Iranian Journal of Fuzzy Systems

Volume *, Number *, (****), pp. **

Original paper

Car-following systems using a multilayer type-2 fuzzy neural network integrated with recurrent neural networks

N. V. Binh [1] and T. L. Le [2]

[1,2] *Faculty of Mechatronics Engineering, Lac Hong University, Dong Nai, Vietnam*
[1,2] *Faculty of Postgraduate Studies, Lac Hong University, Dong Nai, Vietnam*

ngovanbinh@lhu.edu.vn, tienloc@lhu.edu.vn

Abstract

Car-following systems play a critical role in advanced driver assistance systems and autonomous driving applications, where accurate detection and stable tracking of the preceding vehicle are essential for ensuring safety and smooth motion. This paper presents a low-cost vision-based car-following control system that combines monocular vision sensing with an intelligent speed controller. A convolutional neural network detects the target vehicle and extract spatial features from input images. The resulting distance measurements are then refined using a Kalman filter to suppress noise and enhance estimation stability. Based on the visual feedback, the vehicle speed is regulated by a Recurrent Multilayer Interval Type-2 Fuzzy Neural Network (RMIT2FNN), the enabling adaptive responses to the motion of the leading vehicle and maintenance of a safe following distance. The proposed approach is first validated through simulations, in which its tracking performance is compared with conventional FNN and IT2FNN controllers under the same operating conditions. The simulation results demonstrate improved tracking accuracy and rapid responses to velocity variations. Subsequently, real-time experimental validation is conducted on a 1/10-scale vehicle platform, confirming that the system can detect and track the preceding vehicle stably while maintaining the desired following distance under indoor operating conditions. These findings provide preliminary evidence of the feasibility of the proposed method as a cost-efficient vision-based car-following solution under controlled experimental conditions.

Keywords: Interval type-2 fuzzy system, recurrent neural networks, adaptive control, trajectory tracking, car-following.

1 Introduction

Car-following is a fundamental function in advanced driver assistance systems (ADAS) and autonomous driving, because it governs how a host vehicle regulates its longitudinal motion with respect to a preceding vehicle [4, 16, 17, 20, 23, 27]. From a research perspective, car-following has evolved from classical stimulus-response, safety-distance, and psychophysical models toward adaptive cruise control (ACC/CACC) strategies and, more recently, data-driven learning-based approaches [2, 7, 15]. Recent review studies have further shown that car-following research now spans traffic flow theory, control engineering, cognitive modeling, and machine learning [31]. Despite this broad literature, achieving safe, smooth, and robust following performance under nonlinear vehicle dynamics, heterogeneous traffic conditions, and uncertain sensing remains a persistent challenge [1, 5, 25].

Among intelligent control techniques, fuzzy logic has long been considered attractive for automotive applications because it can encode heuristic expert knowledge through linguistic IF–THEN rules and does not require an exact mathematical model of the plant [29]. In automotive engineering, fuzzy methods have been used in driver identification [26], vehicle dynamics control [30], environment recognition [28], and other tasks in which the available information is limited or imprecise. In the specific context of car-following, the study by Mădălin-Dorin Pop demonstrates that the

Corresponding Author: T. L. Le

Received: April 2026; Revised: August 2026; Accepted: September 2026.

https://doi.org/10.22111/ijfs.2026.55222.9795

selection of defuzzification methods significantly affects the calibration accuracy of Mamdani FIS-based car-following models, with the SoM method achieving lower MAE and RMSE. This result highlights the importance of designing efficient fuzzy inference mechanisms to improve the performance of intelligent car-following systems [14]. However, conventional type-1 fuzzy systems still face limitations when the membership functions, rule bases, or measurements are affected by uncertainty, noise, and time-varying operating conditions [7, 8, 24].

To address these limitations, type-2 fuzzy logic systems extend type-1 fuzzy sets by allowing the membership grade itself to be fuzzy. In particular, interval type-2 fuzzy logic systems provide a practical balance between uncertainty-handling capability and computational tractability. Foundational studies have shown that interval type-2 fuzzy systems are especially useful when measurements are corrupted by nonstationary noise, when uncertainty exists in the antecedent or consequent membership functions, or when the rule base is not precisely defined [3, 10, 19, 22]. These properties are highly relevant to vehicle-following control, where perception errors, actuator nonlinearities, and varying traffic scenarios can significantly degrade the performance of a purely type-1 controller. Therefore, interval type-2 fuzzy control offers a promising framework for enhancing robustness in longitudinal vehicle control problems.

Alongside fuzzy logic, recurrent neural networks (RNNs) provide a complementary mechanism for learning temporal dependencies from sequential data [12]. Because recurrent connections preserve information from previous states, RNNs are well-suited to dynamic control and prediction problems in which the current action depends not only on instantaneous measurements but also on recent history [21]. This temporal-memory characteristic is particularly important for car-following, since safe and smooth longitudinal behavior depends on the evolution of relative distance, relative speed, and leader motion over time. Prior studies have shown that RNN-based car-following models can better reproduce traffic oscillations and memory effects than static formulations, while more recent CNN–LSTM structures further improve feature extraction and generalization across different driving scenarios [11, 18, 32]. From the perception side, machine-vision-based sensors are among the earliest and most widely used sensing modalities for vehicle detection in autonomous driving, and modern deep-learning detectors provide an effective basis for extracting target-vehicle information from image data [6]. This makes monocular vision a promising solution for single-camera car-following systems, particularly when the uncertainty of visual measurements is explicitly handled at the control stage.

Motivated by the above considerations, this paper proposes a vision-based car-following system that integrates monocular vehicle perception with an adaptive speed controller based on a multilayer type-2 fuzzy neural network and recurrent neural learning. A convolutional neural network is employed to detect the preceding vehicle and extract spatial information from image sequences captured by a monocular camera. The estimated inter-vehicle distance is then processed by a Kalman filter to suppress noise and improve the stability of the feedback signal. Based on this perception output, the proposed controller regulates the host vehicle speed to maintain a safe following distance and to adapt to changes in the motion of the leading vehicle. By combining the uncertainty-handling capability of interval type-2 fuzzy logic with the temporal learning ability of recurrent neural structures, the proposed framework is expected to achieve both stable tracking performance and responsive longitudinal control.

The main contributions of this work are summarized as follows. First, a low-cost monocular-vision-based perception scheme is developed for preceding-vehicle detection and distance estimation. Second, a Kalman-filter-based processing stage is introduced to enhance the robustness and stability of the distance feedback used for control. Third, a multilayer type-2 fuzzy neural speed controller with recurrent neural learning is designed to address nonlinear longitudinal dynamics, perceptual uncertainty, and temporal dependence in car-following behavior simultaneously. Finally, the effectiveness of the proposed system is validated through both simulation and real-time experiments, demonstrating stable following behavior, rapid response to speed variations, and safe operation under different driving conditions. Overall, the study provides preliminary evidence of the feasibility and practical applicability of the proposed vision-based car-following approach under controlled simulations and small-scale experimental conditions.

2 Problem formulation

This section formulates the mathematical model of the vehicle-following problem and defines the control objective. Figure 1 illustrates the vehicle platoon structure, where Vehicle 0 acts as the leading vehicle, and the following vehicles maintain a desired inter-vehicle distance d_i based on their positions p_i. Consider a platoon consisting of N vehicles moving along the same lane, where each follower vehicle aims to maintain a safe distance from its immediate predecessor.

2.1 Vehicle platoon kinematics

Let $p_i(t)$ denote the longitudinal position of the i-th vehicle and $v_i(t) = \dot{p}_i(t)$ its velocity. The inter-vehicle spacing between vehicle i and its preceding vehicle $(i-1)$ is defined as

Figure 1: Structure of the car-following system showing the leading vehicle (Vehicle 0), following vehicles, inter-vehicle distance d_i, and vehicle position p_i.

$$d_i(t) = p_{i-1}(t) - p_i(t). \tag{1}$$

To ensure safe operation, the desired spacing is assumed to follow a velocity-dependent policy of the form

$$d_i^*(t) = h_i v_{i-1}(t) + D_i, \tag{2}$$

where $h_i \geq 0$ is the spacing-policy coefficient and $D_i > 0$ is the standstill spacing. The desired spacing is generated from the motion of the preceding vehicle and is therefore treated as an exogenous reference with respect to the control input of Vehicle i. The case $h_i = 0$ corresponds to a constant-spacing policy.

The spacing error for the i-th vehicle is then defined as

$$e_i(t) = d_i(t) - d_i^*(t). \tag{3}$$

Substituting the definitions yields

$$e_i(t) = p_{i-1}(t) - p_i(t) - h_i v_{i-1}(t) - D_i. \tag{4}$$

The control objective is to design an input signal such that

$$\lim_{t\to\infty} e_i(t) = 0, \tag{5}$$

which guarantees that the follower vehicle maintains the desired safe distance from its leader.

