Highway Driving – Path Planning

In this document I want to describe my approach in order to implement the project for highway driving in the Udacity Simulator.

The path planner operates as follows:

- In line 5 we are setting the lane to drive in
- In line 58 we set the reference velocity which will be used throughout the code. It needs to be 0 at first and will be gradually increased or decreased according to the situation
- Lines 110-150 are initializing various starting variables (the purpose can be found in the code)
- Lines 158-255 grab and evaluate the data from the sensor fusion
- In this segement we are checking ithe three different lanes in the same manner
- First we check if the lane is safe and set the according bool by checking if another car is in a certain range of our car in that lane (e.g. Line 182)
- Following to the "safety" check we look for the data of the car that is closest to our car and in front of us on that lane (e.g. Line 190-192)
- This is executed analog for all three lanes
- In lines 245-255 we check if a car is in our current lane and is in a range that is classified as too close to our car and set the flag accordingly
- This concludes grabbing the data from the sensor fusion. Next we are using this information to decide for our next action
- In lines 259-273 we can see which lane is the best lane in theory. This includes comparing speeds of the closest vehicles in all lanes and and checking if a lane is completely empty for the next 500 meters. With this we set the integer "best_lane" accordingly
- In lines 276-289 we are finally deciding which lane too choose if: there is a car too close to us, we are not already in that lane, the lane is safe and is considered the best lane at that point in time
- Lines 290-298 accelerate and decelarte if there is a car too close or the road is free to drive up until the max velocity 49.5
- Now we are setting up for path creation with the help of the spline library
- If we don't have any previously left over points in the path planner to execute we set the current pose of the car for the starting points
- If there are still points left over these get stored for spline creation and are used for the new spline creation (lines 316-333)
- In lines we create 3 widely spread points for the spline that will make sure that we have minimum jerk and already incorperate the lane object from the beginning of this document
- The points then get pushed into the spline creation
- With the help of evenly spread bits of the spline according to the vehicle velocity we want and the update rate we can achieve smoothly spread path points to then push these into the next values for the path (lines 382-404)