Project Write Up, Kidnapped Car, Term 2, SDCND

All relevant code is the file oarticle_filter.cpp

1. Initialization

We need to initialize all the particles. I chose 100 particles as shown in line 23.
In the Particle_fitler.h file there eis the helper function init which
Initializes particle filter by initializing particles to Gaussian distribution around first position and all the weights to 1.

2. Prediction

In the prediction step we calculated the position of the car at the next time step. We use delta_t for the time step. Lines 66 et al. show the math calculating the prediction model. We distinguish between yaw rate =0 and non zero.
We also have to update the particles location after each time step. We add gaussian noise to the velocity and yaw rate.

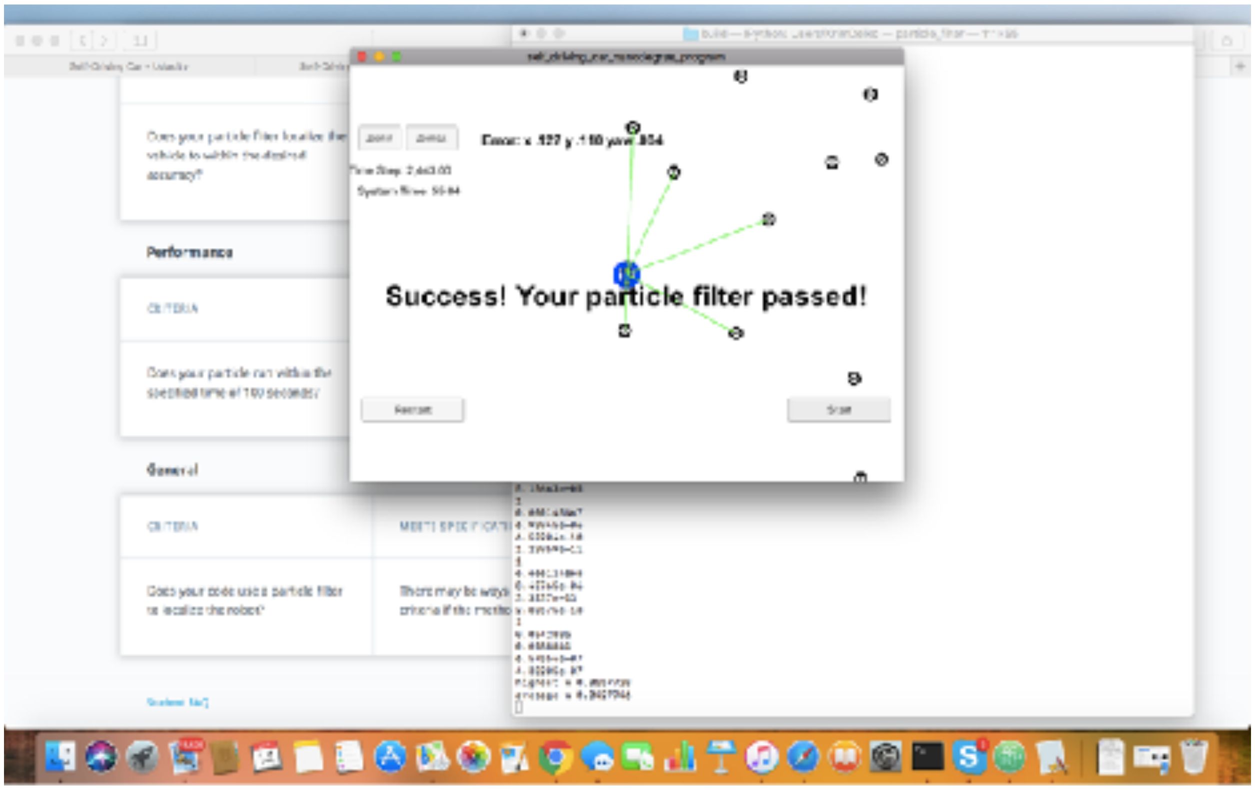
3. weights update

The ultimate goal is to figure out how well a particle represents the actual position of the car. We use car sensor and map measurements to weight each particle.
Car sensor measurements have to be transformed from car coordinate system to map coordinate system. The transformation is done in lines 134 et. all.

Before we can update the beliefs of our position we have to solve the data association problem. Data association combines the position of landmarks with measurements. We make sure every measurement is correctly associated with a landmark.
We choose nearest neighbor to pick the right correspondence between the measurement and the landmarks. this is the closest measurement.

Now we have all the pieces we need to calculate the particle's final weight. This is done according to the formula in lesson 15 section 18. In the code it’s line 174 et all.

Here is the final screen shot from the car simulation:



Does your particle filter localize the vehicle to within the desired accuracy?

Performance

CRITERIA

Does your particle run within the specified time of 100 seconds?

General

CRITERIA | MEETS SPECIFICATIONS

Does your code use a particle filter to localize the robot?

There may be ways [illegible] criteria if the method [illegible]

Error: x .127 y .110 yaw .004

Success! Your particle filter passed!

Further thoughts

The concept of this project is well explained in lessons 14 and 15. However the details of the code our much tougher. I spent hours trying to figure out bugs and surfing the forums and the web for answers. Finally I got it to work but the relationship between learning the concepts in class and time spent actually implementing all the details of the code is a bit off. My suggestion would be to either base the projects more on concepts and less on detailed nitty gritty implementation or structure the course work more toward the coding part so students have more ammunition to fight the problems of implementation.

Another question I wonder about is how machine learning fits into the particle filter concept. Is this a real time tool or can it be trained and hence accelerated thourgh AI.