2.2 Longitudinal vehicle dynamics

The longitudinal dynamics of each vehicle are described by a simplified nonlinear model incorporating drivetrain lag and external resistive forces. Let $F_i(t)$ denote the traction/braking force generated by the powertrain and $u_i(t)$ be the control input.

The actuator dynamics can be approximated by a first-order system

$$\dot{F}_i(t) = \frac{1}{\tau_i}\left(-F_i(t) + u_i(t)\right), \tag{6}$$

where $\tau_i > 0$ is the actuator time constant.

The longitudinal motion of the vehicle is governed by Newton's second law

$$\dot{v}_i(t) = \frac{1}{m_i}\left(F_i(t) - R_i(v_i) - f_i\right), \tag{7}$$

where m_i is the vehicle mass, $R_i(v_i)$ represents aerodynamic drag, and f_i denotes rolling resistance and other disturbances.

The aerodynamic drag is modeled as

$$R_i(v_i) = c_i v_i^2, \tag{8}$$

where c_i is a drag coefficient.

Differentiating the longitudinal dynamics in Eq. (7) gives

$$\ddot{v}_i(t) = \frac{1}{m_i}\left[\dot{F}_i(t) - 2c_i v_i(t)\dot{v}_i(t) - \dot{f}_i(t)\right]. \tag{9}$$

From Eq. (7), the traction force can be expressed as

$$F_i(t) = m_i\dot{v}_i(t) + c_i v_i^2(t) + f_i(t). \tag{10}$$

Substituting Eq. (6) and Eq. (10) into Eq. (9) yields

$$\ddot{v}_i(t) = -\left(\frac{1}{\tau_i} + \frac{2c_i}{m_i}v_i(t)\right)\dot{v}_i(t) - \frac{c_i}{m_i\tau_i}v_i^2(t) - \frac{f_i(t)}{m_i\tau_i} - \frac{\dot{f}_i(t)}{m_i} + \frac{1}{m_i\tau_i}u_i(t). \tag{11}$$

Since $v_i(t) = \dot{p}_i(t)$, Eq. (11) can be written as

$$\dddot{p}_i(t) = \phi_i\big(v_i, \dot{v}_i, t\big) + \psi_i u_i(t), \tag{12}$$

where

$$\phi_i\big(v_i, \dot{v}_i, t\big) = -\left(\frac{1}{\tau_i} + \frac{2c_i}{m_i}v_i\right)\dot{v}_i - \frac{c_i}{m_i\tau_i}v_i^2 - \frac{f_i}{m_i\tau_i} - \frac{\dot{f}_i}{m_i}, \tag{13}$$

and

$$\psi_i = \frac{1}{m_i\tau_i} > 0. \tag{14}$$

The disturbance $f_i(t)$ is assumed to be continuously differentiable with bounded $f_i(t)$ and $\dot{f}_i(t)$. If the rolling resistance is modeled as a constant, then $\dot{f}_i(t) = 0$.

2.3 Tracking error representation

For controller design, define the error state vector as

$$\mathbf{z}_i = \begin{bmatrix} e_i & \dot{e}_i & \ddot{e}_i \end{bmatrix}^\top. \tag{15}$$

The goal of the control system is to regulate $\mathbf{z}_i$ to the origin, ensuring convergence of the spacing error and its derivatives.

If the nonlinear dynamics $\phi_i(\cdot)$ and $\psi_i(\cdot)$ were exactly known, an ideal control input could be constructed to yield the linear error dynamics

$$\dddot{e}_i + a_1\ddot{e}_i + a_2\dot{e}_i + a_3 e_i = 0, \tag{16}$$

where a_1, a_2, and a_3 are positive constants chosen such that the characteristic polynomial

$$s^3 + a_1 s^2 + a_2 s + a_3 = 0, \tag{17}$$

is Hurwitz.

However, in practical applications the vehicle parameters and nonlinear functions are typically uncertain or time-varying. Therefore, a data-driven or adaptive control strategy, is required to approximate the unknown dynamics and achieve stable car-following behavior.

Remark 2.1 (Relationship between the Platoon Formulation and the Single-Follower Controller)**.** *To clarify the transition from the general platoon formulation to the controller developed in Section 3, consider the two-vehicle case ($N = 2$), where Vehicle 0 is the leader and Vehicle 1 is the controlled follower. The leader follows the longitudinal trajectory $p_0(t)$, with*

$$v_0(t) = \dot{p}_0(t), \qquad \dot{v}_0(t) = \ddot{p}_0(t).$$

The leader-motion variables are treated as bounded exogenous signals, whereas the control input $u_1(t)$ is applied only to Vehicle 1.

For Vehicle 1, the actual spacing, desired spacing, and tracking error are respectively given by

$$
\begin{aligned}
d_1(t) &= p_0(t) - p_1(t), \\
d_1^*(t) &= h_1 v_0(t) + D_1, \\
e_1(t) &= p_0(t) - p_1(t) - h_1 v_0(t) - D_1.
\end{aligned} \tag{18}
$$

Since h_1 and D_1 are constant, differentiating the spacing error gives

$$
\begin{aligned}
\dot{e}_1(t) &= v_0(t) - v_1(t) - h_1 \dot{v}_0(t), \\
\ddot{e}_1(t) &= \dot{v}_0(t) - \dot{v}_1(t) - h_1 \ddot{v}_0(t), \\
\dddot{e}_1(t) &= \ddot{v}_0(t) - \ddot{v}_1(t) - h_1 \dddot{v}_0(t).
\end{aligned} \tag{19}
$$

Accordingly, the tracking-error vector of the controlled follower is constructed as

$$
\boldsymbol{\chi}_1(t) = \begin{bmatrix} e_1(t) & \dot{e}_1(t) & \ddot{e}_1(t) \end{bmatrix}^\top . \tag{20}
$$

The same formulation applies to an arbitrary follower Vehicle i and its immediate predecessor Vehicle $i-1$. Because the controller design in Section 3 is presented for a representative follower, the vehicle index is omitted for notational simplicity. The error-state components are therefore denoted by

$$
z_1 = e, \qquad z_2 = \dot{e}, \qquad z_3 = \ddot{e}, \tag{21}
$$

and the controller input vector becomes

$$
\boldsymbol{\chi}(t) = \begin{bmatrix} z_1(t) & z_2(t) & z_3(t) \end{bmatrix}^\top . \tag{22}
$$

Here, the subscripts in z_1, z_2, z_3 denote the components of the single-follower tracking-error state rather than vehicle indices. By differentiating the filtered error and substituting the follower dynamics in Eq. (9), the leader-motion terms, spacing-policy terms, and uncertain follower dynamics are collected in $\Xi(\boldsymbol{\chi}, t)$, while $u_1(t)$ is denoted by $u(t)$. Substituting the follower dynamics (12) *into the third derivative of the spacing error leads to the compact filtered-error dynamics derived explicitly in Eq.* (37).

3 Proposed recurrent multilayer interval type-2 fuzzy neural network controller

This section develops a recurrent multilayer interval type-2 fuzzy neural network (RMIT2FNN) controller for the car-following problem formulated in Section 2. The controller is designed to approximate the ideal nonlinear control action in real time while preserving robustness against modeling uncertainty, approximation residuals, and measurement disturbances. Figure 2 resents the architecture of the proposed recurrent multilayer interval type-2 fuzzy neural network (RMIT2FNN) controller. The proposed controller is composed of seven layers, namely the input layer, context layer, interval type-2 fuzzy membership function layer, rule layer, type-reduction layer, weight layer, and output-feedback layer. The context layer introduces recurrent connections to incorporate temporal information, allowing the controller to capture system dynamics effectively. The interval type-2 fuzzy sets improve robustness against uncertainties, while the Cartesian-product rule construction enhances the mapping capability between inputs and outputs. The final control signal is obtained through a weighted summation of type-reduced rule outputs and a feedback structure, which ensures stable and adaptive control performance in nonlinear car-following systems.

To this end, the final control input is composed of a neural approximation term and a robust compensator,

$$
u(t) = u_n(t) + u_r(t), \tag{23}
$$

where $u_n(t)$ is generated by the proposed RMIT2FNN and $u_r(t)$ is a stabilizing compensation term.

