Data-Driven Adaptive Cruise Control for Trucks with Lane-Keeping Model

Shuyou Yu, *Senior Member, IEEE* , Junting Liu, Shuo Wang, Huan Chang, Zhuqing Shi, Hong Chen, *Fellow, IEEE*

***Abstract*—This paper proposes a data-driven adaptive cruise control (ACC) strategy for trucks, incorporating the lane-keeping assistance. It addresses both the maintainace of a safe longitudinal distance and lateral tracking of the road centerline. To handle system constraints, model predictive control (MPC) is employed. The nonlinear characteristics of trucks are approximated using Koopman operator theory and deep neural network (DNN) techniques, yielding a global linear predictive model. Furthermore, a comparative analysis between first-principle modeling and data-driven modeling methods is conducted using TruckSim and Simulink. The results demonstrate the proposed ACC strategy's effectiveness in reducing computational time while ensuring longitudinal safety and lateral lane-keeping capabilities.**

***Index Terms*—Adaptive cruise control, model predictive control, Koopman operator, deep neural networks.**

I. Introduction

THE progress of intelligent technologies significantly enhances vehicle safety and reduces drivers' workload [1]. Advanced Driver Assistance Systems, including Adaptive Cruise Control (ACC), Lane Keeping Assist, and Automatic Emergency Braking, are key components of intelligent automation systems [2]–[4]. In the late 1970s, ACC system emerges by integrating a Safe Distance Keeping System into the traditional fixed-speed cruise control system [5]. ACC strategy utilizes onboard radar sensors to measure the relative distance and speed of the leading vehicle, actively adjusts the vehicle's speed through throttle and braking mechanisms to guarantee safety. To deal with complex scenarios, a target selection strategy is proposed to identify the critical leading vehicle and prevent collision [6]. Additionally, to calculate the safe following distance under varying vehicle accelerations, various models are presented [7], describing vehicle acceleration and deceleration processes.

This work was supported by the National Natural Science Foundation of China (No. 62473167) and the Natural Science Foundation of Jilin Province (No. 20240402079GH). (Corresponding author: Shuo Wang, e-mail: shuow23@mails.jlu.edu.cn)

Shuyou Yu is with the State Key Laboratory of Chassis Integration and Bionics and the Department of Control Science and Engineering, Jilin University, Changchun, Jilin, China. (e-mail: shuyou@jlu.edu.cn)

Shuo Wang is with the Department of Control Science and Engineering, Jilin University, Changchun, Jilin, China. (e-mail: shuow23@mails.jlu.edu.cn)

Junting Liu is with CRRC Technology Innovation Co., LTD, Beijing, China. (e-mail: 787853223@qq.com)

Huan Chang is with Beijing Jingwei Hirain Technology Co., LTD Beijing, China. (e-mail: huanchang19@mails.jlu.edu.cn)

Zhuqing Shi is with Beijing Automotive Technology Center Co.,Ltd., Intelligent and Connected Vehicles Center, Beijing,China. (e-mail: shizhuqing@baicgroup.com.cn)

Hong Chen is with the College of Electronics and Information Engineering, Tongji University, Shanghai, China. (e-mail: chenhong2019@tongji.edu.cn)

To ensure the effectiveness of ACC in turning scenarios, a radar is used to measure the relative speed to the leading vehicle [8], and a vehicle-rode relative model is designed [9], where sensors are adopted to capture road curvature. A road-based predictive speed planner is designed in [10], using a fuzzy logic and a decoupled control strategy, to ensure vehicles navigate curved roads at the expected speed. The implementation of an ACC system enhances vehicle safety and reduces the workload of drivers. In [11], a target vehicle selection algorithm is developed based on the prediction of the lane-changing intention for the leading vehicle to improve the safety of the target vehicle. When a truck is fully loaded, its center of mass is elevated, increasing the possibility of rollover [12]. The driver's position further increases the difficulty in detecting proximity to the rollover threshold [13]. Therefore, ensuring the lateral stability of fully loaded and high speed trucks is crucial for reliable steering. A Kalman filter is designed in [14] to estimate the tire roll angle online, and a rollover index is defined to indicate the rollover possibility. In [15], A sliding mode controller is integrated into the anti-lock braking system to improve the lateral stability of trucks. To address the H_∞ controller design problem for ACC systems with unknown system dynamics, an online data-driven learning technique is proposed [16].

Model predictive control (MPC) demonstrates its superiority in adjusting potential conflicts between longitudinal and lateral control, leading to the development of vehicle control strategies [17], [18]. In [19], a speed-predicting strategy for the leading vehicle is proposed, which can minimize energy consumption through speed trajectory optimization. A nonlinear model predictive controller, designed in [20], improves energy utilization efficiency by planning the vehicle's speed trajectory in real-time, incorporating look-ahead traffic information and road conditions. Furthermore, a hierarchical neural networks structure is established to estimate the vehicle's deceleration by integrating road and weather conditions [21]. Based on the identified driver driving style, a predictive controller is designed to improve driving comfort and vehicle tracking performance [22]. In [23], a model predictive controller is designed to guarantee driving comfort and minimize fuel consumption by imposing constraints on the vehicle's acceleration and its rate of change. Additionally, driving safety is guaranteed by maintaining a minimum safety distance from the leading vehicle. A hierarchical structure based on cloud control systems is presented to solve the challenges of real-time computation and apply on vehicle-cloud [24]. In addition, based on the inertial triggering mechanism and

multi-objective optimization, an ACC system with universal hardware configuration is proposed [25].

Koopman operator theory provides a framework for transforming nonlinear dynamical systems into an infinite-dimensional linear representation [26]. Conventional linearization techniques rely on local Taylor expansions around fixed operating points, the accuracy of these methods rapidly degrades when the system deviates from these operating points. In contrast, Koopman operator theory enables a global linearization of the underlying nonlinear dynamics by constructing a data-driven predictor that remains valid across the full operational envelope. Moreover, whereas first-principles vehicle models require precise calibration of parameters (e.g., tire stiffness, aerodynamic drag, inertial properties), the Koopman operator theory can directly construct a linear model from the data and effectively capture the nonlinear dynamics. Through the construction of suitable state lifting functions, a nonlinear system is mapped into an infinite-dimensional linear space, without loss of any information of the original system. However, implementing an infinite-dimensional state space is infeasible in practice. To address this issue, the extended dynamic mode decomposition (EDMD) method is proposed [27], approximating the infinite-dimensional Koopman operator in finite dimensions. In [28], the EDMD method is applied to controlled systems, which increases the scope of application of Koopman operator theory.

Koopman operator theory has been developed in vehicle dynamics and advanced driver assistance systems. Classical EDMD is utilized to model a full-vehicle lateral dynamics system incorporating sideslip, and an optimal controller is designed [29]. To enhance robustness under model uncertainties, stochastic model predictive control schemes are formulated based on Koopman-predicted model [30]. A modified resolvent-type method, a Koopman-based data-driven technique, is developed for system identification of ACC [31]. Results show the new Koopman-resolvent method yields very accurate ACC models even with low sampling rates and noise. An MPC torque vectoring strategy is developed based on a finite-dimensional Koopman model of the 3-DOF vehicle obtained via EDMD [32]. Compared to a linear time-varying (LTV) MPC, the method achieves better handling performance. However, the aforementioned studies handle lateral and longitudinal control separately, which do not take the influence of lateral dynamics on longitudinal dynamics into account.

Traditional EDMD methods rely on manual selection of lifting functions, which limits their flexibility. To address this limitation, the deep neural network (DNN)-based Koopman methods are proposed [33]. This technology has been applied in diverse domains, including fluid mechanics, robotics, and power systems [34], [35], which highlights its generality as a data-driven modeling method.

In [33], neural networks, capable of fitting arbitrary nonlinear functions, are employed to improve the lifting function. Furthermore, an auxiliary control network is designed to elevate prediction and control performance [36]. In [37], the DNN-based Koopman method is employed to derive a linear model of the heavy unmanned tracked vehicle, upon which an MPC strategy for trajectory tracking is formulated. The extended state observer is utilized to estimate the total disturbances in the lifted space online, which effectively compensates the modeling errors and residuals [38]. In [39], a deep learning framework based on the Koopman operator is presented for establishing a global linear model of vehicle dynamics. Furthermore, a linear MPC strategy is presented to track the time-varying reference. These works focus on passenger vehicles or low-speed vehicles, which exhibit relatively mild longitudinal–lateral coupling dynamics. In contrast, trucks exhibit strong longitudinal-lateral coupling and strong nonlinear dynamics. This phenomenon is further amplified under high-speed driving conditions.

This paper proposes a data-driven ACC method for trucks with the lane-keeping assistance. The key contributions of this paper are summarized as follows:

1) A data-driven ACC strategy for trucks is proposed. Based on Koopman operator theory, a global linear model of trucks is established, preserving the nonlinear characteristics of truck systems. This model is further integrated with a lane-keeping model to design a unified controller that ensures both longitudinal safety and lateral lane-keeping performance.

2) Based on the global linearization model, a model predictive controller is designed to avoid solving non-convex optimization problems which can significantly reduce the computational time.

3) Scenarios of high-speed turning on highways for fully loaded trucks are taken into account. Furthermore, the influence of the nonlinear dynamics of the tires on trucks safety is considered. Comparative simulations conducted in TruckSim and Simulink illustrate the effectiveness of the proposed method.

The paper is organized as follows: Section II introduces Koopman operator theory, model validation and control objective of ACC. Section III proposes a DNN-ACC algorithm based on high-dimensional linear model. The first-principle models are established and the related MPC referred to as 5DOF-ACC and LTV-ACC are proposed in Section IV. Section V conducts the comparative experiments. Section VI concludes this paper.

