PID Controller project write-up

Rubric's Reflection

The PID algorithm has three components,

- **Kp** (Proportional Component) –

  The proportional component is responsible for correcting the Cross Track Error based on its current value. It is affects how quickly the vehicle should steer if it is at a distance from its expected position, depending upon the difference of its current and expected position.

  If we have a low Kp value, then the vehicle comes to the correct position very slowly, therefore is not able to steer correctly along the curves/turns in the road.

  If we have a high Kp value, then the vehicle oscillates along its expected path as it tries to correct its position fast and therefore overshoots.

- **Ki** (Integral Component) –
  The integral component is responsible for nullifying any bias, if present, in the system. Its value should be the smallest compared to Ki & Kd.
  If it has a large value, then the vehicle is likely to overshoot and move away from its expected path.
  If it has a very small value, then the vehicle will nullify any bias very slowly.

- **Kd** (Differential Component) –
  The differential component operates on change of CTE (Cross Track Error) over time and is responsible for smooth convergence of vehicle along its expected track without undergoing unnecessary oscillations.
  If Kd has a low value, then the vehicle wobbles as it is not able to change steering much with reduction of CTE.
  If Kd has a high value, oscillations start increasing again.

In the project the initial values of **P**, **I**, **D** were [0.1, 0.1, 0.1]. These value led to very bad steering as the vehicle was continuously oscillating. The I reduced **P** and **I** to very low value, approx. in the range of 1e-6. Now the oscillations disappeared, but the vehicle moved out of track and wasn't able to steer properly in turns. Next I started increasing **P** and also **D** component. Finally, by trial and error method I reached the following values:

P	I	D
1e-1	1e-5	2.8

Next I implemented the **Twiddle** algorithm, which started after first 100 iterations and continued to tune the **PID** parameters upto 2500 iterations, at a step count of 80. This helped me in getting the final **PID** values for tuning the steer angle as:

P	I	D
1.2e-1	1e-5	3.5

After that I also implemented the **PID** for speed, to take the desired speed of the vehicle to *50mph*. The **PID** parameters for the speed/throttle system are:

P	I	D
1.0e-1	1e-5	3.0