Vaasan yliopisto
UNIVERSITY OF VAASA
OSUVA Open Science

This is a self-archived – parallel published version of this article in the publication archive of the University of Vaasa. It might differ from the original.

Data-Driven Lyapunov-Based Model Predictive Control for Improved Trajectory Tracking in Multi-Wheel-Independent-Drive Electric Vehicle

Author(s): Zhang, Yongkang; Chen, Jicheng; Wei, Henglai; Simões, Marcelo Godoy; Zhang, Hui

Title: Data-Driven Lyapunov-Based Model Predictive Control for Improved Trajectory Tracking in Multi-Wheel-Independent-Drive Electric Vehicle

Year: 2025

Version: Accepted manuscript

Copyright ©2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Please cite the original version:

Zhang, Y., Chen, J., Wei, H., Simões, M. G. & Zhang, H. (2025). Data-Driven Lyapunov-Based Model Predictive Control for Improved Trajectory Tracking in Multi-Wheel-Independent-Drive Electric Vehicle. *IEEE Transactions on Vehicular Technology*. https://doi.org/10.1109/TVT.2025.3587541

Data-Driven Lyapunov-Based Model Predictive Control for Improved Trajectory Tracking in Multi-Wheel-Independent-Drive Electric Vehicle

Yongkang Zhang, *Student Member, IEEE*, Jicheng Chen, *Member, IEEE*, Henglai Wei, *Member, IEEE*, Marcelo Godoy Simões, *Fellow, IEEE*, Hui Zhang, *Fellow, IEEE*

***Abstract*—This paper proposes a data-driven Lyapunov-based Model Predictive Control (LMPC) method for multi-wheel-independent-drive electric vehicles to enhance the trajectory tracking accuracy while ensuring the vehicle stability. To improve the accuracy of the vehicle dynamics model, we first develop a temporal residual network to learn the residual between the nominal vehicle dynamics and the actual vehicle dynamics from a lot of training data offline. The temporal residual network predicts the vehicle dynamics residual online based on the vehicle states within a past time window. Then, by combining the nominal vehicle dynamics model with the temporal residual network, a more accurate compensation model is obtained. Building on this, we propose a novel data-driven control strategy specifically optimized for trajectory tracking. To ensure vehicle stability, a Lyapunov-based constraint based on the designed backstepping controller is incorporated into the data-driven LMPC. Subsequently, theoretical analysis is presented to validate the stability of the system. In the Carsim & Simulink co-simulation environment, we validated the effectiveness of the proposed temporal residual network and tracking control algorithm through open-loop and closed-loop simulations.**

***Index Terms*—Multi-wheel vehicle, Deep learning, Data-driven modeling, Lyapunov-based MPC, Trajectory tracking.**

I. Introduction

MULTI-wheel-independent-drive electric vehicles (MWIDEVs) are attracting increasing research interest for their enhanced maneuverability and dynamic performance [1]. A typical mechanical layout of an eight-wheeled MWIDEV, powered by individual hub motors at each wheel and energized by rechargeable high-voltage batteries, is illustrated in Fig. 1. The high-voltage batteries are powered by a generator. Two sets of steering mechanisms are installed on the first two axles, and the steering angle of each wheel is kept consistent. Owing to their advanced independent torque control capabilities, MWIDEVs have demonstrated highly versatile, effectively handling complex road conditions and diverse driving demands. From urgent demands such as post-disaster search, heavy-duty transport, and emergency rescue to precision-driven agricultural cultivation, mining exploration in harsh environments, and even extraterrestrial missions in space exploration, MWIDEVs demonstrate broad application potential [2]. Based on these diverse application scenarios, the autonomous driving capabilities of MWIDEVs emerge as a technological cornerstone for enabling reliable operations across such heterogeneous environments. Particularly, effective trajectory tracking control enables the vehicle to travel quickly, accurately and safely along a preset trajectory, which is of great significance for MWIDEVs to achieve efficient and reliable operations [3]. The performance of such tracking control systems critically depends on the accuracy of the vehicle dynamics model. For MWIDEVs, the increased number of actuators renders the conventional two-degree-of-freedom (2-DoF) dynamics model insufficient to accurately characterize vehicle motion states. This consequently results in discrepancies between model predictions and actual vehicle responses. The model residuals primarily arise from two aspects: unmodeled dynamics—such as the omission of suspension characteristics, drivetrain and body vibrations, and the tire elastic deformation—and model mismatches, including tire force estimation errors caused by the use of linear tire models and the small-angle assumption for front wheel steering.

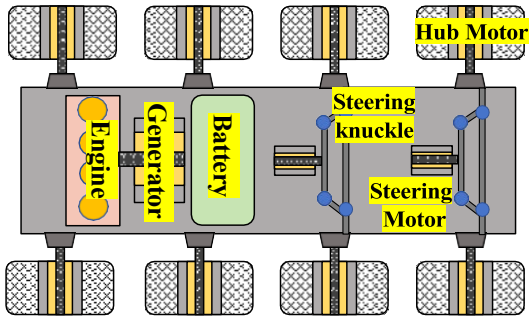


Fig. 1. The mechatronics layout of an MWIDEV with eight hub motors. Eight hub motors can provide sufficient power for the vehicle, and at the same time, they can distribute the vehicle vertical load. The generator on the vehicle can provide additional electrical power supply for the vehicle.

To address these challenges, many research efforts have been devoted to improving model accuracy, primarily cat-

Manuscript created January, 2025; This work was supported in part by Key R&D Program of Shandong Province, China under Grant 2023CXGC010111. *(Corresponding author: Hui Zhang.)*

Yongkang Zhang, Henglai Wei and Hui Zhang are with the School of Transportation Science and Engineering, Beihang University, Beijing, 100191, China (e-mail: yk1995@buaa.edu.cn; henglai.wei@ntu.edu.sg; huizhang285@buaa.edu.cn).

Jicheng Chen is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, 999077, China (e-mail: eejichengc@ust.hk).

Marcelo Godoy Simoes is with the School of Technology and Innovations, Electrical Engineering, University of Vaasa, Vaasa, 65200, Finland (e-mail: marcelo.godoy.simoes@uwasa.fi).

egorized into traditional control-based approaches [4], [5] and data-driven methods [6], [7]. In control-based methods, some states of the vehicle can be estimated by designing an observer, which improves the accuracy of the dynamics model. However, this method relies on some assumptions, and a high-precision vehicle dynamics model is still required when designing the observer. Existing data-driven methods rarely consider the temporal characteristics of vehicle states and overlook the state transitions of vehicles between adjacent time points, which affects estimation accuracy. Meanwhile, some data-driven models function as complete "black-box" models, making it difficult to guarantee system stability.

Motivated by insights from the preceding discussions, this paper introduces a novel data-driven technique for modeling vehicle residuals, complemented by a Lyapunov-based Model Predictive Control (LMPC) strategy tailored for trajectory tracking in MWIDEVs. The main contributions of this paper are as follows:

1) A temporal neural network-based residual compensation method is developed for MWIDEVs to improve the model accuracy. The temporal neural network takes the vehicle states over a past time window as input and predicts the model residual. A new driving cycle with broad driving condition coverage is designed to establish the training dataset. The model accuracy is improved by compensating the MPC prediction model with the output of the residual network model.
2) A novel data-driven LMPC trajectory tracking controller is developed to enable the MWIDEVs to track a given trajectory. We use the compensation model constructed by the temporal residual network combined with the nominal dynamics model as the prediction model for MPC. Backstepping-based Lyapunov constraints are incorporated into the MPC formulation to guarantee the stability of the tracking controller.

The rest of this paper is organized as follows. Related studies are discussed in Section II. Section III presents the nominal vehicle dynamics model of MWIDEVs and provided the problem formulation. Subsequently, we illustrate the proposed control structure. Section IV details the construction of the temporal residual network. In Section V, the design and analysis of data-driven LMPC trajectory tracking controller are provided. The simulation results and conclusion of this paper are given in Section VI and VII, respectively.

II. Related Work

The classical approaches for nonlinear trajectory tracking control include feedback linearization [8], [9], gain scheduling control [10], [11], sliding mode control (SMC) [12], [13], [14], backstepping control (BSC) [15], [16], MPC [17], [18], [19], and among others. The trade-offs among accuracy, safety, stability, and algorithmic efficiency in vehicle tracking control are an important direction of current research [20], [21]. Existing literature largely bases its analyses on vehicle kinematic properties or 2-DoF dynamics representation, both of which are well-suited to meeting the tracking control demands of conventional four-wheel vehicles without independent drive capabilities. Despite the advantages in state prediction and handling constraints and disturbances, MPC is confronted with critical challenges due to its sensitivity to model accuracy. Unmodeled dynamics and model mismatches can reduce the accuracy of the predictive model, leading to degraded control performance or even instability.

Researchers have made significant efforts in areas such as Linear Parameter-Varying (LPV) system design [22], [23], robust control design [24], and observer design [25], [26] to improve modeling accuracy or mitigate the effects of modeling residuals. Although LPV systems can improve modeling accuracy, the selection of vertices may introduce modeling conservatism [27], [28]. To better handle uncertainties and modeling residuals, the construction of safety regions in robust controller design can also introduce controller conservatism [29]. It's important to recognize that designing a robust observer or controller generally still relies on a precise vehicle model and accurate initial conditions.

Data-driven approaches reduce reliance on detailed physical models and hold substantial potential for improving modeling accuracy [30]. In the area of data-driven approaches to vehicle modeling and control, a significant portion of research efforts is dedicated to formulating data-centric control strategies [31], constructing predictive vehicle models [32], and approximating model residuals [33] or controller [34]. A class of data-driven modeling approaches involves directly using neural networks to capture the vehicle dynamics characteristics, resulting in a fully data-driven vehicle model [35]. Long Short-Term Memory (LSTM)-based neural networks are employed to estimate the vehicle's lateral dynamics model and applied in model predictive control [36]. Chen et al. [37] employed a set of convolutional LSTM-based neural networks and model-based methods to predict the states of surrounding vehicles. Deep neural networks are used to learn the nonlinear dynamics in the linearized lifted feature space represented by the Koopman operator, thereby achieving better tracking control performance [38].

Another class of data-driven modeling approaches involves using neural networks to estimate model residuals, which are then combined with the nominal reference model to improve modeling accuracy [33]. The vehicle model residual quantifies the discrepancy between the measured vehicle states and the model-predicted states. A representative work involves using Gaussian processes to estimate the model residuals [39]. Besides, neural networks are used to estimate the upper bound of the residual model and applied in robust MPC control to address the effects of unknown dynamics and disturbances [40]. It is noteworthy that these approaches frequently overlook the temporal coherence in vehicle state evolution which undermining their potential for high accuracy and broad applicability. Simultaneously, the stability guarantee of data-driven model predictive control remains a challenge.

III. Problem Formulation and Control Structure

This section focuses on the fundamental concepts and control structure presented in this paper. First, the nominal vehicle dynamics model of MWIDEV is introduced. Then,

based on the dynamics model, the model residual of vehicle dynamics and the problem formulation of this paper are described. Finally, the data-driven LMPC trajectory tracking control structure is presented.

A. Nominal Vehicle Dynamics Model of MWIDEV

The nominal vehicle dyanmics model describes the MWIDEV's theoretical dynamics characteristics under ideal operating conditions, excluding unmodeled dynamics, external disturbances and sensor errors. The 2-DoF vehicle dynamics model of MWIDEV is shown in Fig. 2. The steering of the vehicle is realized by the front four wheels, each of which shares the same steering angle $\delta(t)$. The longitudinal and lateral speeds are denoted as $u(t)$ and $v(t)$, respectively. $\beta(t)$ and $\varphi(t)$ are the sideslip angle and heading angle of the vehicle, respectively. $\gamma(t)$ is the yaw rate of the vehicle, F_{xij} and F_{yij} are longitudinal and lateral tire force in the tire coordinate system where $i = 1, \cdots, 4$ and $j = 1, 2$. α_{ij} is the sideslip angle of tire. The wheelbase of the MWIDEV is L, l_k $(k = 1, \cdots, 4)$ is the k-th axle to the center of mass of the vehicle.

Based on Fig. 2, the nominal vehicle dynamics model can be written as:

$$
\begin{aligned}
\dot{x}_o(t) &= -v(t)\sin\varphi(t) + u(t)\cos\varphi(t),\\
\dot{y}(t) &= v(t)\cos\varphi(t) + u(t)\sin\varphi(t),\\
\dot{\varphi}(t) &= \gamma(t),\\
\dot{u}(t) &= \gamma(t)v(t) + \frac{\eta_T}{mr_w}(T_1(t) + T_2(t)) - \frac{C_A}{m}u(t)^2 - gf_r,\\
\dot{v}(t) &= -\frac{\sum_{i=1}^{4} C_i}{mu(t)}v(t) + \frac{\sum_{i=1}^{2} l_i C_i - \sum_{i=3}^{4} l_i C_i}{mu(t)} + u(t)\\
&\quad \gamma(t) + \frac{\sum_{i=1}^{2} C_i}{m}\delta(t),\\
\dot{\gamma}(t) &= -\frac{\sum_{i=1}^{2} l_i C_i - \sum_{i=3}^{4} l_i C_i}{I_z u(t)}v(t) - \frac{\sum_{i=1}^{4} l_i^2 C_i}{I_z u(t)}\gamma(t)\\
&\quad + \frac{\sum_{i=1}^{2} l_i C_i}{I_z}\delta(t) + \frac{\eta_T L}{2r_w}(-T_1(t) + T_2(t)),
\end{aligned}
\tag{1}
$$

where $x_o(t)$ and $y(t)$ denote the vehicle position in global coordinate system, respectively. η_T is drivetrain mechanical efficiency, C_D denotes the resistance coefficient, r_w is the radius of the wheel, f_r is rolling resistance coefficient of road, the windward area and atmospheric density are represented by A and ρ_a, respectively. T_1 and T_2 represent the torque inputs to the four wheels on the left and right sides of the vehicle, respectively.