II. Problem Setup and Preliminaries

Since the highly nonlinear characteristics of truck systems, constructing a model that accurately capture the vehicle's dynamic behavior is challenging in general. Furthermore, as the number of degrees of freedom considered increases, so does the complexity of the model. To address these issues, a data-driven approach is adopted to obtain a "global" linear model that describes the dynamic behavior of trucks.

A. Koopman Operator Theory

1) Koopman operator for autonomous systems: Review the Koopman operator method used for analyzing autonomous systems:

$$x_{k+1} = h(x_k) \tag{1}$$

The Koopman operator $\mathcal{K}$: $\mathcal{F} \to \mathcal{F}$ acts on observable function $\phi : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$(\mathcal{K}\phi)(x_k) = \phi(h(x_k)) \tag{2}$$

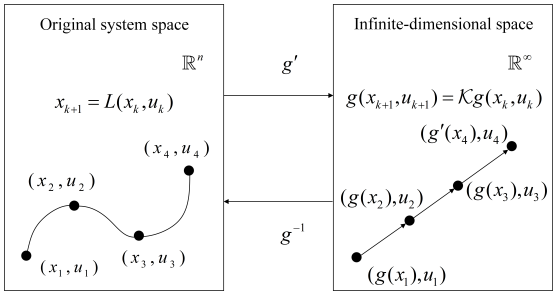

Fig. 1. System evolution process.

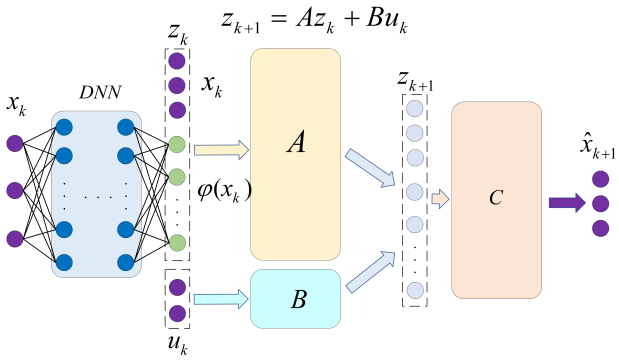

Fig. 2. DNN structure diagram.

Importantly, the Koopman operator provides a linear representation of the dynamics, despite the possible nonlinearity of the original system, with the trade-off that the operator is generally infinite-dimensional.

2) Koopman operator for controlled systems : The discrete time nonlinear system is described as follows:

$$x_{k+1} = L(x_k, u_k) \tag{3}$$

where $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ denote the system state and control input at the time instant k, respectively.

Based on [28], the controlled system is viewed as an autonomous system on the extended state space $\mathbb{R}^n \times \jmath(\mathcal{U})$, where $\jmath(\mathcal{U})$ denotes the space of all infinite control sequences $(u_k)_{k=0}^{\infty}$ with $u_k \in \mathcal{U}$. Writing the extended state as

$$\mathcal{X} = \begin{bmatrix} x \\ \boldsymbol{u} \end{bmatrix} \in \mathbb{R}^n \times \jmath(\mathcal{U}) \tag{4}$$

and

$$\mathcal{X}_{k+1} = F(\mathcal{X}_k) := \begin{bmatrix} L(x_k, u_0) \\ \mathcal{P}\boldsymbol{u} \end{bmatrix} \tag{5}$$

where $\mathcal{P}$ is the left shift operator, i.e. $\mathcal{P}\boldsymbol{u}_k = u_{k+1}$.

The Koopman operator $\mathcal{K}$: $\mathcal{H} \to \mathcal{H}$ acts on observable function $g : \mathbb{R}^n \times \jmath(\mathcal{U}) \to \mathbb{R}$ is defined by

$$(\mathcal{K}g)(\mathcal{X}_k) = g(\mathcal{X}_{k+1}) = g(F(\mathcal{X}_k)) \tag{6}$$

The Koopman operator (6) is a linear operator fully describing the nonlinear dynamical system (1) provided that $\mathcal{H}$ contains the components of the non-extended state $x_i, i = 1, \cdots, n$.

The linear evolution process of system (3) from the original space $\mathbb{R}^n$ to the Koopman infinite-dimensional space $\mathbb{R}^\infty$ is illustrated in Fig. 1. The evolution of the infinite-dimensional state is expressed as:

$$g(x_{k+1}, u_{k+1}) = \mathcal{K}g(x_k, u_k) \tag{7}$$

where $\mathcal{K}$ is the Koopman operator.

In practice, implementing an infinite-dimensional $\mathcal{K}$ is infeasible. Therefore, a finite-dimensional approximation can be adopted [28].

Assumption 1: The discrete time nonlinear system (3) is lifted into a finite-dimensional state space, i.e., a finite-dimensional approximation matrix K exists such that:

$$\Phi(x_{k+1}, u_{k+1}) = K\Phi(x_k, u_k) \tag{8}$$

where $\Phi(x_k, u_k) \in \mathbb{R}^{N_\Phi}$ is the lifting state and N_Φ denote the dimension of the high-dimensional state space, which is specified as 30.

According to (4), $\Phi(x_k, u_k)$ can be separated into two parts in terms of states and control inputs:

$$\Phi(x_k, u_k) = \begin{bmatrix} \Phi_x(x_k)^{\mathrm{T}} & u_k{}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{9}$$

Deep Neural Networks

DNN technology improve the accuracy of model prediction and avoid to select manually the lifting function [39]. The structure of DNN is illustrated in Fig. 2.

To facilitate the design of a linear controller, the original state is added to the high-dimensional state, thereby constructing a new state $z_k \in \mathbb{R}^{N_\Phi+n}$:

$$z_k = \begin{bmatrix} x_k{}^{\mathrm{T}} & \varphi(x_k)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{10}$$

where z_k is the approximation of the function $\Phi_x(x_k)$. The term $\varphi(x_k)$ is the output of the neural networks.

According to (8), the evolution of finite dimension space is expressed as:

$$\begin{bmatrix} z_{k+1} \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} A & B \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} z_k \\ u_k \end{bmatrix} \tag{11}$$

where $A \in \mathbb{R}^{(N_\varphi+n)\times(N_\varphi+n)}$, $B \in \mathbb{R}^{(N_\varphi+n)\times m}$, $K_{21} \in \mathbb{R}^{m\times(N_\varphi+n)}$ and $K_{22} \in \mathbb{R}^{m\times m}$. Note that the control input u_k is consistent in the original state space.

Rewriting (11) as

$$z_{k+1} = Az_k + Bu_k \tag{12a}$$

$$\hat{x}_k = Cz_k \tag{12b}$$

where $C = \begin{bmatrix} I^{n\times n} & 0^{n\times N_\varphi} \end{bmatrix}$ represents the state extraction matrix and $\hat{x}_k$ is the estimate of the original state x_k, used to verify the accuracy of the prediction.

Note that the Koopman operator approach based on DNN follows prior work [34], [35] and is not presented as a methodological novelty. This paper focuses on demonstrating the applicability of the proposed method to truck dynamics and on integrating it with a lane-keeping model to implement adaptive cruise control.

Remark 1. In this work, the original state is defined as $x_k = [v_x, v_y, \omega]^{\mathrm{T}}$, where v_x, v_y, and ω denote the longitudinal velocity, lateral velocity, and yaw rate, respectively. The

Koopman lifting process are performed only on this three-dimensional truck dynamics state [38], [39]. Tracking errors such as longitudinal position error e_x, lateral position error e_y, and heading error e_a are road-relative geometric quantities that are directly computed from the truck attitude and the known road information.

Remark 2. The complexity of the nonlinear system determines the number of hidden layers. To achieve a balance between the fitting accuracy and the network complexity, four hidden layers are used in Fig.2. The observation function is obtained through a fully connected DNN, completing the mapping from a low-dimensional to a high-dimensional space. This allows the state x_k of the nonlinear system to be transformed into the high-dimensional state variables z_k. Predicting the state z_{k+1} of high-dimensional linear systems through two hidden layers A and B without activation function and bias [40]. The estimated future state of the nonlinear system $\hat{x}_{k+1}$ is obtained by state extraction matrix C.

To sure that the linear model accurately approximates the truck's nonlinear dynamic model, a p-step prediction equation is formulated:

$$\begin{aligned} z_{k+p} =& Az_{k+p-1} + Bu_{k+p-1} \\ =& A(Az_{k+p-2} + Bu_{k+p-2}) \\ &+ Bu_{k+p-1} \\ &\vdots \\ =& A^p z_k + \sum_{i=1}^{p} A^{i-1} B u_{k+p-i} \end{aligned} \tag{13}$$

To demonstrate the precision of the fitting model in the high-dimensional state space, the following error equation is proposed:

$$L_{zp} = \frac{1}{p} \sum_{i=1}^{p} \left\| z_{k+i} - \left[{x_{k+i}}^{\mathrm{T}} \varphi(x_{k+i})^{\mathrm{T}} \right]^{\mathrm{T}} \right\|_2^2 \tag{14}$$

where x_{k+i} represents the actual value at time instant $k+i$, and z_{k+i} is a high-dimensional space state.

Furthermore, an average prediction error L_{xp} is defined, to illustrate the fitting accuracy between the estimated states $\hat{x}_{k+i}$ and the true state x_{k+i}:

$$L_{xp} = \frac{1}{p} \sum_{i=1}^{p} \left\| x_{k+i} - \hat{x}_{k+i} \right\|_2^2 \tag{15}$$

where $\hat{x}_{k+i} = C z_{k+i}$.

