**Meets Specifications**

Excellent work. You put a lot of effort into this project and I think it was worth it because this is real life in Automotive and in self-driving in particular. We are dealing with a lot of unknowns and many times we have to do manual work to adjust the different parameters. PID is a typical case.

*I would love some guidance on how to implement both PID's. I struggled to get the throttle PID to work sucessfully.  
Also, any suggestions on a better way to do Twiddle. I don't feel like I was able to fully utilize the benefits (as I struggled to keep the car from crashing while using it.)*  
1- Throttle PID: The following line would set a target speed to zero:

// pid\_speed.UpdateError(cte\_speed);

Instead, you need to set a target speed. Let's say 40:

#define TARGET\_SPEED 40

pid\_speed.UpdateError(cte\_speed - TARGET\_SPEED);

2- For TWIDDLE:  
a- Try to make the code simpler. You are repeating the same code for the 3 parameters.  
b- You need to start with good parameters and then fine tune them with TWIDDLE

If you have more questions you can use the knowledge base

Congratulations on passing this project! Keep up the good work for the next project!

**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.

Code properly compiles with cmake and make.

**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 implementation, your code follows the general guidelines of the PID algorithm.

In addition to that, I liked the fact that you tried another instance of the PID controller to handle the throttle and TWIDDLE.

**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.

You clearly passed this requirement. you explained the effect of each of the P, I , D components on the behavior of the car.

However, the PD controller was incorrect. Both in terms of the parameters {0.15,0.0003,0} and the behavior. This was a PI controller. Note: The 3 parameters in order are: P, **I**, D  
 *PD Controller :  
To enable only the P and D components, the hyperparams = {0.15,0.0003,0} configuration is used. The behavior is similar, slightly more controlled. A fine tuned parameter makes the car steer stablely along the central line.*

Here are some links that have some more information on the role of each component.  
<https://www.youtube.com/watch?v=4Y7zG48uHRo&t=31s>  
<https://www.wikiwand.com/en/PID_controller#/Derivative_term>  
<http://oa.upm.es/30015/1/INVE_MEM_2013_165545.pdf>

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!

Manual tuning works fine for this project! You properly explained the logic on choosing the parameters.

For the Kd parameters it was set too high.

**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).

Your car drives fast and safely through the track!

The ride was not comfortable. There were to many small oscillations. Most probably this is why you increase the Kd this much. May be you can try to lower the Kd and the Kp.