ISSN:2790-1688 Volume-14-(2025)

Path-following control for autonomous vehicles considering the uncertainties of yaw dynamics

Yiming Gong

School of Nanjing University of Science and Technology, Nanjing 210094, China;

18057137398@163.com

Abstract. With the rapid development of autonomous driving technology, achieving precise path-followingcontrol in uncertain environments has become a key challenge. To address the problem mentioned above, this paper proposes a path-following controller based on input-output feedback linearization and neural network-based approximation of uncertainties. It cancels out the nonlinear dynamics by designing a feed-forward control law, in which a neural network is employed to approximate the uncertainties in the vehicle yaw dynamics. High-fidelity co-simulations were carried out using CarSim and Simulink. The results show that after introducing neural network-based compensation, the steady-state error of path-following is reduced by 23%-30%, verifying the effectiveness of the proposed method.

Keywords: Autonomous vehicles; path-following control; neural networks; input-output feedback linearization.

1. Introduction

Path-following control for autonomous vehicles originated from exploring autonomous driving technology, aiming to achieve precise path following in complex environments. Its applications span urban traffic management, logistics, agricultural automation, and more [1]. In recent years, by integrating model predictive control, robust control, and machine learning, path-following control has been continuously optimized to effectively address system uncertainties and external disturbances, enhancing the reliability and adaptability of autonomous vehicles.

There is a wide variety of current research on path-following control methods. The research in [2] proposes a line-of-sight-based adaptive path-following controller that compensates for the drift force through sideslip angle estimation and demonstrates exponential convergence of parameter estimation. However, it assumes that the sideslip angle is a constant parameter and does not consider dynamic uncertainties such as time-varying perturbations. The authors of [3] adopt the least-squares support vector machine method to construct a path-following controller, which utilizes a data-driven model to reduce dependence on an accurate dynamic model. However, it does not introduce a robust mechanism to deal with the model and external disturbances. Literature [4] improves the path following performance by mixing the Pure Pursuit and Stanley controllers. However, it does not quantitatively analyze the effects of road friction coefficient changes and tire nonlinear characteristics on control stability. In [5], researchers propose a fast nonlinear model predictive control framework but fail to consider the uncertainties of the tire magic formula parameter change and longitudinal/lateral coupling dynamics. As for [6], researchers systematically investigate the relationship between path tolerance and vehicle dynamics model complexity, revealing the applicability of simple models under loose tolerances. However, it does not establish an uncertainty propagation model to analyze the path following error boundaries and does not address the effects of realistic constraints such as actuator delays. Existing studies generally suffer from inadequate modeling of uncertainty factors such as time-varying environmental perturbations, sensor noise, and tire-pavement interaction parameter uptake, and lack a unified adaptive robust framework to achieve synergistic suppression of multi-source uncertainty.

Neural network-based approximation of uncertainties is a good method to address the above problems. Leifsson proposes a novel adaptive global surrogate modeling algorithm [7] that efficiently guides neural network-based prediction by cyclically refining sampling points. Chen proposes a neural network-based epistemic uncertainty estimation method, which efficiently quantifies model

uncertainty through low-rank representation and last-layer Laplace approximation [8]. Zhang proposes a Bayesian neural network-based method, which efficiently estimates epistemic uncertainty in deep convolutional neural networks via α-divergence objectives and hierarchical sampling [9]. Literature [10] proposes a Bayesian convolutional neural network-based approach, which introduces a novel uncertainty metric derived from posterior predictive samples to effectively discriminate between correct and incorrect predictions in classification tasks.

To address the above issues, this paper proposes a path-following controller for autonomous vehicles using input-output feedback linearization, and a neural network is employed to approximate uncertainty. The contribution of this paper can be summarized as follows.

(1) Proposed a path-following control method based on input-output feedback linearization and conducted high-fidelity co-simulation to verify its effectiveness.

(2) Approximated the uncertainties of vehicle yaw motion using a neural network and compensated in the control law, which reduces the effect of uncertainties on path-following accuracy.

In this article, the vehicle model is introduced in Section 2. A path-following controller for autonomous vehicles is designed in Section 3. The high-fidelity co-simulation results are presented in Section 4. Finally, we summarized and analyzed this study in Section 5.

