Exploring the Impact of Car-Following Models and Controller Formulations on Autonomous Vehicle Motion Planning

Raju, Narayana

DOI
10.1061/AJRUA6.RUENG-1595

Publication date
2025

Document Version
Final published version

Published in
ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering

Citation (APA)
Raju, N. (2025). Exploring the Impact of Car-Following Models and Controller Formulations on Autonomous Vehicle Motion Planning. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, *11*(4), Article 04025067. https://doi.org/10.1061/AJRUA6.RUENG-1595

Exploring the Impact of Car-Following Models and Controller Formulations on Autonomous Vehicle Motion Planning

Narayana Raju[1]

Abstract: Effectively planning the behavior of autonomous vehicles (AVs) while accounting for their mechanical properties and traffic flow characteristics is a challenging task. This study evaluates different car-following models in combination with various controllers to model the longitudinal behavior of AVs. Controllers regulate a vehicle's speed, ensuring smooth acceleration in accordance with its planned path. Specifically, conventional models, interaction-based models, and artificial intelligence-based models were tested alongside standard controllers. The respective transfer functions of the system were derived, and the weights were tuned accordingly. Nanoscopic simulation runs were conducted to assess performance. The results indicate that the choice of car-following model and controller significantly impacts the longitudinal planning of AVs, with certain combinations demonstrating superior performance. This study thus provides a framework for identifying the optimal pairing of a car-following model and controller to enhance longitudinal behavior planning in AVs. **DOI: 10.1061/AJRUA6.RUENG-1595.** *© 2025 American Society of Civil Engineers.*

Author keywords: Automated vehicles; Car-following; Longitudinal driving behavior; Adaptive cruise control; Intelligent driver model; Nanoscopic simulation.

Introduction and Background

With advancements in technology, autonomous vehicles (AVs) and connected vehicles have the potential to enhance traffic safety and efficiency. However, effectively planning the longitudinal behavior of AVs remains a challenge due to their reliance on car-following models and controller mechanisms. Previous studies have explored various approaches to modeling and controlling AV longitudinal dynamics. While car-following models are often based on traffic flow concepts, real vehicle interactions are more complex. Controllers work alongside these models to regulate speed and acceleration, ensuring AVs follow their planned paths smoothly and efficiently. However, existing literature lacks a systematic evaluation of how different combinations of car-following models and controllers impact the longitudinal behavior of AVs.

In modeling the longitudinal behavior of connected and automated vehicles (C/AVs), researchers tested various car-following model formulations, such as the conventional car-following models (Wen-Xing and Li-Dong 2018), interaction-based models (Gong et al. 2016), backward looking effect (Ma et al. 2021), and artificial intelligence models (Gao et al. 2018). In a review paper on replicating the behavior of C/AVs in microsimulation (Raju and Farah 2021), it was found that researchers primarily tended to use the intelligent driver model (IDM), adaptive cruise controller (ACC), cooperative adaptive cruise controller (CACC), and optimum velocity model. The main reason behind this is the ability of their mathematical formulation in generating consistent and stable car-following behavior. These models were applied with the assumption that C/AVs will have similar behavior to human car-following behavior but with shorter response time. Most microsimulation studies predict that C/AVs would improve the safety and efficiency of traffic flow characteristics (Makridis et al. 2020).

Given the advancements in technology, AVs of lower Society of Automotive Engineers levels are presently available in the consumer market. Considering this, several researchers tested the performance of AVs in an empirical manner. Makridis et al. (2020) investigated the string stability impacts of AVs. They identified that the response time of the controllers was in the range of 1.7 to 2.5 s. Gunter et al. (2020) who tested a platoon consisting of seven vehicles, observed that a small 6 mph amplitude by the first vehicle in the platoon is amplified to 25 mph by the last vehicle in the platoon, and, therefore, the automated system was found to disengage. Makridis et al. (2021) inferred that AVs could lead to a higher energy consumption and introduce new safety risks with the rise of their penetration level. Raju et al. (2022) tested a commercially available adaptive cruise control system and found that different system settings affect the car-following indicators, and that the system response times are comparable to human response times.

The above literature shows that microsimulation studies and other traffic modeling studies indicate that AVs will improve the traffic flow efficiency and safety standards. In contrast, empirical based studies highlight the noninstantaneous response time of AVs and emphasize the potential negative impacts on traffic flow efficiency, safety, and energy consumption.

Typically, AVs' planning (Zaghari et al. 2021) takes place in four stages: global planning, behavior planning, local planning, and hierarchical planning. In the *global planning* stage, the routes of the AVs are planned between their origins and their destinations considering different optimization strategies. *Behavioral planning* mainly focuses on the driving behavior decisions at the tactical

[1] Consultant, cimt BV, Vught 5262 LR, Netherlands; formerly, Postdoctoral Researcher, Dept. of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft Univ. of Technology, Delft 2628 CD, Netherlands. ORCID: https://orcid.org/0000-0002-3561-5676. Email: s.narayanaraju.10@gmail.com

Note. This manuscript was submitted on December 21, 2024; approved on April 21, 2025; published online on August 11, 2025. Discussion period open until January 11, 2026; separate discussions must be submitted for individual papers. This paper is part of the ***ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering***, © ASCE, ISSN 2376-7642.

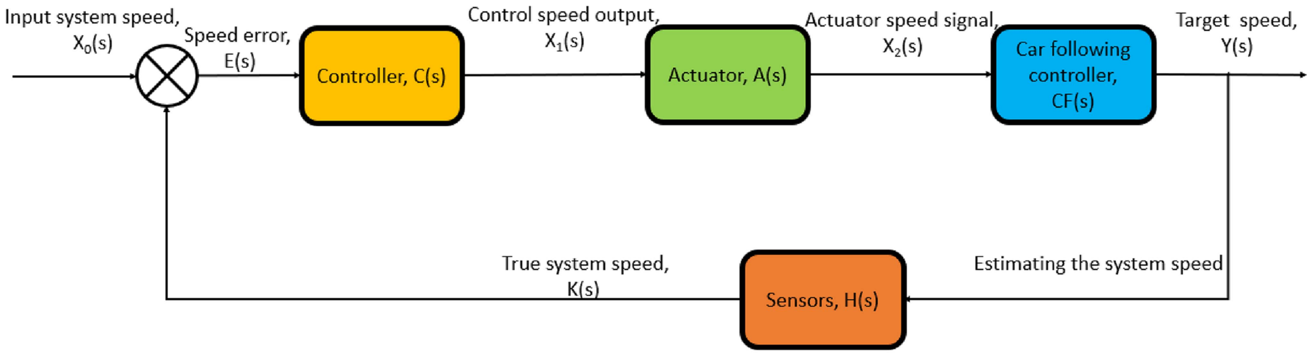

Fig. 1. Longitudinal controller block diagram in a typical AV system. See Lasch et al. (2024) for the conceptual basis of the controller mechanism.

level in a traffic stream, such as speed tracking, decision to follow the leader, merging, and stopping decisions. Lateral movements are governed at the *local planning* stage. The longitudinal movement is regulated by car-following controllers. In control systems modeling, transfer functions play a crucial role in revealing and characterizing the system's responsiveness to inputs. For instance, Zhang et al. (2014) employed transfer functions to implement a proportional and differential model for regulating traffic flow stability. Similarly, Sun et al. (2018) used transfer functions to test the stability of car-following models under various criteria and model conventional and connected traffic. Notably, fine-tuning the weights of transfer functions significantly impacts performance. In this context, Ziegler-Nichols's approach using the process reaction curve (El-Badry 2021) serves as a popular method for optimizing controller weights and tuning transfer functions of car-following models.

