Robotik

Priv.-Doz. Dr. Thomas Wiemann
Institut für Informatik
Autonome Robotik

SoSe 2021

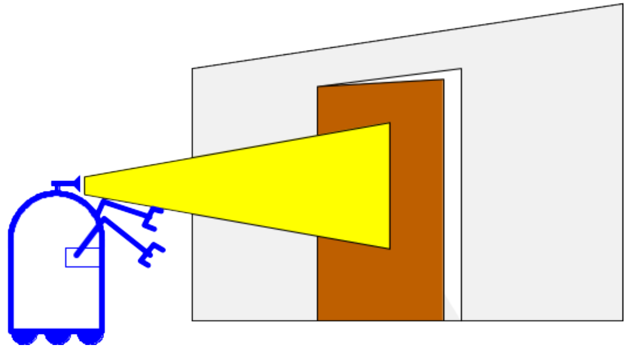
UNIVERSITÄT OSNABRÜCK

Bayes-Filter - Motivation

- Angenommen, ein Roboter macht die Messung z
- Was ist $P(\text{open}, z)$?
- Kausales vs. diagnostisches schließen
- $P(\text{open} \mid z)$ is diagnostisch
- $P(z \mid \text{open})$ ist kausal
- Die Bayes'sche Regel erlaubt es uns kausales Wissen zu nutzen:

Zählen!

$$\boxed{P(\text{open} \mid z) = \frac{P(z \mid \text{open}) P(\text{open})}{P(z)}}$$



UNIVERSITÄT OSNABRÜCK

Beispiel Bayes-Filter

- Beispiel einer einfachen Messung

$$P(z \mid \text{open}) = 0.6 \qquad\qquad P(z \mid \neg\text{open}) = 0.3$$

$$P(\text{open}) = P(\neg\text{open}) = 0.5$$

$$P(\text{open} \mid z) = \frac{P(z \mid \text{open})P(\text{open}}{P(z \mid \text{open})P(\text{open}) + P(z \mid \neg\text{open})P(\neg\text{open}}$$

$$P(\text{open} \mid z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67$$

- z erhöht die geschätzte Wahrscheinlichkeit, dass die Tür offen ist

Bayes-Filter - Rekursives updaten

- Angenommen, unser Roboter führt eine weitere Messung durch
- Wie können wir diese Information berücksichtigen?
- Allgemein: Wie können wir $P(x \mid z_1 \ldots z_n)$ berechnen?

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x, z_1, \ldots, z_{n-1}) P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

- **Markov-Annahme**: z_n ist unabhängig von $z_1, \ldots, z_{n-1}$ wenn wir x kennen.

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x) P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

$$= \alpha P(z_n \mid x) P(x \mid z_1, \ldots, z_{n-1})$$

$$= \alpha_n \sum_{i=1 \ldots n} P(z_i \mid x) P(x)$$

Zweite Messung

- Zweite Messung

$$P(z_2 \mid \text{open}) = 0.5 \qquad P(z_2 \mid \neg\text{open}) = 0.6$$

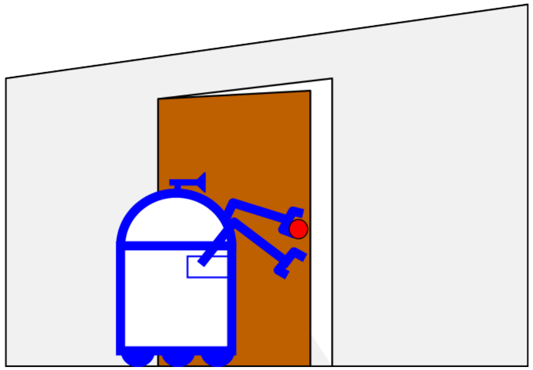
$$P(\text{open} \mid z_1) = 2/3$$

$$P(\text{open} \mid z_1, z_2) = \frac{P(z_2 \mid \text{open}) P(\text{open} \mid z_1)}{P(z_2 \mid \text{open}) P(\text{open} \mid z_1) + P(z_2 \mid \neg\text{open}) P(\neg\text{open})}$$

$$P(\text{open} \mid z_1, z_2) = \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625$$

- z_2 reduziert die Wahrscheinlichkeit, dass die Tür offen ist.

Aktionen

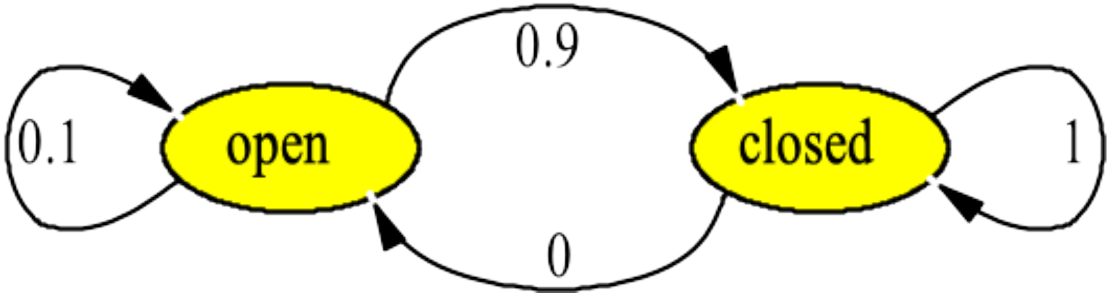


- In den meisten Fällen ist die Welt dynamisch
- Aktionen des Roboters oder von anderen Agenten
- Wie können wir solche Aktionen in unser Modell integrieren?
- Typische Aktionen:
  - Der Roboter dreht seine Räder
  - Der Roboter benutzt seinen Manipulator, um ein Objekt zu greifen
  - Planzen wachsen über die Zeit
- Aktionen haben nicht mit 100% Wahrscheinlichkeit das gewünschte Resultat
- Im Gegensatz zu Messungen erhöhen Aktionen im Allgemeinen die Unsicherheit
- Zur Modellierung fügen wir die Aktion u in den aktuellen Belief ein