2. Vehicle Modeling

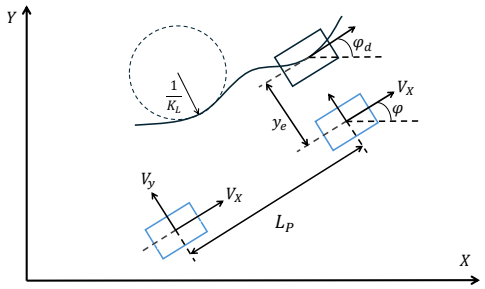


Fig. 1 Pre-scan-based kinematics model

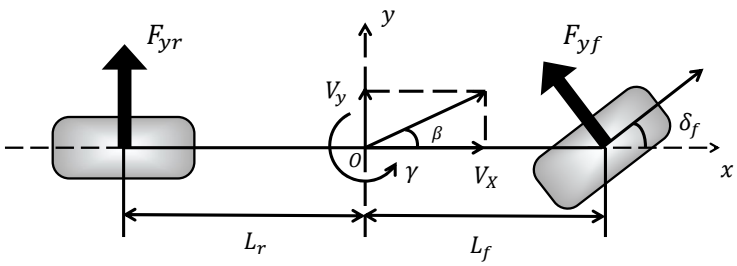


Fig. 2 Dynamics model

Equations (1a) and (1b) describe the kinematic model.

$$\dot{y}_e = V_X\varphi_e - V_X\beta - L_P\gamma \tag{1a}$$

$$\dot{\varphi}_e = V_X K_L - \gamma \tag{1b}$$

where y_e is the (straight-line) deviation of the center of gravity of the pre-scanned vehicle from the desired trajectory. φ_e is the vehicle heading angle deviation. K_L is the road curvature. V_X is the vehicle longitudinal speed. γ is the vehicle yaw rate. β is the vehicle sideslip angle.

Equations (2a), (2b), (2c), and (2d) describe the dynamics model.

$$\dot{\beta} = \frac{F_{yf}(\beta, y, V_X, \delta_f) + F_{yr}(\beta, \gamma, V_X)}{mV_X} - \gamma \tag{2a}$$

$$\dot{\gamma} = \frac{L_f F_{yf}(\beta, y, V_X, \delta_f) + F_{yr}(\beta, \gamma, V_X)}{I_z} \tag{2b}$$

$$F_{yf} = -2C_f\alpha_f = -2C_f\left(arctan\left(\beta + \frac{L_f}{V_X}\gamma\right) - \delta_f\right) \tag{2c}$$

$$F_{yr} = -2C_r\alpha_r = -2C_r\left(\arctan\left(\beta - \frac{L_f}{V_X}\gamma\right)\right) \tag{2d}$$

where m is the vehicle mass. I_z is the moment of vehicle inertia about the vertical ground axis. L_f is the distance from the vehicle's center of gravity to the front axle. L_r is the distance from the vehicle's center of gravity to the rear axle. C_f is the corning stiffness of front tires. α_f is the tire sideslip angle of front wheel. δ_f is the front wheel steering angle.

After sorting, we can obtain the following state equation.

$$\dot{x} = f(x) + gu + EK_L \tag{3}$$

where,

$$f(x) = \begin{bmatrix} V_X x_2 - V_X x_3 - L_P x_4 \\ -x_4 \\ \dfrac{-2C_f \arctan\left(x_3 + \frac{L_f}{V_X} x_4\right) - 2C_r \arctan\left(x_3 - \frac{L_r}{V_X} x_4\right)}{mV_X} - x_4 \\ \dfrac{-2L_f C_f \arctan\left(x_3 + \frac{L_f}{V_X} x_4\right) + 2L_r C_r \arctan\left(x_3 - \frac{L_r}{V_X} x_4\right)}{I_z} \end{bmatrix} := \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

$$x = \mathrm{col}(y_e, \varphi_e, \beta, \gamma) := \mathrm{col}(x_1, x_2, x_3, x_4) \ , \ g = \mathrm{col}(0,0, \tfrac{2C_f}{mV_X}, \tfrac{2L_f C_f}{I_z}) := \mathrm{col}(g_1, g_2, g_3, g_4) \ ,$$

