ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

Assignment 4

Team-7

EE P 545 A - The Self Driving Car: Introduction to AI for Mobile Robots

November 22, 2023

Motion Model (20 Points)

Q1. Run the code that visualizes your motion model three times. For each run, choose a test speed, steering angle, and time interval that result in different robot motions. Submit the visualizations for all three runs.

A1. When we visualized the Motion Model with the default control parameters given, we get the following output :

The following control parameters are used speed (s), steering angle (δ), time-step length (t)

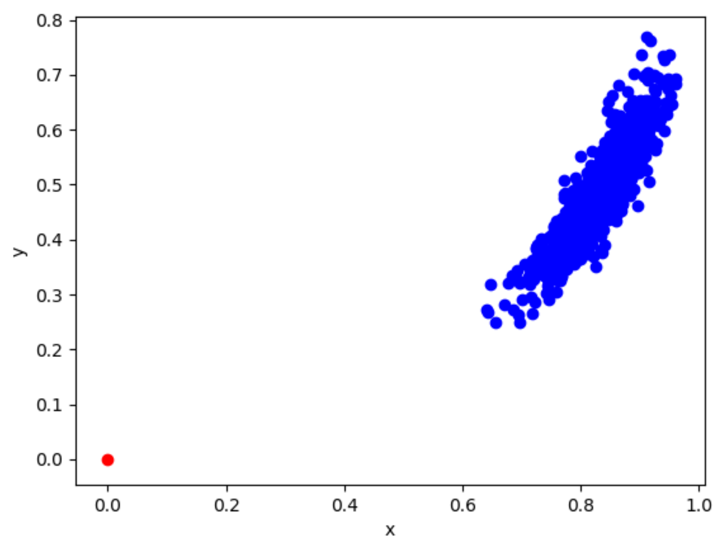
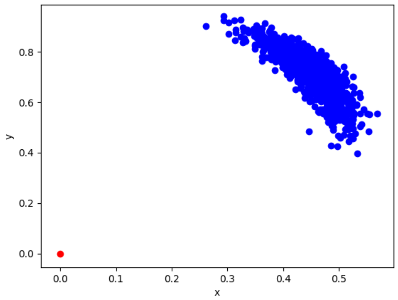
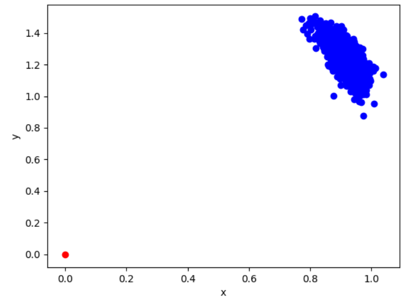


Figure 1: Motion model visualization on default parameters : s = 1; δ = 0.34; t = 1

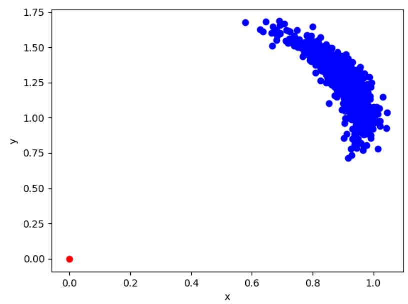
Now, to observe the nature of robot motion using different control parameters, we have visualized the MotionModel under 3 different sets of control parameters. In each set, we have given high weightage to the one of the parameters, while allocating less weightage to other, to see the impact of each control parameter on Robot motion.



(a) Effect of steering angle (δ) `s=1;`δ`=0.64;t=1`



(b) Effect of speed (s) `s=1.8;`δ`=0.34;t=1`



(c) Effect of time-step t `s=1;`δ`=0.34;t=1.8`

Figure 2: Effect of control parameters like s, δ & t on Robot Motion

- We see that when we increased the steering angle (δ), we see the particles have turned/steered more to the left by looking at the x, y pose of the robot. *See Figure 2.(a)*
- We see that when we increased the speed (s), we see the particles have moved more to the left but have not steered much ;by looking at the x, y pose of the robot. Secondly, the particles are clumped together. *See Figure 2.(b)*
- We see that when we increased the time-step (t), we see the particles are more dispersed and more farther by the to the left;by looking at the x, y pose of the robot. This sparse nature corresponds to a more curved nature of the particle cluster. This not necessarily mean the particles are more steered, we are observing a longer path taken by the robot due to the increase in time-step. *See Figure 2.(c)*

Q2. How did you initially choose the noise parameter values?

A2. We first need to visualize how the noise on speed (s_n) and steering angle (δ_n) influences the robot motion. Let's introduce a minimal noise of 0.05 to both the variables and check its visualization. Hence, we chose the initial parameters as:

- $s_n = 0.005$
- $\delta_n = 0.005$

the robot should complete its motion without any deviation.

