During the completion of my PID implementation I experimented with a lot of different P, I, and D values. At first I didn’t understand what each was really doing, however, I eventually started changing them one at a time to find the best values and this lead me to some insight about what was really going on. For this project I altered the throttle of the car to change based on the current speed. By doing this I was able to keep the car at a comfortable 25 mph. This is the basis for my final decisions regarding the following hyper-parameters, P, I, and D. So given a different speed of the car the P, I, and D constants would most likely also need to be changed for equal results based on this.

The P hyper-parameter is matched in the PID equation with the current CTE error measurement, which is provided by the simulator. After trying different values for this parameter ranging from 0.05 all the way up to about 0.5 I started to realize that it was responsible for the intensity of the steering angle of the car given a certain intensity of CTE error. This translates in the simulator to how much the car corrects when the CTE starts to go up like in the case of a curve in the road. With a low P value, the car might not take a small increase in the CTE error seriously and the car might be late to a turn. On the other extreme the car might be a little too paranoid of a CTE change with a higher P value, causing an extreme wobbling motion even on a straight portion of the track.

Due to this, I ended up choosing a P value of 0.2. This value allowed the car to take turns in a gentle but effective manor while also minimizing the wobbling effects on the straight parts of the course.

Next up is the D hyper-parameter, which I found to be a key part of this project. With the P variable alone I noticed plenty of wobbling from my car. However, when I started to apply this D parameter the car started to self stabilize. I ended up looking into the PID equation online to find some further intuition as to why this was happening. My findings online lead me to understand it like this. The D variable is matched with the *change* in the CTE error from the frame before to the current frame. This means that the greater the change of the CTE the more the the steering angle can be increased. This is great, but it really helps more when the CTE is not changing so dramatically, like in the case of the wobbling car on a straight portion of road. In this situation the car is reading in some level of CTE error but because the error is more constant and not changing too much frame to frame the D variable tells the PID controller to reduce the steering angle. This actually helps the controller to not over correct when it is heading towards the middle of the lane, in fact the steering angle levels out when it starts to approach the center of the lane due to this D parameter.

For my D parameter I ended up choosing the value of 2.5. This value allowed the car to stabilize significantly when close to the center of the lane, and this is most clear at the beginning and the end of the track where it straightens out a bit.

Finally, there is the I parameter. This parameter is attached to the total of all of the CTE errors accumulated and needs to be extremely small since the total CTE error will always be increasing. The point of this I parameter is to allow the car to drift towards the center of the lane if it is off by a little bit or if the system has a small amount of bias.

My I term ended up being exactly 0.0. This gave me tremendously better results than anything else. Matter of fact, when this term is made anything but 0.0, my car steers of the road before reaching the first turn via “death by wobbles.” I believe this term might be better calculated instantaneously dependent on the other terms and the incoming data from the car/simulator.