$$E = \mathrm{col}(0, V_X, 0, 0) \ , \ u = \delta_f$$

3. Path-Following Controller Design

3.1 Overview

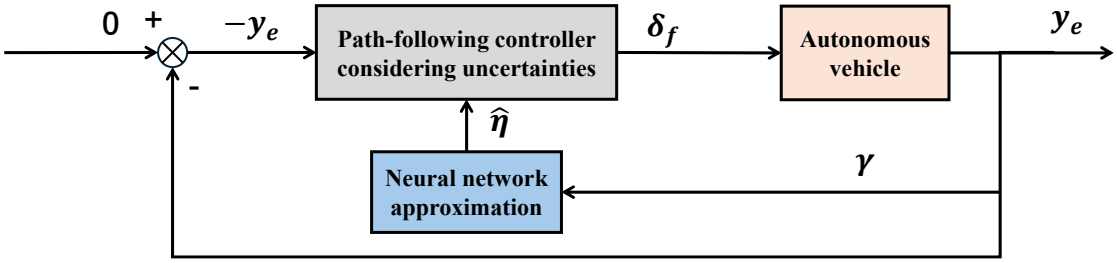


Fig. 3 The proposed control strategy

To realize the path-following control, we design a unit negative feedback control system as Fig. 3. In this control system, we use δ_f as the system input and y_e as the controlled variable. We want to achieve zero deviation from the vehicle's center of gravity to the desired trajectory, so we set the setting value to zero. In the neural network-based approximation of uncertainties, we take γ as input and add the output $\hat{\eta}$ to the control rate to compensate for it.

3.2 Feed-forward Control Based on Input-Output Feedback Linearization

Without loss of generality, we provide the following analysis. Firstly, we select x_1 and x_2 as the controlled variables and adjust the ratio of the two by ϖ:

$$y = x_1 + \varpi \cdot x_2 \tag{4}$$

Next, we find the mapping of the control input u to the system outputs:

$$\ddot{y} = V_X \cdot (f_2 - f_3) - (L_P + \varpi) \cdot f_4 + V_X^2 \cdot K_L + \varpi \cdot V_X \cdot \dot{K}_L - (V_X \cdot g_3 + (L_P + \varpi) \cdot g_4) \cdot u \tag{5}$$

To cancel the nonlinear dynamics, we design control input as $u = u_1 + u_2$, where

$$u_1 = \frac{V_X \cdot (f_2 - f_3) - (L_P + \varpi) \cdot f_4 + V_X^2 \cdot K_L + \varpi \cdot V_X \cdot \dot{K}_L}{V_x \cdot g_3 + (L_P + \varpi) \cdot g_4} \tag{6}$$

3.3 Feedback Control Based on Lyapunov Analysis

After simplification, we get the mapping from the input u_2 to the output y

$$\ddot{y} = -(V_x \cdot g_3 + (L_P + \varpi) \cdot g_4) \cdot u_2 \tag{7}$$

We provide the following state-space model

$$\dot{\boldsymbol{e}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{e} + \begin{bmatrix} 0 \\ \theta \end{bmatrix} u_2 \coloneqq \boldsymbol{f_e e} + \boldsymbol{g_e} u_2 \tag{8}$$

where $\boldsymbol{e} = \begin{bmatrix} y \\ z \end{bmatrix}, z = \dot{y},\ \theta = -(V_X \cdot g_3 + (L_P + \varpi) \cdot g_4),\ \dot{z} = \theta \cdot u_2.$

We define the gain vector as

$$K = [k_1 \quad k_2] \tag{9}$$

and

$$u = -Ke \tag{10}$$

To determine a suitable $\boldsymbol{K}$ to stabilize the system, we design the first Lyapunov function:

$$V_1 = \frac{1}{2} y^2 \tag{11}$$

Its time derivative can be obtained as

$$\dot{V}_1 = y \cdot z \tag{12}$$

We define z as a virtual control input z^*, and

$$z^* = -k_3 y \tag{13}$$

Then the time derivative of the first Lyapunov function becomes

$$\dot{V}_1 = -k_3 y^2 = -2k_3 V_1 \tag{14}$$

whereby the system is stable.