Let's check the following nature :

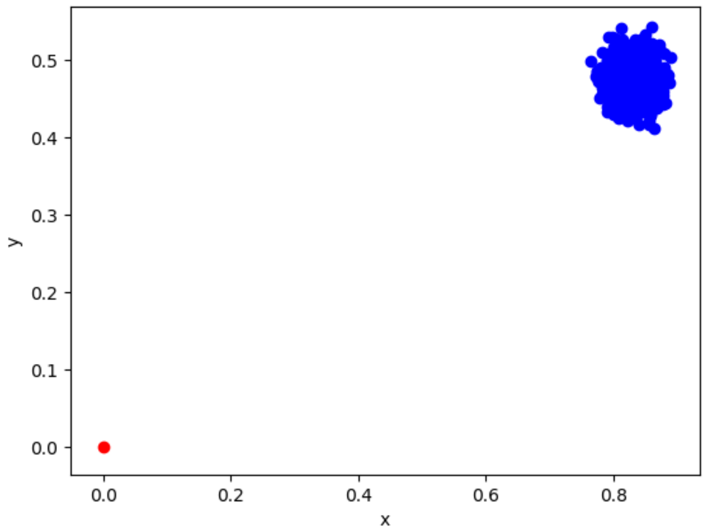


Figure 3: Motion model visualization on default parameters : $s_n = 0.005$; $\delta_n = 0.005$

Looking at *Figure 3*, we have confirmed our hypothesis. The robot steered and traveled a distance, according to the motion equation and the particles are clumped to a region.

Q3. After picking the initial values, did you tune them? If yes, how so?

A3. In order for our Motion model to mimic the behaviour given in the Assignment 4 Question PDF, we shall slowly increase each parameter and see its effect on the Robot Motion. The expected behaviour given in the Assignment 4 PDF is shown in *Figure 4*.

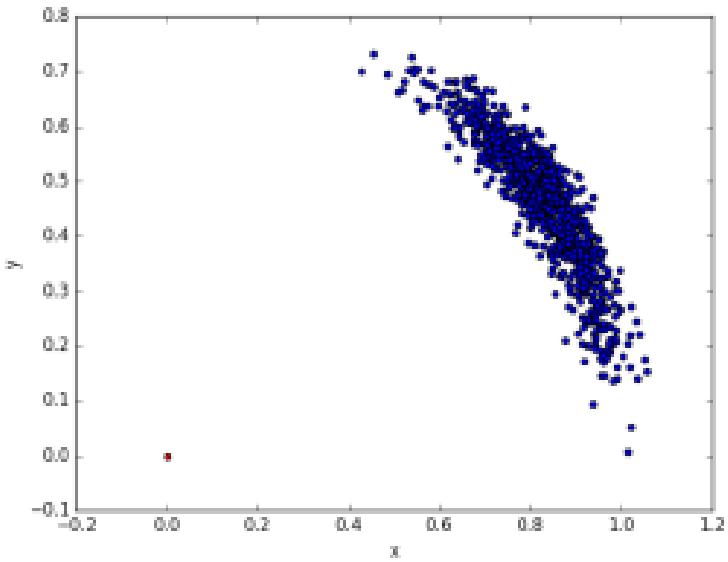
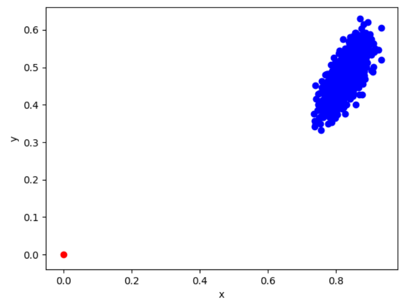
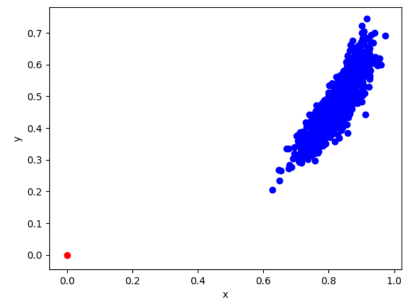


Figure 4: Expected Motion model behaviour we are trying to mimic

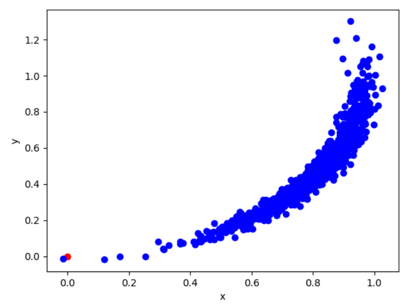
At first, Let's see the effect of noise on the speed of the robot (s_n). To check this out, we shall be keeping the noise on steering angle ($\delta_n = 0.005$) as minimal and constant, and slowly increasing s_n.