Based on (14) and (15), the overall training loss for the neural network is:

$$Loss = \alpha_1 L_{xp} + \alpha_2 L_{zp} + \alpha_3 \left\| \xi \right\|_2^2 \tag{16}$$

where $\left\| \xi \right\|_2^2$ serves as a the regularization term to prevent overfitting in neural networks, where $\alpha_1 > 0$, $\alpha_2 > 0$, and $\alpha_3 > 0$ are the weights.

B. Data Sets

The data are collected from TruckSim, using a sampling model of a 2-axle LCF Van with a total mass of 18,000 kg [41]. The road adhesion coefficient $\mu = 0.85$, with a sampling time of $T_s = 10$ ms. The longitudinal force F_x and the steering angle of the front tires δ have range of $[-6000, 6000]$ N and $[-10, 10]$ deg, respectively. The longitudinal velocity v_x, lateral velocity v_y, and yaw rate ω have ranges of $[16, 25]$ m/s, $[-0.6, 0.6]$ m/s, $[-6, 6]$ deg /s, respectively. A total of 70 data sets are collected in the truck, where each data set contains 6200 time steps.

C. Model Validation

To evaluate the predictive accuracy, the root mean square error (RMSE) is employed as the primary evaluation metric, expressed as:

$$\mathrm{RMSE} = \frac{\sqrt{\sum_k \left\| x_{pred}(k) - x_{true}(k) \right\|_2^2}}{\sqrt{\sum_k \left\| x_{true}(k) \right\|_2^2}} \times 100\% \tag{17}$$

where $x_{true}(k)$ and $x_{pred}(k)$ represent the true values of the actual truck states and the predicted values of the linear model at time instant k, respectively.

The available data sets are divided into three subsets: 60 sets for training, 5 for validation, and 5 for testing. The learning rate, which significantly influences the convergence of the DNN, is empirically set to 10^{-3}. According to the trial, the learning rate is set to 10^{-3}, and the weights in (16) are chosen as $\alpha_1 = 0.1$, $\alpha_2 = 1$ and $\alpha_3 = 10^{-6}$.

To evaluate the accuracy of the Koopman model approximated by the DNN, two test scenarios are designed. For each scenario, the dimension of the lifting function is set to 10, 30, 50, and 100, respectively. The specific test scenarios are set as follows:

Scenario 1: The deceleration scenario is constructed, with the initial state set to $x_0 = [25 \ \ 0 \ \ 0]^{\mathrm{T}}$ and the control inputs specified as $F_x = -1000$ N and $\delta = 0$. In Fig. 3 , the predicted states of the Koopman model in different dimensions are presented and compared with true data of trucks. The green, red, blue, and orange dotted lines represent the predicted states of Koopman linear model in 10 dimensions, 30 dimensions, 50 dimensions, and 100 dimensions, respectively, while the black solid lines indicate the actual states of the truck. The RMSE values under the deceleration scenario are shown in Table I.

TABLE I
RMSE VALUES FOR SCENARIO 1

Time step	10	80	200	400
Koopman-10	0.15%	0.35%	0.52%	0.66%
Koopman-30	0.06%	0.13%	0.26%	0.39%
Koopman-50	0.08%	0.15%	0.20%	0.18%
Koopman-100	0.17%	0.30%	0.44%	0.60%

From the Fig. 3 and Table III, the model achieves higher fitting accuracy when the dimensionality is set to 30 or 50.

Scenario 2: The lane-changing scenario is constructed, with the initial state set to $x_0 = [25 \ \ 0 \ \ 0]^{\mathrm{T}}$ and the control

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2026.3664228

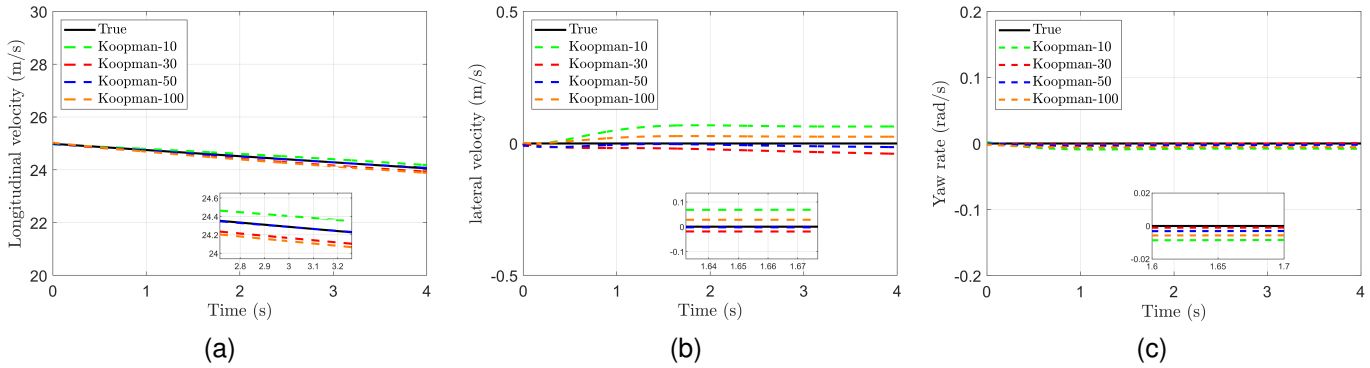

Fig. 3. (a) Longitudinal velocity (b) Lateral velocity (c) Yaw rate.

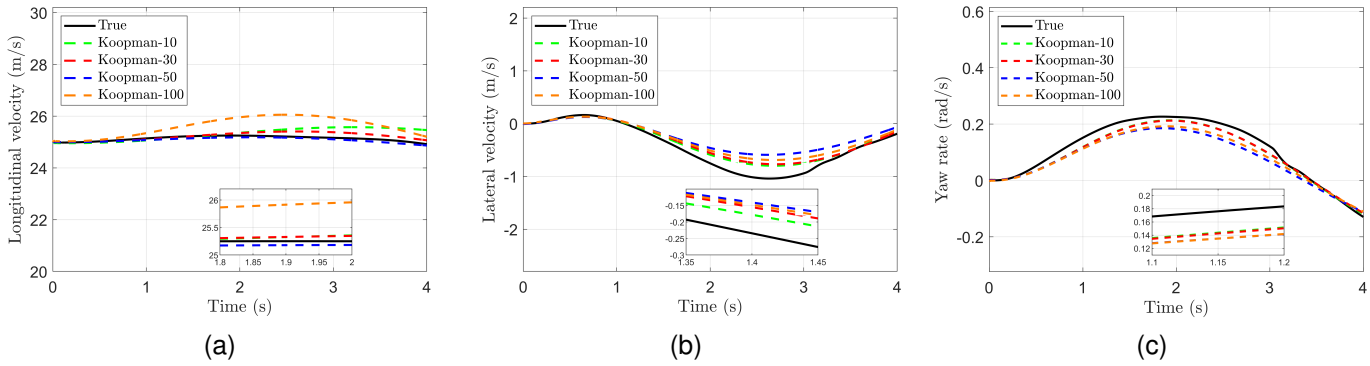

Fig. 4. (a) Longitudinal velocity (b) Lateral velocity (c) Yaw rate.

inputs specified as $F_x = 5000\sin(t)$+1000 N and $\delta = 5\sin(t)$ rad [42]. In Fig. 4, the predicted states of the Koopman model in different dimensions are presented and compared with true data of trucks. The green, red, blue, and orange dotted lines represent the predicted states of Koopman linear model in 10 dimensions, 30 dimensions, 50 dimensions, and 100 dimensions, respectively, while the black solid lines indicate the actual states of the truck. The RMSE values under the lane-changing scenario are shown in Table II.

TABLE II
RMSE VALUES FOR SCENARIO 2

Time step	10	80	200	400
Koopman-10	0.18%	1.01%	1.45%	1.75%
Koopman-30	0.11%	1.00%	1.50%	1.51%
Koopman-50	0.05%	1.04%	2.14%	2.58%
Koopman-100	0.19%	1.11%	2.43 %	3.06%

From the Fig. 4 and Table II, the 50-dimensional model achieves the best performance in short time steps. In contrast, both the 10-dimensional and 30-dimensional models maintain relatively lower RMSE values within 400 time steps. As the complexity of the neural network increases, the prediction accuracy of the 100-dimensional model decreases.

The Koopman linear models with different dimensions are all capable of approximating the actual truck states, thereby demonstrating the effectiveness of the Koopman operator theory for predicting the states of nonlinear systems. In summary, the 30-dimensional Koopman linear model demonstrates consistently high fitting accuracy and is therefore adopted as the prediction model in this paper.

Remark 3. In this paper, a prediction horizon of 10 steps with a sampling time of 0.01 s is employed, yielding a total prediction time of 0.1 s. The 4 s validation window, being longer than the 0.1 s prediction time, is sufficient to assess the model's performance.

D. Lane-Keeping Model

In this subsection, the lane-keeping model is introduced to describe the lateral distance and yaw angle deviation between the truck and the road centerline.

Fig. 5 illustrates the geometric representation of the vehicle-road relationship, where D_L represents the look-ahead distance for detecting the road information, the term e_y is the lateral error between the current truck position and the center position of the road at D_L ahead, and e_a is the heading angle error between the tangent to the road and the truck orientation. Additionally, the term $K_L = 1/R_{road}$ denotes the road curvature, where R_{road} is the road radius.

The vehicle-road relative position model is described as follows [43]–[45]:

$$\dot{e}_y = v_x e_a - v_y - \omega D_L \tag{18a}$$
$$\dot{e}_a = v_x K_L - \omega \tag{18b}$$

where D_L is adjusted to guarantee the safety of the truck on curved road. Suppose that the road curvature is known and time-varying. This curvature information can be obtained in real time or in advance through high-definition maps, cameras, or LiDAR.

