Kevin La Ra
PID Controller
Aug 21, 2017

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

   The PID controller uses three separate control terms to compute a final control output. Each letter in the name stands for one of the components:

   P = proportional term

   I = integral term

   D = differential term

   The P term deals with the current cross-track error (CTE), i.e. the difference between the vehicle position and the center of the road. The larger the difference, the greater the angle of the wheels. This term affected how much and how fast the wheels would turn based on the CTE. A larger P coefficient was needed to negotiate the sharp turns.

   The D term dampens the effect of the P term to cause the car wheels to start straightening as it gets closer to the center track. This helped with overshooting the centerline, making the vehicle track more smoothly.

   The I term had a bigger effect on the vehicle than I thought it would. As the simulation progressed, the vehicle would track the center more closely. The steering also was smoother the longer it lasted. Because the loop is counter-clockwise, most of the CTE is to the right. The I term tended to pull the car slightly to the left of the centerline. Not much, but just enough to keep it centered better than a PD solution alone.

2) Describe how the final hyper-parameters were chosen.

   The final parameters were chosen via manual trial-and-error. From my external reading, PID tuning is more an art than a science. After working on this project I can appreciate that sentiment.

   I tried some experiments with two of the term coefficients set to 0. This allowed me to observe the effect of each term individually.

   My first honest attempt at tuning has the I term coefficient set to 0. Because of the left/right turning of the vehicle and the short duration that the simulation was going to run, I didn't think the integral term would have any effect.

I set some values for P and D and ran the simulation. When I needed more turning angle for the sharp turns, I increased the P coefficient (negative sign assumed from now on). When the car oscillated across the centerline, I added more D coefficient. After finding reasonable values with a 0.35 throttle, I worked on increasing the speed. When I did, I discovered that I also needed to increase the P and D coefficients. It was now that I decided to see what adding the integral term back into the equation would do. The car drove smoother after about a lap of input.

I decided to try a simple throttle control. I added an if statement to the throttle to set it to a different value based on the steering angle value. I tried 0, and even some negative throttles to act as a brake during the sharp turns. The problem was that the PID controller would overshoot given the decreased speed and sometimes would go off the track.

During these experiments, I learned that the PID coefficient values depend on the speed of the vehicle. This makes sense. As speed increases, the steering angle does not need to be so large to make the same correction in the same amount of time. This was too much for me to consider in this project. However, I suspect it is something that is part of autonomous car PID controllers. I also noticed that by applying the brakes or reducing the throttle the vehicle lost too much speed in the corners. Having a constant value while not achieving higher top speeds, did seem to cause shorter overall lap times.

I settled on a throttle value of 0.45 and dialed in the P, I and D values by hand. My final result was based on reasonable smoothness, without over- or understeering.
Top speed is 51.8 MPH. Lap time is approximately 50.5 seconds.

Suggestion for the future: Instead of seeing which student can achieve the highest top speed, see who can achieve the lowest lap time.