Lateral control of intelligent vehicles using radial basis function neural networks with sliding mode control based on fractional order calculus

Received 18 September 2022
Revised 26 September 2022
Accepted 26 September 2022

Peiqing Li and Taiping Yang
*School of Mechanical and Energy Engineering,
Zhejiang University of Science and Technology, Hangzhou, China*
Hao Zhang
*UK-China Joint Intelligent Center of Special Equipment and Robotics Ltd,
Ashford, UK, and*
Lijun Wang and Qipeng Li
*School of Mechanical and Energy Engineering,
Zhejiang University of Science and Technology, Hangzhou, China*

Abstract

Purpose – This paper aimed a fractional-order sliding mode-based lateral lane-change control method that was proposed to improve the path-tracking accuracy of vehicle lateral motion.

Design/methodology/approach – In this paper the vehicle presighting and kinematic models were established, and a new sliding mode control isokinetic convergence law was devised based on the fractional order calculus to make the front wheel turning angle approach the desired value quickly. On this basis, a fractional gradient descent algorithm was proposed to adjust the radial basis function (RBF) neuron parameter update rules to improve the compensation speed of the neural network.

Findings – The simulation results revealed that, compared to the traditional sliding mode control strategy, the designed controller eliminated the jitter of the sliding mode control, sped up the response of the controller, reduced the overshoot of the system parameters and facilitated accurate and fast tracking of the desired path when the vehicle changed lanes at low speeds.

Originality/value – This paper combines the idea of fractional order calculus with gradient descent algorithm, proposed a fractional-order gradient descent method applied to RBF neural network and fast adjustment the position and width of neurons.

Keywords Trajectory tracking, Lateral motion control, Sliding mode control, Radial basis neural network, Fractional order calculus

Paper type Research paper

Funding: This project was supported by the Zhejiang Lingyan Project (2022C04022) and Natural Science Foundation of Zhejiang Province (LGG20F020008) and Key (team) Project of Zhejiang University of Science and Technology (Grant No. 2021JLZD004).

Declaration of conflicting interests: The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Journal of Intelligent Manufacturing and Special Equipment
Vol. 3 No. 2, 2022
pp. 117-133
Emerald Publishing Limited
e-ISSN: 2633-660X
p-ISSN: 2633-6596
DOI 10.1108/JIMSE-09-2022-0017

1. Introduction

Intelligent vehicles identify nearby vehicles and road conditions through the feedback provided by on-board sensors to the vehicle controller (Mozaffari *et al.*, 2022). When the vehicle moves laterally, the tracking error needs to be adjusted in time to ensure accurate road trajectory tracking accuracy, and the controller is required to eliminate external interference online to satisfy the steering lane change requirements. Thus, vehicle lateral control has become a hot topic in intelligent vehicle research (Tian *et al.*, 2022).

The vehicle lateral control system is complex and has many control variables that require an onboard controller to calculate and adjust the wheel deflection angle in real time. Several studies have proposed multiple algorithms for the vehicle lateral control problem. Nadia *et al.* (Zendehdel and Gholami, 2021) combined fuzzy algorithm with a proportional integral derivative (PID) controller to design a robust PID feedback system to autonomously adjust PID parameters to achieve lateral path tracking. Yang *et al.* (2021) designed a feed forward and predictive linear quadratic regulator (LQR) coupled with the lateral control algorithm based on the path tracking error model, which was adapted to the system environment with complex and unknown disturbance quantities. Chen *et al.* (2021) proposed a feed forward and feedback system based on the LQR control to reduce the lateral tracking error and improve system robustness by solving for the front wheel lateral deflection force.

The proposal of intelligent control algorithms has helped in the development of the automotive industry, and the control algorithm with neural networks as the system solution has gradually become one of the research objectives for scholars and engineers aiming to optimize traditional controllers (Aalizadeh and Asnafi, 2018; Fan *et al.*, 2022). Intelligent control, the frontier technology of automatic control, solves complex linear and uncertain control problems (Guo *et al.*, 2021). The sliding mode control algorithm is simple and easy to implement with high robustness. Therefore, the combination of intelligent control and sliding mode control has become a primary solution in engineering. However, jitter is the most important problem many scholars have aimed to solve. Gao *et al.* (2017) used adaptive control and fuzzy control as the sliding mode control parameter adjustment optimization algorithm, using the online fitting capability of the adaptive algorithm to adjust the fuzzy control algorithm parameters in real time, to achieve lateral path tracking for changes in its own parameters and external disturbances, which effectively reduced the jitter problem of the sliding mode control algorithm. Zhang *et al.* (2019) established a two-point presighting system with fuzzy control and designed fuzzy rules to adaptively adjust the weight factors of near and far points to solve the heading error; the system had a good path tracking capability. Fan *et al.* (2022), to solve the problems such as parameter time variation and output jitter, used the radial basis function (RBF) neural network to optimize the sliding mode controller convergence law to eliminate the time-varying vehicle.

The designed control system had strong robustness to various problems such as time-varying parameters and output jitter. Luo and Guo (2021) designed an adaptive neural network sliding mode controller based on an optimized sliding mode control algorithm using an RBF neural network online to compensate for adverse external disturbances and achieve fast and accurate tracking of wheel turning angles to improve the robustness of the system. Zhou *et al.* (2019) used an improved particle swarm algorithm to dynamically solve the RBF neural network structure and fit the sliding mode control switching parameters online to significantly reduce the time to reach the sliding mode surface, eliminate the system modeling uncertainty and other adverse effects, and alleviate the system oscillations.

The system variables of the vehicle lateral control are strongly coupled, and, therefore, have extremely high requirements for the solution speed of the controller (Li, 2021). To overcome the problems of slow convergence speed of the sliding mode controller and poor anti-disturbance capability of the system, the isokinetic convergence law based on fractional-order calculus was optimized in this study for fast convergence to the sliding mode surface.

Furthermore, an RBF neural network was used to dynamically compensate the uncertain parameters in the vehicle model and the unknown disturbances of the system online, and the Lyapunov function was used for the stability analysis of the proposed control system proposed (Muñoz-Vázquez *et al.*, 2022; Wang *et al.*, 2019). On this basis, for the problem of updating neuron parameters in the RBF neural network, the idea of fractional order was introduced and a caputo-based fractional order gradient descent algorithm was designed to solve the width and position of neurons in the RBF neural network in real time. Compared with the traditional sliding mode control and the sliding mode controller following RBF optimization, the proposed control algorithm proposed can effectively improve the system by eliminating jitter, dynamically compensating the unknown disturbance quantity of the system and achieving fast tracking of the desired path.

The remainder of this paper is organized as follows. The kinematic mathematical model of the vehicle is presented in Section 2. The structure of the sliding-mode controller with RBF optimization is presented in Section 2. In Section 4, the simulation results for two different scenarios are presented. The conclusion is presented in Section 5.