$$P(x_t \mid u, x_{t-1})$$

Zustandsübergänge

- $P(x_t \mid u, x_{t-1})$ für $u =$ "schließe die Tür"



- Wenn die Tür offen ist, ist die Aktion in 90% aller Fälle erfolgreich
- Integration über die Ausgänge:

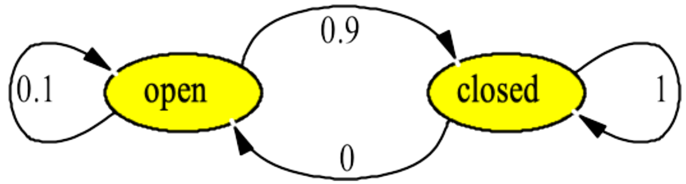
$$P(x_t \mid u) = \int P(x \mid u, x_{t-1}) P(x_{t-1}) dx_{t-1}$$

(kontinuierlich)

$$P(x_t \mid u) = \sum P(x \mid u, x_{t-1}) P(x_{t-1})$$

(diskret)

Dritte Aktion: “Tür öffnen”



$$
\begin{aligned}
P(closed \mid u) &= \sum P(closed \mid u, x')P(x') \\
&= P(closed \mid u, open)P(open) \\
&\quad + P(closed \mid u, closed)P(closed) \\
&= \frac{9}{10} * \frac{5}{8} + \frac{1}{1} * \frac{3}{8} = \frac{15}{16}
\end{aligned}
$$

$$
\begin{aligned}
P(open \mid u) &= \sum P(open \mid u, x')P(x') \\
&= P(open \mid u, open)P(open) \\
&\quad + P(open \mid u, closed)P(closed) \\
&= \frac{1}{10} * \frac{5}{8} + \frac{0}{1} * \frac{3}{8} = \frac{1}{16} \\
&= 1 - P(closed \mid u)
\end{aligned}
$$

Bayes-Filter: Modell (1)

- Gesucht: evidenzbasierte Zustandsschätzung

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = f\left(\mathbf{e}_{1:t+1}, \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})\right)$$

- Für irgendeine Funktion f und Evidenzen (Aktion oder Messung) e
- Unter der **Markov-Annahme**: Eine Evidenz ist unabhängig von allen anderen, gegeben der aktuelle Zustand
- Ansatz:

$$\begin{aligned}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) &= \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}, \mathbf{e}_{t+1}) \\ &= \eta \cdot \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}, \mathbf{e}_{1:t}) \cdot \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) && \text{[Bayessche Regel]} \\ &= \eta \cdot \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \cdot \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) && \text{[Markow-Annahme]}\end{aligned}$$

Filterung

Prädiktion

Bayes-Filter: Modell (2)

- Vereinfachung der Prädiktion:

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) = \int_{\mathbf{x}_t} \left[\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t, \mathbf{e}_{1:t}) \cdot P(\mathbf{x}_t|\mathbf{e}_{1:t})\right]$$

Aktuelle Wahrscheinlichkeit für Zustand

$$= \int_{\mathbf{x}_t} \left[\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t) \cdot P(\mathbf{x}_t|\mathbf{e}_{1:t})\right]$$

[Markow-Annahme]

- Zusammengefasst:

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = \eta \cdot \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \cdot \int_{\mathbf{x}_t} \left[\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t) \cdot P(\mathbf{x}_t|\mathbf{e}_{1:t})\right]$$

- So nicht praktikabel:
  - kontinuierliche Verteilungen, Integration
- Unterschiedliche Vereinfachungen / Approximationen möglich

Spezieller Bayes-Filter (Idee)

- Voraussetzung: Roboterhandeln + Umgebungsdynamik wird beschrieben als Folge von abwechselnd Aktion / Ereignis und Zustandsmessung, gegeben Start-Zustandsschätzung

1.Prädiktion:

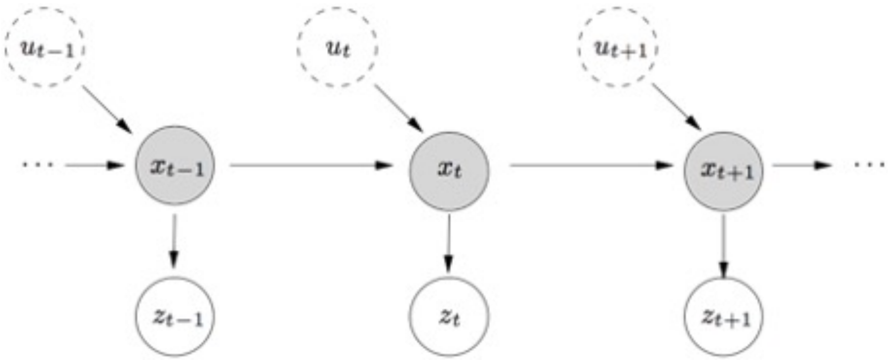
- Sage resultierenden Zustand für Aktion/Ereignis im aktuellen Zustand voraus (a-priori-Zustandsschätzung)
- (z.B. berechne erwartete Zielpose für Fahrbefehl);
- führe die Aktion aus (bzw. das unabhängige Ereignis geschieht)

