Nathan Sherrit
2019/01/17
Self-driving Car Engineer - 2
Project 2 - Highway Driving

**Project Overview**

The goal for this project was to simply have a simulation of a car drive on a highway without having any major incidents and as quickly as possible. The car must drive the speed limit with smooth controls and maintain its position in its lane. If there is traffic obstructing the path, the car must change lanes or reduce its speed to remain as safe and as efficient as possible. The car is not allowed to collide with any other vehicles, cannot have large acceleration or jerk, cannot violate speed limits, and must stay in the center of the lane unless performing a lane change.

**Project Results**

The car was able to complete several laps around the track without any incidents.

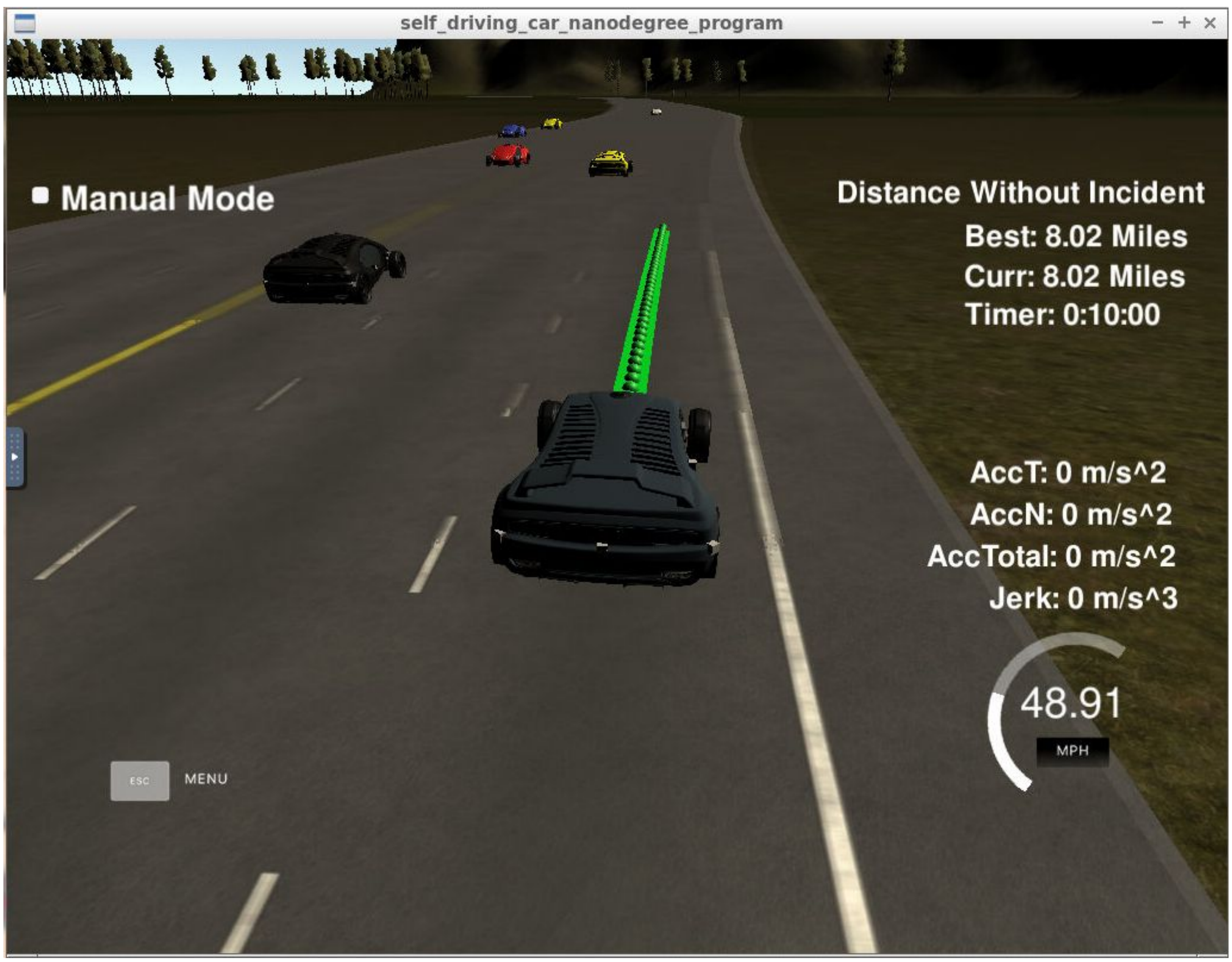


Figure 1: Results of the autonomous car driving on the highway simulation.

