

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/335775497>

# Complex Trajectory Tracking Using PID Control for Autonomous Driving

Article in *International Journal of Intelligent Transportation Systems Research* · September 2019

DOI: 10.1007/s13177-019-00204-2

---

CITATIONS

3

---

READS

74

1 author:



Wael A. Farag  
Cairo University

45 PUBLICATIONS 544 CITATIONS

SEE PROFILE



# Complex Trajectory Tracking Using PID Control for Autonomous Driving

Wael Farag<sup>1,2</sup>

Received: 24 October 2018 / Revised: 12 March 2019 / Accepted: 1 September 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

In this paper, a Proportional-Integral-Differential (PID) controller that facilitates track maneuvering for self-driving cars is proposed. Three different design approaches are used to find and tune the controller hyper-parameters. One of them is “WAF-Tune”, which is an ad hoc trial-and-error based technique that is specifically proposed in this paper for this specific application. The proposed controller uses only the Cross-Track-Error (CTE) as an input to the controller, whereas the output is the steering command. Extensive simulation studies in complex tracks with many sharp turns have been carried out to evaluate the performance of the proposed controller at different speeds. The analysis shows that the proposed technique outperforms the other ones. The usefulness and the shortcomings of the proposed tuning mechanism are also discussed in details.

**Keywords** PID control · Self-driving Car · Autonomous driving · PID tuning

## 1 Introduction

Increasing safety, reducing road accidents, improving energy efficiency and enhancing comfort and driving experience are the major motivations behind equipping modern cars with Advanced Driving Assistance Systems (ADAS) [1–5]. In the past couple of decades, major car manufacturers introduce many sophisticated ADAS functions like Lane Departure Warning (LDW), Lane Keep Assist (LKA), Electronic Stability Control (ESC), Anti-lock Brake System (ABS), etc. Both LKA and ESC functions are examples of how important for the car to follow and track the lane or road boundaries accurately and on time.

Future ADAS functions like Collision Avoidance, Automated Highway Driving (Autopilot), Automated Urban Driving [6], Automated Parking, Connected Vehicles [7] and Cooperative Maneuvering requires more and more fast and reliable road boundaries follower functionalities, which should be designed to maintain the control over the vehicle in extreme circumstances, and being very effective at high

speeds [8]. The road boundaries follower functionality requires the localization of the road, the determination of the relative position between vehicle and road/lane centerline, and the analysis of the vehicle’s heading direction. The mentioned systems represent incremental steps toward a hypothetical future of safe fully self-driven vehicles [9, 10].

There are two categories of vehicle control [11]. The first category is the longitudinal control which deals with the movement of the vehicle in the forward and backward directions (responsible for regulating the vehicle cruise velocity) [12]. The second one is the lateral control which deals with sideways movements perpendicular on the vehicle’s heading, in other words, the steering of the vehicle to follow a given trajectory [13]. Usually, two separate controllers are employed to deal with each of them. The focus of this paper is on the second category with the objective of minimizing the tracking error.

In previous literature, various studies have been carried out to explore different hypotheses and techniques. These studies incorporate the PID control strategies [14, 15], the predictive control paradigms [16] and the model reference adaptive control methods [17, 18].

Moreover, some of these studies focused on intelligent approaches like fuzzy control techniques [19, 20] and neural-network based control strategies [21, 22]. In addition to that, some unpopular techniques are also been exercised such as the SVR (support vector regression) method [23], the fractional-order control method [24], etc.

✉ Wael Farag  
wael.farag@aum.edu.kw; wael.farag@cu.edu.eg

<sup>1</sup> College of Engineering & Technology, American University of the Middle East, Egaila, Kuwait

<sup>2</sup> Electrical Power Engineering Department, Cairo University, Giza, Egypt

However, even though, as mention above, there are a variety of control techniques with good performance, still, the PID control is the most popular as still attracting attention. The PID control has several points of strength such as a straightforward structure, overall good control performance, robust design and simple implementation [25]. Moreover, the control methods introduced above do not consider the real-time environment of the driving, the constraints of the automotive applications and deals with it as a hypothetical problem. The controller performance should be fast enough in real-time without much overhead that consumes computational resources (big software footprint) and may suffer from stability problems, therefore, offline-tuning methods are advantageous in this regard than adaptive or intelligent techniques.

However, the PID method needs supportive algorithms to find and fine-tune its hyper-parameters. It is challenging to find good optimized values for these hyper-parameters that well fit the environment caused by the complexity of vehicle dynamics, uncertainty of the external disturbances and the non-holonomic constraint of the vehicle [9].

Here in this paper, a further step towards the prospects of autonomous driving is presented. The main objective is to attain a car capable by itself of following a pre-defined trajectory or route generated by the prospected path planner of the autonomous car, in both highway and urban traffic with the highest comfort level. Accordingly, the developed PID controller is suitable to be employed at level of autonomy of three and above.