(a) s_n=`0.05`;δ_n=`0.005`



(b) s_n=`0.1`;δ_n=`0.005`

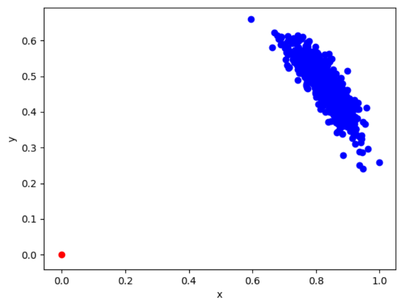


(c) s_n=`0.25`;δ_n=`0.005`

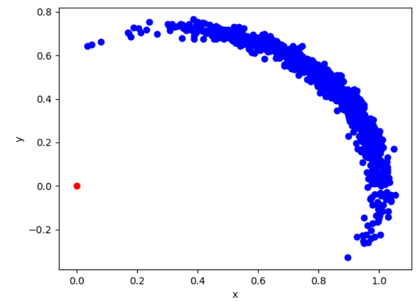
Figure 5: Effect of noise on the speed of the robot (s_n) affecting Robot motion

When we introduced noise in the speed of the robot, we see the sparseness of the particles along the trajectory that the robot will take when it is steered and moved in a angle.

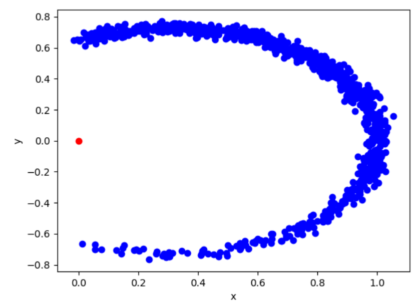
Secondly, Let's see the effect of noise on the steering angle (δ_n). To check this out, we shall be keeping the noise on steering angle ($s_n = 0.005$) as minimal and constant, and slowly increasing δ_n.



(a) s_n=`0.005`;δ_n=`0.05`



(b) s_n=`0.005`;δ_n=`0.2`



(c) s_n=`0.005`;δ_n=`0.5`

Figure 6: Effect of noise on the steering angle of the robot (δ_n) affecting Robot motion

When we introduced noise in the steering angle of the robot, we see that the steering angle becomes well distributed such that the particles steer in different steering angles. Sometimes, the noise makes the net steering angle negative such that the some of the particles of the robot turn right instead of left. *(See Figure 6.c)*

Looking *Figure 5* & *Figure 6*, we find that, in order to mimic *Figure 4*, we need to add noise in the steering angle but keep less noise in the speed of the robot. On following our observations, we arrived at our final parameter set. The final parameter set's visualization is promising. The final values for these parameters are given in answer of Q4.

Q4. What were your final values for these parameters?

A4. The final parameters are as follows :

- $s_n = 0.015$
- $\delta_n = 0.125$
- $s = 1$
- $\delta = 0.34$
- $t = 1$

The visualization for these parameters is as follows:

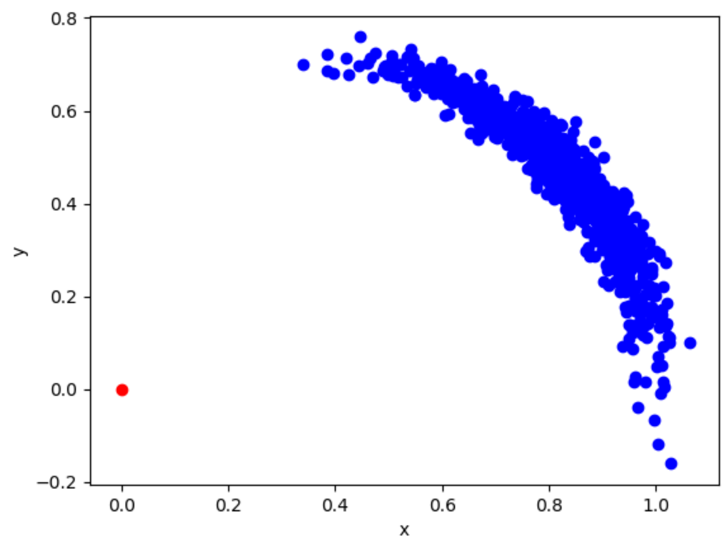


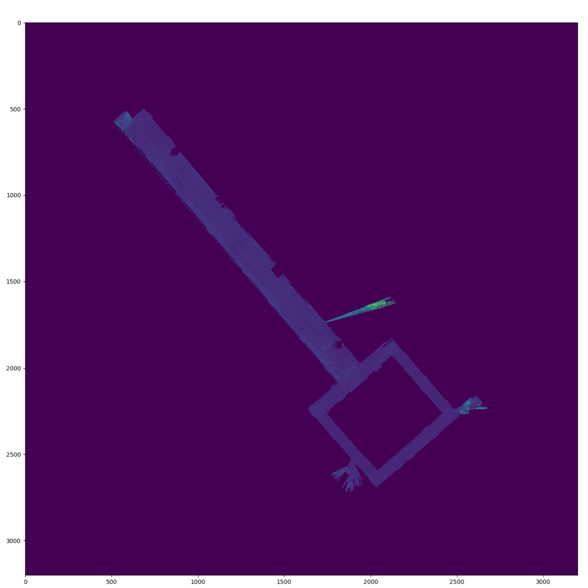
Figure 7: Visualization of Chosen Final Parameter values.

Sensor Model (20 Points)

Q5. Run the code that visualizes your sensor model on each of the three provided laser scan bags. Submit the visualizations for all three runs.

A5. Insert answer here.

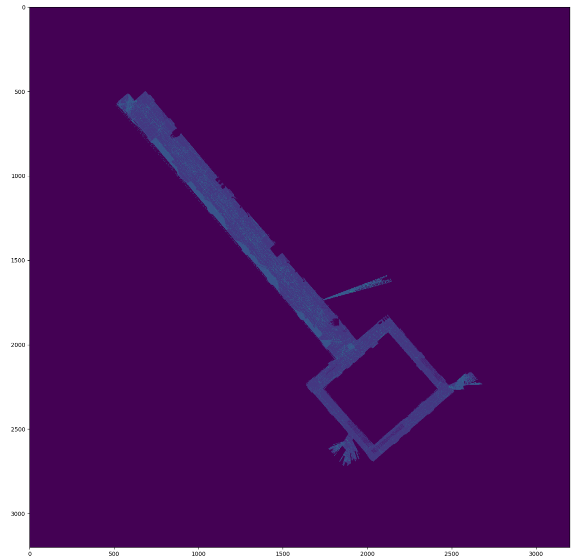
The visualizations for the 3 bag files are given below :



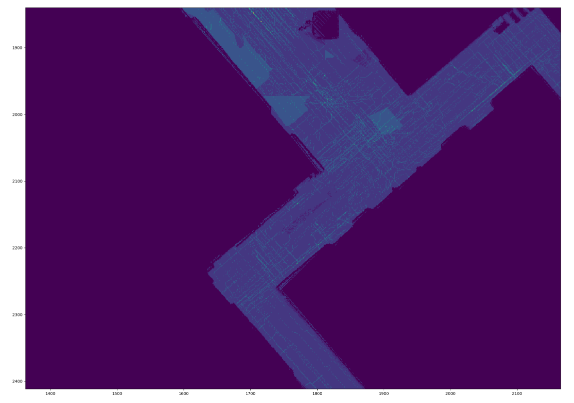
(a) Normal size image

(b) Zoomed image

Figure 8: `laser_scan1.bag`

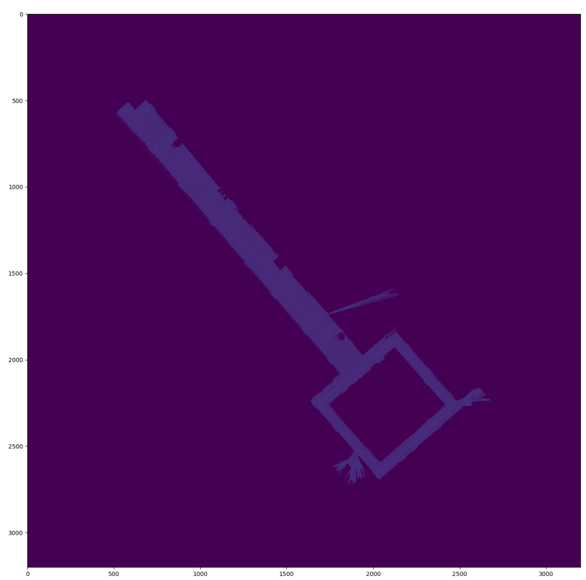


(a) Normal size image



(b) Zoomed image

Figure 9: `laser_scan2.bag`



(a) Normal size image

(b) Zoomed image

Figure 10: `laser_scan3.bag`

Q6. How did you initially chose the mixing weights? How did you initially choose the value of SIGMA HIT?

