# Project Writeup

Anirudh AV (anirudh.av93@gmail.com)

## 1. Model documentation

The model can be split up and analyzed into modules that tackles specific problems described below:-

- Staying in he center of the lane
- Collision prevention
- Lane changing
- Jerk prevention and speed

### *1.1. Staying in the center of the lane*

The path calculations are all done in the Frenet frame. Once we generate the next s coordinate using spline function (Explained in the section ”jerk prevention and speed”) the points are then transformed to the cartesian coordinate system that the simulator is expecting. The helper function **getXY** comes in handy for this. The d coordinate is calculated as d = (2 + 4*lane_number). This ensures that the d values fall exactly in the center of one of the three lanes to the right of the yellow line. See lines 393 - 395 in ”main.cpp”.

### *1.2. Collision prevention*

The sensor fusion data gives us s and d information about other cars on the track. Looking at the d value of the cars we can find the cars in our lane. We can estimate the s coordinate of the time in the future as we get the speed information of the other vehicle from sensor fusion. If the distance between our car and the estimated position of other car in future is less than 30. We do the following :-

- if the cars distance is greater than 20 from our car, we match the cars speed.
- if the cars distance is lesser than 20 from our car , we go slightly slower than the car.

See lines 274 - 306 in ”main.cpp”.

### *1.3. Lane changing*

”Staying in the center of the lane” section describes how the lane number gives us the corresponding s coordinate of the car in the lane specified by the lane variable. Three variables help us change lanes when needed. A lane change is needed when there is a car ahead of us slowing us down, this is captured by the car ahead boolean. When a lane change is needed, we first check to see for a possible left lane change, and then the right. And then increment or decrement the lane number to parse it to the motion model. See lines 308-365 in ”main.cpp”

### *1.4. Jerk prevention and speed*

The only way we can affect speed is by changing the spacing between 2 way points. In order to generate a smoother path, we interpolate between 2 waypoints that are 30 units apart. We use the spline library to perform this interpolation. The number of points within a fixed distance is increased to slow down the car and vice - versa.

See lines 369 - 465

*Preprint submitted to Udacity Self Driving Nano Degree* *May 8, 2018*