The proposed control system uses sensors to provide the actual speed of the controlled car and the relative position of the car with respect to the supplied trajectory. The controller then calculates the optimized actions on the steering wheel. These actions are controlling the direction of the vehicle, and a Proportional-Integral-Derivative (PID) control algorithm is employed to compute them.

## 2 Overview of PID Control

The Proportional Integral Derivative (PID) control technique [24] is a mechanism widely used in various industrial control applications. As an example, one of these uses is in robotics, where its main function is to smoothen the robot's movement inputs in a way that it allows the robot to reach its objective state with as little vibration or jitter as possible.

The output of a PID depends on three hyperparameters, one for each term of the controller abbreviation (P-I-D). These hyperparameters are specific to each application, which in our case is driving a car smoothly around a test track.

**The Proportional Term** The proportional term [25, 26] produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant  $K_p$ ,

called the proportional gain constant. The proportional term is given by:

$$P_{out} = K_p * Error \quad (1)$$

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable. In contrast, a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances. Tuning theory and industrial practice indicate that the proportional term should contribute to the bulk of the output change. The chart in Fig. 1 shows how different values of  $K_p$  would impact the system response.

**The Integral Term** The contribution from the integral term [26] is proportional to both the magnitude of the error and the duration of the error. The integral term contribution is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain ( $K_I$ ) and added to the controller output.

$$I_{out} = K_I * [Error \, dt \quad (2)$$

The chart in Fig. 2 shows how different values of  $K_I$  would impact the system response.

**The Differential Term** The derivative of the process error is calculated [25, 26] by determining the slope of the error over time and multiplying this rate of change by the derivative gain  $K_D$ . The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain,  $K_D$ . The derivative term is given by:

$$D_{out} = K_D * \frac{d}{dt} (Error) \quad (3)$$

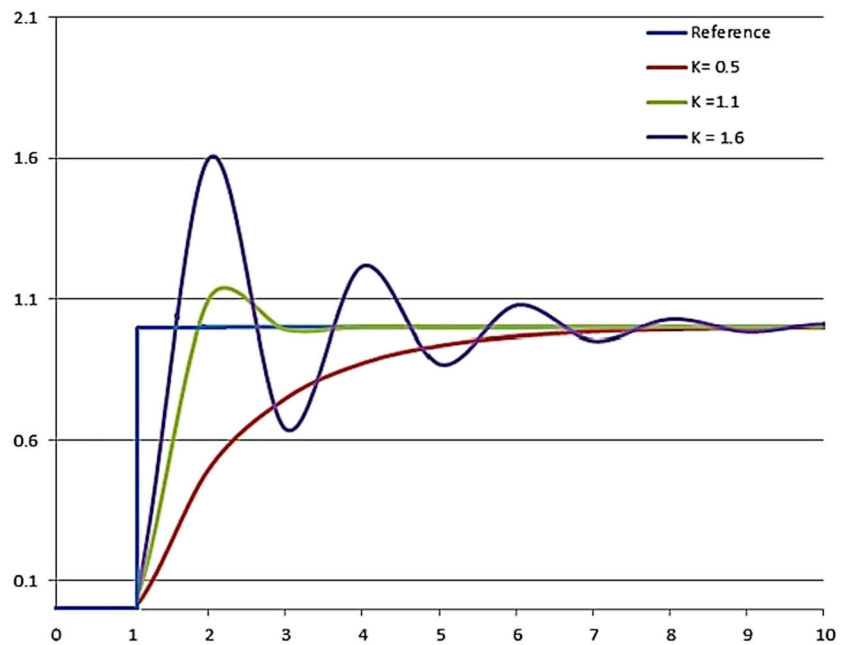
The chart in Fig. 3 shows how different values of  $K_D$  would impact the system response.

In vehicle control, the velocity and steering set-points are picked up by the vehicle's low-level PID controllers. Separate PID loops for speed and heading control are implemented in the current vehicles platform. The steering PID controller determines the steering angle based on the input heading angle in human-driven vehicles or the cross track error in self-driven vehicles. While two separate PID controllers for the accelerator and the brake works together to maintain the desired velocity. Additionally, all low-level hardware controller PID's are written in C/C++ languages with an average update rate of around 100 Hz.

The overall control action is shown in Fig. 4 can be given as

$$PID_{out} = P_{out} + I_{out} + D_{out} \quad (4)$$

**Fig. 1** The  $K_p$  effect in the system response [22]



### 3 The Vehicle Model

In this paper, the Kinematic Bicycle Model [9] is used to simulate the behavior of the self-driving car. The nonlinear continuous time equations that describe a kinematic bicycle model shown in Fig. 5 in an inertial frame are:

$$\dot{x} = v * \cos(\psi + \beta) \quad (5a)$$

$$\dot{y} = v * \sin(\psi + \beta) \quad (5b)$$

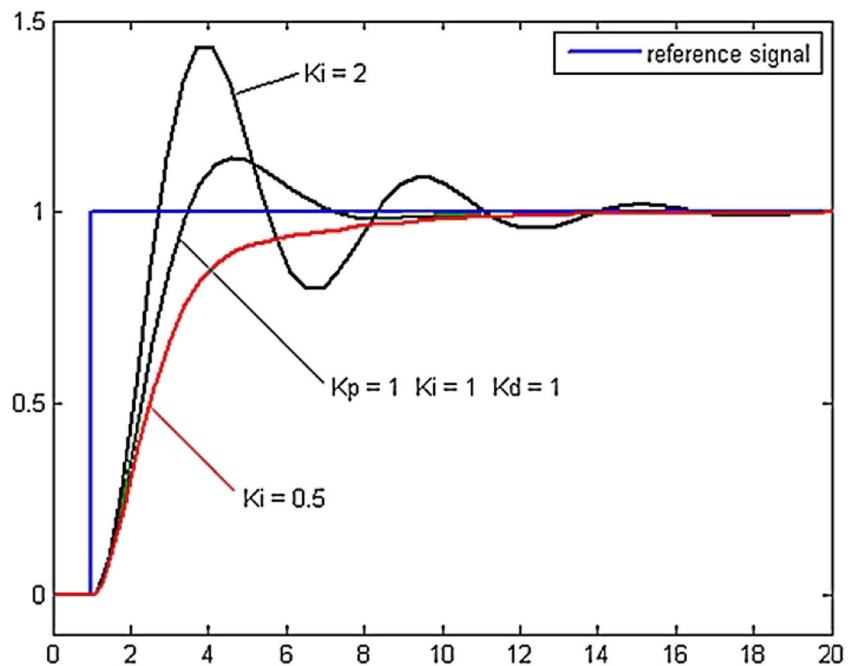
$$\dot{\psi} = \frac{v}{l_r} * \sin(\beta) \quad (5c)$$

$$\dot{v} = a \quad (5d)$$

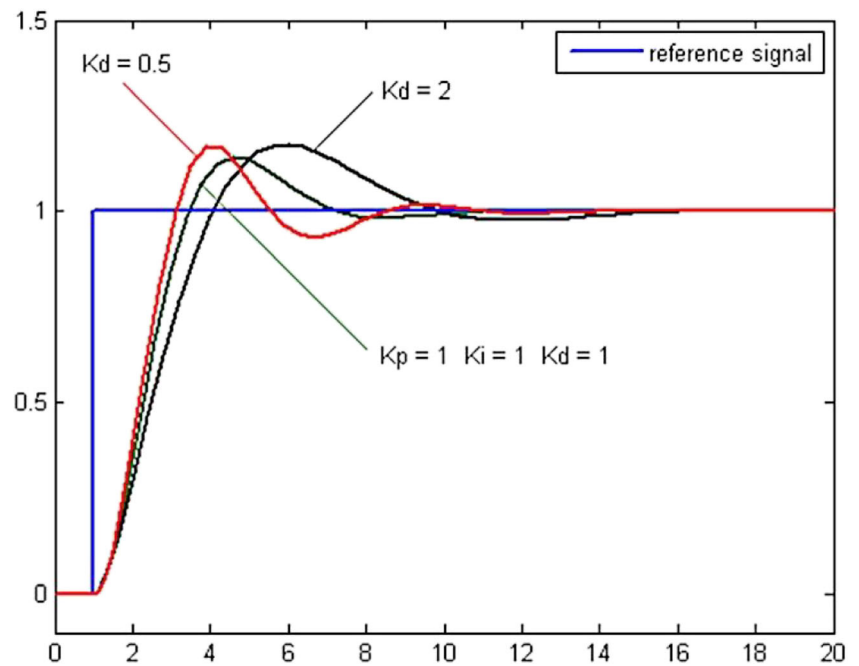
$$\beta = \tan^{-1} \left( \frac{l_r}{l_f + l_r} * \tan(\delta_f) \right) \quad (5e)$$

where  $x$  and  $y$  are the coordinates of the center of mass in an inertial frame (X, Y) [21].  $\psi$  is the inertial heading and  $v$  is the speed of the vehicle.  $l_f$  and  $l_r$  represent the distance from the center of the mass of the vehicle to the front and rear axles, respectively.  $\beta$  is the angle of the current velocity of the center

**Fig. 2** The  $K_i$  effect in the system response [27]



**Fig. 3** The  $K_D$  effect in the system response [24]



of mass with respect to the longitudinal axis of the car.  $a$  is the acceleration of the center of mass in the same direction as the velocity. The control inputs are the front and rear steering angles  $\delta_f$  and  $\delta_r$ . Since in most vehicles the rear wheels cannot be steered, we assume  $\delta_r = 0$ .

Compared to higher fidelity vehicle models [28], the system identification on the kinematic bicycle model is easier because there are only two parameters to identify,  $l_f$  and  $l_r$ . This makes it simpler to port the same controller or path planner to other vehicles with differently sized wheelbases.

