Department of Engineering Cybernetics

Sensor fusion

Graded assignment 3

Group:
Sensor fusion Group 65

Authors:
Eivind Heldal Stray (eivindhs)
Magnus Dyre-Moe (magnudyr)

November 13, 2020

1 Introduction

SLAM stands for simultaneous localization and mapping and is known as a chicken-and-egg problem. This is because, initially, in order to make a map, a pose is required. To obtain the pose, a map is needed. Thus SLAM is regarded as a hard problem in robotics.

A possible solution to the SLAM problem is EKF-SLAM. Here one uses a combination of the SLAM method with a filter, namely the extended kalman filter. Tuning an EKF-SLAM is a difficult task. It requires more precise tuning compared to other target tracking methods such as the IMM-PDAF method. Hence, careful tuning is required to achieve a stable response. Bad tuning will result in divergence quickly. Through this paper we will investigate the different tuning parameters in EKF-SLAM. Furthermore, we will introduce potential difficulties one may encounter while working with SLAM. Also, we will discuss upon possible improvements to SLAM.

The latest code used in this project can be found at `https://github.com/eivindhstray/TTK4250.git`. This repository contains code and reports for all project in the course TTK4250.

1.1 Parameters and Tuning Strategy

When tuning the EKF-SLAM filter on data, there are a few physical assumption that have to be made. This will set a baseline for further tuning. The measurement covariance, in range and angle for each landmark observation is: $\mathbf{R} = \begin{bmatrix} \sigma_r & 0 \\ 0 & \sigma_\theta \end{bmatrix}$. Therefore, the σ values can be chosen according to assumptions regarding sensor measurements (this could be radars, lidars, lasers). As an initial assumption, let the standard deviation σ_r in range be $0.01m$ and the angular standard deviation σ_θ be 2 degrees for both of the datasets in question.

The next parameter is $\mathbf{Q}$, the process noise covariance on the form $\mathbf{Q} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{bmatrix}$ if $\mathbf{Q}$ is a diagonal matrix. The general formulation will be $\mathbf{Q} = \mathbf{ZAZ}^T$ where $\mathbf{A}$ contains correlation coefficients and $\mathbf{Z}^T = [\sigma_x, \sigma_y, \sigma_\psi]$. These values as well, should physically make sense. Assuming that the odometry of the system is fairly accurate, $\sigma_x = \sigma_y = 5 \cdot 10^{-3}m$ and $\sigma_\psi = 2$ degrees $\approx 0.017rad$ can arguably be reasonable for the two datasets. It is important to note that the magnitude of these values will depend on the timestep. For simulated data there is no timestep given, but there is for the Victoria Park Dataset that will be discussed in Section 3. Albeit quite random numbers, the magnitude of these should not be too far off for sensible NEES and NIS values. The aim is simply to achieve reasonable values to begin with.

The final parameter is α_{JCBB} and is related to gating measurements both for individual compatibility and joint compatibility. For individual compitability, $\alpha_{individual}$ acts as gate size for the individual measurements. Choosing this low allows more measurements to be gated (i.e setting it to 0 would let through all measurements). α_{joint} gates measurements by how well the joint probabilities overlap the landmarks in question. If this is set too high, too few measurements get gated forcing the filter to create more landmarks. This in turn leads to very many computations and the filter quickly gets slower and slower. Too high values for these parameters made the algorithm slow as it was accumulating too many landmarks mistaking existing ones for new. Since there is no clutter or false alarms in the simulated dataset, these can be set low without giving them too much thought. This is not necessarily the case for the Victoria Park dataset, however, due to clutter. In both datasets, these were set to the same values (see Table 1). Increasing these values led to a higher number of landmarks, resulting in too slow computations, especially in the Victoria Park dataset.

