NTNU
Kunnskap for en bedre verden

Department of Engineering Cybernetics

TTK4250 - Sensor Fusion

Graded Assignment 2

Authors:
Aleksander Østensen
Olav Livsønn Bekken
Henrik Horpedal

13th November 2024

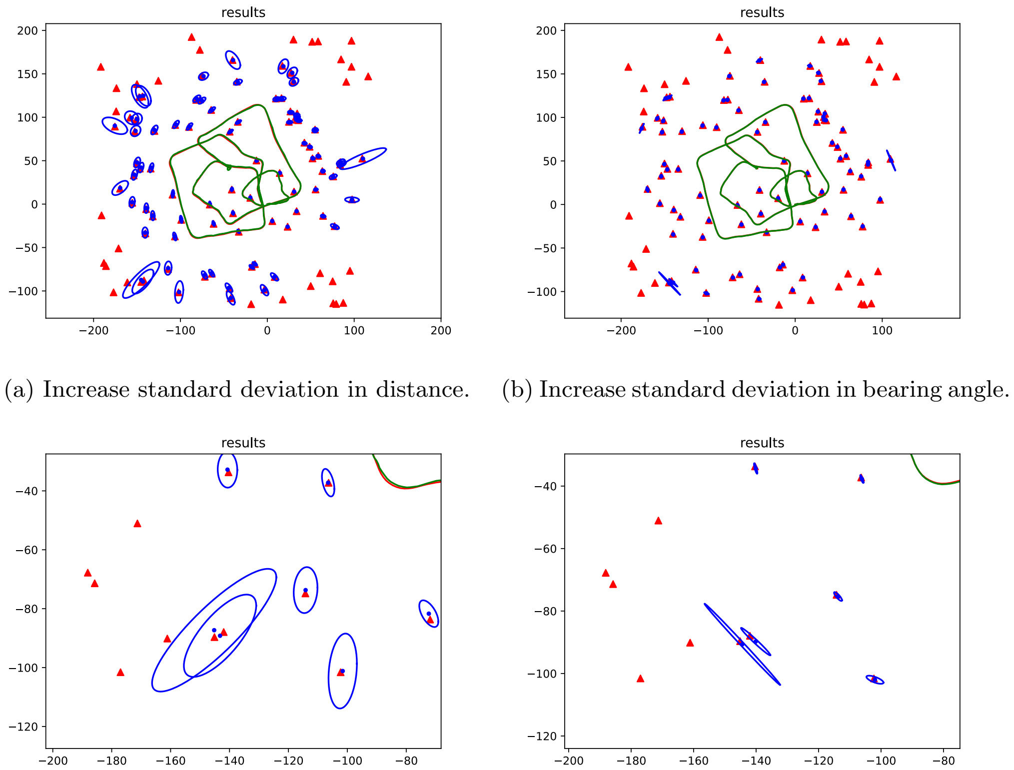
Task 2: Run EKFSLAM on simulated data

Initial parameters:

```
Q = np.diag([0.1, 0.1, 1 * np.pi / 180]) ** 2
R = np.diag([0.5, 5 * np.pi / 180]) ** 2
JCBBalphas = np.array([0.001, 0.0001])
```

Tuning parameters 1

The first entry in R corresponds to standard deviation in distance to landmarks. We increased this by a factor of 100, and noticed a significant increase in uncertainty regarding distance to landmarks. This is seen in figure 1a and 1c. The second entry represents standard deviation in the bearing angle to landmarks. By increasing this by a factor of 10. The system became more uncertain about the bearing angle to landmarks. The results are shown in 1b and 1d. This illustrates how different R values independently affect the system's uncertainty in distance and angle measurements. Each entry controls uncertainty in a specific direction.



(a) Increase standard deviation in distance.

(b) Increase standard deviation in bearing angle.

(c) Zoomed in plot of increase standard deviation in distance to landmarks.

(d) Zoomed in increase standard deviation in bearing angle.

