Plateforms and Algorithms for Autonomous Driving-Module 1

3. Particle Filter

Asif Khan Pattan

Masters in Artificial Intelligence

University of Bologna

August 29, 2023

Objective

The objective of the assignment is to find out the forklift's position in a warehouse.

Solution

Initialization Noise

Initialization noise **(sigma_init)** is a parameter used during the initialization of particles. It adds randomness to the initial position and orientation of particles to represent uncertainty in the initial estimate.
sigma_init is an array with three values: [sigma_x, sigma_y, sigma_theta]. The values represent the standard deviations of the Gaussian noise to be added to the initial x, y, and theta (orientation) values, respectively. Larger values of sigma_init will result in more spread-out initial particles, reflecting higher uncertainty in the initial estimate.

Movement Noise

Movement noise **(sigma_pos)** is used in the prediction step to account for uncertainty in the vehicle's motion. The prediction step calculates the new position based on the vehicle's velocity and yaw rate. Movement noise is added to the predicted position to represent the uncertainty introduced by factors like wheel slippage, imperfect control, etc.
Similar to sigma_init, sigma_pos is an array with three values: [sigma_x, sigma_y, sigma_theta]. Larger values of sigma_pos will result in more spread-out particles after the prediction step, reflecting higher uncertainty in the vehicle's motion.

Measurement Noise

Measurement noise **(sigma_landmark)** is used in the update step to account for sensor measurement uncertainty. It represents the uncertainty in the measurements obtained from the LiDAR sensor.
sigma_landmark is an array with two values: [sigma_x, sigma_y]. These values represent the standard deviations of the Gaussian noise to be added to the observed x and y values of landmarks. Larger values of sigma_landmark will result in wider probability distributions for the likelihood calculation, accommodating more measurement uncertainty.

Number of Particles

The number of particles **(NPARTICLES)** determines the granularity of the particle filter's estimation. More particles allow for a finer representation of the state space, providing a more accurate estimate.
However, a larger number of particles also require more computation and can slow down the algorithm. A balance must be struck between accuracy and computational efficiency.

Results

1. When Measurement Noise is High:

Configuration:
sigma_init [3] = 0.04, 0.04, 0.4;
sigma_pos [3] = 0.05, 0.05, 0.05;
sigma_landmark [2] = 0.5, 0.5;

When measurement noise is high in a particle filter:

1. Particle weights become less informative and similar.
2. Particles spread widely, causing pose estimates to be more uncertain as shown in figure 1
3. Localization accuracy decreases and convergence is slower.
4. Computational demands may increase due to more particles needed and time required to execute at each point also increase (as seen in plot 3 the amplitude of the plot is closer to 0.02 and some peaks reaching 0.03) though not significantly.

