Project 4 - CarND-PID-Control

For this project, major part is to identify what is PID parameter should be set. And I also did some research and below are some explaination from http://www.ni.com/white-paper/3782/en/:

Proportional Response:
The proportional component depends only on the difference between the set point and the process variable. This difference is referred to as the Error term. The proportional gain (Kc) determines the ratio of output response to the error signal.

Integral Response:
The integral component sums the error term over time. The result is that even a small error term will cause the integral component to increase slowly. The integral response will continually increase over time unless the error is zero, so the effect is to drive the Steady-State error to zero. Steady-State error is the final difference between the process variable and set point. A phenomenon called integral windup results when integral action saturates a controller without the controller driving the error signal toward zero.

Derivative Response:
The derivative component causes the output to decrease if the process variable is increasing rapidly. The derivative response is proportional to the rate of change of the process variable. Increasing the derivative time (Td) parameter will cause the control system to react more strongly to changes in the error term and will increase the speed of the overall control system response. Most practical control systems use very small derivative time (Td), because the Derivative Response is highly sensitive to noise in the process variable signal. If the sensor feedback signal is noisy or if the control loop rate is too slow, the derivative response can make the control system unstable

----

Reflection on this project:

Back to our project, to better understand the real effect on our project of P, I, D individually, I have tried P, I, D individually on car driving parameter, ans also record the video inside project folder:

only P record.mp4
only I record.mp4
only D record.mp4

My observation on PID impact on this project:

From these 3, we can see:

"P" will help to reduce error in a back and force way, but if too much error, it will not be able to reduce immediately.
Also it is in charge of steering direction, based on the track middle position, if it is too different, it will try to reduce the error but in a waving way, the larger the parameter is, the larger the waving effect will be. So this parameter can't be too large based on experience.

"I" stand alone doesn't have too much effect if the speed is low for this project, this will help to judge the angel in a smart way, for instance if the error is too large, then reducing error spend will be fast based on this parameter. If error is always small, then will not have too much effect but as a monitoring role.

"D": And this is to reduce the difference between direction of the steering and real time track line. It can be large, but too large will make car go out of track sometimes. So be large, but don't be too large, too small will make the effect too slow, which is also not good.

Based on large number of try and error, do it via *manual tuning*, I have chosen final parameter as below:

pid.Init(0.1, 0, 2)

So car can run properly under this parameter.

In the end, car can run nicely like video attached: ”Car record_PID.mp4”

For running the project, you can refer to below steps after your environment setup done:

- If this is not first time running, remove previous folder first using:

```
rm -R CarND-PID-Control-Project-master
```

- You may need to select ‘y’ for certain file while removing folder. Post that, you can run below code to start project running:

```
git clone https://github.com/MagicSHX/CarND-PID-Control-Project-master.git
cd CarND-PID-Control-Project-master
mkdir build && cd build
cmake .. && make
```

- Last step: making sure simulation started before running below code in ubuntu:

```
./pid
```