According to the linear tire dynamics model, the lateral tire force is :

$$
\begin{aligned}
F_{yij} &= C_{ij}\alpha_{ij},\\
C_i &= C_{i1} + C_{i2},
\end{aligned}
\tag{2}
$$

where C_{ij} is the cornering stiffness of ij-th tire, α_{ij} is the slip angle of ij-th tire, C_i is the combine cornering stiffness of i-th pair tires. We use a linear tire model instead of nonlinear models such as the Magic Formula, as nonlinear tire models require more parameters to achieve higher accuracy compared to linear ones. If the parameters of the nonlinear tire model are inaccurate, the prediction performance will degrade significantly. At the same time, the road of the vehicle in this study is a flat with good adhesion, and the tires are in the linear range.

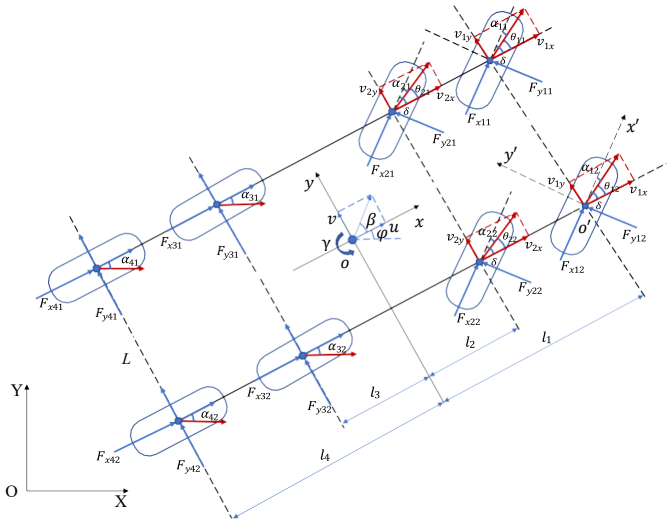


Fig. 2. The 2-DoF vehicle dynamics model of MWIDEV. This model includes three coordinate systems: the global coordinate system (XOY), the vehicle coordinate system (xoy) fixed at the vehicle's center of gravity, and the tire coordinate system ($x'o'y'$).

The nominal vehicle dynamics model of (1) can be expressed in a compact form as follows:

$$
\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{\mu}(t)),
\tag{3}
$$

where $f : \mathrm{R}^{6\times1} \times \mathrm{R}^{3\times1} \to \mathrm{R}^{6\times1}$ is a nonlinear function representing nominal vehicle dynamics. $\boldsymbol{x}(t) = [x_o(t), y(t), \varphi(t), u(t), v(t), \gamma(t)]^{\mathrm{T}}$ is the nominal state vector, $\boldsymbol{\mu}(t) = [\delta(t), T_1(t), T_2(t)]^{\mathrm{T}}$ is the control input vector. Since the Jacobian matrix of (1) is bounded within the range of vehicle states, it can be deduced from the differential mean value theorem that f is Lipschitz continuous. The system (3) is discretized using the fourth-order Runge-Kutta method as follows:

$$
\boldsymbol{x}_{k+1} = f_s(\boldsymbol{x}_k, \boldsymbol{\mu}_k),
\tag{4}
$$

where $f_s(\cdot)$ denotes the discrete nominal vehicle dynamics.

Remark 1. *This paper focuses on establishing a data-driven neural network model capable of compensating for model residuals, thereby improving modeling accuracy while ensuring stability. Therefore, this paper focuses solely on the trajectory tracking control at the upper layer. At the lower layer, the control torque is divided into the total torque T_1 for the left four wheels and T_2 for the right four wheels, with the torque evenly distributed among the individual wheels.*

B. Problem Formulation

Assume that the residual of the vehicle model is additively incorporated into the dynamics model [39]. According to (3), the actual vehicle dynamics model with model residuals is established as follows:

$$
\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{\mu}(t)) + H_t\boldsymbol{\psi}(t),
\tag{5}
$$

where $\boldsymbol{\psi}(t) \in \mathrm{R}^{3\times1}$ represents the actual model residuals, $H_t = [\mathbf{0}_{3\times3}; \boldsymbol{I}_{3\times3}]$ extracts the dynamics components from the vehicle state $\boldsymbol{x}(t)$ as the dynamics characteristics of the vehicle are mainly influenced by $u(t)$, $v(t)$, and $\gamma(t)$. Let actual vehicle state vector $\boldsymbol{x}(t) = [\boldsymbol{x}_1(t), \boldsymbol{x}_2(t)]^{\mathrm{T}}$, where $\boldsymbol{x}_1(t) = [x_o(t), y(t), \varphi(t)]^{\mathrm{T}}$, and $\boldsymbol{x}_2(t) = [u(t), v(t), \gamma(t)]^{\mathrm{T}}$. In this paper, we consider only the model residuals that lie in the sub-state $\boldsymbol{x}_2(t)$. To obtain a more accurate vehicle dynamics model, the primary objective of this paper is to develop a data-driven neural network that estimate model residuals $\boldsymbol{\psi}(t)$ based on the collected vehicle states and control inputs. The discrete formulation of $\boldsymbol{\psi}(t)$ is provided in (8).

The desired trajectory $\boldsymbol{x}_d(t)$ consists of discrete-point trajectories that include the vehicle's target position and heading angle, longitudinal and lateral velocities, as well as velocities and accelerations in the inertial coordinate system, and is defined as follows:

$$\boldsymbol{x}_d(t) = [\boldsymbol{x}_{1d}(t), \boldsymbol{x}_{2d}(t), \boldsymbol{x}_{3d}(t), \boldsymbol{x}_{4d}(t)]^{\mathrm{T}}, \tag{6}$$

Specifically, $\boldsymbol{x}_{1d}(t) = [x_{od}(t), y_d(t), \varphi_d(t)]^{\mathrm{T}}$ represents the desired vehicle posture and $\boldsymbol{x}_{2d}(t) = [u_d(t), v_d(t), \gamma_d(t)]^{\mathrm{T}}$ represents the desired speed states and yaw rate in the vehicle body coordinate system. The $\boldsymbol{x}_{3d}(t) = [\dot{x}_{od}(t), \dot{y}_d(t), \dot{\varphi}_d(t)]$ is the derivative of $\boldsymbol{x}_{1d}(t)$. $\boldsymbol{x}_{4d}(t) = [\ddot{x}_{od}(t), \ddot{y}_d(t), \ddot{\varphi}_d(t)]$ is the derivative of $\boldsymbol{x}_{3d}(t)$. This trajectory is represented by discrete points sampled from a smooth curve that is continuously differentiable up to the second order. Additionally, curvature and lateral acceleration constraints are applied during trajectory generation. The regularity properties of the trajectory encompass two aspects: (1) the trajectory must be explicitly time-parameterized; and (2) the velocity and acceleration profiles must remain within physically admissible bounds, expressed as $\|x_{1d}(t) - x_{1d}(t-1)\|/\Delta t \le v_{max}$, $\|x_{2d}(t) - x_{2d}(t-1)\|/\Delta t \le a_{max}$. In this paper, the vehicle mainly tracks the position and heading of the trajectory points $\boldsymbol{x}_{1d}(t)$. The components $\boldsymbol{x}_{2d}(t)$, $\boldsymbol{x}_{3d}(t)$ and $\boldsymbol{x}_{4d}(t)$ are obtained by performing numerical differentiation on $\boldsymbol{x}_{1d}(t)$. Due to the varying requirements of different control algorithms, the tracking controller uses the $[\boldsymbol{x}_{1d}(t), \boldsymbol{x}_{2d}(t)]^{\mathrm{T}}$ as the reference trajectory. The backstepping controller uses the $[\boldsymbol{x}_{1d}(t), \boldsymbol{x}_{3d}(t), \boldsymbol{x}_{4d}(t)]^{\mathrm{T}}$ as the reference trajectory. The construction of trajectory is described in detail in Section VI.

Define the tracking error of the MWIDEV with respect to the given desired trajectory as follows:

$$\boldsymbol{x}_{1e}(t) := [x_{oe}(t), y_e(t), \varphi_e(t)]^{\mathrm{T}} = \boldsymbol{x}_1(t) - \boldsymbol{x}_{1d}(t). \tag{7}$$

The second goal of this paper is to design a control law u by developing a data-driven residual modeling and tracking control method that ensures the tracking error $\|\boldsymbol{x}_{1e}(t)\|$ converges to a small neighborhood around the origin.

Remark 2. *The residuals of the vehicle dynamics model in this paper mainly arise from unmodeled dynamics and model mismatches. To ensure computational feasibility, the nominal model incorporates several simplifications of vehicle dynamics, such as the small steering angle approximation and the linear tire model. These assumptions result in unmodeled dynamics and model mismatches, which depend on the vehicle's states and control inputs. Since disturbances vary across different scenarios, it is challenging to use a neural network to accurately estimate them in each scenarios. Therefore, our network does not take into account the impact of external disturbances and noise. Consequently, the model residual is denoted as* $\boldsymbol{\psi}(t) = \xi(\boldsymbol{x}(t), \boldsymbol{\mu}(t))$.

C. Overall Control Structure

The overall control structure is shown in Fig. 3. The data-driven modeling section involves the offline training of a temporal neural network model, aimed at predicting model residuals. This temporal neural network, composed of LSTM layers and fully connected (FC) layers, is referred to as the LSFC residual network. Prior to training, various vehicle driving cycles are configured, enabling the acquisition of state evolution data from both the high-fidelity vehicle simulation and the nominal vehicle dynamics model during the execution of these driving cycles. The LSFC residual network acquires the capability to predict residuals through offline training on residual data obtained from the comparison between the vehicle and nominal models. We incorporate the trained LSFC residual model into an LMPC algorithm for trajectory tracking control, referring to it as LSFC-LMPC. In tracking part of LSFC-LMPC, the prediction model combines the nominal model and trained LSFC residual model to obtain a more accurate predicted vehicle state. The backstepping controller is designed to calculate the Lyapunov constraints for LMPC. Based on the error between the current position of the vehicle and the reference trajectory, combined with the Lyapunov constraints, an optimization problem is formulated and solved to obtain the control input of the vehicle through receding horizon optimization.

IV. Data-driven Residual Modeling of MWIDEV

In this section, we introduce the modeling approach of the data-driven residual network. This network takes into account both the temporal feature extraction capabilities of LSTM and the strong fitting capabilities of fully connected neural networks, thereby improving prediction accuracy. This section begins by explaining the method used to acquire the dataset, followed by an illustration of the structure of the residual network.

A. Dataset Acquisition

During dataset acquisition, the vehicle driving cycle which includes steering angle and torque should be configured first. Note that the driving cycle should cover the entire operating range of the vehicle as comprehensively as possible. As shown in Fig. 5, the configured driving cycle is continuous. To facilitate dataset acquisition, the continuous driving cycle should be discretized. The discrete vehicle driving cycle consists of a series of control inputs $U_c = [\boldsymbol{\mu}_{c,1}, \boldsymbol{\mu}_{c,2}, \cdots, \boldsymbol{\mu}_{c,m}]^{\mathrm{T}}$, where $\boldsymbol{\mu}_{c,k} = [\delta_{c,k}, T_{1c,k}, T_{2c,k}]^{\mathrm{T}}$, for $(0 \le k \le m)$, and m denotes the length of driving cycle. Once the inputs $\boldsymbol{\mu}_{c,k}$ are configured, they are fed into both the high-fidelity simulation

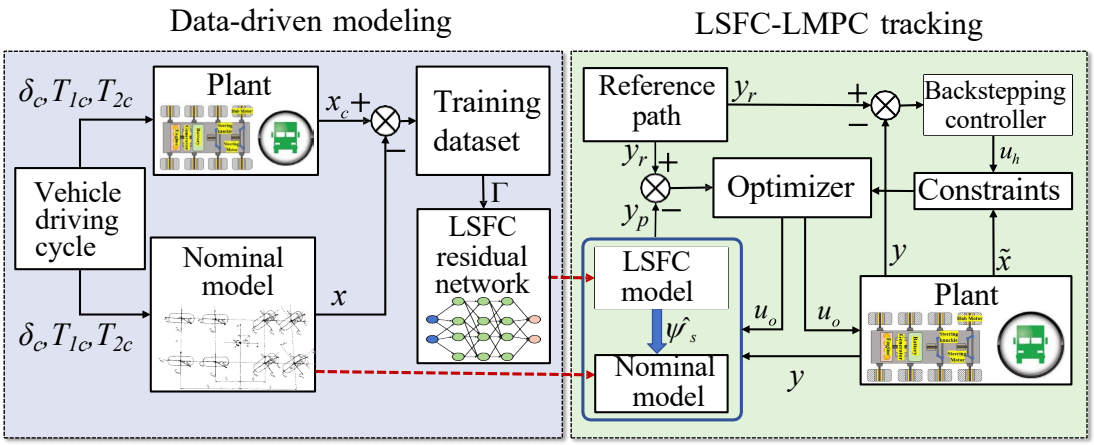


Fig. 3. The control structure. The Data-driven modeling part is used to train the LSFC residual network offline. In LSFC-LMPC tracking part, the LSFC residual network is applied to the LSFC-LMPC for trajectory tracking control.

vehicle and the nominal vehicle dynamics model to elicit their responses $\boldsymbol{x}_{c,k}$ and $f_s(\boldsymbol{x}_{c,k-1}, \boldsymbol{\mu}_{c,k-1})$, respectively. According to (5), the model residuals are expressed as follows:

$$\boldsymbol{\psi}_{c,k} = H_t^{\dagger}(\boldsymbol{x}_{c,k} - f_s(\boldsymbol{x}_{c,k-1}, \boldsymbol{\mu}_{c,k-1})), \tag{8}$$

where the $H_t^{\dagger}$ is the Moore-Penrose pseudo-inverse of H_t. The training dataset consisting of m samples can be constructed as follows:

$$\Gamma = [X_c, U_c, \Psi_c], \tag{9}$$

where $X_c = [\boldsymbol{x}_{c,1}, \boldsymbol{x}_{c,2}, \ldots, \boldsymbol{x}_{c,m}]^{\mathrm{T}}$ represents the recorded vehicle states during driving cycle, and $\Psi_c = [\boldsymbol{\psi}_{c,1}, \boldsymbol{\psi}_{c,2}, \ldots, \boldsymbol{\psi}_{c,m}]^{\mathrm{T}}$ denotes the model residual results.