2.Filterung:

- Miss resultierenden Zustand; ggf. mach mehrere unabhängige Messungen (die sich widersprechen dürfen)
- z.B. miss Pose durch Odometrie und durch IMU;
- berechne aktuelle Zustandsverteilung, die maximal gut zu allen Messungen und der Vorhersage passt (a-posteriori-Zustandsschätzung)
- weiter bei 1.

UNIVERSITÄT OSNABRÜCK

Handeln und Messen



- Erziele Unabhängigkeit durch Abwechseln von Messen-Handeln-Messen-Handeln-…
- **Konvention**: Sortiere Evidenzen in Aktionen u_i, Messungen z_i
- **Konvention**: Wahrscheinlichkeit des Zustands x heißt *Bel*(x) (“Belief”)

$$Bel(x_t) = P(x_1 \mid u_1, z_1, \ldots, u_t, z_t)$$

- Vereinfachung: Diskrete Zustände ➠ Ersetze $\int$ durch $\sum$

$$Bel(x_t) = \eta P(z_t \mid x_t) \sum_{x_{t-1}} P(x_t \mid x_{t-1}, u_t) \cdot Bel(x_{t-1})$$

(Statisches) Sensormodell **Aktionsmodell**

Algorithmus Bayes-Filter

Eingabe : Aktueller Belief $Bel(x_t)$, sowie ein Paar (u, z).

Ausgabe: Aktualisierter Belief $Bel(x_{t+1})$

```
for alle Zustände x_t do                                    // Update nach Aktion
    \overline{Bel}(x_{t+1}) = Σ_{x_t} P(x_{t+1} | u, x_t) Bel(x_t)
end for
η = 0
for alle Zustände x_{t+1} do                                // Update nach Messung
    Bel(x_{t+1}) = P(z | x_{t+1}) \overline{Bel}(x_{t+1})
    η = η + Bel(x_{t+1})
end for
for alle Zustände x_{t+1} do                                // Normierung
    Bel(x_{t+1}) = η^{-1} Bel(x_{t+1})
end for
return Bel(x_{t+1})
```

$$\overline{Bel}(x_{t+1}) = \sum_{x_t} P(x_{t+1} \mid u, x_t) Bel(x_t)$$

Belief a priori

$$Bel(x_{t+1}) = P(z \mid x_{t+1}) \overline{Bel}(x_{t+1})$$

$$\eta = \eta + Bel(x_{t+1})$$

Belief a posteori

$$Bel(x_{t+1}) = \eta^{-1} Bel(x_{t+1})$$

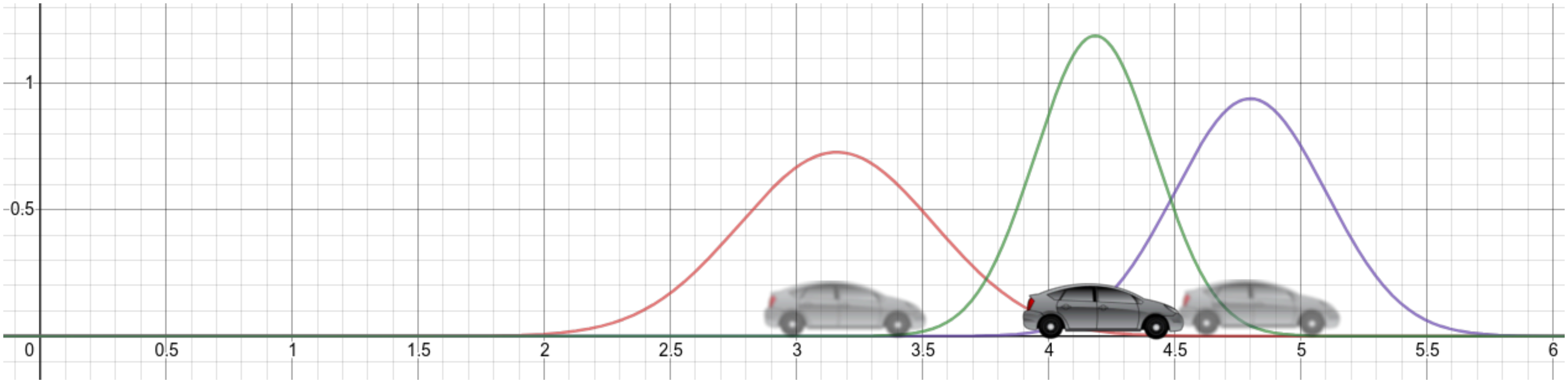
UNIVERSITÄT OSNABRÜCK

Fazit Bayes-Filter

- Die Bayes'sche Regel erlaubt es uns Wahrscheinlichkeiten zu berechnen, die sonst nur schwer zu erheben wären
- Mit Hilfe der Markov-Annahme können rekursive Bayes-Filter benutzt werden, um Evidenzen zum kombinieren
- Bayes-Filter sind **das** Standardwerkzeug für Zustandsschätzungen in dynamischen Systemen
- Bayes-Filter werden in vielen Bereichen verwendet
  - Kalman-Filter (siehe gleich)
  - Partikelfilter (siehe Kapitel 6)
  - Hidden Markov Models
  - Dynamic Bayesian Networks
  - Partial Observable Markov Decision Processes (POMDPs)

