PID Control

Reflection

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

First I started with just Proportional control keeping Differential and integral to zero.

Proportional Term: Here the control input is only proportional to CTE. I started with the value 0.5 and the car kept on the road for the most straight line (with oscillations) but went off the track as soon as it encountered a turn. This is due to the overshooting effect. Also greater the value of P, greater were the oscillations and eventually faster the car went off the track as soon as it saw a large turn. I finally used a value of 0.15 for P term. However, it was not possible to keep the car on track with the P term alone. So I start using Derivative term to counteract the overshooting.

Derivative Term: The derivative term makes the control input to vary the rate of change of CTE. I used it in conjunction with Proportional term in order to reduce the oscillations and smoothen out the behaviour. The starting value was 2 for D term along with previous mentioned 0.15 for P term. The car made the full track. But I also tried to play with different values to see the effect. Large values of D (~10) still made the full track but the driving was not smooth. It contains a lot of last minute corrections and too small a value of D term was not enough to dampen out the oscillations enough. I chose the value 3 as my final value.

Integral Term : The integral term is used to correct systematic bias or residual error. It considers all the past values of CTE. I started with the value 0.009 which just made the car go off the track on the straight line path itself. I reduced the value which induces some oscillations in the beginning. Finally I used the I value of 0.001.

Describe how the final hyperparameters were chosen.

The final parameters are chosen through manual trial and error method. I used the following guidelines from the link below to

https://robotics.stackexchange.com/questions/167/what-are-good-strategies-for-tuning-pid-loops

1. Set all gains to zero.
2. Increase the P gain until the response to a disturbance is steady oscillation.
3. Increase the D gain until the the oscillations go away (i.e. it's critically damped).
4. Repeat steps 2 and 3 until increasing the D gain does not stop the oscillations.
5. Set P and D to the last stable values.
6. Increase the I gain until it brings you to the setpoint with the number of oscillations desired (normally zero but a quicker response can be had if you don't mind a couple oscillations of overshoot)

The final parameters I used in my implementation are :

P : 0.15
I: 0.001
D: 3