2. Vehicle dynamics modeling

2.1 Vehicle preview model

The driver completes the steering operation by adjusting the steering wheel angle to reduce the deviation of the desired path from the body attitude, and controlling the accelerator pedal to complete the speed regulation of the vehicle when steering (Alexandru, 2017). Therefore, the controller dynamically adjusts the front-wheel rotation angle such that the deviation is zero when the smart vehicle tracks the desired path. We studied the vehicle single-point pretargeting model (point L in Figure 1) as the object of study and considered the center point of the front and rear axles of the vehicle (point O in Figure 1) as the coordinate origin and established the coordinate system, as shown in Figure 1.

The vehicle kinematic model was established based on the coordinates shown in Figure 1,

$$\begin{cases} \dot{y}_L = v_x \varepsilon_L - v_y - \dot{\varphi} D_L \\ \dot{\varepsilon}_L = v_x K_L - \dot{\varphi} \end{cases} \tag{1}$$

where y_L is the relative lateral distance between points L and O, v_x and v_y represent the forward and lateral velocities of the vehicle, respectively. ε_L is the heading angle of the vehicle at point L. φ is the transverse sway angle of the vehicle. D_L is the relative longitudinal distance between points L and O, and K_L is the curvature of the desired road trajectory ahead.

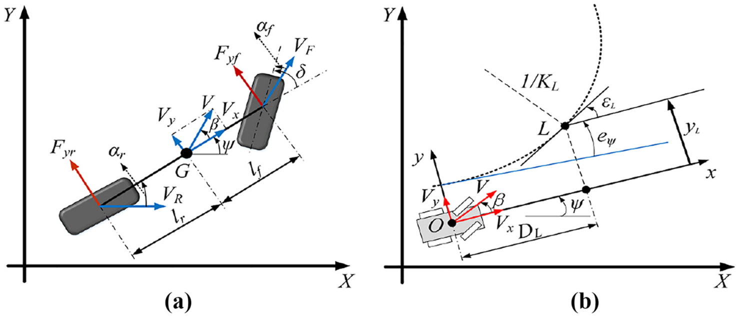


Figure 1.
(a) Vehicle 2-degree-of-freedom model and (b) Single-point presighting model

2.2 Vehicle two-degree-of-freedom model

The lateral motion of the intelligent vehicle adjusts the deviation of the self-vehicle position from the pretargeting point by dynamically adjusting the front-wheel turning angle through the actuator. To reduce modeling complexity and control system arithmetic, this study was premised on the following assumptions about the vehicle modeling problem:

(1) The model completely ignored the influence of the vertical motion of the vehicle on the lateral motion characteristics.

(2) The model took the front-wheel angle as the control system input and assumed that the left and right wheel angles were equal at all times during steering.

(3) The effect of the lateral deflection force on lateral motion owing to the tires was not considered, and it was assumed that both sides of the body dynamics parameters responded similarly when the vehicle was moving.

Based on the above assumptions, the two-degree-of-freedom vehicle motion model was considered as the main research object and the model differential equations were established as follows:

$$\begin{cases} (k_1 + k_2)\dfrac{v_y}{v_x} + (l_1k_1 - l_2k_2)\dfrac{\dot{\varphi}}{v_x} - k_1\delta = m(\dot{v}_y + v_x\dot{\varphi}) \\ (l_1k_1 - l_2k_2)\dfrac{v_y}{v_x} + \left({l_1}^2k_1 + {l_2}^2k_2\right)\dfrac{\dot{\varphi}}{v_x} - l_1k_1\delta = I_z\dot{\varphi} \end{cases} \tag{2}$$

where k_1 and k_2 are the lateral deflection stiffnesses of the front and rear axle tires, respectively. l_1 and l_2 are the distances from the center of the front and rear axles to the center of mass of the vehicle, respectively. δ is the front-wheel angle, where m is the overall mass of the vehicle. I_z is the rotational inertia of the vehicle in the vertical direction.

2.3 Vehicle motion state model

Based on the above model analysis, the system state equation was established as (Lai and Huang, 2022)

$$\dot{X} = \begin{bmatrix} a_1 & a_2 & 0 & 0 \\ a_3 & a_4 & 0 & 0 \\ -1 & -D_L & 0 & v_x \\ 0 & -1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ 0 \\ 0 \end{bmatrix}\delta + \begin{bmatrix} 0 \\ 0 \\ 0 \\ v_x \end{bmatrix}K_L \tag{3}$$

where X represents the system state, $X = \begin{bmatrix} v_y & \dot{\varphi} & y_L & \varepsilon_L \end{bmatrix}^T$, $a_1 = -\frac{k_1+k_2}{mv_x}$, $a_2 = -\frac{l_2k_2 - l_1k_1}{mv_x}$, $a_3 = -\frac{l_2k_2 - l_1k_1}{I_zv_x}$, $a_4 = -\frac{{l_1}^2k_1 - {l_2}^2k_2}{I_zv_x}$, $c_1 = \frac{k_1}{m}$, $c_2 = \frac{l_1k_1}{I_Z}$.

The ideal lateral motion of the vehicle is a zero deviation of the self-driving vehicle from the desired trajectory at the presighting point (point L in Figure 1). The equation of motion state for vehicle motion with body attitude deviation p is

$$\ddot{p} = A\dot{p} + Bp + C\delta + D \tag{4}$$

where $p = \begin{bmatrix} y_L \\ \varepsilon_L \end{bmatrix}$, $A = \begin{bmatrix} 0 & A_1 \\ A_2 & A_3 \end{bmatrix}$, $B = \begin{bmatrix} B_1 & B_2 \\ B_3 & 0 \end{bmatrix}$, $C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$, $D = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix}$.

The parameters in the matrix are specifically expressed as

$$\& \begin{cases} A_1 = v_x - a_1 D_L + a_2 - a_3 {D_L}^2 + a_4 D_L \\ A_2 = a_3 \\ A_3 = a_4 - a_3 D_L \end{cases}, \begin{cases} B_1 = a_1 + a_3 D_L \\ B_2 = -a_1 v_x - a_3 v_x D_L \\ B_3 = -a_3 v_x \end{cases},$$

$$\& \begin{cases} C_1 = -c_1 - c_2 D_L \\ C_2 = -c_2 \end{cases}, \begin{cases} D_1 = \left(a_1 D_L - a_2 + a_3 {D_L}^2 - a_4 D_L\right) v_x K_L \\ D_2 = (a_3 D_L - a_4) v_x K_L + v_x K_L \end{cases}$$

In the above analysis, the vehicle front-wheel turning angle was considered as the problem entry point and a complete vehicle motion model was established. The ideal vehicle attitude deviation was taken as the desired value, and the actual attitude deviation p was compared with the desired value to obtain the change in attitude deviation, thus establishing the vehicle–vehicle lateral control expectation model.