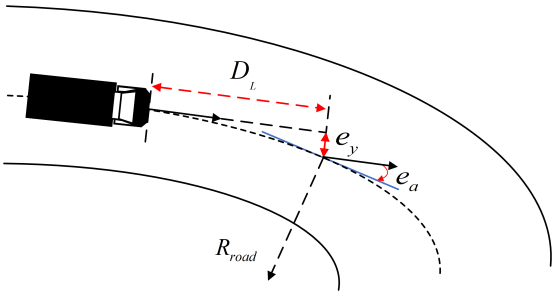

Fig. 5. Lane-keeping model.

Fig. 6 illustrates the relative longitudinal position between the following truck and leading truck. Denote d_{ref} as the desired inter-truck distance, l_{Truck} the truck's length, x_h and x_f the longitudinal position of the following truck and the leading truck, respectively. The longitudinal position error between the two trucks is denoted as:

$$e_x = x_f - x_h - d_{ref} - l_{Truck} \tag{19}$$

The time derivative of the longitudinal position error is:

$$\dot{e}_x = v_f - v_x \tag{20}$$

where v_x and v_f are the longitudinal velocity of the following truck and leading truck, respectively.

E. Control Objective of Adaptive Cruise Control Integrated the Lane-Keeping Model

In this subsection, the control objective of ACC is proposed including the longitudinal and lateral control objective.

1) The longitudinal control objective of trucks is to track the longitudinal velocity of the leading truck, and to ensure that the longitudinal position error is within a desired range, i.e.,

$$\begin{cases} \text{minimize} \left\| v_f - v_x \right\|_2^2 \\ \text{minimize} \left\| e_x - 0 \right\|_2^2 \end{cases} \tag{21}$$

where $\|\vartheta\|_2$ represents the L_2-Norm of function ϑ, i.e., $\|\vartheta(k)\|_2 := \sqrt{\sum_{k=0}^{\infty}(\vartheta(k))^2} < \infty$, and $\lim_{k\to\infty} \vartheta(k) = 0$.

In order to avoid collision of trucks, a collision avoidance constraint is presented

$$e_{xmin} \le e_x \le e_{xmax} \tag{22}$$

where e_{xmin} and e_{xmax} represent the minimum and maximum longitudinal position error.

2) The lateral control objective of trucks is to travel as close as possible to the centerline of the road:

$$\begin{cases} \text{minimize} \left\| e_y - 0 \right\|_2^2 \\ \text{minimize} \left\| e_a - 0 \right\|_2^2 \end{cases} \tag{23}$$

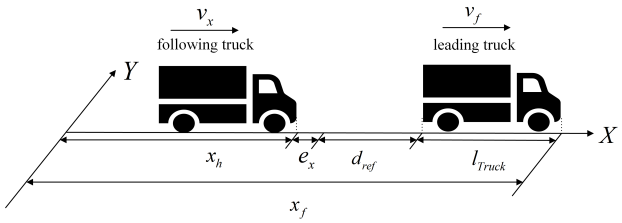

Fig. 6. Relative position of the following truck and leading truck.

To ensure the truck travels along the road centerline, the following lateral safety constraints are expressed as:

$$\begin{cases} e_{ymin} \le e_y \le e_{ymax} \\ e_{amin} \le e_a \le e_{amax} \end{cases} \tag{24}$$

where e_{amin} and e_{amax} represent the minimum and maximum of the heading errors. The terms of e_{ymin} and e_{ymax} are the minimum and maximum lateral position errors.

III. Predictive Cruise Control with Linear Dynamics Model

In this section, the proposed data-driven ACC method, referred to as the DNN-ACC method, is developed based on linear MPC.

A. Predictive Controller Based on Linear Model

By combining (12), (18), and (20), the model used in the DNN-ACC is:

$$\begin{aligned} X(k+1) &= A_l X(k) + B_l u(k) + E_l v_{xref} \\ Y(k) &= C_l X(k) \end{aligned} \tag{25}$$

where $X(k) = [z(k)^{\mathrm{T}} \;\; e_x(k) \;\; e_y(k) \;\; e_a(k)]^{\mathrm{T}}$ denotes state, $u(k) = [F_x(k) \;\; \delta(k)]^{\mathrm{T}}$ is the control input, and the term v_{xref} represents the reference longitudinal velocity of the truck. The matrices A_l, B_l, C_l, and E_l are:

$$A_l = \left[\begin{array}{cccc|ccc} \multicolumn{4}{c|}{A} & \multicolumn{3}{c}{0^{N_\varphi \times 3}} \\ \hline T_s & 0 & 0 & 0^{1\times N_\varphi} & 1 & 0 & 0 \\ 0 & -T_s & -T_s D_L & 0^{1\times N_\varphi} & 0 & 1 & T_s v_x(k) \\ 0 & 0 & -T_s & 0^{1\times N_\varphi} & 0 & 0 & 1 \end{array}\right]$$

$$B_l = \begin{bmatrix} B \\ 0^{3\times 2} \end{bmatrix} \quad E_l = \begin{bmatrix} 0^{(N_\varphi+3)\times 1} \\ T_s \\ 0 \\ 0 \end{bmatrix} \quad C_l = \begin{bmatrix} I^{1\times 3} & 0^{1\times N_\varphi} & 0^{1\times 3} \\ 0^{3\times 3} & 0^{3\times N_\varphi} & I^{3\times 3} \end{bmatrix}$$

Suppose that $v_{xref}(k)$ remains constant during the prediction horizon N_p, the model (25) is linear. The control input sequence is denoted as:

$$U(k) = \{u(k|k), u(k+1|k) \cdots u(k+N_p-1|k)\} \tag{26}$$

where $k+i|k$ represents the predicted value at the time instant $k+i$ starting from the current time instant k.

Based on the control objectives presented, the desired output is denoted as follows:

$$r(k) = [v_f(k) \;\; e_{xref}(k) \;\; e_{yref}(k) \;\; e_{aref}(k)]^{\mathrm{T}} \tag{27}$$

where the reference values of $e_{xref}(k)$, $e_{yref}(k)$ and $e_{aref}(k)$ are 0.

In the prediction horizon $[0 \quad N_p-1]$, the optimization problem be described as:

Problem 1

$$\underset{U(k)}{\text{minimize}}\, J(Y(k), r(k), U(k)) \tag{28}$$

subject to

$$X(k+i+1|k) = A_l X(k) + B_l u(k) + E_l \xi(k) \tag{29a}$$
$$Y(k+i|k) = C_l X(k+i|k) \tag{29b}$$
$$Y_{min} \le Y(k+i|k) \le Y_{max} \tag{29c}$$
$$u_{min} \le u(k+i|k) \le u_{max} \tag{29d}$$
$$i{=}0, \cdots, N_p-1 \tag{29e}$$

where the objective function is:

$$\begin{aligned} &J(Y(k), r(k), U(k)) \\ &= \sum_{i=0}^{N_p-1} \|Y(k+i|k) - r(k+i|k)\|_Q^2 + \|U(k+i|k)\|_R^2 \end{aligned}$$

In the constraint conditions, Equation (29c) defines the output variable constraints, which include longitudinal velocity v_x, longitudinal position error e_x, heading error e_a, and lateral position error e_y. Equation (29d) specifies the system control variable constraints, which include longitudinal force F_x and front wheel steering angle δ. In the objective function, the weight matrices are Q and R.

Define the optimal solution of Problem 1 as:

$$U^*(k) = \{u^*(k|k), u^*(k+1|k) \cdots u^*(k+N_p-1|k)\} \tag{30}$$

The first element $u^*(k|k)$ of the optimal solution is applied to the truck at the time instant k. Then, Problem 1 is solved at the next time instant $k+1$, repeatedly.

Remark 4. At each time instant k, state $x(k)$ is processed by the pre-trained encoder $\varphi(\cdot)$ to yield the high-dimensional lifting state $\varphi\big(x(k)\big)$. The lifting state is combined with $x(k)$ to form (5). Subsequently, $e_x(k)$, $e_y(k)$, and $e_a(k)$ are incorporated, producing the augmented state $X(k) = [z(k)^{\mathrm{T}} \quad e_x(k) \quad e_y(k) \quad e_a(k)]^{\mathrm{T}}$. The state $X(k)$ then serves as the initial condition for the linear model (25), which is integrated over the prediction horizon. The output $Y(k)$ extracts the original state $x(k)$ from augmented state $X(k)$ at each time instant k. Only the first control action $u^*(k|k)$ is applied to the truck, and the procedure is repeated at the next time instant. Notably, both the encoder $\varphi(\cdot)$ and the Koopman matrices (A_l, B_l) remain fixed during online optimization.

IV. A COMPARISON: ADAPTIVE CRUISE CONTROL WITH 5DOF NONLINEAR DYNAMICS MODEL

To highlight the advantages of the proposed approach, comparative studies are carried out against two control algorithm: the first-principles-based 5DOF-ACC method and the LTV model-based LTV-ACC method.

A. Vehicle Dynamics Model

Trucks encompass various mechanical components such as steering mechanisms, suspensions, and tires. The degree of freedom and complexity increase with the increase of modeling accuracy. Accordingly, truck models are categorized into 2DOF [46], 3DOF [47], and 5DOF [48], respectively.

