Mechanical Systems and Signal Processing 238 (2025) 113211

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Full length article

Reinforcement learning-based fusion framework for vehicle sideslip angle estimators using physically guided neural networks

Chaofan Gong [a], Yan Kong [b], Dong Zhang [c,*], Yao Ma [d], Bo Lu [e], Shuyong Xing [e], Changfu Zong [a]

[a] National Key Laboratory of Automotive Chassis Integration and Bionics, Jilin University, Changchun, 130022, China
[b] School of Engineering, University of Surrey, Guildford, GU2 7XH, UK
[c] College of Engineering, Design and Physical Sciences, Brunel University of London, London, UB8 3PH, UK
[d] College of Electromechanical Engineering, Harbin Engineering University, Harbin, 150001, China
[e] Shandong Linglong Tyre Co., Ltd., Zhaoyuan, 265400, China

ARTICLE INFO

Communicated by P. Borghesani

Keywords:
Reinforcement learning
Estimation fusion
Physically guided neural network
Vehicle sideslip angle
Soft actor–critic

ABSTRACT

Accurate and robust estimation of the vehicle sideslip angle is crucial for maintaining safety under extreme and variable driving conditions. In this paper, we first propose a reinforcement learning-based framework for estimator fusion in vehicle sideslip angle estimation, where the estimator is constructed using physically guided neural networks incorporating gated recurrent unit (GRU) and self-attention mechanisms. First, the physical knowledge of vehicle kinematics and dynamics is analyzed to design the input configuration of the dynamically and kinematically guided neural network estimators. Second, a GRU-based neural network with self-attention is developed to capture both instantaneous and long-range dependencies in time-series signals, serving as the employed neural network estimator. Third, a reinforcement learning-based estimator fusion framework is proposed to integrate the dynamically and kinematically guided neural network estimators. The estimator fusion is modeled as a Markov decision process (MDP) and implemented using soft actor–critic with auto-entropy, extending reinforcement learning to estimation scenarios where actions do not affect state transitions. Finally, the accuracy and robustness of sideslip angle estimation, as well as the generalizability and adaptability of the reinforcement learning-based estimator fusion framework, are validated through diverse real vehicle experiments using both self-collected and public datasets under normal and extreme maneuvers.

1. Introduction

Autonomous driving and collaborative autonomous driving technologies help reduce traffic congestion, improve vehicle safety, and optimize energy efficiency, paving the way for safer, more efficient, and sustainable intelligent transportation systems [1,2]. The effectiveness of these advanced technologies relies on an accurate vehicle state. The vehicle sideslip angle, as a key indicator of lateral stability, is critical for maintaining the vehicle's stability boundaries within active safety control and autonomous driving systems, particularly in collaborative autonomous driving under complex and varying driving scenarios [3,4].

However, due to the cost limitation of mass production vehicles, it is often infeasible to equip additional sensors that can directly and accurately measure sideslip angle. As a result, estimating sideslip angle using measurement signals from on-board sensors

* Corresponding author.
E-mail addresses: gongcf21@mails.jlu.edu.cn (C. Gong), y.kong@surrey.ac.uk (Y. Kong), Dong.Zhang@brunel.ac.uk (D. Zhang), yao.ma@hrbeu.edu.cn (Y. Ma), bo_lu@linglong.cn (B. Lu), shuyong_xing@linglong.cn (S. Xing), zong.changfu@ascl.jlu.edu.cn (C. Zong).

https://doi.org/10.1016/j.ymssp.2025.113211
Received 3 June 2025; Accepted 7 August 2025
Available online 21 August 2025

has become a practical solution [5]. In parallel, combining multiple estimators via fusion methods has emerged as a promising approach in state estimation – not limited to sideslip angle estimation – offering improved robustness and generalization under varying conditions, and proving valuable in safety-critical applications such as sideslip angle estimation.

1.1. Related work

Accordingly, related work is discussed in two areas: vehicle sideslip angle estimation and estimator fusion methods.

1.1.1. Vehicle sideslip angle estimation

Current sideslip angle estimation methods are categorized into model-based, data-driven, and hybrid approaches.

Model-based methods are further divided into dynamics-based, kinematics-based, and combined-model approaches in terms of physical knowledge [1,6]. Dynamics-based methods integrate vehicle dynamics and tire models with advanced observers or filters to achieve high estimation accuracy under strong dynamic conditions [7,8]. Since filter parameters significantly affect estimation, adaptive filters have been proposed [3,9,10]. However, their strong dependence on accurate vehicle and road parameters, and the computational burden of nonlinear models limits real-time performance. Kinematics-based approaches use vehicle motion relationships, decoupling inherent vehicle parameters [11]. However, they are prone to drift and require reset strategy to limit cumulative errors. Global navigation satellite systems (GNSS)-assisted methods have been introduced to correct accumulated errors [12,13] with event-trigger mechanism [14]. Kinematics-based approaches are less accurate under aggressive maneuvers, while GNSS signals are easily affected by obstacles and have low update rates. To address the limitations of single model, some researchers have combined dynamics- and kinematics-based approaches [15–18]. A fuzzy logic-based approach integrating a velocity Kalman filter (KF) from kinematics and an augmented KF from dynamics was proposed [19]. However, the combined-model approaches largely depends on the fusion strategy, current strategies struggle to adaptively adjust weights in time-varying driving conditions, reducing estimation accuracy. A detailed discussion of fusion strategies is given in the next subsection.

Data-driven methods have gain attention for the ability to capture complex nonlinear relationships and overcome the limitations of model-based approaches [20]. Sun applied long short-term memory (LSTM) and bidirectional LSTM (BiLSTM) networks with particle swarm optimization for hyperparameter tuning [21]. Polzleitner proposed an recurrent neural network (RNN)-based method incorporating feature importance, uncertainty prediction, and Monte Carlo dropout for confidence interval estimation [22]. Novotny developed a hybrid neural network combining convolutional neural network (CNN) with GRU for snowy roads [23]. However, existing methods like LSTM, RNN, and GRU excel at capturing short-term correlations, they struggle with long-term temporal dependencies, limiting adaptability to varying driving conditions. Moreover, most methods use all available measurements as inputs without selection, introducing irrelevant or redundant information. As the importance of each signal varies with driving conditions, these overload inputs reduce estimation accuracy and increase computational cost.

Hybrid methods combine model-based and data-driven approaches to take advantage of both physical modeling and learning flexibility [2,3,24,25]. Bertipaglia proposed a CNN-assisted unscented Kalman filter (UKF), using CNN output as a pseudo-measurement for sideslip angle estimation, but parameter sensitivity and model mismatch across vehicles and scenarios hinder accuracy [26]. Lio introduced a neural network-based lateral velocity estimator with modular design and nonlinear feedback, but reliance on kinematic assumptions limits performance under dynamic conditions, while sensor noise causes long-term drift [27]. Most hybrid methods rely on a single physical model, supplemented by the neural network, and thus inherit the limitations of the single model. For instance, dynamic-model-based hybrid method ensures accuracy when the dynamic equations are applicable [26], while kinematic model-based hybrid methods accumulate errors under aggressive conditions [27]. When real-world scenarios deviate from single-model assumptions, neural network is forced to learn under incorrect physical constraints, ultimately reducing estimation accuracy, making it insufficient to rely solely on a single physical model.

To achieve accurate and robust sideslip angle estimation across various driving scenarios, a fusion framework that integrates multiple physical knowledge estimators is needed by adaptively assigning weights. By fusing estimators based on different physical principles (e.g., dynamics-based and kinematics-based models) or sensors at the decision level [28], reliance on any single source is effectively reduced, and fault tolerance is improved, making it an indispensable component of safety-critical applications.

1.1.2. Estimator fusion methods

Beyond vehicle sideslip angle estimation, estimator fusion has been adopted in a broad applications. With the increasing complexity of modern engineering systems and the need for precise control under various conditions, estimator fusion has been widely applied in industrial devices [28,29], power systems [30,31], and autonomous driving technologies [32,33]. Thus, developing a flexible and adaptive estimator fusion framework is essential for both vehicle state estimation and general estimation tasks.

Existing estimator fusion can be divided into rule-based, Bayesian-based, and data-driven methods [28,31,34].

Rule-based methods usually assign estimator weights using predefined logic from expert knowledge. Common strategies include fuzzy logic [19,35,36], error weighting [37–39], entropy weighting [40,41], and custom-designed indicators [42]. These methods are usually concise and straightforward, but performance is limited in dynamically uncertain environments. Chen proposed a fuzzy logic-based fusion strategy for sideslip angle estimation, using vehicle speed, tire sideslip angle, and slip ratio to optimize weights [43]. A matrix-weighted fusion strategy with confidence and error covariance was proposed to combine multiple parallel estimators and mitigate errors caused by non-line-of-sight (NLOS) conditions [38]. However, these strategies are based on predefined static or semi-static rules, making them lack robustness in dynamically changing scenarios.

Bayesian-based methods assign estimator weights through probabilistic modeling [33,44], typically using Kalman filters and their extensions [15,24]. Park proposed an interacting multiple model (IMM) Kalman filter to adjust weights based on real-time vehicle states [15]. But performance is highly dependent on hyperparameters (e.g., process and measurement noise), and suffer from computational complexity in nonlinear modeling.

Data-driven methods learn fusion strategies from datasets, exhibiting strong adaptability and the ability to capture nonlinear relationships, where machine learning (ML) and deep learning (DL) techniques are typically implemented [29,32,45]. Lee proposed a feed-forward neural network to adaptively determine weights of two finite memory estimator (FME) estimators based on vehicle dynamics and kinematics [45]. A light gradient boosting machine (LightGBM)-based fusion method was employed to estimate instantaneous energy consumption by fusing the outputs from multiple regression algorithms [32]. Despite their flexibility, these supervised data-driven models often lack generalizability across varying environments and require large amounts of high-quality training data to maintain robustness.Reinforcement learning (RL) has recently emerged as a promising method for estimator fusion, as its exploration-based mechanism suits dynamic scenarios well. It has shown excellent results in fields such as agriculture [46] and simultaneous localization and mapping (SLAM) [47]. Sharma proposed a deep Q-network (DQN)-based estimator fusion method to select the best baseline model for estimating reference evapotranspiration in water management.

However, existing RL-based fusion strategy still has limitations. For example, Sharma's single-model voting method is difficult to fully exploit model complementarity. Furthermore, DQN tends to overestimate, leading to reduced estimation robustness and accuracy [46]. Wong proposed a proximal policy optimization (PPO)-based multi-sensor fusion method in SLAM, adjusting weights based on environment and anomaly detection, but its limited exploration and sensitivity to input noise weaken robustness [47]. In general, present RL-based estimator fusion methods face challenges such as overestimation, limited exploration, and sensitivity to noise. Moreover, to the best of the author's knowledge, there is an absence of methodical mathematical framework for applying RL theory to estimator fusion. Unlike classical RL-based control problems, the state transition probability distribution in estimator fusion is independent of the agent's actions. This key difference complicates the direct implementation of classical RL framework to estimator fusion and highlights the need for theoretical reformulation to expand the applicability of RL theory.

Therefore, the main gaps to be bridged in the literature are summarized as follows:

(1) Data-driven methods for vehicle sideslip angle estimation often ignore the long-term temporal dependencies and most available measurements are used as inputs without proper guidance from vehicle physical knowledge. While some studies incorporate either kinematic or dynamic information, the fusion of both remains largely unexplored, resulting in reduced accuracy and robustness.
(2) Current estimator fusion methods lack flexibility and adaptability, while existing reinforcement learning-based methods rarely derive a methodical mathematical formulation for the key distinction that the environment's state transition probability function is independent of the agent's actions. They also struggle to effectively address challenges such as overestimation, limited exploration, and noise sensitivity.

1.2. Contributions

To address the above challenges, this paper proposes a reinforcement learning-based fusion framework for vehicle sideslip angle estimation. The framework integrates dynamically and kinematically guided neural networks built on GRU and self-attention as complementary estimators to enhance accuracy and robustness under diverse driving conditions. The main highlights of this paper are summarized as follows:

(1) A novel physically guided neural network is proposed for vehicle sideslip angle estimation. Vehicle dynamics and kinematics guide the dynamics-based and kinematics-based neural network estimators, the neural network captures transient and long-term dependencies in time-series signals by combining GRU and self-attention mechanisms.
(2) A new reinforcement learning-based estimator fusion framework is developed using the soft actor–critic algorithm with auto-entropy, along with sound mathematical formulation. The fusion process is modeled as a Markov decision process, explicitly accounting for the independence of state transitions from agent actions. This framework extends RL beyond control-oriented problems to estimator fusion tasks with theoretical clarity.
(3) An innovative sideslip angle estimator is presented by integrating dynamically and kinematically guided neural network estimators within the proposed RL-based fusion framework. Experiments on two different real vehicle datasets verify estimation accuracy and robustness of the GRU-based neural network estimator with self-attention, along with the generalizability and adaptability of RL-based fusion framework. The potential deployability of the proposed estimator in production vehicles is also discussed.

