International Journal of Science and Engineering
http://www.scieng.nutn.edu.tw/ch/item.php
International Journal of Science and Engineering (issued by NUTN) Vol.12 No.1(2022):15-24

Adaptive Cruise Vehicle Using Deep Learning Model Predictive Control

Pachara Opattrakarnkul and Chung-Neng Huang*

Graduated Institute of Mechatronic System Engineering, National University of Tainan, Tainan, 70005, Taiwan
Email: pachara_35@hotmail.com, kosono@mail.nutn.edu.tw

Abstract

Adaptive cruise control (ACC) systems are designed to provide longitudinal assistance to enhance safety and driving comfort by adjusting vehicle velocity to maintain a safe distance between the host vehicle and the preceding vehicle. Generally, using model predictive control (MPC) in ACC systems provides high responsiveness and lower discomfort by solving real-time constrained optimization problems but results in computational load. This paper presents an architecture of deep learning based on model predictive control in ACC systems to avoid real-time optimization problems required by MPC, which in turn, reduces computational load. The learning dataset is acquired from the simulation data of the input/output of the MPC controller. We designed the proposed deep learning controller using long short-term memory networks (LSTMs) and simulated it in MATLAB/Simulink using the vehicle's characteristics from the advanced vehicle simulator (ADVISOR). Finally, the safety and driving comfort are compared with the PID-based control to demonstrate the performance of the proposed deep-learning architecture.

***Keywords* -- Adaptive Cruise Control, Model Predictive Control, Deep Learning.**

----

* Corresponding author: kosono@mail.nutn.edu.tw
DOI：10.53106/222344892022041201002

I. INTRODUCTION

Every year, around 20 to 50 million people suffer from injury and disability from road traffic crashes and approximately 1.35 million deaths [1]. The most common type of road traffic crash is rear-ended collision. It occurs when a vehicle crashes into another vehicle in the front. Rear-ended collision accounts for 29% of all crashes and has 7.2% of fatality [2, 3], and most of them occur due to human error. Advanced driver-assistance systems (ADAS) are automated systems developed to enhance safety and driving comfort by alerting, warning, assisting, or taking control of the vehicle if necessary. ADAS vehicles are integrated with modern technology, such as sensors and cameras to detect anomalies or obstacles and respond accordingly or alert drivers through a human-machine interface [4].

One of the most important features of ADAS is adaptive cruise control (ACC). ACC is a longitudinal control system that can automatically adjust vehicle velocity to maintain a safe distance between the host and the preceding vehicle. The behavior of the system can separate into two modes – cruise mode and following mode. In cruise mode, in the absence of the preceding vehicle, it acts like conventional cruise control by maintaining the driver's desired constant velocity. However, when its sensors, either radar, lidar, or even camera, detect a preceding vehicle that moves slower than its cruise velocity, it will enter the following mode by braking or accelerating to adjust its speed to maintain a safe distance.

In the ACC systems, there is an essential part called the spacing policy, it refers to the desired spacing distance between the host and preceding vehicle during vehicle following. The spacing policy of an ACC system is the key to satisfy not just the safe distance but also various aspects such as traffic capacity, energy consumption, driving comfort, and safety [5, 6]. To achieve these goals, many approaches have been done to improve the spacing policy [7, 8, 9]. However, the current policies cannot satisfy stability, safety, and comfort at the same time, so some trade-off must be made [10]. To ensure that specification is met, an appropriate controller is needed.

In general, the ACC system consists of at least two levels of controller [11] – a lower-level controller and an upper-level controller. The lower-level controller determines throttle and brake commands from vehicle dynamics and driveline dynamics data to track the desired acceleration. And the upper-level controller determines the desired acceleration to ensure that the spacing stability as mentioned is met. There is plenty of research on ACC that has been done by applying a different control approach, PID-based control [12, 13, 14] is the most commonly used. Besides PID, there are also fuzzy logic controls [15, 16], model predictive control (MPC) [17, 18], and even neural networks (NN) [19, 20, 21] that have been proposed in recent literature.

The MPC controller has been used in the industrial process since the 1980s, as the name suggests, is a feedback control method that relies on a model of complex dynamical systems and a real-time optimization solver. Until now, there is much research related to longitudinal control using MPC – stop and go [22], energy-optimal [23], safety guarantee [24], and multi-objective [25, 26, 27]. The advantages of using MPC in the ACC systems are – it is easy to achieve precise and optimal control, capable of real-time multi-objective optimal control, and even provides high responsiveness during traffic jams [28, 29]. This is due to its capability of optimizing a finite time horizon, handling multiple input/output (MIMO), and constraints in a receding horizon.

In recent years, deep learning has been in the spotlight of a wide range of systems and applications. Deep learning is a part of machine learning (ML) methods based on artificial neural networks (ANN) with representation learning [30]. The word "deep" in deep learning refers to the use of multiple layers and neurons between the input and output layers of the network, which significantly improve learning performance and accuracy. With that, Deep learning can resolve various large complex applications, features, or problems such as speech recognition [31], handwriting recognition [32], and even forecasting [33] et cetera. This has led to the increasing research on deep learning in the ACC systems [34, 35, 36].

