UNIJOURN

International Journal of Sustainable Transportation Technology
Vol. 3, No. 2, October 2020, 58-65

Optimization of Longitudinal Control of an Autonomous Vehicle using Flower Pollination Algorithm based on Data-driven Approach

Fadillah Adamsyah Ma'ani[1], Yul Yunazwin Nazaruddin[1,2]*

[1]Instrumentation and Control Research Group, Faculty of Industrial Technology, Institut Teknologi Bandung, Indonesia
[2]National Center for Sustainable Transportation Technology, Indonesia
**Email: yul@tf.itb.ac.id*

Abstract

Some challenges in the development of autonomous vehicles, such as generating a model representing the dynamics of the speed, designing a longitudinal controller, and the optimization method, are still explored until now. In this paper, a longitudinal controller based on the proportional-integral-derivative controller with an additional feed-forward term is proposed, where the Flower Pollination Algorithm is employed for optimizing the controller. The feed-forward term and the model used in the optimization are generated using the data-driven approach. For the optimization, a cost function considering mean absolute error and mean absolute jerk will be minimized. The simulation study was performed using the CARLA simulator, and the results show that the proposed scheme represents the dynamics of the speed very well inside the range of the training data and does not overfit the training data. It is also demonstrated that the proposed longitudinal controller can track the desired speed satisfactorily in a non-straight path.

Keywords

Autonomous vehicle; CARLA simulator; Data-driven approach; Flower pollination algorithm; Longitudinal control; Proportional-integral-derivative

1 Introduction

Until now, researches and developments of the autonomous vehicle still grow quite rapidly. One of the important aspects in the development of an autonomous vehicle is in the design of control systems. For autonomous vehicles, the control mechanism can be divided into two systems separately, i.e., longitudinal and lateral control. Longitudinal control is required for controlling the speed of the vehicle, whereas lateral control drives the vehicle to follow a path. The longitudinal control is not a simple problem because of the nonlinearity occurs in the dynamics, such as the drag force acting on the vehicle. One of the most popular controllers applied in longitudinal control is the proportional-integral-derivative (PID) type of controller and its variation [1–3]. PID controller is widely used because of inexpensive computation and its simplicity. Many works in the literature, e.g. [4], clearly explain how to tune the parameter of the PID controller, the variations of the PID controller, and issues regarding practical implementation. The response of the controlled variable can be adjusted by changing the PID parameters. Mostly, because of the integral component, the PID controller will give zero steady-state error if the set-point is a step function. To obtain a satisfactory tracking performance, an additional term should be added in order to compromise the changing set-point. For this reason, a new longitudinal controller form will be proposed in this investigation

One of the very challenging aspects of the design of a longitudinal controller is the tuning of the parameters. Manually tuning the parameters will lead to a non-optimal solution. To gain satisfying results, optimization procedures must be performed. Among many available techniques, one of the most promising optimization techniques to solve this kind of problem is the nature-inspired optimization technique, such in [5–8]. A nature-inspired optimization technique tries to mimic a natural phenomenon, mostly from an animal or a plant. In general, this technique performs well if it optimizes linear or non-linear, unconstrained or constrained, problems due to no need for the gradient. Currently, there are a number of nature-inspired optimization techniques that exist in the literature; the most popular one is Particle Swarm Optimization (PSO) [9]. However, some other techniques are more efficient than PSO, e.g., flower pollination algorithm (FPA). FPA works well to solve various optimization problems [10]. For that reason, FPA will be explored in this investigation for solving the optimization problems.

Received July 24, 2020, Revised September 18, 2020, Accepted for publication September 22, 2020

To optimize the longitudinal control system, a model that represents the vehicle longitudinal dynamics is needed. In theory, the vehicle longitudinal dynamics can be represented as a non-linear model. The nonlinearities at the model are generated by physical phenomena, e.g., drag forces, gravitational forces, longitudinal tire forces, and rolling resistance [11]. The model generated by the laws of physics is usually very complex, e.g. [12], and needs an extensive calibration. As a result, on some occasions, the modeling process is time-consuming, and in addition, the model also has a high computational cost due to its complexity. To address that problem, a new two-dimensional vehicle forward longitudinal model based-on the data-driven approach is introduced in this work. By using the data-driven approach, the model can estimate the training data quite well with low-cost calibration. However, the data-driven approach has some limitations. One of the limitations is the operating speed, which should be in the range of the simulation value.