The remainder of this paper is organized as follows. Section 2 introduces the proposed methodology, including the guidance from vehicle dynamics and kinematics for the neural network sideslip angle estimator, the GRU and self-attention architecture, and the reinforcement learning-based estimator fusion framework. The experimental results on both self-collected and public real-vehicle datasets are analyzed in Section 3. The conclusion is provided in Section 4.

2. Methodology

The structure of the vehicle sideslip angle estimation method is shown in Fig. 1. The guidance of vehicle dynamic and kinematic for neural network sideslip angle estimator are introduced in Section 2.1. The GRU-based neural network with self-attention for sideslip angle estimation is built in Section 2.2. The reinforcement learning-based estimator fusion framework using soft actor–critic algorithm with detailed mathematical implementation is proposed in Section 2.3.

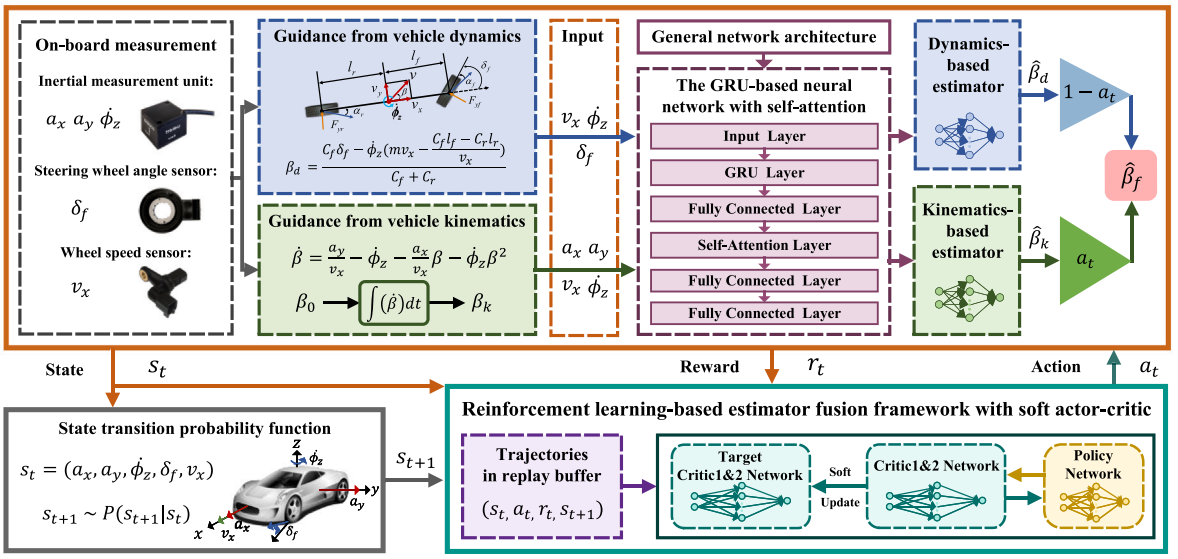


Fig. 1. Structure of sideslip angle estimation method within the reinforcement learning-based fusion framework using dynamically and kinematically guided neural network estimators.

2.1. Guidance of vehicle dynamic and kinematic for physical neural network estimators

Production vehicles are typically equipped with inertial measurement unit to measure longitudinal acceleration a_x, lateral acceleration a_y, and yaw rate $\dot{\phi}_z$, the wheel speed sensor for longitudinal velocity v_x indirectly, and the steering wheel angle sensor for the front wheel steering angle δ_f. Common data-driven methods for sideslip angle estimation use all these measurements as inputs. While this approach aims to capture the full vehicle information, it may introduce irrelevant or redundant information, as signal importance varies across diving condition. For example, during high-speed cornering, a_y and $\dot{\phi}_z$ are critical for accurately estimating sideslip angle, while a_x has minimal impact. In contrast, during straight-line acceleration or braking, a_x becomes more relevant, while a_y and $\dot{\phi}_z$ contribute little useful information. Including all measurements without accounting for varying importance of each signal under driving conditions may lead to increased noise and estimation error. Therefore, the physical mechanisms underlying the vehicle kinematic and dynamic perspectives are analyzed in the following to construct kinematics-based and dynamics-based neural network estimators.

2.1.1. Guidance for kinematics-based neural network estimator

The sideslip angle β is the angle between the vehicle's longitudinal axis and its actual motion direction, defined by the longitudinal velocity v_x and the combined velocity v at the center of gravity (CG), as shown in Fig. 2(a). For small angles (in radians), the approximation applies:

$$\beta = \arctan\left(\frac{v_y}{v_x}\right) \approx \frac{v_y}{v_x}. \tag{1}$$

In a non-inertial reference frame attached to the vehicle body, the longitudinal and lateral accelerations satisfy:

$$a_y = \dot{v}_y + \dot{\phi}_z v_x, \quad a_x = \dot{v}_x - \dot{\phi}_z v_y, \tag{2}$$

where $\dot{v}_x$ and $\dot{v}_y$ are the derivatives of the longitudinal and lateral velocities. The time derivative $\dot{\beta}$ can be further expanded as:

$$\dot{\beta}(t) = \frac{a_y(t)}{v_x(t)} - \dot{\phi}_z(t) - \frac{a_x(t)}{v_x(t)}\beta(t) - \dot{\phi}_z(t)\beta^2(t), \tag{3}$$

where $\dot{\beta}$ follows an ordinary differential equation (ODE), numerical methods like Euler or Runge–Kutta iteratively integrate the ODE to compute $\beta_k(t)$.

$$\beta_k(t) = \beta(0) + \int_0^t \dot{\beta}(\tau)\, d\tau. \tag{4}$$

where $\beta(0)$ represents the initial condition. Thus, the kinematics-based method relies on measurable signals $\{a_x, a_y, v_x, \dot{\phi}_z\}$ and applies universally across vehicle types. However, as $\dot{\beta}$ follows an ODE requiring numerical integration, its accuracy depends on robust integration rules for starting and resetting, as well as filtering and backward integration techniques to mitigate drift and cumulative errors.

2.1.2. Guidance for dynamics-based neural network estimator

The 3-DOF vehicle dynamics model is adopted, as shown in Fig. 2(a), with the following assumptions:

(1) The vehicle's degrees of freedom are lateral, longitudinal, and yaw motion.
(2) The front wheel steering angle δ_f is small (less than 8 degrees), with $\cos\delta_f \approx 1$, under an error of 1%.
(3) The vehicle is rear-wheel drive (RWD), and the front-wheel longitudinal force (F_{xf}) is neglected in the lateral dynamics equation.

F_{yf} and F_{yr} are the lateral tire forces on the front and rear axles, respectively. m is the vehicle mass, l_f and l_r are the distances from the CG to the front and rear axles, respectively. δ_f is given by $\delta_f = \delta_{sw}/i_{sw}$, where δ_{sw} is the steering wheel angle and i_{sw} is the steering gear ratio.

Linear tire model has limitations under large tire sideslip angles. To more accurately describe nonlinear tire behavior, the nonlinear cornering stiffness C_y derived from the UniTire model is adopted:

$$C_y = K_y(S_x - 1)\bar{F}\boldsymbol{\Phi}^{-1}, \tag{5}$$

where S_x is longitudinal slip ratio, K_y is lateral slip stiffness, $\bar{F}$ denotes normalized resultant tire force, and $\boldsymbol{\Phi}$ accounts for the combined slip ratio, more details can be found in [19]. C_{yf} and C_{yr} are the cornering stiffnesses of the front and rear axles, respectively. Therefore, the equivalent lateral tire forces are expressed using the above nonlinear tire model:

$$F_{yf} = C_{yf}\left(\delta_f - \beta - \frac{l_f\dot{\phi}_z}{v_x}\right), \quad F_{yr} = C_{yr}\left(-\beta + \frac{l_r\dot{\phi}_z}{v_x}\right). \tag{6}$$

The lateral dynamics equation is expressed as:

$$m(v_x\dot{\beta} + \dot{v}_x\beta + v_x\dot{\phi}_z) = C_{yf}\left(\delta_f - \beta - \frac{l_f\dot{\phi}_z}{v_x}\right) + C_{yr}\left(-\beta + \frac{l_r\dot{\phi}_z}{v_x}\right). \tag{7}$$

The sideslip angle can be calculated using first-order Euler approximation :

$$\begin{aligned}\beta_d(t) = \beta(t-\Delta t) &+ \frac{\Delta t}{mv_x(t-\Delta t)}\Big[\, C_{yf}(t-\Delta t)\cdot\delta_f(t-\Delta t) \\ &- \big(C_{yf}(t-\Delta t) + C_{yr}(t-\Delta t) + m\dot{v}_x(t-\Delta t)\big)\cdot\beta_d(t-\Delta t) \\ &+ \big(-C_{yf}(t-\Delta t)l_f + C_{yr}(t-\Delta t)l_r - mv_x^2(t-\Delta t)\big)\cdot\frac{\dot{\phi}_z(t-\Delta t)}{v_x(t-\Delta t)}\,\Big].\end{aligned} \tag{8}$$

From the above formulations, it is found that β_d is derived from the dynamics-based method using measurable signals $\{v_x, \dot{\phi}_z, \delta_f\}$ and vehicle parameters $\{C_{yf}, C_{yr}, l_f, l_r, m\}$. The dynamics-based method suits rapid lane changes or emergency maneuvers. But its reliance on precise vehicle modeling and accurate nonlinear tire characteristics, the need for comprehensive parameter calibration raises the threshold for practical deployment. Otherwise, inappropriate parameter settings may lead to degraded estimation accuracy.

2.2. GRU-based neural network with self-attention for sideslip angle estimator

The GRU holds the ability to capture instantaneous and continuous correlations through its gating mechanism, effectively modeling short-range dependencies [48,49]. In contrast, the self-attention layer complements this by providing a global contextual understanding, thereby addressing long-range dependencies [50]. The fully-connected layer integrates these features seamlessly, ensuring they are compatible for subsequent processing. The illustration of the GRU-based network with self-attention is shown in Fig. 2(b). The network architecture consists of six layers: an input layer incorporating guidance from vehicle physics, a GRU layer, a fully connected layer, a self-attention layer, and two subsequent fully connected output layers, following the data processing sequence. It effectively combines the strengths of both sequential and global dependency modeling in vehicle sideslip angle estimation.

Given the input time-series data $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t\}$, where $\boldsymbol{x}_t \in \mathbb{R}^{d\times n_w}$, where d is the dimension of input features, t represents the current time step in the sequence, and n_w denotes the length of the temporal input window. Thus $\boldsymbol{x}_t$ represents the input signals over the consecutive time steps from t to $t + n_w - 1$, where $t \geq 1$. The output is given as $\boldsymbol{Y} = \{\boldsymbol{\beta}_{n_w}, \boldsymbol{\beta}_{1+n_w}, \ldots, \boldsymbol{\beta}_{t+n_w-1}\}$.