#### 4 The Proposed PID Tunning Method

In order to design and implement the proposed Proportional-Integral-Differential (PID) controller for an autonomous vehicle to successfully maneuver around a complex track which

has lots of sharp turns, the following measurements are received by the controller:

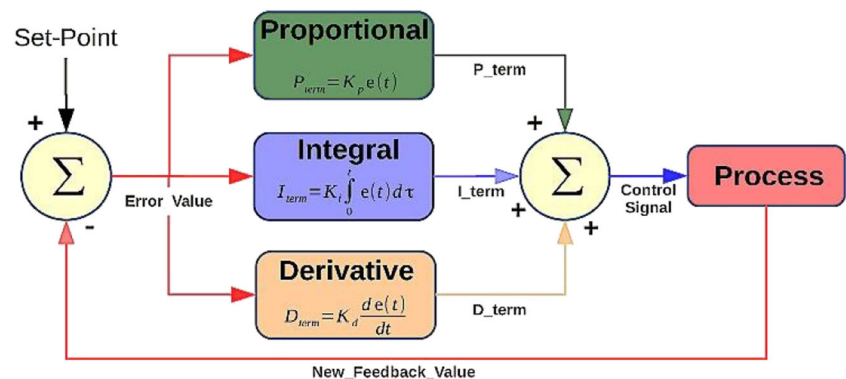
- 1) The CTE (the cross track error) which represents the misalignment of the vehicle with respect to the center of the track at a given instance and can be given by:

$$CTE = \text{desired position} - \text{vehicle position} \quad (6)$$

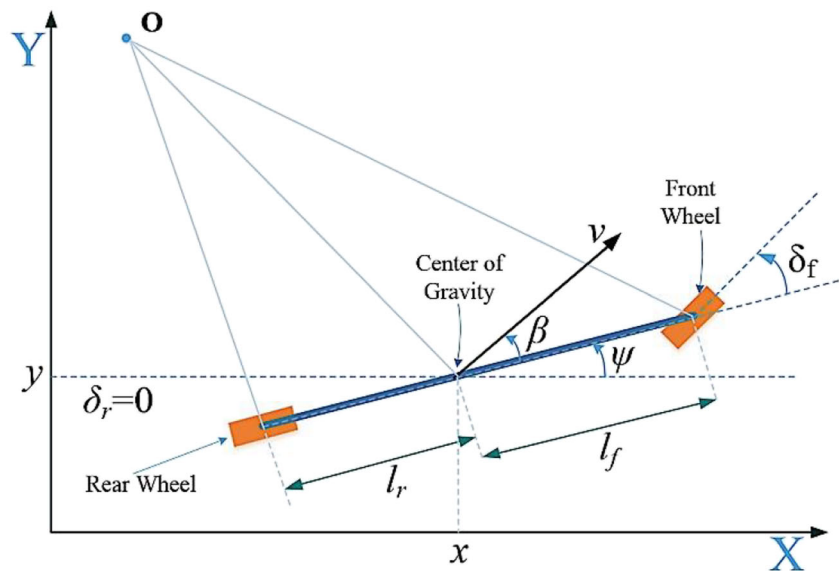
- 2) The vehicle speed at the given instance.
- 3) The instantaneous vehicle orientation angle (–ve for left and +ve for right).

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.

**Fig. 4** Structure of the PID controller



**Fig. 5** The Kinematic Bicycle Model



The steering command is produced after applying 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 centerline 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 eq. (7):

*Objective Function* = minimize{*MSE*}

$$= \min \frac{1}{N} \sum_{i=0}^{i=N} CTE_i^2 \quad (7)$$

The PID coefficients are tuned using a proposed tuning method given the name “WAF-Tune”. The following procedural steps illustrate this method:

- 1) The tuning will be done through the ad hoc technique “trial and error” which makes it a transparent procedure, unlike other automating algorithms like “Twiddle” [29] (will be described later) that is considered a kind of opaque approach (it is not known exactly, how each change in a hyperparameter affects the performance). This manual transparent approach incorporates the intuition as well as the experience in the tuning process.
- 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 eq. (4)). The number of samples is big enough to get a sufficiently accurate evaluation. It represents the measurements from the vehicle while going around the

designated track more than once (several times in case of the high enough speeds).

- 3) At first, the throttle value is kept 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 controller coefficients as follows ( $K_p = 0.1$ ,  $K_I = 0.0$  and  $K_D = 0.0$ ).
- 5) The simulation for several iterations (at least two) in which each iteration produces an individual measurement of the objective function has been run. Then, the average of these measurements is taken to produce the designated performance indicator (average Mean Squared Error “MSE”).
- 6) In the next set of iterations, the  $K_p$  has been then incremented small amounts based on intuition while keeping the other coefficients constant. Afterward, the new performance indicator is then calculated. If the performance gets better,  $K_p$  is kept incremented, otherwise; return back to 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 the increment of  $K_I$ , then  $K_D$  will be picked and getting step by step incremented until 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 fail to control the vehicle (the vehicle jumps out of the track) at an upper throttle value and the performance indicator shows a large unacceptable value of MSE.
- 12) The final set of coefficients  $\{K_P, K_I \text{ and } K_D\}$  that keep the vehicle within the track and an utmost reached value of throttle are 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.