A6. The mixing weights include :

- `Z_HIT` = 0.25
- `Z_SHORT` = 0.25

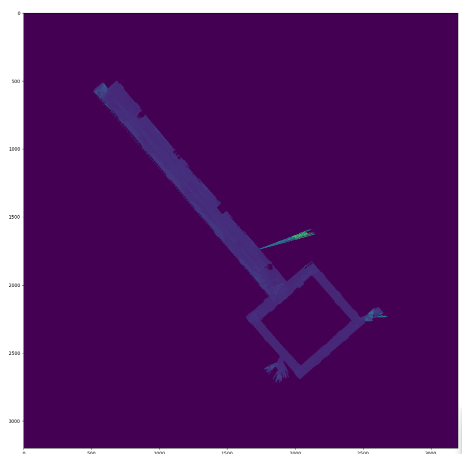
- Z_MAX = 0.25
- Z_RAND = 0.25

As all the mixing weights should add upto 1. We first decided to give equal weightage to all of the mixing weights. This balanced starting point allowed us to observe the overall impact before fine-tuning.

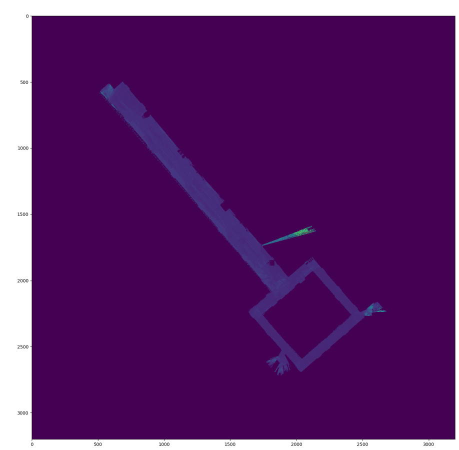
For SIGMA_HIT, we chose an initial small value (0.05). This decision aimed to focus on the distribution filtering out most of the noise, providing a direct observation of its impact on the sensor model.

Q7. After picking the initial values, did you tune them? If yes, how so?

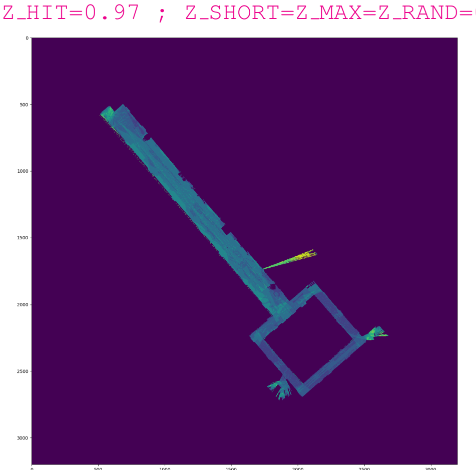
A7. **Mixing Weights** : After observing the generated heatmaps for each bag, we engaged in a tuning process. The tuning process is visualized on the `laser_scan1.bag`. We gave more weightage to each of the mixing weights while decreasing the others to check the impact of each mixing weight *(see provided images below)*.



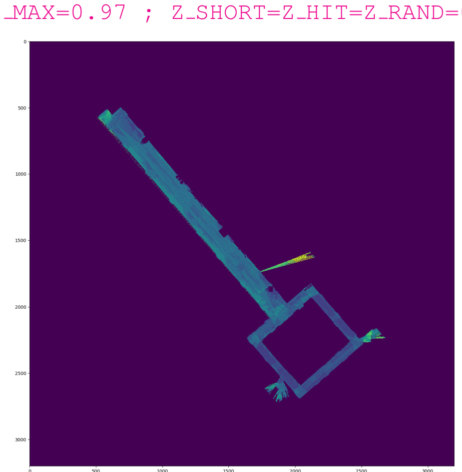
(a) `Z_HIT=0.97 ; Z_SHORT=Z_MAX=Z_RAND=0.01`



(b) `Z_MAX=0.97 ; Z_SHORT=Z_HIT=Z_RAND=0.01`



(c) `Z_RAND=0.97 ; Z_SHORT=Z_MAX=Z_HIT=0.01`



(d) `Z_SHORT=0.97 ; Z_HIT=Z_MAX=Z_RAND=0.01`

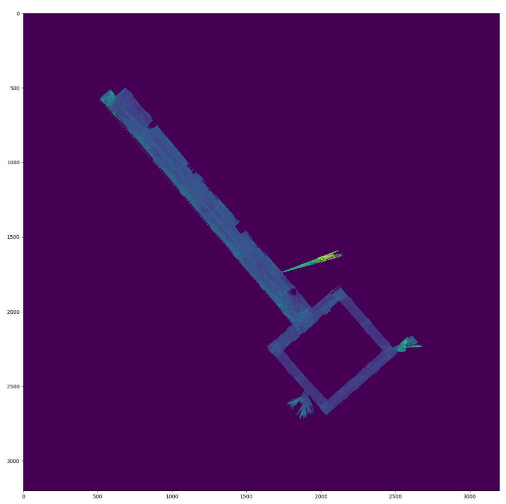
Figure 11: Effect of SIGMA_HIT on Sensor Model

We see that when we allocated more weightage to Z_RAND & Z_SHORT *(see Figure 11.(c) & Figure 11.(d) respy.)*, we see that are more noise dispersed in the heat map; which is not good!