3.1 Filtered error dynamics

For notational simplicity, the vehicle index is omitted in this section. Let the spacing error states be

$$
z_1 = e, \qquad z_2 = \dot{e}, \qquad z_3 = \ddot{e}. \tag{24}
$$

Define the filtered error (or sliding variable) as

$$
s = z_3 + \lambda_2 z_2 + \lambda_1 z_1, \tag{25}
$$

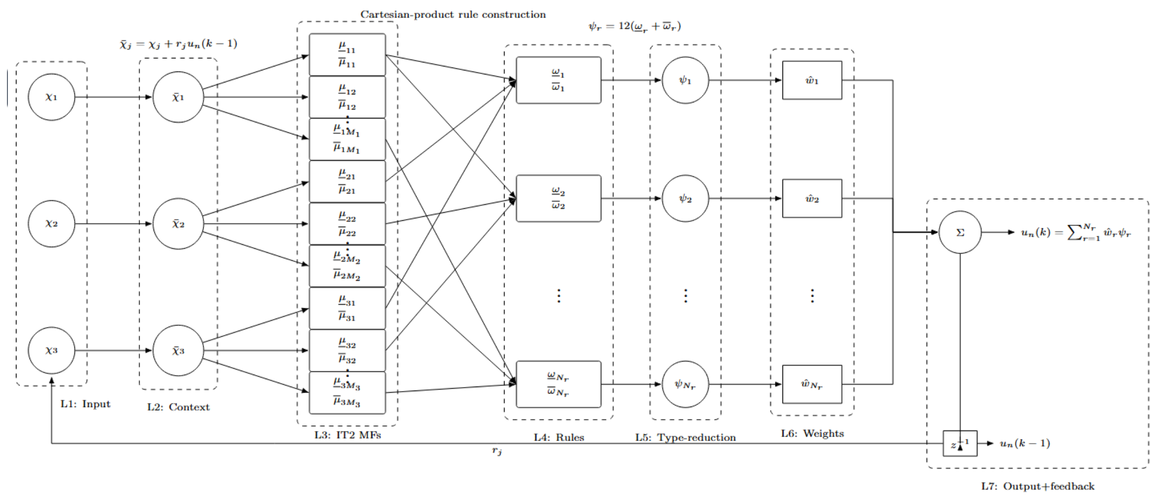

Figure 2: Architecture of the proposed RMIT2FNN controller.

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are chosen such that the polynomial

$$\varpi(\nu) = \nu^2 + \lambda_2 \nu + \lambda_1, \tag{26}$$

is Hurwitz. The variable s summarizes the tracking objective in a scalar form and will be used for both adaptation and stability analysis.

To derive the filtered-error dynamics explicitly, first retain the vehicle index. From Eqs. (2) and (4), the first three derivatives of the spacing error are

$$\dot{e}_i = v_{i-1} - v_i - h_i \dot{v}_{i-1}, \tag{27}$$
$$\ddot{e}_i = \dot{v}_{i-1} - \dot{v}_i - h_i \ddot{v}_{i-1}, \tag{28}$$
$$\dddot{e}_i = \ddot{v}_{i-1} - \ddot{v}_i - h_i \dddot{v}_{i-1}. \tag{29}$$

Substituting the follower dynamics (12), namely,

$$\ddot{v}_i = \phi_i(v_i, \dot{v}_i, t) + \psi_i u_i,$$

into Eq. (29) gives

$$\dddot{e}_i = \ddot{v}_{i-1} - h_i \dddot{v}_{i-1} - \phi_i(v_i, \dot{v}_i, t) - \psi_i u_i. \tag{30}$$

For the filtered error

$$s_i = \ddot{e}_i + \lambda_2 \dot{e}_i + \lambda_1 e_i, \tag{31}$$

its derivative is

$$\dot{s}_i = \dddot{e}_i + \lambda_2 \ddot{e}_i + \lambda_1 \dot{e}_i = \Xi_i(\boldsymbol{\chi}_i, t) - b_i(\boldsymbol{\chi}_i, t) u_i, \tag{32}$$

where

$$\Xi_i(\boldsymbol{\chi}_i, t) = \ddot{v}_{i-1} - h_i \dddot{v}_{i-1} - \phi_i(v_i, \dot{v}_i, t) + \lambda_2 \left(\dot{v}_{i-1} - \dot{v}_i - h_i \ddot{v}_{i-1}\right) + \lambda_1 \left(v_{i-1} - v_i - h_i \dot{v}_{i-1}\right), \tag{33}$$

and

$$b_i(\boldsymbol{\chi}_i, t) = \psi_i = \frac{1}{m_i \tau_i} > 0. \tag{34}$$

Notice that the follower velocity and acceleration can be expressed as

$$v_i = v_{i-1} - h_i \dot{v}_{i-1} - \dot{e}_i, \tag{35}$$
$$\dot{v}_i = \dot{v}_{i-1} - h_i \ddot{v}_{i-1} - \ddot{e}_i. \tag{36}$$

Therefore, for bounded predecessor-motion signals, Ξ_i can be represented as a smooth function of $\boldsymbol{\chi}_i = [e_i, \dot{e}_i, \ddot{e}_i]^\top$ and time.

The predecessor trajectory is assumed to be sufficiently smooth, with $p_{i-1}(t) \in C^4$, and the signals v_{i-1}, $\dot{v}_{i-1}$, $\ddot{v}_{i-1}$, and $\dddot{v}_{i-1}$ are assumed to be bounded. These signals are treated as bounded exogenous inputs to the controlled follower.

For notational simplicity, the vehicle index is omitted in the remainder of the controller design. Accordingly, Eq. (32) is written as

$$\dot{s} = \Xi(\boldsymbol{\chi}, t) - b(\boldsymbol{\chi}, t)u. \tag{37}$$

If the nonlinear functions in (37) were exactly known, the ideal control law could be defined as

$$u^\star(\boldsymbol{\chi}, t) = b^{-1}(\boldsymbol{\chi}, t)\big[\Xi(\boldsymbol{\chi}, t) + k_s s\big], \tag{38}$$

with $k_s > 0$, yielding

$$\dot{s} = -k_s s. \tag{39}$$

Since $u^\star(\boldsymbol{\chi}, t)$ is unavailable in practice, the proposed RMIT2FNN approximates it.

3.2 RMIT2FNN architecture

The proposed RMIT2FNN consists of seven computational layers: the input layer, recurrent context layer, interval type-2 membership layer, rule-firing layer, type-reduction layer, consequent-weight layer, and output-feedback layer. Their functions are described below.

Layer 1: Input Layer

The input vector is defined as

$$\boldsymbol{\chi}(t) = \begin{bmatrix} z_1(t) & z_2(t) & z_3(t) \end{bmatrix}^\top \in \mathbb{R}^3. \tag{40}$$

Layer 2: Recurrent Context Layer:

To embed temporal memory, each input channel is augmented by a recurrent term,

$$\bar{\chi}_j(t) = \chi_j(t) + r_j u_n(t - \Delta t), \qquad j = 1, 2, 3. \tag{41}$$

where r_j is the recurrent coefficient associated with the jth channel, $u_n(t - \Delta t)$ denotes the previous network output, and Δt is the sampling interval. The recurrent feedback enables the network to capture short-term temporal dependencies that frequently arise in vehicle-following maneuvers.

Layer 3: Interval Type-2 Membership Layer

For each input $\bar{\chi}_j$, let $\mathcal{F}_{j\ell}$, $\ell = 1, \dots, M_j$, denote the ℓth interval type-2 Gaussian antecedent set. The lower and upper membership grades are defined as

$$\underline{\mu}_{j\ell}(\bar{\chi}_j) = \exp\left[-\frac{(\bar{\chi}_j - c_{j\ell})^2}{\underline{\sigma}_{j\ell}^2}\right], \tag{42}$$

$$\underline{\mu}_{j\ell}(\bar{\chi}_j) = \exp\left[-\frac{(\bar{\chi}_j - c_{j\ell})^2}{\underline{\sigma}_{j\ell}^2}\right], \tag{43}$$

where $c_{j\ell}$ is the center, and $0 < \underline{\sigma}_{j\ell} \le \overline{\sigma}_{j\ell}$ are the lower and upper widths, respectively. The footprint of uncertainty is therefore induced by the interval $[\underline{\sigma}_{j\ell}, \overline{\sigma}_{j\ell}]$.

Layer 4: Rule Firing Layer

Assume that the rth fuzzy rule is formed by one antecedent from each input channel. Its lower and upper firing strengths are given by

$$\underline{\omega}_r = \prod_{j=1}^{3} \underline{\mu}_{j\ell_j(r)}, \tag{44}$$

$$\overline{\omega}_r = \prod_{j=1}^{3} \overline{\mu}_{j\ell_j(r)}, \tag{45}$$

where $\ell_j(r)$ denotes the antecedent index selected by rule r for input j.

Layer 5: Type-Reduction Layer

For each rule r, the interval type-2 inference process produces an interval-valued firing strength

$$[\underline{\omega}_r, \overline{\omega}_r].$$

The type-reduction layer maps this interval into the scalar basis function

$$\psi_r = \frac{1}{2}\left(\underline{\omega}_r + \overline{\omega}_r\right), \qquad r = 1, \ldots, N_r. \tag{46}$$

where $N_r = \prod_{j=1}^{3} M_j$ denoting the total number of fuzzy rules.

Thus, ψ_r represents the midpoint of the lower and upper rule-firing strengths. This non-iterative type-reduction operation provides a crisp rule activation for the subsequent weighted aggregation while retaining the effect of the footprint of uncertainty introduced by the interval type-2 membership functions. Compared with iterative type-reduction procedures, the adopted midpoint operation has a lower computational burden and is therefore suitable for real-time implementation.