Kalman-Filter - Die Motivation

- Einsatz mehrerer Sensoren, deren Messwerte mit Unsicherheiten belegt sind
- Ziel: Zustandsschätzung basierend auf mehreren unabhängigen Messungen, so dass
  - … die Gesamtunsicherheit reduziert wird
  - … glaubwürdige Informationen stärker gewichtet werden als unplausible
  - … die abgeleitete Lösung statistisch “besser” ist als die Einzellösungen



UNIVERSITÄT OSNABRÜCK

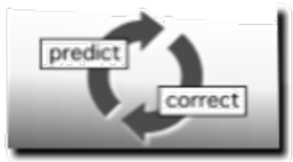
Kalman-Filter als spezieller Bayes-Filter

$$\mathbf{P}(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \boldsymbol{\Sigma}_\mathbf{u})(\mathbf{x}_{t+1})$$

$$\mathbf{P}(\mathbf{z}_t \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{H}\mathbf{x}_t, \boldsymbol{\Sigma}_\mathbf{z})(\mathbf{z}_t)$$

n	Dimension Zustand
m	Dimension Aktion
l	Dimension Messung

$\mathbb{1}$	Einheitsmatrix $(n \times n)$-dimensional
$\mathbf{A}$	Transitionsmodell $(n \times n)$-dimensional, modelliert spontane Transitionen. Ist gleich $\mathbb{1}$, wenn keine spontanten Transitionen erfolgen
$\mathbf{B}$	Aktionsmodell $(n \times m)$-dimensional, konvertiert $\mathbf{u}$ in den Zustandsraum
$\boldsymbol{\Sigma}_\mathbf{u}$	Kovarianzmatrix für Aktionsmodell $(n \times n)$-dimensional
$\mathbf{H}$	Sensormodell, $(l \times n)$-dimensional. Konvertiert Zustand in den Messraum
$\boldsymbol{\Sigma}_\mathbf{z}$	Kovarianzmatrix für Sensormodell $(l \times l)$-dimensional



Kalman-Filter - Das Vorgehen

1.Prädiktion:

- Sage resultierenden Zustand für Aktion/Ereignis voraus
- Schätze den Fehler nach aktuellem Fehlermodell ab
- Führe die Aktion aus (bzw. unabhängiges Ereignis geschieht)

2.Filterung:

- Miss resultierenden Zustand
- Ggf. mach mehrere unabhängige Messungen (die sich widersprechen dürfen)
- Nimm als aktuellen Zustand den an, der maximal gut zu allen Messungen und dem letzten bekannten Fehlermodell passt
- Aktualisiere Fehlermodell (!)
- Weiter bei 1.

Schätzung von Mittelwert und Varianz

A-priori-Vorhersagen für $t+1$:

$$\underline{\mathbf{x}}_{\mathbf{t+1}} := \mathbf{A}\mu_{\mathbf{t}} + \mathbf{Bu}$$
$$\underline{\mathbf{\Sigma}}_{\mathbf{t+1}} := \mathbf{A}\mathbf{\Sigma}_{\mathbf{t}}\mathbf{A}^{\mathbf{T}} + \mathbf{\Sigma}_{\mathbf{u}}$$

Rote Kurve

← Vorhergesagter Zustand zu $t+1$ nach u

← Vorhergesagte Kovarianz zu $t+1$ nach u

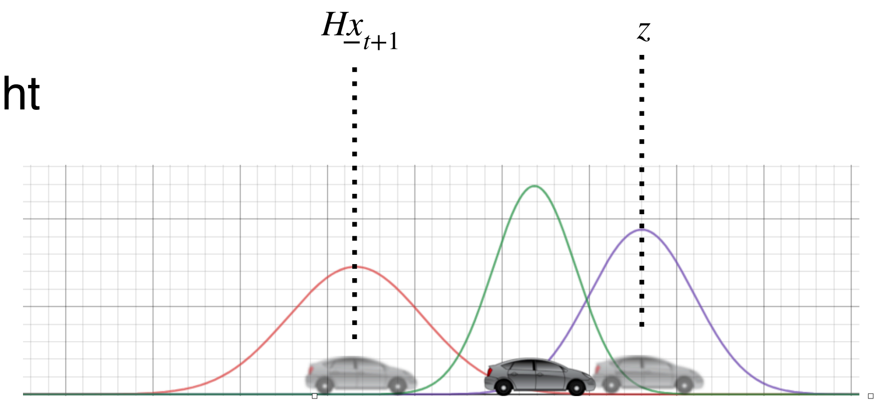
Filterung / Korrektur:

$$\mu_{\mathbf{t+1}} = \underline{\mathbf{x}}_{\mathbf{t+1}} + \mathbf{K}_{\mathbf{t+1}}(\boxed{\mathbf{z}} - \mathbf{H}\underline{\mathbf{x}}_{\mathbf{t+1}})$$
$$\mathbf{\Sigma}_{\mathbf{t+1}} = (\mathbf{1} - \mathbf{K}_{\mathbf{t+1}}\mathbf{H})\underline{\mathbf{\Sigma}}_{\mathbf{t+1}}$$