With the advantage and robustness of MPC in the ACC system and the upcoming development of deep learning, this research aims to investigate the performance of deep-learning model predictive control (DMPC) for ACC systems. Several pieces of research have implemented deep learning from MPC and point out that DMPC can reduce computational load, maintain robustness, and compute faster in a dynamical real-time environment [37, 38, 39]. We designed and simulated the proposed model in the MATLAB/Simulink environment. For a better simulation result, the vehicle dynamics model needs to be as realistic as possible. To reduce the complication of designing vehicle dynamics models, we used the vehicle models from advanced vehicle simulators (ADVISOR) and integrated them with a constant time-headway (CTH) policy to create the ACC systems. Then, we designed the MPC controller that is capable of achieving optimal control performance. After that, we trained the deep learning model via supervised learning from MPC's simulated input and output data. Finally, we demonstrated the comparison of the PID and DMPC-controlled ACC systems is shown to demonstrate the performance of the proposed controller.

II. MODEL PREDICTIVE CONTROL DESIGN

A. State-space model for ACC system

The vehicle dynamics model is needed when designing, analyzing, and simulating an MPC controller. In fact, the dynamics of vehicles are nonlinear. The characteristics such as engine, transmission, brake, and drag force are all nonlinear. The dynamics between input – desire acceleration a_{des} and output – actual acceleration a can be described as an ordinary differential equation [17]:

$$a = \frac{K_L}{T_L s + 1} a_{des}. \tag{1}$$

where K_L is the system steady-state gain and T_L is the time constant of acceleration using the engine. For ACC systems, the inter-vehicle dynamics state variables are defined as – inter-velocity error Δv and inter-distance error Δd:

$$\Delta v = v_{lead} - v_{follow}. \tag{2}$$

$$\Delta d = d_{rel} - d_{safe}. \tag{3}$$

where v_{lead} and v_{follow} are velocity of the lead and follow vehicle respectively. The d_{rel} and d_{safe} are inter-vehicle relative distance and inter-vehicle safe distance given by:

$$d_{rel} = \int \Delta v(t) dt. \tag{4}$$

$$d_{safe} = v_{follow} t_h + d_{min}. \tag{5}$$

where t_h is the constant time-headway and d_{min} is the minimum stopping distance at zero velocity. With the formulation of (1), (2), and (3), the continuous-time state-space model of ACC systems can be formulated as [29]:

$$\dot{x} = Ax + Bu + Gv + Hw. \tag{6a}$$

$$y = Cx + Jw. \tag{6b}$$

where

$$A = \left[0\ 1\ -t_h\ 0\ 0\ -1\ 0\ 0\ \frac{-1}{T_L}\right], B = \left[0\ 0\ \frac{K_L}{T_L}\right], G = [0\ 1\ 0], H = \left[\frac{t_h}{m}\ \frac{1}{m}\ 0\right], C = I_{3x3}, J = \left[0\ 0\ \frac{1}{m}\right], \tag{7a}$$

$$x = [\Delta d\ \Delta v\ a\ ], u = a_{des}, v = a_{lead}, w = r_{travel}, y = [\Delta d\ \Delta v\ a\ ] \tag{7b}$$

where x represents state variable of the control system which include Δd, Δv and a, u represents control input a_{des}, v represents disturbance of the system which is acceleration of lead vehicle a_{lead}, w represents unmeasured disturbance which is travel resistance r_{travel} and y represents output variable Δd, Δv and a.

B. Model predictive control

Most of the MPC algorithms are usually designed and implemented in the discrete-time domain. So, the state-space model in equation (6) needed to be transformed into discrete-time state space model, yielding

$$x(k+1) = Ax(k) + Bu(k) + Gv(k) + Hw(k). \tag{8a}$$

$$y(k) = Cx(k) + Jw(k). \tag{8b}$$

MPC solves an optimization problem – specifically, a quadratic program (QP) at each control interval. The solution determines the control input u to be used in the plant until the next control interval. This research uses MPC architecture from MATLAB's Model Predictive Control toolbox [40]. The control input u can be calculated by solving the constrained optimization problem below in every time step:

$$J(z_k) = \sum_{i=0}^{p-1} \left\{ \left[e_y^T(k+i) Q e_y(k+i) \right] + \left[e_u^T(k+i) R_u e_u(k+i) \right] + \left[\Delta u^T(k+i) R_{\Delta u} \Delta u(k+i) \right] \right\} + \rho_\varepsilon \varepsilon_k^2. \tag{9}$$