Collecting the type-reduced basis functions gives

$$\boldsymbol{\psi} = \begin{bmatrix} \psi_1 & \psi_2 & \cdots & \psi_{N_r} \end{bmatrix}^\top. \tag{47}$$

Layer 6: Consequent-Weight Layer

Let $\hat{\boldsymbol{w}} \in \mathbb{R}^{N_r}$ denote the adjustable consequent-weight vector. The weighted output of the r-th rule is

$$\phi_r = \hat{w}_r \psi_r. \tag{48}$$

Layer 7: Output-Feedback Layer

The neural control signal is obtained by aggregating the weighted outputs of all rules:

$$u_n(t) = \sum_{r=1}^{N_r} \hat{w}_r \psi_r = \hat{\boldsymbol{w}}^\top \boldsymbol{\psi}. \tag{49}$$

The delayed network output $u_n(t - \Delta t)$ is fed back to the recurrent context layer and contributes to the recurrent inputs defined in Eq. (41). This feedback mechanism supplies short-term temporal information without introducing an additional external input.

3.2.1 Numerical Illustration of the RMIT2FNN Computation

To illustrate the computational procedure in Eqs. (41)–(49), consider three recurrent inputs

$$\bar{\boldsymbol{\chi}} = \begin{bmatrix} 0.2 & -0.1 & 0.3 \end{bmatrix}^\top,$$

with two interval type-2 membership functions assigned to each input. The Cartesian-product rule construction therefore produces $N_r = 2^3 = 8$ rules. Suppose that Eqs. (42) and (43) yield the following lower and upper membership intervals:

$$\begin{aligned}
[\underline{\mu}_{11}, \overline{\mu}_{11}] &= [0.4650, 0.6126], \quad [\underline{\mu}_{12}, \overline{\mu}_{12}] = [0.8688, 0.9139], \\
[\underline{\mu}_{21}, \overline{\mu}_{21}] &= [0.7788, 0.8521], \quad [\underline{\mu}_{22}, \overline{\mu}_{22}] = [0.5698, 0.6977], \\
[\underline{\mu}_{31}, \overline{\mu}_{31}] &= [0.3679, 0.5273], \quad [\underline{\mu}_{32}, \overline{\mu}_{32}] = [0.9394, 0.9608].
\end{aligned}$$

For example, Rule 1 corresponds to the antecedent combination (F_{11}, F_{21}, F_{31}). Its firing-strength interval and type-reduced basis value are

$$[\underline{\omega}_1, \overline{\omega}_1] = [0.1332, 0.2753], \qquad \psi_1 = 0.2043.$$

The results obtained for all eight rules are summarized in Table 1.

Using the illustrative consequent-weight vector

$$\hat{\boldsymbol{w}} = \begin{bmatrix} 0.4 & -0.2 & 0.5 & 0.1 & -0.3 & 0.6 & -0.1 & 0.2 \end{bmatrix}^\top,$$

the final neural control signal calculated from Eq. (49) is

$$u_n = \hat{\boldsymbol{w}}^\top \boldsymbol{\psi} = 0.5093.$$

This example shows how the interval membership grades are propagated through the rule-firing and type-reduction stages before being combined with the consequent weights to generate u_n. The values used above are provided solely for illustration.

Table 1: Illustrative computation of the eight fuzzy rules.

Rule	(ℓ_1, ℓ_2, ℓ_3)	$\underline{\omega}_r$	$\overline{\omega}_r$	ψ_r
1	$(1, 1, 1)$	0.1332	0.2753	0.2043
2	$(1, 1, 2)$	0.3402	0.5016	0.4209
3	$(1, 2, 1)$	0.0975	0.2254	0.1614
4	$(1, 2, 2)$	0.2489	0.4107	0.3298
5	$(2, 1, 1)$	0.2489	0.4107	0.3298
6	$(2, 1, 2)$	0.6356	0.7483	0.6920
7	$(2, 2, 1)$	0.1821	0.3362	0.2592
8	$(2, 2, 2)$	0.4650	0.6126	0.5388

3.3 Robust compensator

Since the neural approximator cannot perfectly reconstruct $u^\star$, an additional robust term is introduced:

$$u_r = k_r s + \hat{\rho}\,\mathrm{sat}\left(\frac{s}{\mu}\right). \tag{50}$$

where $k_r > 0$, $\mu > 0$ is a small boundary-layer constant, and $\hat{\rho}$ is an adaptive estimate of the lumped uncertainty bound. The saturation function is defined as

$$\mathrm{sat}(\varsigma) = \begin{cases} \mathrm{sgn}(\varsigma), & |\varsigma| > 1, \\ \varsigma, & |\varsigma| \le 1. \end{cases} \tag{51}$$

Compared with a discontinuous sign function, (50) alleviates high-frequency chattering while preserving robustness.

3.4 Approximation model and parameterization

Let the complete parameter vector of the RMIT2FNN be

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{\boldsymbol{w}}^\top & \hat{\boldsymbol{c}}^\top & \hat{\underline{\boldsymbol{\sigma}}}^\top & \hat{\overline{\boldsymbol{\sigma}}}^\top & \hat{\boldsymbol{r}}^\top \end{bmatrix}^\top. \tag{52}$$

where $\hat{\boldsymbol{c}}$, $\hat{\underline{\boldsymbol{\sigma}}}$, $\hat{\overline{\boldsymbol{\sigma}}}$, and $\hat{\boldsymbol{r}}$ collect the centers, lower widths, upper widths, and recurrent coefficients, respectively.

It is assumed that there exists an ideal bounded parameter vector $\boldsymbol{\theta}^\star$ such that

$$u^\star = u_n(\boldsymbol{\chi}, \boldsymbol{\theta}^\star) + \varepsilon, \qquad |\varepsilon| \le \rho^\star, \tag{53}$$

for some unknown constant $\rho^\star > 0$. Using a first-order local expansion around the current estimate $\hat{\boldsymbol{\theta}}$, one obtains

$$u^\star - u_n(\hat{\boldsymbol{\theta}}) = \tilde{\boldsymbol{\theta}}^\top \boldsymbol{g} + \varepsilon_c, \tag{54}$$

where $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}$, $\boldsymbol{g} = \partial u_n / \partial \hat{\boldsymbol{\theta}}$ is the network Jacobian, and ε_c is the residual of the local linearization, bounded by

$$|\varepsilon_c| \le \bar{\rho}. \tag{55}$$

The constant $\bar{\rho}$ absorbs both the approximation error and the higher-order Taylor remainder.

The Jacobian vector is partitioned as

$$\boldsymbol{g} = \begin{bmatrix} \boldsymbol{g}_w^\top & \boldsymbol{g}_c^\top & \boldsymbol{g}_{\underline{\sigma}}^\top & \boldsymbol{g}_{\overline{\sigma}}^\top & \boldsymbol{g}_r^\top \end{bmatrix}^\top, \tag{56}$$

where

$$\boldsymbol{g}_w = \frac{\partial u_n}{\partial \hat{\boldsymbol{w}}} = \boldsymbol{\psi}, \tag{57}$$

$$\boldsymbol{g}_c = \frac{\partial u_n}{\partial \hat{\boldsymbol{c}}}, \qquad \boldsymbol{g}_{\underline{\sigma}} = \frac{\partial u_n}{\partial \hat{\underline{\boldsymbol{\sigma}}}}, \qquad \boldsymbol{g}_{\overline{\sigma}} = \frac{\partial u_n}{\partial \hat{\overline{\boldsymbol{\sigma}}}}, \qquad \boldsymbol{g}_r = \frac{\partial u_n}{\partial \hat{\boldsymbol{r}}}. \tag{58}$$

For a rule r containing the antecedent $\mathcal{F}_{j\ell}$, the elementary derivatives used in (56) are

$$\frac{\partial \underline{\omega}_r}{\partial c_{j\ell}} = \underline{\omega}_r \frac{2(\bar{\chi}_j - c_{j\ell})}{\underline{\sigma}_{j\ell}^2}. \tag{59}$$

$$\frac{\partial \overline{\omega}_r}{\partial c_{j\ell}} = \overline{\omega}_r \frac{2(\bar{\chi}_j - c_{j\ell})}{\overline{\sigma}_{j\ell}^2}. \tag{60}$$

$$\frac{\partial \underline{\omega}_r}{\partial \underline{\sigma}_{j\ell}} = \underline{\omega}_r \frac{2(\bar{\chi}_j - c_{j\ell})^2}{\underline{\sigma}_{j\ell}^3}. \tag{61}$$

$$\frac{\partial \overline{\omega}_r}{\partial \overline{\sigma}_{j\ell}} = \overline{\omega}_r \frac{2(\bar{\chi}_j - c_{j\ell})^2}{\overline{\sigma}_{j\ell}^3}. \tag{62}$$

$$\frac{\partial \underline{\omega}_r}{\partial r_j} = -\underline{\omega}_r \frac{2(\bar{\chi}_j - c_{j\ell})}{\underline{\sigma}_{j\ell}^2} u_n(t - \Delta t). \tag{63}$$

$$\frac{\partial \overline{\omega}_r}{\partial r_j} = -\overline{\omega}_r \frac{2(\bar{\chi}_j - c_{j\ell})}{\overline{\sigma}_{j\ell}^2} u_n(t - \Delta t). \tag{64}$$

and therefore

$$\frac{\partial \psi_r}{\partial(\cdot)} = \frac{1}{2}\left(\frac{\partial \underline{\omega}_r}{\partial(\cdot)} + \frac{\partial \overline{\omega}_r}{\partial(\cdot)}\right). \tag{65}$$