To verify the proposed longitudinal control method for the autonomous vehicle, a simulation study will be conducted using CARLA Simulator [13]. This software gives a realistic environment for self-driving car research. The track and set-point used for simulation are obtained from the Introduction to Self-Driving Cars course provided by Coursera, with permission from the author of the course. For the optimization, the code is written using Python language in the Jupyter Notebook environment.

2 Flower Pollination Algorithm

Flower Pollination Algorithm (FPA) is an optimization method that mimics the flower pollination process in the real world with some simplifications. FPA is introduced in 2012 by Yang [10]. The global convergence of FPA has been proven in [14] by using Markov Chain. In this algorithm, a set of parameters is described as $x_i^k \in \mathbb{R}^N$ where x_i^k is the i-th flower at iteration k and N is the number of parameters.

FPA consists of two different update equations, i.e. global pollination and local pollination. The use of global and local pollination is switched concerning switching probability p. In global pollination, the parameters of a flower x_i^k will be updated based on its value and the best solutions $g \in \mathbb{R}^N$ so far. The global pollination is given as follows

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k + \gamma \boldsymbol{L}\left(\boldsymbol{g} - \boldsymbol{x}_i^k\right) \tag{1}$$

where $L \in \mathbb{R}^N$ is a Levy-flights-based step size. A random number s which follows the Levy distribution can be determined by an algorithm proposed in [15]. That random number can be determined using the following expression

$$s = X/|Y|^{\alpha} \tag{2}$$

where α is a constant, X is a zero-mean Gaussian distribution with variance σ^2 and Y is a zero-mean Gaussian distribution with variance 1. The variance σ^2 is calculated using formula

$$\sigma^2 = \left[\frac{\Gamma(1+\alpha)}{\alpha\Gamma(0.5+\alpha/2)}\frac{\sin(\pi\alpha/2)}{2^{\alpha/2-0.5}}\right]^{1/\alpha} \tag{3}$$

where Γ is the Gamma function. In this work, the step size is limited to a positive value s_0 so that $s \geq s_0 > 0$ for faster convergence. In local pollination, the parameters x_i^k will be updated based on the parameters of other flowers. The update equation can be written as

$$x_i^{k+1} = x_i^k + \epsilon\left(x_j^k - x_l^k\right) \tag{4}$$

where ϵ is a random number at interval $[0,1]$ and $i \neq j \neq l$.

Both (1) and (4) is not directly updated to the flowers or population. If the result of those equations gives a smaller cost or loss than the previous one, the new parameters will be updated to the population. This process is repeated for a pre-determined maximum number of iterations.

3 Longitudinal Controller

The proposed longitudinal controller consists of a feed-forward and feedback term. The feedback controller is the PID type controller, consisting three parts, namely: proportional, integral, and derivative. This controller works by considering the difference between the set-point and the controlled variable. On the other hand, the feed-forward controller is employed to improve the tracking performance. The feed-forward term which is used in the simulation is the throttle value at the steady-state condition given the set-point r. The feed-forward term is designed based on the simulation data.

In practice, the control signal is usually bounded by some constants because of actuator saturation. This phenomenon can generate undesirable overshoot. The integral part accumulates the error even though the control signal is not increasing or decreasing anymore. Hence, overshoot is needed to decrease the integral value. The simplest solution to tackle this problem is clamping the integral term by some constants. With clamping, the integral term stops accumulating value of

the signal when not being needed. In the proposed scheme, the integral term is clamped based on the difference between the saturated value of the actuator and the feed-forward term.

In discrete-time, the proposed longitudinal controller can be written as follows:

$$e_k = r_k - v_k \tag{5}$$

$$E_l = \text{sat}\Big(\big(M_l - s_{r,k}\big)/k_i, -\infty, 0\Big) \tag{6}$$

$$E_u = \text{sat}\Big(\big(M_u - s_{r,k}\big)/k_i, 0, \infty\Big) \tag{7}$$

$$E_k = \text{sat}\left(\sum_{i=1}^{k} e_k \Delta T\,, E_l, E_u\right) \tag{8}$$

$$u_k = s_{r,k} + k_p e_k + k_i E_k + k_d (e_k - e_{k-1})/\Delta T \tag{9}$$

where $(\cdot)_k$ is a variable at timestep k, v is the speed of a vehicle, e is the difference between the set-point and actual speed, u is the control signal, $\text{sat}(\cdot, b, c)$ is the saturated function with lower bound b and upper bound c, ΔT is the sampling time, M_l is the minimum value of the throttle, M_u is the maximum value of the throttle, s_r is the throttle value given set-point at steady-state condition, k_p is the proportional gain, k_i is the integral gain, and k_d is the derivative gain.