## 5 The Twiddle Algorithm

The *Twiddle* algorithm [30] is a search algorithm that tries to find a kind of optimized selection of the hyper-parameters values of the PID controller based on the returned track error. The pseudo code for the implementation of the *Twiddle* algorithm is as follows:

```
# Choose an initialization for the parameter vector  $\zeta = [K_P, K_I, K_D]$ 
 $\zeta = [0, 0, 0]$ 
# Define potential changes
 $\Delta\zeta = [1, 1, 1]$ 
# Define Drive_CTE( $\zeta$ ) as the function that returns the  $\sum CTE$  for driving
  for the track with parameter vector  $\zeta$ 
# Calculate the error
best_err = Drive_CTE( $\zeta$ )

threshold = 0.001

while sum( $\Delta\zeta$ ) > threshold:
  for  $i$  in range(length( $\zeta$ )):
     $\zeta_i = \zeta_i + \Delta\zeta_i$ 
    err = Drive_CTE( $\zeta$ )

    if err < best_err: # There was some improvement
      best_err = err
       $\Delta\zeta_i = \Delta\zeta_i * 1.1$ 
    else: # There was no improvement
       $\zeta_i = \zeta_i - 2 * \Delta\zeta_i$  # Go into the other direction
      err = Drive_CTE( $\zeta$ )

  if err < best_err: # There was an improvement
    best_err = err
     $\Delta\zeta = \Delta\zeta * 1.05$ 
  else # There was no improvement
     $\zeta = \zeta + \Delta\zeta$ 
    # As there was no improvement, the step size in either
    # direction, the step size might simply be too big.
     $\Delta\zeta = \Delta\zeta * 0.95$ 
```

## 6 The Ziegler–Nichols Algorithm

The Ziegler–Nichols tuning method [31] is a heuristic method of tuning a PID controller. This method requires to set  $K_D$  and  $K_I$  to 0 and gradually increase  $K_P$  until it reaches the ultimate gain  $K_u$  before the vehicle runs with stable and consistent oscillations as shown in Fig. 6.  $K_P$  and the oscillation period  $T_u$  are used to set the  $K_P$ ,  $K_I$ , and  $K_D$  gains based on the type of controller used as shown in Table 1.

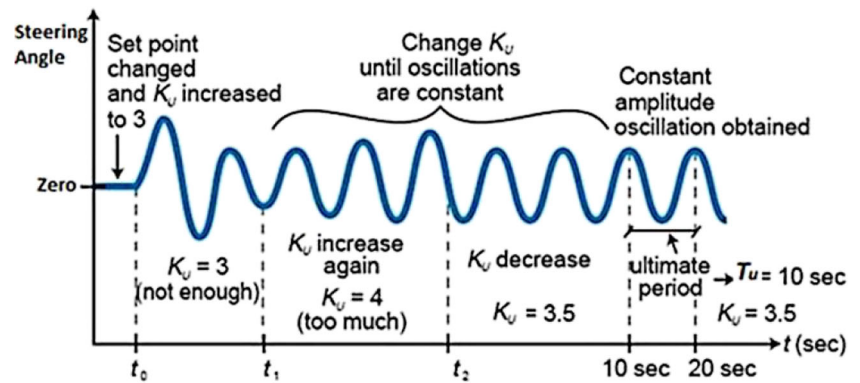
## 7 Tuning and Testing Results

The tuning of the PID controller is carried out using the upper three algorithms by driving the car (simulated using the presented model in Section III) around the test track shown in Fig. 7. Table 2 shows samples of the endeavors of the “WAF-Tune” (Section IV) to reach fine-tuned values for the controller coefficients  $\{K_P, K_I \text{ and } K_D\}$ .

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 3 below.

To tune the PID using *Twiddle* (Section V), the hyper-parameters have to be tuned manually at first. This was necessary because the narrow track left little room for error, and when attempting to automate parameter optimization it was very common for the car to leave the track, thus invalidating the optimization. Once the initial preliminary parameters were found that were able to get the car around the track reliably, *Twiddle* is then implemented. It was necessary to complete a full lap with each change in parameter because it was the only way to get a decent “score” (total error) for the parameter set. For this reason, the changes in each parameter are allowed to “settle in” for 100 steps and are then evaluated for the next 2000 steps. In all, *Twiddle* is allowed to continuously run for over 1 million steps (or roughly 500 trips around the track) to fine tune the parameters to their final values  $\{K_P: 0.134611$ ,  $K_I: 0.000270736$ ,  $K_D: 3.05349\}$  with an MSE of 0.1823.