grüne Kurve

$\mathbf{K}_{\mathbf{t+1}}$ sagt, wie hoch der Vorhersagefehler in neuen Wert eingeht
"Kalman Gewinnmatrix (*Kalman Gain*)":

$$\mathbf{K}_{\mathbf{t+1}} = \underline{\mathbf{\Sigma}}_{\mathbf{t+1}}\mathbf{H}^{\mathbf{T}}(\mathbf{H}\underline{\mathbf{\Sigma}}_{\mathbf{t+1}}\mathbf{H}^{\mathbf{T}} + \mathbf{\Sigma}_{\mathbf{z}})^{-1}$$



Zeitdiskreter Kalmanfilter-Algorithmus

Eingabe : $\boldsymbol{x}_t$ aktueller Zustand, $\boldsymbol{\Sigma}_t$ aktuelles Fehlermodell, sowie ein Evidenz-Paar $(\boldsymbol{u}, \boldsymbol{z})$

Ausgabe: Die aktualisierten $\boldsymbol{x}_{t+1}, \boldsymbol{\Sigma}_{t+1}$

1: *Prädiktion mit Hilfe der Aktion* $\boldsymbol{u}$

2: $\bar{\boldsymbol{x}}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}$

3: $\bar{\boldsymbol{\Sigma}}_{t+1} = \boldsymbol{A}\boldsymbol{\Sigma}_t\boldsymbol{A}^T + \boldsymbol{\Sigma}_u$

4: *Korrektur mit Hilfe der Messung* $\boldsymbol{z}$

5: $\boldsymbol{K}_{t+1} = \bar{\boldsymbol{\Sigma}}_{t+1}\boldsymbol{H}^T\left(\boldsymbol{H}\bar{\boldsymbol{\Sigma}}_{t+1}\boldsymbol{H}^T + \boldsymbol{\Sigma}_z\right)^{-1}$

6: $\boldsymbol{x}_{t+1} = \bar{\boldsymbol{x}}_{t+1} + \boldsymbol{K}_{t+1}(\boldsymbol{z} - \boldsymbol{H}\bar{\boldsymbol{x}}_{t+1})$ // *Zustand* $\boldsymbol{x}_i \widehat{=}$ *Mittelwert* $\boldsymbol{\mu}_i$

7: $\boldsymbol{\Sigma}_{t+1} = (\mathbb{1} - \boldsymbol{K}_{t+1}\boldsymbol{H})\bar{\boldsymbol{\Sigma}}_{t+1}$

8: **return** $\boldsymbol{x}_{t+1}, \boldsymbol{\Sigma}_{t+1}$

Optimalität des Kalman-Filters

- Gauß / Normalverteilung nicht zwingend für Kalman-Filter:
  - Systemrauschen mit Mittelwert 0
  - Fehlerquellen untereinander mit Kovarianz 0
  - Lineare Transitionen
  - Lineare Abhängigkeiten zwischen Zuständen und Sensormessungen

➡ **Unter diesen Bedingungen ist der Kalman-Filter optimal!**
  - Kein anderes Verfahren schätzt die Systemzustände mit geringeren erwarteten Abständen zwischen geschätztem und tatsächlichem Zustand
  - Kalman-Filter nutzt die vorhandenen Informationen optimal aus

Beispiel: Gleichförmige Bewegung + GPS

Erinnerung:

$$\underline{\mathbf{x}}_{\mathbf{t+1}} := \mathbf{A}\mu_{\mathbf{t}} + \mathbf{Bu} \qquad \mu_{\mathbf{t+1}} = \underline{\mathbf{x}}_{\mathbf{t+1}} + \mathbf{K}_{\mathbf{t+1}}(\mathbf{z} - \mathbf{H}\underline{\mathbf{x}}_{\mathbf{t+1}}) \qquad \mathbf{K}_{\mathbf{t+1}} = \underline{\boldsymbol{\Sigma}}_{\mathbf{t+1}}\mathbf{H}^{\mathbf{T}}(\mathbf{H}\underline{\boldsymbol{\Sigma}}_{\mathbf{t+1}}\mathbf{H}^{\mathbf{T}} + \boldsymbol{\Sigma}_{\mathbf{z}})^{-1}$$

$$\underline{\boldsymbol{\Sigma}}_{\mathbf{t+1}} := \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^{\mathbf{T}} + \boldsymbol{\Sigma}_{\mathbf{u}} \qquad \boldsymbol{\Sigma}_{\mathbf{t+1}} = (\mathbf{1} - \mathbf{K}_{\mathbf{t+1}}\mathbf{H})\underline{\boldsymbol{\Sigma}}_{\mathbf{t+1}}$$

Start bei $x_0 = 0$, $\sigma_0 = 0$, $\Sigma_0 = [0]$, sei $u = \Delta x = 3{,}2$

Sei $\sigma_u = (|\Delta x|)/10 = 0{,}32$

➟ $\Sigma_u = [\sigma_u^2] = [0{,}32^2]$

➟ $\underline{x}_1 = 3{,}2$

➟ $\underline{\Sigma}_1 = \Sigma_0 + \Sigma_u = [0{,}32^2]$

Weiter gelten $\mathbf{H} = [\mathbf{1}]$ und $\Sigma_z = [\sigma_z^2] = [0{,}05^2]$