From the literature, it can be inferred that research on car-following models and vehicle motion planning has largely treated these two areas independently. However, in real-world conditions, both aspects must work together to plan the longitudinal behavior of AVs. In traffic modeling studies, car-following models are typically used at the microscopic level to simulate detailed vehicle movements, often with little consideration for controller dynamics. However, recent field studies indicate that AVs do not react instantaneously, as is often assumed in traffic models. Instead, AVs exhibit response times and delays when adapting to their surroundings. This highlights a critical research gap: the need for a more integrated modeling approach that explicitly accounts for response delays in car-following behavior.

Research Methodology

Following the main objective, this study was performed in four stages: (1) in the initial stage, the system transfer function was developed in line with the chosen car-following model and the controller; (2) in the second stage, selected car-following models were calibrated and validated using the publicly available Waymo data set (Hu et al. 2022); (3) in the third stage, the selected controllers were modeled in line with their respective system transfer functions, and the controller weights were tuned; and (4) finally, in the last stage, nanoscopic simulation runs were performed. The system was tested using various performance measures based on the controller and car-following combinations.

The following subsections further elaborates these four stages described above.

Planning the Behavior of AVs with Controllers

Car-following models estimate AVs' longitudinal target states (acceleration/speed). To keep the AVs in a balanced state in stable conditions, the target speed $Y(s)$ that is predicted by the car-following model is processed regarding the input system speed $X_o(s)$ of the AV with the help of a controller $C(s)$. The controller mainly focuses on the speed error $E(s)$, which is the difference between the input system speed $X_o(s)$ and the true system speed $K(s)$. The AV sensors (e.g., radar, LIDAR) play a significant role in providing the necessary update to the controller to assess the system state. Further, the controller passes the speed information to the actuator to operate the AV. This loop repeats itself; the system speed is processed by the car-following model and the controller for predicting the target speed. The block diagram of a typical longitudinal controller loop is shown in Fig. 1.

To express the controller loop dynamics in a better manner, the transfer functions for the loop are derived as follows:

$$Y(s) = \mathrm{CF}(s) \cdot X_2(s) \tag{1}$$

$$Y(s) = \mathrm{CF}(s) \cdot A(s) \cdot X_1(s) \quad \text{where } X_2(s) = A(s) \cdot X_1(s) \tag{2}$$

$$Y(s) = \mathrm{CF}(s) \cdot A(s) \cdot C(s) \cdot E(s) \quad \text{where } X_1(s) = C(s) \cdot E(s) \tag{3}$$

$$Y(s) = \mathrm{CF}(s) \cdot A(s) \cdot C(s) \cdot [X_0(s) - K(s)] \quad \text{where } E(s) = X_0(s) - K(s) \tag{4}$$

$$Y(s) = \mathrm{CF}(s) \cdot A(s) \cdot C(s) \cdot [X_0(s) - H(s) \cdot Y(s)] \quad \text{where } K(s) = H(s) \cdot Y(s) \tag{5}$$

$$[1 + \mathrm{CF}(s) \cdot A(s) \cdot C(s) \cdot H(s)]Y(s) = CF(s) \cdot A(s) \cdot C(s) \cdot X_0(s) \tag{6}$$

Further by rearranging the terms in Eq. (6), the closed-loop transfer function for the entire system is given as

$$G(s) = \frac{Y(s)}{X_0(s)} = \frac{CF(s) \cdot A(s) \cdot C(s)}{[1 + CF(s) \cdot A(s) \cdot C(s) \cdot H(s)]} \tag{7}$$

where $Y(s)$ = target speed outcome from the car-following model in a complex domain; $CF(s)$ = transfer function of the car-following model; $A(s)$ = transfer function of the actuator; $C(s)$ = transfer function of the controller; $H(s)$ = transfer function of the sensor;

$X_0(s)$ = input system speed in a complex domain; $X_1(s)$ = control speed outcome from the controller; $X_2(s)$ = control speed outcome from the actuator; $E(s)$ = speed error; $K(s)$ = true speed of the system measured by the sensor; and $G(s)$ = transfer function of the system.

Car-Following Models

Numerous car-following models are available in the literature to replicate the longitudinal behavior of vehicles, ranging from traditional response–stimulus models to advanced psychophysical models (Olstam and Tapani 2004). In the case of AVs car-following behavior, researchers assumed that the behavior will be smooth compared to human car-following behavior. Considering this, certain car-following models were favored in the literature for modeling the longitudinal behavior of AVs. In particular, the IDM (Kesting et al. 2010), which provides a mathematical formulation for determining how a vehicle adjusts its speed and acceleration to maintain safe and smooth driving in response to surrounding traffic conditions, was commonly used to describe the car-following behavior of AVs. Furthermore, ACC (Milanés and Shladover 2014) is effective for AVs in reaching their desired speed in free-flow conditions and adopting a linear safety distance model in their following behavior. In line with the literature and considering the above aspects, IDM and ACC models were selected to plan the AVs' longitudinal behavior. Furthermore, considering the strength of deep learning (DL) model in predictions, it has also been selected to be tested along with the IDM and ACC.

Intelligent Driver Model

To model the continuous movement of vehicles on the German freeways as individual lane-wise movement, Treiber et al. (2000) introduced the concept of IDM as presented in Eq. (8). in which acceleration of the vehicle is assumed to be a function of maximum acceleration $a_{\max}$, maximum deceleration b, desired time headway T, speed v, relative distance Δx, and speed difference with leader Δv

$$a = a_{\max} \cdot \left[1 - \left(\frac{v}{V_0}\right)^{\delta} - \left(\frac{x_0 + v \cdot T + \frac{v \cdot \Delta v}{2 \cdot \sqrt{a_{\max} \cdot b}}}{\Delta x}\right)^2\right] \qquad (8)$$

where x_0 = minimum safe distance; V_0 = desired speed; and δ = exponential parameter.

Adaptive Cruise Control

To generate a consistent cruise control process in the following behavior of vehicles, researchers (Xiao et al. 2017) have used the ACC and CACC models. The ACC and CACC are two regime controllers including cruising and following. In the case of cruising, subject vehicles tend to match their desired speeds. In the case of following behavior, subject vehicles accommodate their relative speeds and distances with their leaders to maintain the time headway set in the system. In the present work, considering the single leader combination for the subject vehicle, the ACC model is used for the analysis.

The acceleration of the subject vehicle in cruising conditions is given as

$$a_1 = k_1 \cdot (V_0 - v) \qquad (9)$$

where k_1 = multiplication factor; and a_1 = acceleration under cruising conditions. The acceleration of the subject vehicle in following conditions is given as

$$a_2 = k_2 \cdot (\Delta x - x_0 - v \cdot T) - k_3 \cdot \Delta v \qquad (10)$$

where Δx = relative distance; x_0 = minimum safe distance; Δv = speed difference with leader; T = desired time headway; v = speed; k_2, k_3 = parameters; and a_2 = acceleration under following conditions.

Finally, the acceleration a of the subject vehicle is computed as the minimum of the two accelerations from the cruise control and the following conditions as follows:

$$a = \min(a_1, a_2) \qquad (11)$$

Deep Learning

Considering the prediction power of artificial intelligence (AI) in recent years, deep learning from the branch of AI was selected in this study to be tested. The architecture of the deep learning model is comparable to neurons in the brain cells. Like the way electrical signals travel across the living cells, each subsequent layer of nodes will be activated when it receives stimuli from its neighboring neurons. Given this, the deep learning model (Patil et al. 2021) predictions' accuracy can be increased with the right amount of training data. There are three different layers in deep learning models: input layers, hidden layers, and output layers. The input layers are provided with the input vectors to map the outcomes in the output layer. Given this, the input data are filtered through a series of hidden layers. The hidden layers are in between the input and output layers, as shown in Fig. 2.

In the present study, TensorFlow's keras sequential (Manaswi 2018) model variant is used with three hidden layers of 128, 64, and 16 nodes that were adopted. In limiting the overfitting, 250 epochs were used for training. Further, the rectified linear unit (ReLU) activation function is adopted for activating the nodes. The final layer has just one neuron, indicating the model predicts a single continuous value. An optimizer root-mean-square propagation (RMSprop) and learning rate (0.01) are specified to guide the training process. The model is compiled to use mean-square error as the loss function and monitors both mean-square error and mean absolute error during training.