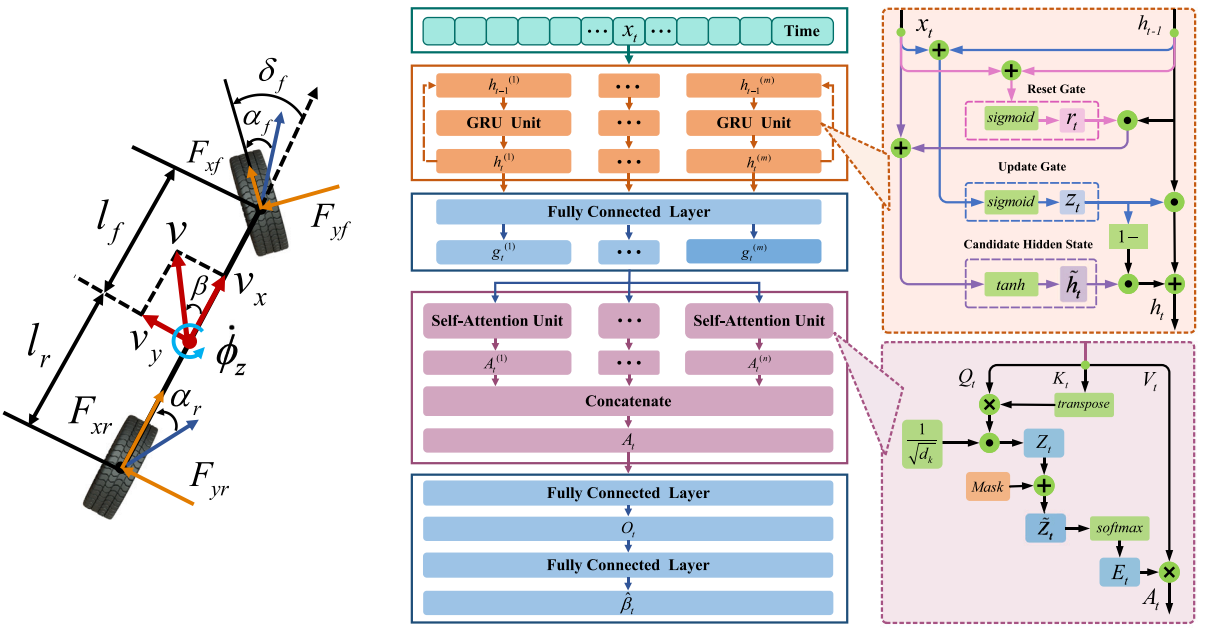
Specifically, for the dynamics-based estimator, $\boldsymbol{x}_t = \{v_x, \dot{\phi}_z, \delta_f\}$; for the kinematics-based estimator, $\boldsymbol{x}_t = \{a_x, a_y, v_x, \dot{\phi}_z\}$; and for the all-input estimator, $\boldsymbol{x}_t = \{a_x, a_y, \dot{\phi}_z, \delta_f, v_x\}$. The neural network architecture remains consistent across configurations. The GRU layer processes the sequence as follows:

The reset gate $\boldsymbol{r}_t$ is designed to regulate the extent to which the previous hidden state $\boldsymbol{h}_{t-1}$ is forgotten:

$$\boldsymbol{r}_t = \sigma\left(\boldsymbol{W}_r\boldsymbol{x}_t + \boldsymbol{U}_r\boldsymbol{h}_{t-1} + \boldsymbol{b}_r\right). \tag{9}$$

The update gate $\boldsymbol{z}_t$ is employed to control the degree to which the previous hidden state $\boldsymbol{h}_{t-1}$ is updated with the new input $\boldsymbol{x}_t$:

$$\boldsymbol{z}_t = \sigma\left(\boldsymbol{W}_z\boldsymbol{x}_t + \boldsymbol{U}_z\boldsymbol{h}_{t-1} + \boldsymbol{b}_z\right). \tag{10}$$



(a) Vehicle model. (b) GRU-based neural network with self-attention.

Fig. 2. Illustration of vehicle model (left) and GRU-based neural network with self-attention (right).

The candidate hidden state $\tilde{\boldsymbol{h}}_t$ is computed as:

$$\tilde{\boldsymbol{h}}_t = \tanh\left(\boldsymbol{W}_h \boldsymbol{x}_t + \boldsymbol{r}_t \odot \left(\boldsymbol{U}_h \boldsymbol{h}_{t-1}\right) + \boldsymbol{b}_h\right). \tag{11}$$

The final hidden state $\boldsymbol{h}_t$ is derived as:

$$\boldsymbol{h}_t = \left(1 - \boldsymbol{z}_t\right) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t. \tag{12}$$

The vectors $\boldsymbol{r}_t, \boldsymbol{z}_t, \tilde{\boldsymbol{h}}_t, \boldsymbol{h}_t \in \mathbb{R}^m$ represent the reset gate, update gate, candidate hidden state, and final hidden state at time step t, where m is the hidden state size (number of GRU units). The weight matrices $\boldsymbol{W}_r, \boldsymbol{W}_z, \boldsymbol{W}_h \in \mathbb{R}^{m\times d}$ map input features to the hidden state space, while $\boldsymbol{U}_r, \boldsymbol{U}_z, \boldsymbol{U}_h \in \mathbb{R}^{m\times m}$ perform hidden-to-hidden transformations. The bias vectors $\boldsymbol{b}_r, \boldsymbol{b}_z, \boldsymbol{b}_h \in \mathbb{R}^m$ refine the reset gate, update gate, and candidate hidden state computations. Here, $\odot$ denotes element-wise multiplication, $\sigma(\cdot)$ is the sigmoid activation function, and $\tanh(\cdot)$ refers to the hyperbolic tangent activation function.

For each time step t, the fully connected layer computes the output $\boldsymbol{g}_t$ incrementally as:

$$\boldsymbol{g}_t = \boldsymbol{W}_g \boldsymbol{h}_t + \boldsymbol{b}_g, \tag{13}$$

where $\boldsymbol{g}_t \in \mathbb{R}^m$, $\boldsymbol{W}_g \in \mathbb{R}^{m\times m}$ is the weight matrix, and $\boldsymbol{b}_g \in \mathbb{R}^m$ is the bias vector. The collection of outputs from the fully connected layer up to time step t forms the sequence $\boldsymbol{G}_t = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_t\}$, where $\boldsymbol{G}_t \in \mathbb{R}^{t\times m}$. This sequence is then fed into the self-attention mechanism, with the queries $\boldsymbol{Q}_t$, keys $\boldsymbol{K}_t$, and values $\boldsymbol{V}_t$ computed as:

$$\boldsymbol{Q}_t = \boldsymbol{G}_t \boldsymbol{W}_Q, \quad \boldsymbol{K}_t = \boldsymbol{G}_t \boldsymbol{W}_K, \quad \boldsymbol{V}_t = \boldsymbol{G}_t \boldsymbol{W}_V, \tag{14}$$

where $\boldsymbol{Q}_t, \boldsymbol{K}_t, \boldsymbol{V}_t \in \mathbb{R}^{t\times d_k}$, $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{m\times d_k}$ are the weight matrices, and d_k is the head dimension.

The scaled dot-product attention is computed at time t:

$$\boldsymbol{Z}_t = \frac{\boldsymbol{Q}_t \boldsymbol{K}_t^T}{\sqrt{d_k}}. \tag{15}$$

To enforce causality, a mask matrix $\boldsymbol{M} \in \mathbb{R}^{t\times t}$ is applied:

$$\tilde{\boldsymbol{Z}}_t = \boldsymbol{Z}_t + \boldsymbol{M}, \quad \boldsymbol{M}_{ij} = \begin{cases} 0, & \text{if } i \geq j, \\ -\infty, & \text{if } i < j. \end{cases} \tag{16}$$

The masked attention scores $\boldsymbol{E}_t$ are normalized using the softmax function:

$$\boldsymbol{E}_t = \mathrm{softmax}(\tilde{\boldsymbol{Z}}_t). \tag{17}$$

The attention output for each head $\boldsymbol{A}_t^{(i)} \in \mathbb{R}^{t\times d_k}$ is calculated as:

$$\boldsymbol{A}_t^{(i)} = \boldsymbol{E}_t \boldsymbol{V}_t. \tag{18}$$

The concatenated output $\boldsymbol{A}_t \in \mathbb{R}^{t\times nd_k}$ from all attention heads is given as follows:

$$\boldsymbol{A}_t = \mathrm{Concat}(\boldsymbol{A}_t^{(1)}, \boldsymbol{A}_t^{(2)}, \ldots, \boldsymbol{A}_t^{(n)}), \tag{19}$$

where n is the number of head, $\boldsymbol{A}_t$ is then passed through the first fully connected layer, producing the intermediate output $\boldsymbol{O}_t$:

$$\boldsymbol{O}_t = \boldsymbol{A}_t \boldsymbol{W}_O + \boldsymbol{b}_O, \tag{20}$$

where $\boldsymbol{W}_O \in \mathbb{R}^{nd_k\times m}$, $\boldsymbol{b}_O \in \mathbb{R}^m$, and $\boldsymbol{O}_t \in \mathbb{R}^{t\times m}$.

Finally, the second fully connected layer computes the predicted sideslip angle $\hat{\boldsymbol{\beta}}_t$:

$$\hat{\boldsymbol{\beta}}_t = \boldsymbol{O}_t \boldsymbol{W}_\beta + \boldsymbol{b}_\beta, \tag{21}$$

where $\boldsymbol{W}_\beta \in \mathbb{R}^m$, $\boldsymbol{b}_\beta \in \mathbb{R}$, and $\hat{\boldsymbol{\beta}}_t \in \mathbb{R}^t$.

The above parameters are automatically learned via backpropagation, and key hyperparameters are tuned using Bayesian optimization to ensure training efficiency and robustness.

2.3. Reinforcement learning-based estimator fusion framework with soft actor–critic

A reinforcement learning-based framework with soft actor–critic is proposed for estimator fusion. Reinforcement learning is extended to scenarios where the agent's actions do not affect the environment's state transition probability function. The soft actor–critic with auto-entropy algorithm is employed within the reinforcement learning-based framework to derive the detailed mathematical formulation. The kinematics-based and dynamics-based neural network estimators are integrated into the proposed reinforcement learning-based framework for sideslip angle fusion estimation.

2.3.1. Formulation of reinforcement learning-based estimator fusion framework

For a bank of N parallel estimation models $\{\hat{\boldsymbol{x}}_1(t), \hat{\boldsymbol{x}}_2(t), \ldots, \hat{\boldsymbol{x}}_N(t)\}$ with the weights $\{w_1(t), w_2(t), \ldots, w_N(t)\}^\top$, the objective function is to minimize the fusion estimation error :

$$\mathcal{L} = \sum_{t=1}^{T} \left\| \boldsymbol{x}_g(t) - \hat{\boldsymbol{x}}_{\mathrm{fusion}}(t) \right\|^2, \quad \hat{\boldsymbol{x}}_{\mathrm{fusion}}(t) = \sum_{i=1}^{N} w_i(t)\hat{\boldsymbol{x}}_i(t), \tag{22}$$

where $\boldsymbol{x}_g(t)$ represents the ground truth, and the weights $\{w_i(t)\}$ satisfy the constraint:

$$\sum_{i=1}^{N} w_i(t) = 1, \quad w_i(t) \geq 0, \forall i. \tag{23}$$

The optimal weights $\{w_i^\star(t)\}$ are determined by solving:

$$\min_{\{w_i(t)\}} \sum_{t=1}^{T} \mathcal{L}(t), \quad \text{subject to } \sum_{i=1}^{N} w_i(t) = 1, \quad w_i(t) \geq 0, \quad \forall t. \tag{24}$$

The goal of fusion estimation is to derive the optimal global estimate $\hat{\boldsymbol{x}}_{\mathrm{fused}}(t)$ by effectively combining the estimates from parallel models, addressing the challenge of achieving accuracy and robustness in varying and uncertain environments with high dimensionality and nonlinearity.

Reinforcement learning provides an exploration-based framework for adaptively determining the optimal fusion weights through interaction with the environment, minimizing the expected estimation error and offering a robust, flexible solution for multi-model fusion in complex and uncertain scenarios. As the foundation of RL, MDP provides a mathematical representation for modeling sequential decision-making problems [51], which is defined as:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r), \tag{25}$$

where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the state transition probability function, and r is the reward function. The next state s_{t+1} depends on both the current state s_t and the action a_t. The state transition function can be written as:

$$P\left(s_{t+1} \mid s_t, a_t\right) = \Pr\left(S_{t+1} = s_{t+1} \,\middle|\, S_t = s_t,\ A_t = a_t\right). \tag{26}$$

It is worth mentioning that the state of environment is unaffected by the actions and only the existing state is observed in estimation scenario. Although actions in RL do not have a direct impact on state evolution, the state still evolves according to its inherent laws or defined dynamics. In this context, a reinforcement learning-based estimator fusion framework is proposed, where the state transition P_p is given by

$$P_p\left(s_{t+1} \mid s_t, a_t\right) = P\left(s_{t+1} \mid s_t\right), \tag{27}$$

where $a_t = \{w_1(t), w_2(t), \ldots, w_N(t)\}^\top$, $w_i(t) \geq 0$ and $\sum_{i=1}^{N} w_i(t) = 1$.

The return at time step t for an infinite-horizon task is defined as the discounted sum of future rewards:

$$R_t = r_t + \gamma\, r_{t+1} + \gamma^2\, r_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k\, r_{t+k}, \quad r(t) = \mathcal{R}\big(\boldsymbol{x}_g(t), \hat{\boldsymbol{x}}_{\text{fusion}}(t)\big), \tag{28}$$

where k represents the future step relative to t, the discount factor $\gamma \in (0, 1]$ controls the impact of future rewards, and $\mathcal{R}$ is the reward function about estimation error.