Subject to:

$$y_{min} \le y_j(k+i|k) \le y_{max} \tag{10a}$$

$$u_{min} \le u(k+i-1|k) \le u_{max} \tag{10b}$$

$$\Delta u_{min} \le \Delta u(k+i-1|k) \le \Delta u_{max} \tag{10c}$$

where z_k is QP decision of control input given by:

$$z_k^T = \left[u(k|k)^T\ u(k+1|k)^T\ \cdots\ u(k+p-1|k)^T\ \varepsilon_k\ \right]. \tag{11}$$

where p is the prediction horizon, k is the current control interval. Q, R_u, and $R_{\Delta u}$ are positive-semi definite weight matrices for the following error. e_y, e_u, and Δu is error tracking of output, control input, and control input rate, which can be described as:

$$e_y(k+i) = S_y^{-1}[r(k+i+1|k) - y(k+i+1|k)]. \quad (12a)$$

$$e_u(k+i) = S_u^{-1}[u_{target}(k+i|k) - u(k+i|k)]. \quad (12b)$$

$$\Delta u(k+i) = S_u^{-1}[u(k+i|k) - u(k+i-1|k)]. \quad (12c)$$

where S_y and S_u are the scaling factor in the engineering unit of output and input respectively, r is the reference value of plant output, y is plant output and u_{target} is the target value of control input. The term $\rho_\varepsilon \varepsilon_k^2$ in (9) is for constraint softening in case of an unavoidable constraint violation, where ρ_ε is constraint violation penalty weight and ε_k is a slack variable.

III. DEEP-LEARNING ARCHITECTURE

A. Long short-term memory (LSTM) layer

Long short-term memory networks (LSTMs) are a special kind of recurrent neural network (RNN) which is one of the deep-learning architectures. It was introduced in 1997 by Sepp Hochreiter and Jürgen Schmidhuber. It was designed to handle long-term dependencies between input and output.

The LSTM unit is composed of a memory cell illustrated in Figure. 1 which contains three gates – forget gate, input gate, and output gate that control the flow of information into or out of the cell.

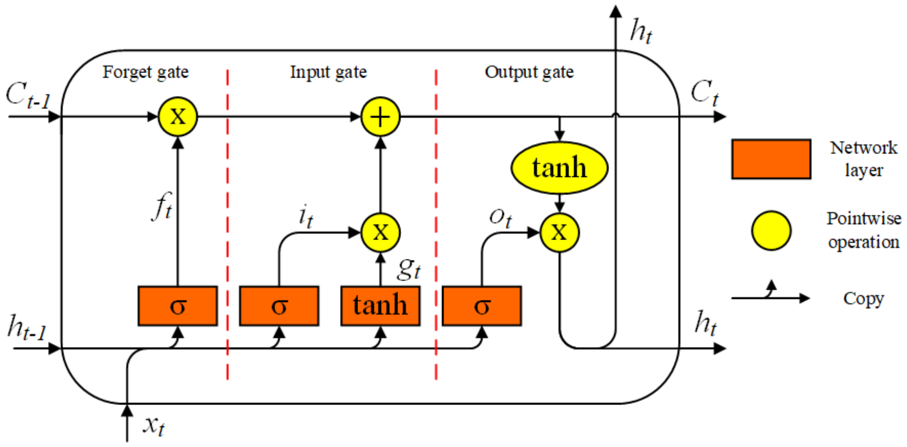


Figure 1: LSTMs cell adapted from [37]

The forget gate's activation, $f_t \in R^h$, determine how much information is allowed to pass through the next cell state C_t. Forget gate's activation can be described as:

$$f_t = \sigma(W_f x_t + R_f h_{t-1} + b_f). \quad (13)$$

where σ is the gate activation function of the sigmoid function, $x_t \in R^d$ is the cell input vector, h_{t-1} is the past hidden state, $W \in R^{h\times d}$ and $R \in R^{h\times h}$ are input and recurrent weights matrices respectively and $b \in R^h$ is the bias vector. The superscripts h and d refer to the number of hidden units and number of input features respectively.

The input gate's activation or update gate, $i_t \in R^h$, determine which new information will be updated with cell candidate $g_t \in R^h$. Input gate's activation and cell candidate can be described as:

$$i_t = \sigma(W_i x_t + R_i h_{t-1} + b_i). \quad (14a)$$

$$g_t = tanh \tanh(W_g x_t + R_g h_{t-1} + b_g). \quad (14b)$$

where $tanh$ is the state activation function of the hyperbolic tangent function.

The old cell state, C_{t-1}, can now be updated into a new cell state, $C_t \in R^h$, by using information from forget gate and input gate as shown in the equation below:

$$C_t = f_t C_{t-1} + i_t g_t. \quad (15)$$

The output gate's activation vector, $o_t \in R^h$, determines how much information will be passed to the next layer via the output vector, $h_t \in R^h$, also known as the hidden state. The output gate's activation and hidden state can be described as:

$$o_t = \sigma(W_o x_t + R_o h_{t-1} + b_o). \quad (16a)$$
$$h_t = tanh\ o_t \tanh(C_t). \quad (16b)$$

B. Regression layer

A regression layer computes the half-mean-squared error loss for regression tasks. For a single observation, the mean-squared-error is given by:

$$MSE = \sum_{i=1}^{R} \frac{(t_i - y_i)^2}{R}. \quad (17)$$

where R is the number of responses, t_i is the target output, and y_i is the network's prediction for response i.