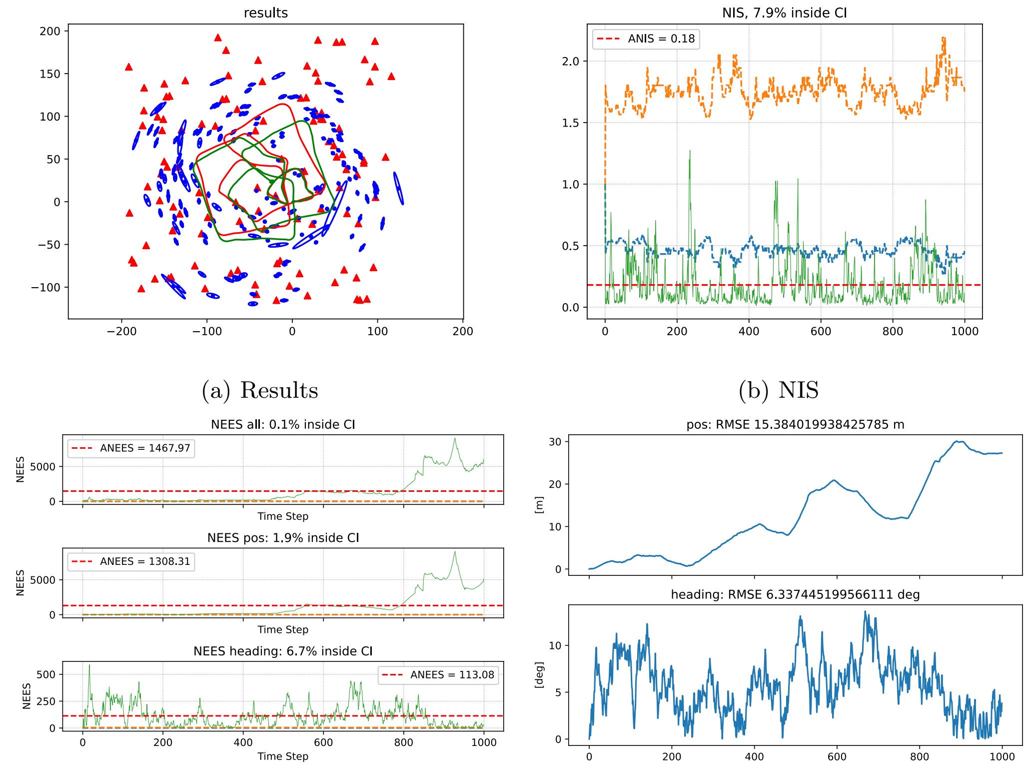
Figure 1: Increasing values in R

Tuning parameters 2

The tuning parameters were changed to:

```
Q = np.diag([0.1, 0.1, 0.1 * np.pi / 180]) ** 2
R = np.diag([0.5, 5 * np.pi / 180]) ** 2
JCBBalphas = np.array([0.1, 0.1])
```

Setting R higher, while lowering Q, makes the filter less confident in the measurements. Additionally we tuned the alphas of JCBB to be much higher, making the gating test (11.29) harder to pass [1, p. 214]. This leads to fewer associations being made and rejected measurements are considered as new landmarks. Measuring the same landmark more than once, may therefore lead to duplicates being made. Future measurements will then have multiple candidate landmarks that are close in position, possibly leading to incorrect associations and ultimately the degradation of the map and pose as seen in figure 2a.



(a) Results

(b) NIS

(c) NEES

(d) RMSE

Figure 2: Result, NIS, NEES, and RMSE for high alphas and R

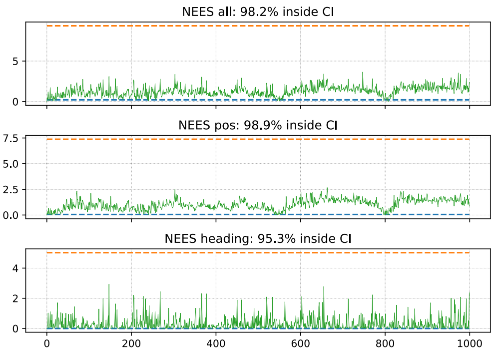
The filter is conservative and won't utilize all the information from the measurements due to the high R, seen in the low NIS value. The RMSE in position grows steadily over time, and the ANEES in position becomes high. See figure 2.

Tuning parameters 3

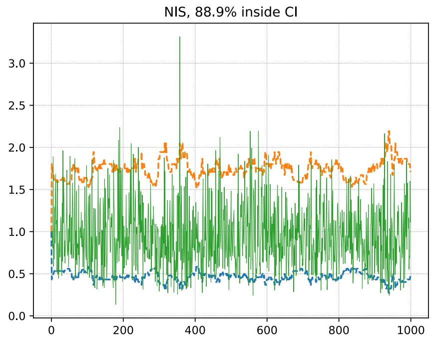
The initial tuning was improved in every given metric by changing to:

```
Q = np.diag([0.07, 0.07, 0.6 * np.pi / 180]) ** 2
R = np.diag([0.1, 0.8 * np.pi / 180]) ** 2
JCBBalphas = np.array([1e-7,1e-7])
```

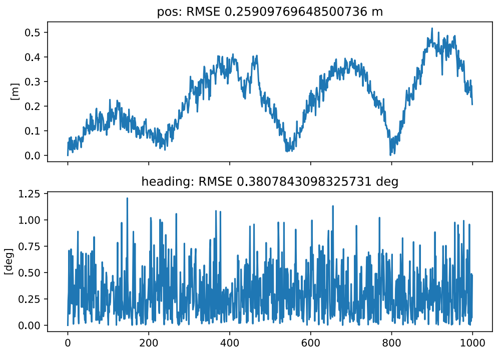
`JCBBalphas` was set to small values in order to overcome the problem with duplicates in tuning set 2. Q and R was slightly lowered because the experienced NEES and NIS of the initial parameters seemed to be quite low in the CI. This improved the consistency as seen in 3b and 3a. The position RMSE was improved the most, while the heading was difficult to improve, see 3c. The RMSE of position is lowest when the robot is traveling in an area it has been before, but increases when detecting new territory. This effect is potentially amplified by the low alphas, the filter may try to associate new landmarks with old ones instead of creating new ones. Why heading RSME was harder to improve will be further discussed in algorithm shortcomings.