3. Fractional-order sliding mode control based on neural-network compensation

Vehicle modeling uncertainty and external unknown disturbances have a large impact on the vehicle path tracking accuracy. Therefore, the RBF neural network was used to compensate the unknown disturbances in the system online dynamically. The working-state structure of the system is shown in Figure 2. The system was realized by deploying the fractional order calculus to build a new equal convergence law for fast convergence to the sliding mode surface, while eliminating the control system oscillations problem in the traditional sliding mode control algorithm.

3.1 Fractional order sliding mode control

Three types of calculus, GL-defined, RL-defined and Caputo-defined are widely used in engineering projects (Swain *et al.*, 2021). The Caputo definition operator does not require the determination of the initial value of the fractional-order derivative, a unique property that is extremely convenient for numerical calculations with nonzero initial values. The Caputo definition has been intensively studied by scholars because it is an extension of the integer order derivative and is suitable for expressing its initial conditions, in terms of simple integer-order derivatives. The three definitions are as follows:

$${}^{RL}_{t_0}D^{\alpha}_{t} f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_{t_0}^{t} \frac{f(\tau)}{(t-\tau)^{\alpha+1-n}} d\tau$$

$${}^{GL}_{t_0}D^{\alpha}_{t} f(t) = \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{[(t-t_0)/h]} (-1)^j \binom{\alpha}{j} f(t - jh)$$

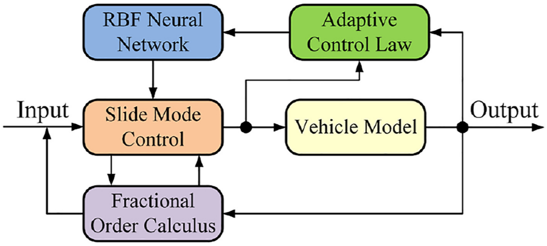


Figure 2.
Control system structure diagram

$$ {}_{t_0}^{C}D_t^{\alpha}f(t)=\frac{1}{\Gamma(n-\alpha)}\int_{t_0}^{t}\frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha+1-n}}d\tau $$

where ${}_{t_0}^{RL}D_t^{\alpha}$ denotes the RL integral operator, ${}_{t_0}^{GL}D_t^{\alpha}$ denotes the GL integral operator, ${}_{t_0}^{C}D_t^{\alpha}$ denotes the Caputo integral operator, α denotes the order of the integral operator and h denotes the integration step. $\begin{pmatrix}\alpha\\ j\end{pmatrix}$ is the number of combinations that can be calculated using the following equation:

$$ \begin{pmatrix}\alpha\\ j\end{pmatrix}=\frac{\alpha(\alpha-1)(\alpha-2)\cdots(\alpha-j+1)}{j!}=\frac{\alpha!}{j!(\alpha-j)!} \tag{5} $$

where $\Gamma(\cdot)$ is the Gamma function, defined as follows:

$$ \Gamma(\eta)=\int_{0}^{\infty}e^{-t}u^{x-1}du \tag{6} $$

Lemma 1. If $x=0$ can obtain the equilibrium condition of Equation (8), then

$$ {}^{C}D^{\alpha}x(t)=f(x,t) \tag{7} $$

where $f(x,t)$ satisfies the Lipschitz condition.

Suppose Lyapunov's theorem satisfies the following conditions

$$ \begin{aligned} &\alpha_1(\|x\|)\le V(t,x(t))\le\alpha_2(\|x\|)\\ &{}^{C}D^{\beta}V(t,x(t))\le-\alpha_3(\|x\|) \end{aligned} \tag{8} $$

where α_1, α_2 and α_3 are all positive integers, $\beta\in(0,1)$

Then, it was demonstrated that the system represented by Equation (8) was stable. Define the expression of the system path-tracking attitude error as

$$ e=p-p_d \tag{9} $$

where p is the actual deviation, p_d is the expected deviation; therefore, $p_d=0$.

The design sliding surface is

$$ s=\lambda e+\dot{e} \tag{10} $$

where λ is the design parameter, $\lambda=\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}$, and λ_1 and λ_2 are both greater than zero.

We integrated the left-hand side s of Equation (10) to obtain

$$ \dot{s}=\lambda\dot{e}+\ddot{e}=\lambda\dot{e}+A\dot{p}+Bp+C\delta+D-\left(f\left(\dot{p},t\right)+\Delta f\left(\dot{p},t\right)+d(t)\right) \tag{11} $$

where $f(\dot{p},t)$ is the nonlinear system term, $\Delta f(\dot{p},t)$ is the unknown disturbance of the system, $d(t)$ is the external unknown disturbance satisfying $d(t)\le M$, where M is the upper limit of the disturbance.

The traditional sliding mode control algorithm converged slowly, resulting in a long convergence time to the sliding mode surface. Further, there was a jitter problem. Therefore, the Caputo definition of the integral operator was used to optimize the equal speed convergence law in this study, as defined below:

$$ {}^{C}D^{\alpha}s = -k_c \operatorname{sgn}(s) \tag{12}$$

where $0 \le \alpha \le 1$.

Therefore, we can obtain

$$ \dot{s} = {}^{C}D^{1-\alpha}(-k_c \operatorname{sgn}(s)) \tag{13}$$

The following Lyapunov function was used

$$ V = \frac{1}{2}s^T s \tag{14}$$

We derived both ends of the above equation simultaneously and obtained

$$ \dot{V} = s^T \dot{s} = s^{T\,C}D^{1-\alpha}(-k_c \operatorname{sgn}(s)) \tag{15}$$

Owing to the

$$ \operatorname{sgn}\left[{}^{C}D^{1-\alpha}(-k_c \operatorname{sgn}(s))\right] = -k_c \operatorname{sgn}(s) \tag{16}$$

Then

$$ \operatorname{sgn}(\dot{V}) = \operatorname{sgn}\left(s^T\right)\operatorname{sgn}\left({}^{C}D^{1-\alpha}(-k_c \operatorname{sgn}(s))\right) = \operatorname{sgn}\left(s^T\right)\operatorname{sgn}(-k_c \operatorname{sgn}(s)) = -k_c \tag{17}$$

We obtained

$$ \dot{V} \le 0 $$

According to *Lemma 1*, the system designed was stable.

The equivalent control input δ_{eq} designed in this study contained the model uncertainty and the unknown disturbance of the system was

$$ \delta_{eq} = -C^T(\lambda\dot{e} + A\dot{p} + Bp + C\delta + D - f(\dot{p}, t) - \Delta f(\dot{p}, t) - d(t) + {}^{C}D^{1-\alpha}(k_c \operatorname{sgn}(s)) \tag{18}$$

By adjusting the magnitude of parameters α and k, the speed of convergence of the system state to the slipform surface could be adjusted to reduce the system overshoot and solve the system jitter problem.

