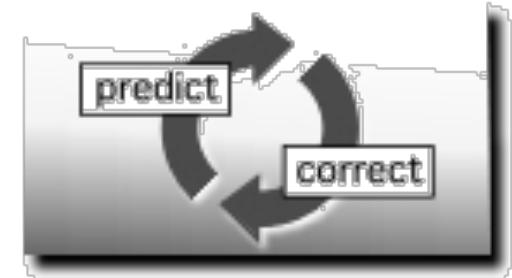


Kalman-Filter: Die Idee



Ausgangspunkte

- n -dimensionale Zustandsschätzung
z.B. Schätzung der Roboterpose in x, z, θ
- Lineares Fehlermodell für Aktionen
z.B. Schätzung des Posefehlers bei Bewegung;

www.cs.unc.edu/~welch/kalman/

Russell/Norvig, Kap.15.4

Thrun/Burgard/Fox Kap.3.2

Ziel

- Gib zu jedem Zeitpunkt aktuelle Zustandsschätzung
z.B. Poseschätzung in nächstem Zustand (vor/nach Aktion)
- Integriere Sensormessungen in Zustandsschätzung
z.B. Aktualisiere Poseschätzung nach Aktion+Gyro-Wert
- **Aktualisiere Fehlermodell optimal**
(bestmögliche Übereinstimmung mit bisherigen Aktionen+Messungen)

Kalman-Filter: Das Vorgehen



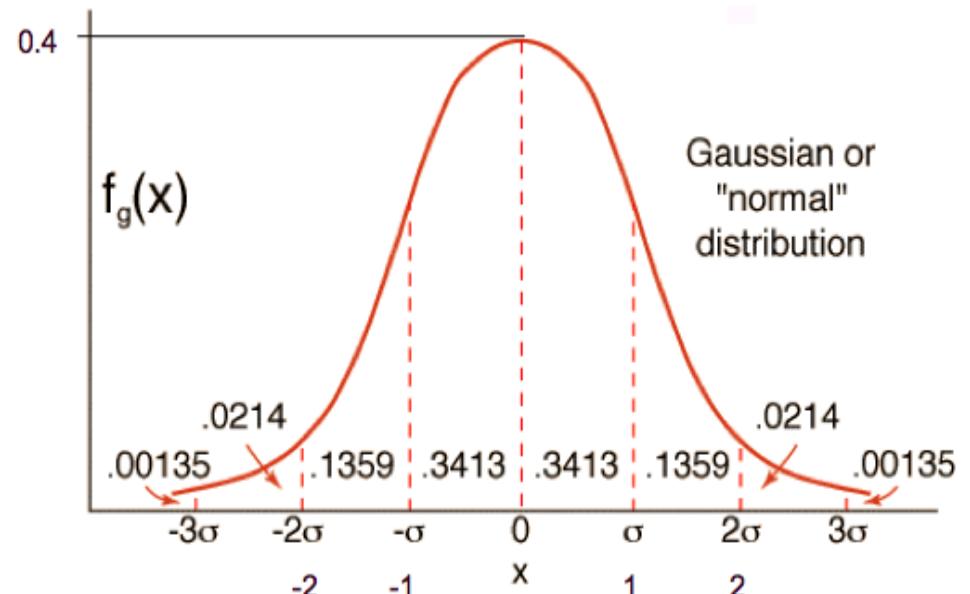
- 1. Prädiktion:** Sage resultierenden Zustand für Aktion/Transition voraus; schätze den Fehler nach aktuellem Fehlermodell ab; führe die Aktion aus (bzw. unabhängige Transition läuft)
- 2. Filterung:** Miss resultierenden Zustand; ggf. mach mehrere unabhängige Messungen (die sich widersprechen dürfen); nimm als aktuellen Zustand den an, der maximal gut zu allen Messungen und dem letzten Fehlermodell passt; aktualisiere Fehlermodell(!); weiter bei 1.

Beschränkung 1: Fehlermodell ist Normalverteilung!
Beschränkung 2: Zustandsverteilung ist Normalverteilung!
Ein Kalman-Filter ist ein spezieller Bayes-Filter

Beispiel, eindimensional (1/3)

- Wert einer Variablen x über die Zeit t bei *random walk*: Bewegt sich in der Nähe der alten Position (Zufall modelliert durch Standardabweichung, s.u.) Das ist ein Beispiel für *Tracking*: Nachfolgezustand entsteht „von allein“, wird nur beobachtet!

- zu $t=0$ gilt: $\mu_0=0.0$; $\sigma_0=1.0$ (Standard-Normalverteilung)
- Transitionsfehler $\sigma_u=2.0$ (erhöht St.-Abw. !)
- Sensorfehler $\sigma_z=1.0$



Beispiel (2/3)

Ausgangszustand, $t=0$: x_0 ist standard-normalverteilt

(α Normalisierungsfaktor)

$$P(x_0) = \mathcal{N}(\mu_0, \sigma_0^2)(x_0) = \alpha e^{-\frac{1}{2}\left(\frac{(x_0 - \mu_0)^2}{\sigma_0^2}\right)}$$

Transitionsmodell für *random walk*:

„Tu nichts“, doch erhöhe St.-Abw. um Transitionsfehler $\sigma_u=2.0$:

$$P(x_{t+1} | x_t) = \mathcal{N}(x_t, \sigma_{t+1}^2(\sigma_t, \sigma_u))(x_{t+1}) = \alpha e^{-\frac{1}{2}\left(\frac{(x_{t+1} - x_t)^2}{\sigma_{t+1}^2}\right)}$$

Sensormodell:

Reduziere St.-A. auf $\sigma_z=1.0$ um „Bereich zwischen z_t und x_t “ bei Messung von z_t :

$$P(z_t | x_t) = \mathcal{N}(x_t, \sigma_z^2)(z_t) = \alpha e^{-\frac{1}{2}\left(\frac{(z_t - x_t)^2}{\sigma_z^2}\right)}$$

Beispiel (3/3)

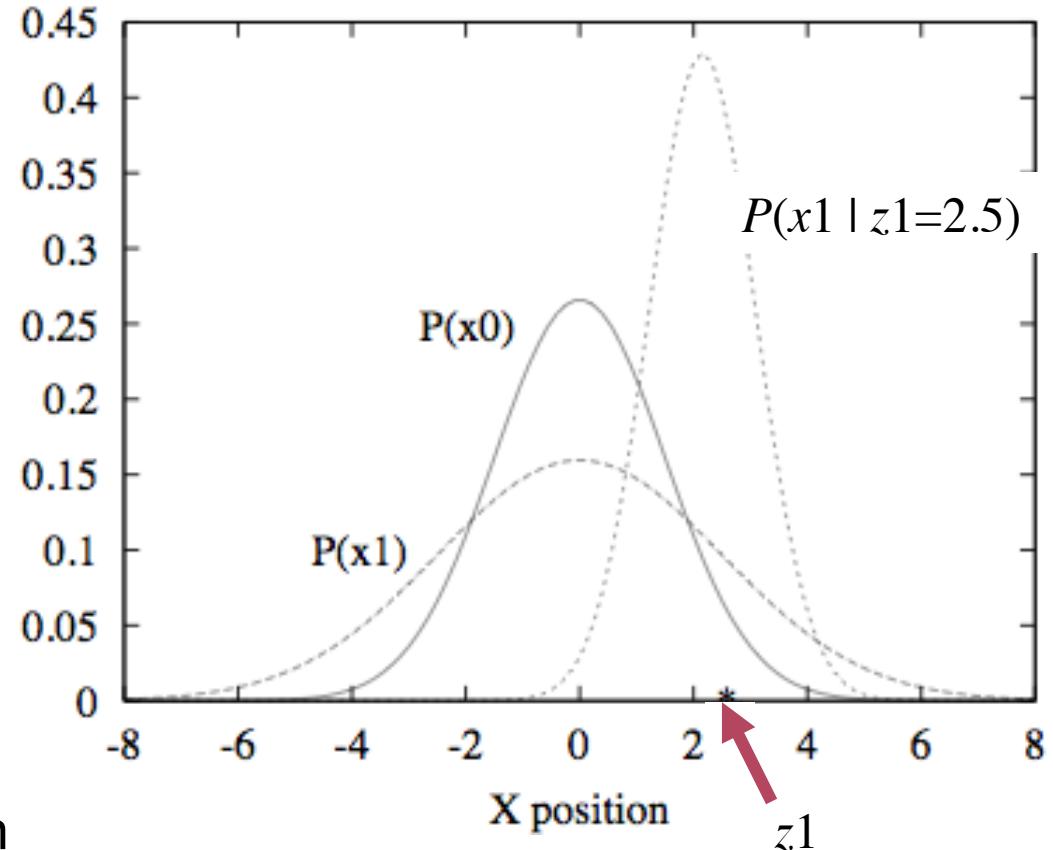
$$\begin{aligned} \mathbf{P}(x_1) &= \int_{-\infty}^{\infty} \mathbf{P}(x_1 | x_0) \mathbf{P}(x_0) dx_0 \\ &= \alpha e^{-\frac{1}{2} \left(\frac{(x_1 - \mu_0)^2}{\sigma_0^2 + \sigma_u^2} \right)} \end{aligned}$$

$$\begin{aligned} \mathbf{P}(x_1 | z_1) &= \alpha \mathbf{P}(z_1 | x_1) \mathbf{P}(x_1) \\ &= \alpha e^{-\frac{1}{2} \left(\frac{(x_1 - f(z_1, \mu_0))^2}{c^2} \right)} \end{aligned}$$

(f neuer Mittelwert: lineare Funktion in σ -Termen;
 c neue St.-A.: Konstante in σ -Termen)

Alle resultierenden Verteilungen sind Gaußverteilungen!

Herleitungen s. Russell/Norvig



- Mittelwert nach Messung i.allg. nicht bei gemessenem Wert!
- der μ_{t+1} ist gewichteter Mittelwert aus z_{t+1} und vorigem Mittelwert

Gauß rein – Gauß raus

Gaußverteilungen sind invariant gegen Faltung mit Gaußverteilungen!