2 Run EKFSLAM on simulated data

2.1 The Simulated dataset

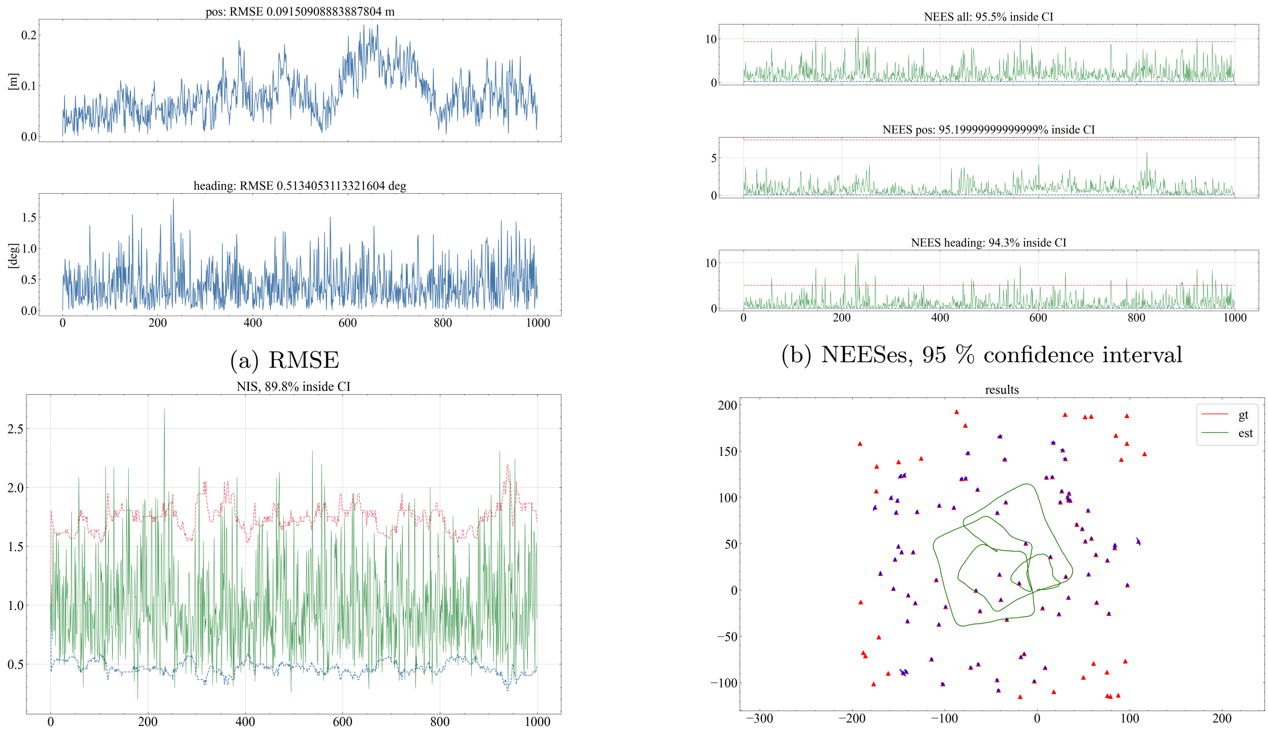
The simulated dataset contains 10.000 time steps. There is no clutter or false alarms, meaning that landmarks are actual landmarks. Furthermore, the odometry is quite accurate. This is can be verified by running the simulated dataset without doing associations but no plot of this will be displayed in the report due to limitations in pages. These identities make the simulated dataset quite robust to tuning parameters compared to the Victoria Park Dataset in section 3.

2.2 Tuning and simulation of Simulated Data

The results of the initial guesses were not bad. NEESes and NIS were acceptably within the 95% confidence interval, and the RMSE for position and angle in pose were approximately $0.2m$ and $0.5degrees$, respectively. By tuning the parameters around this area, the resulting performance is given in Figure 1. The parameters were as in Table 1. Note that the average NEES (ANEES) is on average quite low, while the average NIS (ANIS) is well within the 95% confidence interval. It seems that the position RMSE is generally quite low while the heading varies more. This would impact distant measurements providing a higher NIS than NEES, as deviation in measurements increases with range. The reason this is left unchanged is that these were the best results in RMSE seen together with ANEES and ANIS. Furthermore, the NEES and NIS plots are all satisfactory.

σ_r	sigma$_\theta$	σ_{xandy}	σ_ψ	α_{joint}	$\alpha_{individual}$	ANIS	NIS CI	ANEES	NEES CI
0.06	0.02	0.03	0.00607	10^{-4}	10^{-6}	0.984	[0.465,1.742]	1.99	[2.85,3.15]

Table 1: Parameter values for Simulated data, all confidence intervals (CI) are 95% CI



(a) RMSE

(b) NEESes, 95 % confidence interval

(c) NIS, 95 % confidence interval

(d) Trajectories and landmarks

Figure 1: Simulated results

3 Run EKFSLAM on the *Victoria Park* dataset

3.1 Tuning and considerations about the dataset

General information about the Victoria Park dataset can be found on the Victoria Park dataset website [VP, 2006]. It contains around 64.000 timesteps and uses lasers to measure landmarks in the viscinity of a car navigating through Victoria Park in Sydney. The dataset contains both clutter and false alarms, as well as varying process noise as the car maneuvers through the park. (See Figure 2d for a satellite image of the park).