3.5 Adaptive laws

Using (54), the adaptive laws are selected in the compact form

$$\dot{\hat{\boldsymbol{\theta}}} = \Gamma_\theta \, \boldsymbol{g} \, s, \tag{66}$$

where $\Gamma_\theta = \mathrm{diag}\{\Gamma_w, \Gamma_c, \Gamma_{\underline{\sigma}}, \Gamma_{\overline{\sigma}}, \Gamma_r\}$ is a block-diagonal positive definite gain matrix. Equation (66) can be written elementwise as

$$\dot{\hat{\boldsymbol{w}}} = \Gamma_w \boldsymbol{\psi} \, s, \tag{67}$$

$$\dot{\hat{\boldsymbol{c}}} = \Gamma_c \boldsymbol{g}_c \, s, \tag{68}$$

$$\dot{\hat{\underline{\boldsymbol{\sigma}}}} = \Gamma_{\underline{\sigma}} \boldsymbol{g}_{\underline{\sigma}} \, s, \tag{69}$$

$$\dot{\hat{\overline{\boldsymbol{\sigma}}}} = \Gamma_{\overline{\sigma}} \boldsymbol{g}_{\overline{\sigma}} \, s, \tag{70}$$

$$\dot{\hat{\boldsymbol{r}}} = \Gamma_r \boldsymbol{g}_r \, s. \tag{71}$$

To estimate the uncertainty bound in (50), the following law is adopted:

$$\dot{\hat{\rho}} = \mathrm{Proj}_{[0,\rho_{\max}]}\left(\hat{\rho}, \gamma_\rho |s|\right), \qquad \gamma_\rho > 0. \tag{72}$$

For implementation, the premise parameters are constrained to a compact admissible set through a projection operator, i.e.,

$$\dot{\hat{\boldsymbol{\theta}}} = \mathrm{Proj}_\Theta\left(\hat{\boldsymbol{\theta}}, \Gamma_\theta \boldsymbol{g} s\right). \tag{73}$$

which prevents loss of positivity of the interval widths and avoids parameter drift in the presence of persistent disturbances.

3.6 Closed-loop stability analysis

Theorem 3.1. *Consider the error dynamics* (37) *and the control law* (23)*, where u_n is generated by the RMIT2FNN* (49) *and u_r is given by* (50)*. Suppose that the approximation assumption* (53) *holds and that the input gain is positive and bounded as*

$$0 < b_0 \le b(\boldsymbol{\chi}, t) \le b_1, \tag{74}$$

where b_0 and b_1 are known positive constants. Assume that the ideal parameter vector belongs to the compact admissible parameter set and that the network parameters and adaptive uncertainty estimate are updated using the projection-based adaptive laws (73) *and* (72)*. Then all closed-loop signals remain bounded. Moreover, the filtered error s is uniformly ultimately bounded. Its ultimate bound can be reduced by increasing k_s or k_r, or by selecting a smaller boundary-layer parameter μ.*

Proof. Substituting (23) into (37) and using (38), one obtains

$$\dot{s} = -k_s s + b(\boldsymbol{\chi}, t)\left(u^\star - u_n - u_r\right). \tag{75}$$

Applying (54) and the corrected robust compensator (50) yields

$$\dot{s} = -k_s s + b(\boldsymbol{\chi}, t)\left[\tilde{\boldsymbol{\theta}}^\top \boldsymbol{g} + \varepsilon_c - k_r s - \hat{\rho}\,\mathrm{sat}\left(\frac{s}{\mu}\right)\right]. \tag{76}$$

Define

$$\vartheta(t) = \frac{b(\boldsymbol{\chi}, t)}{b_0}. \tag{77}$$

From (74), it follows that

$$1 \le \vartheta(t) \le \vartheta_{\max}, \qquad \vartheta_{\max} = \frac{b_1}{b_0}. \tag{78}$$

The projection operator constrains the estimated network parameters to a compact admissible set and maintains all interval widths above a strictly positive lower bound. Since the ideal parameter vector is also bounded, $\tilde{\boldsymbol{\theta}}$ remains bounded. Moreover, the Gaussian basis functions and their parameter derivatives are bounded under these constraints. The previous network output is also bounded because $u_n = \hat{\boldsymbol{w}}^\top \boldsymbol{\psi}$, where both $\hat{\boldsymbol{w}}$ and $\boldsymbol{\psi}$ are bounded. Consequently, the network Jacobian $\boldsymbol{g}$ is bounded.

Therefore, the combined residual term

$$\Delta(t) = \vartheta(t)\varepsilon_c + \left[\vartheta(t) - 1\right]\tilde{\boldsymbol{\theta}}^\top \boldsymbol{g}, \tag{79}$$

is bounded. Thus, there exists a positive constant $\bar{\Delta}$ such that

$$|\Delta(t)| \le \bar{\Delta}. \tag{80}$$

Let

$$\tilde{\rho} = \bar{\Delta} - \hat{\rho}, \tag{81}$$

where $\rho_{\max} > \bar{\Delta}$, and consider the Lyapunov candidate

$$V = \frac{1}{2b_0}s^2 + \frac{1}{2}\tilde{\boldsymbol{\theta}}^\top \Gamma_\theta^{-1}\tilde{\boldsymbol{\theta}} + \frac{1}{2\gamma_\rho}\tilde{\rho}^2. \tag{82}$$

Differentiating (82) along (76) gives

$$\dot{V} = \frac{s}{b_0}\dot{s} - \tilde{\boldsymbol{\theta}}^\top \Gamma_\theta^{-1}\dot{\hat{\boldsymbol{\theta}}} - \frac{1}{\gamma_\rho}\tilde{\rho}\dot{\hat{\rho}} = \frac{s}{b_0}\left[-k_s s + b(\boldsymbol{\chi}, t)\left(\tilde{\boldsymbol{\theta}}^\top \boldsymbol{g} + \varepsilon_c - k_r s - \hat{\rho}\,\mathrm{sat}\left(\frac{s}{\mu}\right)\right)\right] - \tilde{\boldsymbol{\theta}}^\top \Gamma_\theta^{-1}\dot{\hat{\boldsymbol{\theta}}} - \frac{1}{\gamma_\rho}\tilde{\rho}\dot{\hat{\rho}}. \tag{83}$$

The projection-based parameter update law satisfies the standard projection property

$$-\tilde{\boldsymbol{\theta}}^\top \Gamma_\theta^{-1}\dot{\hat{\boldsymbol{\theta}}} \le -\tilde{\boldsymbol{\theta}}^\top \boldsymbol{g} s. \tag{84}$$

Therefore,

$$\vartheta\tilde{\boldsymbol{\theta}}^\top \boldsymbol{g} s - \tilde{\boldsymbol{\theta}}^\top \Gamma_\theta^{-1}\dot{\hat{\boldsymbol{\theta}}} \le \left(\vartheta - 1\right)\tilde{\boldsymbol{\theta}}^\top \boldsymbol{g} s. \tag{85}$$

Using (79) and (85), Eq. (83) becomes

$$\dot{V} \le -\left(\frac{k_s}{b_0} + k_r\vartheta\right) s^2 + s\Delta(t) - \vartheta\hat{\rho} s \,\mathrm{sat}\left(\frac{s}{\mu}\right) - \frac{1}{\gamma_\rho}\tilde{\rho}\dot{\hat{\rho}}. \tag{86}$$

Since $k_s > 0$, $k_r > 0$, and $\vartheta(t) \ge 1$, define

$$\kappa_s = \frac{k_s}{b_0} + k_r > 0. \tag{87}$$

Then,

$$\frac{k_s}{b_0} + k_r\vartheta \ge \kappa_s. \tag{88}$$

The projection-based uncertainty adaptation law (72) satisfies

$$-\frac{1}{\gamma_\rho}\tilde{\rho}\dot{\hat{\rho}} \le -\tilde{\rho}|s|. \tag{89}$$

Using (80), (88), and (89) in (86) gives

$$\dot{V} \le -\kappa_s s^2 + \bar{\Delta}|s| - \vartheta\hat{\rho} s \,\mathrm{sat}\left(\frac{s}{\mu}\right) - \left(\bar{\Delta} - \hat{\rho}\right)|s| = -\kappa_s s^2 + \hat{\rho}|s| - \vartheta\hat{\rho} s \,\mathrm{sat}\left(\frac{s}{\mu}\right). \tag{90}$$