3.2 Neural network optimization and compensation strategies

The RBF neural network has strong approximation ability and fast convergence during training. It has been favored by many scholars as an optimization algorithm in recent years. As shown in Figure 3, the input layer is typically composed of the system source signal.

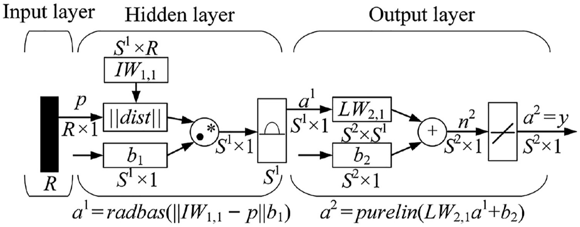


Figure 3.
RBF neural network structure

In this study, the signal of the input layer was designed as $x = \begin{bmatrix} e & \dot{e} \end{bmatrix}$, where e is the vehicle attitude deviation. The implicit layer contained many nodes, and the number of neurons depended on the conditions of the problem to be solved as well as the complexity of the network. The output layer is generally an optimization-type linear function that mainly processes the results of the implicit layer and weighted output externally (Liu *et al.*, 2022).

In this study, we used the Gaussian kernel function as the activation function of the network and set the system perturbation as $\Delta F(\dot{p}, t)$. We used the neural network to approximate $\Delta F(\dot{p}, t)$ as follows:

$$h_i(\sigma_1) = \exp\left(\frac{\|p - c_i\|^2}{2b_i^2}\right) \tag{19}$$

$$\Delta F(\dot{p}, t) = W^{*T}h(p) + V^{*T}h(p) + \varepsilon \tag{20}$$

where p is the external signal input, c_i is the central position of the ith neuron, and b_i is the width of the ith neuron.

We defined the $\Delta F(\dot{p}, t) = \Delta f(\dot{p}, t) + d(t)$. $h(p)$ is the output signal after Gaussian transformation. W^* and V^* are the ideal weights for the network, ε is the approximation error of the network, which satisfies $\varepsilon \leq \varepsilon_N$.

The update law of the design network weight matrix is (Zhao *et al.*, 2022)

$$W = \frac{\sigma_1 h(\sigma_1)}{\eta m} \tag{21}$$

The RBF neural network includes W, b_i and c_i, and three types of parameters must be updated online. The above equation gives the update rule for W, b_i and c_i can be updated by gradient descent (Ji *et al.*, 2018).

In this study, we combined the idea of fractional-order calculus to design a fractional-order gradient descent algorithm to update the neuron center position c_i and width b_i of the RBF neural network, which needs to first find the fractional-order gradient. We defined the performance objective function using the improved iteration rule as

$$E = \frac{1}{2}\left(\sigma_1 \dot{\sigma}_1\right)^2 \tag{22}$$

$$T_{k+2} = T_{k+1} - \rho\, {}^{C}_{T_k}D^{\alpha}_{T_{k+1}} E \tag{23}$$

where ρ is the iteration step, $\rho > 0$.

The updated values at the k+1th moment are as follows:

$${}^{C}_{T_k}D^{\alpha}_{T_{k+1}} E = \sum_{i=0}^{\infty} \frac{E^{(i+1)}(T_k)}{\Gamma(i + 2 - \alpha)} (T_{k+1} - T_k)^{i+1-\alpha} \tag{24}$$

To meet the practicality of the algorithm, as $T_k - T_{k-1}$ goes to 0 and simplify the formula, we can obtain

$$T_{k+2} = T_{k+1} - \frac{\rho}{\Gamma(2 - \alpha)} E^{(1)}(T_{k+1})(T_{k+1} - T_k)^{i+1-\alpha} \tag{25}$$

where $1/\Gamma(2 - \alpha)$ can be considered as part of the step size. Thus, Equation (25) can be further transformed into

$$T_{k+2} = T_{k+1} - \rho E^{(1)}(T_{k+1})(T_{k+1} - T_k)^{i+1-\alpha} \tag{26}$$

To improve the stability of the algorithm and eliminate the singularity of the algorithm itself, a deviation ϕ was added to the above equation to guarantee the gradient descent criterion. Then, Equation (26) can be written as

$$T_{k+2} = T_{k+1} - \rho E^{(1)}(T_{k+1})(|T_{k+1} - T_k| + \phi)^{i+1-\alpha} \tag{27}$$

where $\rho > 0, 0 < \alpha < 2$.

Equation (27) is the updated iteration process based on the gradient descent method defined by Caputo. Applying Equation (27) to the RBF neural network, the update formulas for the width b_i and center position c_i of the neurons are as follows:

$$b_i(k+2) = b_i(k+1) - rt_1 \frac{\partial E}{\partial b_i} \times (|b_i(k+1) - b_i(k)| + \phi)^{1-\alpha} \tag{28}$$

$$c_i(k+2) = c_i(k+1) - rt_2 \frac{\partial E}{\partial c_i(k+1)} \times (|c_i(k+1) - c_i(k)| + \phi)^{1-\alpha} \tag{29}$$

Equations (28) and (29) are the update rules for the neuron width and position parameters of the novel gradient-descent method. The designed network output is:

$$\widehat{F}(\dot{p}, t) = \widehat{W}^T h(p) + \widehat{V}^T h(p) \tag{30}$$

where we defined $\widetilde{F}(\dot{p}, t) = \Delta F(\dot{p}, t) - \widehat{F}(\dot{p}, t)$, and the unknown disturbance $\Delta F(\dot{p}, t)$ of the system was compensated using the RBF neural network as follows:

$$\Delta F(\dot{p}, t) - \widehat{F}(\dot{p}, t) = W^{*T} h(p) + V^{*T} h(p) + \varepsilon - \widehat{W} h(p) + \widehat{V} h(p) = \widetilde{W} h(p) + \widetilde{V} h(p) + \varepsilon \tag{31}$$

We defined the Lyapunov function as

$$V = \frac{1}{2} s^T s + \frac{1}{2\gamma_1} \widetilde{W}^T \widetilde{W} + \frac{1}{2\gamma_2} \widetilde{V}^T \widetilde{V} \tag{32}$$

where $\gamma_1, \gamma_2 > 0$, $\widetilde{W} = W^* - \widehat{W}$, $\widetilde{V} = V^* - \widehat{V}$

The design control law δ_{eq} was

$$\delta_{eq} = -C^T(\lambda \dot{e} + A\dot{p} + Bp + D - f(\dot{p}, t) - \widehat{F}(\dot{p}, t) + {}^C D^{1-\alpha}(k_c \mathrm{sgn}(s)) \tag{33}$$

