PID Controller

1. Files Submitted

- main.cpp – main program
- PID.h – C++ header file.
- PID.cpp – PID C++ code of implementation
- only differential parameter.mov - video of only setup differential parameter
- only integral parameter.mov – video of only setup integral parameter
- only proportional parameter.mov – video of only setup proportional parameter
- PID parameters.mov - video of setup PID parameters

2. Reflection

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

The proportional parameter controls the car to the center of the lane. If only set P component, the car will overshoot against the center line, and eventually oscillate off the track, see video file:

only proportional parameter.mov

The integral parameter or the sum of the crosstrack errors over time helps to compensate system bias. If only set I component, the car will keep in a circle rather than a straight line, see video file:

only integral parameter.mov

The differential parameter controls the car to the center of the lane, when the car has turned enough to reduce the crosstrack error, it won't just go shooting for the center line, but gradually approach to target trajectory. If only set D components, not much change to the car direction, see video file:

only differential parameter.mov

2.2 Describe how the final hyperparameters were chosen

The objective is to make the car drive straightly at the center of the lane. Basically, the hyperparameters are chosen by try and correct method:

1) Initialize the proportional component but the car starts to overshoot.

2) Set differential component to correct overshooting,

3) The integral parameter only drives the car off the track, so it stays as 0

4) Once the car moves within the lane lines, increase the parameters gradually to minimize the average CTE, the final hyperparameters were tuned as [P: 0.15, I: 0.0, D: 2.5].