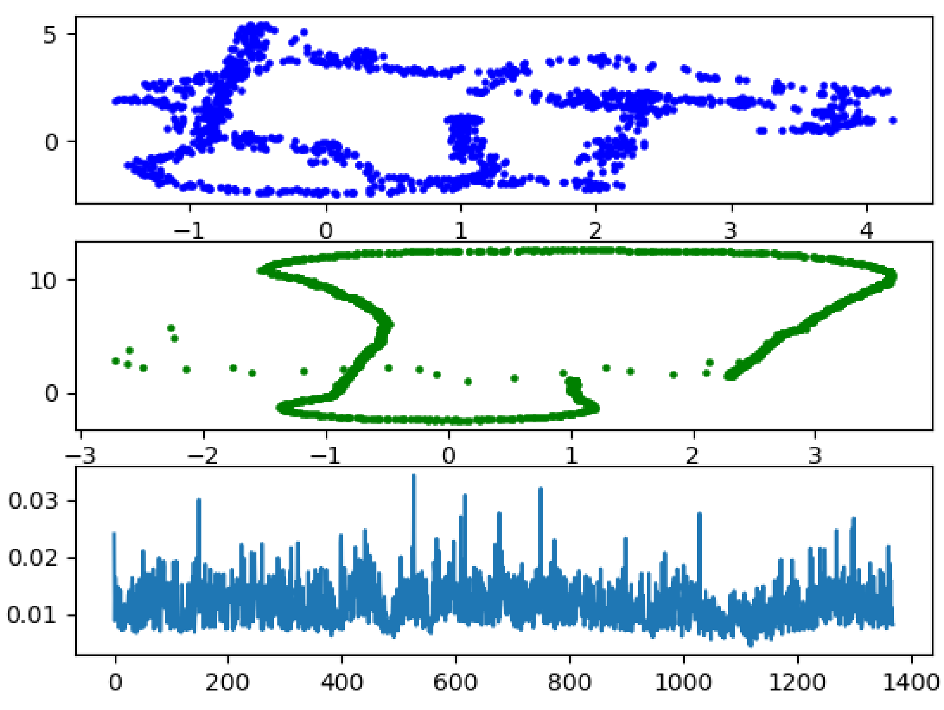

Figure 1: High measurement Noise

2. When Number of Particles is less:

Configuration:
NUMPARTICLES = 50;
sigma_init [3] = 0.04, 0.04, 0.4;
sigma_pos [3] = 0.05, 0.05, 0.05;
sigma_landmark [2] = 0.2, 0.2;

When the number of particles is low in a particle filter:

1. Accuracy decreases due to less representation.
2. Exploration of possible states is limited as shown in figure 2
3. Sampling errors and unreliable resampling occur. Noise has a larger impact on results.
4. But time taken at each step is less because of less particles to work with.(as seen in plot average amplitude is close to 0.015)

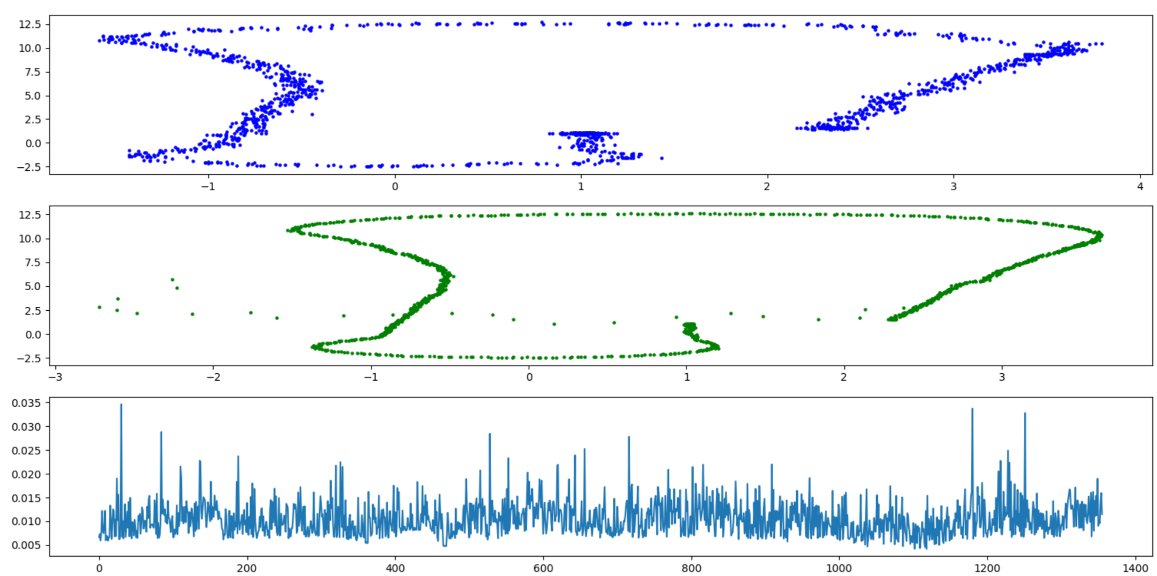

Figure 2: Less particles

3. Better output:

Configuration:
NUMPARTICLES = 300;
sigma_init [3] = 0.04, 0.04, 0.01;
sigma_pos [3] = 0.03, 0.03, 0.01;

sigma_landmark [2] = 0.2, 0.2;