In deep learning, the activation functions play a major role in passing the signal among the nodes. In general, the activation function takes the form as

$$h(x) = \sum_{j=0}^{n} \hat{\theta}_j + \theta_j \cdot x_j \qquad (12)$$

and the loss function (mean-square error) is given as

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} [h_\theta(x_i) - y_i]^2 \qquad (13)$$

where $h(x)$ = activation function outcome; θ_j = random initial weight; $\hat{\theta}_j$ = random initial weight with bias; x_j = input vectors; and y_i = output vectors.

Typically, the car-following model predicts the instantaneous acceleration for the vehicle as a function of a relative distance, relative speed, and current speed. Considering this, in the deep learning model, the acceleration is treated as the dependent variable (i.e., the outcome layer) and $\Delta x, \Delta v, v$ as independent variables (i.e., the input layer).

Car-Following Model Formulation and Transfer Functions

To understand the essence of a car-following model formulation, its performance in terms of stability (i.e., to effectively model the longitudinal movement and keep the system in a balanced state) for

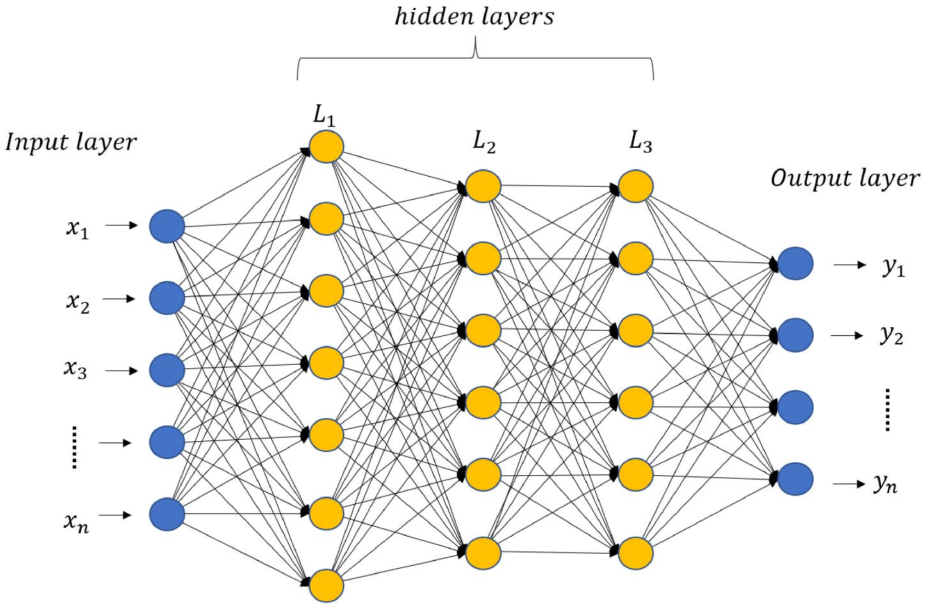

Fig. 2. Deep learning architecture.

varied conditions must be analyzed. To explore this in a better manner, the formulations of the car-following models, mainly IDM and ACC, were investigated.

Typically, in car-following models the response state (acceleration) of the subject vehicle is predicted as a function of relative distance, relative speed, and speed given as

$$a(t) = f[\Delta x, \Delta v, v]_t \tag{14}$$

At equilibrium conditions (Fayolle et al. 2020), given disturbances $\Delta x, \Delta v$ in the system, the above Eq. (14) takes the form as

$$0 = f[\Delta x, \Delta v, 0] \tag{15}$$

In line with the literature (Fayolle et al. 2020; Wang et al. 2020), Eq. (15) defines a second-order linear differential equation. As a result, the fundamental transfer function for a given car-following model takes the form as

$$\mathrm{CF}(s) = \frac{\frac{\partial f}{\partial v} \cdot s + \frac{\partial f}{\partial \Delta x}}{s^2 + \left[\frac{\partial f}{\partial \Delta v} - \frac{\partial f}{\partial v}\right] \cdot s + \frac{\partial f}{\partial \Delta x}} \tag{16}$$

In the case of IDM

$$\frac{\partial f_{\mathrm{IDM}}}{\partial v} = a_{\max} \cdot \left[-\frac{\delta}{V_0^{\delta}} \cdot v^{\delta-1} - \frac{2(x_0 + v \cdot T + \frac{v \cdot \Delta v}{2\sqrt{a_{\max} \cdot b}})}{\Delta x^2} \cdot \left(T + \frac{\Delta v}{2\sqrt{a_{\max} \cdot b}}\right)\right] \tag{17}$$

$$\frac{\partial f_{\mathrm{IDM}}}{\partial \Delta x} = a_{\max} \cdot \left[\frac{-2}{3\Delta x^3} \cdot \left(x_0 + v \cdot T + \frac{v \cdot \Delta v}{2\sqrt{a_{\max} \cdot b}}\right)^2\right] \tag{18}$$

$$\frac{\partial f_{\mathrm{IDM}}}{\partial \Delta v} = \frac{a_{\max}}{\Delta x^2} \cdot \left[-\left(x_0 + v \cdot T + \frac{v \cdot \Delta v}{2\sqrt{a_{\max} \cdot b}}\right) \cdot \frac{v}{\sqrt{a_{\max} \cdot b}}\right] \tag{19}$$

In the case of ACC:

$$\frac{\partial f_{\mathrm{ACC}}}{\partial v} = (-k_1, -k_2 \cdot T) \tag{20}$$

$$\frac{\partial f_{\mathrm{ACC}}}{\partial \Delta x} = k_2 \tag{21}$$

$$\frac{\partial f_{\mathrm{ACC}}}{\partial \Delta v} = -k_3 \tag{22}$$

By substituting the respective terms in Eq. (16), the transfer functions for both the IDM and ACC are obtained as $CF_{\mathrm{IDM}}(s)$ and $CF_{\mathrm{ACC}}(s)$. At the same time, it is highly important in control problems to assess the system functioning by evaluating the bounded input bounded output (BIBO) stability. Given the system as BIBO stable, then the output will be bounded for every input to the system that is bounded. In this direction, to satisfy the criteria for BIBO stability, the poles [i.e., values of s in the transfer function, Eq. (16), where the system becomes uncontrollable] of the respective car-following transfer functions must be on the left-hand complex plane. From the analysis, it can be observed that, when comparing the transfer function of IDM to that of the ACC, the transfer function of ACC is less complex in comparison to that of the IDM. On the other hand, developing any transfer function for the DL model is quite complex and requires complex formulation; as a result, for the DL model no transfer function was developed. At the same time, unlike the IDM and ACC, the outcomes from the DL model are fully dependent on the trained data; given this, the DL model is always expected to provide stable outcomes.

Calibration and Validation with Waymo Data

The open source Waymo (Google 2021) trajectory data are used to calibrate the vehicles' following behavior. The Waymo data set is a part of Google's self-driving project. It contains detailed motion and perception behavior of the Waymo self-driving vehicle to its surroundings in urban traffic conditions. Few researchers so far (Bansal et al. 2019; Hawkins 2019; Hu et al. 2021; Scanlon et al. 2021; Wang et al. 2023) have explored the Waymo data set and revealed insights regarding its driving behavior. Based on this, in the present study, to

Table 1. Calibrated parameters of the models and the errors

Car-following model	Parameters	Units	Calibrated value	Calibration error (MAPE)	Validation error (MAPE)
IDM	$a_{\max}$	$\mathrm{m/s^2}$	0.74	8.41	9.54
	V_0	$\mathrm{m/s}$	13.89		
	x_0	m	2.89		
	T	s	1.60		
	δ	—	3.00		
	b	$\mathrm{m/s^2}$	1.107		
ACC	k_1	$\mathrm{s^{-1}}$	0.4[a]	7.78	8.79
	k_2	$\mathrm{s^{-1}}$	0.53		
	T	s	1		
	k_3	$\mathrm{s^{-1}}$	0.087		
DL	Layers	—	3 (128, 64, 16)	9.82	10.98
	Activation function	—	Relu (all layers)		
	Loss function	—	Mean-square error		
	Optimizer	—	RMSprop		
	Epoch	—	250		
	Metrics	—	Mean-square error		

[a] Data from Xiao et al. (2017).

model the following behavior of Waymo, leader–follower combinations of a Waymo self-driving car-following human-driven vehicle data were used. From the Waymo data set, there are 196 leader–follower pairs of a Waymo vehicle following a human-driven vehicle.