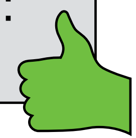
➟ $\mathbf{K_1} = (0{,}32^2)/(0{,}32^2 + 0{,}05^2) = 0{,}97616$

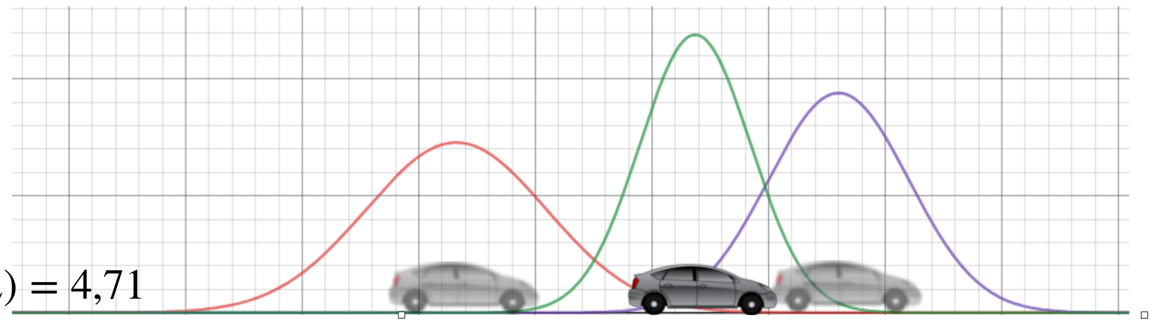
➟ $\boldsymbol{\Sigma}_\mathbf{1} = (\mathbf{1} - \mathbf{K}_\mathbf{1})\underline{\boldsymbol{\Sigma}}_\mathbf{1} = [0{,}00244^2]$

Sei Messwert $z_1 = 4{,}75$

➟ $\mu_1 = \underline{\mathbf{x}}_\mathbf{1} + \mathbf{K}_\mathbf{1}(\mathbf{z}_\mathbf{1} - \mathbf{H}\underline{\mathbf{x}}_\mathbf{1}) = 3{,}2 + 0{,}97616(4{,}75 - 3{,}2) = 4{,}71$

