PROJECT SPECIFICATION

PID Controller

Compilation

CRITERIA	MEETS SPECIFICATIONS
Your code should compile.	Code compile without errors with cmake and make.

Implementation

CRITERIA	MEETS SPECIFICATIONS
The PID procedure follows what was taught in the lessons.	Base algorithm as provided in the lesson followed.

Reflection

CRITERIA	MEETS SPECIFICATIONS

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

P (Proportional Gain) – This part makes the car steer in proportion to the cross track error (CTE). The proportional factor Kp is mulitplied by the CTE to calculate the steering angle. Using a single PID controller may result in suboptimal performance at different speeds. I therefore implemented the possibility to accomodate different setpoints and to vary the parameters linearly according to the speed the car. For example the coefficient K_i is increased linearly as a function of the speed of the car: K_i = alpha_i * v.

I - The "Integral" part corrects systemic bias such as steering drift. The integral factor Ki is multiplied by sum of all the previous CTE. The corresponding contribution to the steering angle is given by -K_i sum(cte). Thereby, biases can be mitigated, for instance if a zero steering angle does not correspond to a straight trajectory. At high speeds this term can also be useful to accumulate a large error signal quickly, for instance when the car is carried out sideways from the reference trajectory. This allows to reduce proportional gain, which causes oscillations at high speeds. It is also beneficial to limit the memory of this term to avoid overshooting. Here, we used an exponentially weighted moving average for this purpose.

CRITERIA	MEETS SPECIFICATIONS
D - The "Differential" part makes the steering angle to decrease as it reaches the reference trajectory. This allows the car to approach the trajectory "gracefully" rather than osclillating wildly. The derivative gain contributes a control output of the form -K_d d/dt cte, with a positive constant K_d.	
Describe how the final hyperparameters were chosen.	
The parameters are chosen through manual tuning. I first tried a set of small value parameters which were introduced in the Udacity lesson. However, the car drove off the track. I realized that the steering drifting is not a issue in the simulator. Therefore, I set Ki to be 0. I decided to take the pedestrian approach, because it provides an intuitive understanding of the importance of the different contributions. The algorithm used was roughly as follows. 1.single P controller for one setpoint, with K_i and K_d = 0 2.increase K_d until oscillations subside. 3.in case of crashes: find cause. a) if slow reactivity is the cause -> reduce steering angle smoothing, increase K_p or K_i b) if oscillations are the cause -> reduce K_p, K_i or brake Rounds topping out at speeds of 95mph can be acheived this way, but a much safer ride is obtained when the target speed is set to 80mph. This is what is shown in the video linked below. The final parameters I am using are : Kp = .12, Ki = 0.0, Kd = 3.5.	

Simulation

CRITERIA	MEETS SPECIFICATIONS
The vehicle must successfully drive a lap around the track.	Please check out my Youtube Video link : https://youtu.be/mfAbv5RJoqs