Voraussetzung: Alle initialen Schätzungen & Fehlermodelle sind normalverteilt

z.B. Roboterpose initial 3-Vektor μ_0 (Mittelwert), 3x3 Kovarianzmatrix Σ_0 ; Σ_u und Σ_z Kovarianzmatrizen für Aktions- und Sensormodelle

Dann ergibt sich: (Evidenz e ist Transition u oder Messung z)

- **Prädiktion:** Sind Zustandsschätzung $\mathbf{P}(\mathbf{x}_t \mid \mathbf{e}_{1:t})$ und Aktions/Trans.-Modell $\mathbf{P}(\mathbf{x}_{t+1} \mid \mathbf{u}_t, \mathbf{x}_t)$ normalverteilt, so auch

$$\mathbf{P}(\mathbf{x}_{t+1} \mid \mathbf{e}_{1:t}) = \int_{\mathbf{x}_t} \mathbf{P}(\mathbf{x}_{t+1} \mid \mathbf{u}_t, \mathbf{x}_t) \mathbf{P}(\mathbf{x}_t \mid \mathbf{e}_{1:t}) d\mathbf{x}_t$$

- **Filterung:** Sind Prädiktion $\mathbf{P}(\mathbf{x}_{t+1} \mid \mathbf{e}_{1:t})$ und Sensormodell $\mathbf{P}(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1})$ normalverteilt, so auch

$$\mathbf{P}(\mathbf{x}_{t+1} \mid \mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1}) \mathbf{P}(\mathbf{x}_{t+1} \mid \mathbf{e}_{1:t})$$

- Also ist $\mathbf{P}(\mathbf{x}_t \mid \mathbf{e}_{1:t})$ für alle t eine Normalverteilung!

n-dimensionale Kalman-Filter

- ... sind eine spezielle Form von Bayes-Filter
- ... approximieren $\text{Bel}(\mathbf{x}_t)$ durch $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)(\mathbf{x})$
- ... aktualisieren $\boldsymbol{\mu}_t$, $\boldsymbol{\Sigma}_t$ je Zeitschritt durch Gauß-Faltung
- ... können folglich nur **unimodale** Verteilungen erzeugen!

Erinnerung: Mehrdimensionale Normalverteilung

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})(\mathbf{x}) = \alpha e^{-\frac{1}{2}((\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}))}$$

$\mathbf{x}, \boldsymbol{\mu}$ n -dimensional, $\boldsymbol{\mu}$ Mittelwert; $\boldsymbol{\Sigma}$ ($n \times n$)-dimensional Kovarianzmatrix

Aktions/Transitions- und Sensormodell

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

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

n-dim. Kalman-Filter: Glossar ☺

$$P(\mathbf{x}_{t+1} | \mathbf{u}_t, \mathbf{x}_t) = \mathcal{N}(\mathbf{Ax}_t + \mathbf{Bu}_t, \Sigma_u)(\mathbf{x}_{t+1})$$

$$P(\mathbf{z}_t | \mathbf{x}_t) = \mathcal{N}(\mathbf{Hx}_t, \Sigma_z)(\mathbf{z}_t)$$

n	Dimension Zustand
m	Dimension Aktion
l	Dimension Messung

1 Einheitsmatrix, ($n \times n$)-dim

A Transitionsmodell, ($n \times n$)-dim, modelliert spontane Transitionen. Ist gleich **1**, wenn keine spontanen Tr.

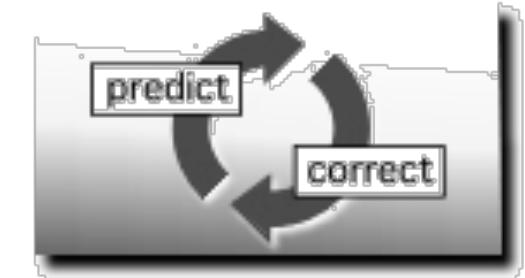
B Aktionsmodell, ($n \times m$)-dim. **Bu** konvertiert **u** in Zust-Raum

Σ_u Kovarianzmatrix für Aktionsmodell, ($n \times n$)-dim

H Sensormodell, ($l \times n$)-dim. Konvertiert Zust. in Mess-Raum

Σ_z Kovarianzmatrix für Sensormodell, ($l \times l$)-dim

Multivariate M'wert und Varianz



Sei $\underline{x}_{t+1} := A\mu_t + Bu$; $\underline{\Sigma}_{t+1} = A\Sigma_t A^\top + \Sigma_u$
(*a-priori*-Vorhersagen) Es ergeben sich:

$$\mu_{t+1} = \underline{x}_{t+1} + K_{t+1}(z - H\underline{x}_{t+1})$$

$$\Sigma_{t+1} = (I - K_{t+1}H)\underline{\Sigma}_{t+1}$$

„Believe it or not, these equations make some intuitive sense.“

Russell/Norvig

\underline{x}_{t+1} vorhergesagter Mittelw./Zust. zu ($t+1$) nach u

$H\underline{x}_{t+1}$ vorhergesagte Sensormessung in ($t+1$)

