PATH PLANNING PROJECT

MODEL DOCUMENTATION

The goal of this project is to have our autonomous vehicle safely drive a simulated highway for at least 4.32 miles without any incidents. Incidents are defined as falling outside of our acceptable acceleration of 10m /s^2, jerk 10m /s^3 and speed limit of 50 mph, collisions (touching another vehicle) and driving outside of the lanes for longer than 3 seconds. The car will change lanes when it makes sense to do so - that is, our autonomous vehicle can safely move into an adjacent lane when a slower car is in front of us and the adjacent lane allows us to get closer to the speed limit.

Our program will first read in all the map_waypoints to help us figure out where our car and the other cars are on the highway. Then we will specify our starting velocity and lane position. Next we read in our car's location information from the simulator. Additionally it will let us know what pts weren't used from our previous path's trajectory list. Finally, the simulator sends us information about the other car's on the road - our sensor_fusion. In the behavioral planning portion of our project our code we will first search through the sensor_fusion list to see if there's any cars in front of us. If a car is very close (the red zone - under 5 meters away) - then we're in danger of an immediate collision - we tell our car to slam on it's brake's to stop - reducing our car's velocity to 1 mph. Otherwise if we sense a car very close (orange zone - greater than 5 meters but less than 10 meters away) then we must slow to a crawl - setting our car's velocity to 5 mph. Next if we sense a car greater than 10 meters away but less than 25 meters - immediately slow quickly. Next if we sense a car in front of us greater than 25 meters away but less than 90 meters we will consider a lane change. Depending on what lane we are in, we now look to an adjacent lane to see if it's safe to move into it. We do this in the changeLane function where we first look to see if there's any cars right where we would like to move and then we also look to predict where any cars might be when we move into that lane. That is, we predict where we will be in .6 seconds (the duration of our lane change multiplied by our current velocity plus our current position) and where any cars in our prospective lane will be in .6 seconds (that car's current velocity multiplied by the duration plus that car's current position). If it's safe we set lc (lane change) to true otherwise false. If true we set our trajectory for that lane otherwise we reduce our car's velocity incrementally depending on how far away the car in front of us is.

Next we calculate our trajectory. We'll first grab our car's localization data and then we grab the last couple of pts left on our previous trajectory list (if available) otherwise we extrapolate where we were from where we are now. Next we add points straight down the middle of the road, 30, 60 and 90 meters away. We add this to our list and we create spline with these five points. We calculate where the y value will be on the spline for our x value 30 meters away. We then create our trajectory list by calculating the distance to move along the spline so that we hit each point every .02 seconds at our desired velocity - ref_vel. And this will be both our trajectory and our velocity which we then send to our simulator. And that's it. Repeat the process.