The optimum values of k_p, k_i, and k_d are determined by minimizing

$$\begin{aligned} \min \quad & J(k_p, k_i, k_d) = \frac{1}{M}\sum_{k=1}^{M}(|e_k| + \lambda_1 \ddot{v}_k^2) \\ \text{subject to} \quad & k_p \geq 0, \\ & k_i \geq 0, \\ & k_d \geq 0, \\ & M_p \leq \lambda_2. \end{aligned} \tag{10}$$

where J is the cost function, M is the number of simulation step, $\lambda_{(\cdot)}$ is a constant, $\ddot{v}$ is the second derivative of the speed or jerk, and M_p is the maximum of percentage overshoot. The cost function in (10) consists of two parts, i.e. mean absolute error (MAE) and mean absolute jerk (MAJ). By injecting the absolute error to the cost function, the error will be minimized so that the vehicle can track the desired speed very well. The jerk relates to passenger comfort. A large jerk can cause the speed of the vehicle to change suddenly. This occurrence can make the passenger feel uncomfortable.

The reference trajectory used in the optimization of the controller is given in Figure 1. The reference acceleration is continuous so that the reference speed is relatively smooth.

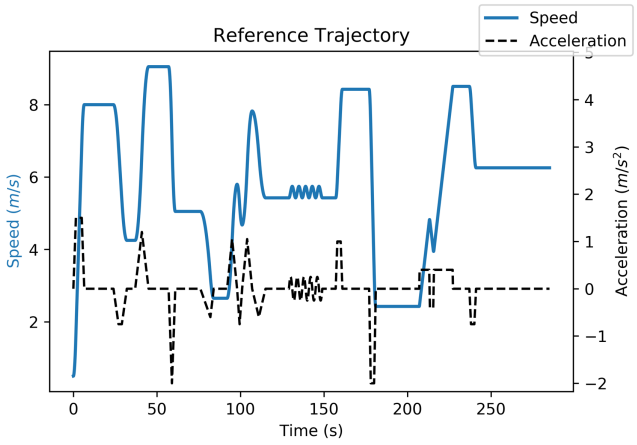


Figure 1 The trajectory which is used in the optimization of the longitudinal control

4 The Proposed Vehicle Longitudinal Model

4.1 Steady-state response

The steady-state relation between speed and throttle resulted from the CARLA simulator is shown in Figure 2. In this simulator, the throttle and brake lie in the interval [0,1]. The proposed relation between speed and throttle at steady-state condition is given as follows

$$s_r = \beta_1\left(1 - e^{\beta_2 v_s + \beta_3 v_s^{0.1}}\right) \tag{11}$$

where v_s is steady-state speed, β_1 is a positive constant, β_2 is a negative constant, and β_3 is a negative constant. With (11), the s_r has saturated value β_1. Therefore, s_r will not blow up when v_s keeps increasing.

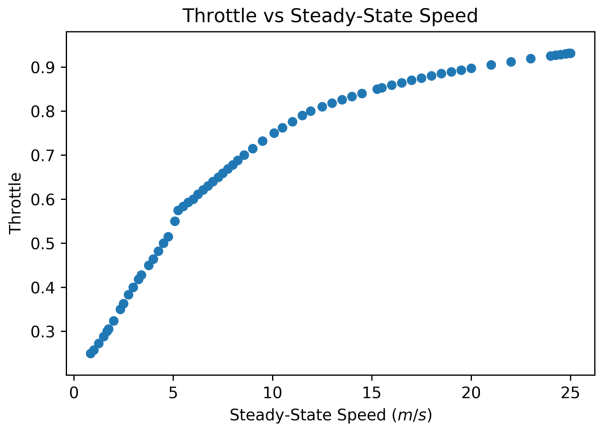


Figure 2 Steady-state response between throttle and speed in CARLA Simulator

4.2 Dynamic model

A dynamic model based on a data-driven system can be obtained using several techniques. Recently, the most popular data-driven method for representing dynamics is neural networks and its variants, e.g. [16] and [17]. Neural networks can construct a complex non-linear model consisting of simple non-linear functions, e.g. relu function, tanh function, and sigmoid function. The neural network algorithm requires significant number of data to obtain a good model. Otherwise, the resulting neural network model will be overfitting, e.g. [18].