For sequence-to-sequence regression networks, the loss function of the regression layer is the half-mean-squared-error of the predicted responses for each time step, not normalized by R:

$$loss = \frac{1}{2S} \sum_{i=1}^{S} \sum_{j=1}^{R} \left(t_{ij} - y_{ij}\right)^2. \quad (18)$$

where S is the sequence length.

IV. METHOD&SIMULATION

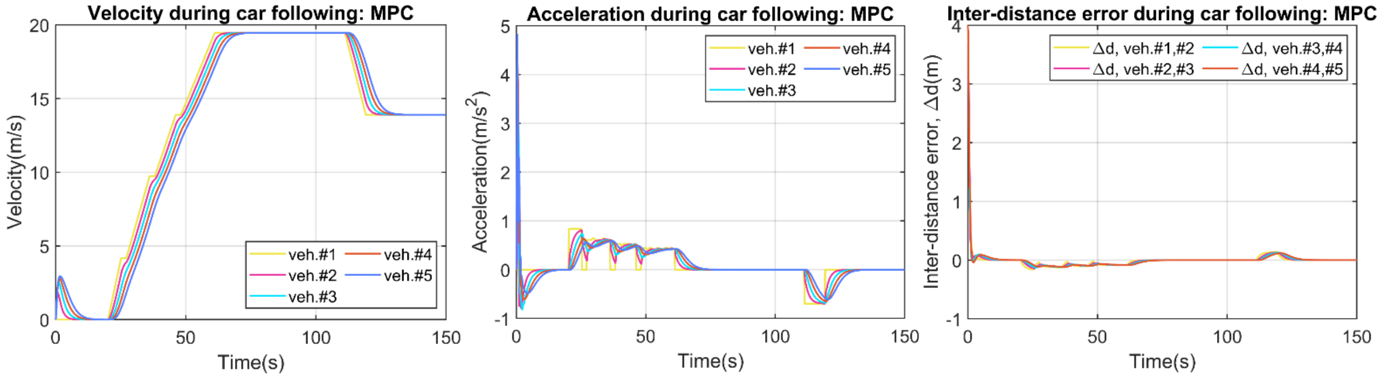
This section demonstrates the research approach and simulation results of the MPC and DMPC. We designed and simulated them in the MATLAB/Simulink environment. The simulation is performed with five vehicles. We implemented the vehicle's characteristics from ADVISOR, all vehicles are initially stopped and the initial inter-vehicle distance is 5 m. Then, the preceding vehicle accelerates following the driving cycle EUDC (extra-urban driving cycle).

Since deep learning is a learning-based architecture, training data is needed. In this study, we chose the MPC as a teaching model of the ACC system to create the training data for the deep learning model to learn. We first designed an MPC controller using the MPC controller block provided by the MPC toolbox. We used Equation (7) as the internal plant model of the MPC. Table 1 shows the parameters of the MPC and vehicle dynamics.

Table 1 The parameters of system model

Parameter	Value	Parameter	Value
$\boldsymbol{K_L}$	0.46	Q	[1 0.1 1]
$\boldsymbol{T_L}$	0.732	R_u	0
$\boldsymbol{t_h}$	1.35	$R_{\Delta u}$	0
$\boldsymbol{d_{min}}$	1	y_{min}	[-1 -5 -3]
$\boldsymbol{T_s}$	0.1	y_{max}	[1 5 a_{max}]
$\boldsymbol{p}$	10	u_{min}	-3
$\boldsymbol{c}$	1	u_{max}	a_{max}
$\boldsymbol{S_y}$	[2 10 8]	Δu_{min}	-1.5
$\boldsymbol{S_u}$	[8 8 1]	Δu_{max}	1.5

Using the above parameters, we designed and simulated the MPC-controlled ACC system. Figure 2 shows the results of the simulated MPC controller. The figures show the result during the first 150s of the EUDC. Each figure shows, from left to right, (a) the velocity of each vehicle, (b) the acceleration of each vehicle, and (c) the inter-vehicle following distance error. It is illustrated that the designed MPC controller is capable of controlling the ACC systems as the system output converges to zero and constraints are satisfied. Then, we collected the manipulated variable against the input disturbance and output variables and recorded it for the deep learning training.



(a) (b) (c)

Figure 2: The simulation results of MPC. (a) velocity of each vehicle, (b) acceleration of each vehicle and (c) inter-vehicle following distance error.

We designed the deep-learning model using a deep learning toolbox [41]. Figure 3 illustrates the architecture of DMPC, which we designed to correspond to the architecture of the MPC. The model has two inputs and one output – output variables, input disturbance, and manipulate variable respectively.