Define a sample input vector at the instance k of the residual network:

$$\boldsymbol{\pi}_k = [\pi_{k,1}, \pi_{k,2}, \cdots, \pi_{k,ns}]^{\mathrm{T}} = [\boldsymbol{x}_{c,k}, \boldsymbol{\mu}_{c,k}], \tag{10}$$

where ns represents the sequence length of an input sample, $\pi_{k,i} = [x_{c,k,i}, \mu_{c,k,i}]$, $i \in \{1, \cdots, ns\}$ denotes the i-th element in the input sample sequence $\boldsymbol{\pi}_k$, $x_{c,k,i}$ and $\mu_{c,k,i}$ are the i-th element in the sequence of $\boldsymbol{x}_{c,k}$ and $\boldsymbol{\mu}_{c,k}$, respectively. The training dataset can be divided into feature data X_f and labeled data $\boldsymbol{y}_l$. The training set Γ can be alternatively represented as:

$$\Gamma = [X_f, \boldsymbol{y}_l], \tag{11}$$

where $X_f := [\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_m]^{\mathrm{T}} = [X_c, U_c]$, and $\boldsymbol{y}_l = \Psi_c$.

B. Data-driven Residual Network

According to (5), the vehicle dynamics model employed by the LSFC residual network is given by:

$$\dot{\tilde{\boldsymbol{x}}}(t) = f(\tilde{\boldsymbol{x}}(t), \boldsymbol{\mu}(t)) + H_t \hat{\boldsymbol{\psi}}(t), \tag{12}$$

where the $\hat{\boldsymbol{\psi}}(t)$ denotes the outputs of the LSFC residual network, $\tilde{\boldsymbol{x}}(t)$ denotes the states vector of the nominal model with model residual predictions. To facilitate the prediction of residual data, we define the discrete form of (12) as follows:

$$\tilde{\boldsymbol{x}}_{k+1} = f_s(\tilde{\boldsymbol{x}}_k, \boldsymbol{\mu}_k) + H_t \hat{\boldsymbol{\psi}}_{s,k}, \tag{13}$$

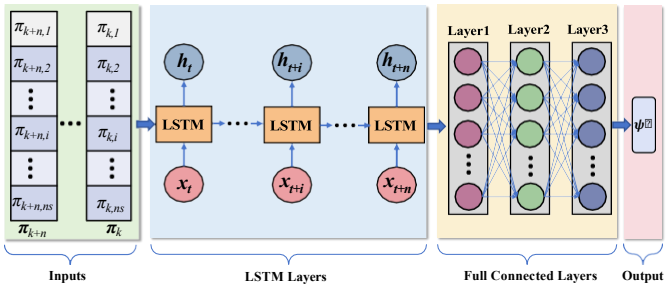


Fig. 4. The LSFC residual network architecture. This network consists of 1 LSTM layers and 3 FC layers.

where $\hat{\boldsymbol{\psi}}_{s,k}$ is the discrete output of the LSFC residual network.

The LSFC residual network shown in Fig. 4 consists of three parts, the input parts, LSTM layers parts [41] and FC layers parts. The LSTM layers predict the residual based on the state and control inputs over a past time window, capturing the relationship between the vehicle state and control inputs over time. The FC layers map the output of the LSTM layers to the required dimensions. Additionally, it introduces nonlinear transformation to enhance the model's fitting ability.

We use the mean squared error to calculate the loss function for error backpropagation. The training dataset Γ is divided into appropriate batches and arranged sequentially over time before being input into the residual model for offline training. After error backpropagation through the neural network, the network parameters are optimized, resulting a decrease in the loss function to decrease. Finally, the trained LSFC residual network is obtained.

Remark 3. *In this paper, since we only consider the model residuals of the velocity state variables, errors in position and heading angle are excluded from the residual calculation. There are mainly three reasons: (1) the velocity state variables are key parameters that connect vehicle dynamics and kinematics, and their accuracy significantly impacts the model's prediction accuracy; (2) factors contributing to model*

uncertainty (such as simplified tire models and the small-angle assumption for front-wheel steering) directly influence changes in the velocity state variables, and (3) considering only the residuals of the velocity state variables can reduce the dimensionality of neural network predictions and improve the model's prediction speed.

V. LSFC-LMPC Trajectory Tracking Control

In this section, we first present the derivation of the backstepping controller, a pivotal element serving as the Lyapunov constraint in the model predictive control architecture. Next, we propose the LSFC-LMPC, which synergistically combines the data-driven LSFC residual network with the backstepping-based Lyapunov constraint. A detailed proof of stability for the LSFC-LMPC technique is provided in Appendix.

A. Backstepping Controller Design

To derive the BSC, the nominal vehicle dynamics model is reformulated based on (1) and (5) as follows:

$$\begin{aligned} \dot{\boldsymbol{x}}_1(t) &= f_1(\boldsymbol{x}_1(t)) + g_1(\boldsymbol{x}_1(t))\boldsymbol{x}_2(t), \\ \dot{\boldsymbol{x}}_2(t) &= f_2(\boldsymbol{x}(t)) + g_2(\boldsymbol{x}(t))\boldsymbol{\mu}_h(t), \end{aligned} \tag{14}$$

where $g_1(\boldsymbol{x}_1(t))$ is an orthogonal matrix. According to the nominal MWIDEVs dynamics model, the tracking error of the BSC is $\boldsymbol{\alpha}_1(t) = \boldsymbol{x}_1(t) - \boldsymbol{x}_{1d}(t)$ with its derivative expressed as $\dot{\boldsymbol{\alpha}}_1(t) = \dot{\boldsymbol{x}}_1(t) - \dot{\boldsymbol{x}}_{1d}(t)$. The purpose of backstepping control is to make the vehicle follow the trajectory under the control input $\boldsymbol{\mu}_h(t) = h(\boldsymbol{x}(t))$. Here, we derive the BSC for MWIDEVs. The first Lyapunov function is defined as follows:

$$V_1(\boldsymbol{\alpha}_1(t)) = \frac{1}{2}\boldsymbol{\alpha}_1^{\mathrm{T}}(t)\boldsymbol{\alpha}_1(t), \tag{15}$$

Taking the derivative of the first Lyapunov function, we obtain:

$$\begin{aligned} \dot{V}_1(\boldsymbol{\alpha}_1(t)) &= \boldsymbol{\alpha}_1^{\mathrm{T}}(t)\dot{\boldsymbol{\alpha}}_1(t) \\ &= \boldsymbol{\alpha}_1^{\mathrm{T}}(t)(f_1(\boldsymbol{x}_1(t)) + g_1(\boldsymbol{x}_1(t))\boldsymbol{x}_2(t) - \dot{\boldsymbol{x}}_{1d}(t)). \end{aligned} \tag{16}$$

In order to ensure that $\dot{V}_1(\boldsymbol{\alpha}_1(t)) < 0$, we set the virtual tracking target $\boldsymbol{x}_{2tg}$ for $\boldsymbol{x}_2$ as follows:

$$\boldsymbol{x}_{2tg}(t) = g_1^{\mathrm{T}}(\boldsymbol{x}_1(t))(-f_1(\boldsymbol{x}_1(t)) + \dot{\boldsymbol{x}}_{1d}(t) - P_1\boldsymbol{\alpha}_1(t)), \tag{17}$$

where P_1 is a positive definite symmetric matrix. By defining the second tracking error as $\boldsymbol{\alpha}_2(t) = \boldsymbol{x}_2(t) - \boldsymbol{x}_{2tg}(t)$, the second Lyapunov function is given by:

$$V(\boldsymbol{\alpha}_1(t), \boldsymbol{\alpha}_2(t)) = V_1(\boldsymbol{\alpha}_1(t)) + \frac{1}{2}\boldsymbol{\alpha}_2^{\mathrm{T}}\boldsymbol{\alpha}_2. \tag{18}$$

Taking the derivative of the Lyapunov function, we obtain:

$$\begin{aligned} \dot{V}(\boldsymbol{\alpha}_1(t), \boldsymbol{\alpha}_2(t)) = \boldsymbol{\alpha}_1(t)^{\mathrm{T}}(&f_1(\boldsymbol{x}_1(t)) + g_1(\boldsymbol{x}_1(t))\boldsymbol{x}_2(t) \\ &- \dot{\boldsymbol{x}}_{1d}(t)) + \boldsymbol{\alpha}_2(t)^{\mathrm{T}}\dot{\boldsymbol{\alpha}}_2(t), \end{aligned} \tag{19}$$

Considering that $\boldsymbol{x}_2(t) = \boldsymbol{\alpha}_2(t) + g^{-1}(\boldsymbol{x}_1(t))(-f_1(\boldsymbol{x}_1(t)) + \dot{\boldsymbol{x}}_{1d}(t) - P_1\boldsymbol{\alpha}_1(t))$, the (19) is rewritten as:

$$\begin{aligned} \dot{V}(\boldsymbol{\alpha}_1(t), \boldsymbol{\alpha}_2(t)) = &-\boldsymbol{\alpha}_1(t)^{\mathrm{T}}P_1\boldsymbol{\alpha}_1(t) \\ &+ \boldsymbol{\alpha}_1(t)^{\mathrm{T}}g_1(\boldsymbol{x}_1(t))\boldsymbol{\alpha}_2(t) + \boldsymbol{\alpha}_2(t)^{\mathrm{T}}\dot{\boldsymbol{\alpha}}_2(t). \end{aligned} \tag{20}$$

In order to guarantee $\dot{V}(\boldsymbol{\alpha}_1(t), \boldsymbol{\alpha}_2(t)) < 0$, it is required that:

$$g_1(\boldsymbol{x}_1(t))\boldsymbol{\alpha}_1(t) + \dot{\boldsymbol{\alpha}}_2(t) = -P_2\boldsymbol{\alpha}_2(t), \tag{21}$$

where P_2 is a positive definite symmetric matrix. Since $\dot{\boldsymbol{\alpha}}_2(t) = f_2(\boldsymbol{x}(t)) + g_2(\boldsymbol{x}(t))\boldsymbol{\mu}_h(t) - \dot{\boldsymbol{x}}_{2d}(t)$, $\dot{\boldsymbol{x}}_{1d}(t) = \boldsymbol{x}_{3d}(t)$, $\ddot{\boldsymbol{x}}_{1d}(t) = \boldsymbol{x}_{4d}(t)$, the control law is defined as follows:

$$\begin{aligned} h(\boldsymbol{x}(t)) = &\, g_2(\boldsymbol{x}(t))^{-1}(-f_2(\boldsymbol{x}(t)) + \dot{g}_1^{-1}(\boldsymbol{x}_1(t))(-f_1(\boldsymbol{x}_1(t)) \\ &+ \boldsymbol{x}_{3d}(t) - P_1\boldsymbol{\alpha}_1(t)) + g_1^{-1}(\boldsymbol{x}_1(t))(-\dot{f}_1(\boldsymbol{x}_1(t)) \\ &+ \boldsymbol{x}_{4d}(t) - P_1\dot{\boldsymbol{\alpha}}_1(t)) - g_1(\boldsymbol{x}_1(t))\boldsymbol{\alpha}_1(t) - P_2\boldsymbol{\alpha}_2(t)). \end{aligned} \tag{22}$$

Although some derivatives are employed in the above control law, these derivatives are either directly accessible or can be easily computed without an estimation process. According to the derivation process of the backstepping controller, the closed-loop system under the action of the control input $\boldsymbol{\mu}_h$ can be asymptotically stabilized by appropriately choosing the values of P_1 and P_2. This backstepping controller will be used to design the LSFC-LMPC controller, whose stability analysis is provided in the Appendix.

B. LSFC-LMPC Design

The LSFC-LMPC control algorithm, which takes into account model residuals, is formulated as the following finite-horizon constrained optimal control problem:

$$\min_{\boldsymbol{\mu}(t)\in S(\Delta)} \int_{t_k}^{t_{k+N_p}} [(\tilde{\boldsymbol{x}}(\tau) - \boldsymbol{x}_d(\tau))^{\mathrm{T}}Q_c(\tilde{\boldsymbol{x}}(\tau) - \boldsymbol{x}_d(\tau)) + \boldsymbol{\mu}(\tau)^{\mathrm{T}}R_c\boldsymbol{\mu}(\tau)]\, d\tau, \tag{23a}$$

$$\text{s.t.} \quad \dot{\tilde{\boldsymbol{x}}}(t) = f(\tilde{\boldsymbol{x}}(t), \boldsymbol{\mu}(t)) + H_t\hat{\psi}(t), \tag{23b}$$

$$\tilde{\boldsymbol{x}}(t_k) = \boldsymbol{x}(t_k), \tag{23c}$$

$$\boldsymbol{\mu}_{\min} \le \boldsymbol{\mu}(t) \le \boldsymbol{\mu}_{\max}, \tag{23d}$$

$$\Delta\boldsymbol{\mu}_{T,\min} \le \Delta\boldsymbol{\mu}_T(t) \le \Delta\boldsymbol{\mu}_{\max}, \tag{23e}$$

$$\frac{\partial V}{\partial \boldsymbol{x}}(f(\tilde{\boldsymbol{x}}(t_k), \boldsymbol{\mu}(t_k))) \le \frac{\partial V}{\partial \boldsymbol{x}}f(\hat{\boldsymbol{x}}(t_k), h(\hat{\boldsymbol{x}}(t_k))), \tag{23f}$$

where Δ is the sampling step, $t \in [t_k, t_{k+N_p})$, $S(\Delta)$ denotes the family of piece-wise constant functions, $\tilde{\boldsymbol{x}}(t)$ is the predicted state of the compensated dynamics model by LSFC residual network, $\hat{\boldsymbol{x}}(t)$ is the nominal state of backstepping controller, N_p represents the prediction horizon, the weighting matrices Q_c and R_c are both positive definite, $\boldsymbol{\mu}_{max}$ and $\boldsymbol{\mu}_{min}$ are the maximum and minimum values of $\boldsymbol{\mu}(t)$, respectively. The control input is discretized as a piece-wise constant function by $\boldsymbol{\mu}(t) \in S(\Delta)$. The purpose is to transform the continuous optimization problem into a discrete optimization problem with a finite number of variables, facilitating its implementation on real hardware. The torque difference between the left and right sides is $\Delta\boldsymbol{\mu}_T(t) = \boldsymbol{\mu}_2(t) - \boldsymbol{\mu}_3(t)$, where $\boldsymbol{\mu}_2(t)$ and $\boldsymbol{\mu}_3(t)$ represent the torque inputs to the left and right sides, respectively. $\Delta\boldsymbol{\mu}_{max}$ and $\Delta\boldsymbol{\mu}_{min}$ denote the maximum and minimum torque differences between the left and right sides, respectively. This constraint is used to limit the torque difference between the left and right sides, thereby preventing excessive differential steering. The optimization objective

function contains two quadratic terms: the first term represents the vehicle's trajectory tracking error, and the second term corresponds to the control input cost.

