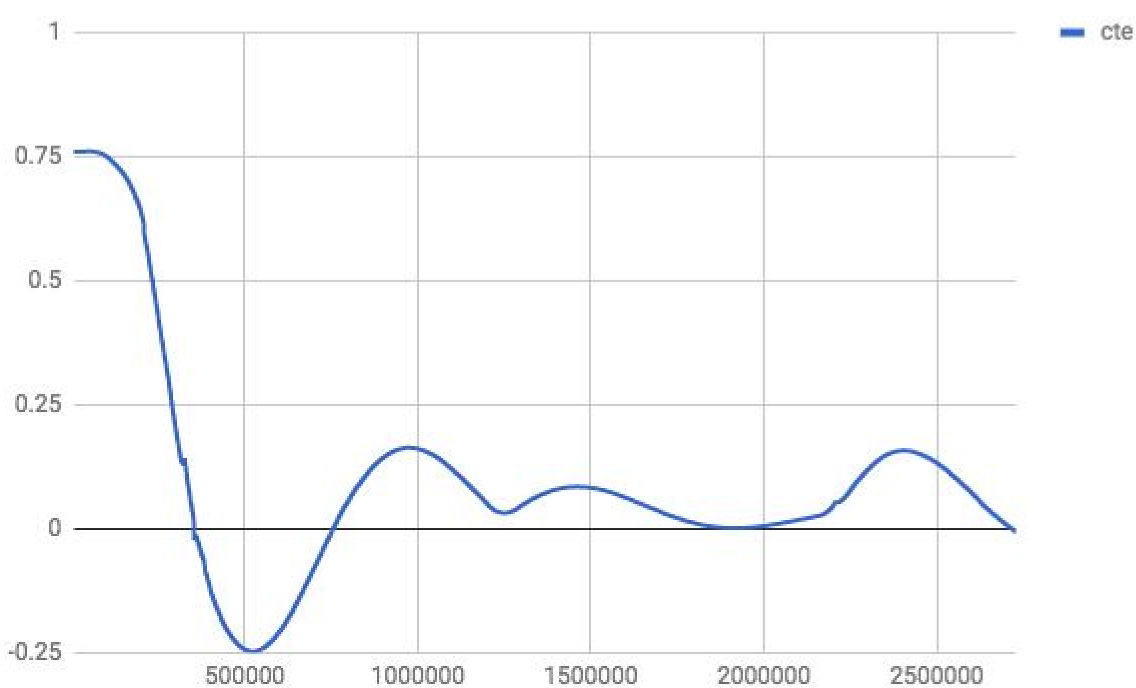
# PID Controller Parameter selection reflection.

Docs, videos and google sheets are stored in a google drive folder here.

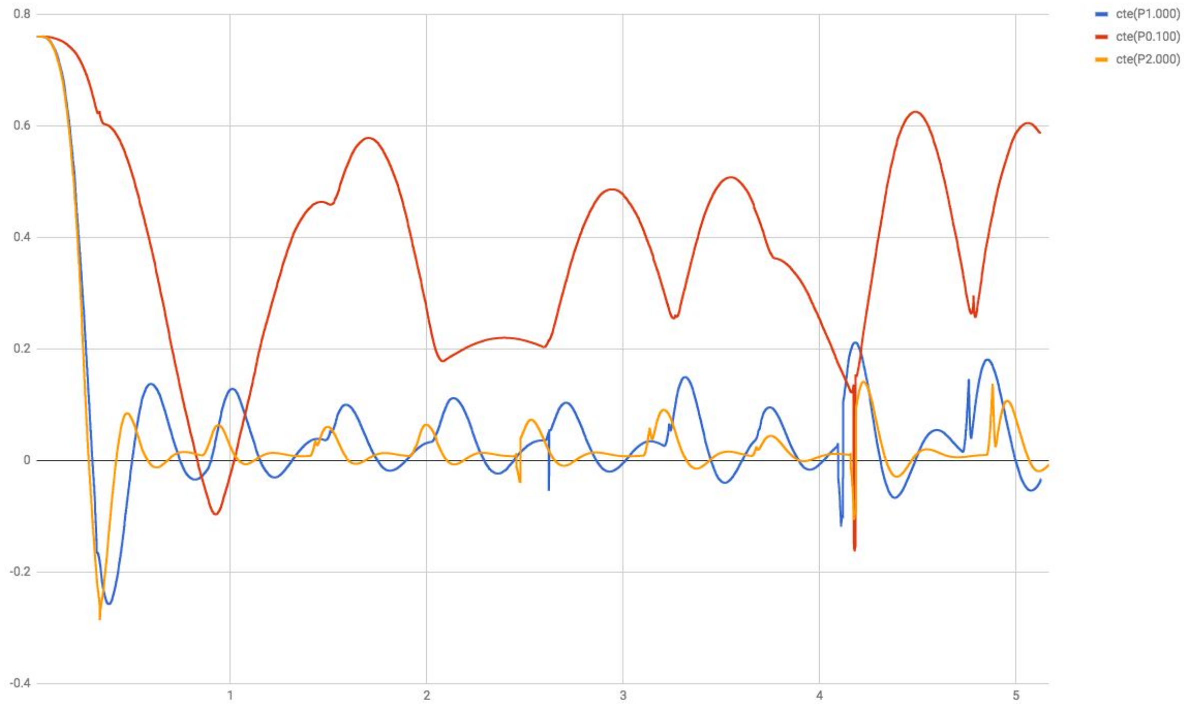
I initially approached the problem through a little bit of trial and error with the various numbers - at least for the P and D control parameters until I got it go around the track albeit not very smoothly.

