Self-Driving Car Engineer Term 2: Project-4
PID Controller

The following report for this project includes:

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

In overall, it is observed that lower *Kp, Ki, Kd* gain make the car slower to response to the cross track error (smother drive, but not very precise or in some case weaving at low frequency). It is opposite to the higher *Kp, Ki, Kd* gain where it is sensitive to the error (more responsiveness, and quick in correcting errors that make the car weaving at high speeds).

In individual case of the PID gain,
Kp reduced the error by moving the process or the car to the right direction point. But without other gains, the car would be oscillating on the track as if Kp kept following the target point.

See video "*Only_Kp.mp4*". *Kp = 0.09, Kd=0, Ki=0*
The car was oscillating and could not go pass the turn

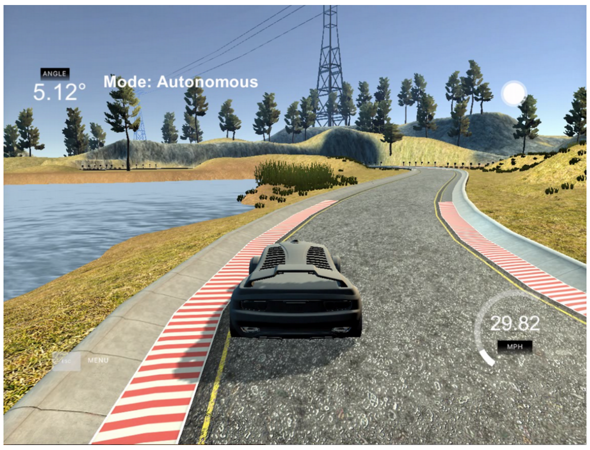
Kd term helped to eliminate/dampen the oscillation to reach the steady state. This gain makes the car react faster as the error increase by observing from the error rate.

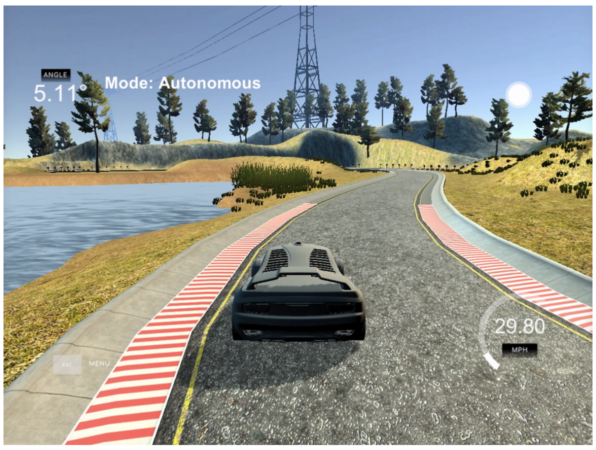
See video "Kp_and_Kd.mp4". *Kp = 0.09, Kd=0.45, Ki=0*
The oscillation was less and the car could go pass the turn.

Ki was used to minimise the systematic error/noise e.g. steering offset. In addition, it was used in this project to improve the track accuracy particularly on the sharp turns. It kept the car running inside the track better than using PD gain alone. This was by effect of the Ki term keeping track of errors overtime. However, too much Ki could make the car weaving/overshooting.

See video "No_Ki_Precision.mp4" and "With_Ki_Precision.mp4"

It can be seen that the car was positioned better.





Without Ki term
Kp = 0.12, Kd=0.45, Ki=0

With Ki term
Kp = 0.12, Kd=0.45, Ki=0.0025

2. Describe how the final hyper-parameters were chosen.

Step 1- Simulate the first initial guess using ***the Twiddling approach on the step response*** with the similar car motion model and target speed e.g 30 mph. It searched through the PID gains that responded with the best accuracy. This was not necessary be the best parameters on the track but however gave us some idea what would be the range of the gains. See *TwiddlePIDTune.py*

The output parameters are as follows:

Kp: 0.237, Kd: 1.025, Ki: 0.005
*With these gains, the car was oscillating and eventually fell of the track.

Step 2- It is the point where we could ***reduce all the gain by a factor of 2-4*** to eliminate the ossification as commonly used as a general guideline.

Kp: 0.12, Kd: 0.5, Ki: 0.002
*With these gains, the car could go round the track. However, it required some more fine tuning for certain places such going around the corners.

Step 3- I then ran these parameters again using ***the Twiddle approach on the track*** to look for the good parameters (see TunePIDTwiddle(..) in PID.h). The final set of the parameters was then derived from the experiment on many trial runs to find the best performance (responsiveness vs smoothness).

Kp: 0.12, Kd: 0.45, Ki: 0.0025
*This gain kept the car on the track at the speed of 30-32 mph. However, with this range of speeds the car would be weaving a little bit that depended on the planned path from the simulator.

Note:
The steering and speed control parameters are in main.cpp line 24-41
The car starts with the low speed 15 mph up until 200 loops then the full speed, main.cpp line 141