

Lab 8: Kalman Filter und Sensorfusion

1. Einleitung

Im heutigen Lab implementieren Sie einerseits einen Kalman Filter Algorithmus, um Messdaten von zwei unterschiedlichen Sensoren zu fusionieren und andererseits werden Sie einen Kalman Filter als Python Klasse implementieren.

Aufgabe 1: Kalman Filter Refactoring

Bisher haben Sie in der Vorlesung primär «Code-Snippets» von Kalman Filter Implementierungen gesehen. Diese eignen sich zu didaktischen Zwecken oder für einfache Implementierungen. Für eine bessere Wiederverwendbarkeit und Übersichtlichkeit lohnt es sich den Kalman Filter als Funktion oder als Klasse zu implementieren.

Im Ordner `Lab 8 Vorlagen` finden sie das Skript `simulate_data.py` mit welchem verrauschte Höhenmessdaten erzeugt werden können (Beispiel 3: Kalman-Filter mit Input). Im Skript `run_kalman_main.py` werden die simulierten Daten eingelesen. Ferner wird ein Objekt der Klasse `KalmanFilter` erzeugt und das Kalman Filter für jeden Messdatenpunkt sequenziell aufgerufen.

Aufgabe

- Implementieren Sie die Klasse `KalmanFilter` im File `kalman_filter_with_input.py` aus und rufen sie die entsprechenden Methoden in `run_kalman_main.py` auf.

Aufgabe 2: Aufzugsgeschwindigkeit schätzen

Bei der Sensorfusion werden häufig unterschiedliche (komplementäre) Sensormessungen fusioniert (z.B. IMU mit GNSS), sodass die Stärken des jeweiligen Sensors zum Tragen kommen. Im Folgenden werden wir Accelerometer- mit Barometermessungen fusionieren.

Ein Accelerometer liefert Beschleunigungsmessungen mit einer hohen Datenrate (typischerweise 100 Hz – 400 Hz). Die Geschwindigkeit- und Positionsschätzung, welche man durch Integration der Beschleunigung erhält, «driftet» jedoch typischerweise schon nach kurzer Zeit weg. Daher bedient man sich oftmals zusätzlicher Positionsmessungen, z.B. mittels GNSS oder Druckmessungen mit einer geringeren Abtastfrequenz, um die geschätzte Position und Geschwindigkeit zu korrigieren.

$$u = 20 \frac{\text{m}}{\text{s}^2}$$

$$h[n+1] = h[n] + \dot{h}[n] \cdot T_s + \ddot{h}[n] \cdot \frac{T_s^2}{2}$$

$$\dot{h}[n+1] = \dot{h}[n] + \ddot{h}[n] \cdot T_s$$

$$x = Ax + Bu$$

$$x = \begin{bmatrix} h \\ \dot{h} \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} T_s^2/2 \\ T_s \end{bmatrix}$$

Versuch

Öffnen Sie die **Phyphox** App und fügen Sie ein neues Experiment hinzu, indem Sie oben rechts auf das « + » Symbol klicken. Wählen Sie “Add simple experiment”. Geben Sie dem Experiment einen Namen (z.B. Treppensteigen oder Aufzug) und wählen Sie den Accelerometer (Beschleunigungssensor) und den Drucksensor aus und nehmen nutzen Sie jeweils die maximale Datenrate.

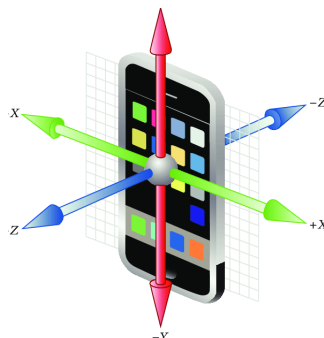


Figure 1: Home-Menü der Phyphox App

Aufgabe:

In diesem Versuch sollen sie die Beschleunigungswerte und den barometrischen Druck messen, während Sie vom 4. Stock des TS-Gebäudes in das Erdgeschoss fahre. Alternativfahren. Statt des Aufzugs können Sie auch die Treppe benutzen.

- Starten Sie das Experiment und gehen Sie vom 4. Stock ins Erdgeschoss. Achten Sie dabei darauf, dass Sie das Smartphone flach in der Hand halten, sodass die z-Achse in entgegengesetzter Richtung zur Schwerkraft zeigt.



- b) Exportieren Sie die Daten (z.B. als .csv) Datei und schicken Sie diese an Ihre ZHAW-Mailadresse. Lesen Sie die Daten in Python (z.B. mit der `pandas` Library) ein und visualisieren Sie das Drucksignal sowie das Beschleunigungssignal. Achtung die zwei Messungen haben jeweils eigene Zeitstempel.
- c) Integrieren Sie das Beschleunigungssignal in z-Richtung ('Z (m/s^2)' Spalte im .csv File). Erhalten Sie damit eine vernünftige Schätzung der Gebäudehöhe und des Höhenverlaufs?
- d) Nun möchten Sie die barometrischen Höhenmessungen mit den Messungen des Beschleunigungssensors fusionieren. Stellen Sie hierfür ein geeignetes Zustandsraummodell der Form:

$$\begin{aligned}x_{k+1} &= A x_k + B u_k \\ y_k &= C x_k\end{aligned}$$

auf.

Tipps

- Der Zustand $x = \begin{pmatrix} h \\ v \end{pmatrix}$ enthält die Höhe und Geschwindigkeit in Z-Richtung
 - Matrix $A = \begin{pmatrix} 1 & T_s \\ 0 & 1 \end{pmatrix}$ beschreibt ein constant-velocity model
 - Die gemessene Beschleunigung kann als Kontrollinput benutzt werden und folglich
 - $B = \begin{pmatrix} 0.5 T_s^2 \\ T_s \end{pmatrix}$
- e) **Falls Sie noch Zeit haben:** Entwerfen Sie einen Kalman Filter, der diese Barometer- und Beschleunigungssensormessungen zu genaueren zeitlich hochaufgelösten Höhenmessungen fusioniert. Hinweis, Sie können den Kalman Filter aus Aufgabe 1 benutzen.

$$x = \begin{bmatrix} h \\ \dot{h} \end{bmatrix}$$

$$h[n+1] = h[n] + \dot{h}[n] \cdot T_s + \ddot{h}[n] \cdot \frac{T_s^2}{2}$$

$$\dot{h}[n+1] = \dot{h}[n] + \ddot{h}[n] T_s$$

$$A = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}$$