**CarND-Path-Planning-Project**

Himanshu Kumar

The goal of this project is to drive a car safely on highway with other vehicles driving at various speeds. The autonomous car should be able to:

1. Follow speed limit

2. Should be comfortable during drive (do not exceed max jerk, acceleration)

3. Should change lane when vehicle ahead is driving slowly and it is safe to change lane

4. Should stay in its lane

5. Avoid collision

The section below will explain how we achieve each criterion.

**1. The car drives according to the speed limit.**

To achieve this, we capped the max speed of the car to 49.5MPH (speed limit is 50MPH).

**2. Max Acceleration and Jerk are not Exceeded.**

We solved the issue of max acceleration/jerk by using a small value of 0.224 mph to increase or decrease the speed of the car. This gives a maximum acceleration of 5/s/s.

**3. Car does not have collisions.**

We used sensor fusion data to detect vehicles around us. If there is any car within 30m of range then, our car will take safety measures like reducing the speed if another car is ahead of us and do not change lane if there is car in another lane.

**4. The car stays in its lane, except for the time between changing lanes.**

Our car mostly stays in its lane until it needs to change lane (if there is slower car ahead of it).

**5. The car is able to change lanes**

This was fun problem to solve. Whenever, we detected a slow car ahead of us, we reduce our speed by 0.224mph to avoid collision but at the same time also started looking at vehicle around us. If it’s safe to change lane (no other car in 30 m range) then and only we change lane. In our logic, we have given left lane preference over right lane (as left lane is fast lane).

**6. There is a reflection on how to generate paths.**

To generate path, we have used spline library and used library as shown in Udacity class. We provided anchor points to it and then asked “Y value for each “X” to get (X,Y) coordinates of our path.

1. We used remaining previous points and added future path points to it for smooth transition.

2. We used car’s “S” value to get future way points in 30m, 60m and 90m range. These points were used as anchor points for spline.

3. Converted these global coordinates anchor points to car’s coordinates for simple math.

4. we are doing path planning for next 30 m (~1.5 sec in future). We avoided planning for too short and too long, as they might not reflect correct situation around our car.

5. First, we divided the distance into N points using our speed.

6. Using spline, got Y coordinated for each X.

7. Converted these car’s X,Y coordinate to map’s (x,y).

**7. Future Enhancements:**

1. I think, putting a strict range of 30 m (ahead and behind) in the other lane for no car, sometimes caused no lane change which I guess might be possible by changing our current speed. For example, if there is car in concerned lane which is within 30 m ahead of us but no car behind, I guess by reducing our speed we should be able to change lane.

This might require cost function.

2. Generating multiple path for lane change for both lanes. I am giving priority to left lane but a right lane change might be better choice.

NOTE: I have used UDACITY video class code as a base for my project. And its heavily based on that.