Path Planning Project

Udacity Self-Driving Car Nanodegree Term3

Jiahao Cui

1. Goals

The major goal of this project is to design a path planner that can create smooth and safe paths for the car to drive on highways. Since the car drives in a discrete state space, the main challenge of the task is to design a reasonable cost function which can lead the car to behave correctly. Besides, we need to minimize the acceleration and jerk in our paths in order to make passengers feel comfortable.

2. States and Behavior Spaces

The car drives in a highway with three lanes and 50 mph speed limit. We should drive as fast as possible in order to reach the destination quickly but not violate the speed limit. So I chose the 49.0 mph as the standard speed. There are four situations to be considered, each of which should be handled differently.

Situations	Desired Behavior
No obstacles vehicles, speed is lower than 49.0mph	Accelerate
No obstacles vehicles, speed is higher than 49.0mph	Brake to control speed.
There is a close vehicle , safe to change lane	Change to the lowest cost lane
There is a close vehicle, unsafe to change	Keep lane and brake to avoid collision

I first used the sensor fusion data to compute the closest vehicle in front of the current lane. If there is a car detected within 30 meters ahead of the car, the **too_close** indicator will be set to true and the cost function will calculate the cost of each lanes in order to choose an appropriate behavior. Otherwise, the car only need to keep lane and tune its speed.

3. Cost Function

The cost function plays a critical role in the path planner. The cost of a lane is decided by the S-axis distance between the car and the nearest vehicle. I also used the sensor fusion data to compute the S-axis distance and recorded the nearest distance. The cost function is written as follows.

$$f(dist) = 2.0/(1+\exp(-10/dist))-1$$

The range of the result is [0,1], the closer the distance, the higher the cost value.

Then, we must penalty the collision risk with an extremely large cost. In this project, I add 5.0 cost to the lanes whose nearest distance is smaller than 10 meters. As a result, such lanes will never be chosen by the car.

4. Smooth Lane Change

The implementation of lane change is relatively easier but we still need to smooth the process in order to minimize jerk and acceleration. As suggested in the course, I built up a spline and set it with generated path points to make the path smoother. The parameter target_x is important because it decides how many steps a lane change process takes. I first set it to 30 but the car still suffered from occasional big jerk when changing lanes. So I settled this parameter to 50 and finally the process of changing lanes is very stable and smooth.

5. Output of the Planner

The performance of my planner is relatively good. As shown below, the car ran 4.60 miles in 6 minutes without any incidents. The average speed of the driving is 46mph, which is close to the 49mph standard speed.