

Bachelor Seminar : Kalman Filter

Struktur

1. Introduction
 - Examples?
2. g-h-Filter
3. Kalman-Filter
4. Outlook: Extended / Unscented Kalman Filter
5. End.
 - Hidden Markov model?

1. Introduction

- Concept of the presented filters: Combine inaccurate measures and worldmodel-based predictions to achieve better accuracy.
- Use case: **Tracking** of real Objects

2. G-H-Filter

- State variables:
 - \hat{x}_k like position
 - \hat{v}_k like velocity
- Input:
 - Measures x_k
- World Model:
 - \hat{x}_k is being predicted each time interval Δt as $\hat{x}_k = \hat{x}_{k-1} + \Delta t \cdot \hat{v}_{k-1}$
- The Update-step with assumedly inaccurate measure x_k uses the **Residual** $\hat{r}_k = x_k - \hat{x}_k$ scaled by **Parameters** g and h to correct predictions:
 - $\hat{x}_k = \hat{x}_{k-1} + \Delta t \cdot \hat{v}_{k-1} + g \cdot \hat{r}_k$
 - $\hat{v}_k = \hat{v}_{k-1} + \frac{h}{\Delta t} \cdot \hat{r}_k$

Choice of g and h

Big Parameters match transients but also emphasize noise. Small Parameters reduce noise but might lead to divergence from real position.

Big values for g that aren't corrected with big h values can push the filter into Resonance and make the velocity increase unreasonably fast.

If chosen well, the algorithm *filters* Measurement-Noise and leads to smoother and more accurate results in tracking

Disadvantages

- Reacts poorly to more complex acceleration

3. Kalman-Filter

- Zustandsvariablen:
 - **Zustand** $\hat{x}_k \in \mathbb{R}^n$
 - * z.B. Position, Geschwindigkeit

- **Kovarianzmatrix** $P_k \in \mathbb{R}^{n \times n}$
- Eingaben:
 - **Messung** z_k (i.Allg. nicht gleiche Einheiten wie x_k)
 - **Messungskovarianz**
 - **Störung** u_k beschreibt den deterministischen & bekannten Einfluss auf den Zustand.
 - * z.B. *Steuerung der Motoren*
-
- Weltmodell:
 - Die **Prediction-Matrix** $F_k \in \mathbb{R}^{n \times n}$
 - * Überführt \hat{x}_k in \hat{x}_{k+1} . Damit lässt sich insbesondere der gesamte g-h-Filter realisieren, aber auch *Beschleunigung* bzw Ableitungen beliebigen Grades.
 - * Die Prediction-Matrix überführt auch P_k zu P_{k+1}
 - Die **Störungs-Dynamik(name?)** B_k überführt die Größe der Störung u_k in die Einheiten des Zustandes.
 - * z.B. *Motorspannung beschleunigt etwas*
 - Die **Sensor-Matrix(name?)** H_k überführt die Zustandsvariablen in den entsprechenden Sensoroutput (falls Sensoren andere Einheiten / Skalen verwenden als das Modell)
 - Der **Kalman-Gain** \hat{K}_k skaliert das Residual im Verhältnis zur Prediction anhand beider Genauigkeiten (?)