The above image shows that the car was able to travel over 8 miles without any incidents in a total time of about 10 minutes. This equates to an average speed of 48mph. This demonstrates that the car was able to complete the route while changing lanes to maintain an optimal speed.

**Project Code**

The project was broken up into several smaller sections each with their or goal to complete the project.

1. The first task was to simply drive the car in a straight line. Using the sample code in the projects FAQ video this task was completed very quickly.

2. The next goal was to drive the car in the lane. Again this was covered in the video and I was able to apply the technique show into my own code. Creating several target points in the Fernet plane, I was able to generate a spline in the XY plane that the car could easily travel. This allowed the car to follow the road in a smooth fashion.

3. After the car could travel in a single lane the next job was to make the car perform a smooth lane change. This was accomplished by changing the target points used to generate the spline. All I needed to do was shift the points over a lane and the spline would create a path that allowed the car to transition to a different lane.

4. Now that the car can change lanes, the next goal was to determine when a lane change was necessary. Each time data is passed to the car the algorithm checks to see if any target vehicles are in front of the car. If there are no targets in the ego lane, nothing happens. If there is a vehicle the front of the subject, and the time to collision is less than 2 seconds, the subject vehicle needs to determine what to do next. The time to collision is calculated by taking the distance between the cars and dividing it by the difference in velocity.

5. When the subject vehicle is close to a target there are several options the subject can do. It can; stay in the current lane and reduce speed, change lanes while maintaining the current speed, change lanes while accelerating, or change lanes while it is decelerating. Each of these possibilities is simulated and a cost function is created for each outcome. The path with the lowest cost is chosen as the path for the subject to follow. Paths with collision generate very high cost values and are typically ignored. The only exception is if every possible path has a predictable collision within 2.5 seconds. In that case the subject we maintain the current lane and slow down to avoid hitting the car in front. Several different variables affect the value of the cost. Variables like the final speed in the new lane, time to collision after changing lanes, acceleration/deceleration amount.

6. Once the best possible path is selected the car will begin to follow that path. While the car is changing lane it will not look for any new paths unless it thinks that a collision will

occur during the lane change. Locking the path during a lane change fixed a bug where the car would get stuck in between lanes trying to figure out which lane was the best. Now a new path will only get picked during a lane change if it is absolutely needed.

**Issues and Updates**

The current solution to the self driving car is far from perfect and still have some issues that could be tackled. The first major issue is that the subject will sometimes get stuck behind a row of cars and not change to the fastest moving lane in heavy traffic. This occurs because the subject tries to drive as defensively as possible to avoid collision. This is great for safety, but sometimes being aggressive is needed to reach the destination faster.

Another issue currently is that the car will change two lanes at once, which some people may consider to be bad driving. This happens before the path planning algorithm is very simple and create a possible path to each lane, ignoring when the car is located currently. If the car is in the right lane and needs to move to the left, it will just drive across to lanes of traffic. Updating the path planner to understand that two lanes of traffic is not allowed would fix this problem. However, I noticed that sometime the subject would not notice the far lane, even when empty, and follow behind a slow moving target when this was implemented. A solution that is used right now is that the cost value is increased greatly when the car must change two lanes. This discourages the behavior, but still allowed it for when a double lane change is needed.

Tracking the target vehicles is another major update that was ignored for this project. Right now the target objects only have their current location and velocity collected by the subject. Using a simple kalman filter for each target would allow the subject to be predict when the target was going to go, and when it was going to be there. This would greatly help with the path planning since the subject would better accuracy predicting collisions.

Finally the last major update I would make to this project if it needed more development is to test for more possible paths before choosing one. Right now the car only checks 9 total paths. One path for each lane, and three different accelerations for each path. This worked well for the project but has its limitations. Increasing the variables to test would take longer to process, but would generate better results. For example testing different final position in the lane, or trying to accelerate before making a lane change, or predicting further into the future. All these different options would allow the car more freedom to choose when it thinks is the best place to be on the road.

**Flowchart of path planner (simplified)**