Exercises for Course on State-Space Control Systems (SSC)
Albert-Ludwigs-Universität Freiburg – Summer Term 2019

Exercise 9: Extended Kalman Filter & Unscented Kalman Filter

Prof. Dr. Moritz Diehl, Dr. Dang Doan, Benjamin Stickan, Katrin Baumgärtner

Exercises

1. *(EKF with dynamical model depending nonlinearly on both state and disturbance)* Consider a discrete nonlinear system with state-space representation as follows

$$\begin{aligned} x_{k+1} &= f(x_k, w_k) \\ y_k &= h(x_k) + v_k \end{aligned}$$

in which disturbance w_k and measurement noise v_k are independent zero-mean, Gaussian white noises with covariances $\mathbf{Q}$ and $\mathbf{R}$.

Derive the extended Kalman filter for this system. (hint: linearize f at the current state estimate and zero disturbance).

2. Consider the discrete-time nonlinear system:

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} &= \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + T \begin{bmatrix} \cos\theta_k & 0 \\ \sin\theta_k & 0 \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} v_k \\ \omega_k \end{bmatrix} + \mathbf{w}_k \right), \quad \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}) \\ \begin{bmatrix} r_k \\ \phi_k \end{bmatrix} &= \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \arctan 2(-y_k, -x_k) - \theta_k \end{bmatrix} + \mathbf{n}_k, \quad \mathbf{n}_k \sim \mathcal{N}(0, \mathbf{R}) \end{aligned}$$

which could represent a mobile robot moving around on the $xy-$plane and measuring the range and bearing to the origin. Set up the EKF equation to estimate the state of the mobile robot. In particular, work out expressions for the Jacobians.

3. *(Implement unscented Kalman filter)* We want to verify several ways of choosing the sigma points z_k^i and weights ω^i for $x \sim \mathcal{N}(\bar{x}_k, P_k)$ at each time step k, and then run the UKF.

   (a) Verify the following symmetric set of $N = 2n+1$ points, where n is the dimension of state x, is the set of sigma points:
   Choose $\lambda > -n$, weights:

$$\begin{aligned} \omega^0 &= \frac{\lambda}{n+\lambda} \\ \omega^i = \omega^{i+n} &= \frac{1}{2(n+\lambda)}, \quad \text{for } i = 1, \dots, n \end{aligned}$$

   and points:

$$\begin{aligned} z^0 &= \bar{x}_k \\ z^i &= \bar{x}_k + \left(\sqrt{(n+\lambda)P_k}\right)_i, \quad \text{for } i = 1, \dots, n \\ z^{i+n} &= \bar{x}_k - \left(\sqrt{(n+\lambda)P_k}\right)_i, \quad \text{for } i = 1, \dots, n \end{aligned}$$

   where the notation $\left(\sqrt{(n+\lambda)P_k}\right)_i$ means the i^{th} column of the matrix A that is square root of $(n+\lambda)P_k$, i.e. $AA = (n+\lambda)P_k$.

   (b) Implement a recursive UKF using this choice of sigma points as functions in MATLAB.