Messung gewinnt, da sie deutlich zuverlässiger ist: $\sigma_z \ll \sigma_u$





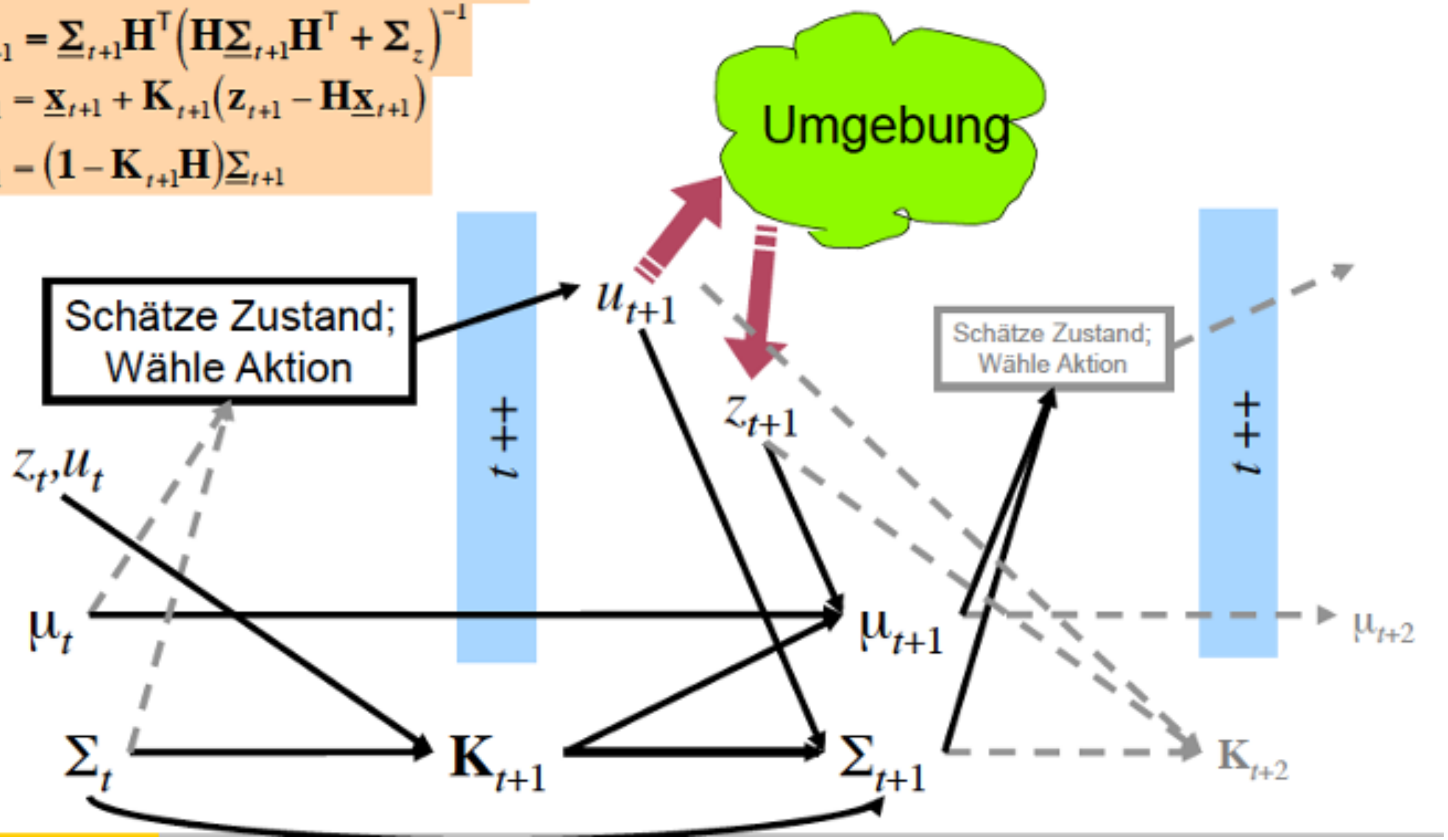
Zusammenfassung Handeln / Messen

$$\underline{\mathbf{x}}_{t+1} := \mathbf{A}\mu_t + \mathbf{B}u;\ \underline{\Sigma}_{t+1} = \mathbf{A}\Sigma_t\mathbf{A}^\mathsf{T} + \Sigma_u$$

$$\mathbf{K}_{t+1} = \underline{\Sigma}_{t+1}\mathbf{H}^\mathsf{T}\left(\mathbf{H}\underline{\Sigma}_{t+1}\mathbf{H}^\mathsf{T} + \Sigma_z\right)^{-1}$$

$$\mu_{t+1} = \underline{\mathbf{x}}_{t+1} + \mathbf{K}_{t+1}\left(\mathbf{z}_{t+1} - \mathbf{H}\underline{\mathbf{x}}_{t+1}\right)$$

$$\Sigma_{t+1} = \left(1 - \mathbf{K}_{t+1}\mathbf{H}\right)\underline{\Sigma}_{t+1}$$



Beispielaufgabe

Für einen hypothetischen mobilen Roboter bestehe der Zustandsraum aus folgenden Komponenten:

- den Koordinaten x, z
- der Orientierung θ
- dem Luftdruck in Koblenz λ
- dem Namen des Papstes π

Unabhängig von den Aktionen des Roboters kann sich der Luftdruck von einem zum nächsten Zustand von selbst mit einer Standardabweichung von 0.001 hPa ändern. Die Roboteraktionen ändern weder den Luftdruck, noch den Namen des Papstes. Der Roboter kann die aktuellen Werte von x, z, θ und λ messen.

(1) Geben Sie die Matrizen $\mathbf{A}, \mathbf{B}$ und $\mathbf{H}$ für die beschriebene Domäne an!

(2) Modellieren Sie die Aktionen *Rotation* und *Translation* mit gegebenen Standardabweichungen!

(3) Beispielrechnung von drei Iterationen für gegebene Initialwerte und Aktionen

Beispielaufgabe - Diskussion

- Besprechung Form und Dimension der erforderlichen Matrizen

Lösung: Sei $x = \begin{pmatrix} x \\ z \\ \theta \\ \lambda \\ \pi \end{pmatrix}$ der Zustandsraum, $u = \begin{pmatrix} \Delta x \\ \Delta z \\ \Delta\theta \end{pmatrix}$ der Aktionsraum und $Z = \begin{pmatrix} x \\ z \\ \theta \\ \lambda \end{pmatrix}$ der Messraum

$$A = \mathbb{1}$$

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Zusammenfassung

Vorhersage:

Sage den Zustand basierend auf Aktions- und Transitionsmodell voraus

Schätze die Kovarianzen

Korrektur:

Berechne den Kalman-Gain

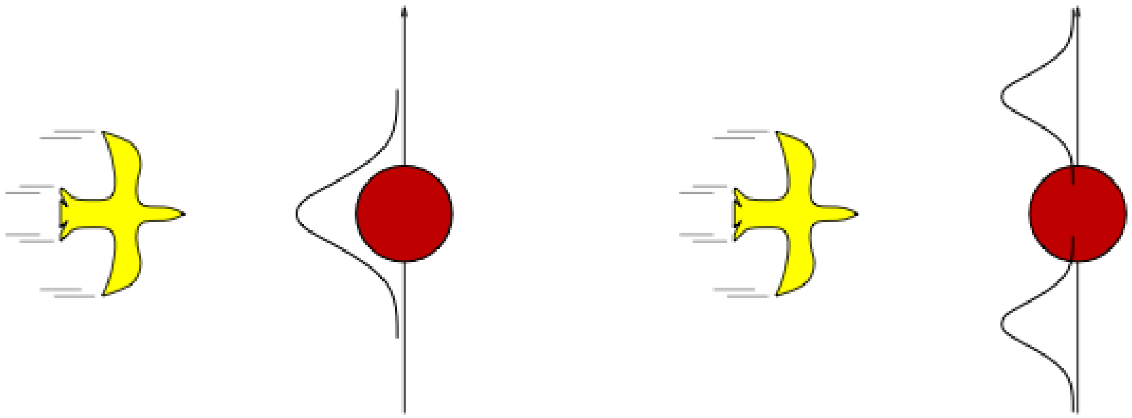
Aktualisiere die Schätzung durch Fusion mit der Messung

Aktualisiere das Fehlermodell (Kovarianzen)

- Berechnung einer Zustandsschätzung, die die Informationen über den Systemzustand und durch Messdaten gewonnen Erkenntnisse optimal ausnutzt
- Unimodales Verfahren, man bekommt immer nur den wahrscheinlichsten Zustand
- Multimodale-Verfahren kommen dann später zur Lokalisierung!