$z - H\underline{x}_{t+1}$ Abweichung in der Sensorvorhersage nach Messung z

Σ_{t+1} vorhergesagte Kovarianz für ($t+1$) nach u

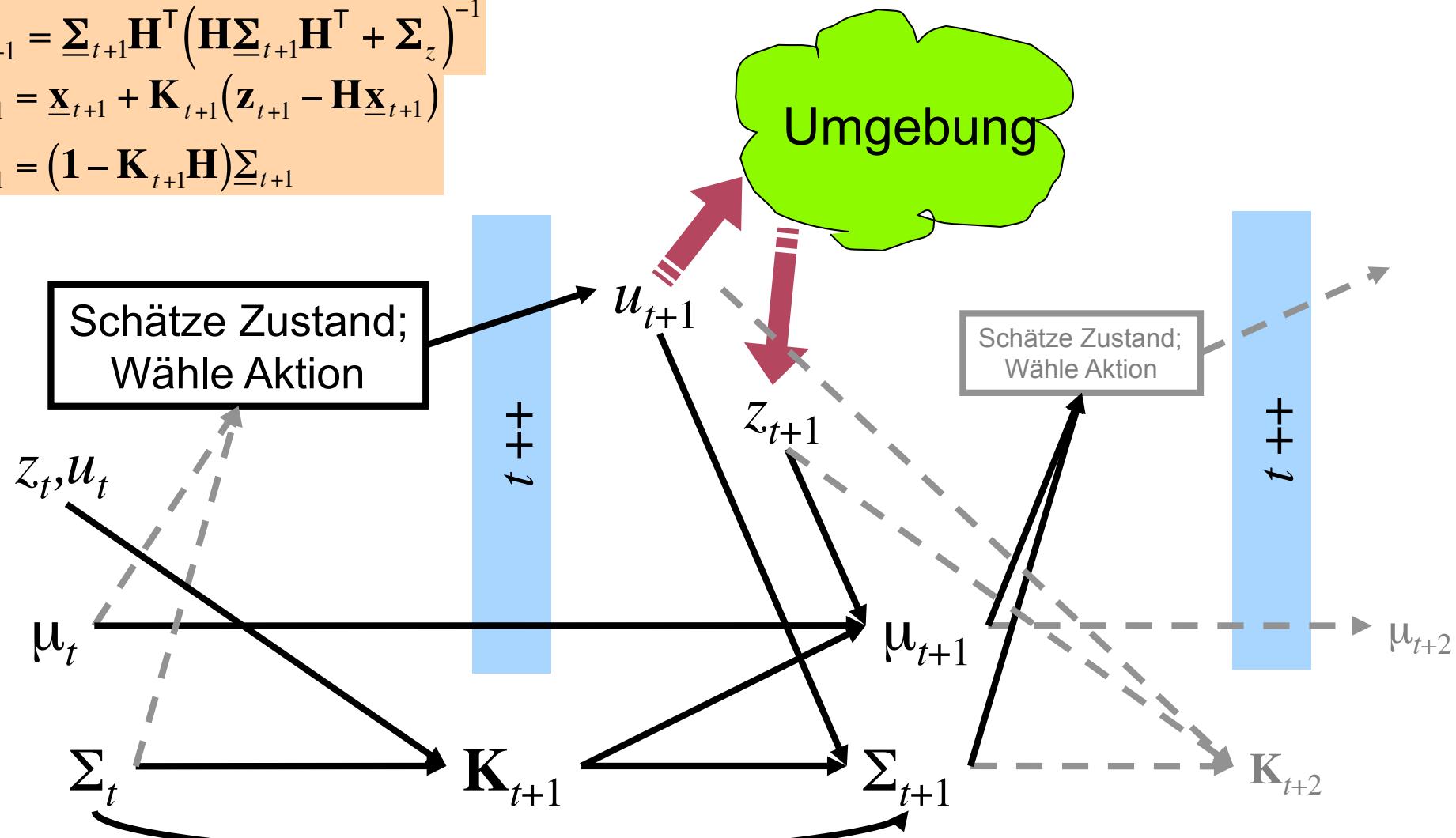
K_{t+1} sagt, wie hoch Vorhersagefehler in neuen Wert eingeht

Kalman-Gewinnmatrix (Kalman Gain) (Dim. $n \times l$)

$$K_{t+1} = \underline{\Sigma}_{t+1} H^\top \left(H \underline{\Sigma}_{t+1} H^\top + \Sigma_z \right)^{-1}$$

Zusammenfassung: Abhängigkeiten

$$\begin{aligned}\underline{\mathbf{x}}_{t+1} &:= \mathbf{A}\mu_t + \mathbf{B}u; \quad \underline{\Sigma}_{t+1} = \mathbf{A}\Sigma_t\mathbf{A}^\top + \Sigma_u \\ \mathbf{K}_{t+1} &= \underline{\Sigma}_{t+1}\mathbf{H}^\top \left(\mathbf{H}\underline{\Sigma}_{t+1}\mathbf{H}^\top + \Sigma_z \right)^{-1} \\ \mu_{t+1} &= \underline{\mathbf{x}}_{t+1} + \mathbf{K}_{t+1}(\mathbf{z}_{t+1} - \mathbf{H}\underline{\mathbf{x}}_{t+1}) \\ \Sigma_{t+1} &= (1 - \mathbf{K}_{t+1}\mathbf{H})\underline{\Sigma}_{t+1}\end{aligned}$$



Rechenbeispiel *random walk* (1/5)

Zeige: Kalman-Filter ergibt dasselbe wie direkte Rechnung (Folie 4-6)!

Aktionsmodell:

$$\mathbf{B} = [1]$$

Transitions-Kov.-Matrix:

$$\Sigma_u = [\sigma_u^2] = [4]$$

Sensormodell:

$$\mathbf{H} = [1]$$

Sensor-Kovarianzmatrix:

$$\Sigma_z = [\sigma_z^2] = [1]$$

Setze 1-dim.
Zustand $x_t = \mu_t$!

