PID Controller

Reflection

Describe the effect each of the P, I, D components had in your implementation.

1. **Proportional component.**

   Proportional component, since it’s proportional to current CTE, is responsible for turning the vehicle towards the desired path. If the proportional coefficient in PID is zero, the vehicle just goes parallel to the desired path without explicit intention to turn for decreasing CTE. An example of that (when P control is turned off) is 1.mp4.

2. **Derivative component.**

   Derivative component is responsible for dampening the oscillation caused by proportional control. Without derivative control (when derivative coefficient in PID is zero) the vehicle tends to loop from side to side. Such behavior was depicted in 2.mp4.

3. **Integral component.**

   Integral component, which is proportional to the sum of CTE over the driving time, is responsible for mitigating the systematic bias of vehicle. However, it seems the vehicle in project does not have a significant bias. The PID controller without Integral component (integral coefficient is zero) works not much worse.

Describe how the final hyperparameters were chosen.

First part of tuning hyperparameters was made manually. That was started from 0.2, 3.0, 0.004 (Pc, Dc, Ic) (like in PID control lesson) and empirically came to 0.3, 5.0, 0.0002. The rest part of the parameters tuning was made by twiddle algorithm (Pid.cpp PID::Twiddle() function). After a couple of hours it showed 0.375, 6.24778, 0.0003. That are the final parameters.

The speed control was also made by using PID control (with manually chosen 0.3, 20.0, 0.0002). However, the cte for speed depends on speed limit 60mph and steering angle in the following way: speedCte = speed – (1-abs(steer)^2)*speedLimit. This makes the vehicle go faster on straight parts of laps and go slower on turns.