With the above configuration we can see:

1. Accurate state estimation because of tuned initialization
2. Effective exploration of possibilities due to more number of particles present
3. Robustness to noise and faster convergence to accurate estimates as shown in figure 3

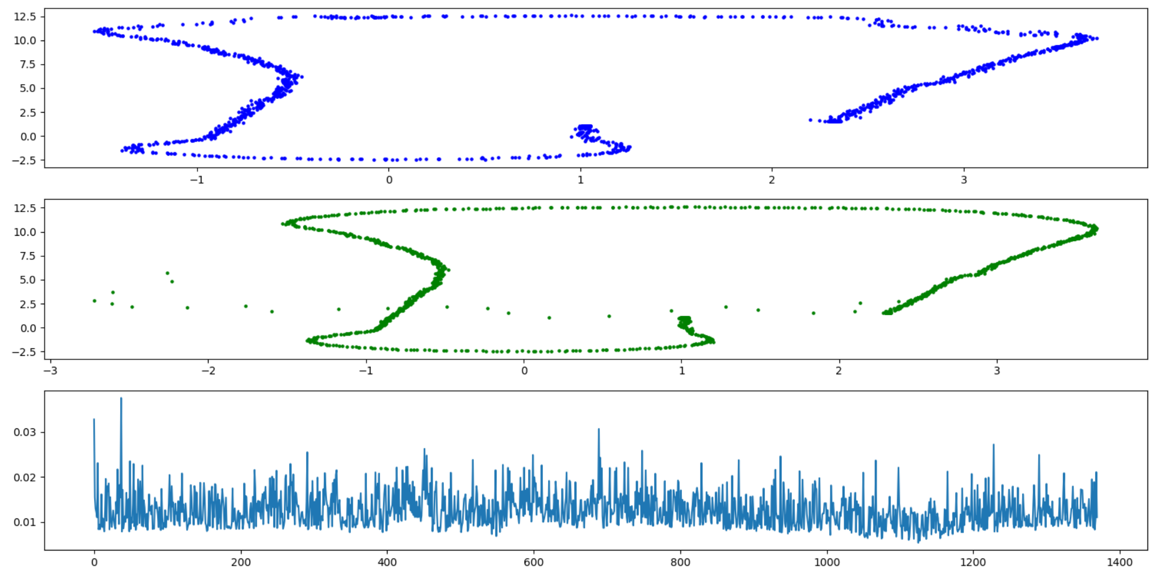

Figure 3: Better results

4. Best configuration:

Configuration:
NUMPARTICLES = 1000;
sigma_init [3] = 0.04, 0.04, 0.01;
sigma_pos [3] = 0.05, 0.05, 0.02;
sigma_landmark [2] = 0.2, 0.2;

This is the best configuration that is obtained after fine tuning the parameters according to the model and input data.
As shown in figure 4, this resulted in a much better estimations compared to the previous configuration due to more number of particles and better initial values of noise also keeping the time taken during each step is lesser than the previous configuration.

