Return to Classroom

Path Planning

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Dear student,

This is a very great attempt as I can see you put a lot of effort to accomplish this project. Throughout the whole simulation, not a single incident occurred which shows that you have very well implemented the prediction, behaviour planning and trajectory generation.

I also liked your reflection as you have nicely explained your implementation combined with well-commented code which makes the project very easy to follow. I enjoyed reviewing your project and I am sure that you also enjoyed working on the project. Congratulations. 🎉

Cheers, and Keep up the Good Work!

Further Reading

You might also check out the these resources for further research:

- Introduction to Robotics #4: Path-Planning
- The path planning problem in depth https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume9/mazer98a-html/node2.html
- http://www.roborealm.com/help/Path_Planning.php
- A discussion on What is the difference between path planning and motion planning?
- Excellent Tutorial on A* Robot Path Planning
- Path Planning in Environments of Different Complexity
- Introduction to robot motion: Robot Motion Planning
- Introduction to robot motion: Path Planning and Collision Avoidance

Compilation

Code must compile without errors with `cmake` and make.

Given that we've made `CMakeLists.txt` as general as possible, it's recommend that you do not change it
will still compile on any platform.


```
Scanning dependencies of target path_planning
[ 50%] Building CXX object CMakeFiles/path_planning.dir/src/main.cp
[100%] Linking CXX executable path_planning
[100%] Built target path_planning
(base) aditya@aditya:~/Documents/review/path planning/9/home/CarND-Path-Planning-P
roject/build$ ./path_planning
Listening to port 4567
```

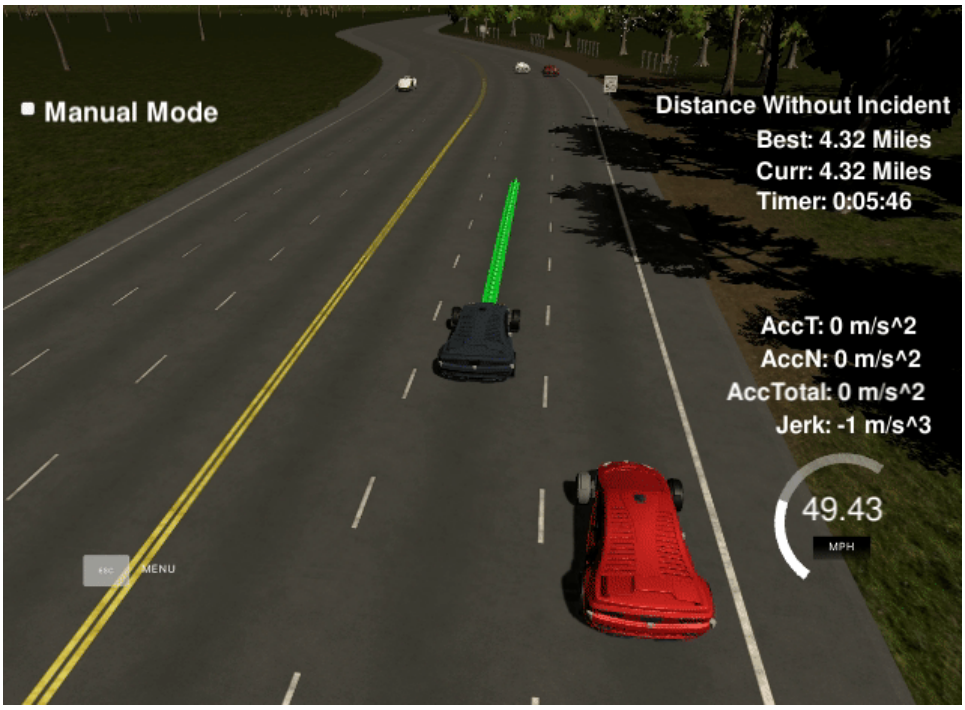
Awesome, your project compiles with `cmake` and `make` . For more information on this build tool, you may refer to the following:

https://cmake.org/runningcmake/
https://rix0r.nl/blog/2015/08/13/cmake-guide/

Valid Trajectories

The top right screen of the simulator shows the current/best miles driven without incident. Incidents include exceeding acceleration/jerk/speed, collision, and driving outside of the lanes. Each incident case is also listed below in more detail.

Your implementation easily meets this requirement. I ran the simulation for over 6 minutes and the car drove through a full lap of the highway without any incident as required. Great work!



The car doesn't drive faster than the speed limit. Also the car isn't driving much slower than speed limit unless obstructed by traffic.