Because $\vartheta \ge 1$, $\hat{\rho} \ge 0$, and

$$s \,\mathrm{sat}\left(\frac{s}{\mu}\right) \ge 0, \tag{91}$$

it follows that

$$\dot{V} \le -\kappa_s s^2 + \hat{\rho}\left[|s| - s \,\mathrm{sat}\left(\frac{s}{\mu}\right)\right]. \tag{92}$$

For the saturation function defined in (51), the following inequality holds for all s:

$$0 \le |s| - s \,\mathrm{sat}\left(\frac{s}{\mu}\right) \le \frac{\mu}{4}. \tag{93}$$

Indeed, the expression is zero when $|s| > \mu$, whereas for $|s| \le \mu$,

$$|s| - s \,\mathrm{sat}\left(\frac{s}{\mu}\right) = |s| - \frac{s^2}{\mu} \le \frac{\mu}{4}. \tag{94}$$

Since the projection operator guarantees

$$0 \le \hat{\rho}(t) \le \rho_{\max}, \tag{95}$$

Eq. (92) leads to

$$\dot{V} \le -\kappa_s s^2 + \frac{\rho_{\max}\mu}{4}. \tag{96}$$

Therefore, $\dot{V} < 0$ whenever

$$|s| > \sqrt{\frac{\rho_{\max}\mu}{4\kappa_s}}. \tag{97}$$

Standard uniform-ultimate-boundedness arguments imply that the filtered error enters and remains within the compact set

$$\Omega_s = \left\{ s : |s| \le \sqrt{\frac{\rho_{\max}\mu}{4\kappa_s}} \right\}. \tag{98}$$

The parameter estimates $\hat{\boldsymbol{\theta}}$ and $\hat{\rho}$ remain bounded because of the projection operators. To establish the boundedness of the original tracking-error states, note that

$$s = z_3 + \lambda_2 z_2 + \lambda_1 z_1,$$

can be rewritten as

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\lambda_1 & -\lambda_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} s. \tag{99}$$

Because the polynomial $\varpi(\nu) = \nu^2 + \lambda_2\nu + \lambda_1$ is Hurwitz, the subsystem (99) is input-to-state stable with respect to s. Hence, the boundedness and uniform ultimate boundedness of s imply the boundedness and uniform ultimate boundedness of z_1 and z_2. Moreover,

$$z_3 = s - \lambda_2 z_2 - \lambda_1 z_1, \tag{100}$$

is also bounded and uniformly ultimately bounded.

Consequently, the spacing error and its first two derivatives converge to compact neighborhoods of the origin. All remaining closed-loop signals are bounded because the network parameters, membership grades, rule firing strengths, neural output, robust compensator, and total control input remain bounded. □

3.7 Implementation remarks

In real-time implementation, the proposed controller is updated according to the following sequence:

1. Measure or estimate e, $\dot{e}$, and $\ddot{e}$, then construct $\boldsymbol{\chi}$ and s.
2. Compute the recurrent inputs $\bar{\chi}_j$ using (41).
3. Evaluate the lower and upper membership functions by (42)–(43).
4. Compute the rule activations and basis vector $\boldsymbol{\psi}$.
5. Generate the neural control signal u_n using (49).
6. Update the adaptive parameters according to (67)–(71) and (72).
7. Apply the total control law (23).

The proposed RMIT2FNN controller combines three favorable properties for vision-based car-following applications: (i) interval type-2 fuzzy inference enhances robustness against measurement uncertainty; (ii) the recurrent context layer captures temporal dependencies in leader-follower motion; and (iii) the Lyapunov-based adaptation mechanism guarantees bounded closed-loop behavior while allowing online learning of both antecedent and consequent parameters.

4 Simulation and experimental results

4.1 Simulation results

The proposed RMIT2FNN controller was evaluated against two benchmark schemes, namely a conventional fuzzy neural network (FNN) and an interval type-2 fuzzy neural network (IT2FNN), under the same simulation scenario. The comparison focuses on the inter-vehicle distance, the follower velocity, the control effort, and the tracking error. Figure 3 illustrates the control structure of the proposed car-following system, in which the vehicle position $p_i(t)$ is processed by the controller and update laws to generate the control input for maintaining the desired inter-vehicle distance.

Fig. 4 presents the inter-vehicle distance responses. Although all three controllers eventually drive the spacing toward the desired operating region, their transient characteristics are noticeably different. The conventional FNN exhibits the largest initial overshoot, while the IT2FNN reduces this deviation but still shows a visible transient excursion. By contrast, the proposed RMIT2FNN reaches the desired spacing more rapidly and with a more restrained overshoot, indicating a better transient regulation capability.

A similar trend can be observed in the velocity responses shown in Fig. 5. After the start-up phase, the follower vehicle controlled by RMIT2FNN reproduces the speed variations of the lead vehicle with the smallest deviation. The FNN response displays the most pronounced overshoot in the early stage, whereas the IT2FNN provides an intermediate behavior. These results are consistent with the spacing trajectories discussed above.

The control signals are depicted in Fig. 6. A brief corrective action appears at the beginning of the RMIT2FNN response, after which the control input settles rapidly and remains bounded. The FNN and IT2FNN require smaller initial corrections, but their responses fluctuate for a longer period. This behavior suggests that the recurrent interval

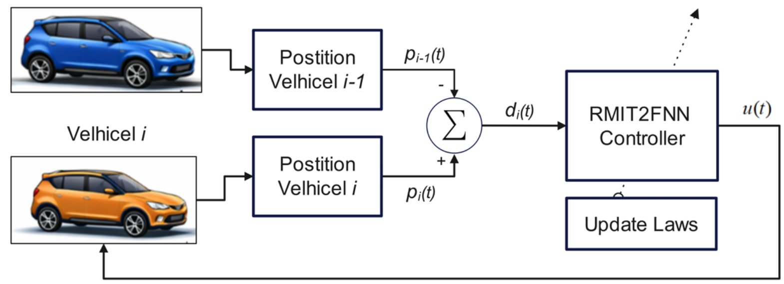

Figure 3: Control structure of the proposed car-following system.

type-2 fuzzy structure can compensate the initial mismatch more decisively and then maintain the closed-loop motion with limited sustained actuation.

The tracking error profiles in Fig. 7, further highlight the difference among the three methods. The FNN yields the largest transient error, whereas the IT2FNN narrows the error band without completely removing the oscillatory behavior for during the transient stage. The proposed RMIT2FNN drives the tracking error to a narrow neighborhood of zero more quickly and preserves the smallest overall error level over the entire simulation interval.

The quantitative comparison reported in Table 2 confirms the visual observations. The proposed RMIT2FNN achieves an RMSE of 0.23043, which is substantially lower than those obtained by IT2FNN (1.5031) and FNN (1.8969). This result indicates that incorporating recurrent learning into the interval type-2 fuzzy neural structure improves the approximation capability for nonlinear and time-varying dynamics, leading to more accurate car-following behavior. In addition to the tracking accuracy, the computation time is also included in Table 2 to provide a balanced comparison between control performance and computational cost.

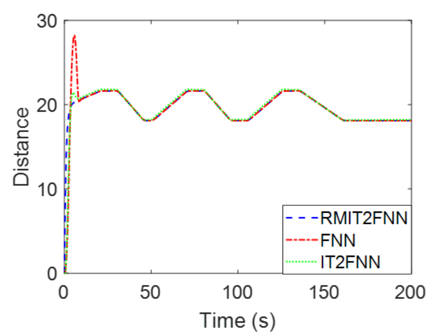

Figure 4: Inter-vehicle distance tracking performance of FNN, IT2FNN, and RMIT2FNN.

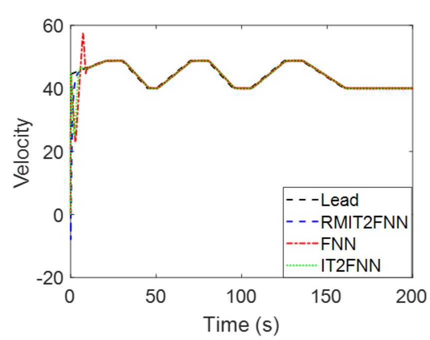

Figure 5: Comparison of velocity responses between the lead vehicle and the follower vehicle under different controllers.

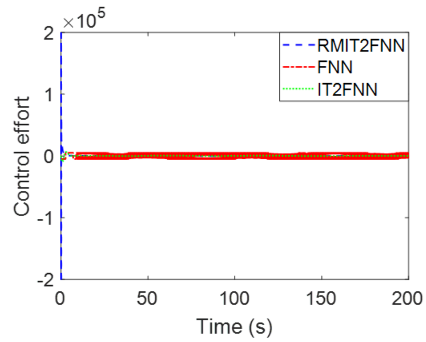

Figure 6: Comparison of control signals generated by FNN, IT2FNN, and RMIT2FNN.

