Instructions: All course participants are requested to submit their exercise solutions electronically to the instructors (sourav.bhattacharya at cs.helsinki.fi and teemu.pulkkinen at cs.helsinki.fi), as well as, to the course lecturer (petteri.nurmi at cs.helsinki.fi) by the due date (latest before the exercise session). In all the exercises, do not just give the answer, but also the derivation how you obtained it. Participants are encouraged to write computer programs to derive solutions to some of the given problems.

1) Filtering
Compare the Kalman filter and particle filters in terms of their advantages and disadvantages. Mention how techniques that can be used to overcome some of the disadvantages of particle filters.

E.g.: Kalman filters assume the system can be modeled using the Gauss-Markov model, i.e. it assumes a linear system. Particle filters can handle the non-linear case (non-linear state space and noise). On the other hand, Kalman gains can be pre-calculated and stored offline. Particle filters also demand a lot of system resources, which can be alleviated by adjusting the number of particles (e.g. using Kullback-Leibler Divergence sampling).

2) Mathematical
a) Assume \( p(y_k|x_k^i) \) (Lecture V, pp. 33) is given by an exponential distribution \( \exp(-\lambda d_i) \), where \( \lambda = 0.01 \). Given the set of particles (see Table 2), their weights and travelled distances \( (d_i) \) determine their new weights. See Table 1 for the new weights. Essentially, we multiply each weight with the value we receive from applying the exponential formula to the relevant distance. So for weight 1 we input 
\[
0.0372 \times \exp(-0.001 \times 3.153) \approx 0.0371,
\]
and similarly for the other weights. After we normalize the weights (to make sure they sum up to one and represent a distribution), we receive the values in the aforementioned table.

b) Given the new weights of the particles, calculate the number of effective particles
Table 1: New weights for particles for exercise 2)

<table>
<thead>
<tr>
<th>Index</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0375</td>
</tr>
<tr>
<td>2</td>
<td>0.0284</td>
</tr>
<tr>
<td>3</td>
<td>0.0005</td>
</tr>
<tr>
<td>4</td>
<td>0.0205</td>
</tr>
<tr>
<td>5</td>
<td>0.0730</td>
</tr>
<tr>
<td>6</td>
<td>0.1136</td>
</tr>
<tr>
<td>7</td>
<td>0.0076</td>
</tr>
<tr>
<td>8</td>
<td>0.0050</td>
</tr>
<tr>
<td>9</td>
<td>0.7138</td>
</tr>
<tr>
<td>10</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

after the update. Assume the resampling threshold is $N_{EFF} = 2$. Is there any need for resampling?

$$N_{EFF} = \frac{1}{\Sigma(w_k)^2} \approx 1.8850 < 2 \implies \text{We should resample.}$$

c) The file in URL\(^1\) contains a distance matrix for 10 points in a Cartesian coordinate system. Each line represents the distance from one point to all the others. In other words, the third value of the fourth line represents the distance between the 3rd and the 4th point, etc. If we assume an Epsilon neighborhood of size 35 and a MinPts threshold of 4, describe which points are i) core objects, ii) non-core objects and iii) noise.

If we calculate how many points are within the defined Epsilon neighborhood for each point, we find that i) points $\{1,4,5,6,10\}$ are core objects because their neighborhood contains 4 points, ii) points $\{3,8,9\}$ are non-core objects because they’re within the neighborhood of a core object (1, in this case) but are not core objects themselves and iii) points $\{2,7\}$ are considered noise because they are outside of the Epsilon neighborhood of all core points. Figure 1 visualizes the locations of the points.

4) Kalman filter, 2pts
Calculate the $m^*_1$ (prior) and $P^*_1$ (posterior) estimates for a Kalman filter for the following parameter settings.