To make z equivalent to z^*, we study the second Lyapunov function:

$$V_2 = \frac{1}{2} y^2 + \frac{1}{2} e_z^2, \qquad e_z \coloneqq z - z^* \tag{15}$$

Its time derivative is

$$\dot{V}_2 = -k_3 y^2 + y \cdot e_z + e_z(\theta \cdot u_2 + k_3 z) \tag{16}$$

Therefore, we can take the control law as

$$u_2 = -\frac{1}{\theta} k_3 z - \frac{1}{\theta} y - \frac{k_4}{\theta} e_z \tag{17}$$

Organize it, and we can obtain

$$u_2 = -\frac{1}{\theta}(k_3 + k_4) z - \frac{1}{\theta}(1 + k_3 k_4) y \tag{18}$$

3.4 Uncertainties approximation based on a neural network

The uncertainties in path-following of autonomous vehicles mainly stem from model simplification, parameter perturbations, and environmental disturbances. Specifically, they include:

(1) The fourth-order simplified model does not cover the pitch and roll dynamics and the nonlinear characteristics of tire forces.

(2) There are time-varying deviations between the actual and nominal values of key parameters such as cornering stiffness and mass.

(3) External conditions like road surface friction coefficient and wind disturbances are unmeasurable.

The above uncertainties may reduce the path-following accuracy. Therefore, we use a neural network to approximate the uncertainties and compensate for them in the control law, aiming to improve the path-following accuracy of vehicles in uncertain environments.

In the existing state equations, considering that the disturbances acting on the yaw motion are complex (including environmental and road surface disturbances, the nonlinear characteristics of tire forces, and vehicle dynamics coupling), we use the variable η to represent the lumped uncertainties of the yaw rate:

$$\dot{x} = f(x) + EK_L + gu + \mathrm{col}(0,0,0,1)\eta \tag{19}$$

Here, the uncertainties can be represented by a neural network as

$$\eta = W^{\top}\varphi(x) + \epsilon \tag{20}$$

where $W = \mathrm{col}(W_1, W_2, \dots, W_m)$ represents the weight vector, $\varphi = \mathrm{col}(\varphi_1, \varphi_2, \dots, \varphi_m)$ represents the activation function vector. ϵ is the inherent approximation error of the neural network. Since ϵ is unmeasurable, $\hat{\eta}$ can be expressed approximately as

$$\hat{\eta} = \hat{W}^{\top}\varphi(x) \tag{21}$$

During uncertainties approximation, we introduce the error filtering online learning model to update the weights. Firstly, we establish the parameterization model and the measurement model. The parameterization model is

$$\dot{\lambda} = -\Theta_{\lambda}\lambda + \Theta_{\lambda}W^{\top}\varphi + \Theta_{\lambda}\epsilon \tag{22}$$

Since ϵ is unmeasurable, we design the online learning model:

$$\dot{\hat{\lambda}} = -\Theta_{\lambda}\hat{\lambda} + \Theta_{\lambda}\hat{W}^{\top}\varphi \tag{23}$$

The measurement model is

$$\dot{\zeta}_x = -\Theta_{\lambda}\zeta_x - \Theta_{\lambda}x - (f(x) + g(x)u) \tag{24a}$$

$$\lambda_x = \Theta_{\lambda}\zeta_x + \Theta_{\lambda}x \tag{24b}$$

In this way, we can obtain the error dynamics model:

$$\dot{\tilde{\lambda}} = \dot{\hat{\lambda}} - \dot{\lambda} = -\Theta_{\lambda}\tilde{\lambda} + \Theta_{\lambda}\tilde{W}^{\top}\varphi - \Theta_{\lambda}\epsilon \tag{25}$$

Next, we provide the theoretical analysis of approximation performance. We define the following Lyapunov function:

$$V_W = \frac{1}{2}k_W \mathrm{tr}\{\tilde{\lambda}^{\top}\Theta_{\lambda}^{-1}\tilde{\lambda}\} + \frac{1}{2}\mathrm{tr}\{\tilde{W}^{\top}K^{-1}\tilde{W}\} \tag{26}$$

where $\tilde{W} = \hat{W} - W$, representing the error between the estimated and actual values of the weights. $K = \mathrm{col}(k, \dots, k)$, and $k > 0$. Obviously, V_W is positive-definite.