Furthermore, to tune the PID using the *Ziegler–Nichols* method, the PID is initialized with hyperparameters values that just allowed the car to drive around the track even with a lot of wobbles. Then  $K_P$  is getting increased till the car makes approximately full swings between the borders (lane lines) of the track at which the value of  $K_P$  is captured and set to given the name  $K_u$ , while the period of the oscillations at this point is called  $T_u$ . The experimentations with different values of  $K_u$  &  $T_u$  is shown in Table 4 below to arrive at the final optimal values of  $\{K_P: 0.09$ ,  $K_I: 0.00144$ ,  $K_D: 1.40625\}$ . Unfortunately, the PID controller with resulted parameters was able to drive the car around the track but with a still lot of wobbling. That is why parameters should further tuned manually by a trial-and-error process. The same process is

**Fig. 6** Fine tuning the PID using “Ziegler–Nichols”

applied for different speeds (throttle values), so different PID parameters were found for different speeds. However, the overall performance is still inferior to that of the “WAF-Tune” and the “Twiddle” algorithms.

Table 5 summarizes the performance comparison among the three algorithms showing the best performance is achieved by WAF-Tune.

## 8 Discussion

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

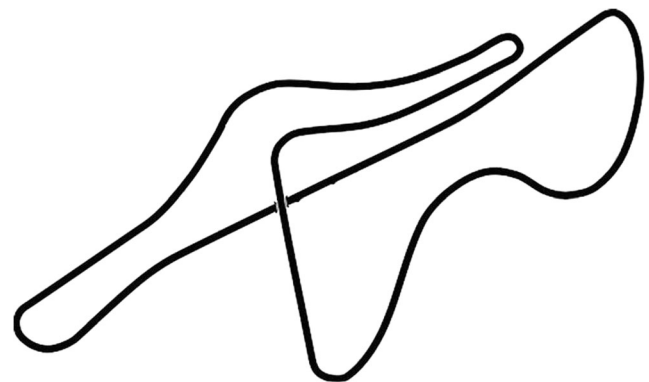
1. The PID controller has a very simple structure, however, it very effective in many control problems. Therefore, it is the most widely used approach by far in the industry (>90%).
2. The main problem with the PID is its tuning. There is no theory or criteria that proves that the optimal value for the coefficients  $\{K_p, K_I \text{ and } K_D\}$  have been reached. All the methods of tuning are mainly based on an extensive search with the incorporation of intuition and experience.
3. From the author point of view, using transparent methods based on extensive “trial and error” endeavors guided by numerical performance indicators is the best approach; as it lets the problem at hand been understood much deeper.

**Table 1** Ine tuning the PID using “Ziegler–Nichols”

Ziegler Nichols Method			
Control Type	$K_P$	$T_i = K_P / K_I$	$T_d = K_D / K_P$
P	$0.5K_u$	—	—
PI	$0.45K_u$	$T_u/1.2$	—
PD	$0.8K_u$	—	$T_u/8$
Classic PID	$0.6K_u$	$T_u/2$	$T_u/8$

Furthermore, it allows the incorporation the intuition and experience of the designer which reduces a lot of the search space; and consequently allows the designer at the end to be more effective.

4. The problem at 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 principle to track changes. In the presented 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, which interns results in lower CTE values as presented in Table 2.
5. The proportional controller ( $K_p$ ) is necessary to feedback the error to the controller with corrective action but it is not playing the principal role in improving the performance in this problem. 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

**Fig. 7** The test track



**Table 2** Fine tuning the PID controller using “WAF-Tune”

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.

of the setting point) and in many scenarios, it has a detrimental effect. The integral component in general counteracts any bias in the CTE and speeds up the approach to the center line, however, in our case, the center line keeps moving with respect to the vehicle which in many cases may cause the integral controller to produce oscillations.

## 9 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

**Table 3** Testing results for the PID controller at different speeds

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

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 at a specific speed range.
3. Applying “*Twiddle*” algorithm to the results of the proposed “*WAF-Tune*” to further optimize the hyper-parameters. However, caution should be taken in in order not to overfit the results to specific test tracks.

**Table 4** Experimentation with  $K_U$  &  $T_U$  to find Optimized  $K_P$ ,  $K_I$  and  $K_D$ 

$K_u$	$T_u$	Comments
2.0	100	Wheels turn very fast, very fast response
1.0	100	Completes a lap but response still very fast
0.5	100	Better than previous but still fast response
0.5	75	Better than previous but still fast response
0.25	75	Better than previous but still can be improved
0.25	60	Bigger oscillations but completes a lap
0.15	75	Completes a lap but needs to reduce oscillations
0.15	80	Much better but 0.25/0.75 still the best so far
0.25	80	Can still reduce oscillations
0.25	80	Increased speed (60 mph) with the same previous settings. High oscillations but still goes through a complete lap.
0.25	90	Several Laps without crashing but still needs to reduce oscillations.
0.25	100	Several Laps without crashing but still needs to reduce oscillations.
0.20	100	Better than previous, but can damp oscillation further.
0.19	110	Better than previous, but can damp oscillation further.
0.17	110	Better than previous, but can damp oscillation further.
0.15	110	Better than previous, but can damp oscillation further.
0.14	110	Better than previous, but can damp oscillation further.
0.14	115	Still, need to try and get inside lines with turn 3
0.13	115	Goes out of the lane
0.14	120	Completes laps. Try and get to drive inside lane on turn 3
0.14	125	Completes laps. Try and get to drive inside lane on turn 3
0.145	125	Completes laps without a problem but can damp oscillations further
0.15	125	Final solution

