

Exercise Round 4.

The answers to the exercises should be returned as follows:

- The deadline for exercise rounds 4–6 (there are 3 exercises on each round) is **May 21, 2010**.

The answers should be sent as email to the teacher (ssarkka@lce.hut.fi) in PDF form. When sending the email, please add "S-114.4202" or "1144202" to subject. The answers can also be returned on paper to the teacher.

Exercise 1. (Extended Kalman filter)

Write extended Kalman filter (EKF) equations for the following dynamic model:

$$\begin{aligned} \mathbf{x}_k &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \mathbf{x}_{k-1} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \mathbf{w}_{k-1} \\ y_k &= \tan^{-1} \left(\frac{x_{k,2} - s_y}{x_{k,1} - s_x} \right) + v_k, \end{aligned} \quad (1)$$

where \mathbf{x}_k is the state, y_k is the measurement, and $\mathbf{w}_k \sim N(0, 1)$ and $v_k \sim N(0, 1)$ are white Gaussian noise processes.

Exercise 2. (Statistically linearized filter)

Consider the following state space model

$$\begin{aligned} x_k &= x_{k-1} - 0.01 \sin(x_{k-1}) + q_{k-1} \\ y_k &= 0.5 \sin(2 x_k) + r_k, \end{aligned} \quad (2)$$

where q_{k-1} has variance 0.01^2 and r_k has variance 0.02 .

A) Derive the required derivatives for EKF and implement EKF for the model.

B) Derive the required expected values for SLF and implement SLF for the model.

Hint: Imaginary part of the inverse Fourier transform of normal distribution.

Simulate state trajectories and measurements from the model, report RMSE values for each method and plot the results. Also include the Matlab codes to the returned document.

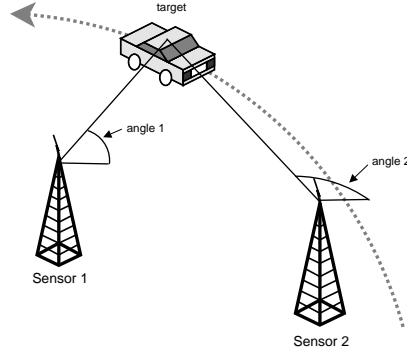
Exercise 3. (Bearings Only Target Tracking with EKF)

Figure 1: In bearings only target tracking problem the sensors generate angle measurements of the target, and the purpose is to determine the target trajectory.

In this exercise we shall consider a classical bearings only target tracking problem, which frequently arises in the context of passive sensor tracking. In this problem there is single target in the scene and two angular sensors are used for tracking it. The scenario is illustrated in the Figure 1.

The state of the target at time step k consist of the position (x_k, y_k) and the velocity (\dot{x}_k, \dot{y}_k) . The dynamics of the state vector $\mathbf{x}_k = (x_k \ y_k \ \dot{x}_k \ \dot{y}_k)^T$ are modeled with the discretized Wiener velocity model:

$$\begin{pmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{pmatrix} + \mathbf{q}_{k-1},$$

where \mathbf{q}_k is a zero mean Gaussian process noise with covariance

$$\mathbf{Q} = \begin{pmatrix} q_1^c \Delta t^3/3 & 0 & q_1^c \Delta t^2/2 & 0 \\ 0 & q_2^c \Delta t^3/3 & 0 & q_2^c \Delta t^2/2 \\ q_1^c \Delta t^2/2 & 0 & q_1^c \Delta t & 0 \\ 0 & q_2^c \Delta t^2/2 & 0 & q_2^c \Delta t \end{pmatrix}$$

In this scenario the diffusion coefficients are $q_1^c = q_2^c = 0.1$ and the sampling period is $\Delta t = 0.1$. The measurement model for sensor $i \in \{1, 2\}$ is the following:

$$\theta_k^i = \tan^{-1} \left(\frac{y_k - s_y^i}{x_k - s_x^i} \right) + r_k, \quad (3)$$

where (s_x^i, s_y^i) is the position of the sensor i and $r_k \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian measurement noise with standard deviation of $\sigma = 0.05$ radians. At each sampling

time, which occurs 10 times per second (i.e., $\Delta t = 0.1$), each of the two sensors produce a measurement.

In file `angle_ex.m` there is a base line solution, which computes estimates of the position from the crossing of the measurements and estimates the velocity to be always zero. Your task is to implement extended Kalman filter for the problem and compare the results graphically and in RMSE sense.

A) Implement an extended Kalman filter to the bearings only target tracking problem, which uses the non-linear measurement model (3) as its measurement model function (not the crossings). Hints:

- Use Matlab/Octave function `atan2` in the measurement model instead of `atan` to directly get an answer at range $[-\pi, \pi]$.
- The two measurements at each measurement time can be processed one at a time, that is, you can simply perform two scalar updates instead of a single two dimensional measurement update.
- Start by computing the Jacobian matrix of the measurement model function with respect to the state components. Before implementing the filter, check by finite differences that the Jacobian matrix is correct.

Report the RMSE values and plot figures of the estimates. Also include the matlab code to the returned document. You may use EKF/UKF toolbox in this exercise, if it helps.