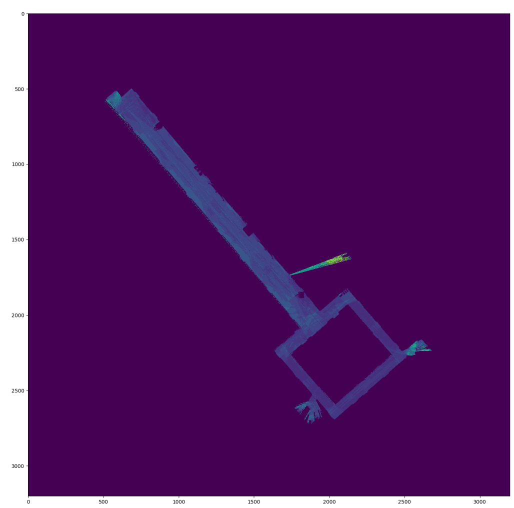
But, when we allocated more weightage to Z_HIT & Z_MAX *(see Figure 11.(a) & Figure 11. (b) respy.)*, we see all the noises generated by the Sensor Model are almost completely dissipated. This is not good because we need some noise information.

Ultimately, we decided to decrease the value of Z_HIT to allow for some noise generation, while slightly increasing other parameters.

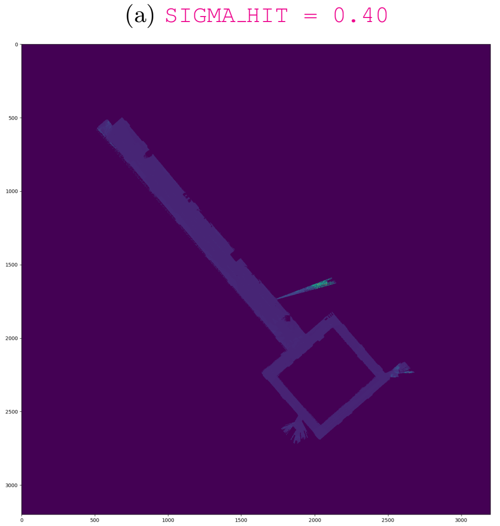
Sigma Hit : We explored the impact of different SIGMA_HIT values on the spread of the sensor model. The goal was to strike a balance that captured measurement uncertainties without making the model overly peaked or diffuse. This can be viewed below on the following plots :



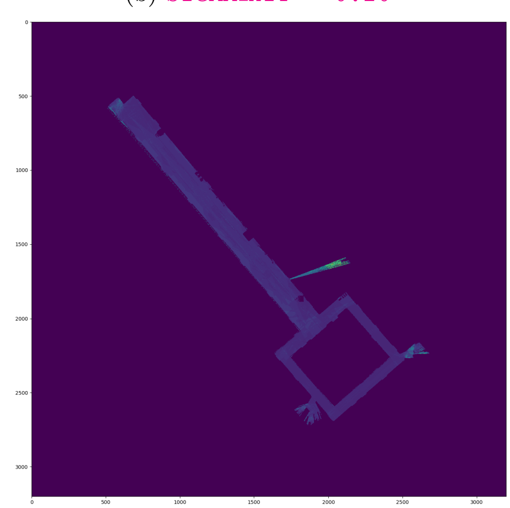
(a) SIGMA_HIT = 0.40



(b) SIGMA_HIT = 0.20



(c) SIGMA_HIT = 0.01



(d) SIGMA_HIT = 0.05

Figure 12: Effect of SIGMA_HIT on Sensor Model

For SIGMA_HIT we first tried to increase the value to 0.4. With the SIGMA_HIT value so high, we see a lot more noise in the images *(see Figure 12-(a))*. Next, we decided to decrease the value to 0.2. As you can see in the image there is still lots of noise *(see Figure 12-(b))*. Finally, we decided to go below our initial value of 0.05 and test a SIGMA_HIT value of 0.01. A extremely low value of SIGMA_HIT cut out almost all the noise in the image, which is not desired *(see Figure 12-(c))*. Therefore, we stuck with our normal SIGMA_HIT value of 0.05, which accurately displayed the desired information *(see Figure 12-(d))*.

The final values chosen for the given paramenters based on our experimentation is given in the Answer for Q8.

Q8. What were your final values for these parameters?