The time derivative of the Lyapunov function is

$$\dot{V}_W = -k_W\tilde{\lambda}^2 - k_W\tilde{\lambda}\epsilon + \tilde{W}^{\top}\left(k_W\tilde{\lambda}\varphi + K^{-1}\dot{\hat{W}}\right) - \tilde{W}^{\top}K^{-1}\dot{W} \tag{27}$$

We assume that there exist constants $b_1 > 0, b_W > 0,$ and $b_{\dot{W}} > 0$, such that $|\epsilon| < b_1, ||W|| < b_W,$ and $||\dot{W}|| < b_{\dot{W}}$. Then, we need to deal with $-k_W\tilde{\lambda}\epsilon$ and $-\tilde{W}^{\top}K^{-1}\dot{W}$. Given that the following inequalities hold:

$$-k_W\tilde{\lambda}\epsilon \leqslant k_W|\tilde{\lambda}||\epsilon| \tag{28a}$$

$$-\tilde{W}^{\top}K^{-1}\dot{W} \leqslant \sum_i \frac{1}{k}|\tilde{W}_i||\dot{W}_i| \leqslant \sum_i \frac{1}{k}|\hat{W}_i - W_i||\dot{W}_i| \leqslant \frac{m}{k}(b_{\dot{W}} + b_W)b_{\dot{W}} \tag{28b}$$

We define the following adaptive law

$$\dot{\hat{W}}_i = \mathcal{P}_i\left[-Kk_W\tilde{\lambda}\varphi\right] \tag{29a}$$

$$\mathcal{P}_i[*_j] = \begin{cases} *_j, & \text{if } \hat{W}_{i,j} \in (\hat{W}_{i,\min}, \hat{W}_{i,\max}), \\ & \text{or if } \hat{W}_{i,j} = \hat{W}_{i,\min} \text{ and } *_j \geqslant 0, \\ & \text{or if } \hat{W}_{i,j} = \hat{W}_{i,\max} \text{ and } *_j \leqslant 0, \\ 0, & \text{otherwise} \end{cases} \tag{29b}$$

where $*_j$ represents the jth component of the variable *.

We use the projection operator (29b) to ensure that W is bounded, and then we have

$$\dot{V}_W \leqslant -k_W\tilde{\lambda}^2 + \tilde{W}^{\top}\left(k_W\tilde{\lambda}\varphi + K^{-1}\dot{\hat{W}}\right) + k_W|\tilde{\lambda}||\epsilon| + \frac{m}{k}(b_{\dot{W}} + b_W)b_{\dot{W}} \tag{30}$$

According to this, we can obtain

$$\dot{V}_W \leqslant -k_W\tilde{\lambda}^2 + k_W|\tilde{\lambda}|b_{\epsilon} + \frac{m}{k}(b_{\dot{W}} + b_W)b_{\dot{W}} \tag{31}$$

The above inequality is a unary quadratic function of $|\tilde{\lambda}|$. According to the properties of a quadratic function of one variable, the function opens downward and has a positive root $|\tilde{\lambda}_1|$. When $\tilde{\lambda} < |\tilde{\lambda}_1|$, the function value is greater than zero, $\dot{V}_W$ is greater than zero, and $\tilde{\lambda}$ diverges; when $\tilde{\lambda} > |\tilde{\lambda}_1|$, the function value is less than zero, and $\tilde{\lambda}$ decreases. Therefore, $\tilde{\lambda}$ is bounded.

4. Simulation Results

High-fidelity co-simulation between CarSim and Matlab/Simulink has been performed to validate the effectiveness of the proposed path-following controller. The relevant parameters are shown in Table 1.