Then, we could obtain

$$\begin{aligned} &\dot{s} = \lambda \dot{e} + A\dot{p} + Bp + C\delta_{eq} + D - f(\dot{p}, t) - \Delta F(\dot{p}, t) = \\ &\lambda \dot{e} + A\dot{p} + Bp + D - f(\dot{p}, t) - \Delta F(\dot{p}, t) + \\ &C(-C^T(\lambda \dot{e} + A\dot{p} + Bp + D - f(\dot{p}, t) - \widehat{F}(\dot{p}, t) + {}^C D^{1-\alpha}\,(k_c \mathrm{sgn}(s))) = \\ &-\widetilde{F}(\dot{p}, t) - {}^C D^{1-\alpha}(k_c \mathrm{sgn}(s)) \end{aligned} \tag{34}$$

We took the adaptive law as

$$\dot{\widehat{W}} = \gamma_1 s h(p), \dot{\widehat{V}} = \gamma_2 s h(p)$$

From the foregoing, we obtained

$$
\begin{aligned}
\dot{V} &= \frac{1}{2}s^T\dot{s} + \frac{1}{2\gamma_1}\widetilde{W}^T\dot{\widetilde{W}} + \frac{1}{2\gamma_2}\widetilde{V}^T\dot{\widetilde{V}} \\
&= s^T[-\widetilde{F}(\dot{p},t) - {}^C D^{1-\alpha}(k_c \mathrm{sgn}(s))] - \frac{1}{\gamma_1}\widetilde{W}^T\dot{\widetilde{W}} - \frac{1}{\gamma_2}\widetilde{V}^T\dot{\widetilde{V}} \\
&= s^T(-\widetilde{W}^T h(p) - \widetilde{V}^T h(p) - \varepsilon - {}^C D^{1-\alpha}(k_c \mathrm{sgn}(s))) - \frac{1}{\gamma_1}\widetilde{W}^T\dot{\widetilde{W}} \\
&\quad - \frac{1}{\gamma_2}\widetilde{V}^T\dot{\widetilde{V}} \leq -s^T\left(\varepsilon + {}^C D^{1-\alpha}(k_c\, \mathrm{sgn}(s))\right) \leq -s^T\varepsilon - {}^C D^{1-\alpha}k_c\left|s^T\right|
\end{aligned}
\tag{35}
$$

We let ${}^C D^{1-\alpha}k_c > |\varepsilon_N|$ obtain $\dot{V} < 0$, which proved that the proposed control law satisfied the Lyapunov stability condition.

4. Simulation and analysis

4.1 Simulation environment

To verify the effectiveness of the sliding mode control algorithm based on the proposed RBF and fractional order calculus optimization, real scenarios and vehicle parameter models were built in Carsim, and different control algorithms were built in Matlab/Simulink respectively. A joint Carsim and Simulink simulation platform was established to complete the comparison between the traditional sliding mode control algorithm and the RBF neural network-based sliding mode. The vehicle parameter response of the traditional sliding mode control algorithm and RBF neural network-based sliding mode control algorithm were compared. The simulation environment of the control system is shown in Figure 4.

The parameters of the vehicle simulation model parameters and the controller and neural network were as shown in Tables 1 and 2, respectively.

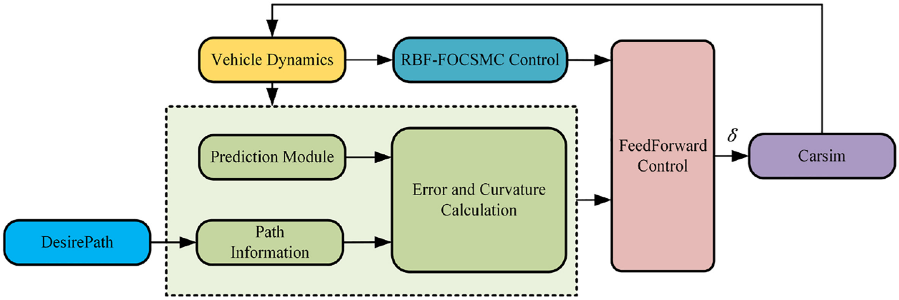


Figure 4. Simulink/Carsim co-simulation environment

Table 1. Vehicle model parameters setting

Parameter	Units	Values
Vehicle overall mass	kg	1,100
Front (rear) wheelbase	mm	1,450 (1,450)
Lateral deflection stiffness of front (rear) tires	$(\mathrm{N{\cdot}rad}^{-1})$	−49000 (−50000)
Steering system ratio	1	15

4.2 Simulation results and analysis

We used a low-speed lane-change simulation condition to simulate the realistic feeling of a driver's active lane-change. We designed three vehicle speeds, 15 km/h, 20 km/h and 25 km/h, to simulate a low-speed lane change, and the simulation results are shown in Figures 5–7.

(1) When the vehicle speed was 15 km/h.

(2) When the vehicle speed was 20 km/h.

(3) When the vehicle speed is 25 km/h.

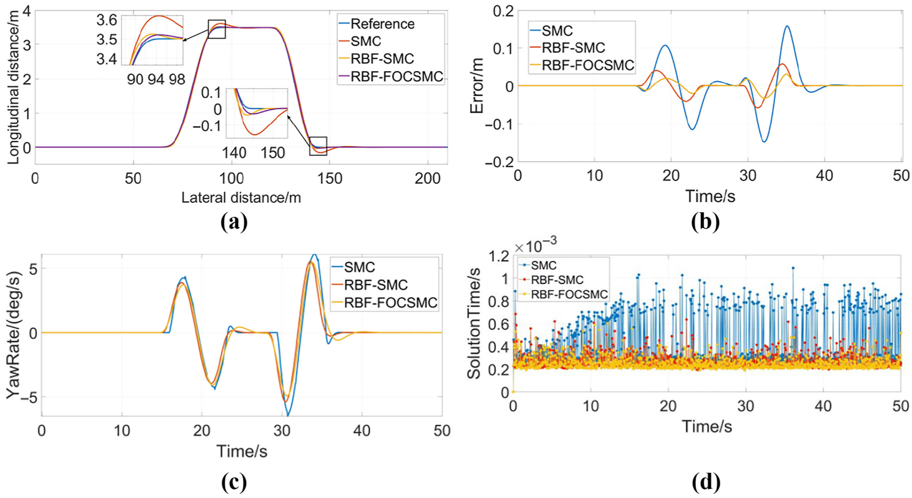
The above simulation environment was set up with three different speed values to simulate vehicle lane changing on low speed, and we compared the path-tracking curves of the vehicle lane changing from Figures 5(a), 6(a) and 7(a). We also compared the following three response parameters, namely, the vehicle path tracking error in Figures 5(b), 6(b) and 7(b); the vehicle transverse sway angle in Figures 5(c), 6(c) and 7(c); and the solving time for each step of the controller in Figures 5(d), 6(d) and 7(d). By comparing the above parameters, we further analyzed the advantages and disadvantages of the three control algorithms, and the specific numerical deviation values are shown in Table 3.