A 5DOF dynamics model of the truck is adopted:

$$\begin{cases} \dot{v}_x = v_y\omega + \dfrac{F_{xf}\cos\delta_f - F_{yf}\sin\delta_f + F_{xr}}{m} \\ \qquad - \dfrac{\mathrm{sgn}(v_x)\cdot C_\alpha A_{L\alpha}\frac{\rho}{2}v_x^2}{m} \\ \dot{v}_y = -v_x\omega + \dfrac{F_{xf}\sin\delta_f + F_{yf}\cos\delta_f + F_{yr}}{m} \\ \qquad - \dfrac{\mathrm{sgn}(v_y)\cdot C_\beta A_{L\beta}\frac{\rho}{2}v_y^2}{m} \\ \dot{\omega} = \dfrac{(F_{xf}\sin\delta_f + F_{yf}\cos\delta_f)a - F_{yr}b}{I_z} \\ \dot{\omega}_f = \dfrac{T_{df} - R_e\cdot F_{xf}}{J_f} \\ \dot{\omega}_r = \dfrac{T_{dr} - R_e\cdot F_{xr}}{J_r} \end{cases} \tag{31}$$

where ω_f, and ω_r are the front tires and rear-tires angular velocity, respectively. The terms C_α and C_β are the coefficients of longitudinal and lateral air resistance, respectively, while $A_{L\alpha}$ and $A_{L\beta}$ denote the longitudinal and lateral frontal area of the truck. The term ρ stands for air density, and the parameter δ_f represents the steering angle for the front tires. The drive torques for the front and rear tires are indicated by T_{df} and T_{dr}, respectively. The moments of inertia of the front and rear tires are represented by J_f and J_r, respectively. The vehicle mass is denoted by m, and the rolling radius of the tires is represented by R_e.

To capture the nonlinear characteristics of tire forces accurately, the Magic Formula tire model is adopted [49]:

$$\begin{cases} F_{xf} = \mathcal{D}\sin\left(\mathcal{C}\arctan\left(\mathcal{B}k_f - \mathcal{E}\left(\mathcal{B}k_f - \arctan\mathcal{B}k_f\right)\right)\right) \\ F_{xr} = \mathcal{D}\sin\left(\mathcal{C}\arctan\left(\mathcal{B}k_r - \mathcal{E}\left(\mathcal{B}k_r - \arctan\mathcal{B}k_r\right)\right)\right) \\ F_{yf} = \mathcal{D}\sin\left(\mathcal{C}\arctan\left(\mathcal{B}\alpha_f - \mathcal{E}\left(\mathcal{B}\alpha_f - \arctan\mathcal{B}\alpha_f\right)\right)\right) \\ F_{yr} = \mathcal{D}\sin\left(\mathcal{C}\arctan\left(\mathcal{B}\alpha_r - \mathcal{E}\left(\mathcal{B}\alpha_r - \arctan\mathcal{B}\alpha_r\right)\right)\right) \end{cases}$$

where F_{xf} and F_{xr} denote the longitudinal forces of the front and rear tires, respectively, while F_{yf} and F_{yr} represent the lateral forces of the front and rear tires, respectively. The slip ratios of the front and rear tires are denoted by k_f and k_r, respectively, and the slip angles of the front and rear tires are represented by α_f and α_r, respectively. The terms of $\mathcal{B}, \mathcal{C}, \mathcal{D}$ and $\mathcal{E}$ represent tire parameters, which are obtained by fitting the tire data in TruckSim. The detailed values are shown in Table III.

The slip ratios of the front and rear tires are calculated as follows:

$$\begin{cases} k_f = \dfrac{\omega_f\cdot R_e - v_x}{\max(\omega_f\cdot R_e, v_x)} \\ k_r = \dfrac{\omega_r\cdot R_e - v_x}{\max(\omega_r\cdot R_e, v_x)} \end{cases} \tag{32}$$

TABLE III
TIRE PARAMETERS

Tire forces	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$	$\mathcal{E}$
F_{xf}	8.61	1.58	22053	0.5624
F_{xr}	8.61	1.58	44625	0.5624
F_{yf}	6.59	2.42	22053	-0.3028
F_{yr}	6.59	2.42	44625	-0.3028

The slip angles of the front and rear tires are calculated as follows:

$$\begin{cases} \alpha_f = \operatorname{sgn}(v_{\alpha f}) \cdot \arctan \dfrac{v_{\beta f}}{v_{\alpha f}} \\ \alpha_r = \operatorname{sgn}(v_{\alpha r}) \cdot \arctan \dfrac{v_{\beta r}}{v_{\alpha r}} \end{cases} \tag{33}$$

where $v_{\alpha f}$, $v_{\beta f}$, $v_{\alpha r}$, and $v_{\beta r}$ are the longitudinal and lateral velocity of the front and rear tires, respectively. The truck parameters are listed in Table IV.

TABLE IV
THE TRUCK PARAMETERS

Parameters	Symbol	Value
Mass	m	18000 kg
Front wheel to center of mass distance	a	3.5 m
Rear wheel to center of mass distance	b	1.5 m
Gravitational acceleration	g	9.8 m/s^2
Moment of inertia about the Z-axis	I_z	130421 kg·m^2
Front wheel rotational inertia	J_f	24 kg·m^2
Rear wheel rotational inertia	J_r	48 kg·m^2
Wheel rolling radius	R_e	0.51 m

B. Nonlinear Model Predictive Controller

The vehicle-road relative position model (18), (20) and the truck dynamics model (31) are formulated as a discrete time nonlinear system:

$$\begin{aligned} x_n(k+1) &= f(x_n(k), u_n(k)) \\ y_n(k) &= C_n x_n(k) \end{aligned} \tag{34}$$

where $C_n = \begin{bmatrix} I^{1\times1} & 0^{1\times7} \\ 0^{3\times5} & I^{3\times3} \end{bmatrix}$ is the output matrix, $x_n(k) = [v_x(k)\ v_y(k)\ \omega(k)\ \omega_f(k)\ \omega_r(k)\ e_x(k)\ e_y(k)\ e_a(k)]^{\mathrm{T}}$ represents the extended state, $u_n(k)$ is the control input.

The optimization problem of 5DOF-ACC is:

Problem 2

$$\underset{U_n(k)}{\text{minimize}}\ \bar{J}(y_n(k), r(k), U_n(k)) \tag{35}$$

subject to

$$x_n(k+i+1|k) = f_n(x_n(k+i|k), u_n(k+i|k)) \tag{36a}$$
$$y_n(k+i|k) = C_n x_n(k+i|k) \tag{36b}$$
$$y_{n,min} \le y_n(k+i|k) \le y_{n,max} \tag{36c}$$
$$u_{n,min} \le u_n(k+i|k) \le u_{n,max} \tag{36d}$$
$$i = 0, \cdots, N_p - 1 \tag{36e}$$

where

$$\bar{J}(y_n(k), r(k), U_n(k)) = \sum_{i=0}^{N_p-1} \|y_n(k+i|k) - r(k+i|k)\|^2_{Q_N} + \|u_n(k+i|k)\|^2_{R_N}$$

In the constraint conditions, Equation (36c) defines the output variable constraints, which include longitudinal velocity v_x, longitudinal position error e_x, heading error e_a, and lateral position error e_y. Equation (36d) specifies the system control variable constraints, which include longitudinal torque T and front wheel steering angle δ. In the objective function, Q_N and R_N are the weight matrices to be determined, and the penalty term $\|y_n(k+i|k) - r(k+i|k)\|^2_{Q_N}$ aims to minimize the output.

Define the optimal solution of Problem 2 as:

$$U_n^*(k) = \{u_n^*(k|k), u_n^*(k+1|k) \cdots u_n^*(k+N_p-1|k)\}$$

where the first element $u_n^*(k|k)$ is applied to the truck at the time instant k. Then, Problem 2 is solved at the next time instant $k+1$, repeatedly.

Note that 5DOF-ACC method employs a nonlinear predictive model, taking into account the nonlinear dynamics of tires and lateral and longitudinal coupling characteristics of trucks.

C. Model Predictive Controller Based on LTV Model

The Jacobian linearization of the system (34) at the *current* operating point is performed, and a discrete time LTV model is obtained:

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{A}(k)\tilde{x}(k) + \tilde{B}(k)\tilde{u}(k) \\ \tilde{y}(k) &= \tilde{C}\tilde{x}(k) \end{aligned} \tag{37}$$

where $\tilde{x}(k)$ is the system state, with the system output matrix $\tilde{C} = \begin{bmatrix} I^{1\times1} & 0^{1\times7} \\ 0^{3\times5} & I^{3\times3} \end{bmatrix}$, and $\tilde{u}(k)$ denotes the control input. The terms $\tilde{A}(k)$ and $\tilde{B}(k)$ represent the Jacobian-linearized state and input matrices, respectively, i.e.,

$$\tilde{A}(k) = \frac{\partial f(\bar{x}, \bar{u})}{\partial \bar{x}}, \quad \tilde{B}(k) = \frac{\partial f(\bar{x}, \bar{u})}{\partial \bar{u}} \tag{38}$$

where $\bar{x}$ and $\bar{u}$ denote the current state and control input, respectively. As the linearized system matrices are updated at each time instant based on the current operating point, the LTV model is obtained.