Table 1 Parameters of the high-fidelity vehicle

Parameters	Values	Units
m	1650	kg
Iz	3234	kg-m2
L_P	5	m
C_f	40000	N/rad
C_r	35000	N/rad
L_f	1.4	m
L_r	1.65	m

4.1 Without Considering Uncertainties

We have already known that

$$u_2 = -\boldsymbol{K} \cdot \boldsymbol{e} = -\begin{bmatrix} \frac{1 + k_3 k_4}{\theta} & \frac{k_3 + k_4}{\theta} \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \tag{32}$$

As shown in Fig. 4, we chose a circular field with a radius of 30 meters for the control test. Then, we choose different controller gians $\boldsymbol{K}$ to analyze the control performance.

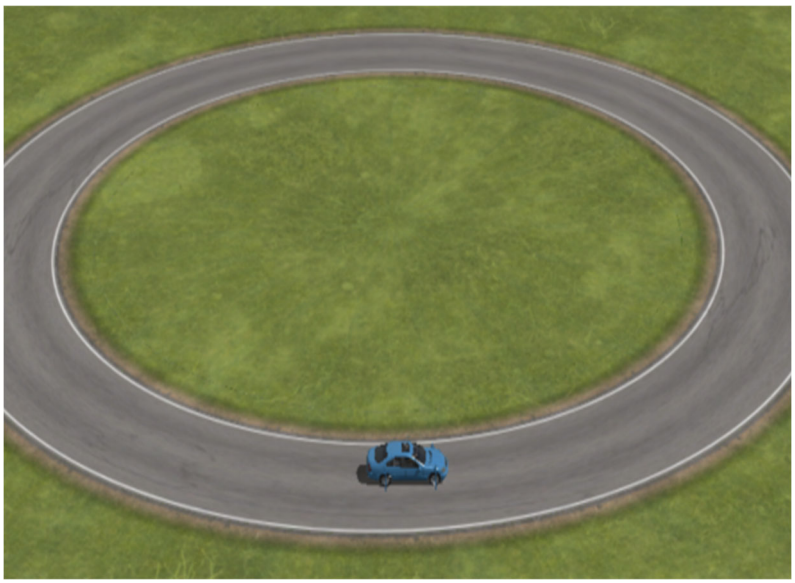


Fig. 4 The first simulation scanerio

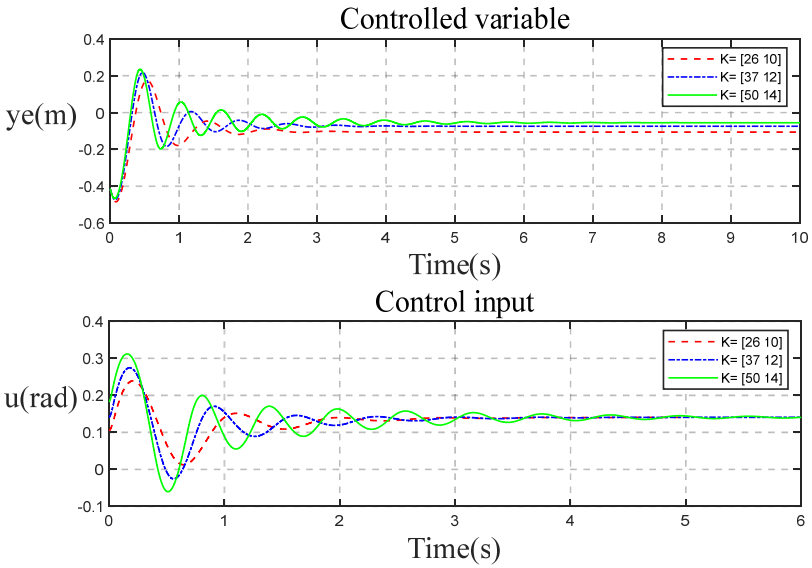


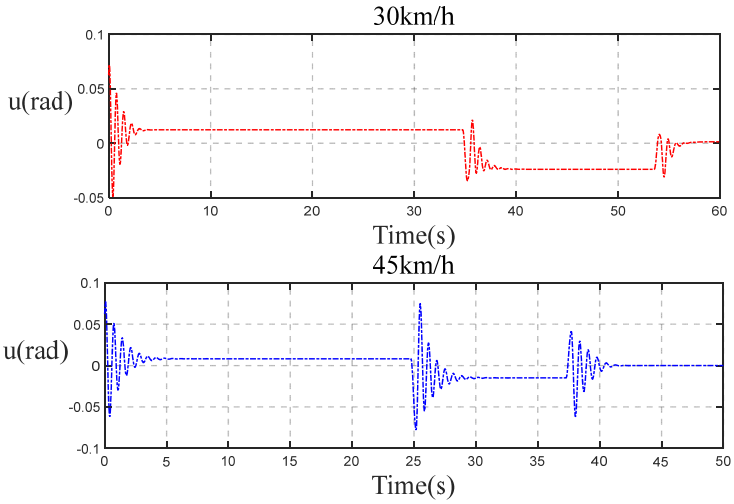
Fig. 5 Control performance with different controller gains

Fig. 5 shows that the adjustment time of the controller is between $t = 2.5\text{s}$ to $t = 6\text{s}$ and the stable state error is between $y_e = 0.1\text{m}$ to $y_e = 0.05\text{m}$. When the values of k_1 and k_2 are appropriate, the longer the adjustment time, the smaller the stabilization error.