The RL agent determines a_t at each time step to optimize the fusion process, the state transition P_p is influenced by the current state s_t, independent of the action a_t, but it does affect the reward r_t. The objective of reinforcement learning-based estimator fusion framework is to maximize the expected return:

$$\mathbb{E}_\pi[R_t] = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k\, r\big(s_{t+k},\, a_{t+k}\big) \,\middle|\, \pi\right], \quad s_{t+1} \sim P_p\big(\cdot \mid s_t\big), \quad a_t \sim \pi\big(\cdot \mid s_t\big). \tag{29}$$

Where $\pi(a|s)$ is the policy, specifies how an agent chooses actions $a_t \sim \pi(\cdot|s_t)$ in different states.

2.3.2. Implementation of reinforcement learning-based estimator fusion framework with soft actor–critic

The reinforcement learning-based estimator fusion framework can be implemented using various RL algorithms. The choice of algorithm depends on the comprehensive consideration of the environment's complexity, sample efficiency, stability and computational constraints. SAC is particularly suitable for fusion estimation problems, because the introduction of entropy maximization allows SAC to encourage exploration while stabilizing policy update [52,53]. In addition, SAC optimizes one policy and two Q-value networks simultaneously, reducing variance in value estimates and preventing policy crashes. Moreover, SAC with auto-entropy algorithm further adjusts the temperature parameter, enhancing SAC's exploration and stability.

Given a sampled transition (s_t, a_t, r_t, s_{t+1}) from replay buffer $\mathcal{D}$. Two Q-networks, Q_{θ_1} and Q_{θ_2}, are employed in SAC with the minimization method to reduce overestimation bias and improve stability. π_ϕ is a stochastic policy parameterized by ϕ, and $Q_{\bar{\theta}}$ is a target network with parameters $\bar{\theta}$ updated via soft update to stabilize training.

The loss function $J_Q(\theta)$ for the soft Bellman residual is expressed as:

$$J_Q(\theta) = \frac{1}{2}\, \mathbb{E}_{(s_t, a_t)\sim\mathcal{D}}\Big[\big(Q_\theta(s_t, a_t) - r(s_t, a_t) - \gamma\, \mathbb{E}_{s_{t+1}\sim P_p(\cdot|s_t)}\big[V_{\bar{\theta}}(s_{t+1})\big]\big)^2\Big], \tag{30}$$

the gradient of $J_Q(\theta)$ is expressed as follows:

$$\hat{\nabla}_\theta J_Q(\theta) = \mathbb{E}_{(s_t, a_t)\sim\mathcal{D}}\Big[\nabla_\theta Q_\theta(s_t, a_t)\Big(Q_\theta(s_t, a_t) - r(s_t, a_t) - \gamma\Big(Q_{\bar{\theta}}(s_{t+1}, a_{t+1}) - \alpha \log \pi_\phi(a_{t+1} \mid s_{t+1})\Big)\Big)\Big], \tag{31}$$

where $\nabla_\theta Q_\theta(s_t, a_t)$ is the gradient of the Q-function with respect to its parameters θ.

The policy objective $J_\pi(\varphi)$ is expressed as:

$$J_\pi(\varphi) = \mathbb{E}_{s_t\sim\mathcal{D},\, \epsilon_t\sim\mathcal{N}}\big[\alpha\, \log \pi_\varphi\big(f_\varphi(\epsilon_t; s_t) \mid s_t\big) - Q_\theta\big(s_t,\, f_\varphi(\epsilon_t; s_t)\big)\big], \tag{32}$$

where ϵ_t is noise sampled from a Gaussian distribution $\mathcal{N}$.

The gradient of $J_\pi(\varphi)$ can be derived through the chain rule, as shown below:

$$\hat{\nabla}_\varphi J_\pi(\varphi) = \mathbb{E}_{s_t\sim\mathcal{D},\epsilon_t\sim\mathcal{N}}\Big[\nabla_\varphi\big(\alpha \log \pi_\varphi(f_\varphi(\epsilon_t; s_t) \mid s_t)\big) + \big(\nabla_{a_t}\alpha \log \pi_\varphi(a_t \mid s_t) - \nabla_{a_t} Q_\theta(s_t, a_t)\big)\nabla_\varphi f_\varphi(\epsilon_t; s_t)\Big], \tag{33}$$

where $a_t = f_\varphi(\epsilon_t; s_t)$ is produced by the reparameterization function of policy, the gradient of the Q-function with respect to the action a_t is denoted by $\nabla_{a_t} Q_\theta(s_t, a_t)$, and the dependence of the action on the policy parameters φ is captured by $\nabla_\varphi f_\varphi(\epsilon_t; s_t)$.

The temperature parameter α in SAC algorithm is optimized to match a target entropy $\mathcal{H}$, defining the desired level of randomness in the policy. The objective for α is formulated as follows:

$$J(\alpha) = \mathbb{E}_{a_t\sim\pi_\varphi(\cdot|s_t)}\big[-\alpha \log \pi_\varphi(a_t \mid s_t) - \alpha\mathcal{H}\big], \tag{34}$$

and the gradient of $J(\alpha)$ is computed as:

$$\hat{\nabla}_\alpha J(\alpha) = \mathbb{E}_{a_t\sim\pi_\varphi(\cdot|s_t)}\big[-\log \pi_\varphi(a_t \mid s_t) - \mathcal{H}\big]. \tag{35}$$

To avoid overly complex detailed derivations affecting the readability of the core content, the complete formulations are given in Appendix.

2.3.3. Application on sideslip angle estimator under RL-based fusion framework with soft actor–critic

The reinforcement learning-based estimator fusion framework with SAC is applied to adaptively fuse estimates from kinematics- and dynamics-based estimators to obtain an accurate fusion estimation $\hat{\beta}_f$. The schematic diagram of sideslip angle under reinforcement learning-based estimator fusion framework with SAC is shown in Fig. 3.

The state is defined as:

$$s_t = \big(a_x,\ a_y,\ \dot{\phi}_z,\ \delta_f,\ v_x\big), \tag{36}$$

which consists of measurable signals available on production vehicles, used to observe the motion state of the vehicle.

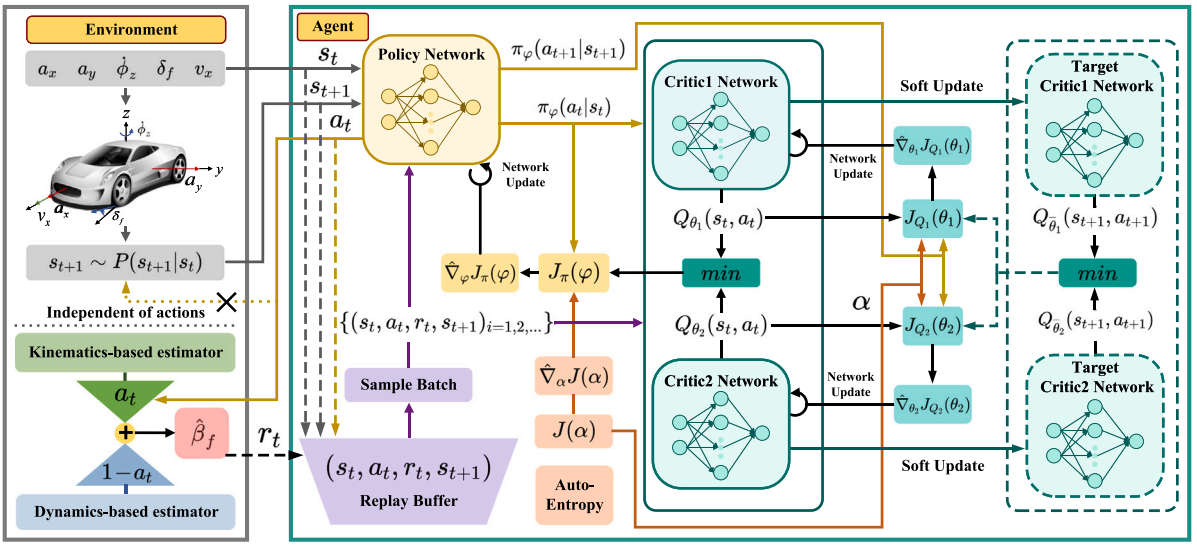


Fig. 3. Diagram of sideslip angle estimator under reinforcement learning-based fusion framework with soft actor–critic.

The action is expressed as the fusion weight assigned to the kinematics-based estimator:

$$a_t = \omega_{k_t}, \quad 0 \le \omega_{k_t} \le 1, \tag{37}$$

and complementary fusion weight for dynamics-based estimator is $(1 - a_t)$.

The state transition is determined by the vehicle dynamics and sensor noise, with no dependence of fusion weights a_t and $(1-a_t)$. The fused sideslip angle estimation is computed as:

$$\hat{\beta}_f = \omega_{k_t}\hat{\beta}_k + (1 - \omega_{k_t})\hat{\beta}_d, \tag{38}$$

where $\hat{\beta}_d$ and $\hat{\beta}_k$ represent the estimate from dynamics- and kinematics-based estimators, respectively.

Considering practical constraints in computational resources, the fusion weights are not updated at every high-frequency signal sampling time of T_s. A longer adjustment duration of T_h is used as the interval for updating the fusion weights to improve computational efficiency.

The reward is represented as a piecewise function, with error weights adjusted according to varying conditions:

$$r(s_t, a_t) = \begin{cases} -w_1\left|\beta_g - \hat{\beta}_f\right|, & \text{if } |\beta_g| > \beta_{\text{th}} \text{ and } |\beta_g - \hat{\beta}_f| > e_{\text{th}}, \\ -w_2\left|\beta_g - \hat{\beta}_f\right|, & \text{if } |\beta_g| > \beta_{\text{th}} \text{ and } |\beta_g - \hat{\beta}_f| \le e_{\text{th}}, \\ -w_3\left|\beta_g - \hat{\beta}_f\right|, & \text{if } |\beta_g| \le \beta_{\text{th}} \text{ and } |\beta_g - \hat{\beta}_f| > e_{\text{th}}, \\ -w_4\left|\beta_g - \hat{\beta}_f\right|, & \text{if } |\beta_g| \le \beta_{\text{th}} \text{ and } |\beta_g - \hat{\beta}_f| \le e_{\text{th}}, \end{cases} \tag{39}$$

where β_{th} is the sideslip angle threshold, e_{th} is the error threshold, and the error weights follow the relationship: $w_1 > w_2 = w_3 > w_4$. This weight hierarchy is designed to prioritize large errors in high-risk scenarios while reducing focus on minor errors in stable conditions.

Subsequently, the update interval T_h is designed as a moving window to accumulate rewards and periodically reset cumulative rewards r_{T_h}. The return R is calculated as the sum of cumulative rewards over all T_h intervals.

$$r_{T_h} = \sum_{t=t_0}^{t_0+T_h/T_s} r(s_t, a_t), \quad R = \sum_{t=T_0}^{T_0+T_a/T_h} r_{T_h}. \tag{40}$$

Where t_0 indicates the starting time during the current T_h interval, the cumulative reward is reset at the end of each T_h, and the process begins anew for the next T_h interval. T_0 indicates the starting time of the overall period T_a.

3. Experimental results and discussion

The results of the GRU-based neural network with self-attention estimator for sideslip angle and the reinforcement learning-based estimator fusion framework with soft actor–critic were analyzed alongside other baseline methods using two different real vehicle datasets to demonstrate the applicability and robustness of the proposed methods.

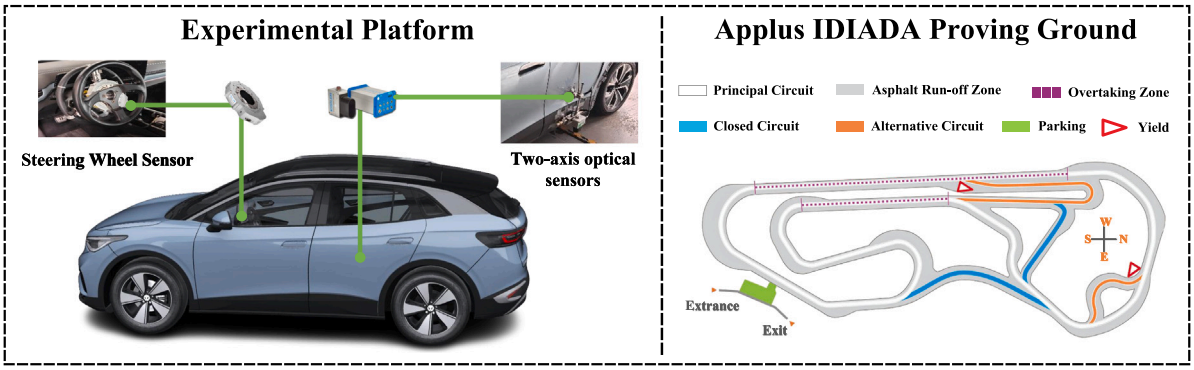


Fig. 4. Experimental platform and test scenario for self-collected dataset.

