UDACITY

Back to Self-Driving Car Engineer

PID Controller

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congratulations. You have finished implementing the PID controller. It can be a tricky project so you should feel proud for successfully completing it. 👍

This is a wonderful submission and all requirements have been met. Good luck on the next project. And keep doing an awesome job.

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.

Excellent. The code compiles and links without incident. You have done a good job here.

However, there were compiler warnings:

```
Scanning dependencies of target pid
[ 33%] Building CXX object CMakeFiles/pid.dir/src/PID.cpp.o
```

```
[ 66%] Building CXX object CMakeFiles/pid.dir/src/main.cpp.o
/Users/***/569961/CarND-PID-Control-Project-master/src/main.cpp:58:18: warning:
      unused variable 'speed' [-Wunused-variable]
          double speed = std::stod(j[1]["speed"].get<std::string>());
                 ^
/Users/***/569961/CarND-PID-Control-Project-master/src/main.cpp:59:18: warning:
      unused variable 'angle' [-Wunused-variable]
          double angle = std::stod(j[1]["steering_angle"].get<std::string>());
                 ^
2 warnings generated.
[100%] Linking CXX executable pid
[100%] Built target pid
```

requirement

Please have this improved in future.

Implementation

It's encouraged to be creative, particularly around hyperparameter tuning/optimization. However, the base algorithm should follow what's presented in the lessons.

Very good work. The update code in PID.cpp is correct and the calculation of total error is also correct. The implementation of the algorithm is as expected.

Reflection

Student describes the effect of the P, I, D component of the PID algorithm in their implementation. Is it what you expected?

Visual aids are encouraged, i.e. record of a small video of the car in the simulator and describe what each component is set to.

Good job. The relative importance of the components was discussed in detail and their respective effects are accurately described.

Student discusses how they chose the final hyperparameters (P, I, D coefficients). This could be have been done through manual tuning, twiddle, SGD, or something else, or a combination!

Very clear presentation. The process that led to the final parameters is well laid out and easy to understand.

Simulation

No tire may leave the drivable portion of the track surface. The car may not pop up onto ledges or roll over any surfaces that would otherwise be considered unsafe (if humans were in the vehicle).

Fantastic work here. The PID controller connects successfully with the simulator and the car accurately drives itself around the track at ~ 40-45 mph.

DOWNLOAD PROJECT

RETURN TO PATH