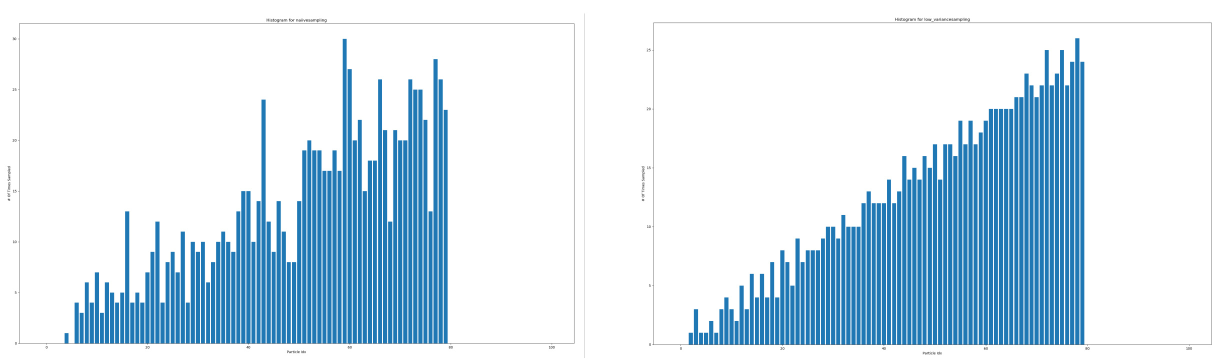
A8. The final values of the parameters that we chose after tuning are :

- Z_HIT = 0.80
- Z_SHORT = 0.01
- Z_MAX = 0.17
- Z_RAND = 0.01
- SIGMA_HIT = 0.05

Re-Sampler (20 Points)

Q9. Run the code that visualizes the re-sampler with the trials parameter set to ten for both methods. Submit both visualizations.

A9. The visualizations for the 2 Re-sampling types are given below :



(a) Naiive Resampling with `trial = 10`

(b) Low-Variance Resampling with `trial = 10`

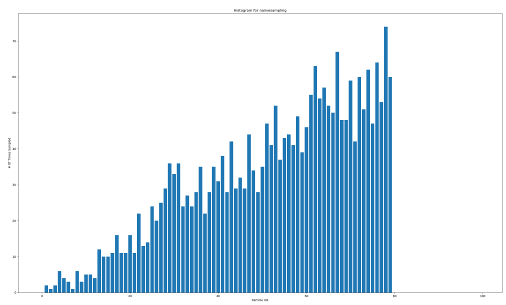
Figure 13: Re-Sampling Visualizations

Q10. Which of the two methods represents the true particle distribution better? How so?

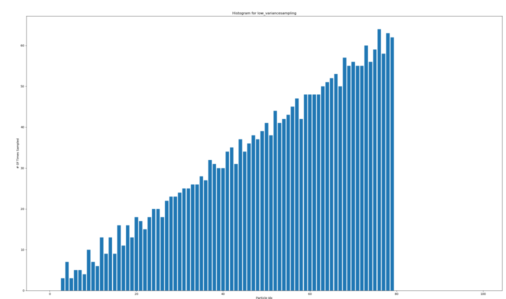
A10. From Q9, we see that **Low-Variance Re-Sampling represents the particle distribution better**. Because, in our visualization and our implementation, we see that at Particle index : 80 is where the maximum number of sampling occurs, telling us the Particle idx : 80 represents true position of the robot exhibited by the dummy values. Low-Variance represents the distribution better and the variance of the number of times different particles being sampled is low. This can be seen on how **smooth the Histogram is for the Low-Variance ReSampling in comparison to the Naiive ReSampling**. In Naiive ReSampling, the distrbution is more chaotic even though we achieve similar results. This **chaotic nature in Naiive Resampling tells the variance in the number of times different particles being sampled is high which is not ideal to the nature of Particle Filter**. Therefore, Low-Variance Re-Sampling represents the particle distribution better.

Q11. Increase the number of trials until the sampling methods starts representing the true particle distribution equally well. Submit the visualizations for both methods.

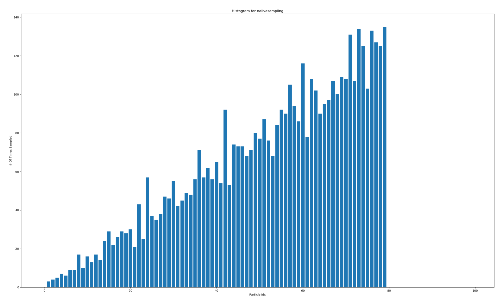
A11. We tried increasing the number of trials from 25 to 200 in the order $25, 50, 100, 200$. The visualizations of how the Particle filter behaves when we increase the number of trials are shown in the plots below :



