IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 August 2024, accepted 3 September 2024, date of publication 24 September 2024, date of current version 7 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3466971

RESEARCH ARTICLE

A Fast Predictive Control Method for Vehicle Path Tracking Based on a Recurrent Neural Network

XIALAI WU[1], **LING LIN**[1], **JUNGHUI CHEN**[2], **AND SHUXIN DU**[1]
[1]Huzhou Key Laboratory of Intelligent Sensing and Optimal Control for Industrial Systems, Huzhou University, Huzhou 313000, China
[2]Department of Chemical Engineering, Chung-Yuan Christian University, Taoyuan 32023, Taiwan

Corresponding authors: Junghui Chen (jason@wavenet.cycu.edu.tw) and Shuxin Du (shxdu@zjhu.edu.cn)

This work was supported in part by Zhejiang Provincial Natural Science Foundation of China under Grant LQ21F030007; in part by Huzhou Key Laboratory of Intelligent Sensing and Optimal Control for Industrial Systems under Grant 2022-17; in part by the Postgraduate Research and Innovation Project of Huzhou University under Grant 2024KYCX40; and in part by the Ministry of Science and Technology, Taiwan, under Grant NSTC 112-2221-E-033-053.

ABSTRACT The paper introduces a Fast Model Predictive Control (FMPC) approach for vehicle path tracking, addressing the challenge of real-time performance in highly nonlinear systems. Utilizing a recurrent neural network with symmetric saturating linear transfer functions (SSL-RNN), our method efficiently constructs an SSL-RNN model for the vehicle. By transforming the MPC optimal control problem into a mixed integer linear programming problem, a swift online solution is achieved. Through simulations on a CarSim/Simulink platform, our FMPC outperforms RNN-based nonlinear MPC and long-short-term memory network-based MPC, demonstrating superior accuracy in vehicle path tracking and enhanced controller solution efficiency.

INDEX TERMS Mixed integer linear programming, model predictive control, recurrent neural network, symmetric saturating linear transfer functions, vehicle path tracking.

I. INTRODUCTION

With the rapid development of driverless technology, its potential to improve traffic efficiency, reduce driver burden, and improve vehicle driving safety has become a current research hotspot in the current automotive industry [1]. As one of the core aspects of the driverless system, the performance of vehicle motion control directly affects the driving safety and user experience of the vehicle. Vehicle motion control includes trajectory tracking and path tracking. The control system calculates appropriate control commands such as acceleration, braking, and steering based on the real-time vehicle status and the target trajectory or path so that the vehicle can achieve accurate trajectory or path tracking [2]. This is also essential for ensuring collision avoidance in autonomous driving systems [3].

The path-tracking control challenge for autonomous vehicles has garnered significant attention [4]. Earlier algorithms employed for path tracking control encompass pure pursuit control [5], proportional-integral-derivative (PID) control [6], and Stanley control [7], among others. Zhang et al. introduced an enhanced pure tracking method employing fuzzy control [8]. Simulation and experimental outcomes demonstrate superior tracking accuracy and convergence in straight-line and turning path tracking, surpassing traditional pure tracking control algorithms. However, the pure tracking control method overlooks vehicle dynamics' influence during path tracking. Chen et al. proposed an adaptive fuzzy PID algorithm for path tracking control of a novel 4WIS (four-wheel-independent-steering) electric vehicle [9]. Simulation results indicate enhanced path tracking performance and robustness compared to traditional PID control. Despite the improved PID control's performance enhancements, it remains incapable of handing system nonlinearity and constraints. Yang et al. presented an enhanced Stanley method for curve path tracking [10]. This method forecasts not only the relative position of the closest point on the path but also the direction of the gaze point. Research findings validate its effectiveness in curve environments, albeit its applicability to high-speed conditions poses challenges.

The associate editor coordinating the review of this manuscript and approving it for publication was Razi Iqbal.

In pursuit of enhancing path tracking performance, vehicle model-based control methods are also under scrutiny. Fan and Chen introduced a path following control method based on linear quadratic regulator (LQR) principles [11]. Balanced weighting of state variables and input weights achieves optimal quadratic performance index. Simulation and field tests affirm stability and rapid convergence of the proposed method. However, LQR controllers are constructed on linear models, rendering them vulnerable to robustness issues. Kapania and Gerdes proposed a feedback-feedforward steering controller [12] adept at maintaining vehicle stability under extreme conditions and minimizing lateral path tracking deviations. However, data collection under extreme conditions poses challenges, necessitating expensive sensors arrays. Pan et al. introduced a model-free adaptive dynamic programming method for autonomous vehicle path tracking amidst actuator faults [13]. Adaptive regulators mitigate the effects of actuator faults, modeling errors, and curvature interference on the vehicle system. While the model-free control method's structural simplicity is notable, stability analysis of the control system proves challenging. Yao and Ge introduced a path tracking method employing deep reinforcement learning (DRL) for autonomous vehicles [14]. Leveraging the deep deterministic policy gradient algorithm of the double critic network, the controller undergoes offline learning to achieve reference route tracking. Results underscore the method's environmental adaptability and tracking performance. However, the computational intensity and time consumption associated with training and debugging DRL-based methods warrant consideration.

The MPC algorithm has been widely studied in path-tracking control due to its ability to predict future trajectories and its advantage in handling multiple constraints [15], [16]. Some studies have used vehicle mechanism models in MPC-based path-tracking control. Jeong and Yim proposed a four-wheel independent steering autonomous vehicle algorithm based on MPC [17]. The MPC controller was designed using a linear time-varying vehicle model, and experimental results show that the algorithm can improve the performance of path and speed-tracking, and reduce the computational complexity compared to the nonlinear MPC. Liu et al. proposed a path-tracking strategy aiming to improve the path-tracking ability and road adaptability of the hitch trailer [18]. The path-following controller was designed based on the MPC and the optimal curvature preview control technique. Simulation results show that the controller improves path tracking capability and driving stability. Wang et al. proposed a MPC for path tracking using a nominal kinematic model and Gaussian processes (GP) models to capture the unmodeled dynamics from the observational data collected in the field experiments [19]. The results show that the proposed MPC algorithm can reduce the path tracking error for different paths and is computationally more efficient. The MPC methods above based on vehicle mechanism models require accurate vehicle model and environment information, but accurate models are difficult to obtain in practice.

To overcome the drawback of the mechanism model-based MPC that is complex and difficult to implement online, data-driven methods are used to as predictive models [20]. The data-driven approach can learn the motion characteristics of the vehicle more accurately using the collected data, with low modeling difficulty. Many studies used neural network models to design MPC controllers [21]. Rokonuzzaman et al. adopted a neural network to learning vehicle dynamics by a large amount of data provided by modern vehicle systems, and integrated the neural network model into the MPC design [22]. Experimental results showed that the controller achieves better control under real road conditions. Spielberg et al. carried out the design and experimental validation of the predictive control based on a neural network model. The trained neural network model is able to predict vehicle dynamics under changing and complex operating conditions [23]. Experimental results showed that the designed predictive control can adapt to different friction conditions and track the vehicle path effectively. However, the high-quality optimal solution of MPC based on the neural network model is difficult to obtain due to the strong nonlinearity of the neural network model.

In summary, there are still challenges in the current MPC-based driverless vehicle path tracking control, including high accuracy requirements for prediction models, high computational complexity for online solutions, and difficulty for controllers to solve in real time. To address these issues, a fast predictive control method for vehicle path tracking based on a symmetric saturating linear transfer function recurrent neural network (SSL-RNN) is proposed. The method establishes a vehicle dynamic model based on the SSL-RNN, designs an MPC controller based on the model, and then transforms the MPC optimal control problem into a mixed-integer linear programming problem (MILP) for a fast solution. The SSL-RNN-based MPC will be detailed in the following sections. To verify the proposed scheme, CarSim/Simulink is used to demonstrate that it achieves more accurate control of the vehicle path and has a faster solution speed.

The rest of this paper is constructed as follows. The vehicle model is introduced in Section II. In this section, the vehicle mechanism model is first established; then the SSL-RNN-based vehicle dynamics model is designed and trained. In Section III, the model predictive controller is designed based on the vehicle SSL-RNN model, and the established MPC optimal control problem is transformed into a MILP. Simulation experiments are carried out to verify the effectiveness of the proposed controller in Section IV. Finally, the paper is concluded in Section V.

II. CONSTRUCTION OF VEHICLE MODELS

In this section, the vehicle mechanism model is first established; then the SSL-RNN-based vehicle dynamics model is designed and trained; and finally, the validation shows that the trained model is characterized by high accuracy and reliability.

A. VEHICLE MECHANISTIC MODEL

For an easy explanation of the proposed scheme. A simplified vehicle model is used to describe the motion state of the vehicle, which mainly includes information of its position, velocity, and acceleration but does not consider the internal dynamics of the vehicle. Due to the simplicity and practicality of the kinematic vehicle model, it is widely used in path-tracking control studies [19]. Under reasonable assumptions and simplification conditions, the kinematic vehicle model can be simplified to a ‘‘single-track model’’, which treats the vehicle body and suspension system as a rigid body mass model. The vehicle kinematics model based on mechanism analysis is shown in Figure 1. In the inertial coordinate system XOY, there is an instantaneous rotation center P, and the vehicle is considered to be in rotational motion only at a certain moment. Through the principles of classical mechanics, the differential equations describing the motion of the vehicle can be derived, and the motion law of the vehicle can be obtained.

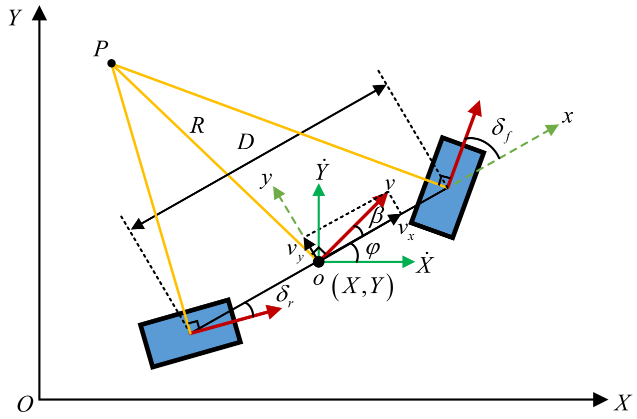

FIGURE 1. Vehicle kinematics model.

The differential equations for the planar motion of the vehicle in the inertial coordinate system XOY are as follows [19]:

$$\begin{cases} \dot{X} = v\cos(\alpha+\beta) \\ \dot{\varphi} = \dfrac{v}{R} = \dfrac{v\left(\tan\delta_f + \tan\delta_r\right)\cos\beta}{D} \\ \dot{Y} = v\sin(\alpha+\beta) \end{cases} \tag{1}$$

where $\dot{X}$ and $\dot{Y}$ represent the changes in the X and Y coordinates of the vehicle in the inertial coordinate system, respectively. The φ denotes the yaw angle of the vehicle, $\dot{\varphi}$ represents the yaw angular velocity of the vehicle, v denotes the velocity of the vehicle's center of mass, β represents the lateral deviation angle of the vehicle's center of mass, δ_f and δ_r represent the steering angles of the front and rear wheels of the vehicle, respectively, R denotes the instantaneous turning radius of the vehicle, and D represents the wheelbase of the vehicle.

It is assumed that during the autonomous steering process of the driverless vehicle, the vehicle has no lateral sliding phenomenon and the rear wheels do not steer, so the vehicle state satisfies the following equation:

$$\begin{cases} v_y \approx 0 \\ \beta = \arctan\dfrac{v_y}{v_x} = 0 \\ \delta_r \approx 0 \end{cases} \tag{2}$$

Selecting $x = [X, Y, \varphi]^T$ and $u = [v, \delta_f]^T$ as the vehicle state and control quantities, the vehicle kinematic model can be obtained by organizing Eq. (1) and Eq. (2) as:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ \frac{\tan\delta_f}{D} \end{bmatrix} v \tag{3}$$

B. VEHICLE SSL-RNN MODEL

In this study, a recurrent neural network (RNN) is utilized to effectively capture the dynamic characteristics inherent in vehicles owing to its exceptional fitting capability. The vehicle dynamic data utilized in our analysis is sourced from the widely-utilized CarSim simulation software. CarSim offers a comprehensive platform for modeling diverse aspects of vehicle behavior, encompassing vehicle dynamics, suspension systems, and tire models, rendering it a staple tool in both automotive industry and academic circles for vehicle systems analysis and design [24]. During the data collection phase, pseudo-random variables serve as the input. To ensure a broad representation of vehicle operating conditions and to enhance data diversity, careful consideration is given to the generation rule and value range of these pseudo-random variables. The schematic representation of the designed vehicle RNN model is illustrated in Figure 2, while the RNN's performance metrics are thoroughly discussed in Section II-C.

During the vehicle data acquisition, the control information of the vehicle at the current time step and the state information of the previous time steps are used as inputs to the SSL-RNN, and the state information of the vehicle at the next time step is used as outputs of the SSL-RNN. The details of the input (x^t) and the output (y^t) are shown in Table 1 and Table 2.

In the realm of RNN models, the selection of an appropriate activation function for the hidden layer holds paramount importance in determining the training efficacy of the model. In this study, we opt for the symmetric saturating linear transfer function (SSL) as the activation function of the hidden layer. Consequently, the recurrent neural network outfitted with SSL as the activation function for the hidden layer is denoted as SSL-RNN. The activation function SSL for each neuron in the hidden layer of the SSL-RNN has the following properties: It restricts the output to the range [−1,1] and saturates the output when it exceeds this range [25]. It is defined as follows:

$$f_h(x) = \max\{-1, \min\{x, 1\}\} \tag{4}$$

The activation function of each neuron in the output layer of SSL-RNN is a linear function, denoted as:

$$f_o(x) = x \tag{5}$$

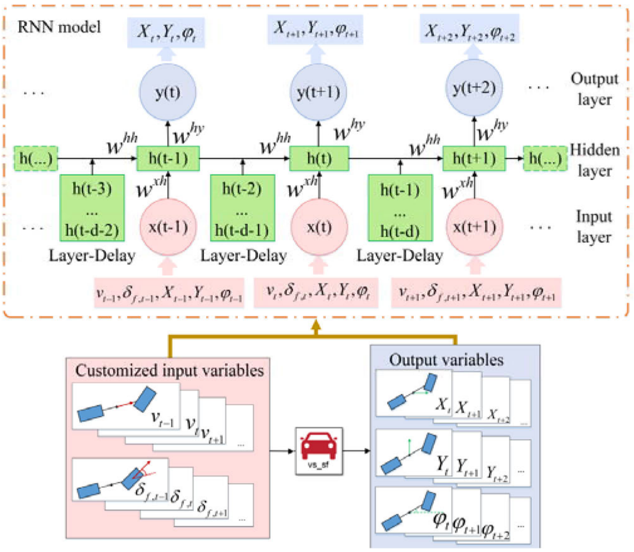
RNN model
X_t, Y_t, φ_t
$X_{t+1}, Y_{t+1}, \varphi_{t+1}$
$X_{t+2}, Y_{t+2}, \varphi_{t+2}$
y(t)
y(t+1)
y(t+2)
Output layer
w^{hh}
w^{hy}
h(...)
h(t-1)
h(t)
h(t+1)
h(...)
Hidden layer
w^{xh}
h(t-3) ... h(t-d-2)
h(t-2) ... h(t-d-1)
h(t-1) ... h(t-d)
x(t-1)
x(t)
x(t+1)
Input layer
Layer-Delay
$v_{t-1}, \delta_{f,t-1}, X_{t-1}, Y_{t-1}, \varphi_{t-1}$
$v_t, \delta_{f,t}, X_t, Y_t, \varphi_t$
$v_{t+1}, \delta_{f,t+1}, X_{t+1}, Y_{t+1}, \varphi_{t+1}$
Customized input variables
v_{t-1} v_t v_{t+1} ...
$\delta_{f,t-1}$ $\delta_{f,t}$ $\delta_{f,t+1}$ ...
vs_sf
Output variables
X_t X_{t+1} X_{t+2} ...
Y_t Y_{t+1} Y_{t+2} ...
φ_t φ_{t+1} φ_{t+2} ...

FIGURE 2. The schematic of a vehicle RNN model.

TABLE 1. Inputs to the SSL-RNN.

Variable	Name	Unit
v_{t-1}	The velocity of the previous time step	m/s
$\delta_{f,t-1}$	The front wheel steering angle of the previous time step	rad
X_{t-1}	The vehicle X-coordinate position of the previous time step	m
Y_{t-1}	The vehicle Y-coordinate position of the previous time step	m
φ_{t-1}	The vehicle yaw angle of the previous time step	rad

TABLE 2. Outputs of the SSL-RNN.

Variable	Name	Unit
X_t	The vehicle X-coordinate position of the next time step	m
Y_t	The vehicle Y-coordinate position of the next time step	m
φ_t	The vehicle yaw angle of the next time step	rad

In each time step, the linear combination of input, weight, and bias, coupled with a nonlinear transformation of the activation function, yields the neuron output of the SSL-RNN hidden layer, expressed as:

$$h_j^t = f_h\left(\sum_{i=1}^{I} w_{ij}^{xh} \cdot x_i^t + w_{ij}^{hh} \cdot h_j^{t-1} + b_j\right), j = 1, 2, \cdots, J \tag{6}$$

where t is the index of the time step, i is the index of each neuron in the input layer, I is the number of neurons in the input layer, j is the index of each neuron in the hidden layer, J is the number of neurons in the hidden layer, h_j^{t-1} is the output of the jth hidden layer neuron at the previous time step $t-1$, h_j^t is the output of the jth hidden layer neuron at time step t, d is the time delay of RNN, x_i^t is the output of each neuron in the input layer, w_{ij}^{xh} is the weight coefficient between the ith neuron in the input layer, and the jth neuron in the hidden layer, w_{ij}^{hh} is the weight coefficient between the ith neuron in the previous hidden layer and the jth neuron in the current hidden layer, and b_j is the bias of the jth neuron in the hidden layer.

The neuron output of the hidden layer of SSL-RNN is taken as the input of the output layer, and the neuron output of the output layer is obtained through the activation function:

$$y_k^t = f_o\left(\sum_{j=1}^{J} w_{jk}^{hy} \cdot h_j^t + c_k\right), k = 1, 2, \cdots, K \tag{7}$$

where k is the index of each neuron in the output layer, K is the number of neurons in the output layer, y_k^t is the output of the kth output layer neuron at the time step t, w_{jk}^{hy} is the weight coefficient between the jth neuron in the hidden layer and the kth neuron in the output layer, and c_k is the bias of the kth neuron in the output layer.

Based on the use of the piecewise function SSL as the activation function of the neurons in the hidden layer, the neural network model can be converted into a mixed integer linear formulation. This can reduce the complexity of the nonlinear model [26], [27]. Therefore, it is a good choice to use the SSL-RNN model as an alternative model for the vehicle model.

The mixed integer linearization of the piecewise function SSL; i.e. Eq. (4) can be achieved by introducing a number of auxiliary variables, including both continuous and binary variables [28]. The piecewise function SSL is composed of three linear segments with four interval breakpoints ($x_1 = a$, $x_2 = -1$, $x_3 = 1$, $x_4 = A$), as shown in Figure 3. The breakpoints a and A respectively represent the lower and upper bounds of the input of the hidden layer neuron x, and their values are obtained by a test. Then a positive continuous variable w_n is introduced for each breakpoint x_n, such that $w_n \in [0, 1]$ $n = 1, \cdots, 4$, and a binary variable q_n associated with the segment $[x_n, x_{n+1}]$ $n = 1, 2, 3$ is introduced.

At this point, for any given x value $(\tilde{x})$ with $x_n \leq \tilde{x} \leq x_{n+1}$, there are always special w_n and w_{n+1} that make the following equation true.

$$\tilde{x} = w_n x_n + w_{n+1} x_{n+1} \tag{8}$$

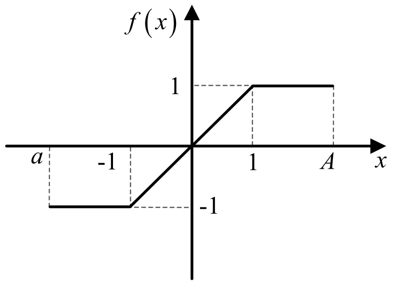

FIGURE 3. The graph of the continuous piecewise function SSL.

Accordingly, the value of the piecewise function SSL at $\tilde{x}$ is

$$f(\tilde{x}) = w_n f(x_n) + w_{n+1} f(x_{n+1}) \tag{9}$$

To force each x value to be associated with the proper pair of consecutive interval breakpoints (i.e. segment) and w_n, the following constraints are imposed.

$$\sum_{n=1}^{3} q_n = 1 \tag{10}$$

$$\begin{cases} w_1 \le q_1 \\ w_2 \le q_1 + q_2 \\ w_3 \le q_2 + q_3 \\ w_4 \le q_3 \end{cases} \tag{11}$$

The constraints in Eq. (10) imposed that only one q_n takes the value 1, i.e., $\tilde{x}$ is associated with the segment $[x_n, x_{n+1}]$. Constraint in Eq. (11) imposed that only w_n and w_{n+1} at the left and right extremes of the segment $[x_n, x_{n+1}]$ are not 0.

Based on the above, the mixed integer linear formulation of the piecewise function SSL $(f(x))$ for any given x value $(\tilde{x})$ can be obtained by imposing the following constraints:

$$\begin{cases} f(\tilde{x}) = w_1 f(a) - w_2 + w_3 + w_4 f(A) \\ w_1 + w_2 + w_3 + w_4 = 1, w_n \in [0, 1] \\ q_1 + q_2 + q_3 = 1, \ q_n \in \{0, 1\} \\ w_1 \le q_1 \\ w_2 \le q_1 + q_2 \\ w_3 \le q_2 + q_3 \\ w_4 \le q_3 \end{cases} \tag{12}$$

where $\tilde{x} = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$. The constraints in Eq. (12) ensure the correct computation of the piecewise function SSL $f(x)$.

C. MODEL TRAINING

To train the vehicle SSL-RNN model, a large number of data based on CarSim vehicle simulation software were collected, including the control signals and the corresponding motion states of the vehicle under different working conditions. To ensure the diversity of training data, pseudo-random variables adopted as inputs during data collection are designed as follows: The speed range is set to be 5 m/s-20 m/s, and the speed variation range is generated by pseudo-random PRBS signals to be [−0.2,0.2] to simulate the change of vehicle speed; the range of the front wheel steering angle is set to be −0.44rad-0.44rad, and the variation range of the front wheel steering angle generated by pseudo-random PRBS signal is [−0.1,0.1] to simulate the change of the front wheel steering angle. The sampling time is 0.1 second. After the data acquisition is completed, the data is preprocessed. First, the input data and output state data are normalized and mapped to appropriate value ranges to eliminate the differences in magnitude between different variables; then, the dataset is divided into a training dataset and a validation dataset, 18,000 sample data for the training and 2,000 sample data for the validation. The time delay in RNN training is set to 5, and the number of hidden layers is set to 20. The training algorithm is the Levenberg-Marquardt algorithm, which combines the characteristics of the gradient descent method and the Gauss-Newton method. By approximating the inverse matrix of the Hessian matrix in the algorithm, the parameters of the model are updated in a faster and more accurate way to minimize the loss function [28]. In this work, the neural network toolbox of MATLAB is used, from which the built-in function "layrecnet" is adopted to complete the modeling and training of the SSL-RNN model.

The training results of the vehicle SSL-RNN dynamic model are depicted in Figures 4-7. In these figures, the blue triangles represent the output values obtained from CarSim validation data, while the magenta dots represent the output values generated by the vehicle SSL-RNN model. A discernible observation from these plots is the close alignment between the predicted outputs of the trained SSL-RNN model and the outputs derived from CarSim data. Given that vehicle speed and steering angle in this work are discrete manipulated variables input to the vehicle, changes in speed or yaw angle occur in relatively discrete steps. This alignment underscores the high precision of the SSL-RNN model in capturing the dynamic behavior of the vehicle. Furthermore, it highlights the model's adeptness in adapting to various vehicle operating states.

III. MODEL PREDICTIVE CONTROLLER DESIGN

To design the control system, the objective expression and the constraints of MPC are first formulated; then the MPC controller is designed based on the SSL-RNN model of the vehicle. Finally, the established MPC optimal control problem is transformed into the framework of MILP to improve computational efficiency.

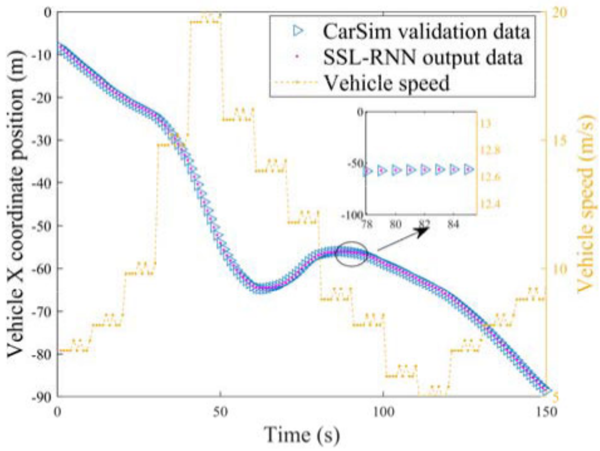

FIGURE 4. The vehicle X-coordinate position in the verification test.

A. OBJECTIVES AND CONSTRAINTS

To ensure that the autonomous vehicle can track the reference path quickly and stably, it is crucial to design the objective function in the MPC algorithm appropriately [29]. In MPC, the optimization objective of the control problem is to find the optimal control input sequence. The optimization objective

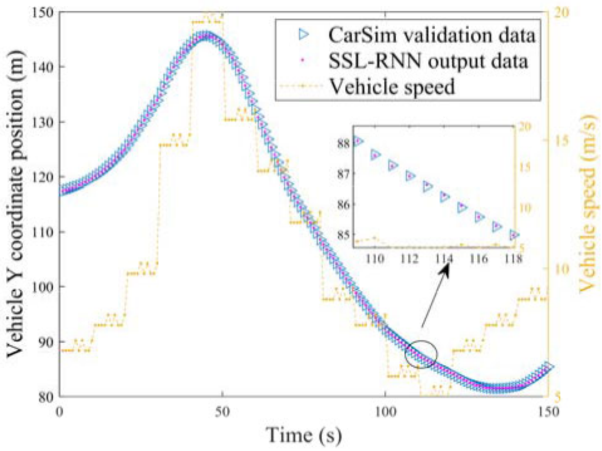

FIGURE 5. The vehicle Y-coordinate position in the verification test.

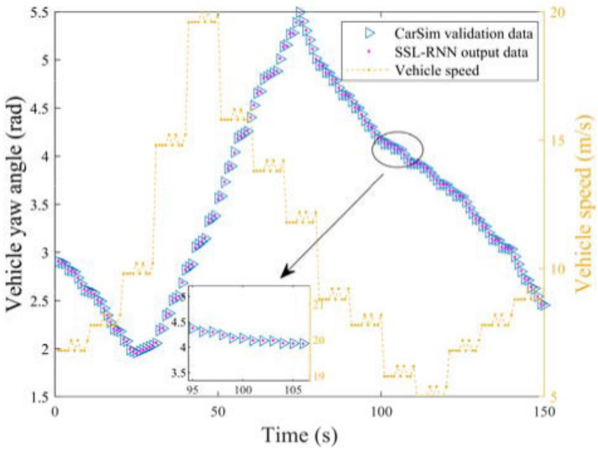

FIGURE 6. The vehicle yaw angle in the verification test.

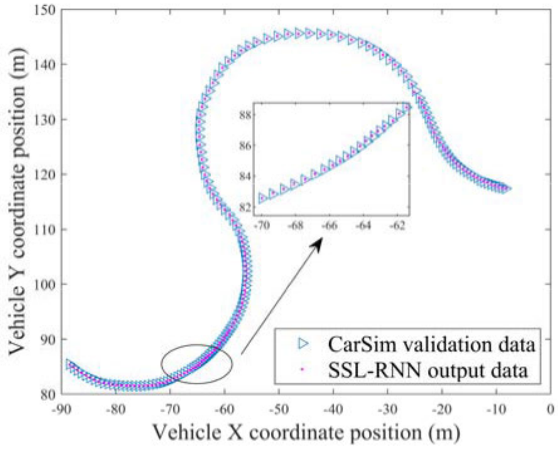

FIGURE 7. Trajectory in verification test.

in this work is to minimize the deviation of the system state trajectory, i.e. the difference between the current predicted state of the vehicle and the reference path, to make the vehicle as close as possible to the reference path and achieve accurate path tracking. The objective of the controller is expressed as:

$$C = \sum_{t=1}^{N_p} p_t \left| X_t - X_{ref}^t \right| + \sum_{t=1}^{N_p} r_t \left| Y_t - Y_{ref}^t \right| + \sum_{t=1}^{N_p} s_t \left| \varphi_t - \varphi_{ref}^t \right| \tag{13}$$

where t is the index of the prediction horizon, N_p is the total prediction horizon, X_t, Y_t and φ_t are respectively the predicted values of the X-coordinate position of the vehicle, the predicted values of the Y-coordinate position of the vehicle, and the predicted values of the yaw angle of the vehicle at the time t, X_{ref}^t, Y_{ref}^t and φ_{ref}^t are respectively the reference values of the X -coordinate position of the vehicle, the Y-coordinate position of the vehicle, and the yaw angle of the vehicle at time t, and p_t, r_t and s_t represent the weight coefficients of the error term of the objective function in each prediction step. To realize the adaptive change of the weight coefficients in the error term of each step, the weight coefficients are specifically defined as [30] and [31]:

$$\begin{cases} p_t = \dfrac{1}{t}, t = 1, 2, \cdots, N_p \\ r_t = \dfrac{1}{2N_p - 1} t, \ t = 1, 2, \cdots, N_p \\ s_t = \sqrt{\dfrac{4N_p}{N_p - t + 1}} N_p, \ t = 1, 2, \cdots, N_p \end{cases} \tag{14}$$

In the design of a predictive control algorithm based on the vehicle SSL-RNN model, MPC needs to add control quantity constraints to ensure the safety and performance of vehicle control. The specific constraints are described as follows:

1) Limit the range of vehicle speed to ensure that the vehicle runs within the safe speed. Here, the maximum speed constraint of the vehicle is set to 20 m/s, and the minimum speed constraint is 5 m/s, expressed as:

$$v_{\min} \le v_t \le v_{\max}, t = 1, 2, \cdots, N_c \tag{15}$$

where N_c is the control horizon.

2) The maximum front wheel steering angle constraint of the vehicle $\delta_{f,\max}$ is set to 0.44 rad, and the minimum front wheel steering angle constraint $\delta_{f,\min}$ is set to -0.44 rad. In addition, to ensure the comfort and safety of the vehicle, the deviation $\Delta\delta_f$ of the front wheel steering angle from the previous time step to the next time step is limited to -0.1 rad-0.1 rad, expressed as:

$$\begin{cases} \delta_{f,\min} \le \delta_{f,t} \le \delta_{f,\max}, \ t = 1, 2, \cdots, N_c \\ \Delta\delta_{f,t} = \delta_{f,t-1} - \delta_{f,t} \\ -0.1 \le \Delta\delta_{f,t} \le 0.1, t = 1, 2, \cdots, N_c \end{cases} \tag{16}$$

B. OPTIMAL CONTROL PROBLEM

Based on the vehicle SSL-RNN model established as the predictive model, the optimal control problem of the MPC controller is established by combining the objective function and constraint conditions, specifically described as:

$$\begin{aligned}
\min_{\mathbf{v_t},\delta_{\mathbf{f,t}}} & \sum_{t=1}^{N_p} p_t \left| X_t - X_{ref}^t \right| + \sum_{t=1}^{N_p} r_t \left| Y_t - Y_{ref}^t \right| \\
& + \sum_{t=1}^{N_p} s_t \left| \varphi_t - \varphi_{ref}^t \right| \\
\text{s.t.} \quad & v_{\min} \le v_t \le v_{\max}, t = 1, 2, \cdots, N_c \\
& \delta_{f,\min} \le \delta_{f,t} \le \delta_{f,\max},\ t = 1, 2, \cdots, N_c \\
& \Delta\delta_{f,t} = \delta_{f,t-1} - \delta_{f,t} \\
& -0.1 \le \Delta\delta_{f,t} \le 0.1, t = 1, 2, \cdots, N_c \\
& \text{Vehicle SSL-RNN dynamic model}
\end{aligned} \tag{17}$$

where $\mathbf{v_t} = [v_1, \cdots, v_{N_c}]$ is the velocity vector, v_t is the velocity at the time step t within the range of the velocity constraint $(v_{\min}, v_{\max})$, $\delta_{\mathbf{f,t}} = [\delta_{f,1}, \cdots, \delta_{f,N_c}]$ is the front wheel steering angle vector, and $\delta_{f,t}$ is the front wheel steering angle at the time step t within the range of the steering constraint $(\delta_{f,\min}, \delta_{f,\max})$.

The vehicle SSL-RNN dynamic model in the MPC controller is described as:

$$\begin{cases}
x^t = [v_{t-1}, \delta_{f,t-1}, X_{t-1}, Y_{t-1}, \varphi_{t-1}], t = 1, 2, \cdots, N_p \\
h_j^t = f_h\left(\sum_{i=1}^{I} w_{ij}^{xh} \cdot x_i^t + w_{jj}^{hh} \cdot \left(h_j^{t-1} + \cdots + h_j^{t-5}\right) + b_j\right) \\
y_k^t = f_o\left(\sum_{j=1}^{J} w_{jk}^{hy} \cdot h_j^t + c_k\right), k = 1, 2, \cdots, K \\
y^t = [X_t, Y_t, \varphi_t]
\end{cases} \tag{18}$$

where k is the index of each neuron in the current layer, y^t is the output vector of the SSL-RNN output layer, which is composed of the state information at the time step t.

Since the activation function of SSL-RNN and the objective function are nonlinear as shown in Eq. (4) and Eq. (13) respectively, the optimal control problem in Eq. (17) becomes a nonlinear programming problem (NLP). Although many existing algorithms can be used to solve this nonlinear optimization problem, such as the interior point method [32], particle swarm optimization (PSO) [33] and sequential quadratic programming (SQP) [34], etc., the global optimal solution cannot be guaranteed. To efficiently obtain high-quality optimal solutions and improve the computational efficiency of the controller, transforming the MPC optimal control problem in Eq. (17) into an MILP for the solution is a feasible strategy. Existing optimization solvers and algorithms can effectively process discrete variables and binary variables in MILP, find the optimal control sequence, and achieve accurate control of autonomous vehicles [35].

To transform the problem in Eq. (17) into an MILP problem, the nonlinear objective expression is first transformed into a linear objective expression by introducing new auxiliary variables and constraints:

$$\mathrm{C} = \sum_{t=1}^{N_p} p_t z_1^t + \sum_{t=1}^{N_p} r_t z_2^t + \sum_{t=1}^{N_p} s_t z_3^t \tag{19}$$

where z_1^t, z_2^t and z_3^t are the introduced three auxiliary variables, representing each absolute error term of the objective expression in each prediction step.

Meanwhile, the following constraints must be satisfied:

$$\begin{cases}
X_t - X_{ref}^t \le z_1^t \\
-\left(X_t - X_{ref}^t\right) \le z_1^t \\
Y_t - Y_{ref}^t \le z_2^t \\
-\left(Y_t - Y_{ref}^t\right) \le z_2^t \\
\varphi_t - \varphi_{ref}^t \le z_3^t \\
-\left(\varphi_t - \varphi_{ref}^t\right) \le z_3^t
\end{cases} \tag{20}$$

Then the linearization of the nonlinear activation function in the SSL-RNN model, i.e. Equation (18), is carried out by introducing auxiliary variables as shown in Equation (12). Finally, the MILP form of the optimal control problem in Equation (17) can be obtained by combining the control objective expressions in Equations (19) and (20) as follows:

$$\begin{aligned}
\min_{\mathbf{v_t},\delta_{\mathbf{f,t}},\mathbf{z},\mathbf{w},\mathbf{q}} & \sum_{t=1}^{N_p} p_t z_1^t + \sum_{t=1}^{N_p} r_t z_2^t + \sum_{t=1}^{N_p} s_t z_3^t \\
\text{s.t.} \quad & v_{\min} \le v_t \le v_{\max}, t = 1, 2, \cdots, N_c \\
& \delta_{f,\min} \le \delta_{f,t} \le \delta_{f,\max},\ t = 1, 2, \cdots, N_c \\
& \Delta\delta_{f,t} = \delta_{f,t-1} - \delta_{f,t} \\
& -0.1 \le \Delta\delta_{f,t} \le 0.1, t = 1, 2, \cdots, N_c \\
& X_t - X_{ref}^t \le z_1^t \\
& -\left(X_t - X_{ref}^t\right) \le z_1^t \\
& Y_t - Y_{ref}^t \le z_2^t \\
& -\left(Y_t - Y_{ref}^t\right) \le z_2^t \\
& \varphi_t - \varphi_{ref}^t \le z_3^t \\
& -\left(\varphi_t - \varphi_{ref}^t\right) \le z_3^t \\
& x^t = [v_{t-1}, \delta_{f,t-1}, X_{t-1}, Y_{t-1}, \varphi_{t-1}], \\
& t = 1, 2, \cdots, N_p \\
& \psi_j^t = \sum_{i=1}^{I} w_{ij}^{xh} \cdot x_i^t + w_{jj}^{hh} \cdot \left(h_j^{t-1} + \cdots + h_j^{t-5}\right) \\
& + b_j, j = 1, 2, \cdots, J \\
& h_j^t\left(\psi_j^t\right) = -w_{1j}^t - w_{2j}^t + w_{3j}^t + w_{4j}^t \\
& w_{1j}^t + w_{2j}^t + w_{3j}^t + w_{4j}^t = 1, w_{nj}^t \in [0, 1] \\
& q_{1j}^t + q_{2j}^t + q_{3j}^t = 1 \\
& w_{1j}^t \le q_{1j}^t \\
& w_{2j}^t \le q_{1j}^t + q_{2j}^t \\
& w_{3j}^t \le q_{2j}^t + q_{3j}^t
\end{aligned}$$

$$\begin{aligned}
&w_{4j}^{t} \leq q_{3j}^{t}\\
&y_k^t = \sum_{j=1}^{J} w_{jk}^{hy} \cdot h_j^t + c_k, k = 1, 2, \cdots, K\\
&y^t = [X_t, Y_t, \varphi_t] \qquad (21)
\end{aligned}$$

where $\mathbf{z} = [z_1^t, z_2^t, z_3^t]$ is the auxiliary variable vector for the linearization of the objective function, $\mathbf{w} = [w_{1j}, w_{2j}, w_{3j}, w_{4j}]$ is the auxiliary variable vector used to linearize the activation function, $\mathbf{q} = [q_{1j}^t, q_{2j}^t, q_{3j}^t]$ is a binary vector determining the segment of the activation function SSL and limiting the value of the auxiliary variable w_{nj}^t. The z_1^t, z_2^t and z_3^t are the three auxiliary variables introduced at the time step t, respectively. The w_{1j}^t, w_{2j}^t, w_{3j}^t and w_{4j}^t are the four auxiliary continuous variables of the jth neuron at the time step t, respectively. The q_{1j}^t, q_{2j}^t, and q_{3j}^t are the three binary variables of the jth neuron at the time step t, respectively, and the ψ_j^t is the input of the jth hidden layer neuron at the time step t.

The optimal control problem in Eq. (17) has been approximated to a MILP (in Eq. (21)), which is known as the fast model predictive control (FMPC) in this study, and the global optimal solution can be theoretically obtained due to its convexity [27]. The block diagram of the constructed vehicle control system is shown in Figure 8. First, the initial motion state and the reference path of the vehicle are input to the MPC controller; Then, based on the SSL-RNN prediction model, the FMPC obtains the optimal solution according to the optimization objective under constraints, and controls the vehicle. Finally, the motion state of the controlled vehicle is fed back to the MPC controller, which serves as the input to the MPC controller in the next moment. The above procedure is continuously looped to realize the path-tracking control of the vehicle.

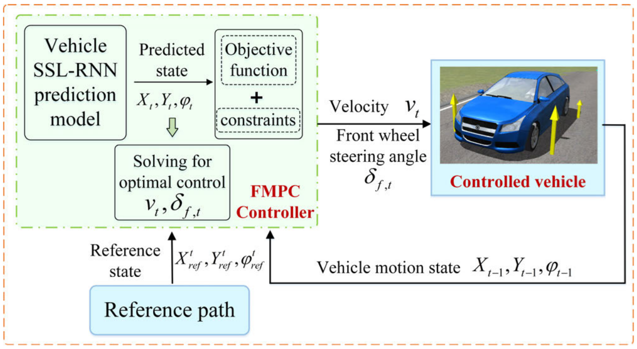
Vehicle SSL-RNN prediction model
Predicted state
X_t, Y_t, φ_t
Objective function
+
constraints
Solving for optimal control
$v_t, \delta_{f,t}$
FMPC Controller
Velocity v_t
Front wheel steering angle
$\delta_{f,t}$
Controlled vehicle
Reference state
$X_{ref}^t, Y_{ref}^t, \varphi_{ref}^t$
Vehicle motion state $X_{t-1}, Y_{t-1}, \varphi_{t-1}$
Reference path

FIGURE 8. Block diagram of the MPC system for vehicle motion.

IV. ANALYSIS AND DISCUSSIONS

In this section, the advantages of the proposed method are demonstrated. The control results of the proposed FMPC method are compared with the predictive control results based on long-short-term memory neural network (LSTM). Also, to show the computational efficiency and solution quality of the controller, the comparison results of the FMPC, the RNN-based nonlinear MPC and the predictive control based on the vehicle mechanism model, i.e. Eq. (3), are provided.

A. DESIGN OF LSTM-BASED PREDICTIVE CONTROLLER FOR VEHICLE PATH TRACKING

LSTM is a variant of recurrent neural networks with a memory unit and a gating mechanism, through which it can effectively capture and remember long-term dependencies in the input sequence and can learn and predict the path and behavior of vehicles in the future time. LSTM has been used in many studies for vehicle path tracking control [36], [37]. To verify the superiority of the proposed control method by comparison, the predictive controller based on LSTM for vehicle path tracking is designed. Like any model predictive control, the first step is to collect data and train an accurate vehicle LSTM model. The training results of the LSTM model are shown in Figure 9. The cyan circle is the state value of the data, while the red dot is the output state value of the LSTM model.

Then the vehicle LSTM model is used as the prediction model and the optimal control problem of the vehicle MPC controller is established as:

$$\begin{aligned}
&\min_{\mathbf{v_t}, \delta_{\mathbf{f,t}}} \sum_{t=1}^{N_p} p^t \left| X^t - X_{ref}^t \right| + \sum_{t=1}^{N_p} r^t \left| Y^t - Y_{ref}^t \right|\\
&\quad + \sum_{t=1}^{N_p} s^t \left| \varphi^t - \varphi_{ref}^t \right|\\
&\text{s.t.} \quad v_{\min} \leq v_t \leq v_{\max}, t = 1, 2, \cdots, N_c\\
&\quad \delta_{f,\min} \leq \delta_{f,t} \leq \delta_{f,\max}, t = 1, 2, \cdots, N_c\\
&\quad \Delta\delta_{f,t} = \delta_{f,t-1} - \delta_{f,t}\\
&\quad -0.1 \leq \Delta\delta_{f,t} \leq 0.1, t = 1, 2, \cdots, N_c\\
&\quad x^t = \left[v_{t-1}, \delta_{f,t-1}, X_{t-1}, Y_{t-1}, \varphi_{t-1}\right], t = 1, 2, \cdots, N_p\\
&\quad e_i^t = \sigma\left(\sum_{i=1}^{I} w_{ij}^{e} \cdot \left[T^{t-1}, x_i^t\right] + b_j^e\right), j = 1, 2, \cdots, J\\
&\quad L_j^t = \sigma\left(\sum_{i=1}^{I} w_{ij}^{L} \cdot \left[T^{t-1}, x_i^t\right] + b_j^L\right)\\
&\quad \tilde{C}_j^t = \tanh\left(\sum_{i=1}^{I} w_{ij}^{C} \cdot \left[T^{t-1}, x_i^t\right] + b_j^C\right)\\
&\quad o_k^t = \sigma\left(\sum_{j=1}^{J} w_{jk}^{o} \cdot \left[T^{t-1}, x_i^t\right] + b_k^o\right), k = 1, 2, \cdots, K\\
&\quad C_k^t = L_j^t * C_k^{t-1} + e_i^t * \tilde{C}_j^t\\
&\quad T_k^t = o_k^t * \tanh\left(C_k^t\right)\\
&\quad y^t = [X_t, Y_t, \varphi_t] \qquad (22)
\end{aligned}$$

where e_i^t is the input gate of the output i at the time step t, L_j^t and $\tilde{C}_j^t$ are the forgetting gates of the output j at the time step t, o_k^t is the output gate of the output k at the time step t, C_k^t is the long-term memory of the output k at the time step t, T_k^t is the short-term memory of the output k at the time step t,

x_i^t is the external input of the output i at the time step t, w_{ij}^e is the weight coefficient of the input gate, w_{ij}^L and w_{ij}^C are the weight coefficients of the forgetting gate, w_{ij}^o is the weight coefficient of the output gate, b_j^e is the bias of the input gate, b_j^L and b_j^C are the biases of the forgetting gate, b_k^o is the bias of the output gate, $*$ represents the Hadamard product, $\sigma(\cdot)$ and $\tanh(\cdot)$ are the sigmoid function and the hyperbolic tangent function, respectively.

Since the activation functions $\sigma(\cdot)$ and $\tanh(\cdot)$ of the LSTM forgetting gate and the output gate are nonlinear, the MPC based on the LSTM model, i.e. Eq. (22), is obviously a nonlinear optimal control problem.

B. COMPARISON OF PREDICTIVE CONTROLS FOR VEHICLE PATH TRACKING

For the purpose of conducting experiments, a CarSim/ Simulink co-simulation platform is built, and an S-shaped single-lane road scene is constructed based on the platform, with the road centerline serving as the path reference trajectory as shown in Figure 10. To verify the advantages of the proposed FMPC method, the experiment test is also conducted with the nonlinear MPC based on the RNN model (Eq. (17)) and a nonlinear MPC based on the vehicle mechanism model. The vehicle mechanism model-based MPC, the LSTM-based MPC, and the RNN-based nonlinear MPC are solved by using the ‘‘fmincon’’ function in the MATLAB toolbox YALMIP. The vehicle path tracking predictive controller proposed in this paper is solved by using GUROBI solver in MATLAB toolbox YALMIP, which is highly efficient for solving MILP problems. In the simulation, the relevant control parameters are shown in Table 3. In this paper, Hatchback model in CarSim is selected as the object of simulation test. The suspension of the vehicle is independent suspension, the tire type of the vehicle is 205/55 R16. The basic parameters of the vehicle model in CarSim are shown in Table 4.

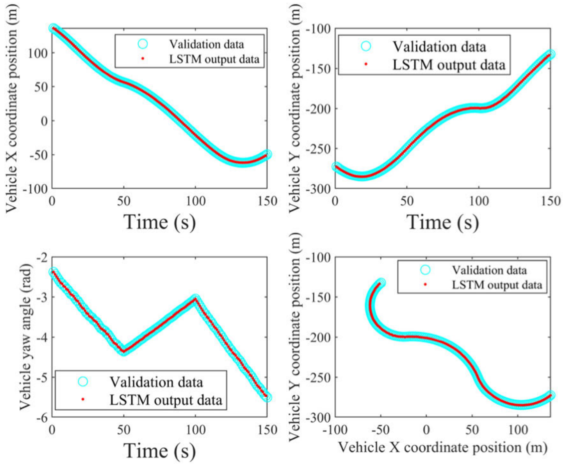

FIGURE 9. Training results of the vehicle LSTM model.

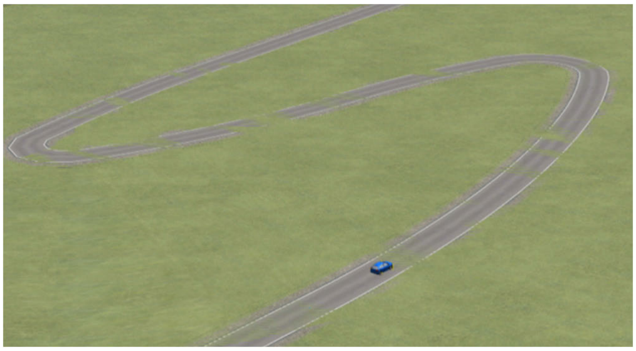

FIGURE 10. Reference path in CarSim simulation.

TABLE 3. Control parameters.

Simulation parameter	Value	Unit
Simulation step	0.1	s
Prediction time domain N_p	14	-
Control time domain N_c	4	-

TABLE 4. CarSim basic vehicle parameters.

Vehicle parameter	Value	Unit
Vehicle mass	1723	kg
Rotational inertia	4129	kg·m²
Distance from center of mass to front axis	1.04	m
Distance from center of mass to rear axis	1.62	m
Height of center of mass	0.54	m
Wheelbase of the vehicle	2.7	m
Wheel track	1.52	m

To assess the merits of the proposed FMPC method, experimental testing is conducted using both nonlinear MPC based on the LSTM model (Equation (17)) and a nonlinear MPC based on the vehicle mechanism model. Utilizing experimental data from CarSim/Simulink, the control results are compared, as illustrated in Figures 11-14. Figure 11 depicts the vehicle path tracking outcomes employing the proposed FMPC, vehicle mechanism model-based MPC, LSTM-based MPC, and RNN-based nonlinear MPC, respectively. The analysis reveals that the proposed FMPC method enhances path tracking precision, particularly in navigating turns. This improvement stems from the highly nonlinear nature of the vehicle model during turning maneuvers, where the FMPC, directly solved via the MILP scheme, yields superior control solutions. Furthermore, the Integral Square Error (ISE) index is computed to provide a quantitative assessment of the control performance of the proposed FMPC method.

$$ISE = \sum_{t=1}^{T_{end}} E^2(t) \tag{23}$$

where $E(t)$ is the deviation of tracking control at each instant t.

The ISE results of the aforementioned four control methods are shown in Table 5. Comparative analysis reveals that the proposed FMPC method in this study exhibits the lowest ISE index among the evaluated approaches. In comparison with the vehicle mechanism model-based MPC, the ISE index

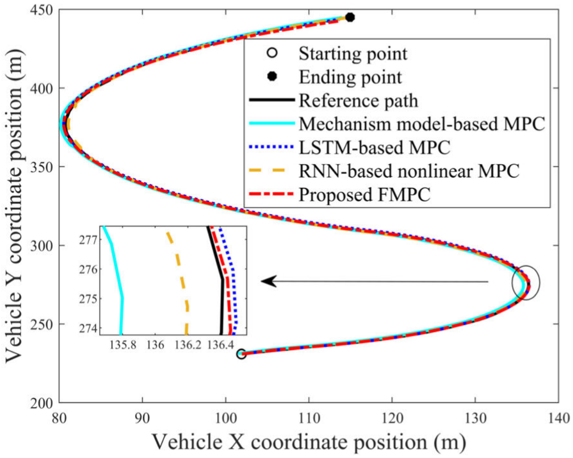

FIGURE 11. Comparison of the path-tracking results. The black solid line represents the reference path, while the cyan solid line represents the driving path based on the mechanism model-based MPC. The blue dotted line corresponds to the driving path based on the LSTM-based MPC, the yellow dotted line depicts the driving path based on the RNN-based MPC, and the red dotted line showcases the driving path based on the proposed FMPC.

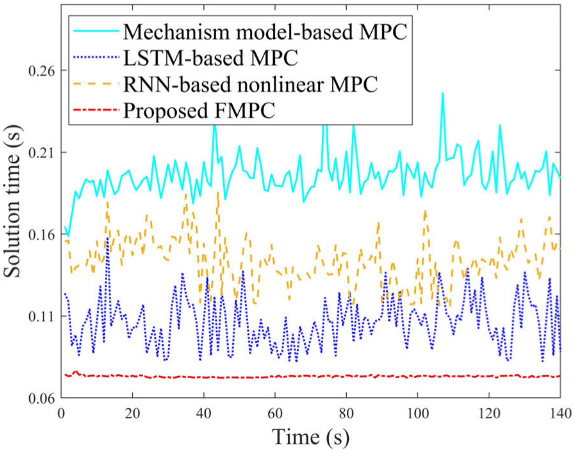

FIGURE 12. Comparison of control solution time. The cyan solid line represents the solution time based on the mechanism model-based MPC, the blue dotted line corresponds to the solution time based on the LSTM-based MPC, the yellow dotted line depicts the solution time based on the RNN-based MPC, and the red dotted line showcases the solution time for the proposed FMPC.

reductions for X, Y, and φ of the proposed FMPC method amount to 46.27%, 19.06%, and 45.95%, respectively. Similarly, compared with the result of the LSTM-based MPC, the ISE index reductions for X, Y, and φ of the proposed FMPC method are 7.53%, 7.13%, and 4.10%, respectively. Compared with the result of the RNN-based nonlinear MPC, the ISE index reductions for X, Y, and φ of the proposed FMPC method are 37.39 %, 10.44%, and 21.43%, respectively. Additionally, the maximum tracking deviations for X and Y using the proposed FMPC method are 0.22 and 0.30, respectively, which represent the smallest deviations among the four control methods evaluated.

The solution time for the three controllers is shown in Figure 12, revealing that the solution time of the proposed FMPC is shorter. Further, Table 6 provides insight into the average solution times, indicating that the proposed FMPC method's solution time is 62.74%, 31.30%, and 49.09% shorter than those of the vehicle mechanism model-based MPC method, LSTM-based MPC method, and RNN-based nonlinear MPC method, respectively. Leveraging the MILP scheme for solution, the proposed FMPC method demonstrates expedited solution times, rendering it favorable for online control applications. In contrast, MPC based on mechanism models exhibits the longest average solution time, attributed to the necessity of solving nonlinear differential equations within its optimal control problem. The comparison of manipulated variables for the mechanism model-based MPC, RNN-based nonlinear MPC, and LSTM-based MPC, as shown in Figures 13-14, indicates that the manipulated variables (velocity and front wheel steering angle) of the proposed FMPC are relatively stable contributing to improved driving stability and comfort. However, to achieve fast solutions, the prediction horizon of the proposed controller is kept short, resulting in noticeable vibration in noticeable

TABLE 5. ISE results.

	ISE (X)	ISE (Y)	ISE (φ)
Mechanism model-based MPC	9.7382	22.6321	0.0346
LSTM-based MPC	5.6578	19.7243	0.0195
RNN-based nonlinear MPC	8.3563	20.4527	0.0238
Proposed FMPC	5.2319	18.3180	0.0187

TABLE 6. Average solution time.

	Average solution time (s)
Mechanism model-based MPC	0.1962
LSTM-based MPC	0.1064
RNN-based nonlinear MPC	0.1436
Proposed FMPC	0.0731

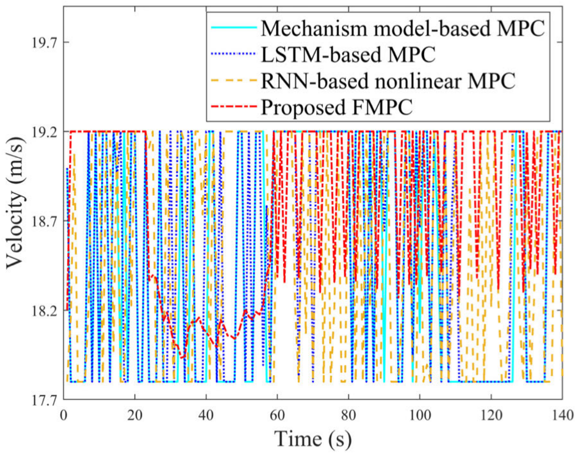

FIGURE 13. Comparison of the velocity control variables. The cyan solid line represents the velocity given by the mechanism model-based MPC, the blue dotted line corresponds to the velocity given by the LSTM-based MPC, the yellow dotted line depicts the velocity given by the RNN-based MPC, and the red dotted line showcases the velocity given by the proposed FMPC.

vibration in the manipulated variables. In summary, the proposed FMPC method effectively achieves high-precision path tracking and demonstrates superior control performance.

V. CONCLUSION

In this paper, a fast predictive control method for vehicle path tracking is proposed for the vehicle path tracking control problem, and the effectiveness and advantages of the proposed FMPC are verified by experiments. Specifically, the data from different vehicle operating states is first collected to train the SSL-RNN model. Then, the FMPC is designed based on the established SSL-RNN model of the vehicle, and the constructed MPC optimal control problem is transformed into a MILP problem structure for solving, which can well improve the computational efficiency and solution quality of the controller. Finally, a joint CarSim/Simulink simulation platform is built for experiments. Compared with the control results of the vehicle mechanism model-based MPC, the LSTM-based MPC and the RNN-based nonlinear MPC, the results show that the proposed FMPC for vehicles has higher path tracking accuracy under turning conditions and effectively improves the solution efficiency of the controller, which is more conducive to the implementation of online control. In summary, the main contributions of this work are stated as follows:

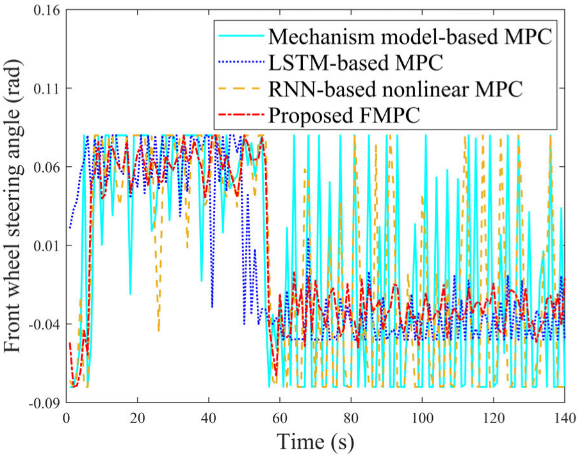

FIGURE 14. Comparison of the front wheel steering angle during control. The cyan solid line represents the front wheel steering angle based on the mechanism model-based MPC, the blue dotted line corresponds to the front wheel steering angle based on the LSTM-based MPC, the yellow dotted line depicts the front wheel steering angle based on the RNN-based MPC, and the red dotted line showcases the front wheel steering angle based on the proposed FMPC.

1) Aiming at the problem of vehicle path tracking control, a fast predictive control method for vehicle path tracking based on the linearized SSL-RNN model is proposed, which has good control performance.

2) By linearizing the activation function and objective expression, the MPC optimal control problem is transformed into MILP, which improves the computational efficiency of the FMPC.

3) A control comparison is conducted, and the results show that the proposed FMPC method had better path tracking control performance. In comparison with the vehicle mechanism model-based MPC, the ISE index reductions for X, Y, and φ of the proposed FMPC method amount to 46.27%, 19.06%, and 45.95%, respectively. Similarly, compared with the result of the LSTM-based MPC, the ISE index reductions for X, Y,and φ of the proposed FMPC method are 7.53%, 7.13%, and 4.10%, respectively. Additionally, when compared with the result of the RNN-based nonlinear MPC, the ISE index reductions for X, Y, and φ of the proposed FMPC method are 37.39 %, 10.44%, and 21.43%, respectively. Meanwhile, among the four control methods, the proposed FMPC method has the shortest average control solution time (0.0731 s), which is 62.74%, 31.30%, and 49.09% shorter than the vehicle mechanism model-based MPC method, the LSTM-based MPC method, and the RNN-based nonlinear MPC method, respectively.

In the future, the proposed method will be applied to more complex verification scenarios to further demonstrate its advantages. Furthermore, it will be tested in a real vehicle to further verify its effectiveness for vehicle path tracking in real environments.

REFERENCES

[1] D. Parekh, N. Poddar, A. Rajpurkar, M. Chahal, N. Kumar, G. P. Joshi, and W. Cho, "A review on autonomous vehicles: Progress, methods and challenges," *Electronics*, vol. 11, no. 14, p. 2162, Jul. 2022.

[2] M. Aledhari, M. Rahouti, J. Qadir, B. Qolomany, M. Guizani, and A. Al-Fuqaha, "Motion comfort optimization for autonomous vehicles: Concepts, methods, and techniques," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 378–402, Jan. 2023.

[3] L. Gharavi, A. Dabiri, J. Verkuijlen, B. De Schutter, and S. Baldi, "Proactive emergency collision avoidance for automated driving in highway scenarios," 2023, *arXiv:2310.17381*.

[4] Q. Yao, Y. Tian, Q. Wang, and S. Wang, "Control strategies on path tracking for autonomous vehicle: State of the art and future challenges," *IEEE Access*, vol. 8, pp. 161211–161222, 2020.

[5] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., DTIC Document, 1992.

[6] W. P. Mounfield and L. T. Grujic, "PID natural tracking control of a robot: Application," in *Proc. IEEE Syst. Man Cybern. Conf. (SMC)*, vol. 4, Le Touquet, France, Oct. 1993, pp. 328–333.

[7] S. Thrun et al., "Stanley: The robot that won the DARPA grand challenge," *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, 2006.

[8] C. Zhang, G. Gao, C. Zhao, L. Li, C. Li, and X. Chen, "Research on 4WS agricultural machine path tracking algorithm based on fuzzy control pure tracking model," *Machines*, vol. 10, no. 7, p. 597, Jul. 2022.

[9] X. Chen, Q. Bao, and B. Zhang, "Research on 4WIS electric vehicle path tracking control based on adaptive fuzzy PID algorithm," in *Proc. Chin. Control Conf. (CCC)*, Guangzhou, China, Jul. 2019, pp. 6753–6760.

[10] J. Yang, H. Bao, N. Ma, and Z. Xuan, "An algorithm of curved path tracking with prediction model for autonomous vehicle," in *Proc. 13th Int. Conf. Comput. Intell. Secur. (CIS)*, Hong Kong, 2017, pp. 405–408.

[11] Z. Fan and H. Chen, "Study on path following control method for automatic parking system based on LQR," *SAE Int. J. Passenger Cars-Electron. Electr. Syst.*, vol. 10, no. 1, pp. 41–49, Sep. 2016, doi: 10.4271/2016-01-1881.

[12] N. R. Kapania and J. C. Gerdes, "Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling," *Vehicle Syst. Dyn.*, vol. 53, no. 12, pp. 1687–1704, Dec. 2015.

[13] H. Pan, C. Zhang, and W. Sun, "Fault-tolerant multiplayer tracking control for autonomous vehicle via model-free adaptive dynamic programming," *IEEE Trans. Rel.*, vol. 72, no. 4, pp. 1395–1406, Dec. 2022.

[14] J. Yao and Z. Ge, ''Path-tracking control strategy of unmanned vehicle based on DDPG algorithm,'' *Sensors*, vol. 22, no. 20, p. 7881, Oct. 2022.

[15] H. Wang, Q. Wang, W. Chen, L. Zhao, and D. Tan, ''Path tracking based on model predictive control with variable predictive horizon,'' *Trans. Inst. Meas. Control*, vol. 43, no. 12, pp. 2676–2688, Apr. 2021.

[16] H. Wei, H. Zhang, K. AI-Haddad, and Y. Shi, ''Ensuring secure platooning of constrained intelligent and connected vehicles against Byzantine attacks: A distributed MPC framework,'' *Engineering*, vol. 33, pp. 35–46, Feb. 2024.

[17] Y. Jeong and S. Yim, ''Model predictive control-based integrated path tracking and velocity control for autonomous vehicle with four-wheel independent steering and driving,'' *Electronics*, vol. 10, no. 22, p. 2812, Nov. 2021.

[18] L. Liu, B. Wang, and Y. He, ''Research on path-tracking control of articulated vehicle with a trailer based on advanced model prediction control strategy,'' *Lateral*, vol. 20, p. 1, Jun. 2021.

[19] J. Wang, M. T. H. Fader, and J. A. Marshall, ''Learning-based model predictive control for improved mobile robot path following using Gaussian processes and feedback linearization,'' *J. Field Robot.*, vol. 40, no. 5, pp. 1014–1033, Aug. 2023.

[20] X. Yu, H. Wang, C. Teng, X. Sun, L. Chen, and Y. Cai, ''DGPR-MPC: Learning-based model predictive controller for autonomous vehicle path following,'' *IET Intell. Transp. Syst.*, vol. 17, no. 10, pp. 1992–2003, Oct. 2023.

[21] P. Stano, U. Montanaro, D. Tavernini, M. Tufo, G. Fiengo, L. Novella, and A. Sorniotti, ''Model predictive path tracking control for automated road vehicles: A review,'' *Annu. Rev. Control*, vol. 55, pp. 194–236, Jan. 2023.

[22] M. Rokonuzzaman, N. Mohajer, S. Nahavandi, and S. Mohamed, ''Model predictive control with learned vehicle dynamics for autonomous vehicle path tracking,'' *IEEE Access*, vol. 9, pp. 128233–128249, 2021.

[23] N. A. Spielberg, M. Brown, and J. C. Gerdes, ''Neural network model predictive motion control applied to automated driving with unknown friction,'' *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 5, pp. 1934–1945, Sep. 2022.

[24] J. S. Stine, B. C. Hamblin, S. N. Brennan, and E. T. Donnell, ''Analyzing the influence of median cross-section design on highway safety using vehicle dynamics simulations,'' *Accident Anal. Prevention*, vol. 42, no. 6, pp. 1769–1777, Nov. 2010.

[25] R. Yogitha and G. Mathivanan, ''Performance analysis of transfer functions in an artificial neural network,'' in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Chennai, India, Apr. 2018, pp. 0393–0397.

[26] S.-B. Yang, Z. Li, and W. Wu, ''Data-driven process optimization considering surrogate model prediction uncertainty: A mixture density network-based approach,'' *Ind. Eng. Chem. Res.*, vol. 60, no. 5, pp. 2206–2222, Feb. 2021.

[27] H. Sildir and E. Aydin, ''A mixed-integer linear programming based training and feature selection method for artificial neural networks using piece-wise linear approximations,'' *Chem. Eng. Sci.*, vol. 249, Feb. 2022, Art. no. 117273.

[28] J. O. B. Lira, H. G. Riella, N. Padoin, and C. Soares, ''Computational fluid dynamics (CFD), artificial neural network (ANN) and genetic algorithm (GA) as a hybrid method for the analysis and optimization of micro-photocatalytic reactors: NOx abatement as a case study,'' *Chem. Eng. J.*, vol. 431, Mar. 2022, Art. no. 133771.

[29] K. Berntorp, R. Quirynen, T. Uno, and S. Di Cairano, ''Trajectory tracking for autonomous vehicles on varying road surfaces by friction-adaptive nonlinear model predictive control,'' *Vehicle Syst. Dyn.*, vol. 58, no. 5, pp. 705–725, Nov. 2019.

[30] M. De Paula and G. G. Acosta, ''Trajectory tracking algorithm for autonomous vehicles using adaptive reinforcement learning,'' in *Proc. OCEANS-MTS/IEEE Washington*, Washington, DC, USA, Oct. 2015, pp. 1–8.

[31] X. Tang, L. Shi, B. Wang, and A. Cheng, ''Weight adaptive path tracking control for autonomous vehicles based on PSO-BP neural network,'' *Sensors*, vol. 23, no. 1, p. 412, Dec. 2022.

[32] I. M. Bomze et al., ''Interior point methods for nonlinear optimization,'' in *Nonlinear Optimization: Lectures Given at the CIME Summer School Held in Cetraro, Italy, July 1–7, 2007*, vol. 1989. Springer, Jan. 2010, pp. 215–276.

[33] J. Kennedy and R. Eberhart, ''Particle swarm optimization,'' in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Perth, WA, Australia, 1995, pp. 1942–1948.

[34] P. E. Gill and E. Wong, ''Sequential quadratic programming methods,'' in *Mixed Integer Nonlinear Programming*, vol. 154. New York, NY, USA: Springer, Nov. 2011, pp. 147–224.

[35] Y. Cao, Z. Zhang, F. Cheng, and S. Su, ''Trajectory optimization for high-speed trains via a mixed integer linear programming approach,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17666–17676, Oct. 2022.

[36] P. Wang, H. Yu, C. Liu, Y. Wang, and R. Ye, ''Real-time trajectory prediction method for intelligent connected vehicles in urban intersection scenarios,'' *Sensors*, vol. 23, no. 6, p. 2950, Mar. 2023.

[37] S. Wang, Z. Sun, Q. Yuan, Z. Sun, Z. Wu, and T.-H. Hsieh, ''Autonomous piloting and berthing based on long short time memory neural networks and nonlinear model predictive control algorithm,'' *Ocean Eng.*, vol. 264, Nov. 2022, Art. no. 112269.

XIALAI WU received the Ph.D. degree in control science and engineering from Zhejiang University, China, in 2019. He is currently a Lecturer with Huzhou University. His current research interests include process control and optimization.

LING LIN received the B.Eng. degree from Huzhou University, China, in 2021, where she is currently pursuing the M.S. degree. Her research interests include machine learning, deep learning, and autonomous driving.

JUNGHUI CHEN received the B.S. degree from the Department of Chemical Engineering, Chung-Yuan Christian University, Taiwan, in 1982, the M.S. degree from the Department of Chemical Engineering, National Taiwan University, in 1984, and the Ph.D. degree from the Department of Chemical Engineering, University of Tennessee, Knoxville, TN, USA, in 1995. He is currently a Full Professor with Chung-Yuan Christian University. His research interests include process system engineering, including process design for operability, nonlinear control, process monitoring and diagnosis, control loop performance assessment, batch control, model predictive control, data mining and analytics, and iterative learning design.

SHUXIN DU received the B.S. degree in industrial electrical automation from Jiangsu University, in 1989, the M.S. degree in theory and application of automatic control from Xi'an Jiaotong University, in 1992, and the Ph.D. degree in aircraft control, guidance and simulation from Northwestern Polytechnical University, in 1995. From September 1995 to September 1997, he was a Postdoctoral Fellow with the Institute of Industrial Process Control, Zhejiang University. From September 1997 to September 2015, he was an Associate Professor with the Department of Control Science and Engineering, Zhejiang University. He is currently a Professor with the Engineering College, Huzhou University. His research interests include control theory and application, pattern recognition and intelligent systems, and the on-line measurement of quality parameters based on spectrum.