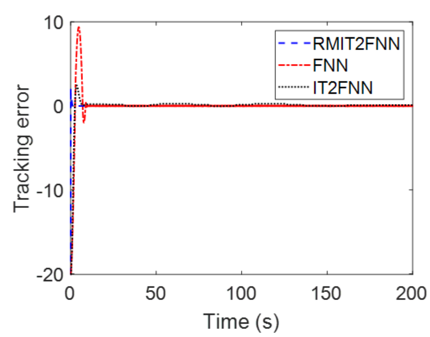

Figure 7: Comparison of tracking error responses among FNN, IT2FNN, and RMIT2FNN.

4.2 Experimental results

To further examine the practical feasibility of the proposed method, the car-following system was implemented on a 1/10-scale autonomous vehicle platform and evaluated in an indoor environment under relatively stable illumination. The experimental setup was designed to reproduce a representative following scenario while preserving repeatability and safe operation.

For distance perception, an ArUco marker, with a side length of 6 cm was mounted on the rear of the leading vehicle. The relative distance was then estimated using the pinhole camera model from monocular images captured by a forward-facing camera installed on the follower vehicle.

The Kalman filter was used only in the experimental perception pipeline to reduce frame-to-frame fluctuations in the monocular distance measurement. It was not included in the RMIT2FNN adaptation mechanism or the closed-loop stability analysis. Let z_k denote the raw camera-based distance measurement and $\hat{d}_k$ the filtered distance. A scalar random-walk model was adopted:

$$\hat{d}_{k|k-1} = \hat{d}_{k-1|k-1}, \tag{101}$$

$$P_{k|k-1} = P_{k-1|k-1} + Q, \tag{102}$$

$$K_k = \frac{P_{k|k-1}}{P_{k|k-1} + R}, \tag{103}$$

$$\hat{d}_{k|k} = \hat{d}_{k|k-1} + K_k \left(z_k - \hat{d}_{k|k-1} \right), \tag{104}$$

$$P_{k|k} = (1 - K_k)\, P_{k|k-1}, \tag{105}$$

where Q and R denote the process-noise and measurement-noise variances, respectively. The filtered value $\hat{d}_{k|k}$ was used as the distance feedback supplied to the controller.

The scalar Kalman filter was initialized using the first camera-based distance measurement, i.e., $\hat{d}_{0|0} = z_0$, with an initial error covariance of $P_{0|0} = 1 \times 10^{-2}\,\mathrm{m}^2$. The process- and measurement-noise variances were selected as $Q = 1 \times 10^{-5}\,\mathrm{m}^2$ and $R = 1 \times 10^{-4}\,\mathrm{m}^2$, respectively. The experimental perception-and-control loop was operated with a sampling interval of approximately $\Delta t = 0.05\,\mathrm{s}$. These values were selected empirically to suppress frame-to-frame distance fluctuations while retaining a sufficiently rapid response to changes in the motion of the leading vehicle.

All perception and control modules were executed in real time on an embedded *Jetson Nano* platform. The processing pipeline consisted of target detection using YOLOv8, ArUco marker recognition, distance estimation, and control signal generation by the proposed RMIT2FNN controller. Fig. 8 shows a representative experimental frame acquired during the vehicle-following test. Fig. 9 presents the real-time inter-vehicle distance response, with the desired distance between the leader and the follower set to 42 cm. The leading vehicle is successfully identified, and the system status is displayed on the interface. The detected marker is highlighted by a red bounding box, whereas the green centroid denotes the visual reference utilized during the lateral alignment process. Based on the estimated inter-vehicle distance and the extracted target position, the controller generates the corresponding speed and steering commands in real time so that the follower vehicle can maintain a safe spacing while remaining aligned with the leader trajectory.

The experimental observations are consistent with the simulation study. In particular, the follower vehicle was able to maintain stable tracking behavior and react promptly to variations in the motion of the leading vehicle. The resulting motion remained smooth throughout the test, indicating that the proposed perception-and-control framework can operate reliably on a small-scale autonomous platform. The real-time experimental results of the proposed system are shown in the video available at: Experimental validation video of the car-following system.

An important advantage of the present implementation is its low-cost sensing configuration and compact embedded architecture, which make it suitable for laboratory-scale autonomous vehicles as well as educational and research applications. At the same time, the current system still depends on favorable visual conditions and on the availability of artificial markers for robust distance estimation. Future work will therefore focus on markerless visual ranging and multimodal sensor fusion to improve robustness in more challenging environments.

Overall, the experimental validation supports the practical applicability of the proposed vision-based car-following framework and demonstrates that the RMIT2FNN-based control strategy can be deployed in real time with satisfactory tracking performance.

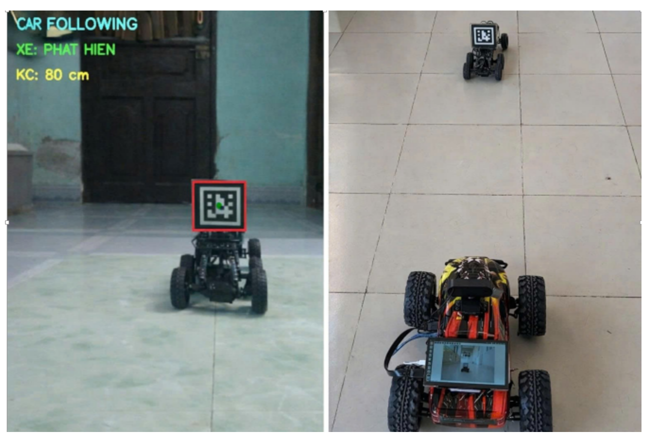

Figure 8: Experimental frame showing ArUco marker detection and real-time inter-vehicle distance estimation in the car-following system.

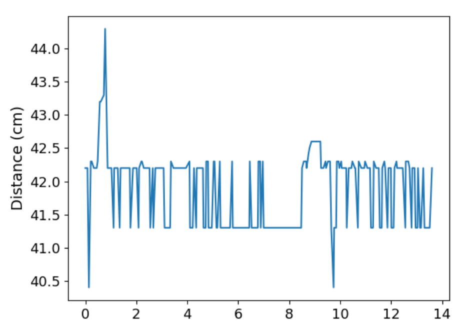

Figure 9: Real-time experimental inter-vehicle distance tracking performance of the car-following system.

Table 2: Quantitative comparison of the simulation results.

Controller	RMSE	Computation time (s)
FNN	1.89690	0.0108
IT2FNN	1.50310	0.0159
RMIT2FNN (proposed controller)	0.23043	0.0182

Although the proposed RMIT2FNN achieved the lowest RMSE among the evaluated controllers, its computation relies on the explicit propagation of lower and upper membership grades, rule firing, midpoint type reduction, and consequent weighting. A complementary research direction is offered by Fuzzy-Informed Neural Networks (FINNs), introduced within the Perception-Informed Neural Network framework [13]. In a FINN, fuzzy IF–THEN knowledge can be incorporated as differentiable constraints in the neural-network learning objective, potentially replacing a separate defuzzification or type-reduction stage. For example, by defining the relative velocity as $\Delta v = v_0 - v_1$, a car-following rule of the form "if the spacing error is negative-large and Δv is negative, then the commanded acceleration should be low or braking" could be represented by a fuzzy-informed loss term that penalizes control outputs violating this safety requirement. Similar constraints could be formulated for safe spacing, acceleration limits, and smooth control actions. Future work will investigate FINN-based and FINN–RNN-based car-following controllers and compare them with the proposed RMIT2FNN in terms of RMSE, transient response, robustness, computational burden, and real-time implementation. Whether such fuzzy-informed constraints can further improve tracking accuracy will require systematic simulation and experimental validation.

5 Conclusion

This paper presented a vision-based car-following system that integrates monocular perception with an adaptive control strategy based on a recurrent multilayer interval type-2 fuzzy neural network (RMIT2FNN). The proposed framework combines a convolutional neural network-based vehicle detection, Kalman-filter-based distance estimation, and an intelligent controller capable of handling nonlinear dynamics, temporal dependencies, and measurement uncertainties.

The RMIT2FNN controller was designed to approximate the ideal control law in real time while ensuring robustness through Lyapunov-based adaptive laws and a compensating mechanism. Theoretical analysis guaranteed the boundedness of all closed-loop signals and the convergence of the tracking error.

Simulation results demonstrated that the proposed method outperforms conventional FNN and IT2FNN controllers in terms of transient response and tracking accuracy, achieving the lowest RMSE among the compared approaches. Furthermore, the experimental study was conducted to validate the real-time applicability and practical feasibility of the proposed system. The results confirmed that the controller can maintain stable inter-vehicle distance, respond effectively to velocity variations of the leading vehicle, and operate smoothly under real-world conditions.

Overall, the proposed system demonstrates the feasibility of a low-cost solution for vision-based car-following under controlled experimental conditions, while further validation using full-scale vehicles and diverse road environments is required before broader application to intelligent transportation and advanced driver assistance systems. Future work will focus on enhancing perception robustness through markerless distance estimation and multi-sensor fusion, as well as extending the framework to more complex and dynamic driving environments.