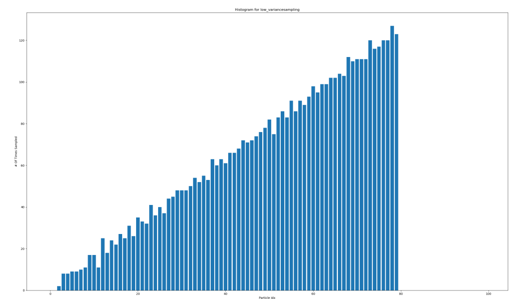
(a) Naiive Re-Sampling with `trial = 25`



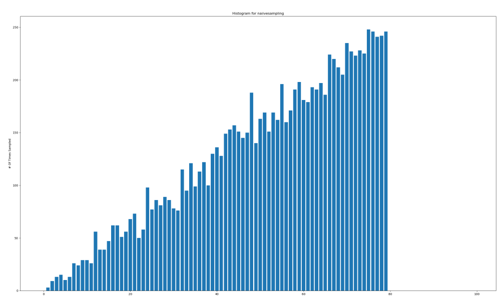
(b) Low_Variance Re-Sampling with `trial = 25`



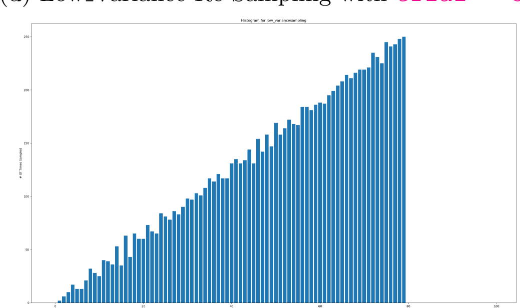
(c) Naiive Re-Sampling with `trial = 50`



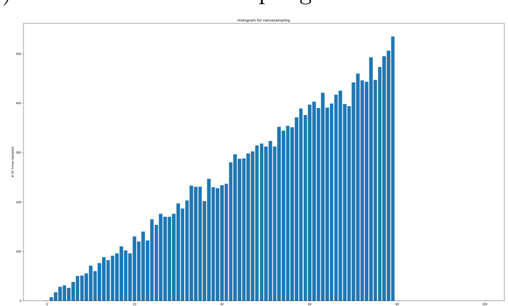
(d) Low_Variance Re-Sampling with `trial = 50`



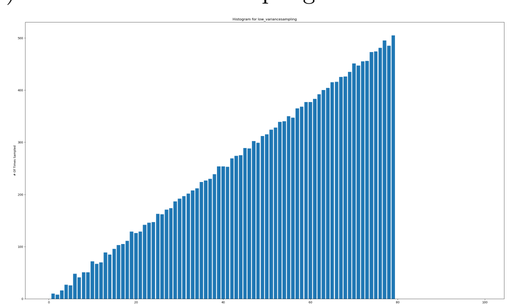
(e) Low_Variance Re-Sampling with `trial = 100`



(f) Low_Variance Re-Sampling with `trial = 100`



(g) Low_Variance Re-Sampling with `trial = 200`



(h) Low_Variance Re-Sampling with `trial = 200`

Figure 14: Effect of `trial` on Re-Sampling Models implemented on the Particle Filter.

Q12. At what number of trials did both methods begin to represent the true particle distribution well?

A12. At `trial` = 200, we see that both Re-Sampling methods represent the true particle distribution.

Particle Filter (40 Points)

Q13. Record a video of your particle filter estimating the robot's state when playing back the *lab4/bags/real-floor4_corridor/full_2x.bag* file. **Make sure to visualize all of the topics mentioned in Section 4.5**. Show your particle filter's estimate of the expected pose in blue, and our recorded estimate in green. Please playback and record the estimates for the whole bag (note that at one point the robot pauses its movement, but then continues moving shortly afterwards). Your estimates will not exactly match ours, but they should be similar at all times.

A13. The drive link for the video of Particle Filter simulation in RVIZ following the given bag file is given here [Link].

Q14. Record a video of your particle filter running successfully on the real robot. You should use the `cse_basement` map, and the video should both show your robot moving via teleoperation around the basement. Any reasonable submission for this question will earn full credit, as long as we can reasonably see the localization capabilities of your system.

A14. The drive link for the video of Particle Filter simulation in RVIZ where the the real robot being modeled is given here [Link].

The drive link for the video of the robot being tele-operated in the real world is given here [Link] .

Note : When we tele-operated the robot, the inferred pose which gets the real world pose of the robot and dictates the particles. The inferred pose correctly follows the given tele-operated robot. But since, the robot model in rviz is not calibrated well with the real-world map. The Robot model is not keeping up with the inferred pose. But in reality, the particles are in synchronous with the tele-operation. This tells that the particle filter does its job, and we can observe better results when the robot model map environment becomes one with the real-world.

Please use your UW email account to access the drive link videos.