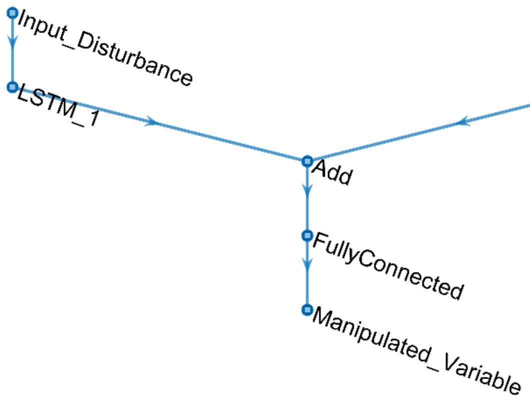
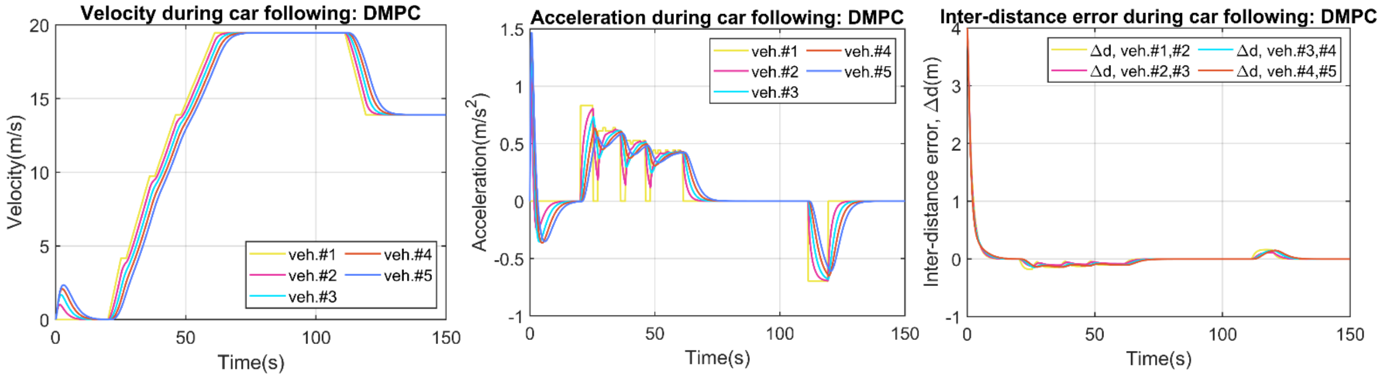


Figure 3: Network layers of deep model predictive control

Since the model has two inputs, *LayerGraph* is used to create the initial network. The input layers of the network are designed using the *feature input layer,* namely - input disturbance and output variable. The input disturbance has one feature and the output variable three features corresponding to ACC systems outputs – Δd, Δv, and a. Then, each input layer is connected with the *LSTM layer* with 200 hidden layers before summing into one element with the *addition layer*. Then, merge into a single output by the *fully connected layer*. Finally, the network output is connected with the *regression layer* named manipulated variable.

And finally, we trained the deep-learning model from the recorded data of the MPC, the driving cycle of WLTC-Class 3 (Worldwide harmonized Light-duty vehicles Test Cycles). We used an Adam optimizer with 20 epochs for the training. The RMSE and loss at the end of the iteration are 0.0123 and 0.0001, respectively.

Then, we deployed the DMPC controller for the ACC systems. Figure 4 illustrated the simulation results during the first 150s. Each figure shows, from left to right, (a) the velocity of each vehicle, (b) the acceleration of each vehicle, and (c) the inter-vehicle following distance error. It can be seen that the DMPC controller can successfully learn the necessary control action from the MPC. Furthermore, the vehicle that uses the DMPC controller has smoother acceleration, especially during the transition from the initial condition.



(a) (b) (c)

Figure 4: The simulation result of DMPC. (a) velocity of each vehicle, (b) acceleration of each vehicle and (c) inter-vehicle following distance error.

V. RESULT&DISCUSSION

This section demonstrates the performance of the DMPC controller compared with the PID-based controllers in the aspects of safety and comfort [36]. For the evaluation of the performance of the proposed controller, we simulated each controller with the driving cycle of the federal test procedure (FTP-75). Figure 5 and Figure 6 show the result in terms of safety and comfort index, and Table 2 shows the root means square (RMS) of the performance.

Figure 5: Comparison of safety index.

