Self-Driving Car Nanodegree

P4: PID controller

Markus Buchholz

1. Introduction

In this project the main goal was to implement PID controller in order to maneuver the vehicle around the track. Controller based on provided CTE (cross track error) computes proper command (steering angle) in order to minimize the CTE.

Beside the steering angle command the throttle of the car can be also controlled. Robric does not specify the minimum speed to pass the project so in order to secure successfully car drive around the track (default speed = 0.3) only ordinary setup of PID was applied.

However, successful implementation of PID controller for default speed does not guaranty the secure drive while the speed is higher. In this case (higher speed then 0.6 of max throttle) the ordinary PID setup was failed. The car lost the control and rolled over from the track. Situation forced the author to look on additional solution securing safe drive with higher speed (as fast as possible). Therefore it was decided to deploy additional software function, which controls the throttle of the car (performs the brake). Normally the car receives the maximal value of the throttle but in case of situation when the combination of CTE and speed is larger the expected so the brakes are activated and the speed is lowered. Additional implementation allows to accelerate car to 85mph.

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

Each component of the implemented PID controller had considerable effect of car motion performance. The project experience can be summarized as follows:

P – Proportional part of controller allows to control the raising time and system stability (overshoots). The higher value of this part of controller is the faster responsivity the system is. In the same time Kp coefficient influences also on system stability, which decreases (higher overshoots). Proper tuning should secure optimization between these two values (time system response and stability of the system).

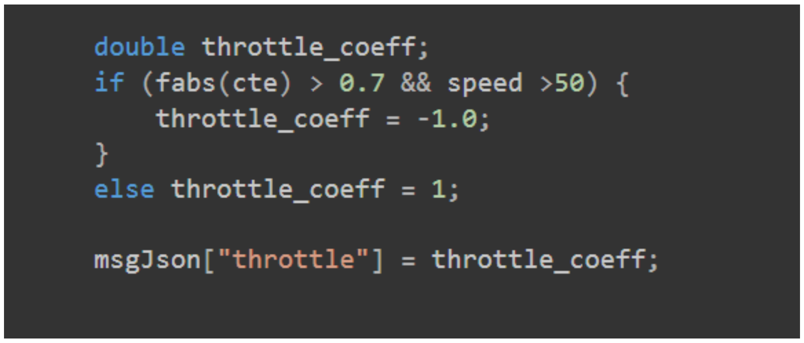
This portion of the controller formula seemed to affect the quickness/aggressiveness of the steering. When the vehicle was trying to correct its path from the cross track error (CTE) a higher "P" coefficient made the car compensate much harder and faster to the correct location on the track. A higher value here also caused more of an over steer problem. It took some trial and error to find the balance of steering fast enough to make the turns on the track, while keeping over steer to a minimum.

I – the integral part of controller takes into consideration all past system error values. The biggest disadvantage of applying this part of controller is that even small errors can be amplified. This can lead that the control command can be too large. In this project the environment (conditions) change very fast, therefore – as it was verified in the project, this part of controller seems to be not very useful. It was verified that each value above 0 negatively influences on system performance. In this project, this coefficient was abandoned.

D – In order to minimize the settling time and overshoots it was necessary to utilize the derivative part of PID controller, which in this case predicts the future errors by linearly extrapolating the change in error value. Larger value of this coefficient makes the system damped. Adjusting proper value of Kd decreases sufficiently overshoots and make system stable.

3. Describe how the final hyperparameters were adjusted

The final PID controller hyperparmaters were adjusted by use of manual tuning. As it was given in introduction the default throttle equal 0.3 was taken into consideration. At the of tuning process it was realized the procedure can take long time to find proper values securing the project goal (setpoint of the controller change very often). The tuning approach was started by adjusting the Kp parameter where the other Kd and Ki were adjusted to zero. First investigations were focused on verification of system response and stability. After adjusting several values of Kp between: 0.5 – 10 it was quickly realized that higher value of Kp implies faster system (reaction on CTE) but also influences on deterioration of system stability (visible large overshoots implies the car rolls over from the track). Author followed general rule and adjusted Kp to reduce rise time. Anyway, using only Kp parameter it was not possible to keep the vehicle on whole track lap. This time the value was adjusted to 0.3.
In order to minimize the settling time and overshoots it was necessary to utilize the derivative part of PID controller, which in this case predicts the future CTE by linearly extrapolating the change in error value. At the beginning the system response for verify on small values, trying to figure out proper response time securing lowest overshoots. Values up to 4 seems not to good enough to secure the car staying on track. Larger values, starting from 6 guaranty fulfillment of project requirements. That time it was noticed that reduction Kp to value 0.2 influences smoother car motion on the track. The preliminary successful setup of PID was Kp=0.2, Kd=6, Ki=0,but the author decided to reduce more overshoots and enlarged lastly the Kd to value 10. Bigger values make system very damped. Finally the integrational part of the controller was investigated, which takes into consideration all past system error values. Adjusting the value to 0.5 made car not possible to start. Accumulated error worked as a bias rolling over the car from the track. Decreasing the value made situation better. Bering in mind the previous successful controller setup it was decided to abandon usage of this part of controller (Ki=0).
As it was written in introduction, the author decided to check the PID setup on other speed variant then default and applied first the throttle = 0.6. Unfortunately, long investigation process with finding proper adjustment of PID values forced to abandon this search. In authors case it was not possible to choose the PID setup which would secure to stay the car on the track. However, the author decided to solve this problem by applying an extra function, which controls the speed (by applying the brakes) while the CTE is large enough to make the system instable and roll over the vehicle from the track. The function is given below. Author used iteration approach in adjusting the correct values of the “if” function. The brakes are active if the CTE is larger then 0.7 and speed of the vehicle exceeds 50mph.



```
double throttle_coeff;
if (fabs(cte) > 0.7 && speed >50) {
    throttle_coeff = -1.0;
}
else throttle_coeff = 1;

msgJson["throttle"] = throttle_coeff;
```

Project PID setup approach can be summarized in following table:

Step	Kp	Kd	Ki	
1	0.3	0	0	
2	0.3	6	0	
3	0.2	6	0	
4	0.2	10	0	BEST CHOICE FOR THROTTLE = 0.3
5	0.2	10	0.5	
6	0.2	10	0.005	
7	0.2	10	0	BEST CHOICE FOR MAX THROTTLE

4. Conclusion

In this project PID controller was deployed and hyperparameters were tuned. Developed controller allows moving safely the vehicle on the track. Beside visible overshoots the car does not pull over the track. The throttle of the car is controllable so when the dangerous for the vehicle motion overshoots appear so the brakes can be engages reducing the speed to taking back car on planed path.

PID controller does not include the dynamic model of the car so it does not guaranty the proper performance when the environment will change (or speed as I this project). In this project it was possible to choose correct values of PID (securing safety movement of the car) but there is no guaranty that this controller setup will succeed when the other configuration (shape or profile) of the track will be taken into consideration (car has to be checked on different tracks).