The error of each parameter was obtained from Table 3, the average value of the error was calculated, and the data are shown in Figure 8.

All three control algorithms exhibited good path-tracking capability when the vehicle was running at a set speed value. As shown in Figures 5–7, the conventional sliding mode control

Parameter	Units	Value
Switching variables	λ	Diag(10, 10, 10, 10, 20)
Number of nodes in the hidden layer of the neural network	n	15
Fractional order calculus order	α	0.9
Constants	γ_1, γ_2, k_c	1.5, 10, diag(0.8,4,4,5,5)
Minimal positive value	δ	1×10^{-8}

Table 2.
Controller parameter setting



(a)

(b)

(c)

(d)

Figure 5.
v = 15 km/h parameter response

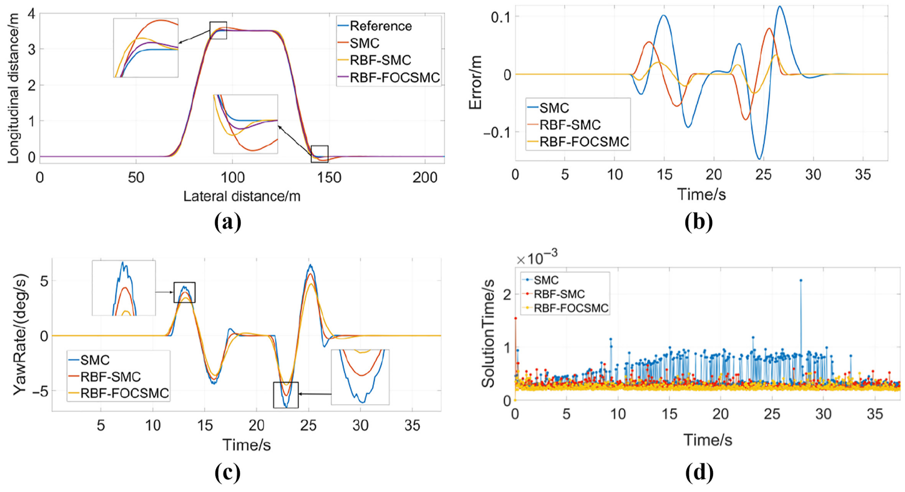


Figure 6.
v = 20 km/h parameter response

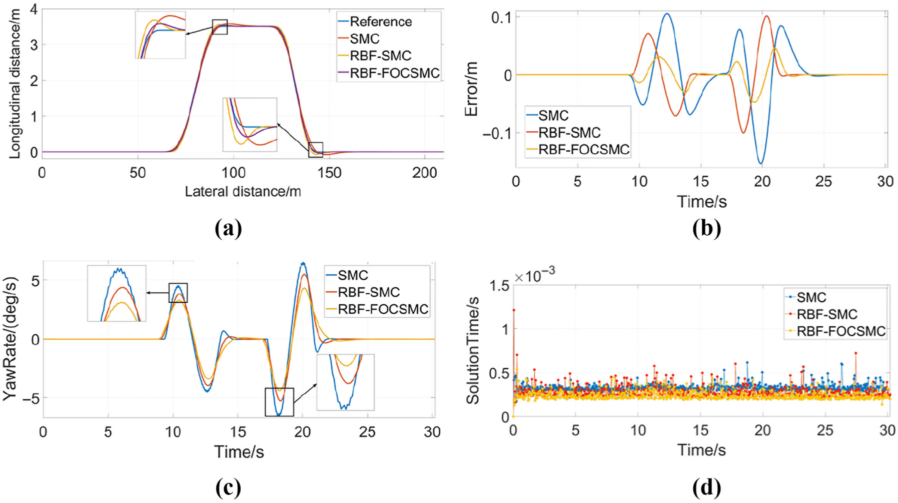


Figure 7.
v = 25 km/h parameter response

Table 3.
Parameter response comparison

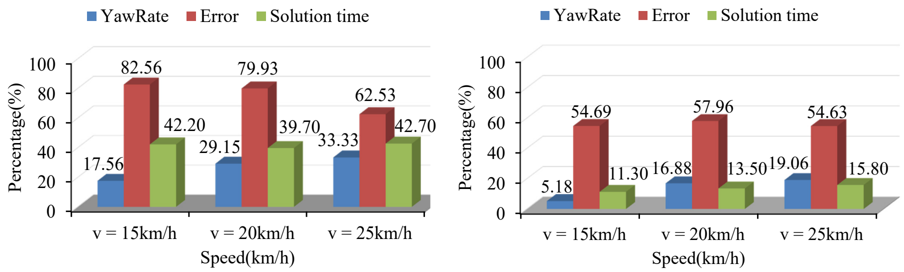
Parameter	Control algorithm	v = 15 km/h	v = 20 km/h	v = 25 km/h
Yaw Rate	SMC	−6.51–6.14	−6.49–6.44	−6.56–6.52
	RBF-SMC	−5.38–5.54	−5.43–5.59	−5.26–5.52
	RBF-FOCSMC	−4.92–5.44	−4.49–4.67	−4.37–4.35
Error	SMC	−0.148–0.158	−0.162–0.167	−0.153–0.104
	RBF-SMC	−0.057–0.058	−0.078–0.079	−0.104–0.101
	RBF-FOCSMC	−0.032–0.020	−0.033–0.033	−0.047–0.046
Mean solution time	SMC	0.435×10^{-3}	0.403×10^{-3}	0.428×10^{-3}
	RBF-SMC	0.283×10^{-3}	0.281×10^{-3}	0.291×10^{-3}
	RBF-FOCSMC	0.251×10^{-3}	0.243×10^{-3}	0.245×10^{-3}

algorithm had significant jitter during turning and exhibited large system delays and values in each parameter of the vehicle lateral control, which was mapped to the actual scenario as a poor driving experience for the driver. The lateral deviation of all the three algorithms was zero when the vehicle was driving in a straight line. When it was steering, the lateral deviation of all the three control algorithms fluctuated. However, the traditional sliding mode control had a larger range of values, with an absolute extreme value of 0.167 m. The absolute extreme values of the lateral deviation of the RBF neural network-based sliding mode control algorithm and the proposed fractional-order optimization-based control algorithm proposed were 0.104 m and 0.046 m, respectively. In terms of the algorithm solution time, the conventional sliding mode controller had the longest step solving time, with the absolute extreme value of 0.435 ms. The absolute extreme value of the proposed fractional order optimization-based control algorithm was 0.243 ms, an of 42.2%, 39.7% and 42.7%, respectively, on that of the conventional SMC control algorithm at the three speeds, 15 km/h, 20 km/h and 25 km/h. The improvements over the RBF-SMC algorithm were 11.3%, 13.5% and 15.8%, respectively.

