Research Article

Implementation of the Human-Like Lane Changing Driver Model Based on Bi-LSTM

Junyu Cai,1 Haobin Jiang,1 and Junyan Wang2

1School of Automobile and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China
2School of Automotive Engineering, Zhenjiang College, Zhenjiang 212003, China

Correspondence should be addressed to Haobin Jiang; cjy415851057@126.com

Received 23 March 2021; Revised 25 March 2022; Accepted 6 April 2022; Published 2 May 2022

Abstract

If the driving behavior of an autonomous vehicle is similar to that of a skilled driver, the human driver can extricate himself from fatigue operation and the comfort of passengers can also be guaranteed. Therefore, this paper studies the human-like lane-changing model of an autonomous vehicle. The lane-changing characteristic data of skilled drivers are collected and analyzed through a real vehicle test. Then, comparing the MPC-based driver model with the steering wheel angle of human drivers, we found that the MPC-based model could hardly reflect the maneuvering characteristics of human drivers, so we proposed a driver model with steering wheel angle continuity for human drivers. This paper uses four neural network models to compare the prediction on the test set, then uses different input types to compare the prediction accuracy of the model, and finally verifies the generalization ability of the model on the verification set. These three test results show that the prediction results of the human-like lane-changing driving model based on Bi-LSTM are closest to the real steering wheel angle sequence of skilled drivers. The test results demonstrate that the Bi-LSTM-based human-like lane-changing driving model achieves 9.8% RMSE and 6.8% MAE, which improves 10.8% RMSE and 10.3% MAE over LSTM. The model can generate the steering wheel angle sequence in the process of lane changing like a human, so as to realize the human simulation control of an autonomous vehicle for lane-changing conditions.

1. Introduction

In recent years, with the development of autonomous vehicles, unmanned vehicles will enter the actual road traffic system in the future and will operate in a mixed driving environment for a long time [1]. In this context, we need to consider the actual driving behavior of human drivers when developing autonomous driving systems [2]. If some driving behaviors of the vehicle are close to humans, then autonomous vehicles are more acceptable to the public. If the driving behavior of AV is similar to that of a highly skilled human driver, the driver can be freed from the operation of enduring fatigue, and the comfort of the passengers can also be guaranteed. At the same time, the human drivers of the surrounding vehicles can better understand and predict the behavior of the vehicle, and the interaction between the vehicles is smoother [3].

The behavior of human drivers includes the perception of information [4], judgment and reasoning, and decision-making and finally produces the steering behavior such as direction control, driving control, and braking control, which has a strong randomness and adaptability. The driver model is the expression of these manipulation behaviors. In order to make autonomous vehicles understand the behavior of human drivers and human decision making and control capabilities, various driver models have been developed based on the classical control theory, modern control theory, and intelligent control theory [5].

The classical control theory is a control theory based on transfer function, and the object of the study is a single-input, single-output automatic control system. MacAdam and Charles [6] in 1980 proposed an optimal preview control for the linear system model successfully and applied it to lane change, which combines pursuit behavior, compensatory
behavior, and human presighted predictive behavior. Since then, more and more studies on the presighted pilot models have been conducted, and according to the number of preview points, the preview driver models can be classified into 3 categories: single-point, two-point, and multipoint preview driver models. Guo and Fancher [7] proposed the theory of the preview-following method in 1982, which is an early proposed single-point preview driver model that considers the driver’s decision to be divided into a pre-scanning phase and a compensated following phase, whose driver’s correction link depends on the vehicle dynamics model. The single-point preview is generally for small curvature working conditions of the path and needs to set a reasonable presighting point. The preview point cannot be too far from the vehicle or the preview information will be invalid, and it cannot be too close to the vehicle or the steering maneuver will be very frequent, which will lead to a very poor final control effect. To address the deficiencies in the single-point preview model, Salvucci and Gray [8] proposed a two-point preview mechanism in 2004. Based on this, Sentouh [9] proposed a two-point preview driver model where the expectation and compensation are related to the angle of the near point data at the far point, respectively. Symonds [10] proposed a multipoint preview path steering control method using the linear quadratic adjustment theory to achieve optimal control.

At present, compensation models, preview compensation tracking models, and preview optimal curvature following models have been established on the basis of transfer function and optimal control. However, due to the nonlinear and time-varying characteristics of driver’s driving characteristics, it is difficult for the abovementioned models to simulate the actual driver’s maneuvering behavior. Considering the driver’s maneuvering characteristics, the corresponding driver model based on the fuzzy control, neural network, and the combination of these two can be closer to the actual driver’s driving behavior by using the intelligent control theory. Rix and Cole [11] simulated the driver’s steering behavior and established a corresponding model using the fuzzy mathematical theory based on real vehicle steering road tests, which integrates the control problems of both the preview link and the compensation link. Model predictive control (MPC) is another advanced and effective trajectory tracking control algorithm that is suitable for solving constrained optimal control problems and is also often applied to build driver models [12]. The MPC is based on a different predictive model, using the principle of iterative online rolling optimization with feedback corrections, so that it is highly robust, well-optimized, and highly stable in control systems [13]. Verschueren et al [14] designed a novel steering control method for driverless automobiles based on the nonlinear MPC, which enables the vehicle to travel stably along a desired or ideal trajectory while satisfying its physical constraints. Based on a linear optimal path tracking controller, Ungoren and Peng [15] proposed an adaptive lateral driver model with the MPC. Bageshwar et al. investigated the intelligent vehicle autonomous cruise control model [16] and active steering control model [17], respectively, based on the model predictive control theory. Bruschetta [18] proposed a real-time MPC motion control model based on human driver characteristics for the requirement of real-time steering control. Yoshida et al. [19] developed a nonlinear vehicle model and used nonlinear MPC to develop a vehicle steering control model for lane change conditions, which can take into account the vehicle’s speed and road adhesion constraints.