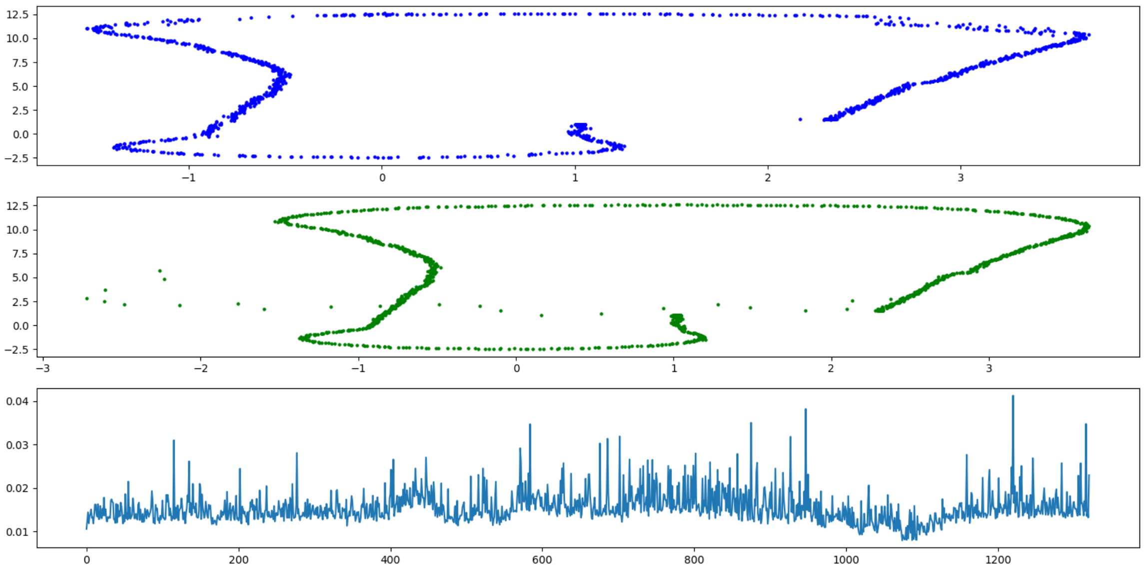

Figure 4: High measurement Noise

5. When Movement Noise is higher:

Configuration:
NUMPARTICLES = 100;
sigma_init [3] = 0.04, 0.04, 0.01;
sigma_pos [3] = 0.02, 0.02, 0.02;
sigma_landmark [2] = 0.2, 0.2;

When the movement noise is set very low, particle filter exploration become limited due to the narrow spread of particles, potentially causing the algorithm to miss true pose possibilites. Incomplete first plot as shown in figure 5
Overconfident predictions, Underestimated uncertainty occure when the movement noise is not properly set.
While the average time taken per step is same compared to other configurations, the time taken at the end is significantly higher due to lost path and estimations do not correspond to the data.

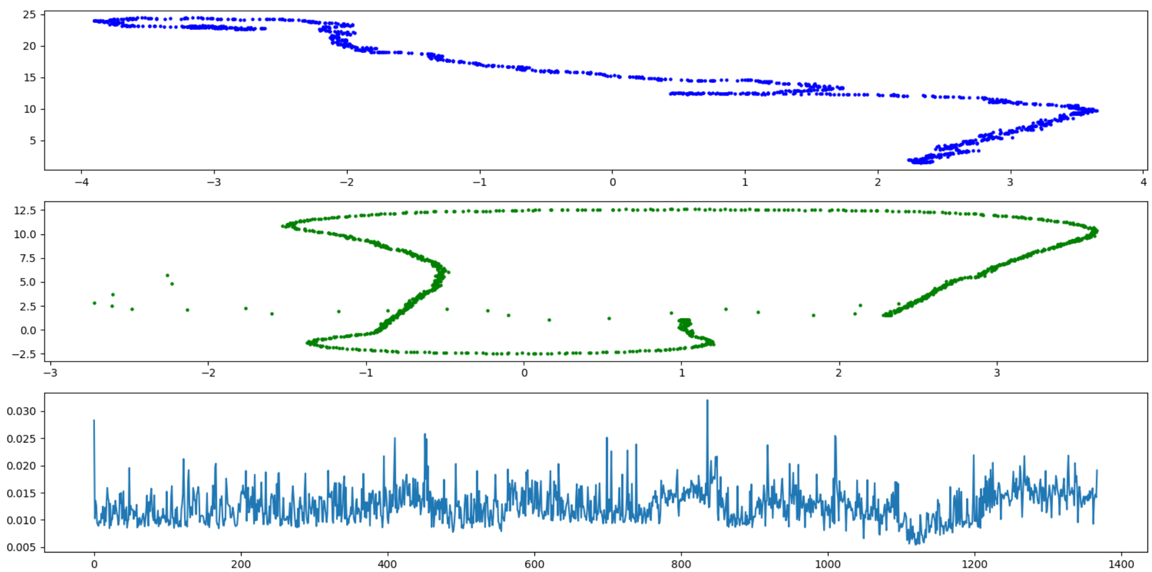

Figure 5: Low movement noise