Similar values to the ones used in the simulated dataset were initially used for Victoria Park dataset. It was apparent that to many landmarks were created – duplicating previously seen landmarks. Duplicating landmarks results in a complex **P**-matrix, cumbersome to invert. As the **P**-matrix and list of landmarks grows, each iteration takes longer and longer time. For the initial tuning, the computational complexity was high and the filter diverged from the GPS measurements quite quickly.

A physical interpretation of this can be viewed as collections of trees. It is better to see collections of features as fewer single landmarks than each tree being considered several individual landmarks. A way of achieving this can be to increase the magnitude of **Q**, conceptually allowing a filter to perceive larger collections of features as one by increasing the area where a landmark may exist relative to the vehicle. By examining Victoria Park in Figure 2d, it is clear that there are many bunches of trees, and computationally it would make sense to allow close-standing trees as single features. The elevation of values in the **Q**-matrix did lead to fewer landmarks being registred throughout the simulation.

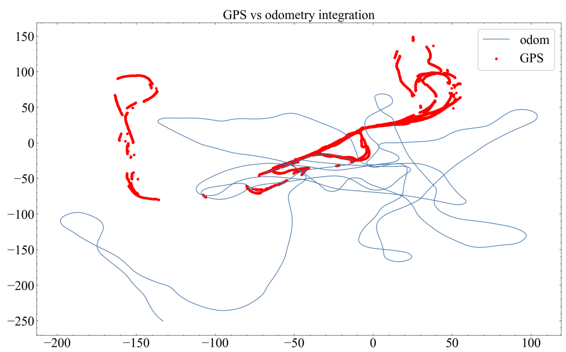
Tuning from the initial values was done by trial and error. The parameters, average NIS and the confidence interval for the average NIS can be found in table 2, and plots for these values over 45.000 iterations can be found in Figure 2.

The average NIS (ANIS) is quite low, with only $\tilde{6}0$ % within the 95% confidence region. Yet, getting the ANIS higher proved to be very difficult, and the results are after all quite pleasing, as the car remains on or close to track (Figure 2b) with a simulation time of less than 30 minutes for the 45.000 iterations. Experience show that this can in fact be hours with badly tuned parameters.

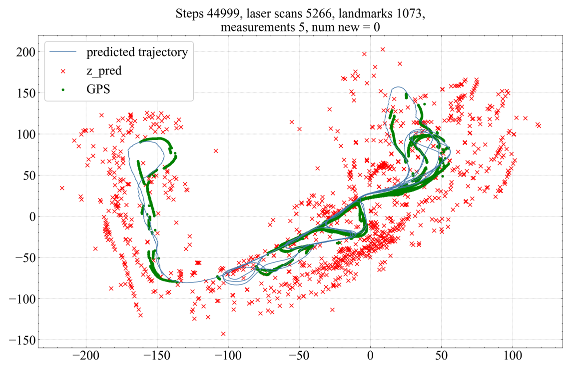
When running EKF-SLAM with these parameters, after about 30.000 iterations the speed of the algorithm was down to around 10 iterations per second. This is too slow, as the laser is sampling at between 0.005 and 0.02 seconds. Once the algorithm is too slow for the sampling frequency, applying it in real time becomes worthless as it is lagging behind the car.

σ_r	sigma$_\theta$	σ_{xandy}	σ_ψ	α_{joint}	$\alpha_{individual}$	Average NIS	95% CI
0.06	0.02	0.0346	0.02	10^{-4}	10^{-6}	0.797	[1.980,2.020]

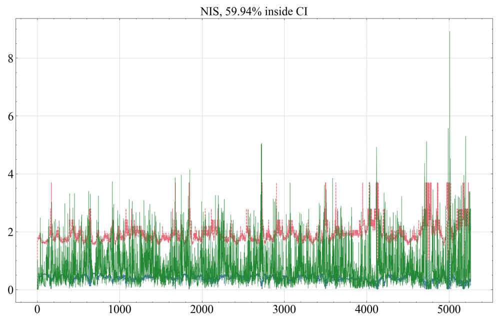
Table 2: Parameter values for the Victoria Park Dataset