The optimization problem of LTV-ACC is:

Problem 3

$$\min_{\tilde{U}(k)}\ \tilde{J}\big(\tilde{y}(k),\ r(k),\ \tilde{U}(k)\big) \tag{39}$$

subject to

$$\tilde{x}(k+i+1 \mid k) = \tilde{A}(k)\,\tilde{x}(k+i \mid k) + \tilde{B}(k)\,\tilde{u}(k+i \mid k) \tag{40a}$$

$$\tilde{y}(k+i \mid k) = \tilde{C}\,\tilde{x}(k+i \mid k) \tag{40b}$$

$$\tilde{y}_{\min} \le \tilde{y}(k+i \mid k) \le \tilde{y}_{\max} \tag{40c}$$

$$\tilde{u}_{\min} \le \tilde{u}(k+i \mid k) \le \tilde{u}_{\max} \tag{40d}$$

$$i = 0, \ldots, N_p - 1 \tag{40e}$$

where the cost function is

$$\begin{aligned} &\tilde{J}(\tilde{y}(k),\, r(k),\, \tilde{U}(k)) \\ &= \sum_{i=0}^{N_p-1} \left\| \tilde{y}(k+i \mid k) - r(k+i \mid k) \right\|_{\tilde{Q}}^2 \\ &+ \sum_{i=0}^{N_p-1} \left\| \tilde{u}(k+i \mid k) \right\|_{\tilde{R}}^2 \end{aligned} \tag{41}$$

The constraints on the output variables, including the longitudinal velocity v_x, longitudinal position error e_x, lateral position error e_y, and heading error e_a, are defined by Equation (40c). Equation (40d) specifies the system control variable constraints, which include both longitudinal torque T and front wheel steering angle δ. The terms $\tilde{Q}$ and $\tilde{R}$ are the weight matrices to be determined.

Denote the optimal solution of Problem 3 as

$$\tilde{U}^*(k) = \left\{ \tilde{u}^*(k \mid k),\, \tilde{u}^*(k+1 \mid k), \ldots, \tilde{u}^*(k+N_p-1 \mid k) \right\}$$

The first element $\tilde{u}^*(k \mid k)$ of the optimal solution is applied to the truck at the time instant k. Then, Problem 3 is solved at the next time instant $k+1$ with new measurements, repeatedly.

V. Simulation Results

In this section, comparative experiments are conducted in TruckSim and Simulink. The proposed method is compared not only with the three methods described in the preceding sections but also with the EDMD method employing an empirically selected basis function (ESBF-ACC). The lifted state spaces selected for both the ESBF-ACC method and DNN-ACC method are 30-dimensional. For DNN-ACC, LTV-ACC and ESBF-ACC methods, quadratic programming problems are formulated, which can be solved by the 'quadprog' function of Matlab Optimization Toolbox. For 5DOF-ACC method, the optimization problem can be solved by the 'fmincon' function, which adopts the sequential quadratic programming method. The controller parameters are shown in Table V, where Q_E, R_E are the weight matrices of the ESBF-ACC method.

Case I: Small curvature road scenario

A highway consisting of straight and curved roads is considered to test the effectiveness of the proposed method. The maximum curvature of the road depicted in Fig. 7(a) is 0.0025, and the width is 3.7 m, which is consistent with the standard highway curvature in China [50]. The adhesion coefficient $\mu = 0.85$. The leading truck travels along the curved road at a constant speed of 90 km/h. The typical dimensions of the truck are 2.5 m in width and 10 m in length.

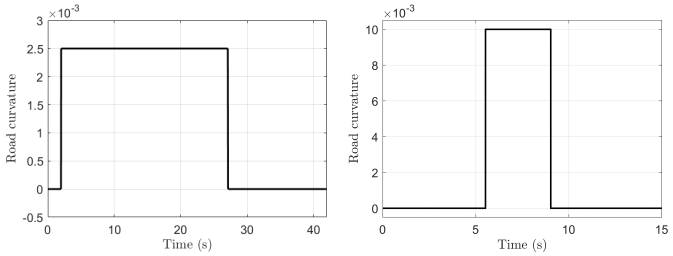

Fig. 7. Road curvature (a) Small curvature road scenario (b) Large curvature road scenario.

TABLE V
Controller Parameters

Parameter	Value
N_p	10
Constraints	
v_x	[0, 100] km/h
v_y	$[-5, 5]$ km/h
ω	$[-0.5, 0.5]$ rad/s
F_x	$[-6000, 6000]$ N
δ	$[-10, 10]$ deg
Weight Matrices	
Q	$\mathrm{diag}\{100, 800, 40, 500\}$
R	$\mathrm{diag}\{0.01, 1\}$
Q_N	$\mathrm{diag}\{4\times10^6, 7\times10^6, 5\times10^7, 5\times10^7\}$
R_N	$\mathrm{diag}\{0.001, 5\times10^7\}$
$\tilde{Q}$	$\mathrm{diag}\{1\times10^8, 100, 50, 1\times10^5\}$
$\tilde{R}$	$\mathrm{diag}\{0.001, 0.01\}$
Q_E	$\mathrm{diag}\{1\times10^4, 1\times10^4, 1000, 100\}$
R_E	$\mathrm{diag}\{0.005, 4\}$

The look-ahead distance D_L is 15 m, and the desired distance between the two trucks is 15 m.

Fig. 8(a)-Fig. 8(c) illustrate the evolution of system state, in which the DNN-ACC exhibits a smaller error in tracking the longitudinal velocity of the leading truck compared to 5DOF-ACC, LTV-ACC, and ESBF-ACC. The red dashed line represents DNN-ACC, the blue dotted line corresponds to 5DF-ACC, the magenta dotted line denotes LTV-ACC, and the green dotted line represents ESBF-ACC. Fig. 8(d) shows that the longitudinal position error of DNN-ACC is less than that of the other three methods. Fig. 8(e) indicates that the maximum lateral distance error of both systems remains within 0.6 m, that is, it does not exceed the road boundary.

As shown in Fig. 8(f), the maximum lateral acceleration under both control strategies remains below $0.16\ \mathrm{m/s^2}$, which is considerably lower than the $4\ \mathrm{m/s^2}$ rollover threshold for a fully loaded truck [51]. This indicates that four control strategies maintain safe lateral acceleration at the corner. However, LTV-ACC and ESBF-ACC result in overshooting when the truck enters the curve and reenters the straightaway. Fig. 8(g) clearly illustrates that DNN-ACC can follow the expected trajectory with higher precision than the other three methods.

Fig. 8(h) demonstrates the computational time of solving the optimization problems of the four schemes. The average computational time and maximum computational time

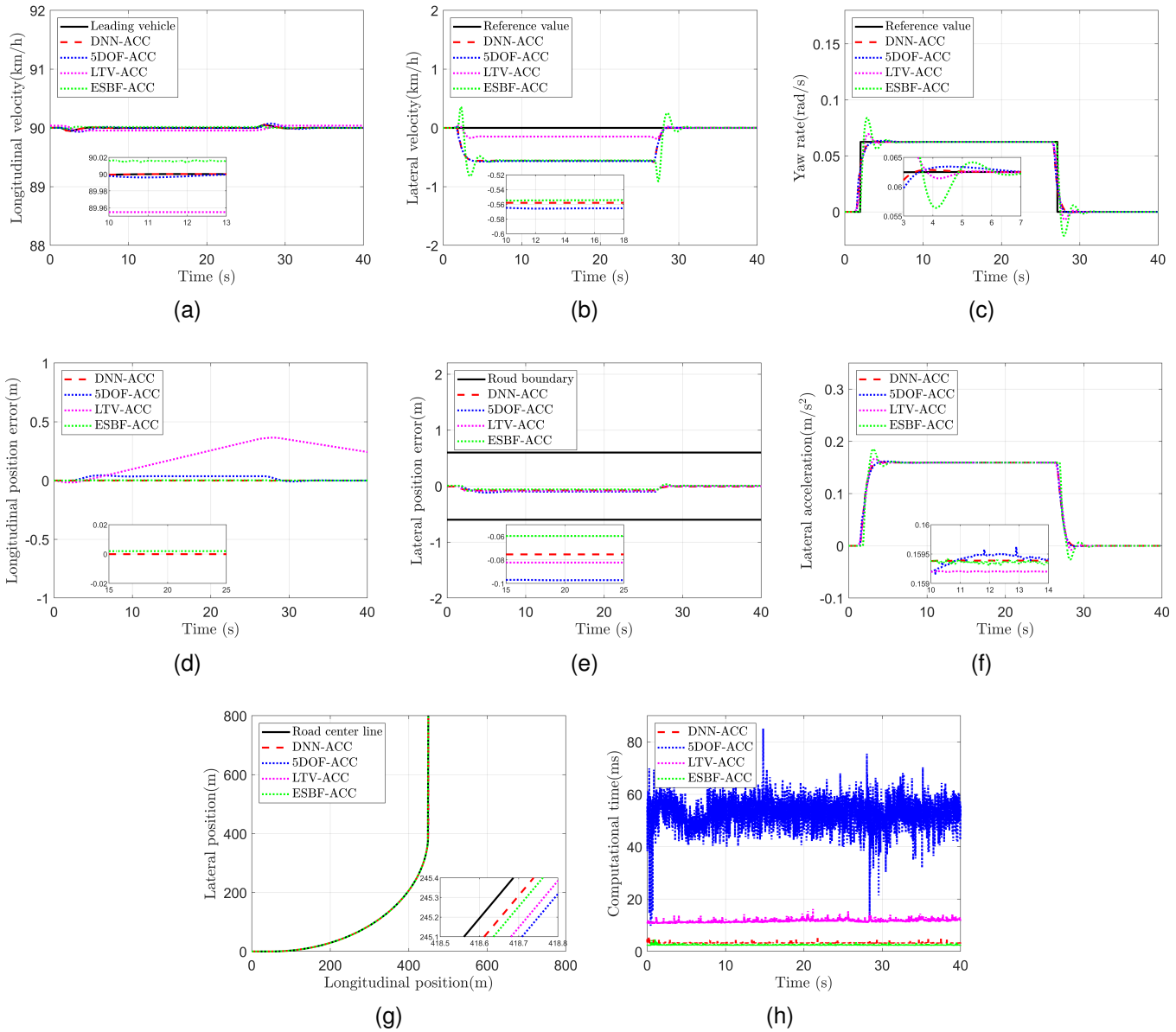

Fig. 8. (a) Longitudinal velocity (b) Lateral velocity (c) Yaw rate (d) Longitudinal position error (e) Lateral position error (f) Lateral acceleration (g) Driving path of trucks (h) Computational time.

are shown in Table VI, demonstrating that the data-driven control strategy achieves a substantial improvement in computational efficiency.