The Lyapunov constraint (23f) is derived by evaluating the backstepping controller $h(\hat{\boldsymbol{x}}(t_k))$. Although this constraint can adjust the control input $\boldsymbol{\mu}(t_k)$ to stabilize the system, it may render the optimization problem infeasible if the adjusted input violates constraint (23d). Therefore, when designing the backstepping control $h(\hat{\boldsymbol{x}}(t_k))$, it is essential to ensure that it is within constraint (23d), which in turn guarantees the feasibility of (23a).

Remark 4. *In LSFC-LMPC, the $\hat{\psi}(t)$ predicted by the LSFC residual network needs to be compensated into the prediction model to improve the accuracy of the dynamics model. Since the dynamics model needs to predict N_s steps at each optimization, ideally the LSFC residual network also predicts N_s steps ahead and compensates the predictions into the dynamics model at each sampling instant. However, we choose to use only the first-step predictions $\hat{\psi}(t)$ from our dynamics model to ensure high accuracy, for two reasons. First, within the prediction time horizon, the accuracy of the states predicted by MPC decreases as time progresses. The residuals predicted based on these inaccurate state predictions are therefore unreliable as well. However, the first-step prediction is based on the actual, measured vehicle state, making its prediction residual more accurate. Second, if the LSFC prediction residual were applied at each step within the MPC prediction horizon, it could increase computation time and degrade the real-time performance of the algorithm. Therefore, using only the first-step prediction can reduce computation time while maintaining the accuracy of the prediction residual, making it more suitable for real-world applications.*

VI. Simulation Results

In this section, we analyze the proposed data-driven residual network and LSFC-LMPC strategy through simulations. The control subject for the simulation is an MWIDEV with eight hub motors, as shown in Fig. 1. To facilitate the simulation, we establish the simulated vehicle shown in Fig. 2 within the Trucksim software as a real vehicle, and the basic parameters of the vehicle are listed in Table I. Our desktop computer is equipped with an i7-1260P CPU, 40 GB of memory, and an integrated graphics card. The simulation setup consists of three parts, the first is the validation of the prediction accuracy of the LSFC residual network, the second is the validation of the LSFC-LMPC trajectory tracking performance, and the last is the validation of different temporal residual networks.

A. Performance Evaluation of LSFC Residual Network

This subsection focuses on the simulation and the validation of data-driven residual networks. The dataset Γ used for training and validation is obtained by designing suitable driving cycles, collecting data from Trucksim and nominal vehicle model, and computing (8). The driving cycle used for data acquisition is shown in Fig. 5. In this cycle, different control inputs are considered, including the case where the left and right sides input torques are the same or different, as well as variations in steering. The total driving cycle time is 360 seconds, with a total travel distance of 2,642 meters and a maximum vehicle speed of 44.34 km/h. The driving scenario is a well-adhered, flat and wide area, and the steering and torque signals of the vehicle are both sine signals. The entire torque input curve is divided into two cycles, each lasting 180 seconds (Cycle 1: 0-180 s; Cycle 2: 180-360 s). Each 180-second cycle is further divided into 9 segments based on a 20-second time step. During each 20-second segment, torque is applied in the first 10 seconds, followed by no torque input in the next 10 seconds. The purpose of this design is to prevent the vehicle's speed from increasing too rapidly and causing instability due to prolonged torque input. In the steering input diagram, the input is divided into two distinct 180-second cycles (Cycle 1: 0-180 s; Cycle 2: 180-360 s). The difference between the two cycles lies in the frequency and amplitude of the steering input.

TABLE I
The parameters of MWIDEV

Parameters	Value	Unit	Note
m	2.30×10^3	kg	The mass of MWIDEV
L	1.60	m	The wheelbase of MWIDEV
l_1, l_4	3.00	m	Center of mass to the 1st, 4th axis distance
l_2, l_3	1.00	m	Center of mass to the 2nd, 3rd axis distance
C_i	3.51×10^5	N/rad	Tire cornering stiffness of i-th axis
I_z	4235.80	kg · m^2	The moment of inertia of MWIDEV
C_D	0.60	−−	Atmospheric drag coefficient
ρ_a	1.20	kg/m^2	Atmospheric density
A	1.84	kg/m^2	Windward area of MWIDEV
η_T	0.95	−−	Drivetrain mechanical efficiency
r_w	0.38	m	Tire radius
g	9.8	m/s^2	Gravitational acceleration
f_r	1	−−	Rolling resistance coefficient

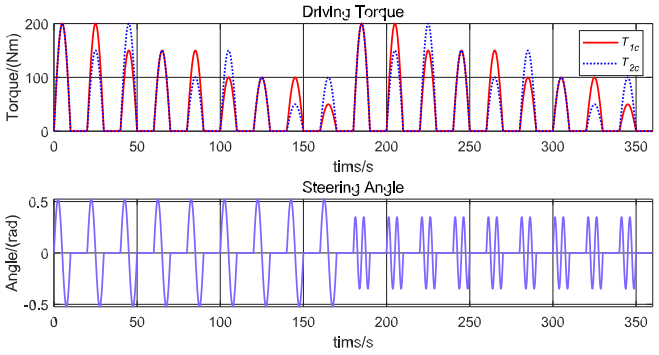


Fig. 5. The driving cycle of MWIDEV for data acquisition.

This dataset is then used to train the LSFC residual network. The trainings parameters of the residual network are illustrated in Table II. The size of input and output layers is consistent with that introduced in the theoretical part introduced of this paper. We choose a single-layer LSTM to fulfill the computational efficiency requirements. We set the dropout rate to 0.25 to ensure a balance between regularization strength and convergence speed during the training process. The dataset is divided into a training set, a validation set, and a test set, with the training set accounting for 70%, the validation set 15%, and the test set 15%. The patience epoch is set to 10 so that training will stop early if the validation performance does

TABLE II
THE TRAINING PARAMETERS OF LSFC RESIDUAL NETWORK

Parameters	Value	Note
Input size	6	Dimension of input features
Hidden size	128	Dimension of hidden state
Output size	3	Dimension of output vector
LSTM layers	1	LSTM stacked layers
FC layer 1 size	[128,64]	Input and output dimensions of the FC layer1
FC layer 2 size	[64,32]	Input and output dimensions of the FC layer2
FC layer 3 size	[32,3]	Input and output dimensions of the FC layer3
Dropout rate	0.25	Dropout rate of LSTM layers
Sequence length	20	Sequence length of LSTM layers
Training dataset ratio	70%	Proportion of the training dataset to the Γ
Validation dataset ratio	15%	Proportion of the validation dataset to the Γ
Test dataset ratio	15%	Proportion of the test dataset to the Γ
Batch size	64	Batch size of training
Learning rate	0.001	Learning rate of training
Max epoch	1000	Max epoch of training
Patience epoch	10	Number of training epochs without raising

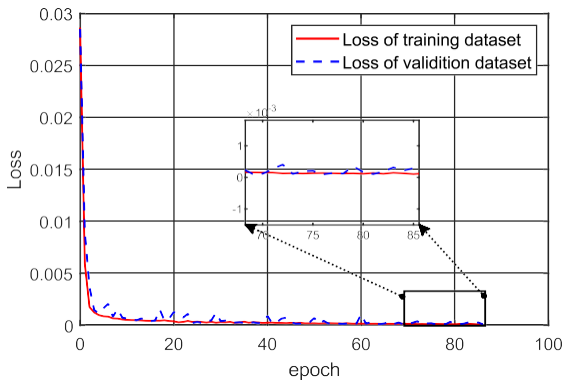


Fig. 6. The loss of training dataset and validation dataset of LSFC residual network.

TABLE III
THE TRAINING PARAMETERS OF DNN-BASED RESIDUAL NETWORK

Parameters	Value	Note
Input size	[6, 1]	Dimension of input features
Output size	[3,1]	Dimension of output vector
FC layer 1 size	[6, 128]	Input and output dimensions of the FC layer 1
FC layer 2 size	[128, 64]	Input and output dimensions of the FC layer 2
FC layer 3 size	[64, 32]	Input and output dimensions of the FC layer 3
FC layer 4 size	[32, 3]	Input and output dimensions of the FC layer 4
Training dataset ratio	70%	Proportion of the training dataset to the Γ
Validation dataset ratio	15%	Proportion of the validation dataset to the Γ
Test dataset ratio	15%	Proportion of the test dataset to the Γ
Batch size	64	Batch size of Training
Learning rate	0.001	Learning rate of Training
Max epoch	1000	Max epoch of Training
Patience epoch	10	Number of training epochs without raising

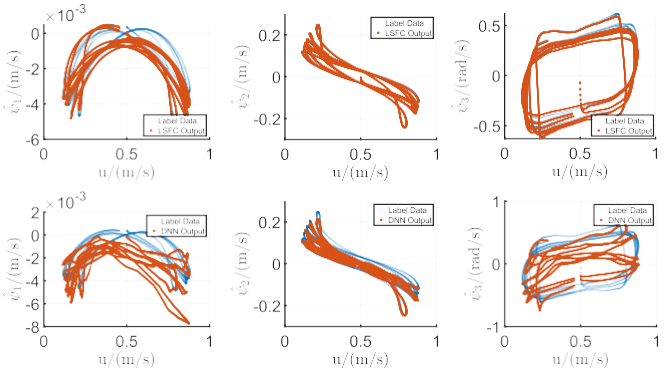


Fig. 7. The outputs of LSFC and DNN-based residual networks for test dataset.

not show significant improvement for 10 consecutive epochs. Based on this parameter, we set the maximum number of epochs to 1000. In practice, the network will stop training before reaching the maximum epoch count. During the pre-processing of the dataset, to avoid the instability in the gradient update direction caused by the differences in feature scales, we performed Z-Score normalization on the dataset. In order to preserve the time series information, the dataset is not shuffled during training. To speed up training, we set the batch size to 64. In order to validate the effectiveness and generalization ability, we compared the loss function values of the training and validation sets of the LSFC residual network in Fig. 6. We observe that as the training epoch increases, the losses of the training dataset and the validation dataset gradually decreases and converges to 0. Since the validation set is the dataset that has never been trained, it can be inferred that the model has acquired a certain degree of generalization ability.

To validate the effectiveness of the proposed LSFC residual network, we establish a regular DNN-based residual network for comparison. The DNN neural network, which is composed of multiple fully connected layers is a classic non-temporal neural network, and is more suitable for comparison with temporal neural networks. The parameters of the DNN-based residual network are made as consistent as possible with the LSFC residual network, as shown in Table III. Based on the parameters in Table II and Table III, we train the LSFC network and the DNN network on the same dataset. In order to fully squeeze the training capacity of the model, the maximum epoch is set to 1000, and the training is stopped when the effect is no longer improved after 10 (patient epoch) consecutive trainings. The results of the two residual networks on the training dataset are shown in Fig. 7. For ease of observation, we set the horizontal axis to the first dimension u of the input feature vector. The vertical axis represents the prediction results of the residual network, which are u_{out}, v_{out}, and r_{out}, respectively. As can be seen from Fig. 7, the LSFC residual network has better prediction results than the DNN-based residual network.

Due to the large differences in the magnitude of the output data, the mean absolute percentage error (MAPE) in (24) is used to measure the quality of the model prediction results. The MAPE is defined as follows:

$$E_m = \frac{1}{n_t}\sum_{i=1}^{n_t}\left|\frac{\hat{\psi}_i - \psi_i}{\psi_i}\right| \times 100\% \tag{24}$$

where n_t denotes the number of samples in the test dataset, $\hat{\psi}_i$ and ψ_i represent the predicted data by the residual network and labels of the test dataset, respectively. It can be observed that the smaller values of MAPE represent the higher accuracy of the model. The comparison of MAPE values of the two residual networks is shown in Fig. 8. It can be seen that all the methods can predict the vehicle's motion state well from 0 to 4 seconds, as there is no steering at this time and the vehicle operating conditions are relatively simple. However, starting from the 4 seconds, the model-based and DNN compensation

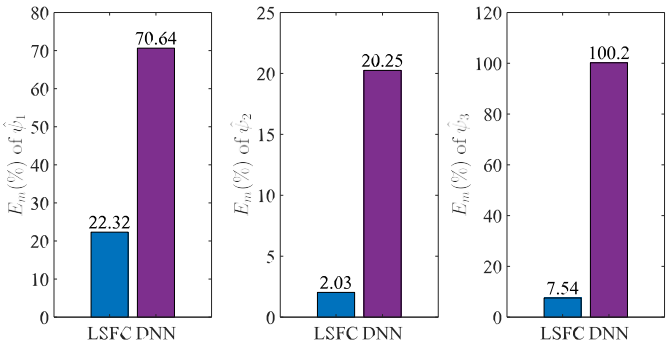


Fig. 8. The MAPE of LSFC and DNN-based residual networks for test dataset.