Table 1
Dataset partitions for self-collected and public datasets.

(a) Self-collected dataset			
Set	Rounds	Samples	Proportion (%)
Training	A, B, E	23 261	59.61
Validation	C	7900	20.25
Testing	D	7860	20.14
(b) Public dataset			
Set	Rounds	Samples	Proportion (%)
Training	A, B, D	24 572	56.06
Validation	C	9600	21.90
Testing	E	9660	22.04

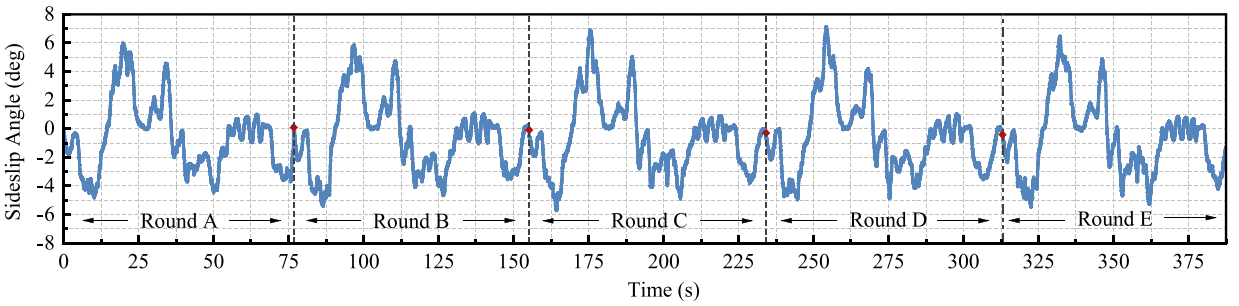
3.1. Experimental setup and data preparation

The self-collected experiments were conducted on the dry handling track of the Applus IDIADA Proving Ground [54]. The SAIC Volkswagen ID.4 served as the test vehicle, equipped with a Kistler Correvit S-Motion DTI for measuring the true sideslip angle, velocity, acceleration, angular rate, and position, as well as a Kistler MSW DTI for recording the steering wheel angle, as shown in Fig. 4. The public dataset is from the REVS vehicle dynamics database [55], collected from experiments carried out at Palm Beach International Raceway using a 1965 Ferrari 250 LM Berlinetta GT, and more details can be found in [56]. All signals in both datasets were sampled at 100 Hz. Multiple consecutive laps of the test were completed on the closed driving course, and each lap was considered a separate experiment in this paper. The two real vehicle datasets differ in terms of sensors, vehicle parameters, and sensor mounting locations.

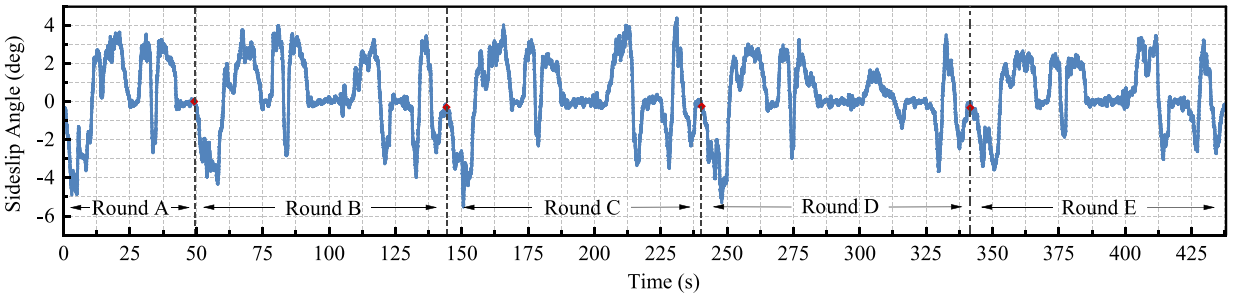
The set partition for training the neural network estimator is introduced. Five rounds (A-E) were selected in the self-collected and public datasets. The true sideslip angles of the two datasets are shown in Fig. 5. Kernel density estimation (KDE) was used to analyze the sideslip angle distributions and guide the partition of training, validation, and testing subsets [57]. The KDE results are presented in Fig. 6. The KDE of the self-collected dataset shows that rounds C and D have similar distributions, making them suitable for consistent validation and testing. Meanwhile, rounds A, B, and E feature diverse scenarios, ensuring that the training set captures a wide range of conditions for robust model training. Thus, rounds A, B, and E were assigned to the training set, round C to the validation set, and round D to the testing set. For the public dataset, comparable distributions between rounds C and E support reliable evaluation, while rounds A, B, and D provide greater variability to improve the network's generalization. Therefore, rounds A, B, and D were used for the training set, round C for the validation set, and round E for the testing set.

The detailed partition is shown in Table 1. For the self-collected dataset, the training set constitutes 59.61% of the total dataset, providing sufficient data diversity. The validation and testing sets represent 20.25% and 20.14% of the data, respectively, ensuring a balanced evaluation. Similarly, for the public dataset, the training set comprises 56.06% of the total data, while the validation and testing sets account for 21.90% and 22.04%, respectively. These proportions provide adequately sized and balanced subsets. The true sideslip angles in geodetic coordinates are presented in Fig. 7. Large sideslip angles mainly occur near high-curvature bends, with the self-collected dataset exhibiting greater sideslip angles, reaching nearly 7 degrees.

As for the evaluation metrics of sideslip angle estimation, root mean squared error (RMSE), mean absolute error (MAE), maximum error (MaxError), relative RMSE, relative MAE, and relative MaxError are used [60], as shown in Table 2. Here, β_g represents the ground truth, $\hat{\beta}$ is the estimate, $\bar{\beta}_g$ is the mean of the ground truth, and n is the total number of samples. Smaller values of RMSE, MAE, MaxError Additionally, for relative RMSE, relative MAE, and relative MaxError, a larger negative percentage indicates a greater improvement in performance of the proposed method compared to the baseline method. These metrics provide a comprehensive framework for evaluation and comparison.

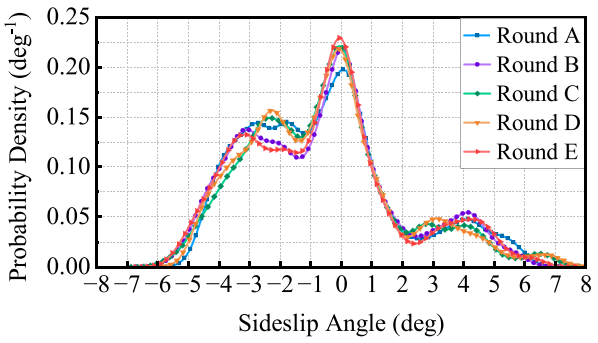


(a) Self-collected dataset.

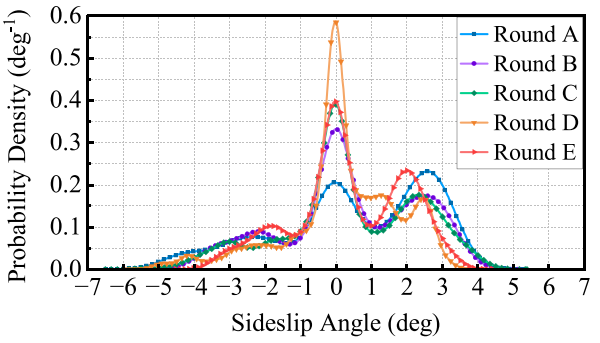


(b) Public dataset.

Fig. 5. True sideslip angles for self-collected and public datasets.

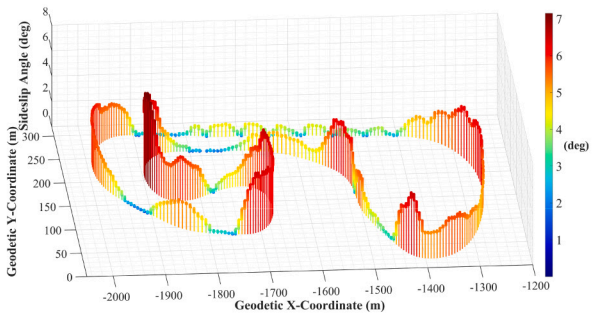


(a) Self-collected dataset.

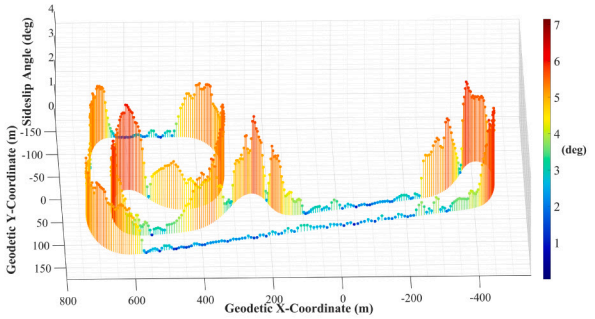


(b) Public dataset.

Fig. 6. KDE results for self-collected and public datasets.



(a) Self-collected dataset.



(b) Public dataset.

Fig. 7. True sideslip angles under geodetic coordinate for self-collected and public datasets.

Table 2
Evaluation metrics.

Metric Name	Meaning	Formula
RMSE	The average estimation error.	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\beta}_i-\beta_{g,i}\right)^2}$
MAE	The average absolute estimation error.	$\frac{1}{n}\sum_{i=1}^{n}\left\lvert\hat{\beta}_i-\beta_{g,i}\right\rvert$
MaxError	The largest estimation error.	$\max_{i=1}^{n}\left\lvert\hat{\beta}_i-\beta_{g,i}\right\rvert$
Relative RMSE	RMSE relative to baseline.	$\frac{\mathrm{RMSE}_{\mathrm{proposed}}-\mathrm{RMSE}_{\mathrm{baseline}}}{\mathrm{RMSE}_{\mathrm{baseline}}}\times 100\%$
Relative MAE	MAE relative to baseline.	$\frac{\mathrm{MAE}_{\mathrm{proposed}}-\mathrm{MAE}_{\mathrm{baseline}}}{\mathrm{MAE}_{\mathrm{baseline}}}\times 100\%$
Relative MaxError	MaxError relative to baseline.	$\frac{\mathrm{MaxError}_{\mathrm{proposed}}-\mathrm{MaxError}_{\mathrm{baseline}}}{\mathrm{MaxError}_{\mathrm{baseline}}}\times 100\%$

Table 3
Comparison of estimation in public dataset.

Methods	RMSE (deg)	Relative RMSE (%)
Factor graph (baseline) [58]	0.57	−48.07
Kalman filter (baseline) [58]	0.87	−65.98
Parametric approach (baseline) [59]	0.539	−45.08
Luenberger (baseline) [27]	0.73	−59.45
GRU (baseline) [27]	0.57	−48.07
LSTM (baseline) [27]	0.52	−43.08
KS-NN (baseline) [27]	0.40	−26.00
GRU-based with self-attention (proposed)	0.296	/

3.2. Evaluation of GRU-based neural network with self-attention for sideslip angle estimator

The training sets for the neural network estimator were as follows: the all-input configuration was applied uniformly to the input layer, the optimizer was Adam, the batch size was 128, and the initial learning rate was 0.01, which was reduced by a factor of 0.5 every 30 epochs. Additionally, a gradient threshold of 10 was applied to prevent gradient explosion during training. The length of temporal input window $n_w = 10$ (0.1 s) was adopted for the sideslip angle estimator based on the vehicle dynamics time scale. This trade-off setting was chosen to capture short-term dynamics while avoiding long-term dependencies that could lead to overfitting.

As for the public dataset, the results of various estimation methods, including the factor graph [58], Kalman filter [58], parametric approaches [59], Luenberger observer [27], GRU [27], LSTM [27], KS-NN [27], and the proposed GRU-based neural network with self-attention estimator in this paper, are compared in Table 3. It can be seen that the estimation performance of traditional model-based methods like the KF and observer is poor, with RMSE ranging from 0.7 to 0.8 degrees. The GRU and LSTM neural networks improve accuracy, reducing RMSE to 0.52–0.57 degrees. The factor graph and parametric approach perform similarly to neural networks. KS-NN, a neural network estimator based on physical knowledge, achieves a lower RMSE of 0.4 degrees. In contrast, the proposed GRU-based neural network with self-attention estimator demonstrates the best estimation performance among all the above methods, reaching the lowest RMSE of 0.296 degrees, which was reduced by 26% compared to KS-NN, highlighting the superiority of the proposed estimator. Its unique architecture effectively captures long-term and transient time-series features to achieve accurate estimation.