TABLE VI
COMPUTATIONAL TIME

Methods	Average	Maximum
DNN-ACC	3.75 ms	5.29 ms
5DOF-ACC	52.82 ms	85.07 ms
LTV-ACC	12.59 ms	15.03 ms
ESBF-ACC	2.46 ms	4.59 ms

The above results show that, although the LTV-ACC method maintains a lower lateral velocity, it incurs a higher computational burden and lower longitudinal control accuracy compared to the proposed DNN-ACC method. The LTV-ACC method incurs a high computational burden due to the requirement of performing linearization at the current operating point. Furthermore, DNN-ACC and ESBF-ACC differ in how the lifting functions are obtained: the former learns the lifting function via data-driven training, whereas the latter relies on empirically or manually chosen basis functions to derive the lifting function. The ESBF-ACC method exhibits limited capability in capturing the nonlinear characteristics of the system, which leads to reduced control accuracy. Consequently, when prior expertise for basis selection is lacking, the proposed method can both reduce the online computational burden and maintain longitudinal and lateral safety.

Case II: Large curvature road scenario

An extreme road scenario is designed to verify the applicability of the proposed method. These large curvature

road scenarios take account of the nonlinear characteristics of the tires.

The maximum curvature of the road depicted in Fig.7(b) is 0.01 and the speed of the leading truck is 80 km/h. Fig. 9(a) reveals that the side slip angle of the following truck's front tires exceeds 5° between 6.8 s and 8.5 s, indicating that the nonlinear characteristics of the tires during this period cannot be neglected [52].

Fig. 9(b) - Fig. 9(d) depict the evolution of the system state, showing that the DNN-ACC achieves a smaller longitudinal velocity error than the other three methods. Fig. 9(c) shows that LTV-ACC has a smaller lateral velocity. Fig. 9(e) infers that LTV-ACC incurs a larger longitudinal position error than DNN-ACC, ESBF-ACC, and 5DOF-ACC. In Fig. 9(f), the lateral position error is presented. The maximum lateral position error of 5DOF-ACC is 0.44 m, which is very close to the road boundary. The maximum lateral position errors of DNN-ACC, ESBF-ACC, and LTV-ACC are 0.12 m, 0.23m, and 0.21 m, respectively, indicating that the proposed method effectively maintains lateral safety. According to Fig. 9(g), 5DOF-ACC and LTV-ACC result in overshooting when the truck reenters the straightaway. Fig. 9(h) illustrates that DNN-ACC can follow the expected trajectory with higher precision than ESBF-ACC.

Fig. 9(i) demonstrates the computational time of solving the optimization problems of the four schemes. The average computational time and maximum computational time are shown in Table VII.

TABLE VII
COMPUTATIONAL TIME

Methods	Average	Maximum
DNN-ACC	3.35 ms	4.89 ms
5DOF-ACC	70.35 ms	123.47 ms
LTV-ACC	12.37 ms	16.75 ms
ESBF-ACC	2.56 ms	3.76 ms

The 5DOF-ACC method relies on first-principles modeling, which has high model complexity and computational burden. In contrast, LTV-ACC, despite maintaining lower lateral velocities, exhibits poorer longitudinal tracking accuracy and also incurs a higher computational burden. The data-driven DNN-ACC and ESBF-ACC utilize Koopman operator theory to construct a linear model that captures the truck's nonlinear dynamics. Consequently, DNN-ACC not only reduces modeling errors and ensures longitudinal and lateral safety but also significantly reduces computational burden.

VI. CONCLUSION

This paper proposed a data-driven ACC method for trucks integrated the lane-keeping model. Firstly, a global linear model of trucks was established based on the Koopman operator theory, where DNN technique was adopted to enhance the model's accuracy. Then, a linear predictive controller was designed by incorporating the lane-keeping model with the Koopman linear model. Compared with conventional 5DOF-ACC, LTV-ACC, and ESBF-ACC algorithm, the proposed DNN-ACC algorithm demonstrated superior trajectory tracking capabilities, which enabled the truck to adhere more closely to the road centerline, and significantly reduced the computational time.

In future work, an adaptive cruise control strategy for trucks that considers variations in road adhesion coefficients will be developed. These efforts will help bridge the gap between theoretical developments and practical implementations in advanced truck ACC.

REFERENCES

[1] Y. Wang, J. Jiang, S. Li, R. Li, S. Xu, J. Wang, and K. Li, "Decision-making driven by driver intelligence and environment reasoning for high-level autonomous vehicles: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 10 362–10 381, 2023.

[2] J. Fischer-Wolfarth and G. Meyer, "Advanced microsystems for automotive applications 2014," *Smart systems for safe, clean and automated vehicles. Switzerland: Springer International Publishing*, 2014.

[3] H. E. Glida, C. Sentouh, A. Chelihi, J. Floris, and J.-C. Popieul, "Event-triggered adaptive fault-tolerant control based on sliding mode/neural network for lane keeping assistance systems in steer-by-wire vehicles," *IEEE Transactions on Intelligent Vehicles*, 2024.

[4] J. Dahl, G. R. de Campos, and J. Fredriksson, "Prediction-uncertainty-aware threat detection for adas: A case study on lane-keeping assistance," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 2914–2925, 2023.

[5] L. Luo, "Vehicle adaptive cruise control and the corresponding macroscopic traffic flow model," *PhD, Zhejiang University, Hangzhou, China*, 2011.

[6] S. Moon, H. J. Kang, and K. Yi, "Multi-vehicle target selection for adaptive cruise control," *Vehicle System Dynamics*, vol. 48, no. 11, pp. 1325–1343, 2010.

[7] D. Z. Hou, "Study on vehicle forward collision avoidance system," *Beijing: Tsinghua University*, 2004.

[8] D. Zhang, "Vehicular adaptive cruise control on curved road," *Beijing: Tsinghua University*, 2011.

[9] J. Guo, K. Li, and Y. Luo, "Coordinated control of autonomous four wheel drive electric vehicles for platooning and trajectory tracking using a hierarchical architecture," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 10, p. 101001, 2015.

[10] P. Ren, H. Jiang, and X. Xu, "Research on a cooperative adaptive cruise control (cacc) algorithm based on frenet frame with lateral and longitudinal directions," *Sensors*, vol. 23, no. 4, p. 1888, 2023.

[11] J. Yao, G. Chen, and Z. Gao, "Target vehicle selection algorithm for adaptive cruise control based on lane-changing intention of preceding vehicle," *Chinese Journal of Mechanical Engineering*, vol. 34, no. 1, pp. 1–18, 2021.

[12] D. J. Cole, "Fundamental issues in suspension design for heavy road vehicles," *Vehicle System Dynamics*, vol. 35, no. 4-5, pp. 319–360, 2001.

[13] B. C. Chen and H. Peng, "Rollover warning for articulated heavy vehicles based on a time-to-rollover metric," *Journal of Dynamic Systems, Measurement, and Control*, vol. 127, no. 3, pp. 406–414, 2005.

[14] J. Hou, H. Lei, Z. Fu, P. Yuan, Y. Yin, H. Feng, Z. Li, M. Zhang, M. Cui, and Y. Xu, "A novel rollover warning approach for commercial vehicles using unscented kalman filter," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–13, 2022.

[15] H. Patil, K. Devika, G. Vivekanandan, S. Sivaram, and S. C. Subramanian, "Direct yaw-moment control integrated with wheel slip regulation for heavy commercial road vehicles," *IEEE Access*, vol. 10, pp. 69 883–69 895, 2022.

[16] J. Zhao, Z. Wang, Y. Lv, J. Na, C. Liu, and Z. Zhao, "Data-driven learning for h_∞ control of adaptive cruise control systems," *IEEE Transactions on Vehicular Technology*, 2024.

[17] D. He, W. He, and X. Song, "Efficient predictive cruise control of autonomous vehicles with improving ride comfort and safety," *Measurement and Control*, vol. 53, no. 1-2, pp. 18–28, 2020.

[18] M. Choi and S. B. Choi, "Model predictive control for vehicle yaw stability with practical concerns," *IEEE transactions on vehicular technology*, vol. 63, no. 8, pp. 3539–3548, 2014.