Further, the speed dynamics is constructed from several functions so that the system is non-linear. The proposed model is written as follows

$$\begin{aligned}\dot{v} = {} & a_1 + a_2 v + a_3 v^2 \\ & + b_1 u_{11} + b_2 \exp(b_3 v + b_4 u_{12}) u_{13} \\ & + c_1 u_{21} + c_2 \exp(c_3 v + c_4 u_{22}) u_{23}\end{aligned} \tag{12}$$

where $\dot{v}$ is the acceleration of the vehicle, u_{1p} is the throttle with time delay d_{1p}, u_{2p} is the brake with time delay d_{2p}, and $a_1, a_2, a_3, b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4$ are constants. This model is proposed to represent the two-dimensional forward longitudinal dynamics. Some functions in (12) are inspired by the law of physics, e.g. drag forces and friction. The constraints and assumptions are given as follows:

- The friction resists the vehicle movement, i.e. $a_1, a_2 < 0$.
- The friction is zero when the vehicle is not moving, i.e. the term a_1 is reset to be zero if v is zero.
- The drag force resists the vehicle movement, i.e. $a_3 < 0$.
- The throttle drives the vehicle, i.e. $b_1, b_2 > 0$.
- The brake resists the vehicle movement, i.e. $c_1, c_2 < 0$.
- The time delay must not be negative, i.e. $d_{11}, d_{12}, d_{13}, d_{21}, d_{22}, d_{23} \geq 0$.

The training data consist of 12 time-series data as shown in Figure 3. The data are sampled at 100 Hz from CARLA simulator.

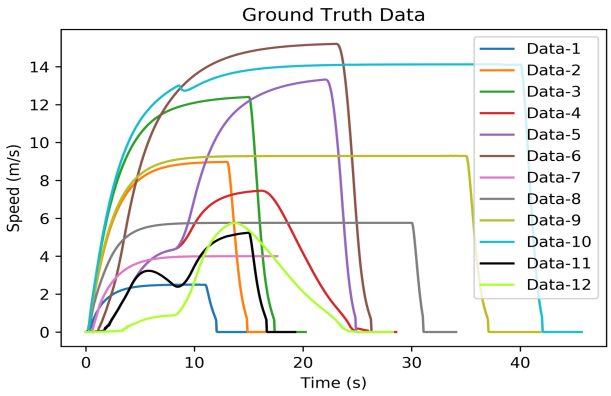


Figure 3 Training data of the vehicle longitudinal system identification

5 Simulation and Results

5.1 Optimization of longitudinal model

The proposed steady-state relation between the speed and throttle is optimized using FPA by minimizing mean squared error (MSE). There are three parameters that must be tuned in (11), i.e. β_1, β_2, and β_3. The optimization is done by using 50 flowers, 10000 iterations and FPA hyperparameters as follows: $p = 0.8$, $\alpha = 1.5$, $\gamma = 0.1$, $\sigma^2 = 0.697$, $s_0 = 0.1$. This training process takes 6 seconds using i5-6200U running at 2.69 GHz. The minimum MSE is 7.19×10^{-5} with $\beta_1 = 0.96$, $\beta_2 = -0.13$, $\beta_3 = -0.15$. The visualization of the result is given in Figure 4. From Figure 4, it can be seen that the proposed steady-state relation between speed and throttle (11) can fit the simulation data very well.

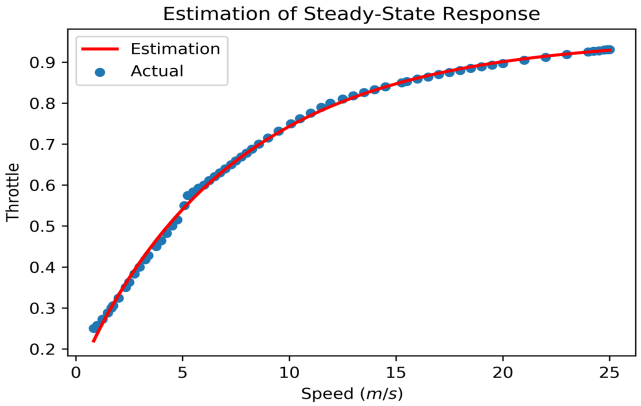


Figure 4 Steady-state response estimation

The proposed model (12) is optimized using FPA by minimizing MSE. Also, the metric used to evaluate the model is an accuracy metric. The accuracy metric is given as follows

