Track Maneuvering using PID-Control

Done by: Wael Farag

1. Objective

To design and implement a Proportional-Integral-Differential (PID) controller for an autonomous vehicle to successfully maneuver around a complex track which has lots sharp turns. The controller receives measurements from the vehicle such as:

1. The <u>CTE</u> (the cross track error) which represents the misalignment of the vehicle with respect to the center of the track at a given instance.
2. The vehicle speed at the given instance.
3. The vehicle orientation angle (-ve for left and +ve for right).

The PID controller then uses some of the above information to produce a steer (angle) command to the vehicle in addition to a throttle command (speed) if required.

2. The PID Controller Design

The PID controller used the CTE and produces the steer command after apply proportional, integral and differential control to it in terms of K_p, K_I and K_d coefficients respectively. Therefore, the main design effort is to carefully tune these three coefficients to get the best possible performance. The performance can be simply defined as to let the vehicle to follow the track center line as closely as possible with the lowest aggregated CTE throughout the entire trip. Therefore, the main goal of the controller is to minimize the aggregated CTE (the objective function) as given by equation (1).

$$Objective\ Function = \text{mininmize}\{MSE\} = min\,\frac{1}{N}\sum_{i=0}^{i=N} CTE_i^2 \qquad (1)$$

The PID coefficients are tuned using the following procedure:

1. The tuning will be done through the ad hoc technique “trial and error”. I didn’t use an automating algorithm like “Twiddle” as it the tuning process in this case will be kind of opaque (will not know exactly, how each change in parameter affected the performance). I preferred to use a manual approach that incorporates intuition and my experience in the field.

2. The time span at which the objective function is evaluated is selected to be considerably big (N=10,000). Ten thousand samples (individual CTE measurements) are used to evaluate the performance indicator (given in equation (1)). The number of samples are big enough to get a kind of accurate evaluation as it represents the measurements from the vehicle while going around the track more than once (several times in case of the higher speeds).
3. First, I set the throttle value as a constant throughout the performance measurement.
4. The tuning process starts by setting the throttle value at a low value (= 0.1 in our case) and the coefficients as follows (K_p = 0.1, K_I = 0.0 and $K_D = 0.0$).
5. I run the simulation for several iterations (at least two) in which each iteration produces an individual measurement of the objective function. And then take the average of these measurements to produce the designated performance indicator (average Mean Squared Error "MSE").
6. In the next set of iterations, I then increment the K_p a small amount based on intuition while keeping the other coefficients constant, then calculate the new performance indicator. If the performance gets better then keep incrementing K_p, otherwise; return back tp the previous value and move on to the next coefficient K_I by incrementing it a small amount (based on intuition as well).
7. After the performance stops improving with K_I, then K_D will be picked and getting step by step incremented till the performance indicator stops improving.
8. After tuning the three coefficients for the throttle value = 0.1, their reached values will be the starting point of the next throttle increment iteration.
9. The next throttle value iteration: the throttle value got incremented to say (=0.15), and steps from 5 → 8, got repeated while holding the throttle value constant at 0.15.
10. Keep repeating steps 5 → 9 while incrementing the throttle value (=> 0.2, 0.25, 0.3, 0.35, 0.4 … etc.).
11. The tuning stops when PID coefficients fails to control the vehicle (the vehicle jumps out of the track) at an upper throttle value and the performance indictor shows a large unacceptable value of *MSE*.
12. The final stet of coefficients {K_p, K_I and K_D} that keep the vehicle within the track and an utmost reached value of throttle is considered the accepted tuned design of the PID.
13. These values are then used at all levels of throttle up to the utmost reached value.

3. Tuning and Testing Results

The tuning of the PID controller is carried out using the upper procedure. Table 1 below show samples of the "trial and error endeavors" to reach fine-tuned values for the controller coefficients {K_p, K_I and K_d}.

Table 1 Fine tuning the PID controller using "trials and errors".

