PID Controller Project

The goals / steps of this project are the following:

- Implementing a PID controller in C++ to control the car from the Behavioral Cloning Project simulator.
- Once the simulator is running, it will provide CTE that will be used to compute the steering angle.
- The CTE will be used to calculate the error for each of the three components P, I and D. Then the final steering angle is calculated to compensate these errors.
- The goal is to tune the P, I and D coefficients that are multiplied with the P, I and D errors respectively to calculate the steering angle, so that the car stays on the track.

Rubric Points:

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

  The P (Proportional) component had the most obvious effect on the car when I tried different values for its coefficient. When using relatively large values, the car would make steep steering angles to compensate for the CTE and ends up oscillating all over the track.

  The I (Integral) component solves the systematic bias problem, when the car drifts over time away from the center due to misalignment of the wheels. The effect of this component was somehow subtle, maybe because I always used small numbers.

  The D (Differential) component made the car turn a little bit smoother towards the center of the lane and minimized the oscillations.

- **Describe how the final hyperparameters were chosen.**

  I started by trying different values for the P, I and D coefficients to find a good starting point for using Twiddle. My goal was to get to the first turn without getting off track. This was achieved by setting P, I and D to 0.1, 0, 0.08 respectively.

  After settling on the initial values, I ran the simulator for 500 steps and then call the Twiddle function and restart the simulator again.

  At each step, I kept track of the current total error and the best error I got so far. At any time step, if the current error exceeds the best error, I restart the simulator and call Twiddle again to update the coefficients.

  These steps were repeated until I observed that the car was driving smoothly around the track (for the 500 time steps only not the whole track).

  I reached this point when using these values: 0.1, 0, 0.435429 for the P, I and D parameters.

  Finally I disabled fine tuning the parameters using Twiddle and let the car drive around the whole track, which it did successfully without getting off.