$$acc = 100\left(1 - \frac{||v - \hat{v}||}{||v - \bar{v}||}\right)\% \quad (13)$$

where v is the real speed, $\bar{v}$ is the mean of the real speed and $\hat{v}$ is the estimated speed from the model. There are 17 parameters that must be tuned. The FPA hyperparameters used in this optimization is the same as the previous FPA hyperparameters. This training process takes 959 seconds. The minimum MSE is 9.34×10^{-3} with 98.01% accuracy for all training data. The resulting parameters are: $a_1 = -0.93$, $a_2 = -0.88$, $a_3 = -3.81 \times 10^{-6}$, $b_1 = 2.33$, $b_2 = 5.2$, $b_3 = 5.57 \times 10^{-2}$, $b_4 = 0.21$, $c_1 = -0.56$, $c_2 = -13.84$, $c_3 = -0.2$, $c_4 = -0.67$, $d_{11} = 0$, $d_{12} = 1.36$, $d_{13} = 0.3$, $d_{21} = 0.89$, $d_{22} = 0.42$, and $d_{23} = 0$. The accuracy, MSE, and maximum absolute error (MAAE) for each training data are shown in Table 1.

Table 1 Accuracy, MSE, and MAAE of each training data

Index of Data	Accuracy (%)	MSE (m^2/s^2)	MAAE (m/s)
1	92.77	0.00591	0.283
2	98.16	0.00408	0.372
3	98.41	0.00274	0.357
4	97.08	0.01719	0.692
5	97.99	0.00907	0.480
6	96.77	0.00772	0.180
7	97.88	0.01182	0.364
8	98.46	0.00751	0.302
9	81.15	0.03639	0.698
10	98.11	0.00125	0.279
11	95.16	0.00780	0.343
12	95.88	0.00677	0.220

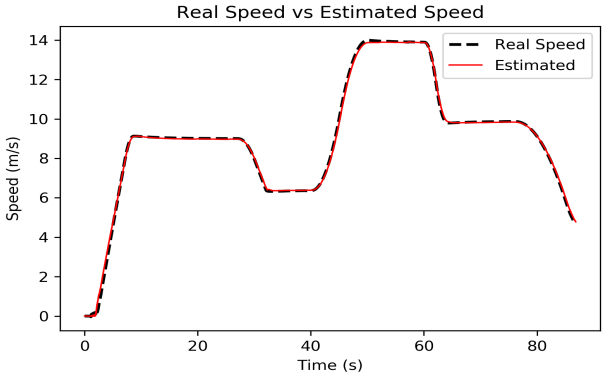


Figure 5 Simulation of proposed forward vehicle longitudinal model using the first test data

Test data are used to validate the model. The test data are not included in the training data. The estimation result using the first test data is demonstrated in Figure 5 with MSE = 9.15×10^{-3}, 96.88% accuracy, and MAAE = 0.527 m/s. The estimation result using the second test data is shown in Figure 6 with MSE = 7.49×10^{-2}, 93.8% accuracy, and MAAE = 0.826 m/s. From the test data, it can be concluded that the model does not overfit the training data and can generalize well. However, the proposed model gives a significant error when the speed is greater than 15. This undesirable error is caused by the range of speed in the training data.

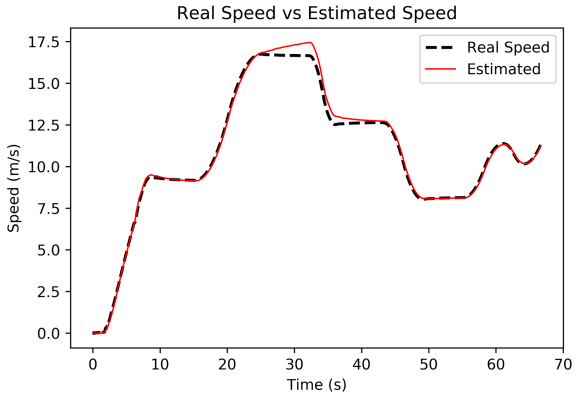


Figure 6 Simulation of proposed forward vehicle longitudinal model using the second test data

5.2 Optimization of longitudinal controller

The proposed longitudinal controller (9) is optimized using FPA by minimizing the cost function (10). The maximum value of throttle and brake is 1 while the minimum value is 0. There are three parameters that must be tuned. The set-point used in this optimization is given in Figure 1. The choice of the value of λ_1 is investigated in the optimization. The maximum overshoot must be less than 15%, i.e. $\lambda_2 = 15\%$. The sampling rate is 100 Hz. The simulation results are given in Table 2.