Further, the calibration of car-following models plays an important role in mimicking the car-following behavior. Traditionally, researchers employed numerous optimization techniques to calibrate car-following models by fine-tuning the parameters. At the same time, the choice of calibration and optimization algorithm depends heavily on the specific nature of the problem, such as the data set, objective function, and computational resources. While some methods are efficient for smooth, convex problems, heuristic approaches may be more suitable for complex or nonconvex problems. But each comes with its own set of limitations, such as converging to local maxima/minima, parameter sensitivity, etc. Considering the complex model formulation of car-following models, over time, researchers relied on genetic algorithms (GAs) to calibrate these models. GAs are flexible and efficient in handling nonlinear and multiparameter optimization problems. Using the GA (Katoch et al. 2021; Li and Li 2021), the parameters of the models were optimized to minimize the mean absolute percentage error (MAPE) between the observed acceleration and the modeled acceleration as shown in Eq. (23). To calibrate the IDM and ACC, the trajectory data of the leader–follower pairs were divided into two batches of 70% and 30% for calibration and validation, respectively. The details of the calibrated parameters, calibration, and validation errors of the respective models are presented in Table 1. From the analysis, it is observed that the errors are in the range of 7.5% to 11% MAPE. Comparing these MAPE values to previous studies (Raju et al. 2020; Zheng et al. 2012) reveals that the car-following models were well calibrated and implies that the deviation from real-world traffic flow is relatively small. The DL model was calibrated using the RMSProp optimizer with 250 epochs

$$\text{objective function} = \operatorname{argmin}\left[\frac{1}{n}\sum_{i=1}^{n}\left[\frac{|a_{\text{observed}} - a_{\text{estimated}}|}{a_{\text{observed}}}\right]_i \cdot 100\right] \tag{23}$$

Modeling the Controllers

Controllers play a major role in passing a controlled input, in this case the speed information, from the response of the car-following model to the system and keeping the system in a balanced state. As a controller the proportional-integrative-derivative (PID) (Patel 2020; Sarhadi et al. 2016) is used in the present study. In a closed-loop control system, the controller checks the error between the target state (target longitudinal speed) and the system state (system longitudinal speed) and passes the control signal (longitudinal speed) to the actuator. The control function of the PID is given as

$$X_1(t) = K_p \cdot E(t) + K_i \int_0^t E(\tau)d\tau + K_d \cdot \frac{dE(t)}{dt} \tag{24}$$

The transfer function of the PID controller in the complex domain is given as

$$C_{\text{PID}}(s) = K_p + \frac{K_i}{s} + K_d s \tag{25}$$

where $X_1(t)$ = control speed outcome from the controller at time t; $E(t)$ = speed error at time t; s = signal form in Laplace domain; K_p = proportional weight; K_i = integral weight; K_d = derivative weight; and $C_{\text{PID}}(s)$ = transfer function of the PID controller.

In the present case, three combinations P, PI, and PID controllers were examined with the selected car-following models. The control function and the transfer function will be adjusted based on the specific combination. It was assumed that both the sensors and actuators work without any lag and pass unconditional information from them. Given this, the transfer function [Eq. (7)] for the entire loop takes the form as in Eq. (26)

$$G(s) = \frac{CF(s) \cdot C(s)}{[1 + CF(s) \cdot C(s)]} \tag{26}$$

The loop transfer function can be derived using Eq. (26) based on the car-following model and the controller combination. The tuning of the controllers plays a major role in balancing the performance and stability. Based on Ziegler and Nichols's method, the

Table 2. Computation of controller weights based on K_u and P_u

Control type	K_p	K_i	K_d
P	$0.5K_u$	—	—
PI	$0.45K_u$	$0.54K_u/T_u$	—
PID	$0.6K_u$	$1.2K_u/T_u$	$3K_uT_u/40$

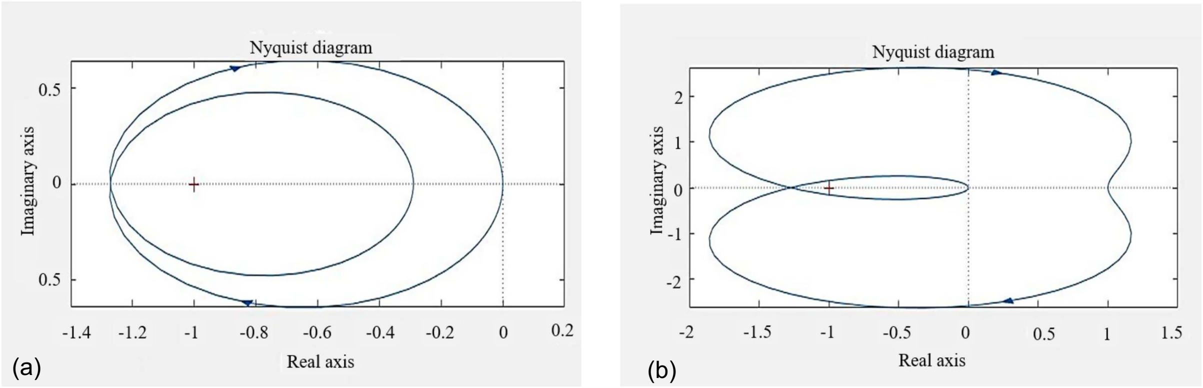

Fig. 3. Nyquist plot for IDM and ACC models: (a) IDM; and (b) ACC.

weights were tuned as reported in Table 2. Further to understand the locus of the poles and assess the stability of the system for a given car-following model, Nyquist plots were used as presented in Fig. 3, and the tuned weights for the IDM and ACC are shown in Table 3. Unlike the other models, the DL model does not have a clear mathematical formula for tuning the controller weights. To overcome this complexity, we considered the weights from either the IDM or ACC model as a starting point for the DL model. Given the linear model of the ACC compared to IDM, the ACC's weights were chosen for the initial DL model training, and the weights of the ACC model were adapted. Ziegler-Nichols' method involves perturbing a system and observing its response to derive critical parameters, aiding in tuning controllers for optimal stability and performance. The Nyquist plot, on the other hand, provides insights into a system's stability by mapping its frequency response. Integrating both, car-following models can be fine-tuned using Ziegler-Nichols' response curve analysis and Nyquist plots to ensure controllers strike a balance between responsiveness and stability. From Fig. 3, it can be noted that, in the case of ACC [Fig. 3(b)], there is a probability for the system being unstable given some part of locus lying on the right side of the complex plane. Meanwhile, in the case of the IDM, the locus of the poles is on the left side of the complex plane demonstrating model stability.

Along with the longitudinal movement, the lateral movement of the vehicles plays a major role in the overall local planning of the AVs' movement. In the present work, the Stanley controller (AbdElmoniem et al. 2020) was used to regulate the lateral movement and avoid the intricacy; the bicycle kinematics model (Polack et al. 2017) is adopted to reduce complexity. The bicycle model effectively captures the essential dynamics of lateral movement, namely, the relationship between steering angle and vehicle heading. Its relatively low dimensionality and its simplicity align well with the Stanley controller's design, enabling smooth integration and efficient trajectory tracking. While more intricate vehicle models can offer higher fidelity representation of dynamics, these often come at the cost of increased complexity and computational demands.