Multiv. Normalverteilung:

$$N(\mu, \Sigma)(\mathbf{x}) = \alpha e^{-\frac{1}{2}((\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu))}$$

Sensormodell $P(z_t | x_t) = N(\mathbf{H}x_t, \Sigma_z)(z_t)$

$$= \eta e^{-\frac{1}{2}((z_t - \mathbf{H}x_t)^T \Sigma_z^{-1} (z_t - \mathbf{H}x_t))}$$

$$= \eta e^{-\frac{1}{2}\left(\frac{(z_t - x_t)^2}{\sigma_z^2}\right)}$$

vgl. Sensormodell
Folie 4!

$$\begin{aligned}\mathbf{B} &= [1] \\ \Sigma_u &= [\sigma_u^2] = [4] \\ \mathbf{H} &= [1] \\ \Sigma_z &= [\sigma_z^2] = [1]\end{aligned}$$

Rechenbeispiel *random walk* (2/5)

$$\mathcal{N}(\mu, \Sigma)(\mathbf{x}) = \alpha e^{-\frac{1}{2}((\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu))}$$

Transitionsmodell

$$P(x_{t+1} | u_t, x_t) = N(\mathbf{B}x_t, \Sigma_u)(x_{t+1})$$

$$= \eta e^{-\frac{1}{2}((x_{t+1} - \mathbf{B}x_t)^\top \Sigma_u^{-1}(x_{t+1} - \mathbf{B}x_t))}$$

$$= \eta e^{-\frac{1}{2} \left((x_{t+1} - x_t) \frac{1}{\sigma_u^2} \cdot (x_{t+1} - x_t) \right)}$$

$$= \eta e^{-\frac{1}{2} \left(\frac{(x_{t+1} - x_t)^2}{\sigma_u^2} \right)}$$

vgl. Transitionsmodell
Folie 4!

$$\begin{aligned}\mathbf{B} &= [1] \\ \boldsymbol{\Sigma}_u &= [\sigma_u^2] = [4] \\ \mathbf{H} &= [1] \\ \boldsymbol{\Sigma}_z &= [\sigma_z^2] = [1]\end{aligned}$$

Rechenbeispiel *random walk* (3/5)

Kalman- Gewinnmatrix

$$\begin{aligned}\mathbf{K}_{t+1} &= (\mathbf{A}\boldsymbol{\Sigma}_t\mathbf{A}^\top + \boldsymbol{\Sigma}_u)\mathbf{H}^\top(\mathbf{H}(\mathbf{A}\boldsymbol{\Sigma}_t\mathbf{A}^\top + \boldsymbol{\Sigma}_u)\mathbf{H}^\top + \boldsymbol{\Sigma}_z)^{-1} \\ &= ([\sigma_t^2] + [\sigma_u^2])[1]([\sigma_t^2 + \sigma_u^2] + [\sigma_z^2])^{-1} \\ &= \left[\frac{\sigma_t^2 + \sigma_u^2}{\sigma_t^2 + \sigma_u^2 + \sigma_z^2} \right]\end{aligned}$$

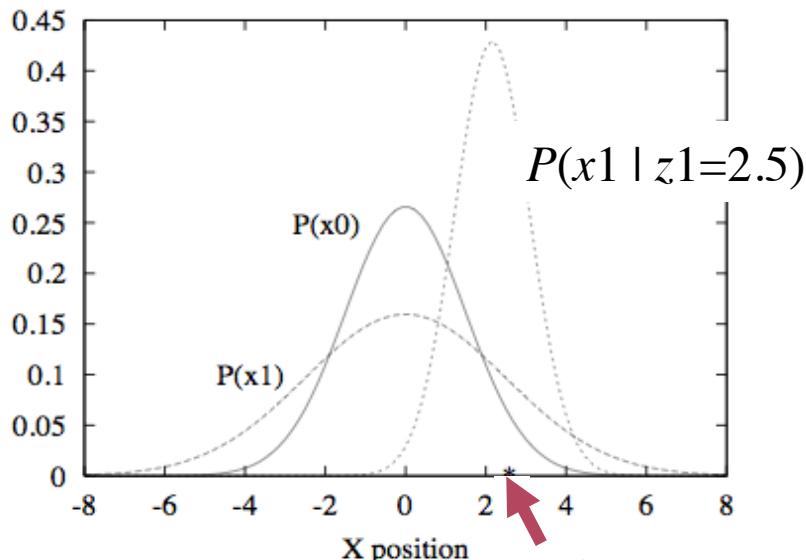
$$\begin{aligned}\mathbf{B} &= [1] \\ \Sigma_u &= [\sigma_u^2] = [4] \\ \mathbf{H} &= [1] \\ \Sigma_z &= [\sigma_z^2] = [1]\end{aligned}$$

Rechenbeispiel *random walk* (4/5)

$$\mathbf{K}_{t+1} = \frac{\sigma_t^2 + \sigma_u^2}{\sigma_t^2 + \sigma_u^2 + \sigma_z^2}$$

$$\mu_{t+1} = (\mathbf{A}\mu_t + \mathbf{B}u) + \mathbf{K}_{t+1}(\mathbf{z} - \mathbf{H}(\mathbf{A}\mu_t + \mathbf{B}u))$$

$$= \mu_t + \frac{(\sigma_t^2 + \sigma_u^2)(\mathbf{z} - \mu_t)}{\sigma_t^2 + \sigma_u^2 + \sigma_z^2} = \frac{\mathbf{z}(\sigma_t^2 + \sigma_u^2) + \sigma_z^2 \mu_t}{\sigma_t^2 + \sigma_u^2 + \sigma_z^2}$$



