Computer Exercise

The exercise report should be returned before **December 15, 2006**. The report should be sent as email to the teacher (ssarkka@lce.hut.fi) in PDF form. The Matlab/Octave-files should be returned as a ZIP or TAR.GZ file together with the PDF report. In the report, in addition to the results and figures, document the Matlab/Octave-codes that you have written. When sending the email, please add "S-114.4220" or "1144220" to subject. The report can also be returned on paper to the teacher.

The Matlab/Octave files for the exercises can be found in URL

`http://www.lce.hut.fi/teaching/S-114.4220/s2006/matlab/`

The files are compatible with Matlab 6.x/7.x and they do not need any additional toolboxes. The files should also work with sufficiently new versions of GNU Octave.

Computer Exercise 1. (Kalman Filter)

Consider the following dynamic model:

$$\mathbf{x}_k = \begin{pmatrix} \cos\omega & \frac{\sin(\omega)}{\omega} \\ -\omega\,\sin(\omega) & \cos(\omega) \end{pmatrix} \mathbf{x}_{k-1} + \mathbf{q}_{k-1}$$
$$y_k = \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{x}_k + v_k$$

where $\mathbf{x}_k \in \mathbb{R}^2$ is the state, y_k is the measurement, $v_k \sim N(0, 0.1)$ is a white Gaussian measurement noise, and $\mathbf{q}_k \sim N(\mathbf{0}, \mathbf{Q})$, where

$$\mathbf{Q} = \begin{pmatrix} \frac{q_c\,\omega - q_c\,\cos(\omega)\,\sin(\omega)}{2\omega^3} & \frac{q_c\,\sin^2(\omega)}{2\omega^2} \\ \frac{q_c\,\sin^2(\omega)}{2\omega^2} & \frac{q_c\,\omega + q_c\,\cos(\omega)\,\sin(\omega)}{2\omega} \end{pmatrix} \tag{1}$$

The angular velocity is $\omega = 1/2$ and the spectral density is $q_c = 0.01$. The model is a dicretized version of noisy resonator model with angular velocity ω.

In the file `kf_ex.m` there is simulation of the dynamic model together with a base line solution, where the measurement is directly used as the estimate of the state component x_1 and the second component x_2 is computed as a weighted average of the measurement differences.

A) Implement a Kalman filter and compare its performance (in RMSE sense) to the base line solution. Plot figures of the solutions and report the RMSE values of for both the methods.

B) Compute the stationary Kalman filter corresponding to the model. Test this stationary filter against the base line and Kalman filter solutions. Plot the results and report the RMSE values for the solutions. What is the practical difference in the stationary and non-stationary Kalman filter solutions?

Computer Exercise 2. (Bearings Only Target Tracking)

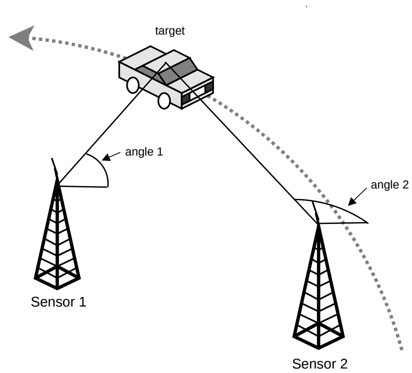

Figure 1: In bearings only target tracking problem the sensors generate angle measurements of the target, and the purpose is to determine the target trajectory.

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:

$$\theta_k^i = \tan^{-1}\left(\frac{y_k - s_y^i}{x_k - s_x^i}\right) + r_k, \tag{2}$$

where (s_x^i, s_y^i) is the position of the sensor i and $r_k \sim \mathrm{N}(0, \sigma^2)$ 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 and bootstrap 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 (2) 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.

B) Implement a bootstrap filter to the bearings only target tracking problem, which uses the dynamic model as its importance distribution and performs resampling after each measurement update. Hints:

- You can use the function `gauss_rnd.m` for simulating samples from the importance distribution and the function `gauss_pdf.m` for evaluating the likelihood of the measurement.
- Use the function `resampstr.m` for performing the resampling. If `SX` is a $n \times m$ matrix of m samples of dimension n and `W` is $1 \times m$ matrix of likelihoods, the resampling can be performed as follows:

```
ind = resampstr(W);
SX = SX(:,ind);
```

Report the RMSE values for each of the methods and plot figures of the estimates.