Trial #	Throttle	K_p	K_I	K_D	Average MSE	Comments
1	0.1	0.18	0.00005	0.36	0.246889	
2	0.1	0.18	0.00005	0.5	0.243526	
3	0.15	0.18	0.00005	0.5	0.280253	MSE jumps higher due to the increment in throttle value.
4	0.15	0.18	0.00005	0.6	0.270583	
5	0.15	0.18	0.00005	0.7	0.277959	
6	0.15	0.18	0.00005	0.9	0.246081	
7	0.2	0.2	0.00005	1.0	0.293409	
8	0.2	0.2	0.0001	1.0	0.253787	
9	0.25	0.2	0.0001	1.0	0.531351	MSE jumps higher due to the increment in throttle value.
10	0.25	0.2	0.0000	1.0	0.509472	
11	0.25	0.2	0.0000	1.25	0.335846	
12	0.25	0.2	0.0000	1.5	0.300634	
13	0.25	0.2	0.0000	1.75	0.280235	
14	0.25	0.2	0.0000	2.0	0.269807	
15	0.25	0.3	0.0000	2.0	0.209112	
19	0.25	0.4	0.0000	2.0	0.489602	An increment in K_P that caused big loss in performance.
20	0.25	0.4	0.0000	2.5	0.314796	
21	0.25	0.4	0.0000	3.0	0.227320	
22	0.25	0.4	0.0000	3.5	0.192307	
23	0.25	0.4	0.0000	4.0	0.159206	
25	0.25	0.4	0.0000	4.5	0.146784	
26	0.25	0.4	0.0000	5.0	0.129811	
27	0.25	0.4	0.0000	5.5	0.119762	
28	0.25	0.4	0.0000	6.0	0.123671	Major improvements of performance after several increments of K_D
29	0.3	0.4	0.0000	6.0	0.161787	MSE jumps higher due to the increment in throttle value.
30	0.3	0.4	0.0000	6.5	0.166268	
31	0.3	0.45	0.0000	6.5	0.165845	
32	0.3	0.35	0.0000	6.5	0.147639	
33	0.3	0.30	0.0000	6.5	0.156575	
34	0.3	0.4	0.0000	6.0	0.150308	
35	0.3	0.4	0.0000	7.0	0.154149	
36	0.3	0.4	0.0000	6.0	0.166513	
37	0.3	0.35	0.0001	6.5	0.116793	Adding a bit of K_I at this stage helped improve the performance significantly.
38	0.3	0.35	0.0002	6.5	0.115610	
39	0.3	0.35	0.0005	6.5	0.116543	This set of coefficients are considered the final tuning.

The final tuning of PID controller is given in trial #39 where (K_p = 0.35, K_I = 0.0005 and K_D = 6.5). Using these coefficients the controller is tested under different throttle values and the performance results are showing in Table 2 below.

Table 2 Testing results for the PID controller @ different speeds.

Throttle	K_p	K_I	K_D	Average MSE	Comments
0.1	0.35	0.0005	6.5	0.062864	~ 12 miles/hour
0.15	0.35	0.0005	6.5	0.072709	~ 17 miles/hour
0.2	0.35	0.0005	6.5	0.080815	~ 23 miles/hour
0.25	0.35	0.0005	6.5	0.098465	~ 28 miles/hour
0.3	0.35	0.0005	6.5	0.1165430	~ 34 miles/hour
0.35	0.35	0.0005	6.5	0.148176	~ 39 miles/hour
0.4	0.35	0.0005	6.5	0.160981	~ 44 miles/hour
0.45	0.35	0.0005	6.5	0.207031	~ 49 miles/hour
0.5	0.35	0.0005	6.5	0.331020	~ 55 miles/hour

4. Discussion and Conclusion

The following are some conclusive remarks on the project and the work done:

1. The PID controller has a very simple structure however, it very effective in many control problems. Therefore, it is the most widely used approach by far in industry (>90%).
2. The main problem with the PID, is its tuning. There is no theory or criteria that proves that you have reached to the optimal value for the coefficients {K_p, K_I and K_D}. All the methods of tuning are mainly based on extensive search with the incorporation of intuition and experience.
3. From my point of view, using transparent methods based on extensive "trial and error" endeavors guided by a numerical performance indicators is the best approach; as it lets you understand the problem at hand much deeper. Furthermore, it allows you to incorporate your intuition and experience which reduces a lot of the search space; and consequently allows you at the end to be more effective.
4. The problem @ hand is a tracking problem in which the set-point (the track center position) keeps changing continuously with time. In such kind of problems, the differential controller (K_D) proved to be very effective and it is designed in principal to track changes. In our case, it played the dominant role. The differential component counteracts the tendency for oscillation or overshoot around the track center line. By properly tuning K_D, it will cause the vehicle to approach the center line smoothly with much lower oscillations.
5. The proportional controller (K_p) is necessary to feedback the error to the controller with corrective action but it is not playing the principle role in improving the performance in this project. The

proportional component had the most directly observable effect on the vehicle's behavior. It causes the vehicle to steer back to the track trying to reduce the CTE.

6. However, the integral controller (K_I) shows to be ineffective is our case (due to the continuous change of the setting point) and in many scenarios it has a detrimental effect. The integral component counteracts a bias in the CTE and speeds up the approach to the center line, however, in our case, the center line keep moving with respect to the vehicle which in many cases may cause the integral controller to produce oscillations.

5. Suggested Improvements

The following list summaries the suggested improvements:

1. Adding another PID controller to control the throttle in such a way to reduce the speed in sharp curves and to allow full thrust in straight portions of the track. This controller should take the vehicle angle as an input and produces the “throttle value”.
2. Use of Adaptive PID control approaches to accommodate the wide range of vehicle speed. One of the popular approaches is the Gain Scheduling (GS) adaptive control, where we use several gain sets (3 or 4 sets), each will be used @ a specific speed range.
3. Use of “Twiddle” algorithm to further fine tune the already found coefficients and get the performance as better as possible.