Table 2 The optimization results of the longitudinal controller

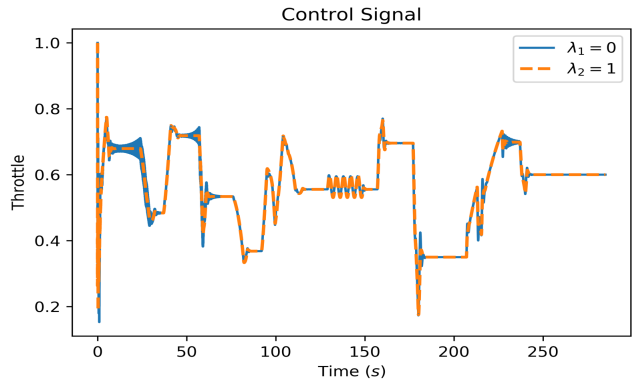
λ_1	k_p	k_i	k_d	MAE (m/s)	MAJ (m/s^3)	M_p (%)
0	0.565	$2.25 \cdot 10^{0}$	$7.82 \cdot 10^{-2}$	0.010	0.244	6.3
2	0.352	$1.14 \cdot 10^{-2}$	$3.91 \cdot 10^{-2}$	0.055	0.148	1.1
1	0.416	$4.49 \cdot 10^{-1}$	$5.15 \cdot 10^{-2}$	0.025	0.175	14.8
0.5	0.467	$8.63 \cdot 10^{-1}$	$5.65 \cdot 10^{-2}$	0.016	0.188	12.2
0.1	0.527	$1.62 \cdot 10^{0}$	$7.31 \cdot 10^{-2}$	0.011	0.207	8.7

From the observation during simulation, the actual speed and the control signal can be made smoother when the MAJ is minimized as shown in Figure 7. However, if the value of λ_1 is too big, the tracking performance will be bad. The λ_1 should be tuned to obtain a satisfying result according to the designer. For

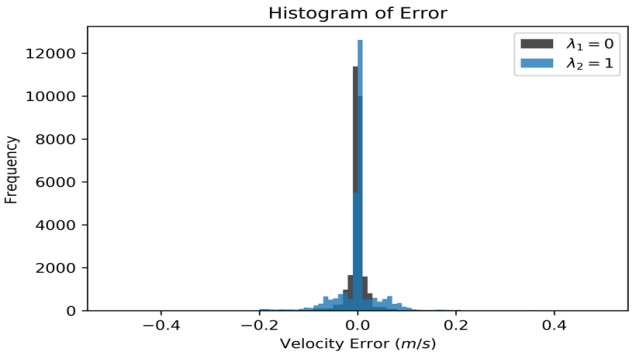
comparison, the control signal and the error histogram of the first and the third attempt is given in Figure 7.

5.3 Simulation of longitudinal control

The designed longitudinal controller is tested using a CARLA simulator. The sampling rate is 100 Hz. The reference speed, acceleration, and position are given in Figure 8. The simulation results are given in Table 3.

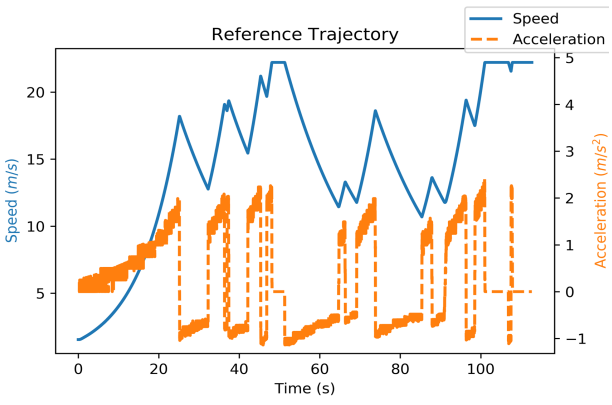


(a)