Grenzen des Kalmanfilters

- Kalman-Filter praktisch auch angewandt, wenn Optimalitäts-voraussetzung nicht gegeben (z.B. korrelierte Fehlerquellen)
- Unimodale Verteilungen modellieren manche Aktionen nicht
- ➠ später arbeiten wir oft mit multimodalen Verteilungen (allg. Bayes-F.)



- In der Robotik sind Transitionen meist/oft nicht linear
- ➠ Erweiterter Kalman-Filter (EKF): Approximative Linearisierung eines nichtlinearen Transitions- und/oder Sensormodells (Jakobi-Matrizen d. nichtlinearen Funktionen)

EKF

- Bring nicht-lineare Aktions- und Sensormodelle *f* und *h* in das Kalman-Filter-Verfahren ein

  Ersetze Zustandsvorhersage $\underline{\mathbf{x}}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}$ durch:

  $$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u})$$

  Ersetze Messvorhersage (Matrix $\mathbf{H}$) durch $h(\underline{\mathbf{x}})$, also die Zustandsaktualisierung, durch

  $$\mathbf{x}_{t+1} = \underline{\mathbf{x}}_{t+1} + \mathbf{K}_{t+1}(\mathbf{z}_{t+1} - h(\underline{\mathbf{x}}_{t+1}))$$

➠ Ersetze die Matrizen $\mathbf{A}, \mathbf{B}, \mathbf{H}$ durch zeitabhängige Versionen $\mathbf{F}_t, \mathbf{H}_t$ der partiellen Ableitungen von f nach $\mathbf{u}$ und h nach $\mathbf{x}$ (Jacobi-Matrizen)

Jacobi-Matrizen für F_t, H_t

Sei $F_t^{[i,j]}$ der (i,j)-te Eintrag in $\mathbf{F}_t$; seien $x^{[i]}$, $f^{[i]}$ i-te Einträge im Vektor $\mathbf{x}$ bzw der mehrdimensionalen Funktion f. (Entsprechend für $\mathbf{H}$, h , u)

$$\mathbf{F}_{t+1} = \left(F_{t+1}^{[i,j]}\right)_{i,j} = \frac{\partial f^{[i]}}{\partial x^{[j]}}(\mathbf{x}_t, \mathbf{u}_t) = \begin{pmatrix} \frac{\partial f^{[1]}}{\partial x^{[1]}} & \cdots & \frac{\partial f^{[1]}}{\partial x^{[n]}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f^{[n]}}{\partial x^{[1]}} & \cdots & \frac{\partial f^{[n]}}{\partial x^{[n]}} \end{pmatrix}$$

$$\mathbf{H}_{t+1} = \left(H_{t+1}^{[i,j]}\right)_{i,j} = \frac{\partial h^{[i]}}{\partial x^{[j]}}(\underline{\mathbf{x}}_{t+1})$$

UNIVERSITÄT OSNABRÜCK

Algorithmus EKF

Eingabe : $\boldsymbol{x}_t$ aktueller Zustand, $\boldsymbol{\Sigma}_t$ aktuelles Fehlermodell, sowie ein Evidenz-Paar $(\boldsymbol{u}, \boldsymbol{z})$

Ausgabe: Die aktualisierten $\boldsymbol{x}_{t+1}, \boldsymbol{\Sigma}_{t+1}$

Prädiktion mit Hilfe der Aktion $\boldsymbol{u}$

$\bar{\boldsymbol{x}}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u})$

$\bar{\boldsymbol{\Sigma}}_{t+1} = \boldsymbol{F}_{t+1}\boldsymbol{\Sigma}_t\boldsymbol{F}_{t+1}^T + \boldsymbol{\Sigma}_u$

Korrektur mit Hilfe der Messung $\boldsymbol{z}$

$\boldsymbol{K}_{t+1} = \bar{\boldsymbol{\Sigma}}_{t+1}\boldsymbol{H}_{t+1}^T\left(\boldsymbol{H}_{t+1}\bar{\boldsymbol{\Sigma}}_{t+1}\boldsymbol{H}_{t+1}^T + \boldsymbol{\Sigma}_z\right)^{-1}$

$\boldsymbol{x}_{t+1} = \bar{\boldsymbol{x}}_{t+1} + \boldsymbol{K}_{t+1}\left(\boldsymbol{z} - h(\bar{\boldsymbol{x}}_{t+1})\right)$ *// Zustand* $\boldsymbol{x}_i \widehat{=}$ *Mittelwert* $\boldsymbol{\mu}_i$

$\boldsymbol{\Sigma}_{t+1} = (\mathbb{1} - \boldsymbol{K}_{t+1}\boldsymbol{H}_{t+1})\bar{\boldsymbol{\Sigma}}_{t+1}$

return $\boldsymbol{x}_{t+1}, \boldsymbol{\Sigma}_{t+1}$ *// Unverändert, falls* $\boldsymbol{e} = \boldsymbol{u}$

Wo sind wir jetzt?