

[< Back to Self-Driving Car Engineer](#)

Path Planning

REVIEW

CODE REVIEW 3

HISTORY

Meets Specifications

Future SDC engineer,

You have done a great job on this project! You understand all parts of this project well. I enjoyed reviewing your work. I added some suggestions on adapting the speed of the car ahead of you when there is no chance of changing lanes in the section below.

Here are some links which I liked going through:

- [Introduction to Robotics #4: Path-Planning](#)
- [The path planning problem in depth](#)
- [A discussion on What is the difference between path planning and motion planning?](#)
- [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 unless you can guarantee that your changes will still compile on any platform.

Awesome Job in this section!! The code compiled without errors with `cmake` and `make` when running from the build directory. Great keep it up!

```

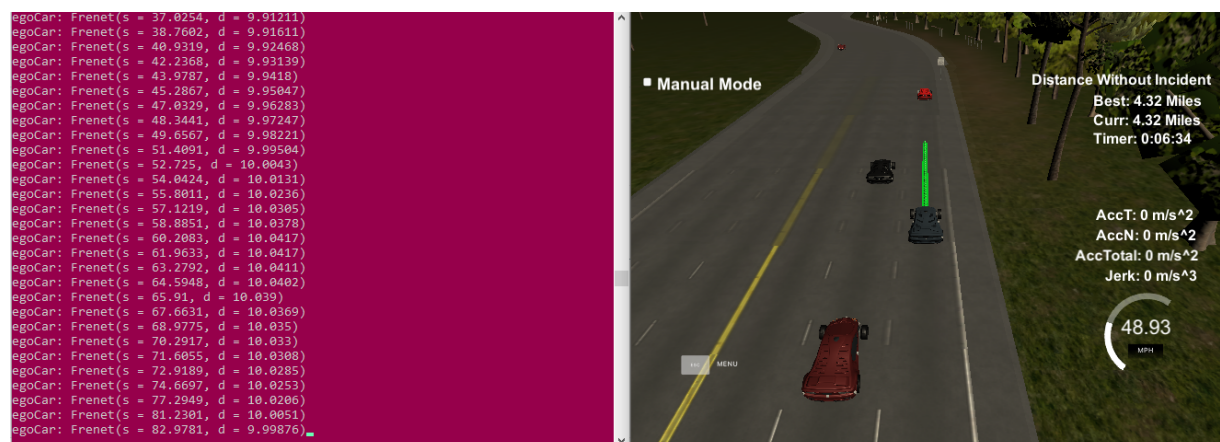
$ cd /mnt/d/Project_reviews/path_planning_SDC/20/KnollFrank-CarND-Path-Planning-Project-b1fa603/build$ cmake -G"Eclipse CD14 - Unix Makefiles" -D
CMAKE_BUILD_TYPE=Debug ../src/ && make
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Could not determine Eclipse version, assuming at least 3.6 (Helios). Adjust CMAKE_ECLIPSE_VERSION if this is wrong.
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/Project_reviews/path_planning_SDC/20/KnollFrank-CarND-Path-Planning-Project-b1fa603/build/googletest-download
Scanning dependencies of target googletest
[ 11%] Creating directories for 'googletest'
[ 22%] Performing download step (git clone) for 'googletest'
Cloning into 'googletest-src'...
Already on 'master'
Your branch is up-to-date with 'origin/master'.
[ 33%] No patch step for 'googletest'
[ 44%] Performing update step for 'googletest'
Current branch master is up to date.
[ 55%] No configure step for 'googletest'
[ 66%] No build step for 'googletest'
[ 77%] No install step for 'googletest'
[ 88%] No test step for 'googletest'
[100%] Completed 'googletest'
[100%] Built target googletest
-- Found PythonInterp: /usr/bin/python3.5 (found version "3.5.2")
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Looking for pthread_create
-- Looking for pthread_create - not found
-- Check if compiler accepts -pthread
-- Check if compiler accepts -pthread - yes
-- Found Threads: TRUE
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/Project_reviews/path_planning_SDC/20/KnollFrank-CarND-Path-Planning-Project-b1fa603/build
[ 5%] Building CXX object googletest-build/googletest/CMakeFiles/gtest.dir/src/gtest-all.cc.o
[ 11%] Linking CXX static library ../lib/libgtestd.a
[ 11%] Built target gtest
[ 17%] Building CXX object CMakeFiles/path_planning.dir/main.cpp.o
[ 23%] Building CXX object CMakeFiles/path_planning.dir/alglib/alglibinternal.cpp.o
[ 29%] Building CXX object CMakeFiles/path_planning.dir/alglib/alglibmisc.cpp.o
[ 35%] Building CXX object CMakeFiles/path_planning.dir/alglib/ap.cpp.o
[ 41%] Building CXX object CMakeFiles/path_planning.dir/alglib/datanalysis.cpp.o [ 47%] Building CXX object CMakeFiles/path_planning.dir/alglib/diffequations.cpp.o
[ 52%] Building CXX object CMakeFiles/path_planning.dir/alglib/fasttransforms.cpp.o
[ 58%] Building CXX object CMakeFiles/path_planning.dir/alglib/integration.cpp.o
[ 64%] Building CXX object CMakeFiles/path_planning.dir/alglib/interpolation.cpp.o
[ 70%] Building CXX object CMakeFiles/path_planning.dir/alglib/lsnalg.cpp.o
[ 76%] Building CXX object CMakeFiles/path_planning.dir/alglib/optimization.cpp.o [ 82%] Building CXX object CMakeFiles/path_planning.dir/alglib/solvers.cpp.o
[ 88%] Building CXX object CMakeFiles/path_planning.dir/alglib/specialfunctions.cpp.o
[ 94%] Building CXX object CMakeFiles/path_planning.dir/alglib/statistics.cpp.o
[100%] Linking CXX executable path_planning
[100%] Built target path_planning

```

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.

Well done in this section! The car passes the requirement of **4.3** miles without incidents.



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.

Well done in this part. The car adjusts the speed automatically. It adjusts the speed based on the nature of traffic. The only suggestion in this regard I have is you can have a better logic to control the speed of the car when there is no chance of changing lanes. You can have your car adapt the speed of the car ahead in that lane when there is no chance of changing lanes. This way it will be far smoother and you don't have to accelerate and decelerate so much and risk of exceeding the max acceleration and jerk limits. And this is also how it will be smoother in a real sense like if we were in a car it would be better to adapt the speed of the car ahead of us when there is no chance of changing lanes than to accelerate and decelerate continuously.

Other than everything looks great.

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

Yes, the car does not exceed the max acceleration and max jerk. It drove safely throughout the track.

Some links to get more insights

[Path Planning for Collision Avoidance Maneuver](#)

[Optimal Trajectory Planning for Glass-Handing Robot Based on Execution Time Acceleration and Jerk](#)

[This discussion on StackExchange can be of interest Which trajectory planning algorithm for minimizing jerk](#)

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

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.

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.

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".

A beautiful write-up!

 [DOWNLOAD PROJECT](#)

3

[CODE REVIEW COMMENTS](#)



RETURN TO PATH

Rate this review