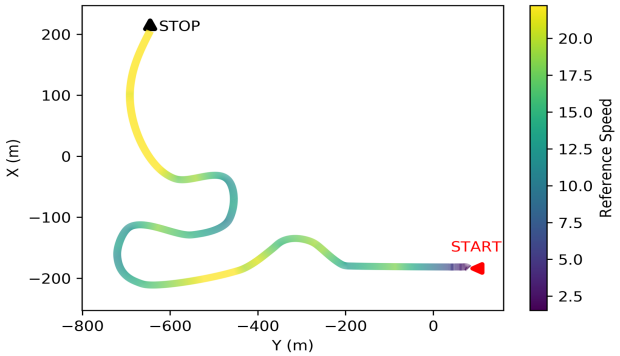


(b)

Figure 7 The comparison of the (a) control signal (the brake is always zero) and (b) the error histogram between the first attempt and the third attempt



(a)

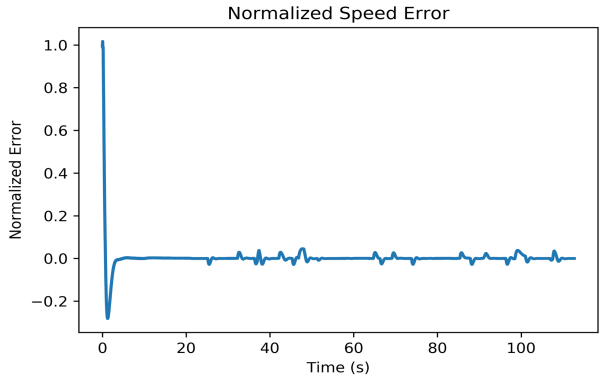


(b)

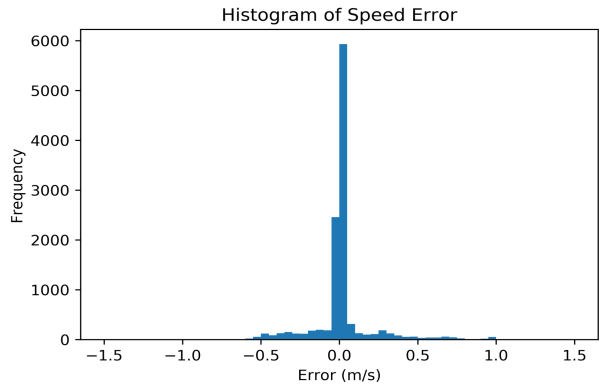
Figure 8 The (a) reference speed and acceleration, and (b) the reference path that used in the simulation of the longitudinal control using CARLA Simulator

Table 3 The simulation results of the designed longitudinal controller using CARLA simulator

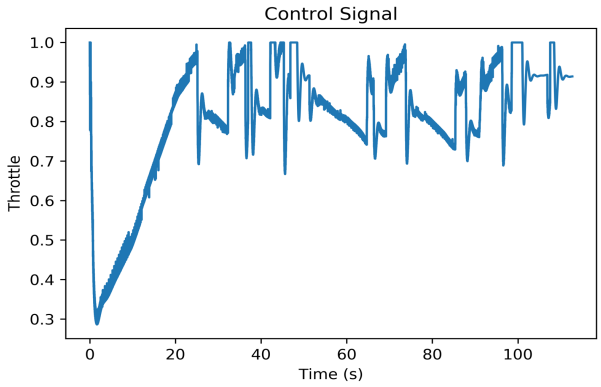
λ_1	MAE (m/s)	MAJ (m/s^3)
0	0.097	1.805
2	0.162	0.874
1	0.087	1.004
0.5	0.088	1.213
0.1	0.091	1.470



(a)



(b)



(c)

Figure 9 The (a) profile of normalized speed error, (b) the histogram of speed error, and (c) the control signal profile (the brake is always zero) from CARLA Simulator

The minimum MAE is given when $\lambda_1 = 1$. The simulation result for $\lambda_1 = 1$ is given in Figure 9. Overall, the results show that the designed longitudinal controller can track the desired speed satisfactorily even though the track is not a straight line. Besides, this good performance indicates that the model used in the optimization step describes the speed dynamics quite well. However, the minimum MAE does not occur when λ_1 is equal to zero. It indicates that by injecting the MAJ term in the cost function, the overfitting to the given trajectory can be reduced.