(a) NEES



(b) NIS



(c) RMSE

Figure 3: Combined figure showing results, NIS, NEES, and RMSE.

Task 3: Run EKFSLAM on real data

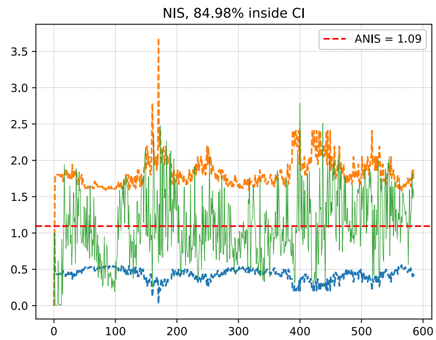
Initial parameters:

```
    sigmas = 0.025 * np.array([0.0001, 0.00005, 6 * np.pi / 180])
    R = np.diag([0.1, 1 * np.pi / 180]) ** 2
    JCBBalphas = np.array([0.00001, 1e-6])
```

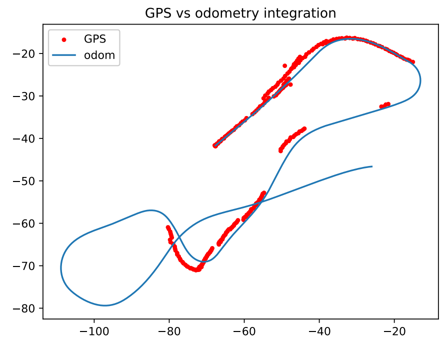
Tuning parameters 1

```
sigmas = 0.025 * np.array([0.0001, 0.00005, 6 * np.pi / 180])
R = np.diag([0.05, 0.5 * np.pi / 180]) ** 2
JCBBalphas = np.array([0.01, 1e-3])
```

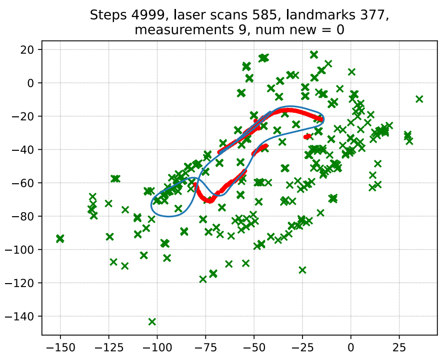
Since we run on real data, the closest we can get to ground truth is the GPS measurements. The problem with the Victoria Park dataset, is that the car drives through areas with dense vegetation, where GPS becomes unavailable. Thus, we base our tuning mostly on the NIS plot.
The NIS tended to be in the lower part of the CI, meaning that the filter was too conservative. We therefore set R lower so the filter would be more confident in the measurements. Additionally, increasing the alphas of JCBB proved to better the consistency. We got an artificially high amount of landmarks with several duplicates, but as will be mentioned in algorithm shortcomings, the NIS is not punished for this. See figure 5. The trajectory follows the GPS measurements pretty well in the first turn, but there is some deviation later on. An improvement strategy in reducing error could be to utilize these GPS measurements in Aiding Slam [2][2].



(a) NIS in 95% CI for good parameters



(b) GPS vs Odometry



(c) Trajectory (blue), GPS (red) and landmarks (green)

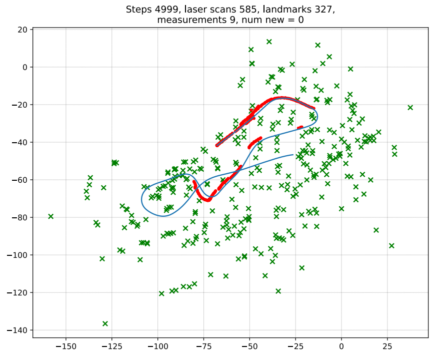
Figure 4: Results for good parameters

Tuning parameters 2

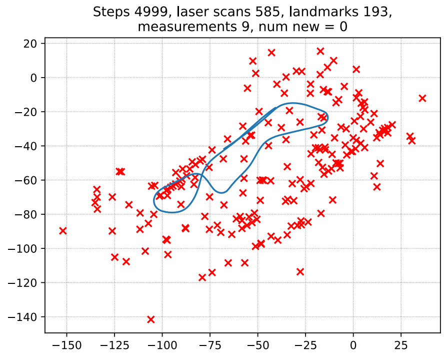
Figure 4b compares the evolution of odometry with the GPS signal, suggesting potential unreliability in the odometry data. By significantly reducing the process noise covariance Q as shown below:

```
sigmas = 0.00001 * np.array([0.0001, 0.00005, 6 * np.pi / 180])
```