## Proportional control factor effects.

Just using Proportional (0.6 factor) I would get a fast reaction but it would overshoot quite a bit. As we went faster on the track this oscillation would get worse and eventually leave the track. You can see that in action in this video.
This is a chart capturing the CTE per iteration - for Proportional control only - as the car accelerates to towards 30mph the oscillations get worse as a function of speed and end up driving off the track.



I also ran a number of experiments with different control parameter factors to get a sense of how the P value might influence the CTE over time.



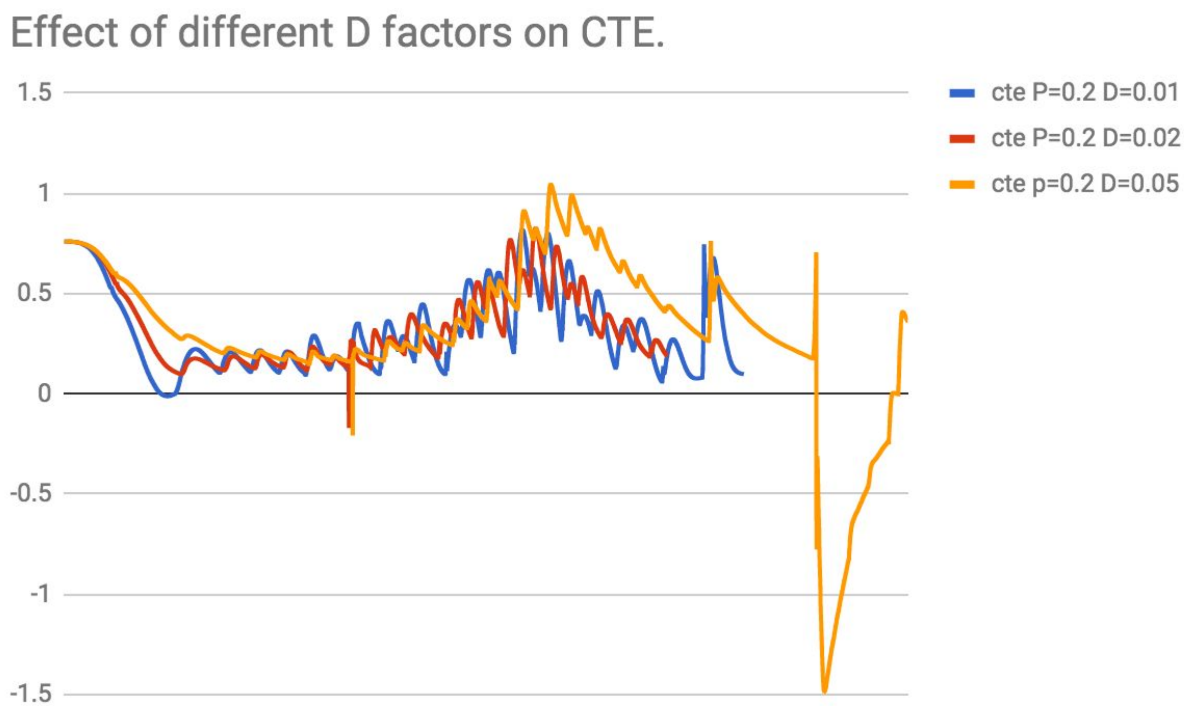

It’s possible to see in this chart that as the P control param increases in size the car can react faster to changes in the CTE , the size of the overshoot is minimized. This works well for the robot on the straight portions of the track.

