Assignment 3

Numerical Optimization / Optimization for CS WS2019

Erich Kobler, erich.kobler@icg.tugraz.at
Sarah Schneider, sarah.a.schneider@student.tugraz.at
Tarik Softic, tarik.softic@student.tugraz.at

December 10, 2019

Submission: Upload your implementation as a single `*.py` file and your report as a single `*.pdf` file to the TU-Graz TeachCenter.
Deadline: January 15$^{\text{th}}$, 2020 at 18:00.

1 Position Estimation

In this task we have to accurately determine the location $\mathbf{x} = (x_1, x_2)^\top \in \mathbb{R}^2$ of a floating sailing boat because we would like to sail through a tight shoal. Since the GPS of the boat is broken, we have to use the compass to bear three surrounding light towers $\mathbf{l}^j = (l_1^j, l_2^j)^\top \in \mathbb{R}^2$ for $j \in \{1, 2, 3\}$. To avoid measurement biases due to environmental factors (water current and wind), three crew members measure the angles z_i^t for $t = 1, \ldots, 60$ rapidly, as sketched in Figure 2. However, each of the determined angles z_j^t is prone to measurements errors.

Our task as a navigator is to determine our position. Given the noisy angle measurements in conjunction with the corresponding light tower positions (`data_position.npz`), we have to estimate our position $\mathbf{x}$. Here we assume that the angle measurement z_j^t of the j^{th} light tower at time t follow the model

$$z_j^t = \arctan\left(\frac{l_2^j - x_2}{l_1^j - x_1}\right) + n_j^t\,,$$

where the noise n_i^t is unknown.

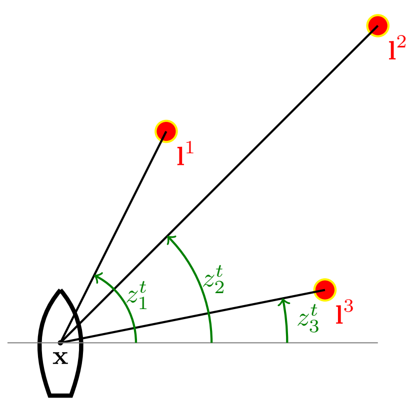

Figure 1: We know the position of the light towers $\mathbf{l}^j$ as well as the measured angles z_j^t and have to determine the position of our ship $\mathbf{x}$.

To complete this task you have to:

- Design the state vector $\mathbf{x}$.
- Choose an initial position in the north east of $\mathbf{l}_1$ around $(2, 2)^\top$.

- Implement an extended Kalman filter to sequentially process the measurements.
- Plot the objective function of the extended Kalman filter for the problem region, mark the position of the light towers, the initial position, all you sequential position estimates, as well as the optimal position of the time varying objective function. Use a contour plot to visualize the level sets of the objective function in the 2D plane.
- Discuss your results for $\lambda \in \{0.6, 0.9\}$.

2 Motion Estimation

After we have determined our position, the weather changes such that dense fog blocks the view to the light towers. However, we still can measure the distances z_i^t to the individual light towers electronically. Despite this bad weather we need to sail trough the narrow shoal. In order to navigate, we use a linear motion model

$$\mathbf{x}_t = \mathbf{x}_0 + t \cdot \mathbf{v}\,, \tag{1}$$

where $\mathbf{x}_0$ is the initial position, t the time, and $\mathbf{v}$ the vector valued course. Every time step t we measure the distances to all light towers

$$z_j^t = \|\mathbf{l}^j - \mathbf{x}_t\|_2 + n_j^t\,,$$

where again n_j^t models the measurement noise. Figure 2 visualizes the geometry.

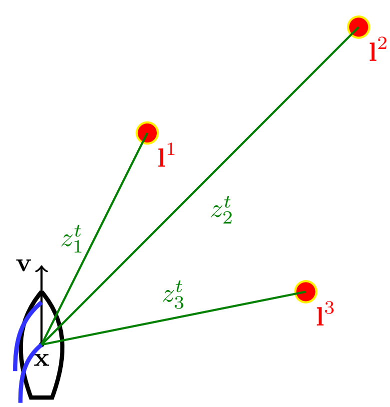

Figure 2: The positions of the light towers are known and we can measure the distance z_j^t to each tower at time t electronically. Our job is to determine our position $\mathbf{x}$ and course $\mathbf{v}$ given the linear motion model.

Now, we would like to determine our position and course. To do so, you have to solve the following tasks:

- Design the state vector $\mathbf{x}$.
- Choose an initial position and course based on the results form the first task.
- Implement an extended Kalman filter to sequentially process the measurements.
- Plot the objective function of the extended Kalman filter for the problem region, mark the position of the light towers, the initial position, all you sequential position estimates, as well as the optimal position of the time varying objective function. Use a contour plot to visualize the level sets of the objective function in the 2D plane. Present your results for two different time steps.
- Plot the course for each time step in a 2D scatter plot.
- Discuss your results for $\lambda \in \{0.6, 0.9\}$.
- How many course changes have been performed?
- Which courses $\mathbf{v}$ were required to sail through the narrow shoal?

Framework

Use the provided python file for your implementation. It already provides the functionality to load data. For this task, you are only allowed to use the `numpy`, and `matplotlib` packages.

Show and describe all your results in the report!