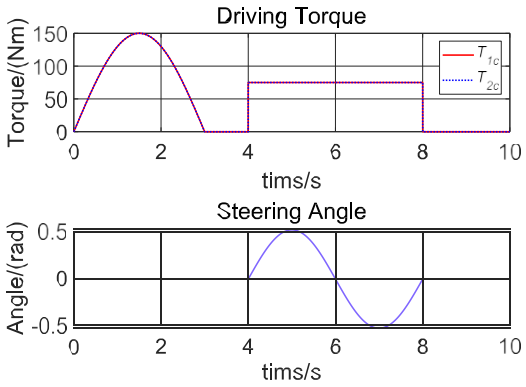


Fig. 9. The testing driving cycle for model validation.

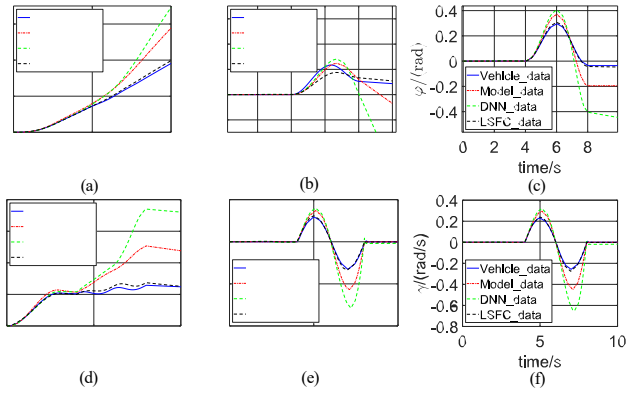


Fig. 10. Comparison of open-loop simulation results for different models.

methods have a relatively large gap from the true value because they cannot capture the nonlinear changes of the vehicle after steering angles are applied. However, the method based on LSFC residual network compensation uses a past time window for prediction, making it more robust.

Next, we will verify the effect of the output from the proposed LSFC residual network after compensation to the nominal vehicle dynamics model. We establish four models: the real vehicle dynamics model, the nominal vehicle dynamics model, the DNN-based compensation model, and the LSFC compensation model. We choose DNN as the baseline for two reasons. On one hand, DNN is a classic regression network that has been widely validated in regression tasks. On the other hand, this can eliminate the impact of "architectural differences" on performance and allows us to directly verify the effectiveness of the residual design and temporal modeling components. Note that the actual vehicle dynamics model is replaced by a model built in Trucksim software, with vehicle parameters shown in Table I. In Fig. 9, we build the vehicle testing driving cycle with sinusoidal torque input and sinusoidal steering angle input as the control inputs of the vehicle. In order to test the generalization performance of the LSFC network, this testing driving cycle differs from the driving cycle in Fig. 5. The differences between the test driving cycle and the training driving cycle lie in the different input frequencies, amplitudes, and different combinations of steering and torque. Our method demonstrates a certain degree of robustness in scenarios involving slight out-of-distribution shifts.

The state evolution trajectories of the four aforementioned models under the testing driving cycle input are shown in Fig. 10. From the simulation results, we find that the LSFC model achieves the lowest prediction error compared to the DNN model and the nominal model. The Model_data denotes the response nominal vehicle dynamics model. At the beginning of the test, the DNN model and the nominal model achieve similar prediction accuracy during the longitudinal acceleration process. In the second half of the test, due to cumulative errors caused by inaccurate predictions of the DNN-based residual model, the response of the DNN-based compensation model deviates further from the actual vehicle dynamics model than that of the nominal model. To quantify the deviation of the model responses, we calculated the Root Mean Square Error (RMSE) of the different models' responses relative to the response of the actual vehicle dynamics model, as shown in Fig. 11. Here, "LSFC" represents the RMSE of the open-loop trajectory generated by the LSFC-based compensation model, "DNN" represents the RMSE of the open-loop trajectory generated by the DNN-based compensation model, and "MODEL" represents the RMSE of the open-loop trajectory generated by the nominal vehicle dynamics model. From Fig. 10 and Fig. 11, we find that the response of LSFC compensation model is closer to the response of the real vehicle model.

B. Trajectory Tracking Performance Evaluation of LSFC-LMPC

This simulation is mainly used to verify the accuracy of different residual neural networks. The reference trajectory is set to a double lane change trajectory [42], following the international vehicle lateral test standard ISO 3888-1. In the simulation results, we use different terms to represent different methods: "LSFC-LMPC" represents the method proposed in this paper, which combines the LSFC residual network with MPC based on lyapunov constraints. "DNN-LMPC" as a comparative method, which combines the DNN residual network with MPC based on Lyapunov constraints. "LMPC" as a comparative method, which does not have a residual network

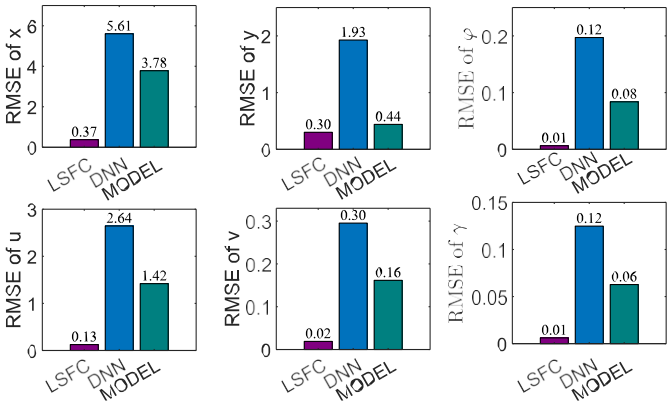


Fig. 11. RMSE of open-loop simulations for different models.

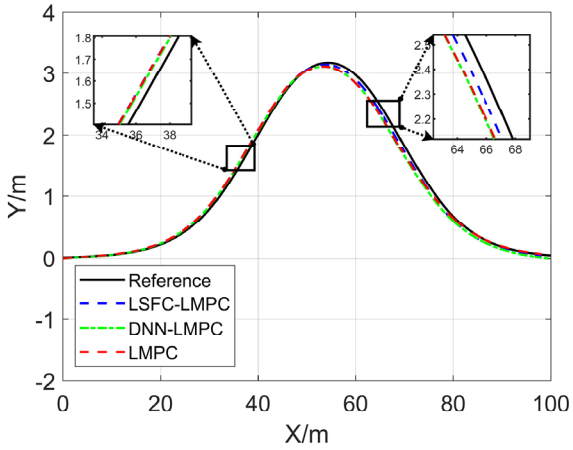


Fig. 12. The tracking trajectory of LSFC-LMPC, DNN-LMPC and LMPC algorithms.

and only retains the MPC based on lyapunov constraints. "LSFC-MPC" as a comparative method, which combines the LSFC residual network with the ordinary MPC algorithm without adding Lyapunov constraints. "MPC" as a comparative method, which adopts the ordinary MPC method without adding the LSFC residual network and Lyapunov constraints. To highlight the advantages of the algorithm, the MPC parameters Q_c, R_c and input constraints $u_{\min}$, $u_{\max}$, $\Delta u_{\min}$, $\Delta u_{\max}$ of the three algorithms are kept consistent across all three algorithms. The basic parameters of the MPC used in the three algorithms are shown in Table IV.

TABLE IV
THE PARAMETERS OF MPC

Parameters	Value	Unit	Note
$Q_c(1,1)$	8e3	−−	Row 1, column 1 of the error weight matrix
$Q_c(2,2)$	1e4	−−	Row 2, column 2 of the error weight matrix
$Q_c(3,3)$	3e4	−−	Row 3, column 3 of the error weight matrix
$Q_c(4,4)$	4e3	−−	Row 4, column 4 of the error weight matrix
$Q_c(5,5)$	0	−−	Row 5, column 5 of the error weight matrix
$Q_c(6,6)$	10	−−	Row 6, column 6 of the error weight matrix
$R_c(1,1)$	350	−−	Row 1, column 1 of the control weight matrix
$R_c(2,2)$	2e-5	−−	Row 2, column 2 of the control weight matrix
$R_c(3,3)$	2e-5	−−	Row 3, column 3 of the control weight matrix
δ_{max}	0.52	rad	Maximum front wheel angle constraint
δ_{min}	-0.52	rad	Minimum front wheel angle constraint
$T_{1,max}$, $T_{2,max}$	350	Nm	Maximum driving torque of T_1 and T_2
$T_{1,min}$, $T_{2,min}$	-350	Nm	Minimum driving torque of T_1 and T_2
Δu_{max}	50	Nm	Maximum torque difference of T_1 and T_2
Δu_{min}	-50	Nm	Minimum torque difference of T_1 and T_2

The tracking trajectories of the three algorithms are shown in Fig. 12. The LSFC-LMPC algorithm refers to the proposed algorithm that uses the LSFC compensation model and Lyapunov constraint. We find that the LSFC-LMPC tracking performs better than the other two algorithms. The DNN-LMPC algorithm results in the worst tracking performance due to the inaccurate estimation of vehicle dynamics residuals. In this comparison, the LMPC algorithm is already capable of enabling the vehicle to track the desired trajectory with a certain degree of accuracy. The RMSE of trajectory tracking for the three algorithms is shown in Table V.

The steering angle and torque inputs of the three algorithms are shown in Fig. 13 and Fig. 14. We find that the steering angle and torque inputs of the DNN-LMPC algorithm oscillate between 6.8 second and 8.4 second. This is because, during

TABLE V
THE RMSE OF TRAJECTORY TRACKING FOR HORIZON COMPARISON

Algorithm	LSFC-LMPC	DNN-LMPC	LMPC
RMSE/m	0.0448	0.0601	0.0496

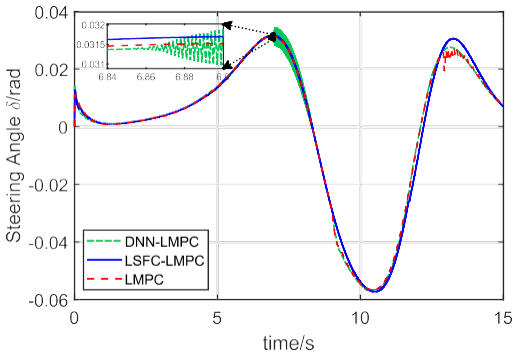


Fig. 13. The steering angle inputs for trajectory tracking of LSFC-LMPC, DNN-LMPC and LMPC algorithms.

this period, the Lyapunov constraint is not satisfied due to inaccurate DNN compensation. And the control inputs of the LMPC algorithm oscillate after 12.9 second because the Lyapunov constraint is not satisfied. The control inputs of the LSFC-LMPC algorithm satisfy the Lyapunov constraints and remain smooth. In terms of the trajectory tracking effects, there is no oscillation in LMPC and MPC method. The average prediction time of the LSFC residual network is 0.0085 seconds. The average computation time of LMPC is 0.0169 seconds.

In the second comparison simulation, we compare our proposed algorithm with LSFC-MPC, which removes the Lyapunov constraint, and MPC, which utilizes the nominal vehicle model without considering Lyapunov constraints. The trajectory tracking performance is shown in Fig. 14. The RMSE of trajectory tracking is shown in Table VI. Compared with LSFC-MPC and MPC algorithms, we find that the LSFC compensation model provides more accurate model residual

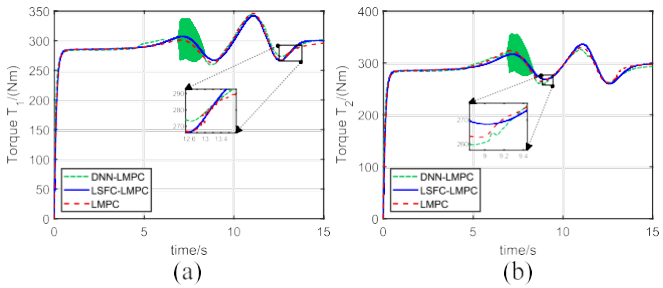


Fig. 14. The torque inputs for trajectory tracking of LSFC-LMPC, DNN-LMPC and LMPC algorithms. (a) is the torque input T_1. (b) is the torque input T_2.

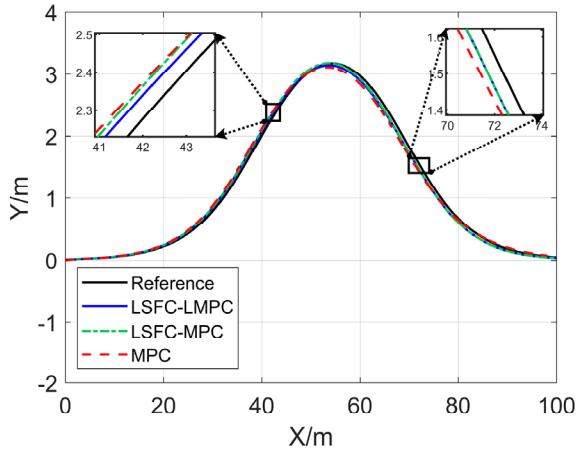


Fig. 15. The tracking trajectory of LSFC-LMPC, LSFC-MPC and MPC algorithms.

estimates, thereby improving trajectory tracking performance. The control inputs of the three algorithms are shown in Fig. 16 and Fig. 17.

TABLE VI
THE RMSE OF TRAJECTORY TRACKING FOR VERTICAL COMPARISON

Algorithm	LSFC-LMPC	LSFC-MPC	MPC
RMSE/m	0.0448	0.0460	0.0596

C. Comparison of different temporal neural networks in residual compensation

In this section, we added simulations and comparisons of temporal residual networks based on Gated Recurrent Unit (GRU) and Transformer. In the designing of the GRU-based residual network, we replace the LSTM layer in the LSFC residual network with a GRU layer to ensure the validity of the comparison. In the Transformer-based residual network, we designed a neural network with a GPT-like structure consisting of three parts: the input layer, the multi-head attention layer (MHA), and the feed-forward network (FFN).

The tracking control results of different temporal residual networks are shown in Fig 18. The RMSE of different temporal residual networks are shown in Table VII. It can be seen that the residual networks based on Transformer and GRU

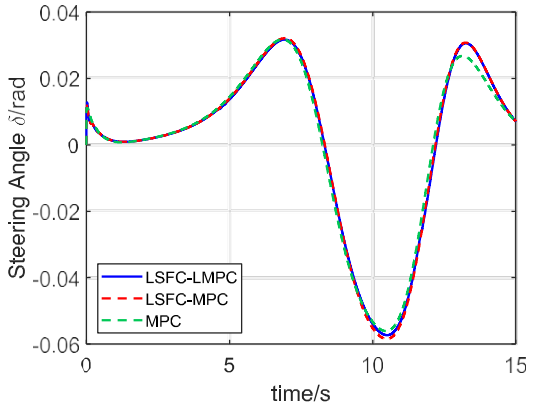


Fig. 16. The steering angle inputs for trajectory tracking of LSFC-LMPC, LSFC-MPC and MPC algorithms.