To better evaluate the effectiveness of the GRU-based neural network with self-attention in sideslip angle estimation, comparisons were also made with two baseline models: one using a GRU layer, referred to as the GRU-only neural network model, and the other employing a self-attention layer, referred to as the self-attention-only neural network model. This comparison is an ablation study to evaluate sole contribution of the main module. GRU, widely used in time-series tasks such as vehicle state prediction, serves as a strong baseline in this study. While self-attention excels at capturing long-range dependencies (e.g., in machine translation), it is not well-suited for transient dynamics modeling. So the self-attention-only model is not a competitive benchmark in this paper, but rather a means to evaluate the role of the self-attention mechanism. Bayesian optimization was employed to tune hyperparameters to ensure that each model was evaluated under conditions that maximized its performance. The hyperparameters included the number of GRU units, attention heads, and training epochs. A maximum of 100 evaluations was performed using an acquisition function based on expected improvement to minimize RMSE on the validation dataset.

The comparative results are shown in Table 4, the similar performance between the GRU-only model (strong baseline) and the proposed model on the training and validation sets indicates that these models have been adequately trained to their optimal configurations. This ensures a fair comparison on the testing set based on models' best performance to reflect models' true generalization ability, and detailed sideslip angle estimation on the testing sets is provided in Fig. 8.

For self-collected dataset, the estimate of testing set is shown in Fig. 8(a). The self-attention-only neural network estimator had the worst estimation accuracy, with RMSE exceeding 0.3 degrees and maximum error approaching 2 degrees. The GRU-only

Table 4
Comparison of GRU-based neural network with self-attention for sideslip angle estimation.

(a) Results of self-collected dataset.

Datasets	Methods	RMSE (deg)	Relative RMSE (%)	MAE (deg)	Relative MAE (%)	MaxError (deg)	Relative MaxError (%)
Training Set	GRU-only	0.158	−5.70	0.112	−6.25	1.206	5.89
	Self-attention-only	0.312	−52.24	0.231	−54.55	1.825	−30.03
	GRU-based with self-attention	0.149	/	0.105	/	1.277	/
Validation Set	GRU-only	0.147	−2.72	0.101	−0.99	0.972	−10.49
	Self-attention-only	0.350	−59.14	0.237	−57.81	1.912	−54.50
	GRU-based with self-attention	0.143	/	0.100	/	0.870	/
Testing Set	GRU-only	0.174	−8.62	0.108	−2.78	1.610	−9.25
	Self-attention-only	0.332	−52.11	0.230	−54.35	2.012	−27.39
	GRU-based with self-attention	0.159	/	0.105	/	1.461	/

(b) Results of public dataset.

Datasets	Methods	RMSE (deg)	Relative RMSE (%)	MAE (deg)	Relative MAE (%)	MaxError (deg)	Relative MaxError (%)
Training Set	GRU-only	0.320	−7.50	0.230	−8.70	1.927	−3.94
	Self-attention-only	0.439	−32.57	0.336	−37.50	1.900	−2.58
	GRU-based with self-attention	0.296	/	0.210	/	1.851	/

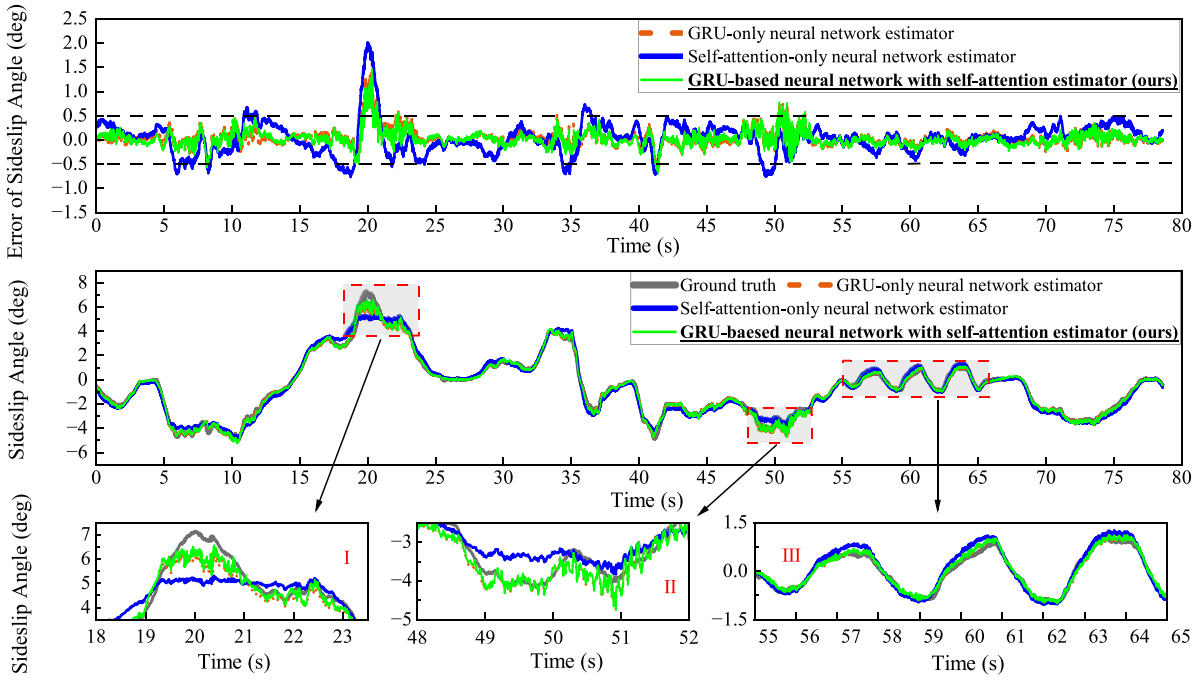
neural network estimator performed better, achieving RMSE of 0.174 degrees, with maximum error of 1.610 degrees. The proposed estimator achieved the lowest RMSE among the three models, with only 0.159 degrees, which is reduced by 8.62% and 52.11% compared to the GRU-only and self-attention-only neural network estimators, respectively. Its Maximum error is 1.461 degrees, which is 9.25% smaller than that of the GRU-only neural network estimator. Its MAE remained around 0.100 degrees, which is also the smallest among the three models. It is observed that the proposed estimator maintains stable and accurate estimation even under extreme dynamic conditions, such as in regions I and II. In contrast, the self-attention-only neural network estimator failed to capture sideslip angle trend, with its maximum error reaching 2 degrees in regions I. The GRU-only neural network estimator could estimate the sideslip angle trend but exhibits larger errors compared to the GRU-based neural network with self-attention estimator. For normal driving conditions, such as in region III, estimation errors of all estimators were relatively smaller. In addition, all three estimators exhibited some degree of estimation ahead of the ground truth, particularly during transitions from zero to positive. The self-attention-only neural network estimator showed the most pronounced early estimate, while the proposed estimator was only slightly ahead.

For public dataset, testing set results are presented in Table 4 due to space constraints. Overall estimation errors were larger than those in self-collected dataset, but differences among the three estimators were reduced. The GRU-based neural network with self-attention estimator achieved the lowest RMSE of 0.296 degrees, and the smallest maximum error of 1.851 degrees among the three estimators, demonstrating the best estimation performance. The self-attention-only neural network estimator had the highest RMSE of 0.439 degrees, which is 32.57% higher than that of the proposed method. The GRU-only neural network estimator exhibited moderate performance, with RMSE of 0.320 degrees and MAE of 0.230 degrees, with most estimation errors remaining within ±1 degrees. However, its RMSE, MAE, and maximum error were 7.50%, 8.70%, and 3.94% larger, respectively, compared to the proposed model. The estimation results for testing set are shown in Fig. 8(b). It can be observed that the self-attention-only neural network estimator cannot estimate trend of sideslip angle accurately, leading to large errors in high-curvature bends, such as in region I, while the proposed estimator had smaller estimation errors. For region II, a sharp transient sideslip state induced by sudden steering, the GRU-only model showed larger estimation errors between 34.5–35.7 s, whereas the GRU-based model with self-attention more accurately captured the transient dynamics due to the enhanced estimation robustness from the self-attention layer. For region III, a high positive sideslip angle condition, the GRU-only model exhibited larger errors between 67.5–79 s. In contrast, the proposed estimator provided more accurate and smoother peak estimating, indicating improved generalization under sustained high dynamic conditions.

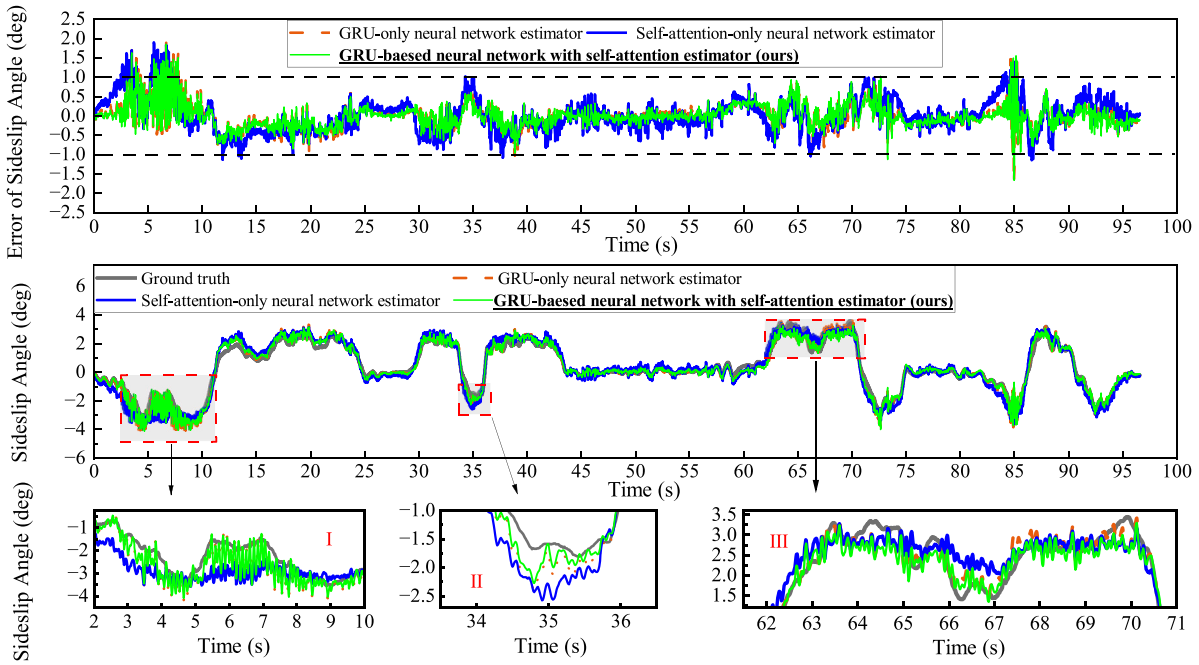
In summary, the GRU-based neural network with self-attention estimator accurately estimates sideslip angles across various driving conditions, as evidenced by the improvements in quantitative relative results. In severe transient dynamic scenarios, it effectively estimates the varying trend while minimizing oscillations. While the GRU-only estimator performs well in moderate scenarios, it shows significant performance degradation under rapid maneuvers, particularly regarding maximum error and local transient accuracy. The self-attention-only neural network estimator, although not as effective as a strong estimator, is beneficial when combined with GRU by capturing extended temporal patterns and sharp local features, thereby mitigating GRU's limitations in handling transient dynamics. The above analysis demonstrates that the proposed GRU-based neural network with self-attention is able to integrate the GRU unit's ability to capture instantaneous dependencies with the strength of self-attention in extracting long-term correlations.

3.3. Evaluation of RL-based fusion framework with soft actor–critic for sideslip angle estimator

Based on the guidance from vehicle dynamics and kinematics for the neural network estimator in Section 2.1, dynamics-based and kinematics-based neural network estimators for sideslip angle were developed to integrate with the reinforcement learning-based estimator fusion framework. Specifically, the kinematics-based and dynamics-based estimators were constructed using the



(a) Estimation comparison on self-collected dataset.