4.3 Real vehicle test

We considered road driving safety, as shown in Figure 9(a), and performed one-quarter model real vehicle experiments to test the superiority of the RBF-FOCSMC algorithm. The vehicle parameters are presented in Table 4.

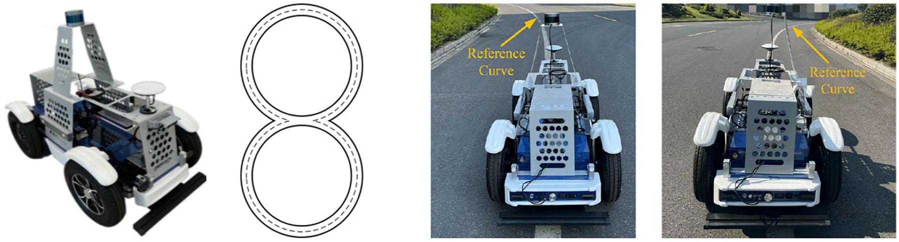
To further compare the RBF-SMC and RBF-FOCSMC algorithms, we set the path for continuous 8-shaped roads to simulate continuous curve conditions to test the tracking performance. We set a low-speed condition of 15 km/h in the experiment, and the experiment results are shown in Figure 10.



(a)

(b)

Figure 8.
(a) Percentage improvement of RBF-FOCSMC vs SMC. (b) Percentage improvement of RBF-FOCSMC vs RBF-SMC



(a)

(b)

(c)

(d)

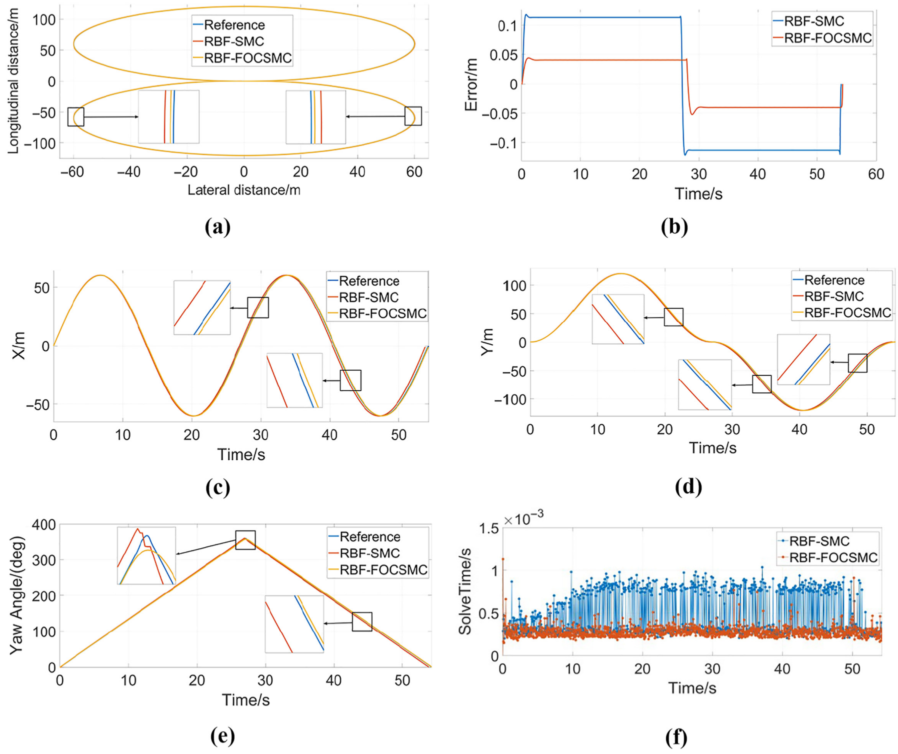
Figure 9.
(a) One-quarter real vehicle model. (b) Schematic diagram of the 8-type curve of the industrial park. (c) Vehicle is driving on the left turn curve in Type "8". (d) Vehicle is driving on the right turn curve in Type "8"

To test the curve path-tracking performance of the proposed algorithm, we designed a real driving scenario on a road with continuous 8-shaped curves to further compare the proposed RBF-SMC and RBF-FOCSMC algorithms. The actual vehicle trajectories of the two algorithms were compared, as shown in Figure 10(a). In Figure 10(b), we compared the path-tracking errors of the two algorithms. In Figure 10(c) and (d), we compared the lateral and longitudinal vehicle displacements of the two algorithms, respectively. In Figure 10(e), we compared the transverse sway angles of the two algorithms. In Figure 10(f), we compared the controller step solution times of the two algorithms. The specific numerical error values of the two algorithms are listed in Table 5.

As shown in Table 5, the extreme values of the path-tracking error of the RBF-SMC algorithm and the RBF-FOCSMC algorithm were 0.120 m and 0.052 m, respectively, in a continuous curved road scenario with a vehicle speed of 15 km/h. The path-tracking performance improved by 59.94%. From Figure 11, it was evident that the RBF-FOCSMC

Table 4.
One-quarter of the experimental vehicle parameters

Parameters	Symbols	Numerical value
Vehicle overall mass	kg	150
Diameter of the tire	mm	450
Vehicle wheelbase	mm	1,000
Vehicle turning radius	m	2.54



(a)

(b)

(c)

(d)

(e)

(f)

Figure 10.
Type 8 road parameters accordingly

algorithm had a smaller error than the reference path. The controller step solution time was compared and the average solution time of the RBF-SMC and RBF-FOCSMC algorithms were 0.549 ms and 0.359 ms, respectively. Thus, the proposed method improved the solving performance by 34.61%.

The above simulation and experiment results showed that the RBF-FOCSMC algorithm outperformed the extant method. The results also revealed that the application of fractional-order calculus can improve the system solution speed. The author applied only the gradient descent method to the control law. From the above simulation results, it can be seen that the designed RBF-FOCSMC converged quickly, compared with the SMC and RBF-SMC algorithms, and the performance of the control system improved significantly.

5. Conclusion

(1) In this paper, the fractional-order calculus was introduced to optimize the sliding mode control algorithm. The proposed method incorporated the RBF neural network to dynamically compensate for external adverse disturbances. The neural network becomes computationally expensive when there are too many external uncertainties. Therefore, the RBF network weights were updated whenever there were problems such as training difficulty and slow convergence speed. Therefore, this paper combines the idea of fractional order calculus with gradient descent algorithm, proposed a fractional-order gradient descent method applied to RBF neural network and fast adjustment the position and width of neurons.