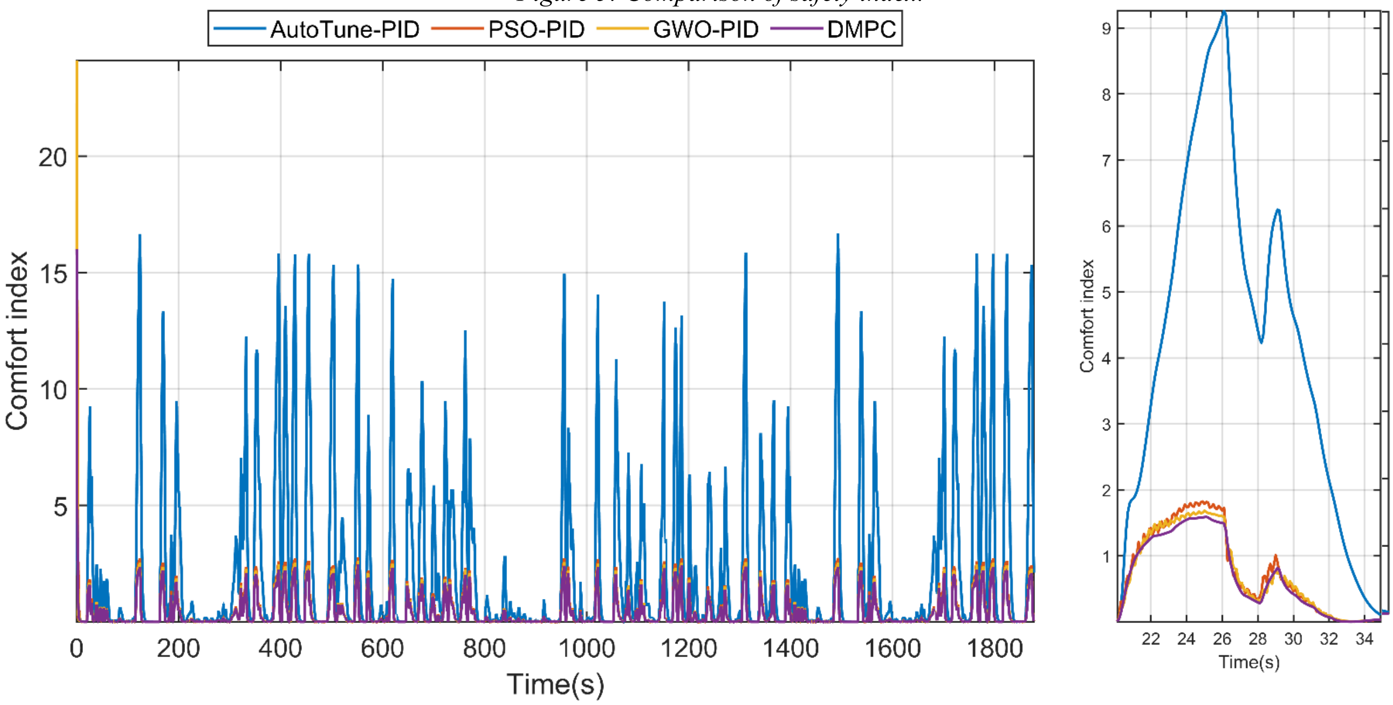


Figure 6: Comparison of comfort index.

Table 2: RMS of control performance.

Control scheme / Performance index	Autotune-PID	PSO-PID	GWO-PID	DMPC
Safety	0.0335	0.0035	0.0037	0.0009
Comfort	4.2689	0.9316	0.9025	0.7751

The safety index is a ratio between relative distance and safe distance. Hence, in Figure 5, the safety index is relatively high when the lead vehicle suddenly accelerates and causes a sudden change in the relative distance while the safe distance remains the same, especially when the lead vehicle accelerates from stationary.

The comfort index is a function of distance error, time-to-collision, and acceleration. Unlike the safety index, the starting position of the vehicles can cause the comfort index to be higher since the initial distance error is high. Furthermore, both acceleration and deceleration also affect the change in comfort index.

From the results, it shows that the overall performance of the proposed DMPC controller outperformed other controllers in safety and comfort whether during the acceleration or deceleration. The DMPC controller provides a smoother and more efficient control response, which results in better acceleration, velocity, and distance error. Hence, better safety and comfort.

Note that the above figures show that the performance of the PSO/GWO PID controller is close to the DMPC. However, the PID-based controller has poor performance during the transition from stationary in the initial condition, which causes the vehicle to accelerate at maximum and unstable acceleration, resulting in a higher comfort index.

Since the DMPC learned the control behavior from the MPC, it guaranteed that each control action on every sampling time is optimal. However, instead of optimizing each control action in real-time like MPC, DMPC selected the optimal response from the relation between input data. Hence, the DMPC provides the optimal control action without solving real-time optimization, which makes it more powerful than the MPC.

VI. CONCLUSION

In this paper, we designed the deep learning controller based on model predictive control to explore the capability of the deep learning controller in ACC systems. We trained the controller via supervised learning from generated data of MPC. The simulation shows that the proposed DMPC controller can successfully learn all of the control responses and behavior from the MPC. By this, the DMPC can provide the control behavior of the MPC but without the need to solve real-time optimization, which greatly reduces the computational load and provides smoother responses. To further investigate the control performance of the DMPC for ACC systems, we compared the DMPC to various types of optimized PID-based controllers. It shows that the DMPC controller yields better performance in both safety and comfort. It increases safety and comfort by 97% and 81% compared to autotune-PID and 75% and 14% compared to GWO-PID.