The preview-based driver model algorithm is simple but allows to characterize the preview characteristics of human drivers, and the MPC algorithm allows to solve the optimal control problem under multiple constraints. All of the above approaches model steering control with human driving characteristics to a certain extent; however, none of them provide a detailed analysis of the key parameters affecting the steering behavior of human drivers. With the development and application of machine learning, especially deep learning, data-driven approaches to model driver models are becoming more and more widespread. The end-to-end (End2End) [20] driver model is a promising research direction nowadays, where steering wheel corners, etc., are obtained directly from images that are closer to human drivers’ driving behavior, without specifying some human rules, and with lower system cost. Bojarski et al. [21] designed an end-to-end automatic steering model based on Pomerleau’s research using only general convolutional neural networks that map the raw image information captured by a single front camera directly to the steering maneuver, and the model learns the task of complete lane-keeping driving. Similarly, Chen et al. [22] designed convolutional long-term short-term memory neural networks to learn human driving data in video games and behavioral reflex approaches that directly map an input image to a driving action by a regressor. Neural network-based models have accurate predictive performance and also have good generalization capabilities [23]. He improved CNN networks and proposed two improved CNN algorithms: the ME mask R–CNN [24] and multiscale one-stage object detector FE-YOLO [25] for rail transit obstacle detection, respectively. The experimental results demonstrate that the improved CNN networks have high detection accuracy and strong generalization. This type of an approach directly constructs the driver model using the driver’s maneuvering data, thus obtaining a driver model that is highly similar to a human driver.

To compare the differences between the MPC-based driver model and skilled drivers, a real vehicle experiment was conducted to collect the steering wheel angle, trajectory, speed, and other driving data from experienced drivers. Ten skilled drivers and four test vehicles were included in the experiment. Then, we built an MPC-based driver model based on CarSim and Simulink. In this paper, LCSS is used to investigate the similarity between the MPC-based driver model and human driver, and the calculation results demonstrate that the difference between the steering wheel turning angle of skilled driver and MPC driver model is large. Finally, based on the shortcomings of MPC drivers and the demand for human-like driving, a human-like driver model is developed. The human-like steering wheel angle generated by the human-like driver model was evaluated for
its performance with different hidden layers and history steps.

The remainder of the paper is organized as follows. Details of skilled driver lane change data acquisition tests are introduced in Section 2. Section 3 represents test data analysis. Section 4 illustrates the classical MPC-based drive model, its simulation setup in CarSim + Simulink, and a comparison with human driver. In Section 5, a human-like lane change driver model is established with four different neural networks. In Section 6, the simulation results and analysis are presented. Finally, the conclusions and the outlook are drawn in Section 7.

2. Data Acquisition

2.1. Skilled Driver Lane Change Data Acquisition. Vehicle lane change is a basic driving behavior during driving. The main purpose of this test is to collect the changed behavior of the driver when changing lanes. We believe that the driving school coach is a skilled and excellent driver who has rich driving experience and knows how to drive the vehicle safely and comfortably. So, we chose 10 driving school coaches, their driving age is 10–28 years, age is 24–55, including 7 males and 3 females. Before the test, the driver is informed to test the circuit and let it freely change lanes according to the daily routine. During the test, no interference is made to the driver to ensure the data obtained are true and reliable.

Figure 1 shows equipment installation drawing and the vehicles used in the test. The test vehicles selected are the general GL8, Skoda Octavia, Honda Accord, and SAIC MG. We think that the four vehicles in Figure 1 can represent most of the models on the market. The test equipment include an S-motion biaxial optical speed sensor, Kimsw force steering wheel sensor, and SDI-600GIGPS/INS; the installation location is shown in Figure 1. These pieces of equipment can measure yaw rate signal, roll angle signal, lateral acceleration signal, steering wheel angle signal, steering wheel torque signal, angular velocity signal, and vehicle trajectories. The test road selects the campus road within Jiangsu University, as shown in Figure 2. During the test, the drivers kept freely changing lanes at a speed from 20 km/h to 50 km/h on the test road.

2.2. Data Processing. The WGS84 coordinate system is the coordinate system currently used by GPS. In actual cases, the GPS measured data need to be converted into the geodetic coordinates of the Beijing 54 coordinate system. We define the lateral offset of the vehicle as to the distance Dy of the vehicle from the road boundary, and the lateral offset is obtained by the transformed geodetic coordinates and the road boundary [26, 27].

The collected steering characteristic parameters and vehicle dynamic parameters are smoothed and filtered, and the parameters are time-aligned to obtain an excellent driver lane change behavior data set. This paper selects bass filters of different orders to process test data. Wherein, the second-order Butterworth filter is for yaw rate, roll angle signal, and lateral acceleration signal; the first-order Butterworth filter is for the steering wheel angle signal, steering wheel torque signal, and angular velocity signal. The number of test groups and test speed in each lane change condition is shown in Table 1.

3. Test Results

3.1. Vehicle Speed. Vehicle speed is one of the indicators that can characterize the driver’s lane-changing behavior. When analyzing the speed, all kinds of test data need to be aligned according to time first. The average speed at each moment is defined as follows:

$$\overline{V}_j = \frac{1}{N} \sum_{i=1}^{N} V_{ij}$$

where $\overline{V}_j$ represents the average speed at the $j$th moment, $T$ represents the total time of lane change, $N$ represents the number of experiments at the specified target speed, and $j$ represents the speed of the $j$th test. At the same time, the standard deviation (SD) of the speed can reflect the dispersion degree of the speed, and its calculation formula is as follows:

$$SD_j = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (V_{ij} - \overline{V}_j)}.$$  

According to the analysis of test data, a stable speed can make passengers feel more comfortable and reduce the impact of vehicle speed on the driver’s steering results when turning. Figure 3 shows the average value and standard deviation of ten skilled drivers at each specified speed at each time on the test road, which can reflect the speed control operation of skilled drivers when changing lanes.

When the specified speed is 20 km/h, the standard deviation of the driver’s speed in the whole lane-changing process is slightly larger than that of the other three speeds, and the average speed exceeds the specified speed. It indicates that the driver’s expected speed at this time is higher than 20 km/h. In the actual driving scenario, the lane-changing operation with the speed as low as 20 km/h may occur when the road traffic is congested.

The standard deviation of vehicle speed is relatively small when the specified vehicle speed is 30 km/h, 40 km/h, and 50 km/h. Especially, in the front section of lane change, the driver controls the speed more accurately. Drivers pay more attention in the front section of lane change. When they reach the target lane, the speed control is not as accurate as in the front section and there will be fluctuations.

When the specified speed is 30 km/h and 40 km/h, there is almost no difference between the average value of the speed and the specified speed. When the specified speed is 50 km/h, the average speed of the drivers is lower than the specified speed. This indicates that when the vehicle speed is too high, the driver’s expected speed may be lower than the specified speed for safety reasons. It can be seen that the speed collected in this paper can reflect the characteristics of human drivers when changing lanes, and the maximum
standard deviation of drivers’ speed on the test road is 4.1 and the minimum is 1.2. This means that the skilled driver’s control of vehicle speed during lane change is accurate and stable, which is in line with the driving habits of human drivers.

3.2. Lateral Offset. The distance between the vehicle center and the leftmost boundary of the road is called the lateral offset, which is used to collect the lane change trajectory of skilled drivers in the real vehicle test. The lateral offset is represented by Dy. The variation law of lateral offset dy directly reflects the control of vehicle trajectory by driver’s lane-changing manipulation. It is the embodiment of the driver’s own lane-changing control characteristics. Figure 4 shows the lateral offset and the test road boundary.

3.2.1. Effect of Vehicle Speed on Lateral Offset. Figure 5 shows the lateral offset of different skilled drivers at the specified speed of 30 km/h and 40 km/h, respectively. By comparing the lateral offsets at different speeds, it is found that the lane-changing trajectories are different, but the difference is small. Another finding is that skilled drivers change lanes more intensively at the specified speed of 40 km/h, while they are relatively sparse at the specified speed of 30 km/h. This indicates that the gap between trajectories is large at lower speeds. The higher the speed, the smaller the difference between trajectories and the higher the similarity. Taking five skilled drivers as an example, the lane-changing trajectories at the specified speed of 30 km/h and
40 km/h are similar. Similar results were found in lane change trajectory analysis at a specified speed from 20 km/h to 50 km/h.

3.2.2. Effect of the Vehicle Type on Lateral Offset. Figure 6 shows 8 sets of lateral offset data of skilled drivers 1 and 3 at the specified speed of 30 km/h. C1 to C4 are the numbers of 4 test vehicles in the experiment. The 8 groups of data in each figure are the data of the driver driving each test vehicle twice. It is found from Figure 6 that the similarity of 8 groups of lateral offsets in the two figures is very high. The lateral offsets of skilled driver 1 are concentrated, while the lateral offsets of skilled driver 3 are relatively loose. The trajectories of the same test vehicle may be scattered. The trajectories of different vehicles may also be clustered together. There are no specific rules. Therefore, the different brands and models of each test vehicle in the lane-changing test have little impact on the lateral offset of skilled drivers.

3.2.3. Similarity Analysis of Lateral Offset. Due to the high similarity between lane-changing trajectories of different skilled drivers, it is necessary to select a suitable method to measure the data similarity when analyzing the similarity of lane-changing trajectories. At present, there are many methods used to measure data similarity, such as the Euclidean distance, dynamic time warping (DTW), edit distance on real sequence (EDR), longest common subsequence (LCSS), cosine similarity, and the Hausdorff distance. In practical application, it is necessary to select appropriate methods for similarity analysis according to the data type and scene type.

DTW and EDR are very sensitive to the difference of individual points of the trajectory. If the two sequences have similar time series in most periods and only show some differences in a short time, the similarity cannot be accurately analyzed by using DTW and EDR. If the sequences have similar shapes for a long time, LCSS can more accurately measure the similarity of the two-time series.

Suppose the sequence $A$ and sequence $B$ are two-lane change trajectory data, with their respective lengths $n$ and $m$, then the length of the longest common subsequence is as follows:

\[
\text{LCSS}(A, B) = \begin{cases} 
1 + \text{LCSS}(a_{i-1}, b_{i-1}), & \text{if } \text{dist}(a_i, b_i) < \epsilon, \\
\max\{\text{LCSS}(a_{i-1}, b_i), \text{LCSS}(a_i, b_{i-1})\}, & \text{otherwise}.
\end{cases}
\]
Among them, \( r \) is a similarity threshold, \( t = 1, 2, \ldots, n; \)
\( i = 1, 2, \ldots, m. \) According to the above formula, the similarity
formula is as follows:

\[
D_{LCSS} = 1 - \frac{LCSS(A, B)}{\min(\text{len}_A, \text{len}_B)},
\]

(4)

Organize all skilled drivers’ lane-changing tracks at different specified speeds into 4 groups of lane-changing trajectories. Calculate the similarity of lane-changing trajectories within and between the four groups, respectively, and obtain the average similarity of lane-changing trajectories within and between the groups according to the following formula:

\[
\overline{D}_{LCSS} = \frac{\sum_{i=1}^{M} D_{LCSSI}}{M},
\]

(5)

where \( \overline{D}_{LCSS} \) is the average value of lane change trajectory similarity and \( M \) is the sample size of each group.

Table 2 shows the LCSS average values of lane-changing trajectories of different skilled drivers collected at different specified speeds. The more similar the lane change trajectory is, the smaller the average value of LCSS is. The average value of LCSS reaches the minimum value of 0.1473 when the specified speed reaches 30 km/h. This shows that the current lane change trajectory has the highest similarity. As the speed increases, the average value of LCSS increases, which means that the similarity between lane-changing trajectories decreases, but the similarity between lane-changing trajectories remains very high. The lane-changing behavior of skilled drivers can be maintained in a stable range at the same specified speed, and the driver’s personality difference has little impact on the lane-changing trajectory. Therefore, we study different skilled drivers as a group.

3.3. Comparison of Lane Changing Trajectories between Skilled Drivers and NGSIM Drivers. According to the analysis results of lane-changing trajectory of skilled drivers in Section 3.2, it can be concluded that different skilled drivers have similar lane-changing control characteristics.
Then, a question arises: what are the differences in lane-changing trajectories between skilled drivers and ordinary drivers? In order to understand the similarities and differences between lane-changing trajectories of skilled drivers and ordinary drivers, this section will randomly select the lateral offset data of some NGSIM drivers as the lane-changing trajectories of ordinary drivers, and the lateral offset obtained from real vehicle test as the lane-changing trajectories of skilled drivers. LCSS similarity is used to compare and analyze the two types of driver lane-changing trajectories. Figure 7 shows the lateral offset data of 4 skilled drivers and 4 NGSIM drivers. The LCSS similarity calculation results of 8 groups of lane-changing tracks in Figure 7 are shown in Tables 3-5: Table 3 shows the LCSS similarity calculation results of lane-changing tracks among skilled drivers; Table 4 shows the calculation results of LCSS similarity of lane change trajectory between ngsim drivers; and finally, Table 5 shows the calculation results of LCSS similarity between skilled drivers and ngsim drivers.

In Table 3, the average LCSS similarity of four groups of lane-changing trajectories of skilled drivers is 0.4032, and the average LCSS similarity of four groups of lane-changing trajectories of ngsim drivers in Table 4 is 0.5238. The results show that the lane change trajectory similarity between skilled drivers is higher than that of ngsim drivers. The lane-changing trajectory of skilled drivers has similar lane-changing characteristics. Similarly, it can be seen that the steering trajectory of skilled drivers can be controlled in a narrower area. In Table 5, the average LCSS similarity of 8 groups of lane change tracks between skilled drivers and ngsim drivers is 0.610. This shows that although the similarity between ordinary drivers and skilled drivers is slightly larger, the changing trend between trajectories is similar. In conclusion, it is considered that skilled drivers can maintain relatively stable lane-changing control characteristics compared with ordinary drivers during lane changing.

### 4. MPC-Based Driver Model

From a control theory perspective, the key capability of a driver model is to control the driverless vehicle to follow the desired trajectory. In order to simulate the behavioral characteristics of driving under different traffic conditions, researchers have built several types of driver models. The MPC is a classic driver model; however, the MPC driver model cannot reflect the characteristics of human drivers’ maneuvers when controlling the steering wheel angles. To this end, we propose a human-like driver model based on the Bi-LSTM neural network.

#### 4.1. Structure of the MPC Driver Model

The MPC driver model trajectory tracking problem involves designing a control input for the system based on MPC theory that enables the autonomous vehicle to follow a specified route given the intended trajectory, as shown in Figure 8. The model predictive control introduces a vehicle dynamics model into the driver model, which can reduce the computational effort of the planning and control process and improve the real-time performance of the system. Meanwhile, using an accurate dynamics model as a predictive model can improve the driver model’s ability to predict the future behavior of the vehicle. The most unique feature of MPC is the addition of multiple constraints to the control process.

The structure of the control algorithm for the MPC driver model is as shown in Figure 8 and divided into three main parts: the linear error model, the system constraints, and the objective function. The linear error equation provides a mathematical description of a trajectory tracking control system. The system constraints include vehicle actuator constraints, control volume smoothing constraints, and vehicle stability constraints. The design of the objective function requires a combination of fast and smooth trajectory tracking.

#### 4.2. Construction of the Simulation

In order to compare the difference between the MPC driver model and the human driver on the same road, an MPC-based driver model was established using CarSim and Simulink. Each simulation...
model is composed of three parts: vehicle model, road environment, and driver model. CarSim is a tool that can conveniently and flexibly define the test environment and test process and define the characteristic parameters and characteristic files of each system of the vehicle in detail. The same road environment as the test road is established in CarSim, and the vehicle model and driver model are connected to CarSim through Simulink. This paper uses a nonlinear model of vehicle dynamics based on the assumption of a small front wheel deflection angle and a linear tire model as follows:

\[
m\ddot{y} = -m\dot{x}\dot{\phi} + 2\left[ C_{\text{cl}}\left( \delta - \frac{\dot{y} + a\phi}{x} \right) + C_{\text{cr}}\frac{b\dot{\phi} - \dot{y}}{x} \right],
\]

\[
m\ddot{x} = m\dot{y}\dot{\phi} + 2\left[ C_{\text{cl}}\delta + C_{\text{cl}}\left( \delta - \frac{\dot{y} + a\phi}{x} \right) \delta + C_{\text{cr}}\right],
\]

\[
I\ddot{\phi} = 2\left[ aC_{\text{cl}}\left( \delta - \frac{\dot{y} + a\phi}{x} \right) - bC_{\text{cr}}\frac{b\dot{\phi} - \dot{y}}{x} \right],
\]

\[
\dot{Y} = \dot{x}\sin \phi + \dot{y}\cos \phi,
\]

\[
\dot{X} = \dot{x}\cos \phi - \dot{y}\sin \phi.
\]

In this unit, state variables are selected as follows:

\[
\xi_{\text{dyn}} = [\dot{y}, \dot{x}, \phi, \dot{\phi}, Y, X]^T. \tag{7}
\]

Control variables are selected as follows:

\[
U_{\text{dyn}} = \delta_f. \tag{8}
\]

The linear time-varying equation obtained by linearizing is as follows:

\[
\xi_{\text{dyn}} = A_{\text{dyn}}(t)\xi_{\text{dyn}}(t) + B_{\text{dyn}}(t)u_{\text{dyn}}(t). \tag{9}
\]

with

\[
B_{\text{dyn}}(t) = \frac{\partial f_{\text{dyn}}}{\partial u_{\text{dyn}}} \left[ \dot{\xi}_t, u_t \right] = \left[ \begin{array}{cccc}
2C_{\text{cl}} & 2C_{\text{cl}}(2\delta_{f,j-1} - ((\dot{y}_t + a\phi_t)/\dot{x}_t)) & 0 & 2aC_{\text{cl}}/I_z \\
\frac{m}{C_{\text{cr}}} & \frac{m}{C_{\text{cr}}} & 2bC_{\text{cr}}/I_z & 0
\end{array} \right],
\]

\[
A_{\text{dyn}}(t) = \frac{\partial f_{\text{dyn}}}{\partial \xi_{\text{dyn}}} \left[ \dot{\xi}_t, u_t \right] = \left[ \begin{array}{cccc}
\frac{-2(C_{\text{cl}} + C_{\text{cr}})}{m\dot{x}_t} & \frac{\partial \dot{f}_y}{\partial \dot{x}} & 0 & -\dot{x}_t + \frac{2(bC_{\text{cr}} - aC_{\text{cl}})}{m\dot{x}_t} \\
\frac{\partial \dot{f}_x}{\partial \dot{x}} & \frac{\partial \delta_f}{\partial \dot{x}} & -\dot{y}_t + \frac{2(aC_{\text{cl}} - \delta f_{j-1})}{m\dot{x}_t} & 0
\end{array} \right],
\]

\[
\frac{\partial}{\partial \dot{x}} \left( \begin{array}{c}
f_y \\
f_x \\
f_\phi \\
\dot{f}_y \\
\dot{f}_x \\
\dot{f}_\phi
\end{array} \right) = \left( \begin{array}{c}
2C_{\text{cl}}(\dot{y}_t + a\phi_t) + 2C_{\text{cr}}(\dot{y}_t - b\phi_t) \\
2C_{\text{cl}}\delta_{f,j-1}(\dot{y}_t + a\phi_t) \\
2C_{\text{cl}}\delta_{f,j-1}(\dot{y}_t + a\phi_t) \\
\frac{2C_{\text{cl}}(\dot{y}_t + a\phi_t)}{m\dot{x}_t} \\
\frac{2C_{\text{cl}}(\dot{y}_t + a\phi_t)}{m\dot{x}_t}
\end{array} \right),
\]
The objective function of the model predictive controller is as follows:

\[
J(\xi_{\text{dyn}}(k), u_{\text{dyn}}(k-1), \Delta u_{\text{dyn}}(k)) = \sum_{i=1}^{N_p} \left\| \eta_{\text{dy}n}(k+i|t) - \eta_{\text{dyn},\text{ref}}(k+i|t) \right\|_Q^2 + \sum_{i=1}^{N_p-1} \left\| \Delta u_{\text{dyn}}(k+i|t) \right\|_R^2 + \rho \varepsilon^2.
\]

Among them, \(N_p\) is the prediction range; \(N_c\) is the control layer; \(\varepsilon\) is the relaxation factor; \(\rho\) is the weight coefficient of the \(\varepsilon\); and \(Q\) and \(R\) are weight matrix. The first two items in the objective function reflect the system’s fast-tracking ability and the requirements for smooth changes in the front wheel angle. Since the prediction model is a complex vehicle dynamics model, which will affect the continuity of the system output, a relaxation factor is introduced into the objective function. In each loop, the trajectory tracking control algorithm should consider the following optimization issues:

\[
\begin{align*}
\min_{\Delta U_{\text{dyn},t}} & \sum_{i=1}^{N_p} \left\| \eta_{\text{dy}n}(t+i|t) - \eta_{\text{dyn},\text{ref}}(t+i|t) \right\|_Q^2 + \sum_{i=1}^{N_p-1} \left\| \Delta u_{\text{dyn}}(t+i|t) \right\|_R^2 + \rho \varepsilon^2, \\
\text{s.t.} & \Delta U_{\text{dyn},\text{min}} \leq \Delta U_{\text{dyn},t} \leq \Delta U_{\text{dyn},\text{max}}, \\
& U_{\text{dy}n,\text{min}} \leq A \Delta U_{\text{dy}n,t} + U_{\text{dy}n,t} \leq U_{\text{dy}n,\text{max}}, \\
& y_{hc,\text{min}} \leq y_{hc} \leq y_{hc,\text{max}}, \\
& y_{sc,\text{min}} - \varepsilon \leq y_{sc} \leq y_{hs,\text{max}} + \varepsilon, \\
& \varepsilon > 0.
\end{align*}
\]

In each loop, the optimization problem is solved to obtain the ideal control input increment sequence \(N_c\):

\[
\Delta U_{\text{dyn},t}^* = \left[ \Delta U_{\text{dyn},t}^*, \Delta U_{\text{dyn},t+1}^*, \ldots, \Delta U_{\text{dyn},t+N_c-1}^* \right]^T.
\]

Add the first element of this sequence to the last control, and we obtain the final control as follows:

\[
u_{\text{dy}n}(t) = u_{\text{dy}n}(t-1) + \Delta U_{\text{dy}n,t}^*.
\]

This loop realizes the tracking control of the required trajectory. The construction of the joint simulation platform is shown in Figure 9.

4.3. Parameters of the MPC-Based Driver Model. The selection of the MPC controller parameters will directly affect the performance of the driver model. The allowed ranges for each parameter of the MPC controller, the vehicle dynamics parameters, and the system constraints are presented in Table 6. The median trajectory of the driver’s driving trajectory at each speed is considered as the desired trajectory at each speed. We aimed to compare the steering wheel angles obtained from the MPC-based driver model with those of a human driver by traversing all parameters.

4.4. MPC Driver Model versus Skilled Drivers. After the simulation is complete, we will calculate the similarity between the steering wheel angle of the MPC driver model and that of skilled human drivers. Taking the steering wheel sequence of skilled drivers at 1 specified speed and the steering wheel sequence of the MPC model at the same speed as an example, the details of the process of calculating the similarity using LCSS are presented. Assume that there are a total of \(N\) sets of skilled drivers steering wheel angle at 1 specified speed. Through simulation, the MPC driver model yields a total of \(M\) sets of the steering wheel angle. For one piece of data \(i\) in a series of \(M\) steering wheel angles, its similarity to the steering wheel angles of all skilled drivers (\(N\) sets) is defined as follows:

\[
\text{LCSS}_i = \frac{1}{N} \sum_{j=1}^{N} \text{LCSS}(i, j), \quad j = 1, 2, \ldots, N,
\]

where \(\text{LCSS}_i\) denotes the steering wheel angle similarity between the \(i\)th MPC driver model and the skilled driver. This section also uses \(\text{LCSS}_{\text{max}}, \text{LCSS}_{\text{min}}, \text{and LCSS}_{\text{ave}}\) to denote the maximum, minimum, and average values of \(\text{LCSS}_i\), respectively.
LCSS\text{max} = \max (\text{LCSS}_i)
\begin{align}
\text{LCSS}_\text{min} &= \min (\text{LCSS}_i), \quad i = 1, 2, \ldots, M. \\
\text{LCSS}_\text{ave} &= \frac{1}{M} \sum_{i} \text{LCSS}_i
\end{align}

(16)

Table 7 presents the calculated similarity between the steering wheel angle of the MPC-based driver model and that of the skilled drivers on the test road at the specified speed of 40 km/h where the prediction model for the kinematic-based MPC driver model is the vehicle kinematic model.

As can be seen in Table 7, the dynamics-based MPC driver model is more similar to the skilled driver than the kinematic-based MPC driver model. To demonstrate the similarities more visually, this paper located the most similar pair of steering wheel angle sequences to skilled drivers in the dynamics-based MPC driver model and these have been plotted in Figure 10.

It should be noted that the steering wheel angles of skilled drivers used in Figure 10 do not belong to the same driver. Obviously, the steering wheel angle of skilled drivers differs considerably from that of the MPC-based model. It is difficult to demonstrate the different driving habits and driving characteristics of different drivers with only an MPC-based driver model.

4.5. Limitations of the MPC Driver Model. The MPC driver model enables the imitation of human drivers following a trajectory based on a given desired trajectory; however, there are some assumptions of ideal conditions in the design. The MPC driver model relies heavily on a predetermined desired trajectory, but no evidence exists to suggest that human drivers have a clear desired trajectory in mind when driving.
In addition, the design of the prediction model, constraints, and controller parameters have an impact on the performance of the MPC-driven model. If the predictive model becomes so simple, then the model will not predict the future state of the vehicle and if so complex, it will increase the computational effort of the system. For the establishment of constraints, in addition to control volume constraints and control increments, vehicle dynamics constraints and road constraints also need to be added. The selection of these constraints is a guarantee for the safe and smooth driving of the vehicle. Nevertheless, these constraints may conflict with each other, resulting in an inability to solve the optimization problem. In summary, due to the limitations of the abovementioned factors, the MPC-based driver model is difficult to reflect the real driving characteristics of different drivers.

Since MPC-based driver models make it difficult for autonomous vehicles to perform lane changes as well as human drivers, we propose a human-like driver lane change model. The driver model considered in this paper is a free road driver model. When driving on free roads, the driving behavior of human drivers depends mainly on their own driving experience and driving habits, without a clear desired trajectory in their minds. Generally, vehicles do not come to a sudden stop and do not move instantaneously. This means that the trajectory of the vehicle is continuous and so are the steering angles of the driver. Consequently, a driver model can be established through a sequential data modeling approach. The framework of the human-like driver model is demonstrated in Section 5.

5. Human-Like Driver Model Based on Bi-LSTM

This section proposes a human simulated lane-changing driver model, in which skilled drivers change lanes according to their subjective wishes, i.e., driving experience and habits, without external interference. From the analysis in Section 3.3, the trajectory and steering wheel angle of skilled drivers is stable and continuous. The driver model can be established by the modeling method of sequence data, and the factors affecting the driver’s lane-changing manipulation are vehicle speed, historical trajectory, and historical steering wheel angle.

5.1. Establishment of the Human-Like Driver Model. The lane-changing driving data collected in the test in Chapter 2 are sorted into sequence data with the characteristics of skilled drivers’ lane-changing behavior. The research object of this paper is the steering wheel angle of skilled drivers when changing lanes. The sequence modeling is carried out, and a human-like driver model using the LSTM network is established.

5.1.1. Recurrent Neural Network (RNN). Human behavior will inevitably be affected by history, such as some common sense, experience, and so on. Historical information plays an important role in human decision-making. The RNN is also called a sequence model to process sequence data. The driver’s driving behavior can also be regarded as sequence data, and the driver’s future behavior is related to the historical behavior.

The RNN is a network-containing cycle, which can transfer information from the previous step to the next step. The RNN structure is shown in Figure 11. The formula is as follows:

\[
\begin{align*}
y_t &= g(V \cdot h_t), \\
h_t &= f(U \cdot x_t + W \cdot h_{t-1}).
\end{align*}
\]

5.1.2. Long Short-Term Memory (LSTM). Using an LSTM network can solve the problem that optimization cannot continue when the gradient is zero or infinite. The input gate, forgetting gate, and output gate can retain or forget the data in the memory unit, which makes the LSTM not rely on too long sequences.

Hypothesis \( x \) is the input of LSTM, \( y \) is the output result of the model, \( i_t, o_t, \) and \( f_t \) represent the output of the input gate, output gate, and forgetting gate in the \( t \) time memory unit, respectively. \( c_t \) and \( m_t \) represent the activation state of neurons and memory units at time \( t \), respectively. \( W \) and \( B \)
represent the weights and offsets of variables in the network input layer, output layer, and memory unit, respectively. The memory unit in LSTM is updated iteratively in the following ways, and its structure is shown in Figure 12:

\[
\begin{align*}
i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b), \\
f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b), \\
c_t &= f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b), \\
O_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_{t-1} + b), \\
m_t &= O_t \odot h(c_t), \\
y_t &= W_{ym}m_t + b.
\end{align*}
\]

(18)

5.1.3. Bi-LSTM. Bi-LSTM (bi-directional LSTM) is an extension of traditional LSTM, which can improve the model performance in sequence prediction. LSTM can only predict the output value of the next time based on the timing information of the previous time. Sometimes, the output of the current time is related not only to the previous information but also to the subsequent information. The bi-LSTM structure is composed of two LSTMs, and the output at each time is determined by LSTM states in different directions. The structure of bi-LSTM is shown in Figure 13, and the calculation process of bi-LSTM is as follows:

\[
\begin{align*}
\overrightarrow{h_t} &= \text{LSTM}_{FW}(\overrightarrow{h_{t-1}}, x_t), \\
\overleftarrow{h_t} &= \text{LSTM}_{BW}(\overleftarrow{h_{t+1}}, x_t), \\
h_t &= \left(\overrightarrow{h_t}, \overleftarrow{h_t}\right).
\end{align*}
\]

(19)

5.1.4. Gated Recurrent Unit (GRU). GRU NN was first proposed in 2014 and is a simplified form of LSTM NN. There are many commonalities between the LSTM NN and the GRU NN. The structure of the GRU NN memory unit is shown in Figure 14. In this structure, \(z_t\) is the update gate, \(r_t\) is the reset gate, \(h_t\) is the GRU NN hidden layer state, \(h_{i_t}\) is the input, and \(\overline{h}_t\) is the candidate value added to the current state; the calculation formula of the update gate \(z_t\) is as follows:

\[
\begin{align*}
z_t &= \pi\sigma(U_zx_t + W_zh_{t-1}), \\
r_t &= \sigma(U_rx_t + W_rh_{t-1}), \\
\overline{h}_t &= \text{tanh}(U_hx_t + W_h(h_{t-1} \odot r_t)), \\
h_t &= (1 - z_t) \odot h_{i_t} + z_t \odot \overline{h}_t.
\end{align*}
\]

(20)
5.2. Input and Output of Human-Like Driver Model. Since the trajectory and steering wheel angle of skilled drivers change continuously, a driver model based on sequence data is established. The factors affecting the driver’s lane-changing manipulation are vehicle speed, historical trajectory, and historical steering wheel angle. Accordingly, the inputs of the human-like lane-changing driver model are as follows:

\[ X(t) = [v_t, d_{x,t}, \delta_t, \text{label}] \]  

where \( v_t \) is the vehicle speed at time \( t \), \( d_{x,t} \) is the lateral offset at time \( t \), and \( \delta_t \) is the steering wheel angle at time \( t \) and the lane change behavior label of the vehicle at time \( t \). The outputs of the human-like lane-changing driver model are as follows:

\[ y(t) = [\delta_{t+1}]. \]  

5.3. Training of the Human-Like Driver Model. In order to verify the performance of the human-like driver model proposed in this paper, the following models are selected for comparison: BP, LSTM, GRU, and Bi-LSTM models. This paper selects two indicators that can characterize the prediction performance of the model, which are root mean square error (RMSE) and mean absolute error (MAE). The expressions of RMSE and MAE are as follows:

\[ \text{RMSE} = \frac{1}{T} \sum_{t=1}^{T} (\delta_t - \hat{\delta}_t)^2, \]
\[ \text{MAE} = \frac{1}{T} \sum_{t=1}^{T} |\delta_t - \hat{\delta}_t|. \]  

Among them, \( \hat{\delta}_t \) is the predicted steering wheel angle of the target vehicle at time \( t \) and \( \delta_t \) is the true steering wheel angle of the target vehicle at time \( t \).

For each BP, LSTM, GRU, and Bi-LSTM neural network models, there are two hidden layers: one is two hidden layers with 50 nodes \( h_{50} \times h_{50} \); the other one also has two hidden layers with 100 nodes \( h_{100} \times h_{100} \). Before model training, the characteristic data are normalized, and the processed value is between 0 and 1; then complete the training of standard BP network, LSTM network, GRU network, and Bi-LSTM network models, respectively. Instep refers to the input time step parameter, whose value is 5, 10, or 15. Different values have different effects on the prediction performance. Outstep indicates that the output time step is set to 15; the parameter learning rate of the model is 0.001 and the number of training steps is 3000. The algorithm in this paper takes TensorFlow as the framework and is implemented by Keras.

6. Simulation and Results

6.1. Prediction Performance in the Test Set. The prediction performance of the BP neural network, LSTM, GRU, and Bi-LSTM models on the test set is shown in Tables 8 and 9. The structure of the neural network is \( h_{50} \times h_{50} \times h_{100} \times h_{100} \). The RMSE and MAE of the BP neural network reach the minimum value when the structure is \( h_{50} \times h_{50} \) and the input step size is 15. The RMSE and MAE of other neural networks LSTM, GRU, and Bi-LSTM reach the minimum value when the structure is \( h_{100} \times h_{100} \) and the input step size is 15. Among them, Bi-LSTM has the best prediction performance. RMSE = 0.229 and MAE = 0.178 show that Bi-LSTM has the strongest ability to understand sequence data. However, the structure of Bi LSTM is more complex, which means that the computational cost is relatively high. The accuracy of Bi-LSTM and LSTM models is higher than that of GRU model, which shows that although GRU has a simple structure and low calculation cost, the prediction accuracy is not as high as that of the LSTM structure. With the increase of the input step size, the RMSE of each model is gradually decreasing, indicating that the prediction accuracy of the model increases with the increase of steps. The prediction accuracy of the three driver models based on RNN model is higher than that based on BP neural network. This shows that the driver model based on RNN can perform lane-changing behavior similar to skilled drivers.

Since the prediction accuracy of the three RNN-based network models is higher than that of the BP neural network-based model, the next experimental design was conducted for the three RNN-based network models, and the model structure with the better prediction structure performance (input step = 15 and structure = \( h_{100} \times h_{100} \)) in Tables 8 and 9 was selected. The model contains a total of 4 layers of sequential deep learning models, first accessing an LSTM layer (the control group used a GRU as well as Bi-LSTM). A dropout regularization layer was accessed in the second layer to discard 50% of the neurons to reduce redundant links, another LSTM layer was accessed in the third layer (GRU as well as Bi-LSTM were used in the control group), and a normalization layer was accessed in the fourth layer to end. The model is applied to the test set, the accuracy is calculated and the results of the model iteration process are counted, and the RMSE and MAE of the epoch process are recorded for model parameter selection. Finally, the parameters were designed to be performed as paired control tests to compare and analyze the test data results and continuously improve the model. Subsequently, the differences between LSTM, GRU, and Bi-LSTM are found, and the data are recorded and the optimal model is saved, and the test results are presented in Table 10.

After the first set of test, it was found that there was overfitting of the model, and the test parameters were adjusted according to the iterative results, and epoch was adjusted to 3000; then, other parameters remained
parameter improvement and record it. To ensure that the variant for parameter tuning to find the direction of possible core layer, so the next control core layer is Bi-LSTM in-group of experiments, Bi-LSTM has better expression as the recorded. According to the results of the third and fourth model were tested separately while keeping all parameters of the second test procedure unchanged, and the results were the second round of test solved the overfitting problem and the model fitted normally. In order to get a better and improved model, the LSTM of the core layer was changed, so the third and fourth sets of test were initiated. The expression effects of the GRU model and the Bi-LSTM model were tested separately while keeping all parameters of the second test procedure unchanged, and the results were recorded. According to the results of the third and fourth group of experiments, Bi-LSTM has better expression as the core layer, so the next control core layer is Bi-LSTM invariant for parameter tuning to find the direction of possible parameter improvement and record it. To ensure that the number of iterations did reach the best-fit default value, the fifth and sixth sets of tests were designed to test the expression effect of Bi-LSTM in epoch+ and epoch-, respectively, and it was found that the expression effect of the model became worse under the conditions of constant other parameters.

Therefore, the subsequent experiments were conducted with epoch = 3000 as the default value, and the batch size was adjusted to 64 in the seventh set of experiments. By observing the feedback results, it was found that the expression effect on the test set was not satisfactory, and the model might have overfitting tendency at this time. When the batch size expanded, the number of epoch iterations should be reduced accordingly to ensure the best fitting effect. In the 8th group of tests, an attempt was made to reduce the strength of the dropout and retain more neuronal connections. The model also produced a tendency to overfit and changes to the percentage of dropout layers were made, but after testing, it was found that a 50% random discard level is still a good level to prevent overfitting and its strength should not be reduced.

By analyzing the experimental results, it is found that adjusting the model parameters when the core layer is Bi-LSTM, the batch-size is 32, epoch = 3000, and dropout = 0.5, an approximate optimal model with RMSE = 0.098 and MAE = 0.068 can be obtained. Overall, the prediction accuracy of the Bi-LSTM-based human-like driving model is very high. Combining the comparison of the LCSS results between the MPC-based driver model and human drivers, it is concluded that the steering wheel angle of the human-like driver model proposed in this paper is closer to the actual steering wheel angle of human drivers during lane changes.

### Table 8: $h_{50} \times h_{50}$ Prediction performance in test set.

<table>
<thead>
<tr>
<th>Model</th>
<th>instep</th>
<th>RMSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP</td>
<td>5</td>
<td>1.767</td>
<td>1.232</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.568</td>
<td>1.065</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1.392</td>
<td>0.957</td>
</tr>
<tr>
<td>LSTM</td>
<td>5</td>
<td>0.859</td>
<td>0.636</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.683</td>
<td>0.515</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.485</td>
<td>0.360</td>
</tr>
<tr>
<td>GRU</td>
<td>5</td>
<td>0.428</td>
<td>0.326</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.421</td>
<td>0.313</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.417</td>
<td>0.312</td>
</tr>
<tr>
<td>Bi-LSTM</td>
<td>5</td>
<td>0.411</td>
<td>0.306</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.289</td>
<td>0.226</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.256</td>
<td>0.192</td>
</tr>
</tbody>
</table>

### Table 9: $h_{100} \times h_{100}$ prediction performance in test set.

<table>
<thead>
<tr>
<th>Model</th>
<th>instep</th>
<th>RMSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP</td>
<td>5</td>
<td>1.542</td>
<td>1.056</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.531</td>
<td>1.055</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1.426</td>
<td>0.995</td>
</tr>
<tr>
<td>LSTM</td>
<td>5</td>
<td>0.348</td>
<td>0.313</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.263</td>
<td>0.206</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.226</td>
<td>0.178</td>
</tr>
<tr>
<td>GRU</td>
<td>5</td>
<td>0.728</td>
<td>0.543</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.410</td>
<td>0.310</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.346</td>
<td>0.242</td>
</tr>
<tr>
<td>Bi-LSTM</td>
<td>5</td>
<td>0.336</td>
<td>0.216</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.226</td>
<td>0.184</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.122</td>
<td>0.156</td>
</tr>
</tbody>
</table>

By analyzing the experimental results, it is found that adjusting the model parameters when the core layer is Bi-LSTM, the batch-size is 32, epoch = 3000, and dropout = 0.5, an approximate optimal model with RMSE = 0.098 and MAE = 0.068 can be obtained. Overall, the prediction accuracy of the Bi-LSTM-based human-like driving model is very high. Combining the comparison of the LCSS results between the MPC-based driver model and human drivers, it is concluded that the steering wheel angle of the human-like driver model proposed in this paper is closer to the actual steering wheel angle of human drivers during lane changes.

6.2. Effect of Inputs on Model Performance. Table 11 shows the effect of inputs on model performance. The input characteristics of the first line are $[6, \delta, \text{label}]$, indicating that only the steering wheel angle history sequence entered; the second line adds the speed history sequence on this basis; the model input displayed in the third line is the historical sequence of the steering wheel angle and lane change trajectory; the fourth line is the input of Bi-LSTM model in this paper. The prediction accuracy of these models with different input characteristics is compared, and the human-like lane-changing driver model is systematically evaluated.

The results in Figure 15 show that the prediction accuracy of the model considering only some input characteristics decreases. In particular, the model that only considers the historical sequence of steering wheel angle has the worst prediction effect. The model accuracy of the second row is lower than that of the third row, indicating that the lateral offset of vehicles has a greater impact on lane-changing behavior. The accuracy of the model considering all input features is the highest.

6.3. Prediction Performance in the Validation Test. The driving data of skilled drivers who did not participate in the model training were used to test the generalization ability of the human-like driver model. The steering wheel angle of the
skilled driver is taken as the real value. Figure 16 demonstrates the test results of three RNN-based driver models. Since the prediction accuracy of the BP neural network model is much lower than that based on the RNN model, three models based on RNN are selected for comparison in this section. The input step of each model is 15 and the network structure is $h_{100} \times h_{100}$. The results show that the prediction results of the human-like driver model based on LSTM, GRU, and Bi-LSTM are very close to the steering wheel angle of the real driver. It is found that the prediction accuracy of the Bi-LSTM model is the highest, and the numerical error between steering wheel angle and the real skilled driver is the smallest.

In order to further study the human-like degree of the driver model based on the RNN, this paper also uses LCSS to compare the similarity of the three steering wheel angle sequences in Figure 16. The similarity between the real value and the standard RNN LCSS$_{RNN}$ is 0.0773, the similarity with LSTM LCSS$_{LSTM}$ is 0.0452, the similarity with Bi-LSTM LCSS$_{Bi-LSTM}$ is 0.0397, and the similarity with GRU LCSS$_{GRU}$ is 0.0614. Figure 17 demonstrates the prediction performance of standard RNN, LSTM, and GRU driver models on the validation set, and the node configuration of these three models is $h_{100} \times h_{100}$.

Similar to the prediction results of the test set, the prediction results of the Bi-LSTM model are closest to the real value in the validation set. However, compared with the validation set, RMSE and MAE increased slightly on the validation set. The results show that the driver model based on Bi-LSTM proposed in this paper not only has strong generalization performance but also achieves a good degree of human imitation. Based on the Bi-LSTM human-like driver model, the steering wheel angle sequence very similar to the skilled driver can be generated to realize the human-like steering control of driverless vehicles.

7. Conclusions

In this study, the MPC-based driver model is simulated jointly in CarSim/Simulink and a vehicle dynamics model is used. LCSS was introduced in order to find the most similar steering wheel angle between the simulation results and the skilled driver. The comparison results indicate that there are significant differences between MPC-based driver models and skilled drivers, and it is difficult for MPC-based driver models to reflect the driving habits and driving characteristics of
different humans. An RNN-based human-like lane change driver model is proposed in order to make intelligent vehicle behave more like human drivers. We collected lane change maneuver data from 10 skilled drivers at 4 specified speeds (20 km/h, 30 km/h, 40 km/h and 50 km/h) and then transformed all collected driving situations into sequential data with same format. The human-like lane change driver model was developed based on multivariate multistep LSTM, GRU, and Bi-LSTM. By comparing the prediction results of RMSE, MAE, and LCSS, it is shown that the Bi-LSTM driver model has the best prediction performance. Finally, the generalization ability of the human-like lane change driver model is tested on the validation set. The computed results of the performance metrics demonstrate that the human-like lane change driver model developed in this paper can produce a human-like lane change steering wheel.

Driver models reflect driving behavior; however, human driving behavior is complex, uncertain, and has individual differences. From the comparison results in Section 4, the MPC-based driver model provides a difficult description of human drivers of varying driving styles. Driver models built using a data-driven approach may be more suitable for human-like implementations than physical models. The human-like lane-changing driver model proposed in this paper enables intelligent vehicles to truly change lanes like human drivers, and the method can be well-applied to areas such as the vehicle lateral control.

The human-like lane-changing driver model is a fundamental model that needs to work with other algorithms in more complex traffic scenarios. The human-like driving model in this paper is primarily concerned with the generation of human-like steering wheel angles and has no control over the speed of the vehicle; the human-like driving model is required to be used in conjunction with a speed control model. In the follow-up study, the human-like driver model was developed to further investigate the influence of the driving road, driving environment, traffic conditions, and the driver’s psychological and physiological state on the driver’s operation, in addition to considering factors such as vehicle driving status, driving trajectory, and the steering wheel angle of skilled drivers. Alternatively, if autonomous vehicles want to perform lane changes on curves, the human-like driving model will need to be improved in combination with other bending algorithms, and the human-like driving model will enable autonomous vehicles to change lanes on curves just like human drivers. The extended model allows for more complex scenarios to be applied to establish more accurate human-like driving models.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this paper.

Acknowledgments

This work was financially supported by the National Natural Science Fund (No. U1564201 and No. U51675235) and the Research Innovation Program for College Graduates of Jiangsu Province (No. KYZZ16_0338).

References