It can be seen that the smaller the steady-state error, the larger the oscillation amplitude of the control input u and the longer the oscillation time. However, the steady-state value of the control input u is the same in all three cases and is about 0.14m.

Furthermore, we selected the controller gain as $K = [37 \quad 12]$. As shown in Fig. 6, we set the road section to have two 90-degree turns (one left-hand turn with a radius of 200m and one right-hand turn with a radius of 100m). Then we validated the control performance of the controller at different vehicle speeds.

Fig. 6 The second simulation scanerio



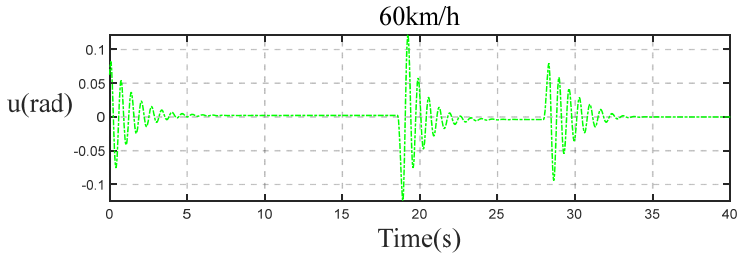


Fig. 7 Control performance with different vehicle speeds

It can be seen from Fig. 7 that when the vehicle speed is maintained between 30 km/h and 60 km/h, the higher the vehicle speed, the longer the adjustment time within the curve, and the smaller the steady-state error.

4.2 Considering Uncertainties

We selected different control gains and compared the control performance before and after adding the neural network-based approximation of uncertainties.