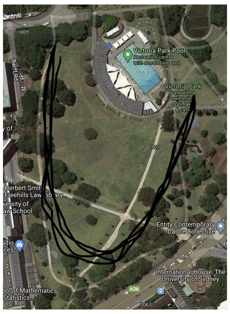
(a) Predicted without correction and GPS for the entire dataset



(b) Trajectory and Landmarks



(c) NIS, 95% confidence interval



(d) Victoria Park and sketched area of maneuvering[1]

Figure 2: Running 45000 iterations with final tuning values

4 Reflection on the results and EKF-SLAM

4.1 Reflection

The EKF-SLAM filter has a few obvious weaknesses, the first being growing computation time for each timestep. As new landmarks are introduced, more computation needs to be done both for gating, as well as a growing $\mathbf{P}-matrix$. While the algorithm is fast to begin with, it gets very slow when processing the Victoria Park dataset. Furthermore, as only landmarks are used for correction, a deviation from ground truth may never be recovered causing permanent deviation due to confusion about old landmarks. There is no finite amount of landmarks, and thus the complexity has no theoretical upper bound.

4.2 Thoughts on improvements

Even human orientation by means of landmarks is difficult, and orientation in the landscape - even with a map - can be misleading. SLAM filtering is a hard problem, due to data association. While using EKF-SLAM on the Victoria park dataset this becomes prevalent. When the number of landmarks increases, data association becomes increasingly

[1]Picture taken from google maps

cumbersome. When the update step eventually is slower than the movement of the robot, the track diverges. Hence, in order to make SLAM available in real time applications the predictions will have to made quickly. Therefore, considering only nearby landmarks could make the algorithm faster.

When discovering old landmarks, the implemented EKF-SLAM filter seems to often add them as new landmarks and thus the old ones are sometimes duplicated, taking up both memory and computation. Another possible solution to the increasing number of landmarks could be clustering. Clustering would group certain landmarks into a region with an centroid. This could make data association faster. However, this could possibly lead to difficulties when discovering new landmarks within the region of the cluster.

Another research topic of SLAM is loop closure. Loop closure is recognizing when revisiting a place. This happens in both of the discussed simulations in section 2 and 3 as the vehicle revisits old landmarks. It is performed in a better way in Simulated data than in Victoria Park, which is to be expected. Without loop closure landmarks may be reintroduced to the map. For the Victoria dataset this means that every landmark could potentially be mapped several times. When running the EKF-SLAM algorithm on the dataset we observe that several landmarks gets mapped multiple times. This adds computational complexity due to the covariance matrix, **P**, which increases in size when adding new landmarks. For a dense **P** matrix with correlating terms this is a problem. In practice this means that the SLAM method gets increasingly more complex for every landmark observed, essentially making the algorithm unscalable. Adding loop closure and the potential of deleting certain landmarks or segmenting them such that only local landmarks within a certain radius are considered could solve this problem. However, loop closure is an active research field.

Noisy odometry is another concern. It introduces uncertainly both in measuring landmarks and predicting motion based on the odometry. Therefore, a precise process state space is essential for these kind of SLAM filters to work. This is a greater issue in the Victoria Park dataset than the simulated, as the simulated is seemingly designed to contain quite accurate odometry.

Lacplace approximations is also a potential way of improving the model. Here, the **P**-matrix becomes redundant as the hessian of the process model $f(\eta_k)$ is used in its place. By using the hessian matrix it is possible to calculate the posterior through Newtons method. However, the hessian matrix may be negative definite. This may lead to divergence. Hence, the Laplace approximation should only be used when the hessian is positive definite. Additionally, the hessian matrix can be cumbersome to calculate analytically. Fortunately, there exist several numerical approaches to calculate the hessian. Nevertheless, substituting the **P**-matrix with a numerical approach to the hessian will make the algorithm faster because we remove the cumbersome computations when inverting the large **P**-matrix.

5 Conclusion

Overall, SLAM is a difficult topic which requires very precise tuning. However, when the tuning is sufficiently good it can perform well. The problem of increasing number of landmarks is still a problem that requires cumbersome computations and may make the track diverge. There exists several different approaches to partly solve this issue, for example Laplace approximation. However, one of the major issues with SLAM today is the loop closure problem, i.e. recognizing when revisiting a location. Solving the loop closure problem may solve the main difficulties to SLAM and make it more applicable to real life systems. However, as loop closure is still an active research topic, an efficient method has not yet been discovered.

Bibliography

Victoria Park, 2006. URL `http://www-personal.acfr.usyd.edu.au/nebot/victoria_park.htm`.