PID Controller

This project implements a PID Controller to drive the car around the track in the Simulator.

Proportional (P) – this parameter controls the error proportionally, so the control signal increases as this parameter value is increased. However, this results in increasing oscillations, as this parameter causes the car to steer proportional but opposite to the cross-track error.

Integral (I) – this parameter integrates, i.e. accumulates the error. This helps in reducing the steady state error. This parameter also helps increase the control signal in case there is a bias, and therefore helps driving the bias error down.

Derivative (D) – this parameter helps to add damping and decreases overshoot, as it controls the rate of change of error. It counteracts the P component's overshoot, and helps the car to approach the center line smoother.

Tuning of parameters was done manually to understand the effects of each parameter. A second PID controller (throttle_pid) was added to control the speed.

I started with setting Kp = 0.05 and setting Ki = 0 and Kd = 0. This caused a lot of oscillations, as expected, due to the P component.

Setting Kd to a large value i.e. > 1 helped reduce overshoot and reduce oscillations. Setting it even larger i.e. > 2 was the optimum value that resulted in stability.

Ki had a small effect, however it controls the bias, and this helps around turns. This only needs to be a small value, as a larger value adds to overshoot and settling time.

These parameters Kp, Ki and Kd for Proportional, Integral and Derivative can be optimized using a Twiddle algorithm.

Here are final values of parameters for both the PID controllers:

	Kp	Ki	Kd
Steering	0.06	0.004	2.15
Speed	0.07	0.002	2.47