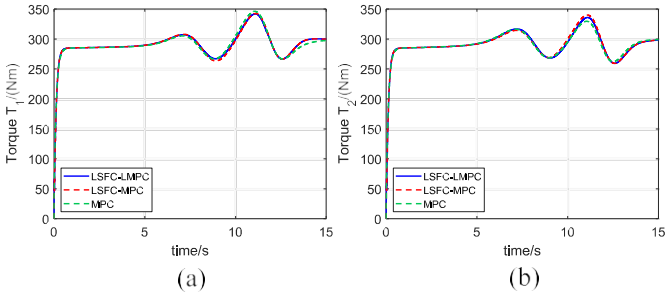


Fig. 17. The torque inputs for trajectory tracking of LSFC-LMPC, LSFC-MPC and MPC algorithms.

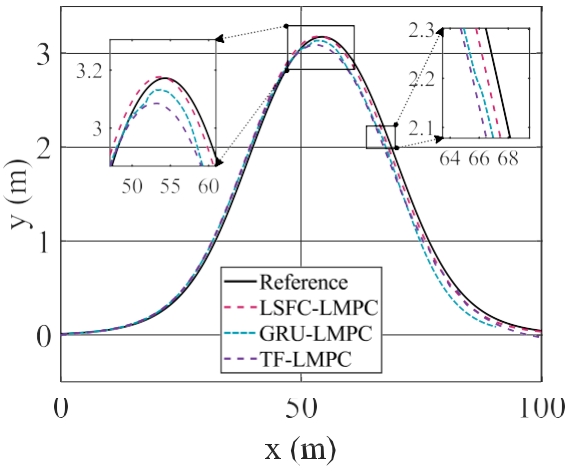


Fig. 18. The tracking control results of different temporal residual networks.

TABLE VII
THE RMSE OF TRAJECTORY TRACKING FOR DIFFERENT TEMPORAL RESIDUAL NETWORKS.

Algorithm	LSFC-LMPC	GRU-LMPC	TF-LMPC
RMSE/m	0.0448	0.0788	0.0735

achieve similar tracking accuracy in closed-loop simulations. The tracking accuracy of the LSFC residual network demonstrates higher tracking accuracy.

VII. Conclusion

This paper proposes an LSFC residual network and an LSFC-LMPC trajectory tracking controller. The LSFC residual network, consisting of LSTM layers and FC layers, captures the temporal characteristics of vehicle states and improves the accuracy of estimation. We compare the performance of LSFC residual network, DNN-based residual network, GRU-based residual network and Transformer-based residual network. LSFC demonstrates higher accuracy in both fitting performance and compensation model accuracy. In the LSFC-LMPC trajectory tracking controller, the BSC controller is configured to obtain the Lyapunov constraint for optimization. We compare the tracking effectiveness of LSFC-LMPC in both vertical and horizontal aspects, and the results show better tracking accuracy. At the same time, we provide a proof of the stability of LSFC-LMPC. This work improves the accuracy of modeling the dynamics of vehicles, especially for multi-wheel-independent-drive electric vehicles with a high degree of state coupling, through a data-driven approach. In this paper, the LSFC-LMPC demonstrates its superiority as an application of data-driven modeling. The model can be applied to more scenarios in the future, such as lateral stability control, torque distribution for MWIDEVs, and other related applications.

Appendix

The asymptotic stability of the backstepping controller indicates that there is a stabilization region for the closed-loop system under the control policy $\boldsymbol{\mu}_h = h(\boldsymbol{x})$. According to the converse Lyapunov theory, the Lyapunov function $V(\boldsymbol{x})$ of the closed-loop system in (18) satisfies the following equations [43]:

$$\beta_1(\|\boldsymbol{x}\|) \le V(\boldsymbol{x}) \le \beta_2(\|\boldsymbol{x}\|), \tag{25a}$$

$$\frac{\partial V}{\partial \boldsymbol{x}} f(\boldsymbol{x}, h(\boldsymbol{x})) \le -\beta_3(\|\boldsymbol{x}\|), \tag{25b}$$

$$\left\| \frac{\partial V}{\partial \boldsymbol{x}} \right\| \le \beta_4(\|\boldsymbol{x}\|), \tag{25c}$$

where the functions $\beta_i(\cdot), i = 1, 2, 3, 4$ belong to class $\mathcal{K}$. Denote that the $\Omega_\rho := \{\boldsymbol{x} \in \mathrm{X} | V(\boldsymbol{x}) \le \rho\}$ is the stability region of the closed-loop system under backstepping control law $h(\boldsymbol{x})$. Due that system $f(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\phi})$ possesses the continuous and local Lipschitz properties, and the Lyapunov function $V(\boldsymbol{x})$ is continuously differentiable, there exists a positive constant $M, L_x, L_\phi, \bar{L}_x, \bar{L}_\phi$ satisfying [44]:

$$\|f(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\phi})\| \le M, \tag{26a}$$

$$\|f(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\phi}) - f(\boldsymbol{x}', \boldsymbol{\mu}, \mathbf{0})\| \le L_x \|\boldsymbol{x} - \boldsymbol{x}'\| + L_\phi \|\boldsymbol{\phi}\|, \tag{26b}$$

$$\left\| \frac{\partial V}{\partial \boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\phi}) - \frac{\partial V}{\partial \boldsymbol{x}} f(\boldsymbol{x}', \boldsymbol{\mu}, \mathbf{0}) \right\| \le \bar{L}_x \|\boldsymbol{x} - \boldsymbol{x}'\| + \bar{L}_\phi \|\boldsymbol{\phi}\|, \tag{26c}$$

where the $\boldsymbol{x}, \boldsymbol{x}' \in \Omega_\rho$.

Proposition 1. *Given the nominal state trajectory $\hat{\boldsymbol{x}}(t)$ of system* (3) *generated by the backstepping control law $h(\boldsymbol{x})$ in* (22) *and assuming that there exist $P_1, P_2 \ge 0$ and $\rho > \rho_s > 0$, then if $\hat{\boldsymbol{x}}(t_k) \in \Omega_\rho / \Omega_{\rho_s}$ holds, the following inequality holds for any $k \ge 0$:*

$$V(\hat{\boldsymbol{x}}(t)) \le V(\hat{\boldsymbol{x}}(t_k)), \forall t \in [t_k, t_{k+1}]. \tag{27}$$

Proof. According to (15) and (18), the Lyapunov function of the backstepping control law for the nominal vehicle dynamics model is given by $V(\hat{\boldsymbol{x}}(t)) = V_1(\hat{\boldsymbol{x}}(t)) + \frac{1}{2}\boldsymbol{\alpha}_2^{\mathrm{T}}\boldsymbol{\alpha}_2$, where $V_1(\hat{\boldsymbol{x}}(t)) = \frac{1}{2}\boldsymbol{\alpha}_1^{\mathrm{T}}\boldsymbol{\alpha}_1$. It can be inferred that $V(\hat{\boldsymbol{x}}(t)) \ge 0$. Taking the derivative of $V(\hat{\boldsymbol{x}}(t))$ yields:

$$\dot{V}(\hat{\boldsymbol{x}}(t)) = \boldsymbol{\alpha}_1^{\mathrm{T}}(f_1(\hat{\boldsymbol{x}}_1(t)) + g_1(\hat{\boldsymbol{x}}_1(t))\hat{\boldsymbol{x}}_2(t) - \dot{\boldsymbol{x}}_{1d}(t)) + \boldsymbol{\alpha}_2^{\mathrm{T}}\dot{\boldsymbol{\alpha}}_2. \tag{28}$$

Let $\hat{\boldsymbol{x}}_2(t) = \boldsymbol{\alpha}_2 + g_1^{-1}(\hat{\boldsymbol{x}}_1(t))(-f_1(\hat{\boldsymbol{x}}_1(t)) + \dot{\boldsymbol{x}}_{1d}(t) - P_1\boldsymbol{\alpha}_1)$, then the derivation of $V(\hat{\boldsymbol{x}}(t))$ is rewritten as:

$$\dot{V}(\hat{\boldsymbol{x}}(t)) = -\boldsymbol{\alpha}_1^{\mathrm{T}}P_1\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_1^{\mathrm{T}}g_1(\hat{\boldsymbol{x}}_1)\boldsymbol{\alpha}_2 + \boldsymbol{\alpha}^{\mathrm{T}}\dot{\boldsymbol{\alpha}}_2. \tag{29}$$

Substituting the backstepping control law $h(\hat{\boldsymbol{x}}(t))$ into the above formula yields:

$$g_1(\hat{\boldsymbol{x}}_1(t))\boldsymbol{\alpha}_1 + \dot{\boldsymbol{\alpha}}_2 = -P_2\boldsymbol{\alpha}_2. \tag{30}$$

Due to $P_1, P_2 \ge 0$, we obtain:

$$\dot{V}(\hat{\boldsymbol{x}}(t)) = -\boldsymbol{\alpha}_1^{\mathrm{T}}P_1\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2^{\mathrm{T}}P_2\boldsymbol{\alpha}_2 \le 0. \tag{31}$$

Integrating the Lyapunov function in $[t_k, t_{k+1}]$, the (27) holds. ∎

Proposition 2. *Given the nominal state trajectory $\hat{\boldsymbol{x}}(t)$ of system* (3) *generated by backstepping control law $h(\boldsymbol{x})$ in* (22) *that satisfy* (26) *and* (27). *Let $\Delta, \epsilon_s, M > 0, \rho > \rho_s > 0$ satisfy:*

$$-\beta_3(\beta_2^{-1}(\rho_s)) + \bar{L}_x M \Delta \le -\epsilon_s/\Delta, \tag{32}$$

then, for any k, if $\hat{\boldsymbol{x}}(t_k) \in \Omega_\rho / \Omega_{\rho_s}$, the following inequality holds.

$$V(\hat{\boldsymbol{x}}(t_{k+1})) \le V(\hat{\boldsymbol{x}}(t_k)) - \epsilon_s. \tag{33}$$

Proof. Taking the time derivative of Lyapunov function $V(\hat{\boldsymbol{x}}(t))$ for the nominal system:

$$\dot{V}(\hat{\boldsymbol{x}}(t)) = \frac{\partial V}{\partial \boldsymbol{x}} f(\hat{\boldsymbol{x}}(t), h(\hat{\boldsymbol{x}}(t))), t \in [t_k, t_{k+1}]. \tag{34}$$

By adding and subtracting the derivative of Lyapunov function $\dot{V}(\hat{\boldsymbol{x}}(t_k))$, we obtain:

$$\begin{aligned}\dot{V}(\hat{\boldsymbol{x}}(t)) = &\frac{\partial V}{\partial \boldsymbol{x}} f(\hat{\boldsymbol{x}}(t_k), h(\hat{\boldsymbol{x}}(t_k))) + \frac{\partial V}{\partial \boldsymbol{x}} f(\hat{\boldsymbol{x}}(t), h(\hat{\boldsymbol{x}}(t))) \\ &- \frac{\partial V}{\partial \boldsymbol{x}} f(\hat{\boldsymbol{x}}(t_k), h(\hat{\boldsymbol{x}}(t_k))).\end{aligned} \tag{35}$$

Based on the local Lipschitz properties of the Lyapunov function in (25) and (26), the (35) can be rewritten as:

$$\dot{V}(\hat{\boldsymbol{x}}(t)) \le -\beta_3(\|\hat{\boldsymbol{x}}(t_k)\|) + L_x\|\hat{\boldsymbol{x}}(t) - \hat{\boldsymbol{x}}(t_k)\|, \tag{36}$$

according to (25)a, we have:

$$V(\hat{\boldsymbol{x}}(t)) \le \beta_2(\|\hat{\boldsymbol{x}}(t)\|). \tag{37}$$

Since the $\hat{\boldsymbol{x}}(t_k) \in \Omega_\rho / \Omega_{\rho_s}$, we have

$$\rho_s \le \beta_2(\|\hat{\boldsymbol{x}}(t)\|), \tag{38}$$

the following inequalities hold:

$$\begin{aligned}-\beta_3(\|\hat{\boldsymbol{x}}(t_k)\|) &\le -\beta_3(\beta_2^{-1}(\rho_s)), \\ \|\hat{\boldsymbol{x}}(t) - \hat{\boldsymbol{x}}(t_k)\| &\le M\Delta.\end{aligned} \tag{39}$$

Then, the (36) is rewritten as:

$$\dot{V}(\hat{\boldsymbol{x}}(t)) \leq -\beta_3(\beta_2^{-1}(\rho_s)) + \bar{L}_x M \Delta, \tag{40}$$

according to (32), the (40) can be rewritten as:

$$\dot{V}(\hat{\boldsymbol{x}}(t)) \leq -\epsilon_s/\Delta. \tag{41}$$

Integrating the (41) on $[t_k, t_{k+1}]$, the (33) holds [45].

