Path Planning Project - Model Documentation

Introduction

The aim of the project is to drive safely on a highway, equipped with three lanes and many cars around, for 7km without incident; this involves avoiding abruptly acceleration and deceleration, containing jerk and preventing collisions with the other vehicles. The speed limit is set to 50 MPH.

Environment and Data Provided

Path planning is that process whereby a vehicle finds safe, efficient and comfortable path analyzing the outer environment. Particularly for this project, the car moves within a given map from a start to a target taking into account the decisions produced by a path planning planner. The path planning algorithm relies on input from sensors whatsoever it is able to produce a certain behavior for the car's driving by predictions and trajectory generation.

The waypoints are given in the highway_map.csv file and are spaced roughly 30 meters apart, so the first step in the process is to interpolate a set of nearby map waypoints and produce a set of much more tightly spaced which help to produce more accurate results.

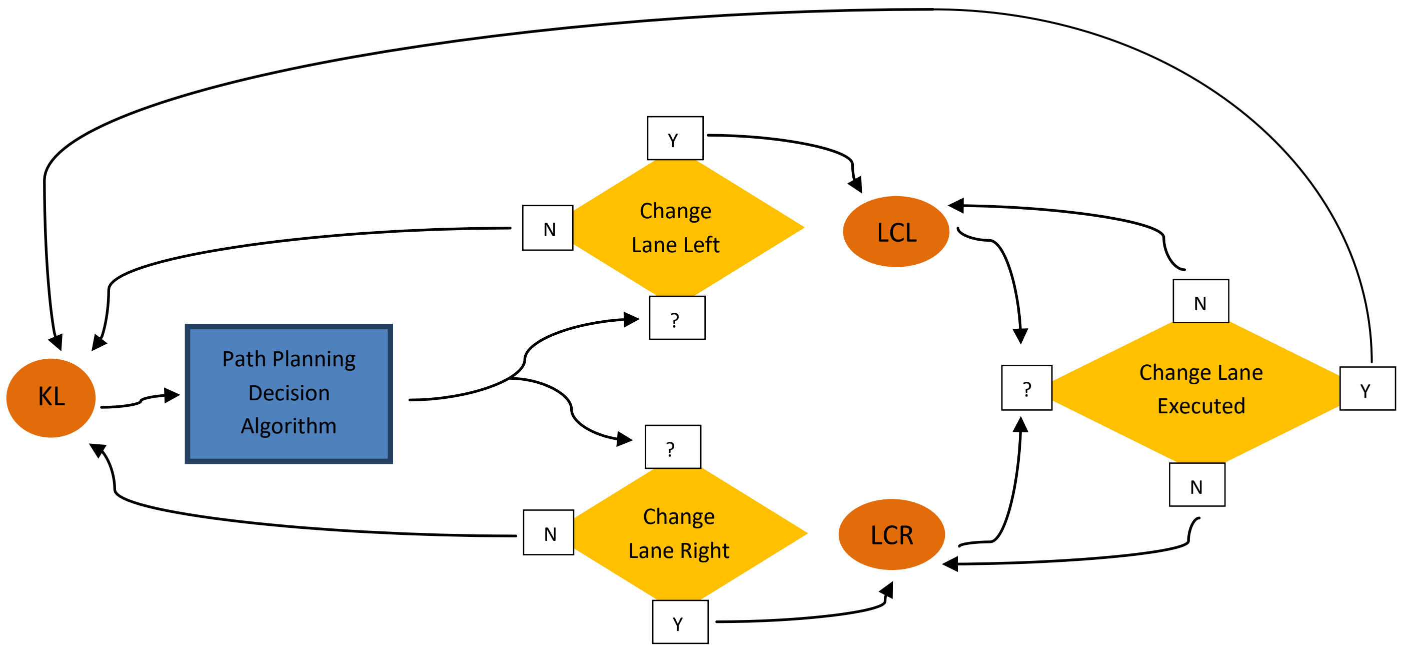
Finite Machine State

Implemented a finite state machine which allows changing target lane relying on the outcome of the lanes analysis.

The states implemented are the following:

- **KL**, Keep Lane
- **LCL**, Lane Change Left
- **LCR**, Lane Change Right

Below is illustrated a scheme of the state machine.



Best Trajectory’s Choice

All the reasoning about lane changing come from the perception of the environment provided by the sensor fusion system dataset.

One of the notably bond has been represented by keeping a safety security distance from the vehicles around as well as preventing collisions, thereby they have been implemented a series mechanisms in order to change lane just at the right time with a minimum risk.

The cost functions created are based on the following principles:

- The ratio of the nearest car velocity and those of the ego car
- The predictions of the positions of the other car at the next step
- The distance from the vehicle ahead and behind
- The numbers of cars in a single lane

For every single lane is associated a penalty if the objective is too far from the current position from the ego car coordinates.

Adas System

Introduced an ADAS system capable to detect if the vehicle ahead is decelerating and promptly responding with a braking assistance without violating the bond of jerk and acceleration of the project. The distance from the vehicle is normally maintained at a distance of 10 meters.

Generate New Path

In order to create smooth trajectories, I used the C++ spline library with the consequence of not taking care of all the math concerning to minimize jerk just adding a header file to the project.

Creating an ideal path for the car to follow the first thing is try creating a spline with the nearest waypoints in our car reference. The new path begins with a certain number of points from the previous path, which is received from the simulator at each iteration.

Therefore, cyclically, a spline is created beginning with the last two points from the previous path and ending with two points at a given distance ahead and in the target lane. This target distance is divided into fixed small steps, each of them containing the information of the target velocity to follow and the points in x and y car’s coordinates to reach the next step. At the end, all points are converted to global world points and passed to the simulator for driving. As mentioned before this approach produces a smooth x and y trajectory. At this stage, the only precaution is decrement or increment the car’s speed by small fractions of it in order to prevent the driving form excessive acceleration and jerk.