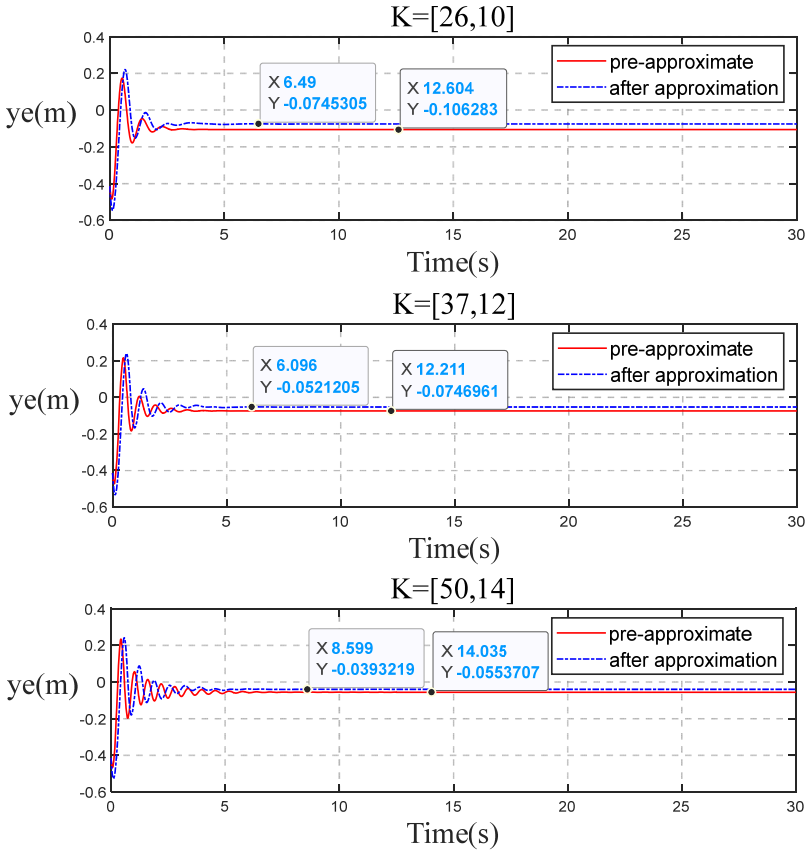


Fig. 8 Control performance with different controller gains

Fig. 8 shows that after introducing the neural network-based approximation, the system adjustment time remains almost unchanged. This is because the disturbances are complex, and the approximation effect of uncertainties during the transient stage is mediocre. However, the steady-state error y_e is reduced by 23% to 30%. This is because the approximation effect of uncertainties is better during the steady-state stage.

5. Conclusions

This paper presents a path-following control method for autonomous vehicles considering the uncertainties of yaw dynamics. The proposed approach combines input-output feedback linearization

with neural network-based approximation of uncertainties to address nonlinear dynamics and uncertainties, such as model simplification, parameter perturbations, and environmental disturbances. The controller design integrates feed-forward and feedback control laws to linearize the system, while the neural network effectively approximate the uncertainties and compensate for them in the control law, reducing steady-state following errors by 23%-30%. Co-simulation results validate the effectiveness of the proposed controller, demonstrating improved accuracy and robustness in complex driving scenarios. This work enhances the reliability and adaptability of autonomous vehicle path-following systems under real-world conditions.

References

[1] Zhang H, Braun, SG. Signal processing and control challenges for smart vehicles.Mechanical Systems and Signal Processing. Vol. 87 (2017), p. 1-3.

[2] Fossen, TI, Pettersen, KY, Galeazzi, R. Line-of-sight path following for Dubins paths with adaptive sideslip compensation of drift forces. IEEE Transactions on Control Systems Technology. Vol. 23 (2015), p. 820-827.

[3] Qian Shi, Hui Zhang. An LS-SVM control method for path following of autonomous ground vehicles. 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall). Victoria, BC, Canada, 18 Nov.-16 Dec. 2020, p. 5.

[4] Domina, A, Tihanyi, V. Path-following controller for autonomous vehicles. 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE). Graz, Austria, 4-8 Nov. 2019, p. 5.

[5] Ningyuan Guo, Xudong Zhang, Yuan Zou. A computationally efficient path-following control strategy of autonomous electric vehicles with yaw motion stabilization. IEEE Transactions on Transportation Electrification. Vol. 6 (2020), p. 728-739.

[6] Lundahl K, Frisk E, Nielsen L. Implications of path tolerance and path characteristics on critical vehicle manoeuvres.Vehicle System Dynamics. Vol. 55 (2017), p. 1909-1945.

[7] Leifsson L, Nagawkar J, Barnet L, et al. Global surrogate modeling by neural network-based model uncertainty. Computational Science. Vol. 13352 (2022), p. 425-434.

[8] Chen Y, Yu S, Eshraghian J.K, et al. Sparse subnetwork inference for neural network epistemic uncertainty estimation with improved Hessian approximation. APL Machine Learning. Vol. 2 (2024), p. 026106.

[9] Ni Zhang, Xiaoyi Chen, Li Quan. Layerwise Approximate inference for bayesian uncertainty estimates on deep neural networks. 2021 International Joint Conference on Neural Networks (IJCNN). Shenzhen, China, 18-22 July 2021, p. 8.

[10] de Waal A, Steyn C. Uncertainty measurements in neural network predictions for classification tasks. 2020 IEEE 23rd International Conference on Information Fusion (FUSION). Rustenburg, South Africa, 6-9 July 2020, p. 7.