Good job with the intelligent speed adjustment by dynamically updating the speed based on if there is a vehicle ahead or not. You have made sure that the speed should increase/decrease by 0.224 while keeping a check that it should not exceed 49.5 mph. This is very well implemented in your code.

The car does not exceed a total acceleration of 10 m/s^2 and a jerk of 10 m/s^3.

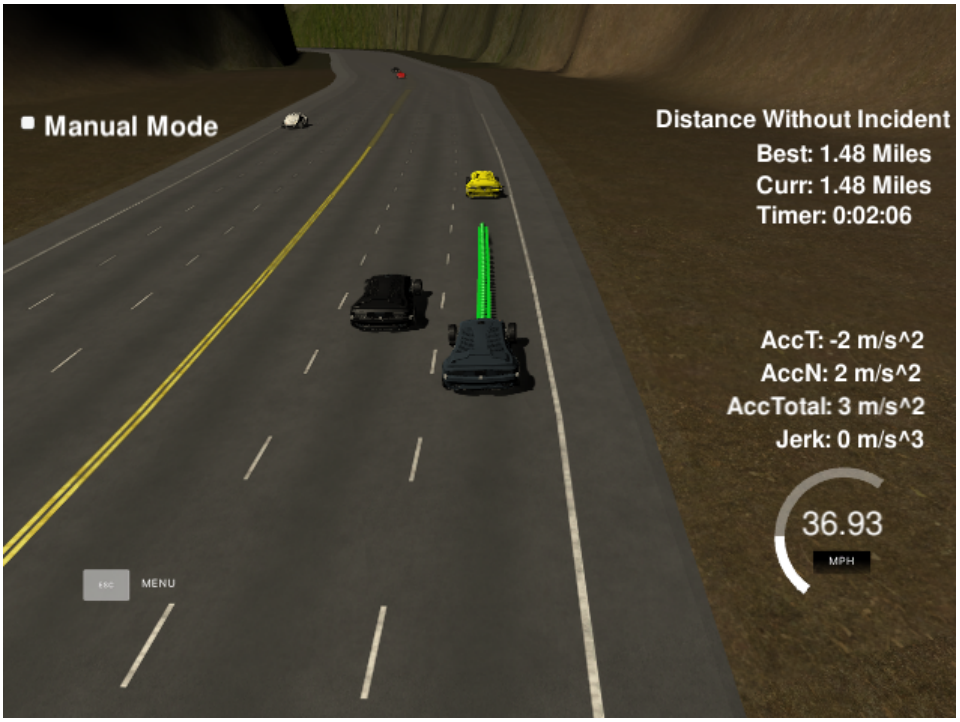
The car drove smoothly by not violating the total acceleration of 10 m/s^2 and max jerk of 10 m/s^3. Nice w

Suggestions and Comments

The following links can also provide more insights in this section of the project:

- Path Planning for Collision Avoidance Maneuver
- Optimal Trajectory Planning for Glass-Handing Robot Based on Execution Time Acceleration and Jerk

The car must not come into contact with any of the other cars on the road.



The vehicle slows down if there is a vehicle in front and it is not possible to change lanes which makes sure that there is no collision with other vehicles. This is well ensured on lines (171 - 173) in the file `main.cpp`. You may also add the logic such that the ego car prefers to drive in the center lane if it is available as that would prevent the ego car from getting stuck on side lanes. It can be simply implemented something like this

