Assignment 8 Jm3zhang

Trajectory generation

1) How is lane following achieved?

Lane following is achieved by updating the x, y coordinates based on the increment of s, d value in Frenet coordinate. The d value in Frenet coordinate keeps track of where the car position is in between lanes and s value in Frenet coordinate follows the trajectory of the lane which can be used to achieve lane following.

2) How to use spline to generate a smooth trajectory?

In order to generate a smooth trajectory, firstly fit a points 30 meter ahead of the starting position using spline. Secondly, using the pervious line fit from spline, we will have an end point of 30 meters away from the x direction. With that end point, we can find the position in y direction. With that x, y coordinates pair, we can separate the line fit from spline to whatever parts needed (in this case, I used 50 parts) base on distance, speed and time required to obtain the smooth path for the car. Finally, since we know how many parts we need, we can find position of the path in y direction by split the 30 meter line fit using spline along x direction into 50 parts which will result the smooth trajectory.

3) How to avoid collision with the car in front?

In order to avoid collision, firstly look at the sensor fusion list to see if there are cars in the same lane as my car current in and if that is the case, we will need to find the next location of the car that is in front of my car. Compare the next location of that car to my car to see if it is closer than 30 meters in distance and if it is, slow down to avoid collision.

4) How to avoid cold start?

In order to avoid cold start, set the initial reference speed (ref_vel) to 0.0 instead of the max speed to start at zero velocity and gradually accelerate by increase the reference speed smoothly.

Behaviour Planning

4) Explain your approach concisely in the writeup

For my approach, the first step is to find out which lane the car is currently in. Since the car needs to change lane where there is a car too close in front, I need to know which car lane the car is currently in to decide whether a shift left only shift right is safe. For example, if the car is currently in the right lane, then the only option for the car is to stay in the right lane or shift left if the front car is too close and the left lane is safe for a lane change. We cannot perform a right shift because there is no lane on the right side in this case.

The second step is to check if there are cars on the left lane or the right lane that prevent my car from safe lane changes. I will look at either left or right lane to see if there is a car that is 40 meters in front of (less than 40 is not worth for change) or 30 meters behind (on a different lane) my car's position (on the current lane). I also find out which car lane my car was pervious at to prevent useless checks. If there are cars on either left or right, a corresponding flag will be used to represent that (i.e. right_lane_has_car).

The third step is to perform the lane change. When my car is too close to the car in front, I will check if the left lane is safe, and my car is not currently on the leftmost lane. If that is the case, change to the left lane and vice versa for the right lane. If both left and right lane are not safe, reduce the car speed to prevent a collision with the car in front.

The fourth step is to change back to the middle lane when it is safe and currently not in the middle lane. I do this because when the car is in the middle lane, I have more choices for lane changes when compared to the other two lanes (This will make the time needed for 2 miles shorter). Also, to prevent over speeding, clamp the ref_vel value to 49.5.

If you want to have more information on how I implement the Behaviour Planning, I have commented my code for that.