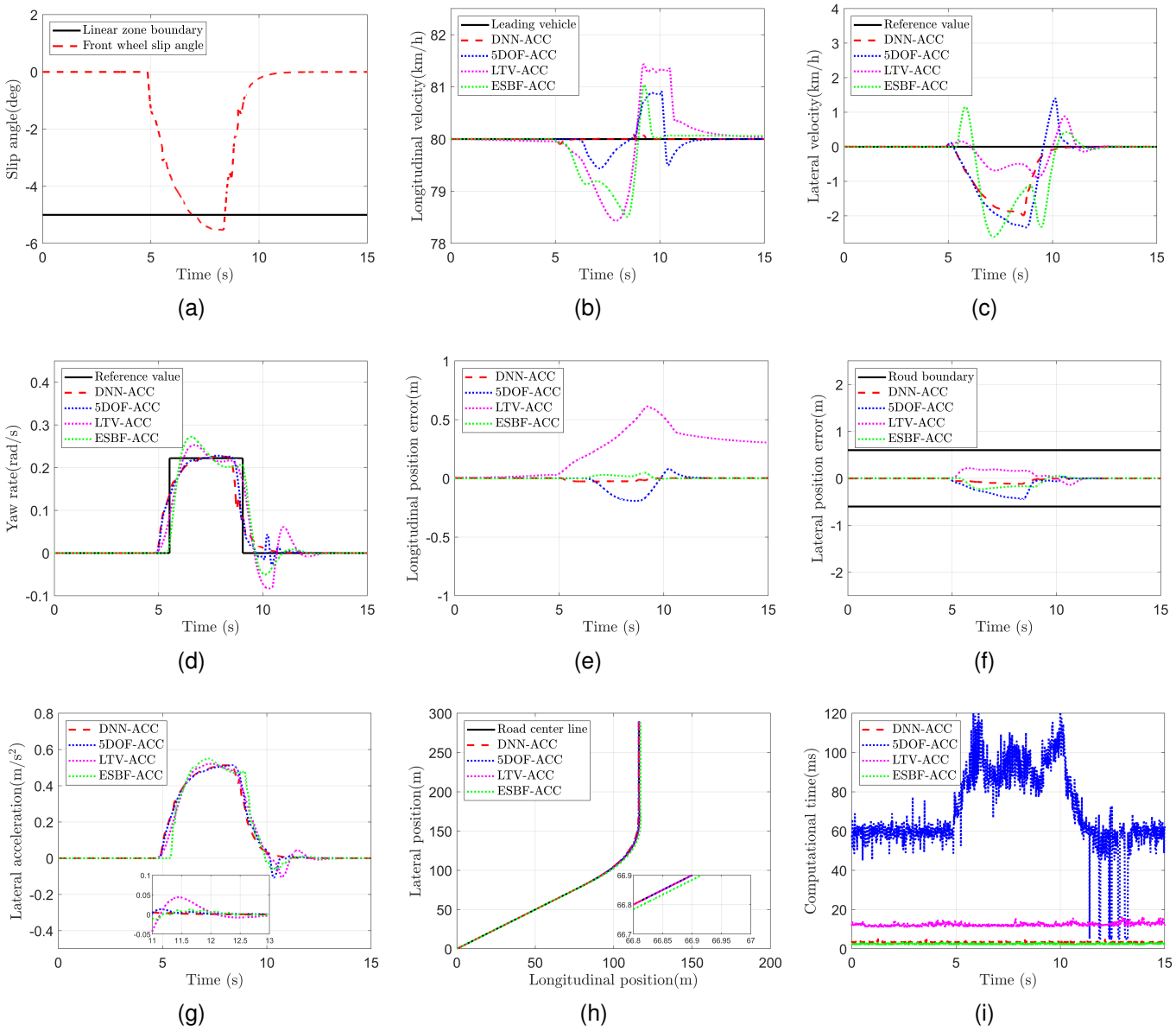

Fig. 9. (a) Slip angle (b) Longitudinal velocity (c) Lateral velocity (d) Yaw rate (e) Longitudinal position error (f) Lateral position error (g) Lateral acceleration (h) Driving path of trucks (i) Computational time.

[19] G. S. Sankar, M. Kim, and K. Han, "Data-driven leading vehicle speed forecast and its application to ecological predictive cruise control," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 11 504–11 514, 2022.

[20] Y. Jia, R. Jibrin, and D. Görges, "Energy-optimal adaptive cruise control for electric vehicles based on linear and nonlinear model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 173–14 187, 2020.

[21] J. David, P. Brom, F. Starỳ, J. Bradáč, and V. Dynybyl, "Application of artificial neural networks to streamline the process of adaptive cruise control," *Sustainability*, vol. 13, no. 8, p. 4572, 2021.

[22] B. Gao, K. Cai, T. Qu, Y. Hu, and H. Chen, "Personalized adaptive cruise control based on online driving style recognition technology and model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 482–12 496, 2020.

[23] Z. Nie and H. Farzaneh, "Adaptive cruise control for eco-driving based on model predictive control algorithm," *Applied Sciences*, vol. 10, no. 15, p. 5271, 2020.

[24] S. Li, K. Wan, B. Gao, R. Li, Y. Wang, and K. Li, "Predictive cruise control for heavy trucks based on slope information under cloud control system," *Journal of Systems Engineering and Electronics*, vol. 33, no. 4, pp. 812–826, 2022.

[25] J. Chen, Y. Ye, Q. Wu, R. Langari, and C. Tang, "Low-cost and high-performance adaptive cruise control based on inertial-triggered mechanism and multi-objective optimization," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 6, pp. 7279–7289, 2023.

[26] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[27] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2015.

[28] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[29] J. S. Kim, Y. S. Quan, and C. C. Chung, "Data-driven modeling and control for lane keeping system of automated driving vehicles: Koopman operator approach," in *2022 22nd international conference on control, automation and systems (ICCAS)*. IEEE, 2022, pp. 1049–1055.

[30] J. S. Kim, Y. S. Quan, C. C. Chung, and W. Y. Choi, "K-smpc: Koopman operator-based stochastic model predictive control for enhanced lateral control of autonomous vehicles," *IEEE Access*, 2025.

[31] Y. Meng, H. Li, M. Ornik, and X. Li, "Koopman-based data-driven techniques for adaptive cruise control system identification," in *27th IEEE International Conference on Intelligent Transportation Systems (ITSC), IEEE*, 2024.

[32] M. Švec, Š. Ileš, and J. Matuško, "Optimizing vehicle handling through koopman-based model predictive torque vectoring: An experimental investigation," *Control Engineering Practice*, vol. 158, p. 106272, 2025.

[33] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, p. 4950, 2018.

[34] R. Mao, T. Meng, K. Wang, J. Lei, and W. Wang, "Deep koopman-operator-based model predictive control for free-floating space robots with disturbance observer," *Aerospace Science and Technology*, vol. 154, p. 109515, 2024.

[35] T. Zhao, M. Yue, and J. Wang, "Deep-learning-based koopman modeling for online control synthesis of nonlinear power system transient dynamics," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 10, pp. 10 444–10 453, 2023.

[36] H. Shi and M. Q. H. Meng, "Deep koopman operator with control for nonlinear systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7700–7707, 2022.

[37] Y. Zuo, C. Yang, S. Li, W. Wang, C. Xiang, and T. Qie, "A model predictive trajectory tracking control strategy for heavy-duty unmanned tracked vehicle using deep koopman operator," *Engineering Applications of Artificial Intelligence*, vol. 159, p. 111698, 2025.

[38] H. Chen and C. Lv, "Incorporating eso into deep koopman operator modeling for control of autonomous vehicles," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 5, pp. 1854–1864, 2024.

[39] Y. Xiao, X. Zhang, X. Xu, X. Liu, and J. Liu, "Deep neural networks with koopman operators for modeling and control of autonomous vehicles," *IEEE transactions on intelligent vehicles*, vol. 8, no. 1, pp. 135–146, 2022.

[40] M. Amrouche and D. M. Stipanović, "A formal characterization of activation functions in deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[41] L. Li, T. Guo, and S. Xu, "Simulation analysis of vehicle handling stability based on trucksim," *Journal of Physics: Conference Series. IOP Publishing*, vol. 1885, no. 3, p. 032043, 2021.

[42] G. Tekin and Y. Samim Unlusoy, "Design and simulation of an integrated active yaw control system for road vehicles," *International journal of vehicle design*, vol. 52, no. 1-4, pp. 5–19, 2010.

[43] J. Guo, K. Li, and Y. Luo, "Coordinated control of autonomous four wheel drive electric vehicles for platooning and trajectory tracking using a hierarchical architecture," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 10, p. 101001, 2015.

[44] C. J. Taylor, J. Košecká, R. Blasi, and J. Malik, "A comparative study of vision-based lateral control strategies for autonomous highway driving," *The International Journal of Robotics Research*, vol. 18, no. 5, pp. 442–453, 1999.

[45] J. Chen, Z. Shuai, H. Zhang, and W. Zhao, "Path following control of autonomous four-wheel-independent-drive electric vehicles via second-order sliding mode and nonlinear disturbance observer techniques," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 3, pp. 2460–2469, 2020.

[46] P. Seetapan and S. Chucheepsakul, "Dynamic responses of a two-span beam subjected to high speed 2dof sprung vehicles," *International Journal of Structural Stability and Dynamics*, vol. 6, no. 03, pp. 413–430, 2006.

[47] L. Liu, S. Shi, S. Shen, and J. Chu, "Vehicle planar motion stability study for tyres working in extremely nonlinear region," *Chinese journal of mechanical engineering*, no. 2, p. 185, 2010.

[48] X. Wang, S. Shi, L. Liu, and L. Jin, "Analysis of driving mode effect on vehicle stability," *International journal of automotive technology*, vol. 14, pp. 363–373, 2013.

[49] M. Belrzaeg, A. A. Ahmed, A. Q. Almabrouk, M. M. Khaleel, A. A. Ahmed, and M. Almukhtar, "Vehicle dynamics and tire models: An overview," *World J. Adv. Res. Rev*, vol. 12, no. 1, pp. 331–348, 2021.

[50] *Technical Standard of Highway Engineering: JTG B01-2014*. Ministry of Transport of the People's Republic of China, 2014.

[51] L. Mai, W. Qu, T. J. Zhu *et al.*, "Rover warning algorithm of heavy commercial vehicle based on model prediction," *Journal of Jilin University(Engineering and Technology Edition)*, vol. 40, no. 4, pp. 906–910, 2010.

[52] J. W. Gong, J. Yan, and X. Wei, *Model Predictive Control for self-driving vehicles*. Beijing: People's Posts and Telecommunications Publishing House, 2020.