As for future tasks, the DMPC controller should achieve a higher level of automation by considering cooperation with a lateral control system and verifying the control performance with real-world experiments.

VII. REFERENCES

[1] "Road traffic injuries," World Health Organization, 7 February 2020. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries. [Accessed 11 May 2021].

[2] "Driving Statistics," Elegant E-Learning, Inc., [Online]. Available: https://driving-tests.org/driving-statistics/. [Accessed 11 May 2021].

[3] "Facts + Statistics: Highway safety," Insurance Information Institute, [Online]. Available: https://www.iii.org/fact-statistic/facts-statistics-highway-safety. [Accessed 11 May 2021].

[4] "Advanced driver-assistance systems," Wikimedia Foundation, Inc., 13 May 2021. [Online]. Available: https://en.wikipedia.org/wiki/Advanced_driver-assistance_systems. [Accessed 20 May 2021].

[5] B. Bayar, S. A. Sajadi-Alamdari, F. Viti and H. Voos, "Impact of Different Spacing Policies for Adaptive Cruise Control on Traffic and Energy Consumption of Electric Vehicles," in *24th Mediterranean Conference on Control and Automation (MED*, Athens, Greece, 2016.

[6] J.-J. Martinez and C. Canudas-de-Wit, "A Safe Longitudinal Control for Adaptive Cruise Control and Stop-and-Go Scenarios," *IEEE Transaction on Control Systems Technology,* vol. 15, no. 2, pp. 246-258, 2017.

[7] J. Wang and R. Rajamani, "The impact of adaptive cruise control systems on highway safety and traffic flow," *Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering,* vol. 218, no. 2, pp. 111-130, 2004.

[8] J. Zhou and H. Peng, "Range Policy of Adaptive Cruise Control Vehicles for Improved Flow Stability and String Stability," *IEEE Transactions on Intelligent Transportation Systems,* vol. 6, no. 2, pp. 229-237, 2005.

[9] J. Zhao, M. Oya and A. El Kamel, "A safety spacing policy and its impact on highway traffic flow," in *2009 IEEE Intelligent Vehicles Symposium*, Xi'an, China, 2009.

[10] C. Wu, Z. Xu and Y. Liu, "Spacing Policies for Adaptive Cruise Control: A Survey," *IEEE Access,* vol. 8, pp. 50149-50162, 2020.

[11] R. Rajamani, "Adaptive Cruise Control," in *Vehicle Dynamics and Control*, New york, Springer, 2012, pp. 141-170.

[12] B. Ganji, A. Z. Kouzani, S. Y. Khoo and M. Shams-Zahraei, "Adaptive cruise control of a HEV using sliding mode control," *Expert Systems with Applications,* vol. 41, no. 2, pp. 607-615, 2014.

[13] H. Balaska, S. Ladaci, H. Schulte and A. Djouambi, "Adaptive Cruise Control System for an Electric Vehicle Using a Fractional Order Model Reference Adaptive Strategy," *IFAC-PapersOnLine,* vol. 52, no. 13, pp. 194-199, 2019.

[14] F. Schrödel, P. Herrmann and N. Schwarz, "An Improves Multi-Object Adaptive Cruise Control Approach," *IFAC PapersOnLine,* vol. 52, no. 8, pp. 176-181, 2019.

[15] C.-C. Tsai, S.-M. Hsieh and C.-T. Chen, "Fuzzy Longitudinal Controller Design and Experimentation for Adaptive Cruise Control and Stop&Go," *Journal of Intelligent & Robotic Systems,* vol. 59, pp. 167-189, 2010.

[16] G. Prabhakar, S. Selvaperumal and P. Nedumal Pugazhenthi, "Fuzzy PD Plus I Control-based Adaptive Cruise Control System in Simulation and Real-time Environment," *IETE Journal of Research,* vol. 65, no. 1, 2019.

[17] S. Li, K. Li, R. Rajamani and J. Wang, "Model Predictive Multi-Objective Vehicular Adaptive Cruise Control," *IEEE Transactions on Control Systems Technology,* vol. 19, no. 3, p. 556–566, 2011.

[18] R. C. Zhao, P. K. Wong, Z. C. Xie and J. Zhao, "Real-time weighted multi-objective model predictive controller for adaptive cruise control systems," *International Journal of Automotive Technology,* vol. 18, pp. 279-292, 2017.

[19] D. Zhao, Z. Hu, Z. Xia, C. Alippi, Y. Zhu and D. Wang, "Full-range adaptive cruise control based on supervised adaptive dynamic programming," *Neurocomputing,* vol. 125, no. 11, pp. 57-67, 2014.

[20] Y.-C. Lin, H.-L. Thi Nguyen and C.-H. Wang, "Adaptive Neuro-Fuzzy Predictive Control for Design of Adaptive Cruise Control System," in *International Conference on Networking, Sensing and Control (ICNSC)*, Calabria, Italy, 2017.