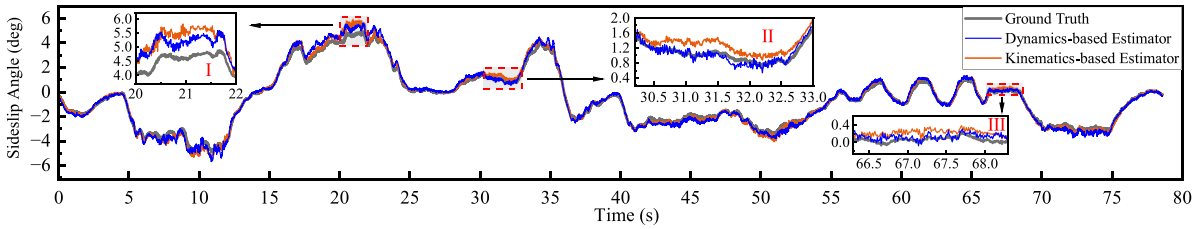


(b) Estimation comparison on public dataset.

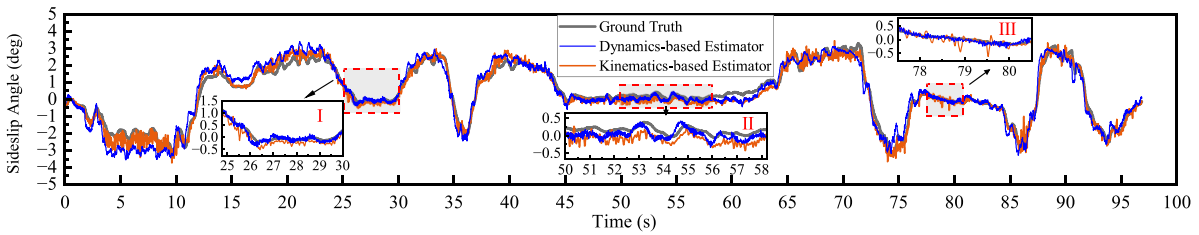
Fig. 8. Results of GRU-based neural network with self-attention for sideslip angle estimation on self-collected and public datasets.

proposed GRU-based neural network with self-attention architecture. The kinematics-based estimator utilized the measurable signals $\{a_x, a_y, v_x, \dot{\phi}_z\}$ as input, while the dynamics-based estimator used $\{v_x, \dot{\phi}_z, \delta_f\}$.

Additional real-vehicle data with different tire configurations were collected on the same testing ground for the self-collected dataset. For the public dataset, an independent real-vehicle dataset, distinct from the training dataset but under the same driving



(a) Estimation comparison on self-collected dataset.



(b) Estimation comparison on public dataset.

Fig. 9. Results of dynamics-based and kinematics-based estimators for sideslip angle on independent datasets of self-collected and public datasets.

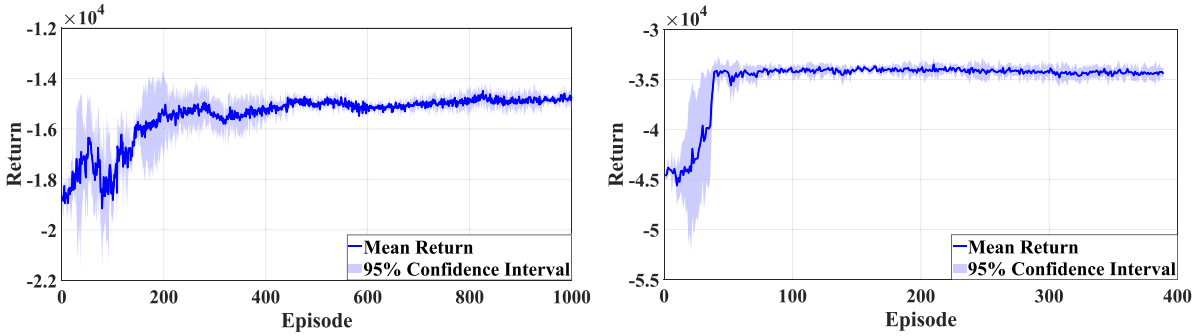
conditions, was selected. These datasets were used to comprehensively evaluate the reinforcement learning-based estimator fusion framework employing dynamically and kinematically guided neural network estimators. To distinguish between them, the testing set from the training dataset is referred to as the trained dataset testing set, while the additional test dataset is termed the independent dataset testing set.

The quantitative results of the kinematics-based and dynamics-based neural network estimators are shown in Table 5. The detailed sideslip angle estimates of dynamics-based and kinematics-based estimators on independent datasets of self-collected and public datasets as shown in Fig. 9. It is observed that the kinematics-based estimator performed better in most driving scenarios. The use of more inputs and simpler integration in the kinematics-based estimator enables easier learning and results in higher accuracy. Although the dynamics-based estimator had lower overall accuracy, it relies on a compact input set with greater model nonlinearity and complexity, which increases learning difficulty and potentially reducing accuracy. Its maximum estimation error under normal driving conditions was smaller, as observed in independent dataset of self-collected dataset. Specifically, in Region I of the self-collected dataset, the sideslip angle showed a sharp transient increase. The dynamics-based estimator closely estimated the ground truth with minimal estimation error, while the kinematics-based estimator exhibited noticeable overestimation. In the small-angle Region II, the dynamics-based estimator again outperformed the kinematics-based estimator, whereas the latter displayed a certain degree of delay and overestimation. In the straight-line Region III, the dynamics-based estimator provided slightly smoother and more stable results. Similarly, in Regions I, II, and III of the public dataset, the dynamics-based estimator better estimated small-amplitude transients with reduced fluctuation compared to the kinematics-based estimator. These results highlight the robustness and superior performance of the dynamics-based estimator, particularly under transient and rapidly changing conditions. While kinematics-based estimator may suffer from larger errors during rapid transients due to weakened motion relevance and input redundancy, dynamics-based estimator showed more stable performance.

Despite the absence of some input signals in kinematics- and dynamics-based estimators, their estimation performance remains acceptable within an acceptable error range, with the RMSE of estimation error usually around 0.5 degrees. Analysis of the above results shows that kinematics-based and dynamics-based estimators each excel in specific estimation scenarios. Accordingly, a reinforcement learning-based estimator fusion framework with soft actor–critic algorithm is proposed to integrate kinematics-based and dynamics-based estimators. This enables adaptive fusion for optimal sideslip angle estimation, thereby improving accuracy and robustness under various driving conditions.

The RL training settings were as follows: the optimizer was Adam with a learning rate of 3×10^{-4}, the discount factor was 0.99, the replay buffer size was 10^6, the batch size was 256, and the entropy target was -1. The policy network consisted of three fully connected layers with ReLU activations. The critic network included a fully connected layer, a ReLU activation layer, and an output layer. The target critic network was stabilized based on Polyak averaging. To demonstrate robustness, training was performed using three different random seeds. The solid line represents the mean return over three trials and the shaded area indicates the 95% confidence interval, as shown in Fig. 10. Convergence was observed in both datasets, with the self-collected dataset exhibiting greater fluctuations compared to the public dataset. This is primarily attributed to the similar estimation accuracy of the kinematics-based and dynamics-based estimators in the self-collected dataset, which required more training episodes to achieve optimal weight assignment.

For comparison, a fuzzy logic-based estimator fusion approach was adopted, where weight allocation was determined by vehicle lateral excitation dynamics, such as lateral acceleration, yaw rate, and steering wheel angle, more details can be found in [19]. For



(a) Self-collected dataset. (b) Public dataset.

Fig. 10. Return plots with episodes under reinforcement learning-based estimator fusion framework with soft actor–critic.

Table 5
Results of three neural network estimators for sideslip angle estimation on multiple testing sets.

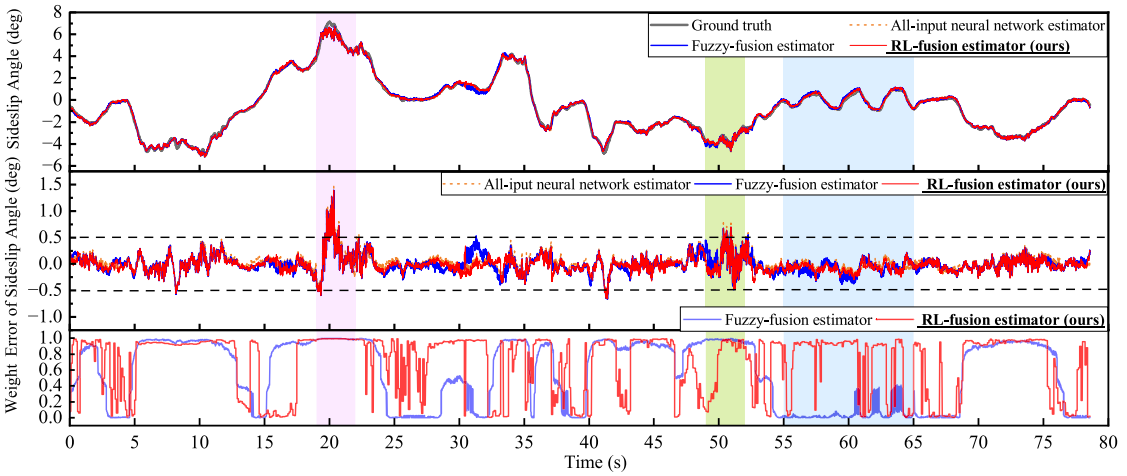
(a) Results for self-collected dataset.				
Dataset	Methods	RMSE (deg)	MAE (deg)	MaxError (deg)
Trained dataset	Kinematics-based	0.160	0.112	1.393
	Dynamics-based	0.269	0.171	1.845
Independent dataset	Kinematics-based	0.241	0.177	1.186
	Dynamics-based	0.271	0.203	0.939
(b) Results for public dataset.				
Dataset	Methods	RMSE (deg)	MAE (deg)	MaxError (deg)
Trained dataset	Kinematics-based	0.297	0.217	1.616
	Dynamics-based	0.469	0.346	1.746
Independent dataset	Kinematics-based	0.294	0.231	1.389
	Dynamics-based	0.449	0.333	1.479

simplicity, this method is referred to as the fuzzy-fusion estimator, while the proposed reinforcement learning-based estimator fusion framework with soft actor–critic is abbreviated as the RL-fusion estimator. From the perspective of input information utilization, the all-input neural network estimator incorporating $\{a_x, a_y, \dot{\phi}_z, \delta_f, v_x\}$ was also constructed for comparison.

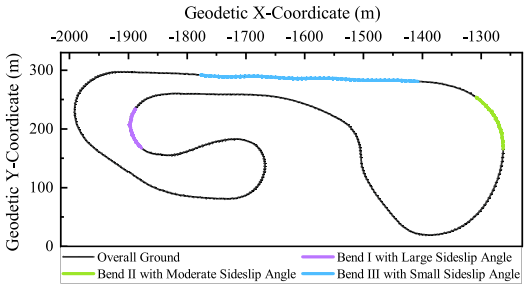
Fig. 11 shows the specific estimation results and corresponding weight allocation for the kinematics-based neural network estimator over time, along with the vehicle geodetic coordinates on both the self-collected and public datasets. For the self-collected dataset, the all-input neural network estimator, fuzzy-fusion estimator, and RL-fusion estimator all demonstrated good estimation performance under most conditions, with estimation errors within 0.5 degrees in linear dynamic conditions (blue zone in Fig. 11(a)). However, in large sideslip angle regions (purple zone), estimation accuracy decreased. The fuzzy-fusion estimator performed poorly between 30–33 s, whereas the RL-fusion estimator maintained higher accuracy. In moderate sideslip angle region (green zone), the fuzzy-fusion estimator exhibited fluctuations and failed to assign appropriate weights, while the RL-fusion estimator dynamically adjusted weights and maintained stable estimation. In terms of weight for the kinematics-based neural network estimator, although both fusion methods prioritized the kinematics-based estimator in large sideslip angle scenarios, the fuzzy-fusion estimator struggled to adapt well under moderate sideslip conditions, such as Bend II in Fig. 11(b). Thus, these observations indicate that the RL-fusion estimator is more flexible in adapting to varying driving conditions compared to the fuzzy-fusion estimator.

For the public dataset, the estimation errors under large sideslip angle conditions (purple zone in Fig. 11(d)) were similar, with both fusion methods assigning close to 1 to the kinematics-based estimator to fully perform its advantage. However, weight assignments differed in moderate and small sideslip angle conditions (green and blue zones). The smaller estimation errors of the RL-fusion estimator indicate that it achieved more appropriate weight allocations, while the fuzzy-fusion estimator was likely influenced by irrelevant information, leading to suboptimal weighting decisions. For straight driving conditions, such as during 45–60 s, the fuzzy-fusion estimator consistently assigned lower weights to the kinematics-based estimator. In comparison, the RL-fusion estimator selectively decreased the weight of the kinematics-based estimator only under specific conditions, thereby maintaining lower estimation errors.

