Path Planning: Highway Driving

Seymour Dadashov

August 11, 2020

Abstract

The purpose of the project is to demonstrate understanding in developing a path-planning model capable of taking in sensor fusion measurements and constraints that define a comfortable driving experience and outputting a simulation demonstrating a vehicle able to smoothly lane shift/navigate a highway environment.

1 Design Parameters

In this project a vehicle is simulated on highway traffic road stretching an approximate $4\frac{1}{3}$ mile loop. A speed of 50 mph is maintained whenever possible and lane changes are made when necessary. In addition, the following constraints are imposed on the motion of the car as it makes its loop around the track:

- The car should not go over 50 mph.
- Maximum acceleration of 10 m/s^2
- Maximum jerk of 10 m/s^3.
- The car should only be outside of a lane for a few seconds when making lane changes.
- The car should not collide with any other vehicles on the road.

2 Model Documentation

To model the behaviour a large portion of sensor information needs to be processed in order to make an appropriate decision about the next action along the given path. The function `processSensorFusion()`, line 236 of main.cpp, processes sensor fusion data multiple times each second and provides a recommendation (stay in lane, change lane left, or change lane right). In addition, a recommended speed is also calculated if a lane change is not possible. To calculate the cost of a lane the distance and speed of the vehicles immediately surrounding our car are calculated. The approximate future position of surrounding vehicles at the time of the end of our current path is used to determine if a lane is safe or we need to move from the lane we are in.

Path planning is done after the call to `processSensorFusion()` on line 501 of main.cpp. To create a path that kept motion within the constraints of maximum acceleration and maximum jerk I used the spline tool recommended in the class. First, on lines 507 - 534, the recommended lane change and speed changes coming out of sensor fusion are made. Since we need at least 2 points to generate a spline we either calculate 2 points based on our current position and heading or use 2 unprocessed points left over from the previous frame's path. The spline control points are then augmented with a few more points further out. These spline points are then used to interpolate many points along the updated path. The new points are then concatenated with the list of previous points and sent back over the socket to the simulator.

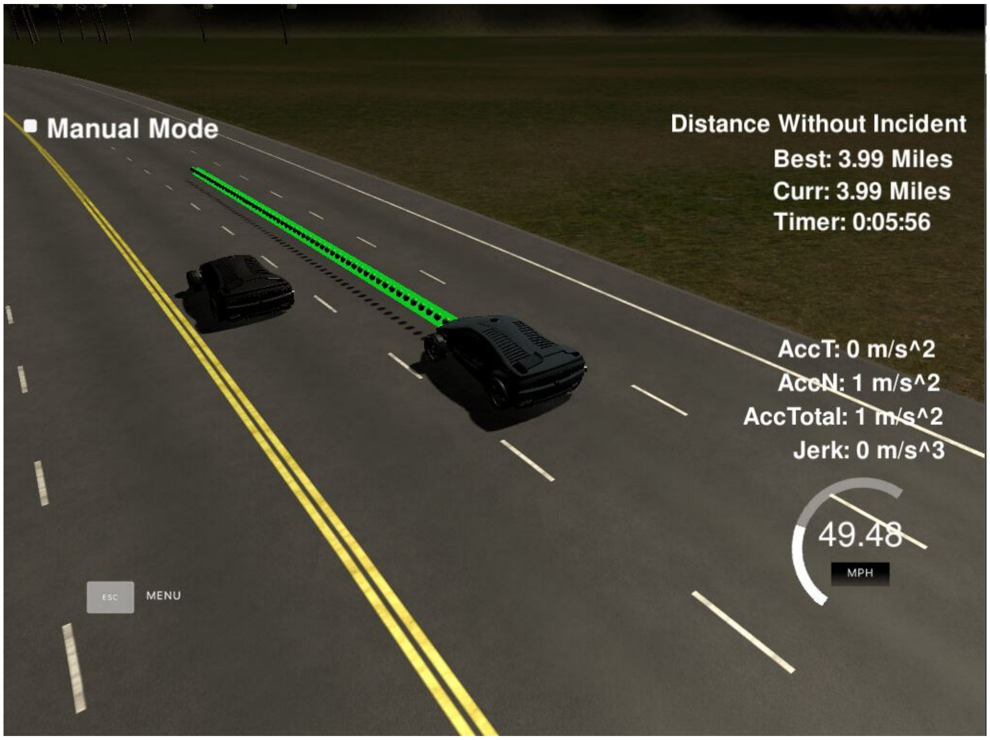

Figure 1: Screenshot of simulation in progress.

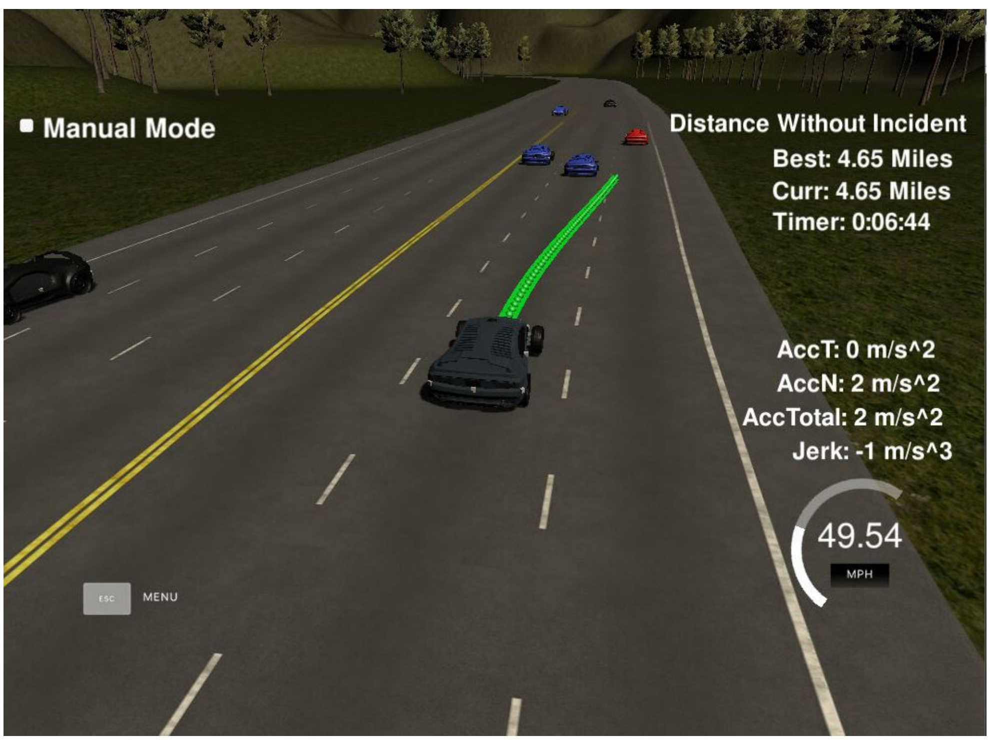

Figure 2: Screenshot of simulation having passed entire lap and in process of performing a lane change.