Exercises for Course on Modeling and System Identification (MSI)
Albert-Ludwigs-Universität Freiburg – Winter Term 2019-2020

Exercise 10: Kalman Filter

(to be returned on February 5th, 2020, 8:30 in HS 00 036 (Schick - Saal), or before in building 102, 1st floor, 'Anbau')

Prof. Dr. Moritz Diehl, Tobias Schöls, Katrin Baumgärtner, Naya Baslan, Jakob Harzer, Bryan Ramos

In this exercise, you get to know the Kalman Filter. We will implement a simple Kalman Filter and use it for state estimation and sensor fusion.

We consider a robot with omni-directional wheels which means it can instantaneously move in any direction. We assume that the robot's orientation does not change and thus we model the robot's state $x \in \mathbb{R}^6$ by

$$x = \begin{bmatrix} p \\ v \\ a \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ v_1 \\ v_2 \\ a_1 \\ a_2 \end{bmatrix},$$

where $p \in \mathbb{R}^2$ denotes the position of the robot in m, $v \in \mathbb{R}^2$ denotes its velocity in $\frac{\mathrm{m}}{\mathrm{s}}$ and $a \in \mathbb{R}^2$ its acceleration in $\frac{\mathrm{m}}{\mathrm{s}^2}$. Note that we omitted the time dependence for cleaner notation, i.e. $x = x(t)$, $v = v(t)$, $a = a(t)$. The time derivatives of p and v are given by $\dot{p} = v$ and $\dot{v} = a$, respectively. The time derivative of a is given by

$$\begin{aligned} \dot{a}_1 &= -\mu_1 v_1 + u_1 - u_2 \\ \dot{a}_2 &= -\mu_2 v_2 + u_1 + u_2 \end{aligned}$$

with control inputs $u = (u_1, u_2)^\top \in \mathbb{R}^2$ and estimated drag coefficients $\mu_1 = 10^{-1} \frac{1}{\mathrm{s}^2}$ and $\mu_2 = 10^{-2} \frac{1}{\mathrm{s}^2}$. The terms $\mu_1 v_1$ and $\mu_2 v_2$ model friction.
We assume that the control inputs u are perfectly known.

1. ON PAPER: Specify the continuous time state-space model, i.e. define matrices $A_c \in \mathbb{R}^{6\times 6}$ and $B_c \in \mathbb{R}^{6\times 2}$ such that the following holds:

   $$\dot{x} = f(x, u) = A_c x + B_c u$$

   Formulate the corresponding discrete time model for the robot's dynamics using a one-step Euler integrator with step length $h = 0.5$ s, i.e. specify matrices $A_d \in \mathbb{R}^{6\times 6}$ and $B_d \in \mathbb{R}^{6\times 2}$ such that

   $$x_{k+1} = F(x_k, u_k) = A_d x_k + B_d u_k$$

   (2 points)

2. ON PAPER: We assume that the discrete time state dynamics are perturbed by additive zero-mean Gaussian noise. Note that we cannot observe the state directly, but can only measure the robot's position p using a GPS sensor. These GPS measurements are perturbed by additive zero-mean Gaussian noise with covariance matrix Σ_{γ_p}.

Summarizing, our state and measurement model has the form

$$x_{k+1} = A_d x_k + B_d u_k + \chi_k, \tag{1}$$
$$y_k = C x_k + \gamma_k, \tag{2}$$

where $\chi_k = (\chi_{p,k}, \chi_{v,k}, \chi_{a,k})^\top$, $\chi_k \sim \mathcal{N}(0, \Sigma_\chi)$ and $\gamma_k \sim \mathcal{N}(0, \Sigma_\gamma)$. We assume

$$\Sigma_{\chi_p} = 2 \cdot 10^{-2} \cdot \mathbb{I}\, \mathrm{m}^2$$
$$\Sigma_{\chi_v} = 4 \cdot 10^{-3} \cdot \mathbb{I}\, \frac{\mathrm{m}^2}{\mathrm{s}^2}$$
$$\Sigma_{\chi_a} = 1 \cdot 10^{-8} \cdot \mathbb{I}\, \frac{\mathrm{m}^2}{\mathrm{s}^4}$$
$$\Sigma_{\gamma_p} = 16 \cdot \mathbb{I}\, \mathrm{m}^2$$

Specify the matrix $C \in \mathbb{R}^{2\times 6}$ such that $y_k \in \mathbb{R}^2$ corresponds to the noisy GPS measurement. Also write down the covariance matrices Σ_χ and Σ_γ. (1 point)

3. MATLAB: Write two functions

```
[x_predict, P_predict]  = predict(x_estimate, P_estimate, A, b, W)
[x_estimate, P_estimate] = update(y, x_predict, P_predict, C, V)
```

that implement the prediction and update step of the Kalman filter.

Hint: See equations (9.19) to (9.24) on page 103 of the lecture notes. Here `x_predict` *corresponds to* $x_{[k|k-1]}$ *and* `x_estimate` *corresponds to* $x_{[k|k]}$.

(2 points)

4. MATLAB: For the given measurement and control trajectories, $y = (y_0, \ldots, y_N)$ and $u = (u_0, \ldots, u_{N-1})$, compute the state estimates $x_{[k|k]}$ and state predictions $x_{[k|k-1]}$ where we assume an estimated initial state $x_0 \sim \mathcal{N}(0, \Sigma_0)$ where $\Sigma_0 = 10^{-5} \cdot \mathbb{I}$, i.e. we assume to know the inital state almost exactly.

(2 points)

5. MATLAB: We already provided code to plot the estimated trajectory, the predicted position $p_{[k|k-1]}$ and the corresponding confidence ellipsoids.

You only have to compute $\Sigma_{p_{[k|k-1]}}$ from $P_{[k|k-1]}$. (1 point)

6. ON PAPER: As the GPS measurements are very noisy, we equip our robot with an additional accelerometer that can measure the robot's acceleration. These measurements are perturbed by additive Gaussian noise with covariance matrix $\Sigma_{\gamma_a} = 2.25 \cdot 10^{-6} \cdot \mathbb{I}\, \frac{\mathrm{m}^2}{\mathrm{s}^4}$.

Consider again the state and measurement equations:

$$x_{k+1} = A_d x_k + B_d u_k + \chi_k, \tag{3}$$
$$\tilde{y}_k = \tilde{C} x_k + \tilde{\gamma}_k, \tag{4}$$

Specify $\tilde{C} \in \mathbb{R}^{4\times 6}$ such that $\tilde{y}_k$ now also includes the (noisy) acceleration measurements. What does $\Sigma_{\tilde{\gamma}}$ look like? (1 point)

7. MATLAB: Repeat part 5 and 6, using now both position and acceleration measurements. This approach is generally referred to as *sensor fusion*, i.e. we combine data from multiple sensors that produce measurements with different units, dimensions and accuracies, in order to obtain a more accurate state estimate.

ON PAPER: Compare your results. (2 points)

This sheet gives in total 11 points.