\(^1\)http://universe.hiit.fi/teaching/location-awareness/ex-3/distance.csv
Figure 1: Plot of object locations for exercise 3 c)

\[ A = H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \]

\[ P = \begin{pmatrix} 0.55 & 0 & 0 \\ 0 & 0.06 & 0 \\ 0 & 0 & 0.06 \end{pmatrix} \]

\[ R = \begin{pmatrix} 7 & 0 & 0 \\ 0 & 0.12 & 0 \\ 0 & 0 & 0.12 \end{pmatrix} \]

\[ y_0 = [54.97, 0.78, 0.86], \ m_0 = [54.05, 0.76, 0.80], \ \sigma = 0.05, \ \sigma I. \] (Hint: Refer to lecture slide V, pp. 22-23).

The formulas we need are as follows:

\[ m_k^* \sim H_{k-1} \ast m_{k-1} \]
\begin{align*}
P^*_k \sim & \ H_{k-1} \ast P_{k-1} \ast H_{k-1}^T + Q \\
& \text{for the priori estimates, and} \\
m_k \sim & \ m^*_k + P^*_k \ast H^T_k \ast (H_k \ast P^*_k \ast H^T_k + R_k)^{-1} \ast (y_k - H_k \ast m^*_k) \\
P_k \sim & \ P^*_k - P^*_k \ast H^T_k \ast (H_k \ast P^*_k \ast H^T_k + R_k)^{-1} \ast P^*_k \ast H^T_k \\
& \text{for the update step. We first perform the prediction step (estimate the priors):} \\

m^*_1 = & \ H \ast m_0 = m_0. \text{ In other words, the prior for } m^* \text{ doesn't change because } H = I. \\
P^*_1 = & \ H \ast P \ast H^T + Q = P + Q = \\
& \begin{pmatrix} 
0.05 + 0.55 & 0 & 0 \\
0 & 0.05 + 0.06 & 0 \\
0 & 0 & 0.05 + 0.06 
\end{pmatrix} \\
\begin{pmatrix} 
0.6 & 0 & 0 \\
0 & 0.11 & 0 \\
0 & 0 & 0.11 
\end{pmatrix} \\
& \text{Note that } H \text{ (and } H^T) \text{ are } I, \text{ which means they can be left out of the calculations. Next, we perform the update step using the given formulas, and receive} \\

m_1 = & \ [54.12, 0.77, 0.83] \\
and & \\
P_1 = \begin{pmatrix} 
0.56 & 0 & 0 \\
0 & 0.07 & 0 \\
0 & 0 & 0.07 
\end{pmatrix}
\end{align*}

5) Programming, \(2\)pts
a) Download GPS trajectory of a user from the URL\(^2\). The columns of the file represent \textit{longitude}, \textit{latitude}, \textit{timestamp}, \textit{satellites} and \textit{hdop}. Perform preprocessing of the data and show the output. (Hint: you can perform your analysis in the Euclidean space. To get a distance measure in meters you can use \(0.005\) unit distance in Euclidian space = \(500\) meters).

b) Perform place identification on the preprocessed data and show the results (plot the centers).

\(^2\)http://universe.hiit.fi/teaching/location-awareness/ex-3/tokyo.csv
<table>
<thead>
<tr>
<th>Index</th>
<th>Weight</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0372</td>
<td>3.153</td>
</tr>
<tr>
<td>2</td>
<td>0.0293</td>
<td>7.244</td>
</tr>
<tr>
<td>3</td>
<td>0.0005</td>
<td>4.423</td>
</tr>
<tr>
<td>4</td>
<td>0.0213</td>
<td>7.666</td>
</tr>
<tr>
<td>5</td>
<td>0.0721</td>
<td>2.705</td>
</tr>
<tr>
<td>6</td>
<td>0.1100</td>
<td>0.793</td>
</tr>
<tr>
<td>7</td>
<td>0.0074</td>
<td>0.733</td>
</tr>
<tr>
<td>8</td>
<td>0.0051</td>
<td>6.465</td>
</tr>
<tr>
<td>9</td>
<td>0.7171</td>
<td>4.468</td>
</tr>
<tr>
<td>10</td>
<td>0.0000</td>
<td>4.412</td>
</tr>
</tbody>
</table>

Table 2: Particles for exercise 2)