→ $f(z_1, \mu_0)$, Folie 5!!

im Rechenbeispiel:
 $\mu_1 = 12,5/6 \approx 2,1$

$$\begin{aligned}\mathbf{B} &= [1] \\ \boldsymbol{\Sigma}_u &= [\sigma_u^2] = [4] \\ \mathbf{H} &= [1] \\ \boldsymbol{\Sigma}_z &= [\sigma_z^2] = [1]\end{aligned}$$

Rechenbeispiel *random walk* (5/5)

$$\mathbf{K}_{t+1} = \frac{\sigma_t^2 + \sigma_u^2}{\sigma_t^2 + \sigma_u^2 + \sigma_z^2}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{t+1} &= \left[\sigma_{t+1}^2 \right] = (1 - \mathbf{K}_{t+1} \mathbf{H}) (\mathbf{A} \boldsymbol{\Sigma}_t \mathbf{A}^\top + \boldsymbol{\Sigma}_u) \\ &= \left(1 - \frac{\sigma_t^2 + \sigma_u^2}{\sigma_t^2 + \sigma_u^2 + \sigma_z^2} \right) (\sigma_t^2 + \sigma_u^2) = \frac{\sigma_z^2 (\sigma_t^2 + \sigma_u^2)}{\sigma_t^2 + \sigma_u^2 + \sigma_z^2}\end{aligned}$$

↳ c^2 , Folie 6!

Algorithmus Kalman-Filter

Eingabe : x_t aktueller Zustand, Σ_t aktuelles Fehlermodell, sowie ein Evidenz-Paar (u, z)

Ausgabe: Die aktualisierten x_{t+1}, Σ_{t+1}

1: *Prädiktion mit Hilfe der Aktion u*

$$2: \bar{x}_{t+1} = Ax_t + Bu$$

$$3: \bar{\Sigma}_{t+1} = A\Sigma_t A^T + \Sigma_u$$

4: *Korrektur mit Hilfe der Messung z*

$$5: K_{t+1} = \bar{\Sigma}_{t+1} H^T (H \bar{\Sigma}_{t+1} H^T + \Sigma_z)^{-1}$$

$$6: x_{t+1} = \bar{x}_{t+1} + K_{t+1} (z - H \bar{x}_{t+1}) \quad // \text{ Zustand } x_i \hat{=} \text{Mittelwert } \mu_i$$

$$7: \Sigma_{t+1} = (I - K_{t+1} H) \bar{\Sigma}_{t+1}$$

8: **return** x_{t+1}, Σ_{t+1}

Optimalität des Kalman-Filters

Gauß/Normalverteilung nicht zwingend für Kalman-Filter:

- Systemrauschen mit Mittelwert 0
- Fehlerquellen untereinander mit Kovarianz 0 (z.B. unabhängig)
- Lineare Transitionen (= durch Matrixmultiplikationen beschreibbar)
- Lineare Abhängigkeiten zwischen Zuständen und Sensormessungen

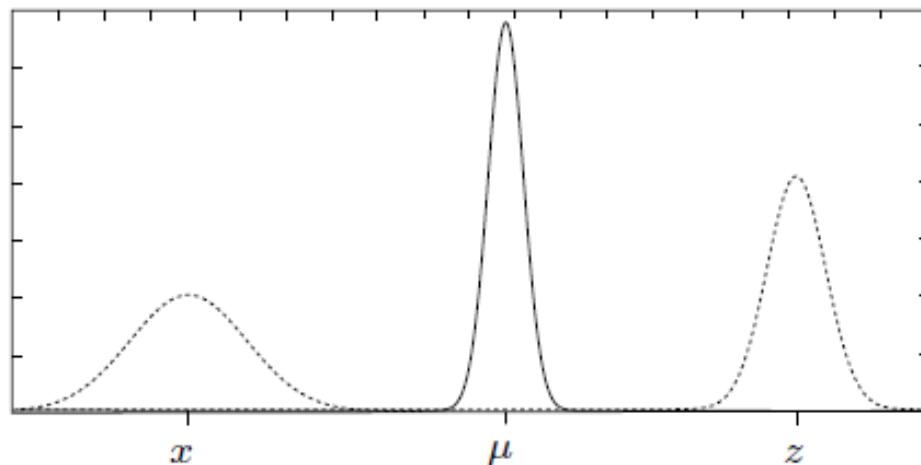
Satz: Unter diesen Voraussetzungen ist Kalman-Filter optimal:

Kein anderes Verfahren schätzt die Systemzustände mit geringeren erwarteten Abweichungen zwischen geschätztem und tatsächlichem Zustand.

D.h.: Kalman-Filter nutzt die vorhandenen Informationen optimal aus.