6 Conclusions and Future Work

This paper delivers a proposed model of steady-state speed response, a model of speed dynamics, and a longitudinal controller of an autonomous vehicle based on a data-driven approach. The FPA is used in the optimizations of the proposed model and the proposed longitudinal controller. The steady-state response model and the dynamics model are constructed from some linear and non-linear functions. The simulation results demonstrate that the model can describe the speed dynamics very well. The model does not overfit the training data, although the optimization utilizes relatively few data. However, the model generates an undesirable error when the speed is outside the range of the training data. The performance of the model can be improved by collecting more data with a broader range of speed and sloping road.

It has also been demonstrated that the proposed longitudinal control scheme can track the reference speed quite satisfactorily. Besides, taking into account the MAJ in the cost function can be used to prevent the controller to overfit the given trajectory. Future works would consider additional assumptions and simulations to improve the robustness of the controller, e.g. simulating the controller tuning in 3D space, cooperating the slope of the road to calculate the feed-forward controller, and generating some disturbances.

Acknowledgment

This work was supported by the P3MI Program 2020 at Institut Teknologi Bandung, and in part by USAID through the Sustainable Higher Education Research Alliances (SHERA) program under grant number IIE00000078-ITB-1.

References

[1] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing," in *2007 American Control Conference*, 2007, pp. 2296–2301.

[2] P. Zhao, J. Chen, Y. Song, X. Tao, T. Xu, and T. Mei, "Design of a Control System for an Autonomous Vehicle Based on Adaptive-PID," *Int. J. Adv. Robot. Syst.*, vol. 9, p. 1, Jul. 2012.

[3] M. Marcano, J. A. Matute, R. Lattarulo, E. Martí, and J. Pérez, "Low Speed Longitudinal Control Algorithms for Automated Vehicles in Simulation and Real Platforms," *Complexity*, vol. 2018, no. March, 2018.

[4] K. J. Åström and T. Hägglund, *PID controllers : theory, design, and tuning*. Research Triangle Park, N.C.: International Society for Measurement and Control, 1995.

[5] J. Zhao, T. Li, and J. Qian, *Application of Particle Swarm Optimization Algorithm on Robust PID Controller Tuning*, vol. 3612. 2005.

[6] A. L. Sangeetha, N. Bharathi, A. B. Ganesh, and T. K. Radhakrishnan, "Particle swarm optimization tuned cascade control system in an Internet of Things (IoT) environment," *Measurement*, vol. 117, pp. 80–89, 2018.

[7] A. Nawikavatan and D. Puangdownreong, "Optimal PIDA Controller Design for Truck Breaking System using Flower Pollination Algorithm," in *2019 7th International Electrical Engineering Congress (iEECON)*, 2019, pp. 1–4.

[8] I. G. N. A. I. Mandala, Franky, and Y. Y. Nazaruddin, "Optimization of Two Degree of Freedom PID Controller for Quadrotor with Stochastic Fractal Search Algorithm," in *2019 IEEE Conference on Control Technology and Applications (CCTA)*, 2019, pp. 1062–1067.

[9] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, vol. 4, no. 6, pp. 1942–1948 vol.4.

[10] X. S. Yang, "Flower Pollination Algorithm for Global Optimization," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7445 LNCS, pp. 240–249, 2012.

[11] R. Rajamani, *Vehicle Dynamics and Control*. 2006.

[12] F. Ahmad, S. Mazlan, H. Zamzuri, H. Jamaluddin, K. Hudha, and M. Short, "Modelling and validation of the vehicle longitudinal model," *Int. J. Automot. Mech. Eng.*, vol. 10, pp. 2042–2056, Dec. 2014.

[13] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017.

[14] X. He, X. S. Yang, M. Karamanoglu, and Y. Zhao, "Global Convergence Analysis of the Flower Pollination Algorithm: A Discrete-Time Markov Chain Approach," *Procedia Comput. Sci.*, vol. 108, pp. 1354–1363, 2017.

[15] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of L\'evy stable stochastic processes," *Phys. Rev. E*, vol. 49, no. 5, pp. 4677–4683, May 1994.

[16] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 4653–4660.

[17] J. Gonzalez and W. Yu, "Non-linear system modeling using LSTM neural networks," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 485–489, 2018.

[18] Y. Y. Nazaruddin, F. A. Ma'ani, P. W. L. Sanjaya, E. R. Muten, G. Tjahjono, and J. A. Oktavianus, "Localization Method for Autonomous Car Using Virtual Sensing System," in *2019 6th International Conference on Electric Vehicular Technology (ICEVT)*, 2019, pp. 202–207