■

Proposition 3. *Consider the open-loop state trajectory of system* (5) *under backstepping control law* $h(\boldsymbol{x})$ *starting at* $\boldsymbol{x}(t_0)$ *with model residuals* $\psi(t)$ *is obtained by solving the following equation:*

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), h(\hat{\boldsymbol{x}}(t))) + H_t\psi(t), t \in [t_0, t_1], \tag{42}$$

let Δ, ϵ_ω, ϵ_s, θ, $\sigma > 0$, $\rho > \rho_s > 0$, $\zeta(\cdot) = \frac{\theta}{L_x}(e^{L_x(\cdot)} - 1)$, *if the nominal system satisfies* (27) *and*

$$-\epsilon_s + \beta_4(\beta_1^{-1}(\rho))\zeta(k\Delta) + \sigma\zeta^2(k\Delta) \leq -\epsilon_\omega, \tag{43}$$

then for $\boldsymbol{x}(t_0) \in \Omega_\rho/\Omega_{\rho_s}$, *the following inequality holds:*

$$V(\boldsymbol{x}(t_1)) \leq V(\boldsymbol{x}(t_0)) - \epsilon_\omega. \tag{44}$$

Proof. According to (18), the Lyapunov function $V(\boldsymbol{x})$ is continuous. Taking the Taylor expansion of $V(\boldsymbol{x})$ at $\hat{\boldsymbol{x}}(t)$, we obtain [45]:

$$\begin{aligned} V(\boldsymbol{x}(t)) \leq V(\hat{\boldsymbol{x}}(t)) + \frac{\partial V}{\partial \boldsymbol{x}}\|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\| \\ + \sigma\|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\|^2, \forall \boldsymbol{x}(t), \hat{\boldsymbol{x}}(t) \in \Omega_\rho, \end{aligned} \tag{45}$$

where σ is the positive constant associated with the higher-order term of the Taylor expansion, and $\hat{\boldsymbol{x}}(t)$ is the nominal trajectory generated by the backstepping control law $h(\hat{\boldsymbol{x}}(t))$. We can derive the following inequality by the (25):

$$\begin{aligned} V(\boldsymbol{x}(t)) \leq V(\hat{\boldsymbol{x}}(t)) + \beta_4(\beta_1^{-1}(\rho))\|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\| \\ + \sigma\|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\|^2. \end{aligned} \tag{46}$$

Define the error vector between the actual and nominal state trajectories as $\hat{e}(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)$. Derive the error vector:

$$\dot{\hat{\boldsymbol{e}}}(t) = f(\boldsymbol{x}(t), h(\hat{\boldsymbol{x}}(t))) + H_t\psi(t) - f(\hat{\boldsymbol{x}}(t), h(\hat{\boldsymbol{x}}(t))). \tag{47}$$

Following the continuity property of $f(\boldsymbol{x})$ in (26):

$$\begin{aligned} \|\dot{\hat{\boldsymbol{e}}}(t)\| &\leq \|H_t\psi(t)\| + L_x\|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\| \\ &\leq \theta + L_x\|\boldsymbol{e}(t)\|, \end{aligned} \tag{48}$$

where $\boldsymbol{x}(t), \hat{\boldsymbol{x}}(t) \in \Omega_\rho$, $\theta = \sup(H_t\|\psi(t)\|)$, where the function $\sup(\cdot)$ denotes the upper bound of the variable. Due to $\hat{\boldsymbol{x}}(t_0) = \boldsymbol{x}(t_0)$, $\hat{\boldsymbol{e}}(t_0) = 0$. We obtain the following equation by first solving the differential equation (48) and then integrating the error vector $|\dot{\hat{\boldsymbol{e}}}(t)|$:

$$\|\hat{e}(t)\| = \|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\| \leq \frac{\theta}{L_x}(e^{L_x(t-t_0)} - 1), \tag{49}$$

where $t = t_0 + k\Delta$. At the first sampling step $t \in [t_0, t_1]$, the (46) can be rewritten as:

$$V(\boldsymbol{x}(t_k)) \leq V(\hat{\boldsymbol{x}}(t_k)) + \beta_4(\beta_1^{-1}(\rho))\zeta(\Delta) + \sigma\zeta^2(\Delta), \tag{50}$$

according to the (33) at $t \in [t_0, t_1]$ and $\boldsymbol{x}(t_0) = \hat{\boldsymbol{x}}(t_0)$, The (50) can be rewritten as:

$$V(\boldsymbol{x}(t_1)) \leq V(\boldsymbol{x}(t_0)) - \epsilon_s + \beta_4(\beta_1^{-1}(\rho))\zeta(\Delta) + \sigma\zeta^2(\Delta). \tag{51}$$

According to Proposition 1, $V(\hat{\boldsymbol{x}}(t)) \leq V(\hat{\boldsymbol{x}}(t_0))$. In the first sampling step, according to (43), we obtain:

$$V(\boldsymbol{x}(t_1)) \leq V(\boldsymbol{x}(t_0)) - \epsilon_\omega. \tag{52}$$

■

Proposition 4. *Consider the open-loop state trajectory of system* (5) *under backstepping control law* $h(\boldsymbol{x})$ *starting at* $\boldsymbol{x}(t_0)$ *with model residuals* $\psi(t)$ *is obtained by solving the following equation:*

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), h(\hat{\boldsymbol{x}}(t))) + H_t\psi(t), t \in [t_0, t_1], \tag{53}$$

let $\bar{L}_x$, Δ, M, $\theta > 0$, $\rho > \rho_s > 0$ *satisfy* (27), (33), *and* (50), *for* $\boldsymbol{x}(t) \in \Omega$, *the following inequality holds:*

$$-\beta_3(\beta_2^{-1}(\rho_s)) + \bar{L}_x M \Delta + \beta_4(\beta_1^{-1}(\rho))\theta \leq 0. \tag{54}$$

Proof. According to (33) and (50), we can conclude that:

$$V(\boldsymbol{x}(t)) \leq V(\hat{\boldsymbol{x}}(t)) + \beta_4(\beta_1^{-1}(\rho))\zeta(\Delta) + \sigma\zeta^2(\Delta), \tag{55}$$

where $t \in [t_0, t_1]$, $\Delta = t - t_0$. According to (27) when $k = 0$ and $\hat{x}(t_0) = x(t_0)$, the (55) can be rewritten as follows:

$$V(\boldsymbol{x}(t)) \leq V(\boldsymbol{x}(t_0)) + \beta_4(\beta_1^{-1}(\rho))\zeta(\Delta) + \sigma\zeta^2(\Delta). \tag{56}$$

From Proposition 2, define the scalar:

$$\rho_{min} = \max\{V(\hat{\boldsymbol{x}}(t + \Delta)) : V(\hat{\boldsymbol{x}}(t)) \leq \rho_s\}, \tag{57}$$

where the ρ_{min} denotes that when the state is within Ω_{ρ_s}, the maximum value that the Lyapunov function may reach after a sampling moment Δ. (55) can be rewritten as:

$$V(\boldsymbol{x}(t)) \leq \rho_{min} + \beta_4(\beta_1^{-1}(\rho))\zeta(\Delta) + \sigma\zeta^2(\Delta). \tag{58}$$

where $\beta_4(\beta_1^{-1}(\rho))\zeta(\Delta) + \sigma\zeta^2(\Delta)$ also represents the increase of the Lyapunov function caused by the model residuals. According to (56) and (58), we have:

$$\begin{aligned} V(\boldsymbol{x}(t)) \leq \max\{V(\boldsymbol{x}(t_0)), \\ \rho_{min} + \beta_4(\beta_1^{-1}(\rho))\zeta(\Delta) + \sigma\zeta^2(\Delta)\}. \end{aligned} \tag{59}$$

It can be inferred that under the backstepping control law $h(\boldsymbol{x})$ with model residuals, the Lyapunov function $V(\boldsymbol{x}(t))$ remains within an invariant region, ensuring the stability of the system.

Taking the time derivative of the $V(\boldsymbol{x}(t))$:

$$\dot{V}(\boldsymbol{x}(t)) = \frac{\partial V}{\partial \boldsymbol{x}}(f(\boldsymbol{x}(t), h(\hat{\boldsymbol{x}}(t))) + H_t\psi(t)), \tag{60}$$

By adding and subtracting the derivative of Lyapunov function $V(\hat{\boldsymbol{x}}(t_k))$, we obtain:

$$\begin{aligned} \dot{V}(\boldsymbol{x}(t)) = \frac{\partial V}{\partial \boldsymbol{x}}f(\hat{\boldsymbol{x}}(t_k) + h(\hat{\boldsymbol{x}}(t_k))) \\ + \frac{\partial V}{\partial \boldsymbol{x}}(f(\boldsymbol{x}(t), h(\hat{\boldsymbol{x}}(t))) + H_t\psi(t)) \\ - \frac{\partial V}{\partial \boldsymbol{x}}f(\hat{\boldsymbol{x}}(t_k) + h(\hat{\boldsymbol{x}}(t_k))). \end{aligned} \tag{61}$$

Based on the local Lipschitz properties of the Lyapunov function in (26), the equality can be rewritten as the following inequality:

$$\dot{V}(\boldsymbol{x}(t)) \leq -\beta_3(\|\hat{\boldsymbol{x}}(t_k)\|) + \bar{L}_x\|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t_k)\| + \beta_4(\|\boldsymbol{x}(t)\|)H_t\boldsymbol{\psi}(t). \tag{62}$$

According to inequalities (39) and $\boldsymbol{x}(t_k) = \hat{\boldsymbol{x}}(t_k)$, we have:

$$\dot{V}(\boldsymbol{x}(t)) \leq -\beta_3(\beta_2^{-1}(\rho_s)) + L_x M\Delta + \beta_4(\beta_1^{-1}(\rho))\theta. \tag{63}$$

Due that the system is stable, the (54) holds. ∎

Theorem 1. *Consider the closed-loop state trajectory of system under the LSFC-LMPC control law $\boldsymbol{\mu}(t)$ starting at $\boldsymbol{x}(t_k)$ with model residuals $\boldsymbol{\psi}(t)$ and data-driven residual network $\hat{\boldsymbol{\psi}}(t)$ is obtained by solving the following equation:*

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{\mu}(t)) + H_t\Delta\boldsymbol{\psi}(t),\ t \in [t_k, t_{k+1}], \tag{64}$$

where the $\Delta\boldsymbol{\psi}(t) = \boldsymbol{\psi}(t) - \hat{\boldsymbol{\psi}}(t)$ denotes the result of model uncertainty and interference being offset by data-driven residual network. Let Δ, ϵ_ω, $\Delta\theta > 0$ satisfy (54)*, then for $\boldsymbol{x}(t_k) \in \Omega_\rho/\Omega_{\rho_s}$, if*

$$\sup(\|\Delta\boldsymbol{\psi}(t)\|) \leq \sup(\|\boldsymbol{\psi}(t)\|), \tag{65}$$

where the function $\sup(\cdot)$ *denotes the upper bound of the function inside the parentheses, then the following inequality holds:*

$$V(\boldsymbol{x}(t)) \leq V(\boldsymbol{x}(t_k)),\ t \in [t_k, t_{k+1}]. \tag{66}$$

Proof. According to $\Delta\boldsymbol{\psi}(t) = \boldsymbol{\psi}(t) - \hat{\boldsymbol{\psi}}(t)$, the (64) include the real model residuals $\boldsymbol{\psi}(t)$ and the predicted model residuals $\hat{\boldsymbol{\psi}}(t)$. $\Delta\boldsymbol{\psi}(t)$ represents the remaining residual after compensation by the temporal neural network. Taking the time derivative of the Lyapunov funvtion $V(x(t))$:

$$\dot{V}(\boldsymbol{x}(t)) = \frac{\partial V}{\partial \boldsymbol{x}}(f(\boldsymbol{x}(t), \boldsymbol{\mu}(t)) + H_t\Delta\boldsymbol{\psi}(t)), \tag{67}$$

According to the constraint (23f), the following inequality holds:

$$\dot{V}(\boldsymbol{x}(t)) \leq \frac{\partial V}{\partial \boldsymbol{x}} f(\hat{\boldsymbol{x}}(t_k), h(\hat{\boldsymbol{x}}(t_k))) + \frac{\partial V}{\partial \boldsymbol{x}}(f(\boldsymbol{x}(t), \boldsymbol{\mu}(t_k)) - \frac{\partial V}{\partial \boldsymbol{x}} f(\tilde{\boldsymbol{x}}(t_k), \boldsymbol{\mu}(t_k)) + \frac{\partial V}{\partial \boldsymbol{x}} H_t\Delta\boldsymbol{\psi}(t)). \tag{68}$$

Based on the local Lipschitz properties of the Lyapunov function in (26), the equality can be rewritten as the following inequality:

$$\dot{V}(\boldsymbol{x}(t)) \leq -\beta_3(\|\hat{\boldsymbol{x}}(t_k)\|) + \bar{L}_x\|\boldsymbol{x}(t) - \tilde{\boldsymbol{x}}(t_k)\| + \beta_4(\|\boldsymbol{x}(t)\|)\Delta\theta, \tag{69}$$

where $\Delta\theta = \sup(H_t\|\Delta\boldsymbol{\psi}(t)\|)$. Since $\tilde{\boldsymbol{x}}(t_k) = \boldsymbol{x}(t_k)$ in (23c), the (69) can be rewritten as follows:

$$\dot{V}(\boldsymbol{x}(t)) \leq -\beta_3(\beta_2^{-1}(\rho_s)) + \bar{L}_x M\Delta + \beta_4(\beta_1^{-1}(\rho))\Delta\theta. \tag{70}$$

According to (65), we can obtain $\Delta\theta \leq \theta$, where $\theta = \sup(H_t\|\boldsymbol{\psi}(t)\|)$. If the condition (54) holds, we can deduce that:

$$\dot{V}(\boldsymbol{x}(t)) \leq 0. \tag{71}$$

Integrating the Lyapunov function in $[t_k, t_{k+1}]$, the (66) holds. ∎

Remark 5. *In Proposition 2, 3 and Theorem 1, several conditions that need to be satisfied. The conditions in Proposition 2 can be satisfied by appropriately designing the controller $h(x)$ and a sufficiently small sampling period Δ. In Proposition 3, in addition to selecting appropriate $h(x)$ and Δ, reducing the disturbance θ is also beneficial. The conditions of Theorem 1 can be satisfied by appropriately designing a residual network such that the predicted result reduce the model residual after compensation.*

Remark 6. *We did not impose explicit threshold constraints on the neural network output but instead adopted indirect methods to ensure the output remained within a reasonable range. First, we ensure that the training data lies within a bounded range. Second, we ensure that the vehicle measurement data used for predicting residuals during tracking control is bounded and falls within the range of the training dataset. Finally, during the training process, the loss of our proposed neural network decreased steadily without significant fluctuations. If the neural network encounters an out-of-distribution scenario,due to the output characteristics of neural networks and the instability of measurement results, situations may arise where* (65) *is not satisfied. In this scenario, since the conditions for controller stability cannot be guaranteed, the algorithm fails to satisfy the Lyapunov stability constraints based on backstepping, rendering it unsolvable.*

REFERENCES

[1] H. W. Yubiao Zhang, Yanjun Huang and A. Khajepour, "A comparative study of equivalent modelling for multi-axle vehicle," *Veh. Syst. Dyn.*, vol. 56, no. 3, pp. 443–460, 2018.