Grenzen des Kalman-Filters

- Kalman-Filter praktisch auch anwendbar, wenn Optimalitätsvoraussetzung nicht gegeben (z.B. korrelierte Fehlerquellen)
- Unimodale Verteilungen modellieren manche Aktionen nicht
→ später behandeln wir multimodale Verteilungen (anders)



- 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)

Erweiterter Kalman-Filter (EKF)

Bring nichtlineare Aktions- und Sensormodelle f und h in das Kalman-Filter-Verfahren ein:

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

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

- Ersetze Messvorhersage (Matrix \mathbf{H}) durch $h(\underline{\mathbf{x}})$, also die Zustandsaktualisierung $\mathbf{x}_{t+1} := \underline{\mathbf{x}}_{t+1} + \mathbf{K}_{t+1}(z_{t+1} - \mathbf{H}\underline{\mathbf{x}}_{t+1})$ durch

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

- => Ersetze die Matrizen \mathbf{A} , \mathbf{B} , \mathbf{H} durch zeitabhängige Versionen \mathbf{F}_t , \mathbf{H}_t : Jacobi-Matrizen der partiellen Ableitungen von f nach u und h nach \mathbf{x} folgendermaßen [\mathbf{A} sei 1]: ...

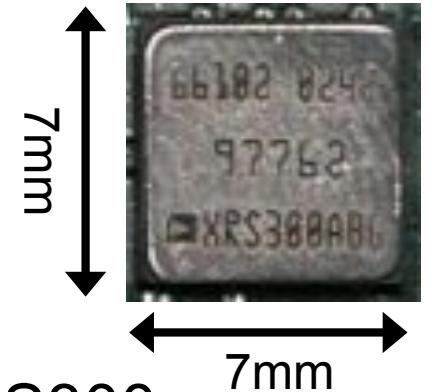
Jacobi-Matrizen für \mathbf{B}_t , \mathbf{H}_t

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 .
(Entspr. für \mathbf{H}, h, u)

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

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

Gyro-Fehler quantitativ

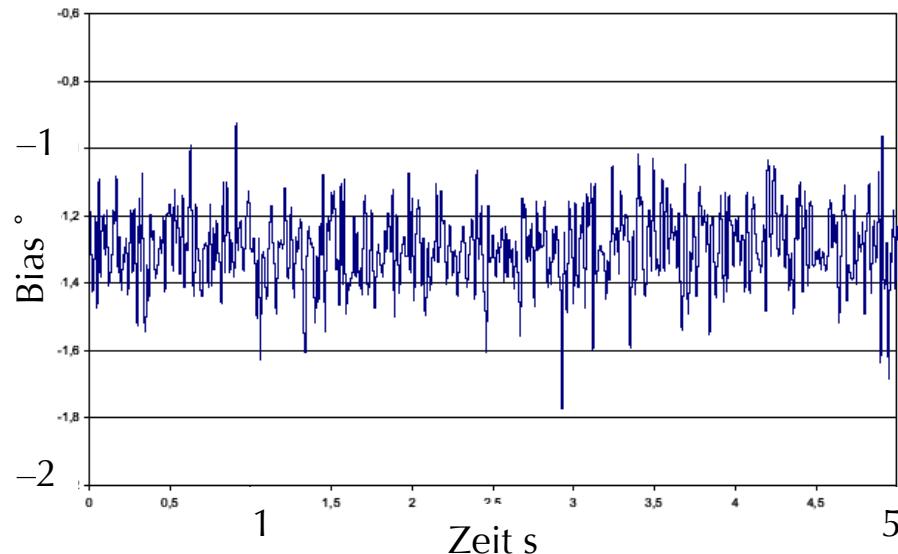


Vorteil beim Kalman-Filter:

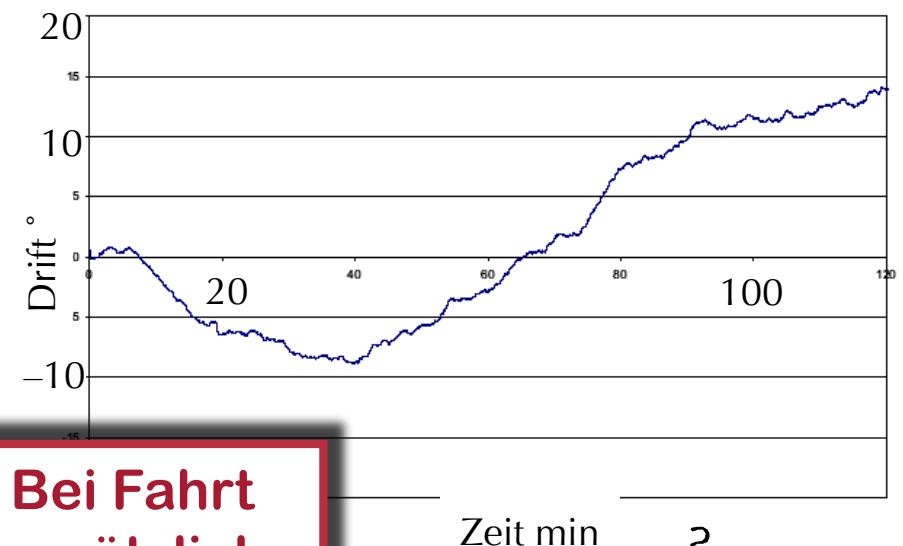
Fehlerkorrektur dynamisch (Gewinnmatrix)

Warum will man das?: Bias-Rauschen des ADXRS300

(Messwerte bei still stehendem Sensor)



Lokal: Gauß-Rauschen
 $(\mu \approx -1.3^\circ, \sigma \approx 0.1^\circ)$



Global:



Gyrodometrie durch Kalman-Filter

... eigentlich komplett Pose (3D) und Gyro 2D (Messwert, Bias) in **EKF** (EKF: Nichtlinearität wg. Gyro-Integration)
(s. Diplomarbeit Solda, bzw. [Solda & al. IAV-2004])

Hier, vereinfacht:

- Eindimensional: Beachte lediglich Orientierung
- Fusion von Odometrie (θ -Schätzung aufgrund erfolgter Radumdrehung) mit Gyro-Messwert
- Bezug auf feste Zeitintervalle Δt (z.B. 1s)

Entspricht strukturell Rechenbeispiel im vorigen Abschnitt!
(random walk, F.186–190)

Transitions- und Sensormodell, 1D

Orientierungsschätzung Odometrie (vgl. *random walk* Folie 186):

- Betrachtung in festen Intervallen von 1 s
- „Aktion“ $u = \Delta\theta = (v_L - v_R)/b$: Drehung innerhalb 1s
- Transitionsmodell: $\mathbf{A} = \mathbf{B} = \mathbf{1}$,
- Varianz wäre empirisch zu ermitteln, hier „geraten“:
Transitions-Kovarianzmatrix: $\Sigma_u = [\sigma_u^2]$, wobei $\sigma_u = (|\Delta\theta| + c)/10$
für festen Transitionsfehleranteil (Drehwinkelrauschen) c

Orientierungsschätzung Gyro:

- Empirisch ermittelt (s. Folie 203):
Feste Messfehler $-1,3^\circ$ behandle außerhalb (konstanter offset),
Standardabweichung $\sigma_z = 0,1^\circ = 0,00175 \text{ rad}$
- Sensormodell: $\mathbf{H} = [1]$
- Sensor-Kovarianzmatrix: $\Sigma_z = [\sigma_z^2] = [0,00175^2]$

Rechenbeispiel 1D-Gyrodometrie

Erinnerung: $\underline{x}_{t+1} := \mathbf{A}\mu_t + \mathbf{B}u; \underline{\Sigma}_{t+1} = \mathbf{A}\Sigma_t\mathbf{A}^\top + \Sigma_u \quad \mu_{t+1} = \underline{x}_{t+1} + \mathbf{K}_{t+1}(\mathbf{z} - \mathbf{H}\underline{x}_{t+1})$

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

Start bei (0,0,0), $\sigma_0 = 0$, $\Sigma_0 = [0]$, sei $u = \Delta\theta = 0,2$

Transitionsfehler (s. vorige Folie) $c := 0,1$, also $\sigma_u = (|\Delta\theta| + c)/10 = 0,03$

→ $\Sigma_u = [\sigma_u^2] = [0,03^2]$,

→ $\underline{x}_1 = 0,2$, $\underline{\Sigma}_1 = \Sigma_0 + \Sigma_u = [0,03^2]$

Weiter gelten $\mathbf{H} = [1]$ und $\Sigma_z = [\sigma_z^2] = [0,00175^2]$ (s.o.), also

→ $\mathbf{K}_1 = (0,03^2) / (0,03^2 + 0,00175^2) = 0,99966$

→ $\Sigma_1 = (1 - \mathbf{K}_1)\underline{\Sigma}_1 = [0,0005532^2]$

Sei Messwert $z_1 = 0,19$

→ $\mu_1 = \underline{x}_1 + \mathbf{K}_1(z_1 - \mathbf{H}\underline{x}_1) = 0,2 + 0,99966(0,19 - 0,2) = 0,1900034$

(Messung „gewinnt“, da sie deutlich zuverlässiger ist: $\sigma_z \ll \sigma_u$)

„Echte“ Ergebnisse mit EKF-Gyrodometrie

- Positionsschätzung mittels Odometrie
- Positionsschätzung mittels Fusion von Odometrie und Gyroskop



[Solda 2003]

Why do I need...

Robotics beta

Questions Tags Users Badges Unanswered Ask Question

got a question about the site itself? [meta.robots](#) is the place to talk about things like what questions are appropriate, what tags we should use, etc.

Why do I need a Kalman filter?



I am designing an unmanned aerial vehicle, which will include several types of sensors: 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer, horizon sensor, GPS and downward facing ultrasound.

6



A friend of mine told me that I will need to put all of this sensor data through a Kalman filter, but I don't understand why. Why can't I just put this straight into my microcontroller. How does the Kalman filter filter the data?



[kalman-filter](#)

[uav](#)

[share](#) [improve this question](#)

asked yesterday



Rocketmagnet

1,028 ● 12

20% accept rate

[feedback](#)

2 Answers

active

oldest

[votes](#)



Sensor data is noisy. If you do not filter it, then your vehicle would at least act erratically if it were even stable enough to fly. Filtering via a Kalman filter or otherwise, can reduce the noise when done correctly.

Welcome!

This is a collaboratively edited question and answer site for professional robotic engineers, hobbyists, researchers and students. It's 100% free, no registration required.

Got a question about the site itself? [meta](#) is the place to talk about things like what questions are appropriate, what tags we should use, etc.

[about](#) » [faq](#) » [meta](#) »

Tagged

[kalman-filter](#) × 2

[uav](#) × 2

Asked **Yesterday**

Viewed **30 Times**

Active **Today**