Table 3. Controller parameters for IDM and ACC

Model	Control type	K_p	K_i	K_d
IDM	P	0.392	—	—
	PI	0.353	0.186	—
	PID	0.463	0.41	0.125
ACC and DL	P	0.391	—	—
	PI	0.352	0.065	—
	PID	0.462	0.141	0.358

The Stanley control is a path tracking approach tht uses the front axle as its reference point. Therefore, the steering angle is given as a function of the heading error, cross-track error, and vehicle's speed as follows:

$$\phi(t) = \psi(t) + \tan^{-1}\left[\frac{k \cdot e_{\text{cross}}(t)}{v(t)}\right]$$
$$\phi(t) \in [\phi_{\min}, \phi_{\max}] \tag{27}$$

where $\psi(t)$ = heading angle; $\phi(t)$ = steering angle; $e_{\text{cross}}(t)$ = cross-track error; k = controller weight (in the present study it is assumed as 2.5); and $v(t)$ = system speed.

Simulation

To test the hypothesized longitudinal behavior, nanoscopic simulation tools can be handy. In this study, town one environment from the CARLA network is used to simulate the urban traffic scenario. To simulate a realistic traffic environment, around 150 unique vehicles were spawned into the road network, creating various levels of vehicle interactions. These vehicles operate as independent entities and are connected to the Unreal game engine, which is responsible for their rendering and behavior. All other traffic-related elements, such as traffic signals and the actions of the vehicles, are controlled by CARLA's traffic manager, which automates these aspects. For the ego vehicle, Tesla Model 3 was selected, and its behavior was externally controlled via a python API. This API primarily captures data from the simulation model, processes it based on the predefined control logic, and then sends commands to the simulation model, which controls the ego vehicle's movement. For each of the test cases, the simulation was run using three different seeds to ensure variability and reliability. Each test ran for a total of 300,000 simulation steps, with the simulation updating at a rate of 20 H.

Results and Analysis

Based on the calibrated parameters of the controllers and the car-following models using the python API, simulation runs were performed, and the trajectory data of the subject vehicle were extracted for the analysis. To understand the system's performance with different control strategies, microscopic variables such as target speed (with the car-following model), system speed (after the control loop), relative distance with leader, acceleration, cross-track error, and heading error were computed. To depict the functioning of the

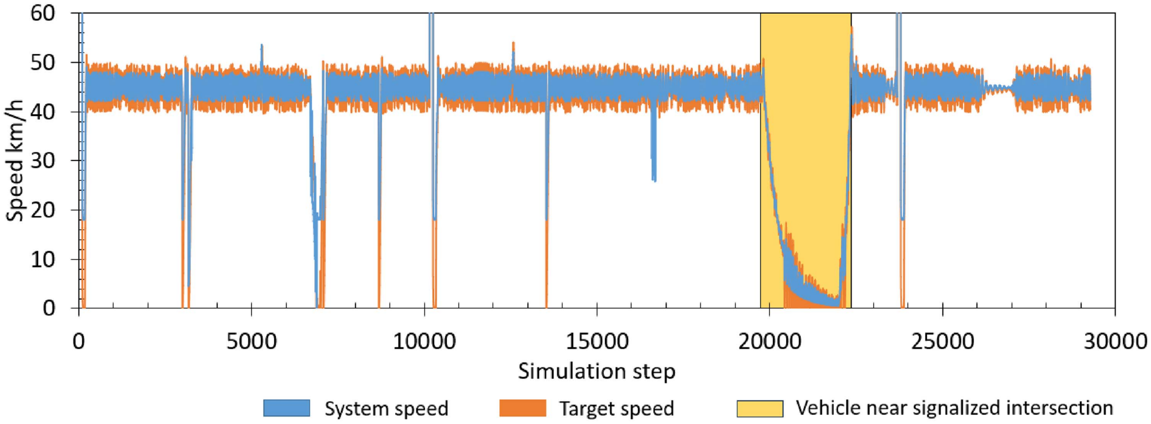

Fig. 4. System speed and target speed of the vehicle over the simulation steps.

car-following model and the controller more effectively, a typical plot of target speed and system speed over the simulation steps is shown in Fig. 4. A proportional-integral (PI) ACC model is used. In the specific scenario presented, the ego vehicle followed its leader within a town network in the Carla simulation setup along with other vehicles on the road, resulting in some variations in the speed profile (speed variation in Fig. 4), and the controller tried to balance these fluctuations. At around the 20,000th simulation step, the ego vehicle approached a signalized intersection and encountered a red light. As a result, there was a drop in speed from 20,000 and a complete standstill at the 22,000th step. It can be observed from Fig. 4 that the outcomes from the car-following model (i.e., the target speed—orange color) tend to have larger perturbations compared to that when the controller is applied (i.e., the system speed—blue color). On the other hand, the controller tends to limit the perturbations and models the control in a much more stable form.

System Speed versus Target Speed

To understand the system's response to the controller and the car-following model, the target speed and system speed were plotted, as shown in Fig. 5. From these plots, it is observed that the correlation trend between the system speed and the target speed varies with the car-following models. In the case of IDM, the correlation has taken a parametric form (i.e., the formulation of the IDM). With ACC, the trend is linear, and with the DL model, the trend seems to also be linear but with a visible randomness when the target speed is higher than the system speed. The trends were further altered with the controller formulation, from P to PID.

For the IDM, its formulation, which targets to achieve the desired speed by harmonizing the relative distance from the leader, played a considerable part in its performance. In the case of free-flow conditions, at more considerable relative distances, the parameter δ of IDM dictates the inclination for the desired speed. For $\delta = 1$ the speed of the vehicle increases linearly to the desired speed like the ACC model. Meanwhile, in the calibrated IDM model, $\delta = 3$, resulting in a cubic parabolic relation in achieving the desired speed. On the other hand, the other interaction terms come into the picture at higher speed differences with the leader vehicle, and the subject vehicle speed is regulated with comfortable deceleration. At smaller relative distances, the distance term comes into the picture and ties for balancing the relative distance to maximize the speed. From the equations (partial derivates of the IDM), it can be noted that the transfer function of the IDM is elaborative in comparison to the ACC. In combination with the formulation aspects, the IDM model adds lag to the system with a parametric relationship, as shown in Fig. 5.

In the case of ACC, a linear trend is observed between the system speed and the target speed. For the proportional controller (P), a lag in the system was observed at low speeds (less than 25 km/h), and it tends to diminish at higher speeds. With the proportional integrative controller (PI), this difference in the lag between low and high speeds is found to decrease compared to the proportional controller. In the case of PID, ACC's performance has improved further and catches a more linear correlation form. The ACC model correlation is more linear in form with an objective of either cruising or matching its leader notions. As a result of these aspects, the trend of ACC is more reliable than the IDM.

Unlike the IDM and ACC, the predicted target speed is more random for the DL model, given its temporal independence in predictions. However, the performance of the DL model is fully dependent on the trained data. From Fig. 5, it can be noted that the correlation is linear between the target speed and system speed. At the same time, unlike the ACC, there is not much improvement in correction from P to PID controllers.

To evaluate the results better, the root-mean-square error (RMSE) of the difference between the target speed and the system speed was computed for each of the combinations, and the results are depicted in Table 4. The results show that the RMSE is in the range of 3.2 to 11.5 km/h. In the case of IDM and the DL model, there is not much decrease in the RMSE as the controller changes from P to PID. Meanwhile, in the case of ACC, the RMSE decreased from P to PID and demonstrated performance improvement.

System Speed versus Relative Distance

To understand the nature of the car-following behavior in combination with the controller, the system's speed was plotted against the relative distance, as shown in Fig. 6. In the case of the IDM, the model tends to follow a linear relationship between the system's speed and the relative distance. The IDM's formula includes a distance term that directly affects the speed. As the relative distance between the ego vehicle and the lead vehicle decreases, this term adjusts the target speed by reducing it and ultimately influencing the system speed. In this case the drop in speed is linear. At the same time, it can be noted that the exponent parameter (δ) of the IDM changes, and the relationship between system speed and relative distance varies in different nonlinear forms. Furthermore, this trend is observed with all controller combinations in IDM. From P [IDM] to PID[IDM], the data become more dispersed.

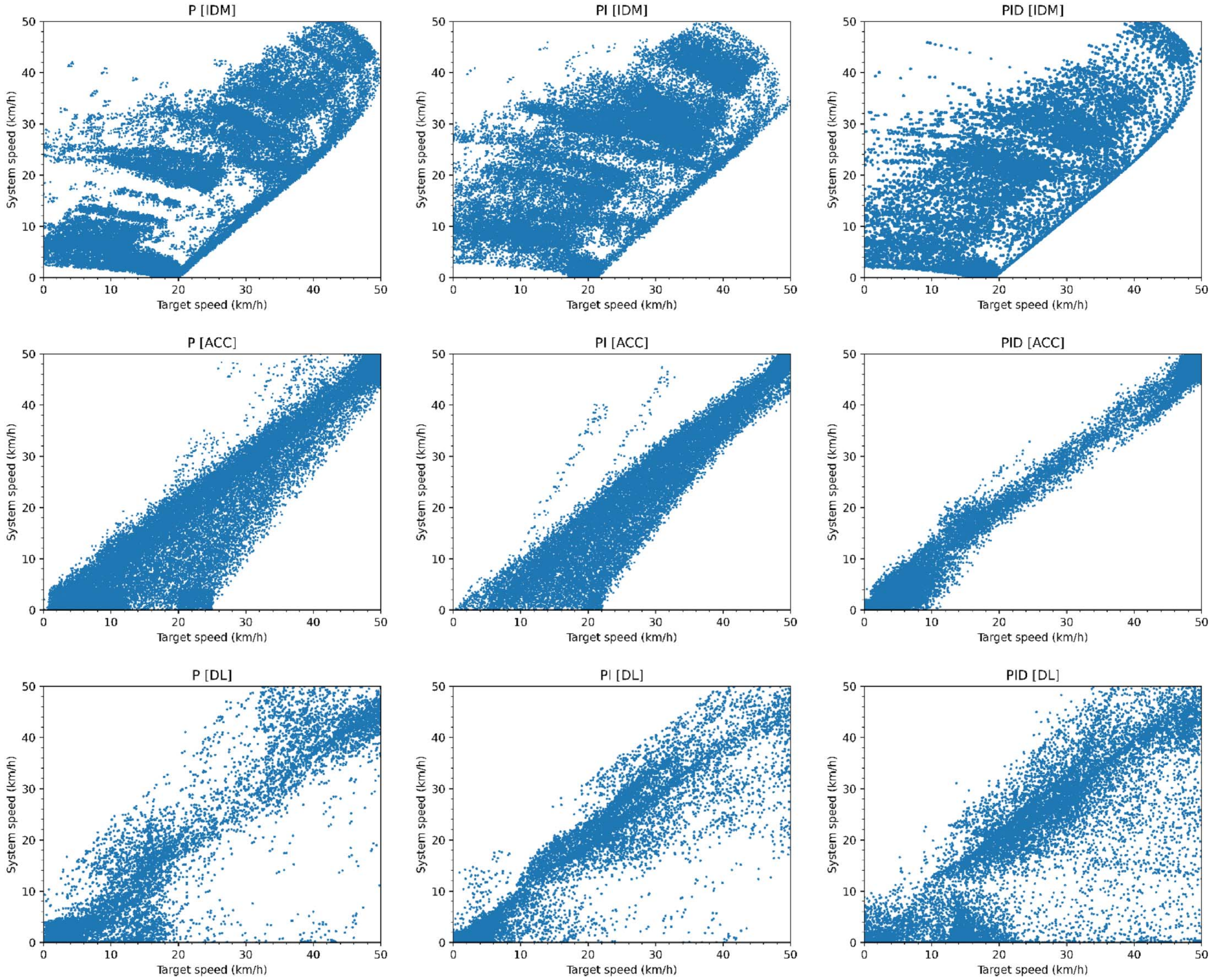

Fig. 5. System speed versus target speed for different car-following and controller combinations.

On the other hand, in the case of ACC, the ego vehicle tends to maintain its desired speed. From the plots, it can be noted that, with ACC, two sets of distinct following patterns are observed. One is cruising phase, where the ego vehicle moves at its desired speeds, and other is following phase, where the ego vehicle adjusts its speed based on the distance to the leading vehicle. When the relative distance is greater than a certain threshold, the system continues to maintain the desired speed without any adjustments {e.g., Fig. 6 PID[ACC] at around 25 m}. However, once the relative distance decreases past this threshold, the system begins to decrease its speed in a linear manner. This means that as the gap between the vehicles gets smaller {e.g., Fig. 6 PID[ACC] less than 20 m}, the ACC reduces the speed of the ego vehicle to ensure safe following behavior.

Table 4. RMSE over the combinations

Controller	Car-following model	RMSE (km/h)
P	IDM	9.6
PI		10.1
PID		8.9
P	ACC	9.1
PI		4.9
PID		3.2
P	DL	11.5
PI		10.8
PID		10.5

In the case of P[ACC] and PI[ACC], there are some inconsistencies observed at certain stages in system (some data points that deviate from the main trend), and the same can be observed in the data patterns. Meanwhile, in the case of the PID[ACC] the system is comparatively more stable, resulting in a clear data pattern.

While the ACC maintains a consistent behavior as it manages the vehicle's speed based on the distance to the leading vehicle, the DL model tends to show more fluctuations in system speed. In some specific instances, the DL model has predicted higher speeds, even when the relative distance to the leading vehicle is shorter (e.g., relative distance 10–15 m and system speed between 20 and 50 km/h). This indicates that the DL model can sometimes prioritize speed over maintaining a safe following distance. The unpredictability of the DL model can be attributed to its reliance on learning from various scenarios and adapting its speed. Further,

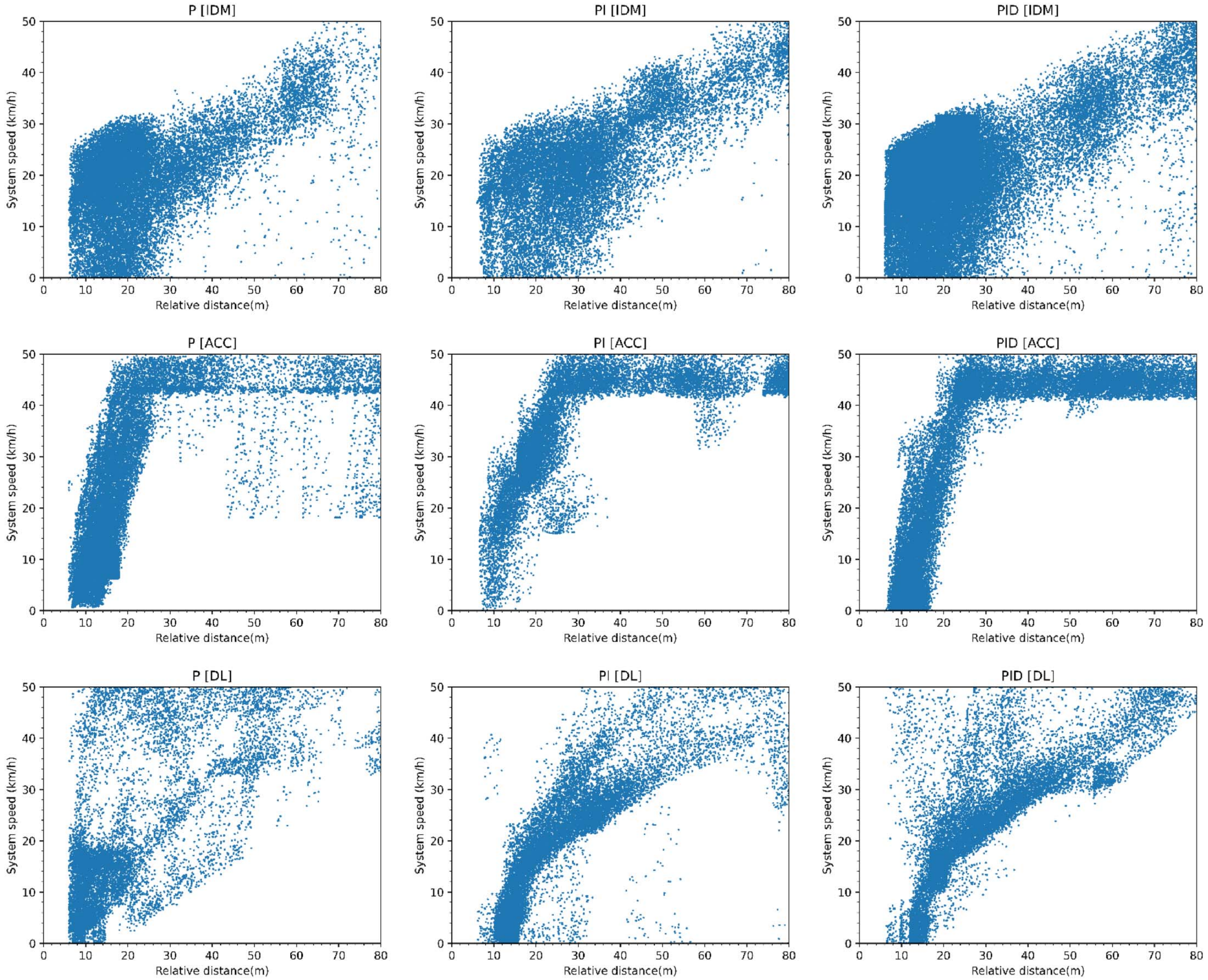

Fig. 6. System speed versus relative distance for different car-following and controller combinations.

moving from PI[DL] to PID[DL], it is observed that there is slight improvement, and the controllers help to stabilize the system, leading to more consistent speed adjustments. As a result, the randomness in the system's behavior is marginally decreased, especially at shorter relative distances. From the analysis, it is observed that the car-following formulations play a significant role in guiding the speed of the ego vehicle. This results in different relational patterns across various car-following models. On the other hand, the combination of controllers contributes to stabilizing speed adjustments, ensuring smoother interactions between vehicles on the road.

Conclusions

Summary

In line with the study's objective, two mathematical car-following models (intelligent driver model and adaptive cruise control), along with a deep learning model from the realm of artificial intelligence (deep learning model), were integrated with different types of controllers (proportional, integrative, and derivative controllers) to evaluate and understand their influence on the longitudinal behavior planning of AVs. It is revealed that the specific formulation of a car-following model in combination of a specific formulation of a controller have an impact on the local planning of AVs. Additionally, various combinations of controllers and car-following models significantly influence the overall system performance. More specifically, it is shown that the ACC model, which uses a linear approach to achieve the desired speed and maintains relative distance by adjusting to the leader vehicle's speed, demonstrates strong compatibility with the tested controllers. Consequently, the ACC model outperformed both the IDM and the deep learning model. In the case of the IDM, the inclusion of the relative distance term and the exponential factor for calculating desired speed, when combined with the controllers, introduced a lag in the system by reducing AV speeds from greater distances. This added delay contributed to less efficient performance compared to the ACC model.

The IDM resulted in a nonlinear correlation between the system and target speeds, meaning limited performance. Even though the DL model has accuracy in line with the IDM and ACC model, the system tends to have irregular speeds when applying the DL car-following. The main problem of the DL model is its temporal independence in predictions. As a result, the outcomes from the DL model at successive time steps could be extensively different. Given this, the DL model impacted the system's performance at all levels.

It can be concluded from the analysis that ACC with PID controller combination tends to have better performance than all the other tested combinations. Given the linear formulation of the ACC, the PID controller can regulate the error in a precise manner without deteriorating the performance. Given this, the PID [ACC] combination has good compatibility with the lateral Stanley controller with minimum eccentricity in cross-track and heading error. However, the performance of the IDM is comparative and in line with the ACC, which is visible in cross-track error, heading error, and acceleration. The present study's DL analysis explicitly highlights the importance of a consistent, continuous model for generating the behavior. At the same time, considering the accuracy, it is inferred that the training data of the DL model play a considerable role in the overall DL performance. The present study suggests that training the DL model with more deterministic data with less randomness will result in more decisive DL predictions and improve performance.

Discussion

From the research, it is noted that in AV microsimulation studies, researchers heavily rely on car-following models to simulate longitudinal movement. However, simply applying the outcomes from these models does not fully capture the true nature of AV behavior. By integrating a controller on top of the car-following models, as adopted in this study, the longitudinal behavior in traffic simulation can be represented more realistically. This approach enables future traffic modeling studies to produce more accurate insights into the impact of increasing AV penetration rates on traffic efficiency.

Findings from this study suggest that car-following models that generate continuous and consistent outputs, such as IDM and ACC, perform better than the DL model, which treats consecutive observations as independent. AV motion planners can utilize the study's methodology to determine the optimal combination of a controller and a car-following model, evaluating system performance before real-world testing. Despite its contributions, this study has certain limitations that future research can address. The selection of IDM and ACC models was based on their formulation and widespread use in AV modeling literature; however, other car-following models can be explored in future studies using the proposed framework. Additionally, while this study assumed perfect sensor and actuator performance, real-world conditions introduce inaccuracies. Future research should consider modeling sensor and actuator transfer functions to assess overall system performance better.

In the present work, only the DL model has been tested, and its performance relies entirely on the patterns learned from the trained data. While this approach enables the model to effectively capture complex relationships within the data set, it does not incorporate adaptive learning from real-time interactions. However, the current framework has the flexibility to be extended with reinforcement learning (RL), which would allow the model to continuously refine its predictions by interacting with dynamic environments. By integrating RL, the system could iteratively improve decision-making based on feedback, leading to more robust and adaptive performance in real-world driving scenarios.

Data Availability Statement

Some data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request. This includes the simulation model and data generated from simulation runs.

Author Contributions

Narayana Raju: Conceptualization; Data curation; Formal analysis; Methodology; Resources; Software; Supervision; Writing – original draft; Writing – review and editing.

References

AbdElmoniem, A., A. Osama, M. Abdelaziz, and S. A. Maged. 2020. "A path-tracking algorithm using predictive Stanley lateral controller." *Int. J. Adv. Rob. Syst.* 17 (6): 1729881420974852. https://doi.org/10.1177/1729881420974852.

Bansal, M., A. Krizhevsky, and A. Ogale. 2019. "ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst." Preprint, submitted December 7, 2018. https://arxiv.org/abs/1812.03079.

El-Badry, W. 2021. "Automated Ziegler Nicholas PID tuning." Accessed May 4, 2022. https://nl.mathworks.com/matlabcentral/fileexchange/47836-automated-ziegler-nicholas-pid-tuning.

Fayolle, G., J.-M. Lasgouttes, and C. Flores. 2020. "Stability and string stability of car-following models with reaction-time delay." *SIAM J. Appl. Math.* 82 (5): 1661–1679. https://doi.org/10.1137/21M1443650.

Gao, H., G. Shi, G. Xie, and B. Cheng. 2018. "Car-following method based on inverse reinforcement learning for autonomous vehicle decision-making." *Int. J. Adv. Rob. Syst.* 15 (6): 1729881418817162. https://doi.org/10.1177/1729881418817162.

Gong, S., J. Shen, and L. Du. 2016. "Constrained optimization and distributed computation based car following control of a connected and autonomous vehicle platoon." *Transp. Res. Part B Methodol.* 94 (Dec): 314–334. https://doi.org/10.1016/j.trb.2016.09.016.