However it’s not enough to just keep increasing the this proportional , as the speed increases the overshoot can lead to instability and cause the robot to leave the track especially during the turns. This is visible in the following table of videos with different values for the P factor, but no I or D Values.

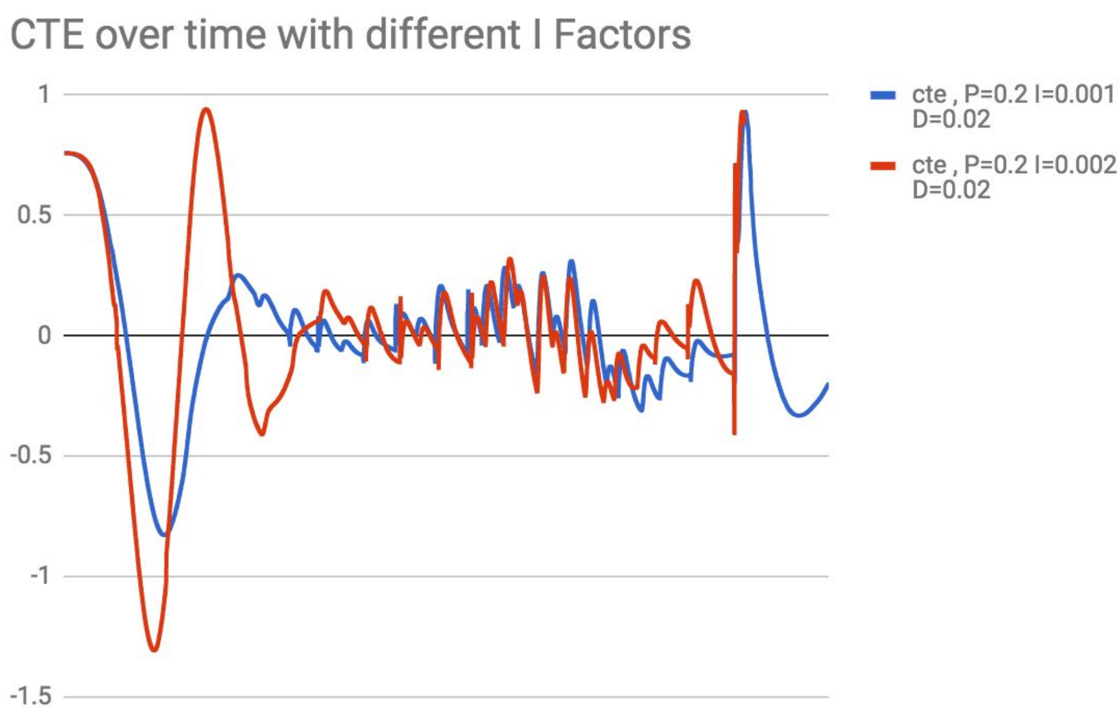
| P Parameter value | Video of simulator |
|---|---|
| 0.1 | video |
| 1.0 | video |
| 2.0 | video |
| 4.0 | video |

# Derivative control factor effects.

Adding proportional plus derivative dramatically improved the situation (video). In these cases the derivative portion significantly reduced the overshooting. However from the chart below which plots CTE over Time with different factors for the D value.
Higher values of the factor slow the rate of change of the CTE but this can have a negative effect , eventually leading to driving off the track at a point where there was a large change in CTE relatively quickly.  Also notice there is a bias towards the positive.



Finally adding in the integral portion resulted in the following type of driving.(video)
The integral portion was less noticeable when observing the driving visually.
In the chart below you can see it converging towards 0 pretty clearly.
The chart also shows how differences in the I factor can affect the stability of the CTE also. In this case even a small difference can cause the car to swing wider initially but as the speed increased the difference was less noticeable and in fact the slightly higher number seems to improve the CTE slightly.




# Final weights selection.

My approach to optimizing the control parameters was to first find a set of parameters that could drive completely around the track and then use an online continuous Twiddle implementation to tune those params.

I implemented a version of the Twiddle algorithm that measured the average squared error over N iterations while driving around the track.
I tried various sample sizes, but realized that turns especially at speed would cause a rapid increase in CTE and so it was important to include these turns in the sample size, so I increased the sample size to cover the entire track at least once. This captured all errors during turns as well as the straight sections.

I ran this version for approx 200 laps to let the twiddle algorithm converge on a good set of weights.

This is a screen capture video of a lap with the final control parameter settings.
[Full track video.]