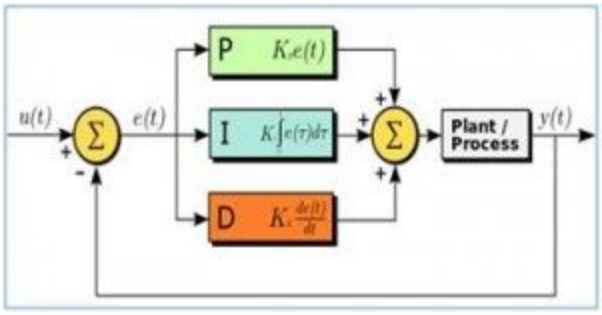
PID Controller

PID controllers are found in a wide range of applications for industrial process control. PID stands for Proportional-Integral-Derivative. These three controllers are combined in such a way that it produces a control signal.



As a feedback controller, it delivers the control output at desired levels. Before microprocessors were invented, PID control was implemented by the analog electronic components. But today all PID controllers are processed by the microprocessors. Due to the flexibility and reliability of the PID controllers, these are traditionally used in process control applications.

PID controller maintains the output such that there is zero error between process variable and set point/ desired output by closed loop operations. PID uses three basic control behaviors that are explained below.

Proportional or P- controller gives output which is proportional to current error e (t). It compares desired or set point with actual value or feedback process value. The resulting error is multiplied with proportional constant to get the output. If the error value is zero, then this controller output is zero. This controller requires biasing or manual reset when used alone. This is because it never reaches the steady state condition. It provides stable operation but always maintains the steady state error. Speed of the response is increased when the proportional constant Kc increases.

I-Controller Due to limitation of p-controller where there always exists an offset between the process variable and set point, I-controller is needed, which provides necessary action to eliminate the steady state error. It integrates the error over a period of time until error value reaches to zero. It holds the value to final control device at which error becomes zero.
Integral control decreases its output when negative error takes place. It limits the speed of response and affects stability of the system. Speed of the response is increased by decreasing integral gain Ki.

D-Controller
I-controller doesn't have the capability to predict the future behavior of error. So it reacts normally once the set point is changed. D-controller overcomes this problem by anticipating future behavior of the error. Its output depends on rate of change of error with respect to time, multiplied by derivative constant. It gives the kick start for the output thereby increasing system response.

Hypertuning parameters:

Trial and Error Method: It is a simple method of PID controller tuning. While system or controller is working, we can tune the controller. In this method, first we have to set Ki and Kd values to zero and increase proportional term (Kp) until system reaches to oscillating behavior. Once it is oscillating, adjust Ki (Integral term) so that oscillations stops and finally adjust D to get fast response.

The final video can be found here in the link
https://github.com/rakshithkeegadi/CarND-PID-Control-Project/blob/master/Video_2018-03-18%2012-27-05.flv

By trial and error method the parameters were chosen. First it started by increasing the value proportional in steps to 0.1 and found the car to be consistent between 0.22 to 0.23 values and I could not get it to perform better. The integral part ended up moving car out of the road hence very often so I had a very small value of 0.0001. The differential parameter which helped car to stay on the road was around 3.0 and the final parameters were 0.225, 0.0001 and 3.0.

Improvements:
The car still wavs a bit to left and right that can be further normalized. The speed can be improved and in tight corners the car needs to be more centered as it goes off almost to the edge of the road.

Reference and resource help:
https://www.elprocus.com/the-working-of-a-pid-controller/