(2) The MATLAB and Carsim simulation platforms were used to build a vehicle lateral control scene to verify the effectiveness of the algorithm and compare it with the traditional SMC and RBF-SMC algorithms using the traditional gradient descent method. Based on the above simulation and experiment results, the proposed optimized RBF-FOCSMC algorithm is superior to both traditional algorithms in terms of system convergence speed and lateral path-tracking accuracy.

Parameter	Control algorithm	v = 15 km/h
Error	RBF-SMC	−0.120–0.117
	RBF-FOCSMC	−0.052–0.043
Mean solution time	RBF-SMC	0.549×10^{-3}
	RBF-FOCSMC	0.359×10^{-3}

Table 5. Comparison of the parameters of the two algorithms

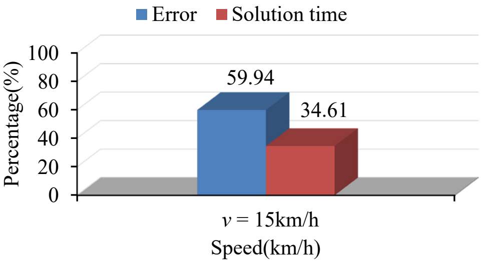


Figure 11. Percentage improvement of RBF-FOCSMC vs RBF-SMC

(3) However, only the performance capability of the control algorithm from the lateral motion and one-quarter of the vehicle model was tested. In future research, we will examine real vehicles under multiple working conditions and driving conditions such as the path-tracking scenario of transverse-longitudinal coupling.

References

Aalizadeh, B. and Asnafi, A. (2018), “Combination of particle swarm optimization algorithm and artificial neural network to propose an efficient controller for vehicle handling in uncertain road conditions”, *Neural Computing and Applications*, Vol. 30 No. 2, pp. 585-593.

Alexandru, C. (2017), “A mechanical integral steering system for increasing the stability and handling of motor vehicles”, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Vol. 231 No. 8, pp. 1465-1480.

Chen, L., Qin, Z.-B., Kong, W.-W. and Chen, X. (2021), “LQR lateral control of intelligent vehicles based on optimal front wheel lateral deflection force”, *Journal of Tsinghua University (Natural Science Edition)*, Vol. 61 No. 9, pp. 906-912.

Fan, B., Zhang, Y., Chen, Y. and Meng, L. (2022), “Intelligent vehicle lateral control based on radial basis function neural network sliding mode controller”, *CAAI Transactions on Intelligence Technology*, Vol. 7 No. 3, pp. 455-468.

Gao, S., Liu, H.-L. and Chen, C. (2017), “Vehicle lateral control method based on adaptive fuzzy sliding mode”, *Control Engineering*, Vol. 24 No. 6, p. 6.

Guo, J., Li, K., Fan, J., Luo, Y. and Wang, J. (2021), “Neural-fuzzy-based adaptive sliding mode automatic steering control of vision-based unmanned electric vehicles”, *Chinese Journal of Mechanical Engineering*, Vol. 34 No. 1, pp. 1-13.

Ji, X., He, X., Lv, C., Liu, Y. and Wu, J. (2018), “A vehicle stability control strategy with adaptive neural network sliding mode theory based on system uncertainty approximation”, *Vehicle System Dynamics*, Vol. 56 No. 6, pp. 923-946.

Lai, F. and Huang, C. (2022), “Path tracking and stability integrated control of intelligent vehicles under steering collision avoidance”, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*.

Li, X. (2021), “Research on integrated control of transverse and longitudinal motion of intelligent vehicles”, *Journal of Chongqing Jiaotong University (Natural Science Edition)*, Vol. 40 No. 4, p. 8.

Liu, L., Wang, J., Zhang, L. and Zhang, S. (2022), “Multi-AUV dynamic maneuver countermeasure algorithm based on interval information game and fractional-order DE”, *Fractal and Fractional*, Vol. 6 No. 5, p. 235.

Luo, Y.T. and Guo, H.W. (2021), “Adaptive neural network sliding mode control for wire-controlled steering systems”, *Journal of South China University of Technology (Natural Science Edition)*, Vol. 49 No. 1, p. 9.

Mozaffari, S., Al-Jarrah, Y.O., Dianati, M., Jennings, P. and Mouzakitis, A. (2022), “Deep learning-based vehicle behavior prediction for autonomous driving applications: a review”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, pp. 33-47.

Muñoz-Vázquez, A.J., Fernández-Anaya, G. and Sánchez-Torres, J.D. (2022), “Adaptive parallel fractional sliding mode control”, *International Journal of Adaptive Control and Signal Processing*, Vol. 36 No. 3, pp. 751-759.

Swain, S.K., Rath, J.J. and Veluvolu, K.C. (2021), “Neural network-based robust lateral control for an autonomous vehicle”, *Electronics*, Vol. 10 No. 4, p. 510.

Tian, Y., Yao, Q., Hang, P. and Wang, S. (2022), “Adaptive coordinated path tracking control strategy for autonomous vehicles with direct yaw moment control”, *Chinese Journal of Mechanical Engineering*, Vol. 35 No. 1, pp. 1-15.

Wang, S., Hui, Y., Sun, X. and Shi, D. (2019), "Neural network sliding mode control of intelligent vehicle longitudinal dynamics", *IEEE Access*, Vol. 7, pp. 162333-162342.

Yang, T., Bai, Z., Li, Z., Feng, N. and Chen, L. (2021), "Intelligent vehicle lateral control method based on feedforward + predictive LQR algorithm", *Actuators*, MDPI AG, Vol. 10 No. 9, p. 228.

Zendehdel, N. and Gholami, M. (2021), "Robust self-adjustable path-tracking control for autonomous underwater vehicle", *International Journal of Fuzzy Systems*, Vol. 23, pp. 216-227.

Zhang, B.L., Lu, M.Y., Cheng, J. and Shen, G. (2019), "Two-point pre-sighted trajectory tracking lateral control system", *Journal of Electronic Measurement and Instrumentation*, Vol. 33 No. 5, pp. 158-163.

Zhao, J., Du, J., Zhu, B., Wang, Z., Chen, Z. and Tao, X. (2022), "Longitudinal cruise control of intelligent vehicles based on adaptive dynamic sliding mode control", *Automotive Engineering*, Vol. 44 No. 1, pp. 8-16.

Zhou, W., Xing, Z., Wenbin, B., Chengchen, D., Xie, Y. and Wu, X. (2019), "Route planning algorithm for autonomous underwater vehicles based on the hybrid of particle swarm optimization algorithm and radial basis function", *Transactions of the Institute of Measurement and Control*, Vol. 41 No. 4, pp. 942-953.

Corresponding author
Qipeng Li can be contacted at: liqipeng2005@163.com