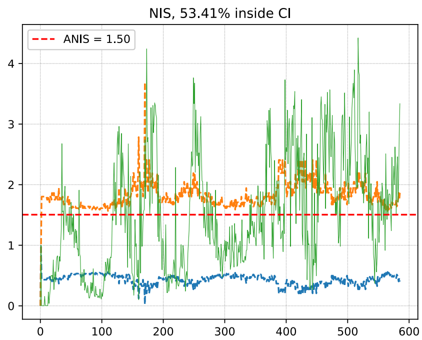
we increase the filters trust in the odometry measurements. As illustrated in figure 5a, this adjustment causes the estimated trajectory to deviate markedly from the GPS signal.



(a) Trajectory (blue), GPS (red) and landmarks (green) with significantly reduced R-matrix



(b) Trajectory (blue), and landmarks (red), initial parameters.



(c)

Figure 5: NIS with significantly reduced R-matrix

Consequently, new measurements of previously mapped landmarks cannot be associated, and new landmarks are generated. (see Figures 5a and 5b). Although the NIS within the CI decreased from 60.75 % to 53.41 %, the reduction is modest compared to the actual divergence of the filter. One reason for this is that NIS is not increased when new landmarks are added. Another is that with the amount of present landmarks, a measurement of a landmark will probably associate to an already nearby landmark, even though it is the completely wrong place.

Tuning parameters 3

The smallest number of landmarks we managed to achieve was 91. See figure 6. The parameters used was:

```
R = np.diag([1, 7 * np.pi / 180]) ** 2
JCBBalphas = np.array([1e-7, 1e-8])
```

Decreasing JCBBalpha resulted in more associations being made, leading to fewer landmarks in total.

The increase in bearing angle's standard deviation made the filter less confident, causing it to associate new landmarks as old ones since it did not trust the measurement.

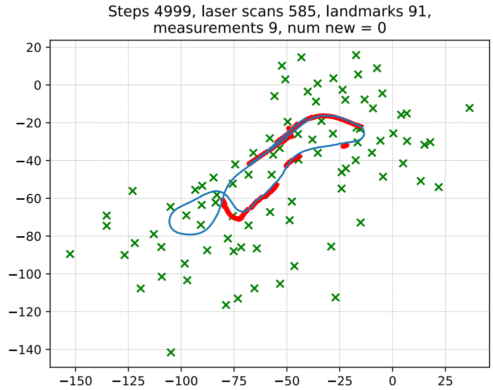


Figure 6: Few landmarks

Algorithm Shortcomings

The standard range-bearing SLAM problem has an unobservable subspace of 3-dimensions (absolute position and heading). [1, p. 211]. This means that a shift in both robot and landmarks, will cause no change in measurements [9][3]. However as shown in the observability analysis in [10][3], the EKF only has an unobservable subspace of 2-dimensions when linearizing around the most recent state estimate. This implies that the EKF attains information along an unobservable direction, i.e. absolute heading, which is baloney. This leads to inconsistency in the filter. A possible solution is to use the first estimates of the landmarks as linearization points.

Further, as experienced when using tuning parameters 2, the filter can be prone to generate duplicates in the internal map. For a given measurement, the innovation is only computed based on the difference from the associated landmark. The rest of the duplicates for that landmark will therefore not contribute in the update step, causing the filter *to not be penalized for having duplicates.*

Certain tuning parameters drastically increases runtime in JCBB association.

Another shortcoming is that recursive SLAM assumes that all landmarks remain stationary [1, p. 205]. In reality, moving objects like cars are detected as landmarks, causing the model to register the same car as multiple landmarks due to its changed position.

Bibliography

[1] Edmund Brekke. *Fundamentals of Sensor Fusion*. Edmund Brekke, 2024.

[2] Jose Guivant, Favio Masson and Eduardo Nebot. 'Simultaneous localization and map building using natural features and absolute information'. In: *Robotics and Autonomous Systems* 40 (Aug. 2002), pp. 79–90. DOI: 10.1016/S0921-8890(02)00233-6.

[3] Guoquan Huang, Anastasios Mourikis and Stergios Roumeliotis. 'Observability-based Rules for Designing Consistent EKF SLAM Estimators'. In: *I. J. Robotic Res.* 29 (Apr. 2010), pp. 502–528. DOI: 10.1177/0278364909353640.