[21] M. Wang, H. Yu, G. Dong and M. Huang, "Dual-Mode Adaptive Cruise Control Strategy Based on Model Predictive Control and Neural Network for Pure Electric Vehicles," in *International Conference on Transportation Information and Safety*, Liverpool, UK, 2019.

[22] Y. Dahmane, R. Abdrakhmanov and L. Adouane, "Stochastic MPC for Optimal Energy Management Strategy of Hybrid Vehicle Performing ACC with Stop&Go Maneuvers," *IFAC-PapersOnLine,* vol. 51, no. 9, pp. 223-229, 2018.

[23] A. Weißmann, D. Görges and X. Lin, "Energy-optimal adaptive cruise control combining model predictive control and dynamic programming," *Control Engineering Practice,* vol. 72, pp. 125-137, 2018.

[24] S. Magdici and M. Althoff, "Adaptive Cruise Control with Safety Guarantees for Autonomous Vehicles," *IFAC-PapersOnLine,* vol. 50, no. 1, pp. 5774-5781, 2017.

[25] L.-h. Luo, H. Liu, P. Li and H. Wang, "Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel-economy, safety and car-following," *Journal of Zhejiang University SCIENCE A,* vol. 11, pp. 191-201, 2010.

[26] L. Guo, P. Ge, Y. Qiao and L. Xu, "Multi-Objective Adaptive Cruise Control Strategy Based on Variable Time Headway," *Intelligent Vehicles Symposium (IV),* pp. 203-208, 2018.

[27] S. Cheng and L. Li, "Multiple-Objective Adaptive Cruise Control System Integrated With DYC," *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY,* vol. 68, no. 5, pp. 4550-4559, 2019.

[28] Y. He, B. Ciuffo, Q. Zhou, M. Makridis, K. Mattas, J. Li, Z. Li, F. Yan and H. Xu, "Adaptive Cruise Control Strategies Implemented on Experimental Vehicles: A Review," *IFAC PapersOnLine,* vol. 52, no. 5, p. 21–27, 2019.

[29] T. Takahama and D. Akasaka, "Model Predictive Control Approach to Design Practical Adaptive Cruise Control for Traffic Jam," *International Journal of Automotive Engineering,* vol. 9, no. 3, pp. 99-104, 2018.

[30] "Deep learning," Wikimedia Foundation, Inc., 18 May 2021. [Online]. Available: https://en.wikipedia.org/wiki/Deep_learning. [Accessed 20 May 2021].

[31] L. Sun, B. Zou, S. Fu, J. Chen and F. Wang, "Speech emotion recognition based on DNN-decision tree SVM model," *Speech Communication,* vol. 115, pp. 29-37, 2019.

[32] C. Boufenar, A. Kerboua and M. Batouche, "Investigation on deep learning for off-line handwritten Arabic character recognition," *Cognitive Systems Research,* vol. 50, pp. 180-195, 2018.

[33] Y. Zhao, J. Li and L. Yu, "A deep learning ensemble approach for crude oil price forecasting," *Energy Economics,* vol. 66, pp. 9-16, 2017.

[34] D. Zhao, B. Wang and D. Liu, "A supervised Actor–Critic approach for adaptive cruise control," *Soft Comput,* vol. 17, pp. 2089-2099, 2013.

[35] X. Chen, Y. Zhai, C. Lu, J. Gong and G. Wang, "A Learning Model for Personalized Adaptive Cruise Control," in *IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, USA, 2017.

[36] D. Zhao, Z. Xia and Q. Zhang, "Model-Free Optimal Control Based Intelligent Cruise Control with Hardware-in-the-Loop Demonstration," *IEEE Computational Intelligence Magazine,* vol. 12, no. 2, pp. 56-69, 2017.

[37] S. S. P. Kumar, A. Tulsyan, B. Gopaluni and P. Loewen, "A Deep Learning Architecture for Predictive Control," *IFAC PapersOnLine,* vol. 51, no. 18, pp. 215-517, 2018.

[38] T. Baumeister, S. L. Brunton and J. N. Kutz, "Deep learning and model predictive control for self-tuning mode-locked lasers," *Journal of the Optical Society of America B,* vol. 35, no. 3, pp. 617-626, 2018.

[39] F. Mohseni, S. Voronov and E. Frisk, "Deep Learning Model Predictive Control for Autonomous Driving in Unknown Environment," *IFAC PapersOnLine,* vol. 51, no. 22, pp. 447-452, 2018.

[40] A. Bemporad, N. L. Ricker and M. Morari, "Model Predictive Control Toolbox™ User's Guide," The MathWorks, Inc., [Online]. Available: https://www.mathworks.com/help/pdf_doc/mpc/mpc_ug.pdf.

[41] M. H. Beale, M. T. Hagan and H. B. Demuth, "Deep Learning Toolbox™ User's Guide," [Online]. Available: https://www.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf.