, u_t) + \varepsilon_t \qquad & x_t \in \Re^n, u_t \in \Re^m \\ z_t &= h(x_t) + \delta_t & z_t \in \Re^k \\ u_t &\sim N(\mu_{u,t}, \Sigma_{u,t}), \quad \varepsilon_t \sim N(0, R_t) \quad \text{und} \quad \delta_t \sim N(0, Q_t) \end{aligned}$$

EKF-Linearisierung (3)

Linearisierung von g:

Mit der gleichen Technik wie in Kap. 4 (Fehlerberechnung beim Odometriemodell):

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

$$G_t = \frac{\partial g(\mu_{t-1}, u_t)}{\partial (x_{t-1}, u_t)} \text{ ist die Jakobi - Matrix}$$

Zweckmäßigerweise wird G_t wie in Kap. 4 in zwei Teilmatrizen zerlegt:

$$G_{x,t} = \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}$$

$$G_{u,t} = \frac{\partial g(\mu_{t-1}, u_t)}{\partial u_t}$$

Linearisierung von h:

$$h(x_t) \approx h(\overline{\mu}_t) + H_t(x_t - \overline{\mu}_t)$$

$$H_t = \frac{\partial h(\overline{\mu}_t)}{\partial x_t} \text{ ist die Jakobi - Matrix}$$

EKF-Algorithmus

Algorithmus ExtendedKalmanFilter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

1. Vorhersage:
2. $\overline{\mu}_t = g(\mu_{t-1}, \mu_{u,t})$
3. $\overline{\Sigma}_t = G_{x,t}\Sigma_{t-1}G_{x,t}^T + G_{u,t}\Sigma_{u,t}G_{u,t}^T + R_t$
4. Korrektur:
5. $K_t = \overline{\Sigma}_t H_t^T (H_t \overline{\Sigma}_t H_t^T + Q_t)^{-1}$
6. $\mu_t = \overline{\mu}_t + K_t(z_t - h(\overline{\mu}_t))$
7. $\Sigma_t = (I - K_t H_t)\overline{\Sigma}_t$
8. return μ_t, Σ_t

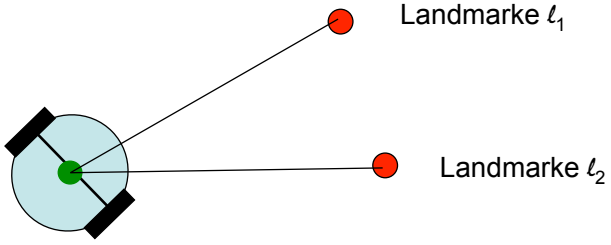
Teil 5:
Lokalisierung mit einem Kalman-Filter

- Idee
- Eindimensionaler Kalman-Filter
- Mehrdimensionaler Kalman-Filter
- Erweiterter Kalman-Filter (EKF) für nicht-lineare Systeme
- Anwendungen
  - Selbstlokalisierung mit Landmarken
  - Selbstlokalisieung mit Matching
  - SLAM-Verfahren

Selbstlokalisierung mit Landmarken (1)

- **Landmarken:**
  Positionen der beiden Landmarken sind bekannt:
  (x_{t1}, y_{t1}) und (x_{t2}, y_{t2})
- **Roboterposition:**

$$x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$



- **Sensordaten z_t:**
  Der Roboter misst für jede Landmarke nach jedem Zeitintervall Δt Abstand d und Winkel α relativ zum lokalen Koordinatensystem:

$$z_t = (d_1, \alpha_1, d_2, \alpha_2)^T$$

- **Geschwindigkeitsmodell als Bewegungsmodell (siehe Kap. 4):**

$$u_t = \begin{pmatrix} d \\ \alpha \end{pmatrix} \qquad x_t = \begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + d\cos\theta \\ y + d\sin\theta \\ \theta + \alpha \end{pmatrix} \quad \text{mit } x_{t-1} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

Selbstlokalisierung mit Landmarken (2)

Systemgleichung (nichtlineares System)

$$x_t = g(\underbrace{x, y, \theta}_{x_{t-1}}, \underbrace{d, \alpha}_{u_t}) = \begin{pmatrix} x + d\cos\theta \\ y + d\sin\theta \\ \theta + \alpha \end{pmatrix} + \varepsilon_t$$

$$\varepsilon_t \sim N(0, R)$$

$$u_t \sim N(u_t, \Sigma_u) \quad \text{mit } \Sigma_u = \begin{pmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_\alpha^2 \end{pmatrix}$$

Jacobi-Matrizen:

$$G_{x,t} = \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}} = \begin{pmatrix} 1 & 0 & -d\sin\theta \\ 0 & 1 & d\cos\theta \\ 0 & 0 & 1 \end{pmatrix}$$

$$G_{u,t} = \frac{\partial g(\mu_{t-1}, u_t)}{\partial u_t} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix}$$

Selbstlokalisierung mit Landmarken (3)

Messgleichung

$$z_t = \begin{pmatrix} d_1 \\ \alpha_1 \\ d_2 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} \sqrt{(x - x_{\ell 1})^2 + (y - y_{\ell 1})^2} \\ \operatorname{atan2}(y_{\ell 1} - y, x_{\ell 1} - x) - \theta \\ \sqrt{(x - x_{\ell 2})^2 + (y - y_{\ell 2})^2} \\ \operatorname{atan2}(y_{\ell 2} - y, x_{\ell 2} - x) - \theta \end{pmatrix} + \delta = h(\underbrace{x, y, \theta}_{x_t}) + \delta$$

$$\delta \sim N(0, Q)$$

Jacobi-Matrix von h:

$$H_t = \frac{\partial h(\overbrace{x', y', \theta'}^{x_t})}{\partial x_t} = \begin{pmatrix} \frac{x' - x_{\ell 1}}{r_{\ell 1}} & \frac{y' - y_{\ell 1}}{r_{\ell 1}} & 0 \\ \frac{-(y' - y_{\ell 1})}{r_{\ell 1}^2} & \frac{x' - x_{\ell 1}}{r_{\ell 1}^2} & -1 \\ \frac{x' - x_{\ell 2}}{r_{\ell 2}} & \frac{y' - y_{\ell 2}}{r_{\ell 2}} & 0 \\ \frac{-(y' - y_{\ell 2})}{r_{\ell 2}^2} & \frac{x' - x_{\ell 2}}{r_{\ell 2}^2} & -1 \end{pmatrix} \quad \text{mit} \quad r_{\ell i} = \sqrt{(x' - x_{\ell i})^2 + (y' - y_{\ell i})^2}, \; i = 1,2$$

Anmerkung zu atan2

Nützliche Erweiterung vom atan:

$$\operatorname{atan}2(y,x) = \begin{cases} \operatorname{atan}(y/x) & \text{falls } x > 0 \\ \operatorname{sign}(y)(\pi - \operatorname{atan}(|y/x|)) & \text{falls } x < 0 \\ 0 & \text{falls } x = y = 0 \\ \operatorname{sign}(y)\pi/2 & \text{falls } x = 0, y \neq 0 \end{cases}$$

Selbstlokalisierung mit Landmarken (4)

Beispielfahrt mit einer Landmarke

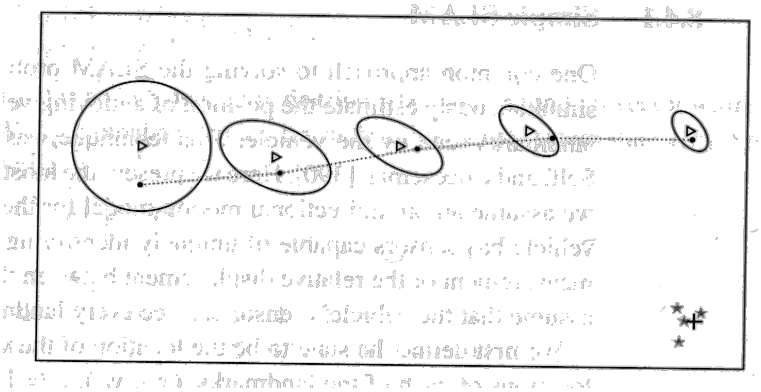


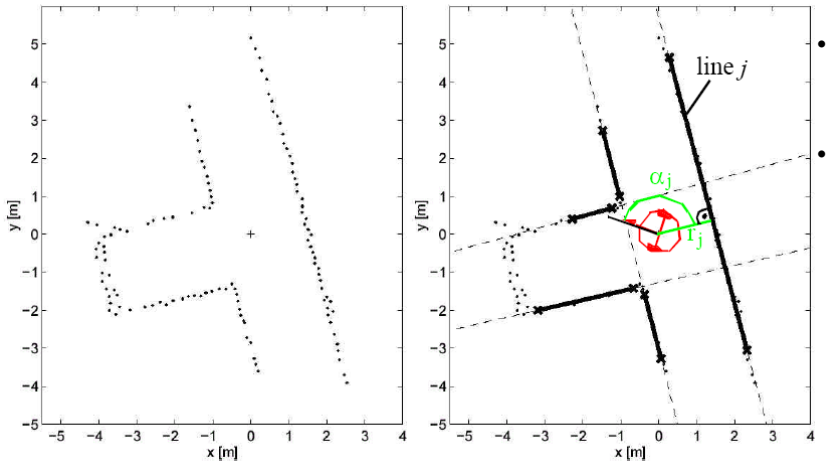
Abbildung aus [3].

- Die gepunktete Linie ist die tatsächliche Wegstrecke.
- Die σ-Ellipsen und die Dreiecke und stellen die geschätze Lage bzw. Orientierung dar.
- Das Kreuz ist die Landmarke. Die Sterne geben die gemessene Position der Landmarke an. Es ist deutlich das Messrauschen zu sehen.
- Die Anwendung kann auf n Landmarken erweitert werden. Zusätzlich kann noch das Problem der Zuordnung der Messdaten zu den Landmarken-Nummern dazukommen. (siehe auch [1])

Selbstlokalisierung mit Matching (1)

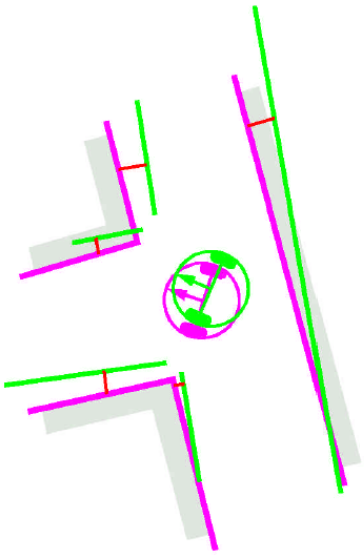
Matching-Verfahren (aus [2])

- Wird ein gemessenes Abstandsprofil (z.B. von einem Laser aufgenommener Scan) gegen eine metrische Karte abgeglichen, dann spricht man von einem Matching-Verfahren.
- Statt des Abstandprofils können auch daraus gewonnene Merkmale wie Linien, Ecken etc. verwendet werden:



- Links: Abstandsprofil von einem Laser (Scan)
- Rechts: Extrahierte Linien

Selbstlokalisierung mit Matching (2)



Matching-Beispiel

- Die grünen Linen stellen die gemessenen Daten dar. Sie wurden aus dem Scan gewonnen.
- Die rosa Linen sind die aufgrund der aktuellen Positionsschätzung und der Umgebungskarte vorhergesagten Linien.

Selbstlokalisierung mit Matching (3)

EKF-Algorithmus:

- Die Positionsvorhersage (Zeile 1 bis 3) kann wie in der vorhergehenden Anwendung durchgeführt werden. Wir erhalten eine Positionsschätzung $\bar{\mu}_t$.
- Der Korrekturschritt (Zeile 4 bis 7) wird in etwa wie folgt realisiert.

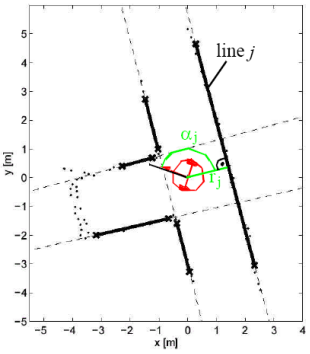
Messdaten z_t:

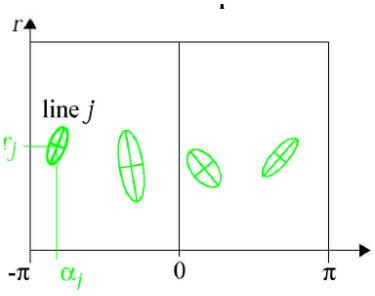
- Aus einem Scan s_t werden einen Menge von n_t Linien mit Kovarianzen gewonnen.
- Eine Linie ist dabei eindeutig festgelegt durch den senkrechten Abstand r und die Orientierung α der Normalen (r und α im lokalen Koordinatensystem)

$$z_t = \left(r_1, \alpha_1, \ldots r_{n_t}, \alpha_{n_t}\right)^T$$

$$Q = \begin{pmatrix} \Sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Sigma_{n_t} \end{pmatrix}$$

$$\text{mit} \quad \Sigma_i = \begin{pmatrix} \sigma_d^2 & \sigma_{d\alpha} \\ \sigma_{d\alpha} & \sigma_\alpha^2 \end{pmatrix}$$

