

Lab1 Writeup

6.1 Motion Model

- **What assumptions is this model making?**

The kinematic car model assumes no acceleration and the wheel rolls on flat ground without slipping. The car is viewed as a bicycle where the wheels are centered on rear axle of the vehicle. Moreover, the car kinematics assume that the car always move in circle. In the derivation, we also assume that the steering angle is a piecewise constant.

- **Why are so many particles in a 10cm radius of the ground truth for test 1 as opposed to test 2/3?**

Observe that the first test has smaller constants for t , v , and θ , we would show that these differences lead to a smaller absolute noise. We show it holds for heading in motion update rule, and then x and y follows assuming δ is a piecewise constant. Consider the term $\Delta\theta = v/L \times \tan(\delta)dt$. A small perturbation in v and δ in our case can be linearized by first order partial derivative, so we have $\partial\theta/dv = 1/L \times \tan(\delta)dt$, and $\partial\theta/\partial\delta = v/(L \cos^2(\delta))$. Therefore, with larger v and δ , we have greater perturbation in θ . Moreover, test 1 has smaller dt than the others, which makes θ perturbation even larger. Therefore, intuitively, we would have more particles in a 10cm radius of the ground truth for test 1, compared with 2 and 3.

6.2 Sensor Model

- **What are the drawbacks of viewing each ray as conditionally independent?**

If some unmodeled object detected by the sensor, each ray are no longer independent because adjacent rays might have similar readings. By viewing them as independent, the car will repeatedly think a hypothesis is incorrect and become overconfidence. When used in particle filters, it might dispose potentially accurate particles.

- **What assumptions is this model making?**

The model assumes complete knowledge of the environment (map), and therefore the conditional independence of rays. In other words, the sensor model stores the map as a lookup table and assume that it does not

account for any dynamic change of the map. Another assumption in our sensor model is that the model for computing the simulated observation given a pose models well the real world scenarios. For instance, there are approximations involved in the library for collision checking. Moreover, in real world the rays might not reach the obstacle as the simulated ray does, and this is hard to model in one distribution. Furthermore, there's an assumption that the four models that we use for our sensor model are somewhat independent and their linear combination make good usage of each individual component. This might not hold if the models themselves have overlaps, not to mention that P_{short} is not a valid distribution(not sum to 1). Lastly, there are many unmodeled errors that are too complicated to compute in our lookup table.

- **Why does the hallway heatmap not converge on one point?**

Because the hallway has less distinct features to distinguish laser readings from different locations. Many spots on the hallway have similar laser scan (for instance the little rock at top right ahead) and thus about the same weight evaluated in the sensor model. Therefore in the heatmap they all have high probability being the ground truth.

6.3 Re-Sampler

- **Why do the samples close to k get sampled more?**

Because the samples close to k are assigned higher probability masses in the test code, they correspond to larger interval in the cumulative weights. Therefore, by the virtue of fixed step in low-variance re-sampling, we are guaranteed to sample more points from the larger intervals. Thus, after a few iterations of resampling, the samples close to k cover more points of sampling.

6.4 Particle Filter

- **Where does the particle filter do well? Where does it fail? Why?**

Particle filter, as a scalable and sampling-based method has its own advantages and drawbacks. In places with a number of features, particle filter is able to converge quickly to a good estimation of the car pose. For instance, in our global localization implementations, the island in the middle of the map has different laser readings from other places, such that particle filter is almost always able to recover this place.

However, it also has some drawbacks such as the requirement for the particles, motion model imperfections, and sensor model evaluations. Although our global localization is able to localize most of the places in the map, we apply a naive way of traversing through ~20 subregions of the map, which

is equivalent to increase particle numbers. This method turns out to be slow.

With our hand-tuned parameter in motion model, the particles spread out when the car passes the turns, which is due to the simplicity of our motion model and noise distribution. At turns, car often takes larger steering angle and therefore particles might suffer from starvation and loses track of the car.

For sensor model, it turns out that we would need a large amount of noise to be able to reach a stable state estimation. However, even with that, in places with similar features or a single place with no features, the sensor model is still unable to provide much useful information. Thus, the distribution that we sample from also affects the quality of our particles. This happens when our car is moving along the hallway.

7 Extra Credits

- The proof for the first extra credit is included as `/ex1_theory_question.pdf`.
- The global localization and kidnapped problems are implemented in `ParticleFilters.py` and `SensorModel.py`. To turn them on, comment out the subscriber to `/initialpose` and set `self.global_localize = True` in the `__init__()` function.
- The videos for global localizations and kidnapped problems are included in `/videos`.