[2] L. Cai, Z. Liao, S. Wei, and J. Li, "Novel direct yaw moment control of multi-wheel hub motor driven vehicles for improving mobility and stability," *IEEE Trans. Ind. Appl.*, vol. 59, no. 1, pp. 591–600, 2022.

[3] L. Zhai, C. Wang, Y. Hou, and C. Liu, "MPC-Based integrated control of trajectory tracking and handling stability for intelligent driving vehicle driven by four hub motor," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2668–2680, 2022.

[4] H. Zhang and J. Wang, "Active steering actuator fault detection for an automatically-steered electric ground vehicle," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 3685–3702, 2017.

[5] W. Wang, Y. Zhang, C. Yang, T. Qie, and M. Ma, "Adaptive model predictive control-based path following control for four-wheel independent drive automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14 399–14 412, 2022.

[6] F. Tian, Z. Li, F.-Y. Wang, and L. Li, "Parallel learning-based steering control for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 379–389, 2022.

[7] J. Wu, Z. Huang, and C. Lv, "Uncertainty-aware model-based reinforcement learning: Methodology and application in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 194–203, 2022.

[8] P. Zhang, T. Liu, and Z.-P. Jiang, "Tracking control of unicycle mobile robots with event-triggered and self-triggered feedback," *IEEE Trans. Autom. Control*, vol. 68, no. 4, pp. 2261–2276, 2023.

[9] J. Pliego-Jime´nez, R. Mart´ınez-Clark, C. Cruz-Herna´ndez, and A. Arellano-Delgado, "Trajectory tracking of wheeled mobile robots using only cartesian position measurements," *Automatica*, vol. 133, p. 109756, 2021.

[10] A. C. Manav, I. Lazoglu, and E. Aydemir, "Adaptive path-following control for autonomous semi-trailer docking," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 69–85, 2022.

[11] J.-X. Zhang, J. Ding, and T. Chai, "Fault-tolerant prescribed performance control of wheeled mobile robots: A mixed-gain adaption approach," *IEEE Trans. Autom. Control*, vol. 69, no. 8, pp. 5500–5507, 2024.

[12] C.-L. Hwang, C.-C. Yang, and J. Y. Hung, "Path tracking of an autonomous ground vehicle with different payloads by hierarchical improved fuzzy dynamic sliding-mode control," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 899–914, 2017.

[13] Z. Zhang, L. Zheng, Y. Li, S. Li, and Y. Liang, "Cooperative strategy of trajectory tracking and stability control for 4wid autonomous vehicles under extreme conditions," *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3105–3118, 2023.

[14] E. A. Gedefaw, C. M. Abdissa, and L. N. Lemma, "An improved trajectory tracking control of quadcopter using a novel sliding mode control with fuzzy pid surface," *PloS One*, vol. 19, no. 11, p. e0308997, 2024.

[15] H. Pang, R. Yao, P. Wang, and Z. Xu, "Adaptive backstepping robust tracking control for stabilizing lateral dynamics of electric vehicles with uncertain parameters and external disturbances," *Control Eng. Pract.*, vol. 110, p. 104781, 2021.

[16] Y. A. Wendemagegn, W. Ayalew Asfaw, C. M. Abdissa, and L. N. Lemma, "Enhancing trajectory tracking accuracy in three-wheeled mobile robots using backstepping fuzzy sliding mode control," *Eng. Res. Express*, vol. 6, no. 4, p. 045204, oct 2024.

[17] L. Zhai, C. Wang, Y. Hou, and C. Liu, "MPC-based integrated control of trajectory tracking and handling stability for intelligent driving vehicle driven by four hub motor," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2668–2680, 2022.

[18] S. Kim, J. Lee, K. Han, and S. B. Choi, "Vehicle path tracking control using pure pursuit with MPC-based look-ahead distance optimization," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 53–66, 2024.

[19] Z. Sun, L. Dai, Y. Xia, and K. Liu, "Event-based model predictive tracking control of nonholonomic systems with coupled input constraint and bounded disturbances," *IEEE Trans. Autom. Control*, vol. 63, no. 2, pp. 608–615, 2018.

[20] R. Soloperto, A. Mesbah, and F. Allgo¨wer, "Safe exploration and escape local minima with model predictive control under partially unknown constraints," *IEEE Trans. Autom. Control*, vol. 68, no. 12, pp. 7530–7545, 2023.

[21] W. Xiao, C. G. Cassandras, and C. A. Belta, "Bridging the gap between optimal trajectory planning and safety-critical control with applications to autonomous vehicles," *Automatica*, vol. 129, p. 109592, 2021.

[22] H. Peng, W. Wang, Q. An, C. Xiang, and L. Li, "Path tracking and direct yaw moment coordinated control based on robust mpc with the finite time horizon for autonomous independent-drive vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6053–6066, 2020.

[23] C. A. Kedir and C. M. Abdissa, "PSO based linear parameter varying-model predictive control for trajectory tracking of autonomous vehicles," *Eng. Res. Express*, vol. 6, no. 3, p. 035229, aug 2024.

[24] S. Mata, A. Zubizarreta, and C. Pinto, "Robust tube-based model predictive control for lateral path tracking," *IEEE Trans. Intell. Veh.*, vol. 4, no. 4, pp. 569–577, 2019.

[25] 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 Trans. Ind. Electron.*, vol. 68, no. 3, pp. 2460–2469, 2021.

[26] W. Zhao, H. Wei, Q. Ai, N. Zheng, C. Lin, and Y. Zhang, "Real-time model predictive control of path-following for autonomous vehicles towards model mismatch and uncertainty," *Control Eng. Pract.*, vol. 153, p. 106126, 2024.

[27] B.-S. Hong, W.-J. Su, and C.-Y. Chou, "LPV modeling and game-theoretic control synthesis to design energy-motion regulators for electric scooters," *Automatica*, vol. 50, no. 4, pp. 1196–1200, 2014.

[28] D. Kapsalis, O. Sename, V. Milanes, and J. J. Molina, "A reduced LPV polytopic look-ahead steering controller for autonomous vehicles," *Control Eng. Pract.*, vol. 129, p. 105360, 2022.

[29] H. Zheng, Y. Li, L. Zheng, and E. Hashemi, "Safe motion planning and control framework for automated vehicles with zonotopic TRMPC," *Engineering*, vol. 33, pp. 146–159, 2024.

[30] J. Berberich, J. Ko¨hler, M. A. Mu¨ller, and F. Allgo¨wer, "Linear tracking MPC for nonlinear systems—Part II: The data-driven case," *IEEE Trans. Autom. Control*, vol. 67, no. 9, pp. 4406–4421, 2022.

[31] L. Consolini and C. M. Verrelli, "Learning control in spatial coordinates for the path-following of autonomous vehicles," *Automatica*, vol. 50, no. 7, pp. 1867–1874, 2014.

[32] R. Zhao, W. Deng, Y. Wang, K. Huang, B. Zheng, and J. Ding, "An improved data-driven method for steering feedback torque of driving simulator," *IEEE/ASME Trans. Mechatron.*, vol. 28, no. 5, pp. 2953–2963, 2023.

[33] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2736–2743, 2020.

[34] M. Zhu, X. Wang, and Y. Wang, "Human-like autonomous car-following model with deep reinforcement learning," *Transp. Res. Part C Emerging Technol.*, vol. 97, pp. 348–368, 2018.

[35] E. Picotti, E. Mion, A. D. Libera, J. Pavlovic, A. Censi, E. Frazzoli, A. Beghi, and M. Bruschetta, "A learning-based nonlinear model predictive controller for a real Go-Kart based on black-box dynamics modeling through gaussian processes," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 5, pp. 2055–2065, 2023.

[36] K. H. Kim, C. Jeong, J. Kim, S. Lee, and C. M. Kang, "Data-Driven LSTM model and predictive control for vehicle lateral motion," *Journal of Electrical Engineering & Technology*, vol. 19, no. 6, pp. 3635–3644, 2024.

[37] G. Chen, T. Wu, X. Li, and Y. Zhang, "Secure and safe control of connected and automated vehicles against false data injection attacks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 9, pp. 12 347–12 360, 2024.

[38] 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 Trans. Intell. Veh.*, vol. 8, no. 1, pp. 135–146, 2023.

[39] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Rob. Autom. Lett.*, vol. 4, no. 4, pp. 3363–3370, 2019.

[40] H. Bao, Q. Kang, X. Shi, L. Xiao, and J. An, "Robust learning-based model predictive control for intelligent vehicles with unknown dynamics and unbounded disturbances," *IEEE Trans. Intell. Veh.*, vol. 9, no. 2, pp. 3409–3421, 2024.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] P. Falcone, H. Eric Tseng, F. Borrelli, J. Asgari, and D. Hrovat, "MPC-based yaw and lateral stabilisation via active front steering and braking," *Veh. Syst. Dyn.*, vol. 46, no. S1, pp. 611–628, 2008.

[43] J. Liu, X. Chen, D. M. Mun˜oz de la Pena, and P. D. Christofides, "Iterative distributed model predictive control of nonlinear systems: Handling asynchronous, delayed measurements," *IEEE Trans. Autom. Control*, vol. 57, no. 2, pp. 528–534, 2012.

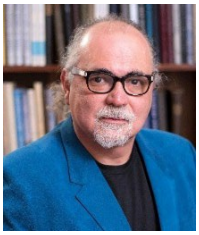
[44] P. D. Christofides, J. Liu, and D. Mun˜oz de la Pen˜a, *Lyapunov-Based Model Predictive Control,* in *Networked and Distributed Predictive Control: Methods and Nonlinear Process Network Applications*. London: Springer London, 2011, pp. 13–45.

[45] D. Munoz de la Pena and P. D. Christofides, "Lyapunov-based model predictive control of nonlinear systems subject to data losses," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2076–2089, 2008.

Yongkang Zhang received the B.Sc. degree form Shandong University of Science and Technology, Qingdao, China, in 2018 and the M.Sc. degree in Beihang University, Beijing, China, in 2021. He is currently a Ph.D. student at the School of Transportation Science and Engineering, Beihang University, Beijing, China. His research interests include automated vehicle dynamics control, automated chassis design and special vehicle collaborative control.

Henglai Wei received the B.S. and M.Sc. degrees in control theory from Northwestern Polytechnical University, China, in 2014 and 2017, respectively, and his Ph.D. degree in mechanical engineering at the University of Victoria, Canada, in 2022.

He is currently an Associate Professor in the School of Transportation Science and Engineering, Beihang University. His research interests include distributed control and optimization, autonomous driving, as well as intelligent unmanned systems. Dr. Wei is an active reviewer for more than ten international journals and conferences.



Marcelo Godoy Simo˜es (Fellow, IEEE) received the B.Sc. and M.Sc. degrees from the University of Sa˜o Paulo, Brazil, in 1985 and 1990, respectively, the Ph.D. degree from The University of Tennessee, Knoxville, TN, USA, in 1995, the D.Sc. degree (Livre-Docecˆncia) from the University of Sa˜o Paulo, in 1998. From 2014 to 2015, he was a U.S. Fulbright Fellow with the Institute of Energy Technology, Aalborg University, Denmark. He is a pioneer in applying neural networks and fuzzy logic in power electronics, motor drives, and renewable energy systems. His fuzzy logic-based modeling and control for wind turbine optimization is used as a basis for advanced wind turbine control, and it has been cited worldwide. His leadership in modeling fuel cells is internationally and highly influential in providing a basis for further developments in fuel cell automation control in many engineering applications. He made substantial and lasting contributions to artificial intelligence technology in many applications, power electronics, motor drives, and fuzzy control of wind generation systems, such as fuzzy logic-based waveform estimation for power quality, neural network-based estimation for vector-controlled motor drives, and the integration of alternative energy systems to the electric grid through AI modeling-based power electronics control. His current research interests include power electronics, power systems, power quality, smart grids, and renewable energy systems. He was elevated to the grade of IEEE Fellow, with the citation: ”for applications of artificial intelligence in control of power electronics systems”.

Jicheng Chen received his B.Sc. degree and M.Sc degree in control science and engineering from Harbin Institute of Technology, China. He received his Ph.D. degree from the Department of Mechanical Engineering, University of Victoria, Victoria, BC, Canada in 2021. He served as a Postdoctoral Research Fellow at Beihang University in Beijing, China from 2021 to 2023. His primary research interests include model predictive control and its applications in vehicular platoon systems and vehicle dynamics control systems.

Hui Zhang (Fellow, IEEE) received the B.Sc. degree in mechanical design manufacturing and automation from the Harbin Institute of Technology, Weihai, China, in 2006, the M.Sc. degree in automotive engineering from Jilin University, Changchun, China, in 2008, and the Ph.D. degree in mechanical engineering from the University of Victoria, Victoria, BC, Canada, in 2012. He is a Professor at Beihang University, China. Dr. Zhang’s research interests include special vehicle design, vehicle dynamics control, and intelligent driving.

He is the recipient of the 2017 IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Award, the 2018 SAE Ralph R. Teetor Educational Award, the IEEE Vehicular Technology Society 2019 Best Vehicular Electronics Paper Award, and the 2019 SAE INTERNATIONAL INTELLIGENT AND CONNECTED VEHICLES SYMPOSIUM Best Paper Award. He is an IEEE Fellow and a member of SAE International and ASME. He serves as Vice Chairman of the IEEE IES Industrial Information Physical Systems Technical Committee, Co-Chairman of the SAE International Intelligent and Connected Vehicle Technical Committee, and is an Editorial Board Member or Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, MECHANICAL SYSTEMS AND SIGNAL PROCESSING, and other prominent journals.