Google. 2021. "Waymo." Accessed July 22, 2021. https://waymo.com/open/download/.

Gunter, G., et al. 2020. "Are commercially implemented adaptive cruise control systems string stable?" *IEEE Trans. Intell. Transp. Syst.* 22 (11): 6992–7003. https://doi.org/10.1109/TITS.2020.3000682.

Hawkins, A. J. 2019. "Waymo is making some of its self-driving car data available for free to researchers." *The Verge*, August 21, 2019.

Hu, X., Z. Zheng, D. Chen, X. Zhang, and J. Sun. 2022. "Processing, assessing, and enhancing the Waymo autonomous vehicle open dataset for driving behavior research." *Transp. Res. Part C Emerging Technol.* 134 (Nov): 103490. https://doi.org/10.1016/j.trc.2021.103490.

Hu, X., Z. Zheng, X. Zhang, D. Chen, and J. Sun. 2021. "Vehicle trajectory data processed from the Waymo Open Dataset." Accessed December 21, 2023. https://doi.org/10.17632/wfn2c3437n.2.

Katoch, S., S. S. Chauhan, and V. Kumar. 2021. "A review on genetic algorithm: Past, present, and future." *Multimedia Tools Appl.* 80 (5): 8091–8126. https://doi.org/10.1007/s11042-020-10139-6.

Kesting, A., M. Treiber, and D. Helbing. 2010. "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity." *Phil. Trans. R. Soc. A* 368 (1928): 4585–4605. https://doi.org/10.1098/rsta.2010.0084.

Lasch, T., J. Imhof, R. Sarc, and K. Khodier. 2024. "Development of a method for shredder characterisation and dynamic control of the output material stream." *J. Sustainable Dev. Energy Water Environ. Syst.* 12 (3): 1–10. https://doi.org/10.13044/j.sdewes.d12.0492.

Li, S., and D. Li. 2021. "Genetic algorithms." In Vol. 312 of *Artificial intelligence for materials science: Springer series in materials science*. Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-030-68310-8_5.

Ma, M., G. Ma, and S. Liang. 2021. "Density waves in car-following model for autonomous vehicles with backward looking effect." *Appl. Math. Modell.* 94 (Jun): 1–12. https://doi.org/10.1016/j.apm.2021.01.002.

Makridis, M., K. Mattas, A. Anesiadou, and B. Ciuffo. 2021. "OpenACC: An open database of car-following experiments to study the properties of commercial ACC systems." *Transp. Res. Part C Emerging Technol.* 125 (Apr): 103047. https://doi.org/10.1016/j.trc.2021.103047.

Makridis, M., K. Mattas, C. Mogno, B. Ciuffo, and G. Fontaras. 2020. "The impact of automation and connectivity on traffic flow and CO_2 emissions. A detailed microsimulation study." *Atmos. Environ.* 226 (Apr): 117399. https://doi.org/10.1016/j.atmosenv.2020.117399.

Manaswi, N. K. 2018. *Deep learning with applications using python*. Berkeley, CA: Apress.

Milanés, V., and S. E. Shladover. 2014. "Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data." *Transp. Res. Part C Emerging Technol.* 48 (Nov): 285–300. https://doi.org/10.1016/j.trc.2014.09.001.

Olstam, J. J., and A. Tapani. 2004. *Comparison of car-following models*. Linkoping, Sweden: Swedish National Road and Transportation Research Institute.

Patel, V. V. 2020. "Ziegler-Nichols tuning method: Understanding the PID controller." *Resonance* 25 (10): 1385–1397. https://doi.org/10.1007/s12045-020-1058-z.

Patil, S., N. Raju, S. Arkatkar, and S. Easa. 2021. "Modeling vehicle collision instincts over road midblock using deep learning." *J. Intell. Transp. Syst.* 27 (2): 257–271. https://doi.org/10.1080/15472450.2021.2014833.

Polack, P., F. Altche, B. d'Andrea-Novel, and A. de La Fortelle. 2017. "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" In *Proc., IEEE Intelligent Vehicles Symp.*, 812–818. New York: IEEE. https://doi.org/10.1109/IVS.2017.7995816.

Raju, N., S. Arkatkar, and G. Joshi. 2020. "Modeling following behavior of vehicles using trajectory data under mixed traffic conditions: An Indian viewpoint." *Transp. Lett.* 13 (9): 649–663. https://doi.org/10.1080/19427867.2020.1751440.

Raju, N., and H. Farah. 2021. "Evolution of traffic microsimulation and its use for modeling connected and automated vehicles." *J. Adv. Transp.* 2021 (1): 2444363. https://doi.org/10.1155/2021/2444363.

Raju, N., W. Schakel, N. Reddy, Y. Dong, and H. Farah. 2022. "Car-following properties of a commercial adaptive cruise control system: A pilot field test." *Transp. Res. Rec.* 2676 (7): 128–143. https://doi.org/10.1177/03611981221077085.

Sarhadi, P., A. R. Noei, and A. Khosravi. 2016. "Model reference adaptive PID control with anti-windup compensator for an autonomous underwater vehicle." *Rob. Auton. Syst.* 83 (Sep): 87–93. https://doi.org/10.1016/j.robot.2016.05.016.

Scanlon, J. M., K. D. Kusano, T. Daniel, C. Alderson, A. Ogle, and T. Victor. 2021. "Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain." *Accid. Anal. Prev.* 163 (Dec): 106454. https://doi.org/10.1016/j.aap.2021.106454.

Sun, J., Z. Zheng, and J. Sun. 2018. "Stability analysis methods and their applicability to car-following models in conventional and connected environments." *Transp. Res. Part B Methodol.* 109 (Mar): 212–237. https://doi.org/10.1016/j.trb.2018.01.013.

Treiber, M., A. Hennecke, and D. Helbing. 2000. "Congested traffic states in empirical observations and microscopic simulations." *Phys. Rev. E* 62 (2): 1824. https://doi.org/10.1103/PhysRevE.62.1805.

Wang, Y., H. Farah, R. Yu, S. Qiu, and B. van Arem. 2023. "Characterizing Behavioral differences of autonomous vehicles and human-driven vehicles at signalized intersections based on waymo open dataset." *Transp. Res. Rec.* 2677 (11): 324–337. https://doi.org/10.1177/03611981231165783.

Wang, Y. N., H. X. Ge, S. M. Lo, K. L. Tsui, and K. K. Yuen. 2020. "Car-following model with considering vehicle's backward looking effect and its stability analysis." In *Proc., Int. Conf. on Computational Methods*, 531–538. New Delhi, India: ST Publisher.

Wen-Xing, Z., and Z. Li-Dong. 2018. "A new car-following model for autonomous vehicles flow with mean expected velocity field." *Physica A* 492 (Feb): 2154–2165. https://doi.org/10.1016/j.physa.2017.11.133.

Xiao, L., M. Wang, and B. Van Arem. 2017. "Realistic car-following models for microscopic simulation of adaptive and cooperative adaptive cruise control vehicles." *Transp. Res. Rec.* 2623 (1): 1–9. https://doi.org/10.3141/2623-01.

Zaghari, N., M. Fathy, S. M. Jameii, M. Sabokrou, and M. Shahverdy. 2021. "Improving the learning of self-driving vehicles based on real driving behavior using deep neural network techniques." *J. Supercomput.* 77 (4): 3752–3794. https://doi.org/10.1007/s11227-020-03399-4.

Zhang, L. D., W. X. Zhu, and J. L. Liu. 2014. "Proportional-differential effects in traffic car-following model system." *Physica A* 406 (Jul): 89–99. https://doi.org/10.1016/j.physa.2014.03.001.

Zheng, J., K. Suzuki, and M. Fujita. 2012. "Evaluation of car-following models using trajectory data from real traffic." *Procedia Social Behav. Sci.* 43 (Jan): 356–366. https://doi.org/10.1016/j.sbspro.2012.04.109.