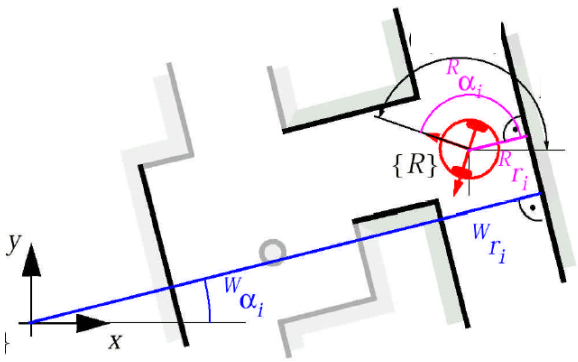


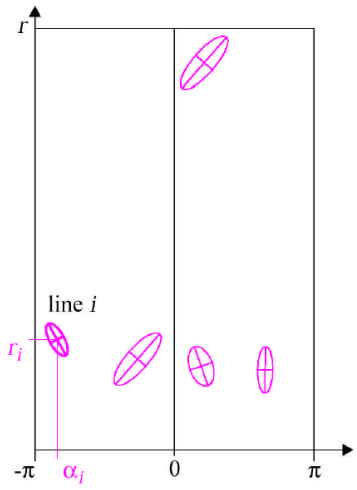


Selbstlokalisierung mit Matching (4)

Vorhersage der Messdaten $h(\bar{\mu}_t)$:

- Mit der geschätzten Position $\bar{\mu}_t$ und der Umgebungskarte m werden alle sichtbaren Linien ermittelt. Die Umgebungskarte ist dabei günstigerweise ein Linenmodell.
- Die Linien werden in das lokale Koordinatensystems des Roboters transformiert.

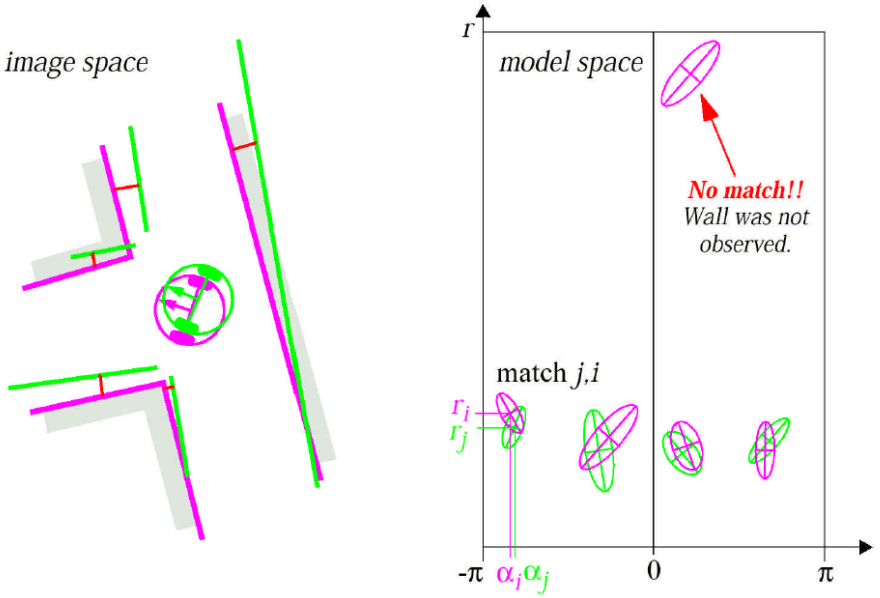




Selbstlokalisierung mit Matching (5)

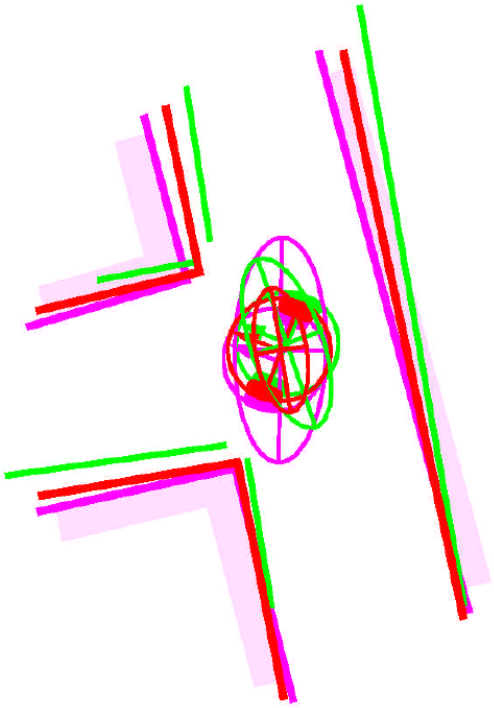
Korrektur:

- Bevor die Korrektur (Zeile 6) durchgeführt werden kann, müssen die gemessenen Linien (grün) so umgeordnet werden, dass sie zu den vorhergesagten Linien (rosa) korrespondieren. Es kann Linien ohne korrespondierenden Linienpartner geben.



Selbstlokalisierung mit Matching (6)

Illustrierung der Korrektur



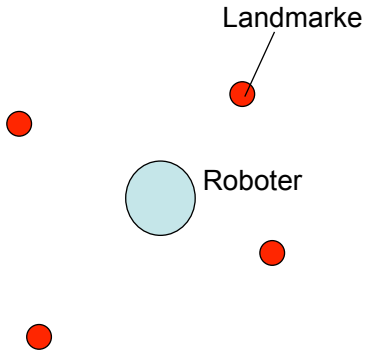
Rosa: aus Steuerdaten geschätzte Position

Grün: durch Messung geschätzte Position

Rot: korrigierte Position x_t

SLAM-Verfahren (1)

- **SLAM = Simultaneous Localization and Mapping**
  (aus [3], stark vereinfachtes Modell)
- **n Landmarken:**
  Positionen sind unbekannt (!):
  $(x_{\ell 1}, y_{\ell 1})$, $(x_{\ell 2}, y_{\ell 2})$ …, $(x_{\ell n}, y_{\ell n})$
- **Roboter mit omnidirektionalem Antrieb:**
  - Antrieb in alle Richtungen
  - Roboterfront zeigt immer in Richtung globale x-Achse
  - Roboterposition (x_r, y_r)



SLAM-Verfahren (2)

- **Zustand x_t:**
  besteht aus Landmarken- und Roboterposition:

$$x_t = (x_r, y_r, x_{\ell 1}, y_{\ell 1}, x_{\ell 2}, y_{\ell 2}, \ldots, x_{\ell n}, y_{\ell n})^T$$

- **Omnidiektionales Bewegungsmodell:**

$$u_t = (v_x \quad v_y)^T$$

- **Systemgleichung:**

$$x_t = x_{t-1} + \begin{pmatrix} \Delta t & 0 \\ 0 & \Delta t \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix} u_t + \varepsilon_t$$

$$\varepsilon_t = (\varepsilon_{vx}, \varepsilon_{vy}, 0, \ldots 0)^T \quad mit \quad (\varepsilon_{vx}, \varepsilon_{vy})^T \sim N(0, R)$$

Landmarken bewegen sich nicht!

SLAM-Verfahren (3)

- **Sensordaten z_t:**
  Landmarken seien eindeutig identifizierbar.
  Für jede Landmarke wird ihre Position relativ zum Roboter gemessen:

$$z_t = \begin{pmatrix} x_{t1} - x_r \\ y_{t1} - y_r \\ \vdots \\ x_{tn} - x_r \\ y_{tn} - y_r \end{pmatrix}$$

- **Messgleichung:**

$$z_t = \begin{pmatrix} H_1 \\ \vdots \\ H_n \end{pmatrix} x_t + \partial_t \text{, wobei}$$

$$H_i = \begin{pmatrix} -1 & 0 & 0 & \cdots & 0 & \underbrace{1}_{(2i+1)\text{-te Spalte}} & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}$$

$$\delta_t \sim N(0, Q)$$

SLAM-Verfahren (4)

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

- **Erweiterungen**
  - Das Bewegungsmodel kann wie in der 1. Anwendung (Lokalisierung mit Landmarken) auf ein Geschwindigkeitsmodell erweitert werden. Zum Systemzustand kommt dann noch die Orientierung dazu.
  - Statt bei jeder Landmarke die relative Position zu messen, kann auch Abstand und Orientierung (im lokalen Koordinatensystem) gemessen werden.
  - Zusätzlich kann noch das Problem der Zuordnung der Messdaten zu den Landmarken-Nummern dazukommen.