Project 6: PID Controller

The goal of this project is to implement and tune a PID controller to maneuver the car around the race track. The simulator provides information about current crosstrack error and car speed to calculate steering angle

---

Code Requirements fulfilled as per the Rubric

- **Code compilation without errors**
- **PID procedure follows what was taught in the lesson –** The PID controller is built as per the instructions from the lesson and Q&A video in following steps
    - Initialize Kp, Ki, Kd in main.cpp (these parameters are tuned as described in the reflection section each time to check the effect on the car driving performance and stability.
    - Calculate steering value from pid class of PID.cpp
    - Initialize proportional, integral and derivative errors to zero
    - Calculate total error = Kp*p_error + Ki*i_error + Kd*d_error
    - Update PID errors based on CTE
- **Describe the effect of each of the P, I, D components –** Described in the error reflection section of the report
- **Describe how final parameters are chosen –** Described in the error reflection section of the report
- **The vehicle must successfully drive a lap around the track –** vehicle drives around the track without any violations at an average speed of 34mph

Reflection

- **Impact of each parameter on the P, I, D component**
  The following table summarizes impact of changing each parameter independently on a step response [ref: https://en.wikipedia.org/wiki/PID_controller]

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Eliminate	Degrade
K_d	Minor change	Decrease	Decrease	No effect in theory	Improve if K_d small

$$\alpha = -\tau_P \, CTE - \tau_D \frac{d}{dt} CTE - \tau_I \sum CTE$$

- As described in the lesson based on above equation -
    - P gain – directly proportional to the CTE. As we increase the magnitude of the p gain the CTE error decreases faster effectively reducing rise time. But this also creates oscillations because signal overshoots when close to target.
    - D gain – directly proportional to the rate of change of CTE. Thus, a non-zero D term reduces oscillations and is therefore accompanied with higher P gain. As the change in error decreases, the effect of D gain drops allowing a more stable driving
    - I gain – directly proportional to summation of error. This is helpful when there is a bias in the system, which is practically all real systems. In addition to the p gain when used, this eliminates the steady state error but higher magnitude of I gain could reduce system stability due to increased overshoot
- Final parameter selection
    - Step1 – [Kp = -1.0, Ki = 0.0, Kd = 0.0]
        - Car starts oscillating in first few seconds and goes off road – P gain is too high
    - Step 2 - [Kp = -0.1, Ki = 0.0, Kd = 0.0]
        - P gain is reduced until a stable non oscillating drive is achieved at least on the straight road
        - Car starts oscillating on the first left turn and goes off road
    - Step 3- [Kp = -0.1, Ki = 0, Kd = -1.0]
        - D gain is introduced to reduce oscillations
        - Car completes one lap around the track without going offroad
        - Crossed yellow line twice for a short duration
        - Fine tuning is required
    - Step 4 - [Kp = -0.2, Ki = 0, Kd = -1.2]
        - This set of gains allows the car to drive around the track without any violations
        - But the car is now less stable (more oscillations) compared to step 3
    - Car speed can be increased by removing system damping (Kd=0) and with introduction of I gain but that will increase oscillations too.