**Table 5** Comparison between the three techniques

Technique	$K_P$	$K_I$	$K_D$	MSE
WAF-Tune	0.35	0.0005	6.5	0.116543
Twiddle	0.134611	0.00027	3.05349	0.1823
Ziegler-Nichols	0.09	0.00144	1.40625	0.3845

4. Investigating the incorporation of the state-of-the-art concepts of optimized adaptive tracking of micro-robotic systems [32–34] that can enhance the maneuvering capabilities of autonomous cars.
5. Exploring the real-world connected vehicles data [35] (such as Safety Pilot Model Data (SPMD)) to be used for validating the controller and fine-tuning its parameters. The majority of the required data are available in the SPMD through vehicle On-Board Units and sensors.

## 10 Conclusion

In this paper, a PID controller is designed to efficiently steer an autonomous car following a pre-calculated track from a path planner. Three different methods are presented as design

approaches, one of them is newly proposed in this paper. The proposed controller uses on the cross-track-error as an input and outputs the steering commands. The method used to tune and minimize the suggested objective function is described in details. The performance of testing the PID controllers at different vehicle speeds (throttle setting) is also shown in details. A comprehensive discussion and analysis regarding the design of the PID as well as suggestions for improvements and future work are presented.

## References

- Mansour, K., Farag, W.: AiroDiag: a sophisticated tool that diagnoses and updates vehicles software over air, IEEE Intern. Electric Vehicle Conference (IEVC), TD Convention Center Greenville, SC, USA, March 4, 2012, ISBN: 978-1-4673-1562-3
- Farag, W., Saleh, Z., Traffic Signs Identification by Deep Learning for Autonomous Driving, Smart Cities Symposium (SCS'18), Bahrain, 22–23 April 2018
- Farag, W.: CANTrack: enhancing automotive CAN bus security using intuitive encryption algorithms, 7th Inter. Conf. on Modeling, Simulation, and Applied Optimization (ICMSAO), UAE, March 2017
- Farag, W., Saleh, Z.: Road lane-lines detection in real-time for advanced driving assistance systems", Intern. Conf. On Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT'18), Bahrain, 18–20 Nov. 2018
- Farag, W., Recognition of traffic signs by convolutional neural nets for self-driving vehicles, International Journal of Knowledge-based and Intelligent Engineering Systems, IOS Press, Vol: 22, No: 3, pp. 205–214, 2018
- Farag, W., Saleh, Z.: Behavior Cloning for Autonomous Driving Using Convolutional Neural Networks", Intern. Conf. on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT'18), Bahrain, 18–20 Nov. 2018
- Khattak, A.J., & Wali, B.: Analysis of volatility in driving regimes extracted from basic safety messages transmitted between connected vehicles, Transportation research part C: emerging technologies, 84, 48–73, (2017)
- Alonso, L., Pérez-Oria, J., Al-Hadithi, B.M., Jiménez, A.: Self-tuning PID controller for autonomous car tracking in urban traffic, in 17th Inter. Conf. on Sys. Theory, Control, and Computing (ICSTCC), Sinaia, Romania, 11–13 Oct. 2013
- Zhao, P., Chen, J., Song, Y., Tao, X., Xu, T., Mei, T.: Design of a Control System for an autonomous vehicle based on adaptive-PID. Int. J. Adv. Robot. Syst. 9(2), 44 (2012)
- Farag, W., Saleh, Z.: Tuning of PID track followers for autonomous driving, Intern. Conf. On Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT'18), Bahrain, 18–20 Nov. 2018
- Chebly, A., Talj, R., and Charara, A.: Coupled longitudinal and lateral control for an autonomous vehicle dynamics modeled using a robotics formalism, IFAC PapersOnLine, Vol 50, Issue 1, July 2017, Pages 12526–12532, <https://doi.org/10.1016/j.ifacol.2017.08.2190>
- Attia, R., Orjuela, R. and Bassent, M.: Longitudinal control for automated vehicle guidance, Workshop on Engine and Powertrain Control, Simulation and Modeling, IFAC, Rueil-Malmaison, France, October 23–25, 2012
- Filho, C., Wolf, D., Grassi, V. Jr, Osório, F.: Longitudinal and lateral control for autonomous ground vehicles, IEEE Intelligent Vehicles Symposium, Dearborn, MI, USA, 8–11 June 2014
- Le-Anh, T., Koster, M.B. De: A review of design and control of automated guided vehicle systems, Erasmus Research Institute of Management (ERIM), Report Series No. 2004–03-LIS, 2004
