CarND: Highway Path Planning in Simulation Environment

Overview

Path planning is one of the hardest and most critical part of self-driving cars. Here you are required to chalk out a path of the car using sensor data of all other vehicles on the highway and road knowledge. In this project task is rather simple, we don't have any exits to take or any junctions to navigate out of. Here we are simply finding our way through different lanes by lane switching, slowing down, or speeding up depending on traffic.

How was it completed?

Largely using the ideas and code provided in Q&A session between Aaron and David Silver, I was able to put things together.

Reading Sensor Fusion

Sensor fusion data made it easy to figure out the position of all other cars on the highway. Using this data I determined whether the ego car is in the same lane as other cars. By using the 60-meter gap (30 in front and 30 behind), on either side (left or right) of the highway, I determine whether to continue with a lane change or not. Lane change decision is taken whenever we find a car can't stay in a slowed down state behind another car.

The starter code from Q&A session and previous lessons made it easy to come up with a lane variable and check each car's lane, and its distance from the origin.

No Jerks

By ensuring the car moves slowly and without jerks, an acceleration of 5m/s is applied until the car reaches its maximum speed of 49.5km/h. Conversely, when a car is found to be less than 30 meters ahead of our car and there appears to be a chance of collision, its speed is slowed down by the same amount.

Lane Switching

While the car performs careful lane switching, sometimes it remains in slowed-down state in a lane if it isn’t safe to switch lanes.

Use of Spline Library

Spline library made it super easy to ensure smooth turns. I essentially use the code provided by Aaron here to chalk out 3 points in the future which are each 30m apart.

Reflections

Since the problem statement in this project was rather simpler than real-life scenarios, I decided not to implement a proper state machine. Variables are fewer in the case of simulation and that too of highway traffic, using the few scenarios code is written to ensure no collision/unnecessarily incidents (jerks) are avoided.

In the real world, the sensor's data may not be as accurate or reliable as has been the case here. With errors from sensors, coding would be a bit more involved to ensure we don’t act on wrong information that occasionally may popup.

Nonetheless, the project gives a good taste of things in the world of path planning and it was a great experience to go through it.