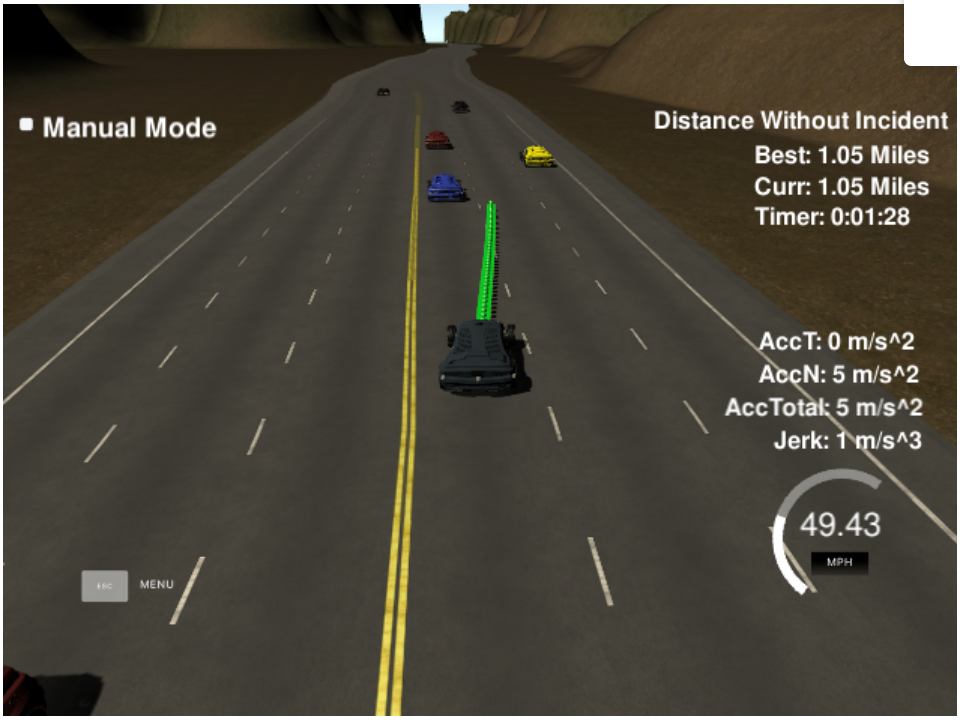
```
if(lane!=1){
              if((lane==0 && !car_right) || (lane==2 && !car_left)){
                lane=1;
              }
            }
```

The car doesn't spend more than a 3 second length out side the lane lanes during changing lanes, and every other time the car stays inside one of the 3 lanes on the right hand side of the road.

An improvement could be to prevent double lane change when a lane change is still on-going. This could be achieved by setting a flag true for say 3 seconds when a lane change event is triggered. This will also be helpful to prevent sudden oscillations in lane change.

The car is able to smoothly change lanes when it makes sense to do so, such as when behind a slower moving car and an adjacent lane is clear of other

The car is able to smoothly change lanes when it makes sense to do so, such as when behind a slower moving car and an adjacent lane is clear of other traffic.




```
if(too_close){
  if((lane == 1 && left_block && right_block)||(lane == 0 && right_block)||(lane == 2 && left_block)){
    ref_vel -= 0.224;
  }
  else if(!left_block && lane > 0)
  {
    lane -= 1;
  }
  else if(!right_block && lane < 2)
  {
    lane += 1;
  }
}
else if(ref_vel < 49.5)
{
  ref_vel += 0.224;
}
```

If there is a slower vehicle ahead then the car changes the lane smoothly. The car also correctly predicts if the adjacent lanes are available to drive by using the sensor fusion data and only then changes the lane thus avoiding collision with the vehicles in the adjacent lanes. Good work. 👍

Reflection

The code model for generating paths is described in detail. This can be part of the README or a separate doc labeled "Model Documentation".

Nice job! You have discussed all the aspects of how the path planning works. You have explained your implementation of all the steps(prediction, behaviour planning and trajectory generation) by referencing your code. 👍 You may also check these awesome resources below based on path planning.

> Though achieving everything required in the rubric, I have not used the cost function in my application but used a scenario based behavioral

Though achieving everything required in the rubric, I have not used the cost function in my application but used a scenario based behavioral planning instead. It works fine for this simulation, but may not work for complicated situations.

- You may go through this medium blog by udacity alumni that is a good read, part 3 of the blog has in

Resources

- This is a great resource based on "A path planning and obstacle avoidance algorithm for an autonomous robotic vehicle" which you might want to read.
- Here is another great paper written by our own Sebastian Thrun which discusses the practical search techniques in path planning.

Additional discussion

You might have noticed in the simulator that other cars might randomly/abruptly change lanes on rare occasions. Since sensor fusion data is not available for cars that are changing lanes, therefore, this sudden change in lanes by other cars might cause a collision with our ego vehicle. A simple workaround to minimize the incident could be to brake more aggressively when this happens based on the distance with the follow-up car, something like this,

`dT = some_value;`

```
if (detectedObstacle.velocity < ego.vel)
{
    const float deceleration = (detectedObstacle.distance <=  5.0) ? y1
                             : (detectedObstacle.distance <= 10.0) ? y2
                             : (detectedObstacle.distance <= 20.0) ? y3
                             : (detectedObstacle.distance <= 40.0) ? y4
                             : 0.0;
    pathVelocity -= deceleration * dT;
}
```

where `y1 > y2 > y3 > y4`

DOWNLOAD PROJECT

RETURN TO PATH