PID Control Write Up

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

To control the steering angle I followed the advice of the given TODO in which it said “Calculate steering value here, remember the steering value is [-1, 1].” and the following code snippet was used to control that.

```
80
81                          if (steer_value > 1 || steer_value < -1)
82                          {
83                              std::cout << "Invalid value. Value must be between -1 and 1" << std::endl;
84                          }
85
86                          // According to the TODO statement the steer value must be -1 and 1
87                          if (steer_value > 1)
88                          {
89                              steer_value = 1;
90                          }
91                          else
92                          {
93                              steer_value = -1;
94                          }
```

P (Proportional) Component - This component calculates steering angles, and it can massively influence how the car will turn. It will potentially cause the car to overshoot and it will be multiplied by the CTE value. It will try to stay in the center and eventually sways to the right and left fast.

I (Integral) Component - This component will add to the cross track error in order to reduce the cumulative error. It will cause the car to go in circles at certain moments.

D (Differential) Component - This component will control the problem of overshooting and it will eventually make it smoother.

```
35  void PID::UpdateError(double cte)
36  {
37      // TODO: Update PID errors based on cte.
38
39      // CREDIT: PID Implementation Solution code snippet
40
41      // Integral
42      i_error += cte;
43
44      // Differential
45      double differential = cte - p_error;
46      d_error = differential;
47
48      // Proportional
49      p_error = cte;
50
51
52  }
```

Describe how the final hyperparameters were chosen.

The final hyper parameters were chosen by trial and error mainly. I wanted my integral component to be low and my differential component to be high so I started out making that very clear. The proportional component was carefully midied to make sure it doesn't overshoot. If the car overshot by a large amount I started to modify the differential component. Eventually I landed on these values: P = 0.09, I = 0.001, D = 2.00, found on line 52 of the main.cpp file.