Writeup

Path Planning Project

This project is built on the starter code provided by Udacity CarND team.

In the main.cpp, I set the variable ref_vel (row 205). It is used to control the target speed. It is set to 5 mph rather then 0, in order to start up faster. The ref_vel is less than 50 mph that is the speed limit.

To avoid collision, I use the d value to detect whether other cars are in front of the ego and close to the ego. If the d is less than 30 meters, we tell the ego there is an obstacle in the front and let the ego to take actions. The actions could be lane changing, or slowing down and lane keeping.

First, lane changing. Check all availabilities to let the ego change lanes safely. The strategies are as follows.

1. Detect obstacle in front of the ego, record its position s and speed.

2. Detect the vehicles in the left lane and driving in front of the ego; search the nearest vehicle (left_up) within 200 meters, and record its position s and speed.

3. Detect the vehicles in the left lane and driving behind the ego; if the nearest vehicle is within 30 meters, the distance is not safe for merging, so I set left_open set to false.

4. Compare the left_up with the obstacle in ego's lane; if the obstacle is faster, there is no need to merge left and set left_open to false.

5. Check the s value of the left_up vehicle; if it is smaller than 30 meters, merging left is not safe; set left_open to false.

6. If no left lane, set left_open to false.

7. For changing to the right lane, the strategies are similar.

8. If both left lane and right lane are open, choose the one with faster traffic.

9. Change lane according to availability. lane-=1 for left lane and lane+=1 for right lane.

Second, slowing down and lane keeping. If either left lane or right lane is available for merging, the ego slows down and keep driving at the current lane, until a lane is open to merge.

The next step is to generate and smooth trajectories. I use the spline library to smooth the trajectories. I select the anchor points based on the previous path, with 30 meters increment in x-axis and fitted by a function in spline. The anchor points are first converted into vehicle's angle, which means theta set as 0. After fitting the polyline, the points are converted back to original x-y coordinates and fed to the simulator.