Acknowledgement

This work was supported by Lac Hong University, Vietnam.

References

[1] H. U. Ahmed, Y. Huang, P. Lu, *A review of car-following models and modeling tools for human and autonomous-ready driving behaviors in micro-simulation*, Smart Cities, **4**(1) (2021), 314-335. https://doi.org/10.3390/smartcities4010019

[2] G. Alite, Å. Cuenca, J. Salt, M. Tomizuka, *Resource-efficient path-following control for a self-driving car in a networked control system*, IEEE Access, **11** (2023), 108011-108023. https://doi.org/10.1109/ACCESS.2023.3321269

[3] S. Askari, S. Zarei, *A fuzzy clustering-based genetic TSK system with fuzzy IF part parameters and nonlinear THEN part*, Iranian Journal of Fuzzy Systems, **22**(6) (2025), 41-65. https://doi.org/10.22111/ijfs.2025.51929.9165

[4] J. Bai, S. Mao, J. J. Lee, *A novel car-following model for adaptive cruise control vehicles using enhanced intelligent driver model*, Transportation Letters, **17**(4) (2025), 702-718. https://doi.org/10.1080/19427867.2024.2376409

[5] Y. Chen, C. Dong, K. Lyu, et al., *A review of car-following and lane-changing models under heterogeneous environments*, Physica A: Statistical Mechanics and its Applications, **654** (2024), 130127. https://doi.org/10.1016/j.physa.2024.130127

[6] W. Cheng, F. Gao, Y. Li, et al., *A machine vision-based dimension measurement method of shaft parts with sub-pixel edge*, Measurement Science and Technology, **36**(5) (2025), 055026. https://doi.org/10.1088/1361-6501/adcc47

[7] Y. Fu, W. Hu, J. Shang, et al., *A fractional-order car-following model with random delay optimized by an enhanced whale optimization algorithm*, Nonlinear Dynamics, **113**(16) (2025), 21277-21303. https://doi.org/10.1007/s11071-025-11271-4

[8] N. M. Hung, H. H. Viet, T. L. Le, *A hybrid parallel decomposed interval type-2 fuzzy-Jordan neural network for controlling uncertain ball-and-beam systems*, Physica Scripta, **101**(7) (2026), 076007. https://doi.org/10.1088/1402-4896/ae442b

[9] M. L. Lagunes, O. Castillo, F. Valdez, J. Soria, *Comparison of fuzzy controller optimization with dynamic parameter adjustment based on of Type-1 and Type-2 fuzzy logic*, Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine, Springer, Cham, (2020), 47-56. https://doi.org/10.1007/978-3-030-34135-0_4

[10] L. T. Le, L. K. Ngo, *Application of combining Type 2 fuzzy CMAC network and Jordan neural network in nonlinear system control*, Iranian Journal of Fuzzy Systems, **22**(2) (2025), 25-39. https://doi.org/10.22111/ijfs.2025.50552.8930

[11] W. Lu, Z. Yi, B. Liang, et al., *Learning car-following behaviors for a connected automated vehicle system: An improved sequence-to-sequence deep learning model*, IEEE Access, **11** (2023), 28076-28089. https://doi.org/10.1109/ACCESS.2023.3243620

[12] H. Ma, J. Qiu, G. Sheng, et al., *Deep simple recurrent unit-based transient modeling method for high-speed circuits*, IEEE Transactions on Microwave Theory and Techniques, **73**(2) (2025), 736-744. https://doi.org/10.1109/TMTT.2024.3426595

[13] M. Mazandarani, M. Najariyan, *Perception-informed neural networks: Beyond physics-informed neural networks*, arXiv preprint, (2025). https://doi.org/10.48550/arXiv.2505.03806

[14] M. D. Pop, *Selection of defuzzification method for calibrating car-following models using Mamdani FIS*, 2025 International Conference on Smart Applications, Communications and Networking (SmartNets), Istanbul, Türkiye, (2025), 1-6. https://doi.org/10.1109/SmartNets65254.2025.11106770

[15] Y. Qin, M. Liu, W. Hao, *Energy-optimal car-following model for connected automated vehicles considering traffic flow stability*, Energy, **298** (2024), 131333. https://doi.org/10.1016/j.energy.2024.131333

[16] B. Qiu, H. Kuang, *A comprehensive review of dynamic modeling methods for extended optimal velocity car-following models in traffic flow*, Nonlinear Dynamics, **113**(10) (2025), 10861-10879. https://doi.org/10.1007/s11071-024-10660-5

[17] V. Shah, H. Patel, *Integrating machine learning and computer vision in advanced driver assistance systems: A comprehensive review*, Augmented Human Research, **11**(1) (2026). https://doi.org/10.1007/s41133-025-00086-y

[18] C. S. Shih, P. W. Huang, E. T. Yen, P. K. Tsung, *Vehicle speed prediction with RNN and attention model under multiple scenarios*, 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, (2019), 369-375. https://doi.org/10.1109/ITSC.2019.8917479

[19] A. Si, S. Das, *MFS: Dynamic decision making approach to select optimal alternative in the presence of uncertainty*, Iranian Journal of Fuzzy Systems, **22**(2) (2025), 81-95. https://doi.org/10.22111/ijfs.2025.49694.8773

[20] V. Siwach, D. Yadav, P. Redhu, *Enhancing driver's attention and overtaking efficiency in car-following model for Advanced Driver Assistance Systems (ADAS) vehicles*, Physica A: Statistical Mechanics and its Applications, **657** (2025), 130207. https://doi.org/10.1016/j.physa.2024.130207

[21] S. Tiwari, A. Gehlot, R. Singh, et al., *Design of an iterative method for disease prediction in finger millet leaves using graph networks, dyna networks, autoencoders, and recurrent neural networks*, Results in Engineering, **24** (2024), 103301. https://doi.org/10.1016/j.rineng.2024.103301

[22] P. V. Toan, T. L. Le, *Parallel hybrid interval type-2 fuzzy NARX network for robust line-following control of autonomous guided vehicles*, Measurement Science and Technology, **37**(10) (2026), 106201. https://doi.org/10.1088/1361-6501/ae4bf8

[23] M. Treiber, A. Kesting, *Car-following models based on driving strategies*, Traffic Flow Dynamics, Springer, Cham, (2025), 239-284. https://doi.org/10.1007/978-3-642-32460-4_11

[24] C. Urrea, C. Domínguez, J. Kern, *Modeling, design and control of a 4-arm delta parallel manipulator employing Type-1 and interval Type-2 fuzzy logic-based techniques for precision applications*, Robotics and Autonomous Systems, **175** (2024), 104661. https://doi.org/10.1016/j.robot.2024.104661

[25] Z. Wang, Y. Shi, W. Tong, et al., *Car-following models for human-driven vehicles and autonomous vehicles: A systematic review*, Journal of Transportation Engineering, Part A: Systems, **149**(8) (2023), 04023075. https://doi.org/10.1061/JTEPBS.TEENG-7836

[26] W. Xiao, G. Xie, H. Liu, et al., *FDAN: Fuzzy deep attention networks for driver behavior recognition*, Journal of Systems Architecture, **147** (2024), 103063. https://doi.org/10.1016/j.sysarc.2023.103063

[27] W. Yang, B. Sun, G. Ma, H. Sun, *Car-following modeling with multi-model driving assistance and stability analysis under heterogeneous time delay*, Chaos, Solitons and Fractals, **204** (2026), 117679. https://doi.org/10.1016/j.chaos.2025.117679

[28] D. Yang, S. Wang, Y. Liao, R. Pan, *An online energy management strategy for fuel cell vehicles based on fuzzy Q-learning and road condition recognition*, IEEE Transactions on Intelligent Transportation Systems, **25**(9) (2024), 12120-12130. https://doi.org/10.1109/TITS.2024.3368149

[29] L. A. Zadeh, *Fuzzy sets*, Information and Control, **8**(3) (1965), 338-353. https://doi.org/10.1016/S0019-9958(65)90241-X

[30] Y. Zha, J. Deng, Y. Qiu, et al., *A survey of intelligent driving vehicle trajectory tracking based on vehicle dynamics*, SAE International Journal of Vehicle Dynamics, Stability, and NVH, **7**(2) (2023), 221-248. https://doi.org/10.4271/10-07-02-0014

[31] T. T. Zhang, P. J. Jin, S. T. McQuade, et al., *Car-following models: A multidisciplinary review*, IEEE Transactions on Intelligent Vehicles, **10**(1) (2025), 92-116. https://doi.org/10.1109/TIV.2024.3409468

[32] Y. Zhou, R. Fu, C. Wang, *Learning the car-following behavior of drivers using maximum entropy deep inverse reinforcement learning*, Journal of Advanced Transportation, **2020** (2020), 4752651. https://doi.org/10.1155/2020/4752651