- Cheein, F.A.A., Cruz, C., Bastos, T.F., Carelli, R.: SLAM-based cross-a-door solution approach for a robotic wheelchair. Int. J. Adv. Robot. Syst. 7(2), 155–164 (2010)
- Lenain, R., Thuilot, B., Cariou, C. and Martinet, P.: Model predictive control for vehicle guidance in presence of sliding: application to farm vehicles' path tracking, IEEE Conf. on robotics and automation, pp. 885–890, 2005
- Byrne, R.H.: Design of a Model Reference Adaptive Controller for vehicle road following. Math. Comput. Model. 22(4–7), 343–354 (1995)
- Li, Z., Chen, W., Liu, H.: Robust control of wheeled Mobile manipulators using hybrid joints. Int. J. Adv. Robot. Syst. 5(1), 83–90 (2008)
- Hessburg, T.: Fuzzy logic control for lateral vehicle guidance. IEEE Control Syst Mag. 14, 55–63 (1994)
- Choomuang, R., Afzulpurkar, N.: Hybrid Kalman filter/fuzzy logic based position control of autonomous Mobile robot. Int J Adv Robotic Syst. 2(3), 197–208 (2005)
- Wang, W., Kenzo, N., Yuta, O.: Model reference sliding mode control of small helicopter X.R.B based on vision. Int J Adv Robotic Syst. 5(3), 233–242 (2006)
- Shumeet, B.: Evolution of an artificial neural network based autonomous land vehicle controller. IEEE Trans Syst Man Cybern. 26(3), 450–463 (1996)
- Lee, Y., Lee, Y.H., Na, S.G., Kwon, O.S., Heo, H.: The improvement of the PID response using support vector regression, The SPRING 8<sup>th</sup> International Conference on Computing, Communications and Control Technologies: CCCT 2010, April, 2010, Orlando, Florida, USA
- Zhuang, D.: The vehicle directional control based on fractional order PD $\mu$  controller. Journal of Shanghai Jiaotong. 41(2), 278–283 (2007)
- Tan, K.K., Qing-Guo, W., Chieh, H.C.: Advances in PID Control. Springer-Verlag, London (1999) ISBN 1-85233-138-0
- Crenganis, M., Bologa, O.: Implementing PID controller for a dc motor actuated mini milling machine. Academic J Manuf. Eng. 14(2), (2016)
- By TimmmmyK - Own work, CC0, <https://commons.wikimedia.org/w/index.php?curid=36662015>. Accessed 12 Mar 2019
- Kong, J., Pfeiffer, M., Schildbach, G. and Borrelli, F.: Kinematic and dynamic vehicle models for autonomous driving, in IEEE Intelligent Vehicles Symposium (IV), Seoul, South Korea, 28 June 2015
- By Skorkmaz [Public domain], from Wikimedia Commons, [https://commons.wikimedia.org/wiki/File:Change\\_with\\_Ki.png](https://commons.wikimedia.org/wiki/File:Change_with_Ki.png). Accessed 12 Mar 2019
- Thrun, S.: CS373: Artificial Intelligence for Robotics. Udacity, San Francisco, California (2018)
- Ziegler, J., Nichols, N.B., and Rochester N.Y.: Optimum settings for automatic controllers, Transactions Of The A.S.M.E., pp. 759–765, November 1942
- Liu, P., Yu, H., Cang, S.: Optimized adaptive tracking control for an underactuated vibro-driven capsule system. Nonlinear Dyn. 94(3), 1803–1817 (2018)
- Liu, P., Neumann, G., Fu, Q., Pearson, S., and Yu, H., Energy-Efficient Design and Control of a Vibro-Driven Robot, in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1464–1469), October 2018
- Liu, P., Yu, H., Cang, S.: Trajectory synthesis and optimization of an underactuated microrobotic system with dynamic

constraints and couplings. *Int. J. Control. Autom. Syst.* **16**(5), 2373–2383 (2018)

35. SPMD Data, <https://catalog.data.gov/dataset?tags=safety-pilot-model-deployment>. Accessed 12 Mar 2019

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Wael Farag** earned his Ph.D. from the University of Waterloo, Canada in 1998; M.Sc. from the University of Saskatchewan, Canada in 1994; and B.Sc. from Cairo University, Egypt in 1990. His research, teaching and industrial experience focus on embedded systems, mechatronics, autonomous vehicles, renewable energy, and control systems. He has a combined of 17 years of industrial and senior management experience in: Automotive (Valeo), Oil & Gas (Schneider) and

Construction Machines (CNH) positioned in several countries including Canada, USA & Egypt. Moreover, he has 10 Years of academic experience at: Wilfrid Laurier University, Cairo University, and the American University of the Middle East. Spanning several topics of electrical and computer engineering. He is the holder of 2 US patents; ISO9000 Lead Auditor Certified and Scrum Master Certified.