LOCALIZATION, NAVIGATION AND SMART MOBILITY
Project Assignment, AY 2019/2020

Mattia Brambilla, Monica Nicoli

Goal

The goal of the project is to assess the student capability in handling a localization problem, by formulating the model, implementing a positioning method and assessing the related performance in Matlab environment. The student is requested to apply the models and methods learned during the course lessons to a practical use case. The project is organized in a number of subtasks, built using a sequential approach, with incremental complexity.

The evaluation of the project will take into account the extension (e.g., number of tasks) and quality of the development, as well as the presentation of the project results.

Scenario and notation

Localization scenario: 2D area of 4x4km.
UE: $\boldsymbol{u} = [u_x \quad u_y]$
AP: $\boldsymbol{s}_i = [s_{i,x} \quad s_{i,y}]$

Tasks:

1. **Learning the AP positions and the measurement model**
   a) Use the data in the variable "*rhoUEAP.mat*" in "Task1a_*rhoUEAP.mat*" to compute the location of each AP $\boldsymbol{s}_i$ with respect to a reference point located in $\boldsymbol{u} = [0 \quad 0]$**.**
   "*rhoUEAP.mat*" is a N_{AP}x2 matrix containing TOA measurements (on the first column) and AOA measurements (in the second column). These measurements are computed as:

$$\rho_i = h_i(\boldsymbol{u}, \boldsymbol{s}_i) = \begin{cases} |\boldsymbol{u} - \boldsymbol{s}_i|_2 & \text{for TOA} \\ tan^{-1}\left(\dfrac{u_y - s_{iy}}{u_x - s_{ix}}\right) & \text{for AOA} \end{cases}$$

   No noise is introduced in the measurements such that the estimated AP locations are the true one.
   Positions are integer numbers, so please round the values (e.g., 1.9 is 2).

   b) Use the TOA measurement matrix "*rhoUEAP.mat*" in "*Task1b_rhoUEAP.mat*" to evaluate the measurement accuracy (standard deviation) of each AP.
   "*rhoUEAP.mat*" is a N_{Meas}xN_{AP} matrix containing TOA measurements of each AP with respect to a UE located in $\boldsymbol{u} = [500 \quad -800]$**.**
   Standard deviations/variances are integer numbers, so please round the values (e.g., 1.9 is 2).

Expected output of Task1: position of APs, covariance matrix of TOA measurements

2. **Learning the motion model dataset for motion models**
   The cell variable "UEtrajectories.mat" in "Task2_trajectory.mat" is a dataset of 100 trajectories obtained with a mobility motion model seen during the lessons. Each cell contains a 200x4 matrix, where the rows scan over time, while the columns indicate the position and velocity ($p_x \quad p_y \quad v_x \quad v_y$) of the UE. The sampling time is 1sec.
   Use this dataset (or part of it) to calibrate the motion model.

   Expected output of Task2: motion model statistics and parameters.

3. **Kalman filter testing on known trajectory**
   Implement a Kalman Filter (KF) to track the UE over time. Use the equations in the lecture notes.
   For each trajectory (out of 100 of task 2) the correspondent matrix of TOA measurements (obtained as in Task 1b) is stored in each cell of the variable "*rhoUEAP.mat*" in *"Task3_rhoUEAP_GRXX.mat".*
   The implementation of KF should consider the TOA measurement accuracy and the mobility properties that have been calibrated in tasks 1b and 2, respectively.
   The best result are obtained when the model used for tracking is matched to the model hidden in the provided trajectory dataset. Any mismatch, either in the TOA measurement or in the mobility model, leads to a lower localization accuracy.
   For the prior pdf: you can choose a uniform prior, or initialize the estimate to the first measurements, or any other reasonable choice.

   Expected output of Task3: the students should provide a detailed analysis of the KF performance, on one or more trajectories. Students are free to use the performance metrics they consider as most appropriate and reliable, given the provided dataset (e.g., distance error, RMSE, CEP, CDF, etc.).

4. **Kalman filter testing on unknown trajectory**
   Implement the KF by using the variable "*rhoUEAP.mat*" in *"Task4_rhoUEAP_GRXX.mat".* The reconstructed trajectory should emulate a car, which at the beginning is moving at 50 km/h along the x-axis.
   Does it work? If yes, can it be improved? If no, try to adapt it

   Expected output of Task4: tracked trajectory.

5. **Kalman filter testing on unknown trajectory**
   Implement the KF by using the variable "*rhoUEAP.mat*" in *"Task5_rhoUEAP_GRXX.mat".*

   Expected output of Task5: tracked trajectory and critical comparison with the output of task 4.

6. **Kalman filter testing on unknown trajectory**
   Implement the KF by using the variable "*rhoUEAP.mat*" in *"Task6_rhoUEAP_GRXX.mat".*
   Note: the code of tasks 3,4,5 *should not work*, you will notice by inspecting the measurement matrix. Try to make it adaptive and make it work.

   Expected output of Task6: tracked trajectory and critical comparison with the output of tasks 4 and 5.