Path Planning: Krishna

Introduction:

In this project your goal is to safely navigate around a virtual highway with other traffic that is driving +-10 MPH of the 50 MPH speed limit. You will be provided the car's localization and sensor fusion data, there is also a sparse map list of waypoints around the highway. The car should try to go as close as possible to the 50 MPH speed limit, which means passing slower traffic when possible, note that other cars will try to change lanes too. The car should avoid hitting other cars at all cost as well as driving inside of the marked road lanes at all times, unless going from one lane to another. The car should be able to make one complete loop around the 6946m highway. Since the car is trying to go 50 MPH, it should take a little over 5 minutes to complete 1 loop. Also the car should not experience total acceleration over 10 m/s^2 and jerk that is greater than 10 m/s^3.

Implementation:

Based on the starter code provided in the classroom, the path planning algorithms start in the main.cpp of the 'src' directory of the project. On the basis of the sensor data received from sensor fusion, we determine the following things regarding the car environment (line 250-294):

- If a car is ahead of us
- If a car is in the left lane in front or back
- If a car is in the right lane in front or back

Based on these predictions, the model generates the behavior part, that is, how the car would react in case of the above situations. So, the car makes the following decisions (line 297-311):

- If the car speeds up or slows down based on if there is car in front
- If the car should change lanes

Based on the predictions made above, the car has to move either in a straight line or make a lane change. The lane change is facilitated by the use of the **spline** library of C++, to ensure a smooth lane change.

Reflection:

The car drives without any incidents which include exceeding acceleration/jerk/speed and driving outside of the lanes:

The car maintains the speed of approximately 50mph. The velocity is set to 49.5 And by the formula, N velocity time = distance, where N is the number of points in the trajectory which gives us the speed control.

Max jerk and acceleration is handled by increasing or decreasing the speed value by 0.224. It increases if speed is less than 49.5 and decreases if speed is above 49.5 or a car is ahead of us.

The jerk problem is reduced by using the spline library which helps in making smooth transitions which works similar to the polyfit function.

Car makes smooth transitions for lane change: If there is a car ahead of us and is moving slow, then our algorithm looks for its immediate left and right lane. If there are cars in our immediate left and right lanes then, our car slows down to avoid imminent collision. If there are no cars in left or right lanes then our car makes a smooth transition in to the respective lane. Also, if the car speed is below the reference velocity then, it tries to achieve the reference velocity.