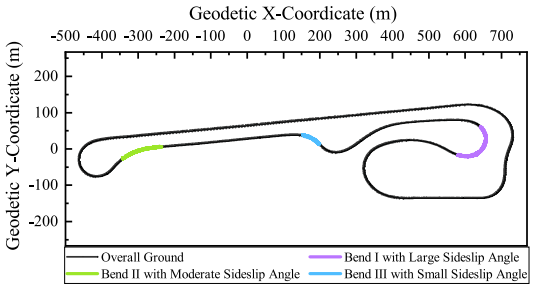
To further validate the generalization and robustness of the proposed method, the results on independent datasets from the self-collected and public datasets were compared, as shown in Figs. 12(a) and 12(b), respectively. In addition, a slalom driving condition from a self-collected real-vehicle experiment was employed for comprehensive verification. This is referred to as the extended dataset, with the results shown in Fig. 12(c). The evaluation metrics for these datasets are summarized in Table 6.



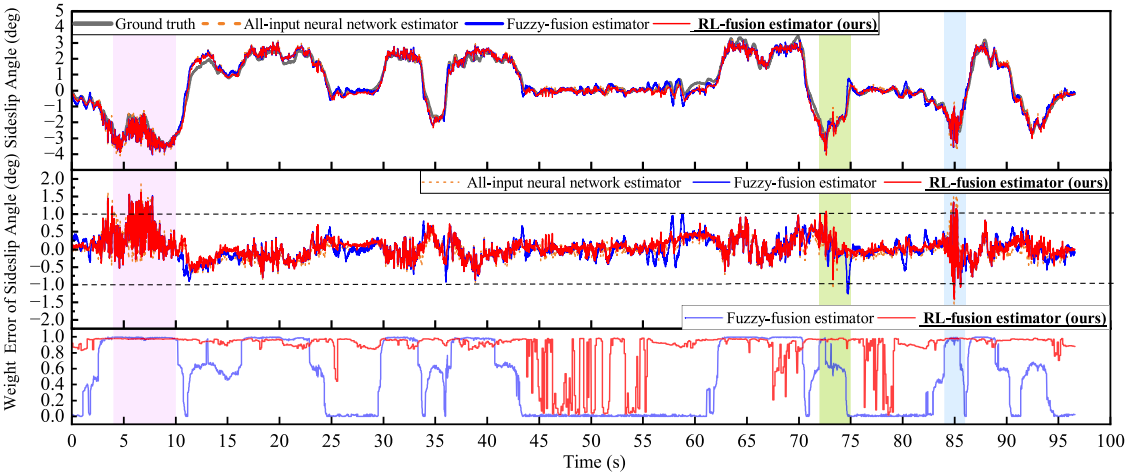
(a) Estimation comparison with weight for kinematics-based estimator from trained dataset in self-collected dataset.



(b) Geodetic coordinates of self-collected dataset.



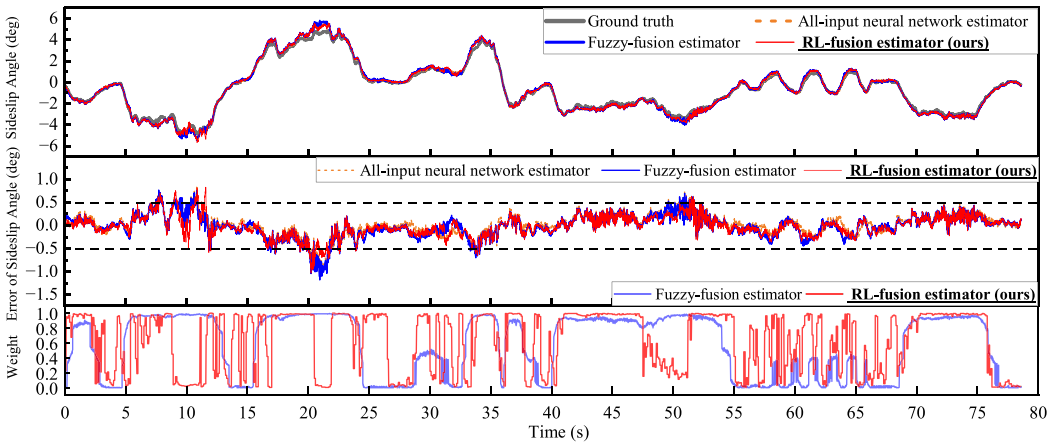
(c) Geodetic coordinates of public dataset.



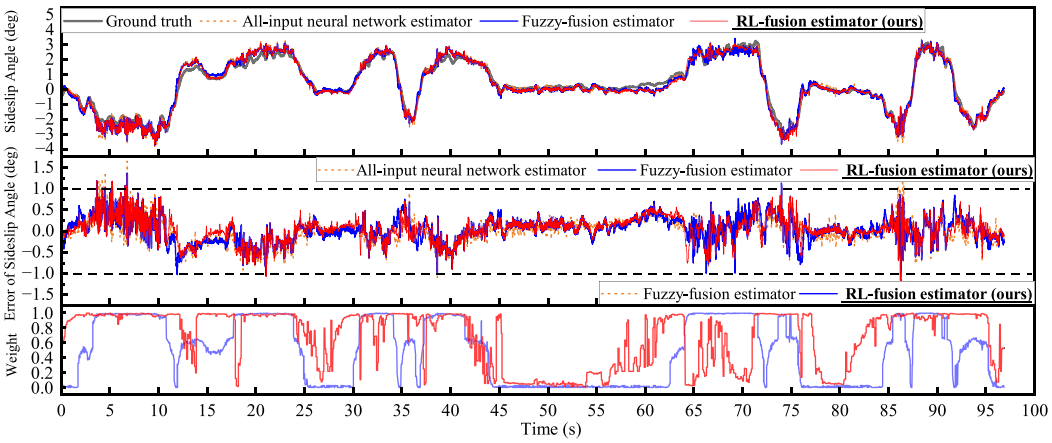
(d) Estimation comparison with weight for kinematics-based estimator from trained dataset in public dataset.

Fig. 11. Comparison of testing set with weight and geodetic coordinates for sideslip angle estimation from trained dataset.

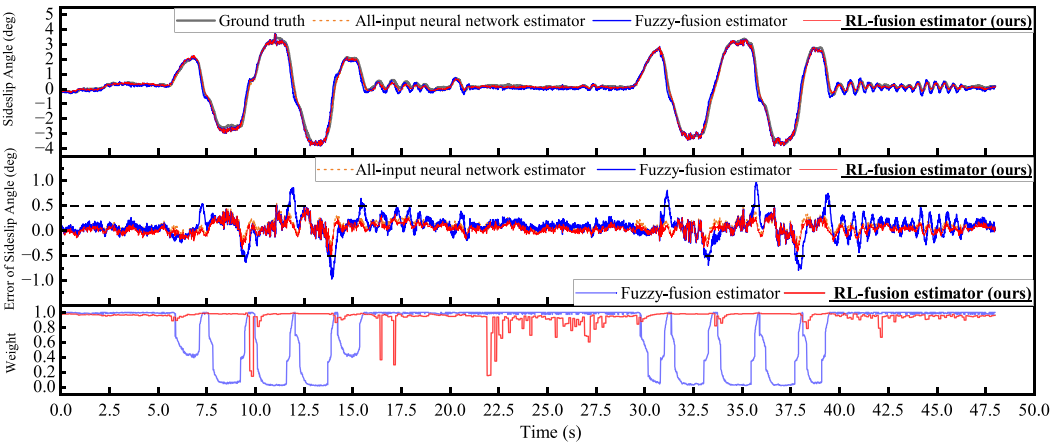
For the self-collected dataset, it is observed that the RL-fusion estimator reduced the RMSE of the estimation error by 3.14%, 5.60%, and 16.55% compared to the all-input estimator on the trained, independent, and extended datasets, respectively. Moreover, the RL-fusion estimator exhibited the smallest maximum estimation error among these three estimators across all datasets. In contrast, the fuzzy-fusion estimator had the largest RMSE among the three datasets, which was 7.23%, 9.50% and 47.27% larger than the proposed RL-fusion method, respectively, indicating inferior estimation performance. For the public dataset, although the



(a) Independent dataset of self-collected dataset.



(b) Independent dataset of public dataset.



(c) Extended dataset of self-collected dataset.

Fig. 12. Estimation comparison with weight for kinematics-based estimator under multiple testing sets.

RMSE of the RL-fusion estimator was not significantly smaller than that of the all-input estimator, its maximum estimation error was substantially decreased, with reductions of 13.12% and 17.33% in the trained and independent datasets, respectively. While the maximum estimation error of the fuzzy-fusion estimator was approximately the same as that of the RL-fusion estimator, its RMSE

Table 6
Results of different fusion models for sideslip angle estimation.

(a) Self-collected dataset.

Datasets	Methods	RMSE (deg)	Relative RMSE (%)	MAE (deg)	Relative MAE (%)	MaxError (deg)	Relative MaxError (%)
Trained dataset	All-input estimator	0.159	−3.14	0.105	0.95	1.464	−4.78
	Fuzzy-fusion estimator	0.166	−7.23	0.118	−10.17	1.394	0.00
	RL-fusion estimator	0.154	/	0.106	/	1.394	/
Independent dataset	All-input estimator	0.232	−5.60	0.170	−2.94	1.078	−4.36
	Fuzzy-fusion estimator	0.242	−9.50	0.179	−7.82	1.182	−12.77
	RL-fusion estimator	0.219	/	0.165	/	1.031	/
Extended dataset	All-input estimator	0.139	−16.55	0.112	−23.21	0.561	−9.98
	Fuzzy-fusion estimator	0.220	−47.27	0.159	−45.91	0.974	−48.15
	RL-fusion estimator	0.116	/	0.086	/	0.505	/

(b) Public dataset.

Datasets	Methods	RMSE (deg)	Relative RMSE (%)	MAE (deg)	Relative MAE (%)	MaxError (deg)	Relative MaxError (%)
Trained dataset	All-input estimator	0.296	−0.34	0.210	1.90	1.851	−13.13
	Fuzzy-fusion estimator	0.327	−9.79	0.244	−12.30	1.614	−0.37
	RL-fusion estimator	0.295	/	0.214	/	1.608	/
Independent dataset	All-input estimator	0.287	−0.35	0.217	0.46	1.650	−17.33
	Fuzzy-fusion estimator	0.300	−4.67	0.233	−6.44	1.371	−0.51
	RL-fusion estimator	0.286	/	0.218	/	1.364	/

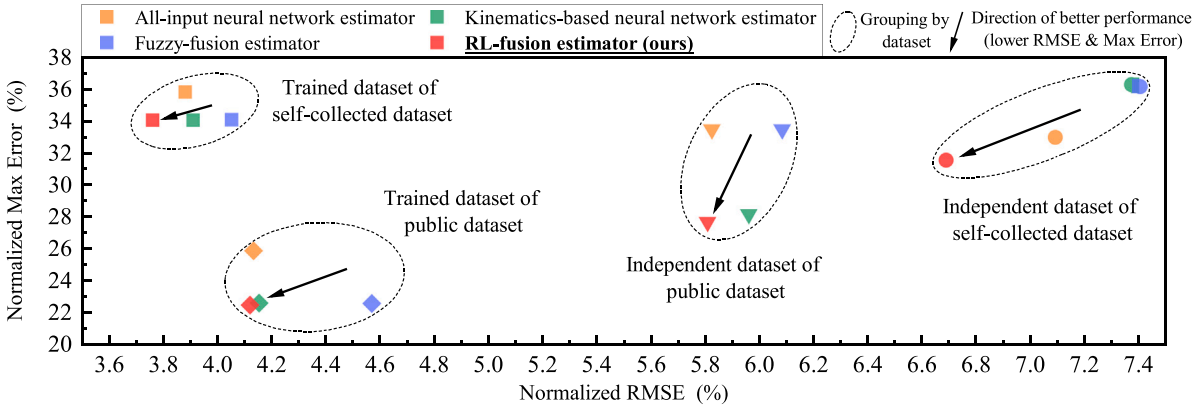


Fig. 13. Comprehensive relative comparison of different estimators on self-collected and public datasets.

was larger, being 9.79% and 4.67% higher than that of the proposed RL-fusion estimator on the trained and independent datasets, respectively.

Fig. 13 presents the comprehensive relative estimation results using normalized RMSE and maximum error across multiple datasets. The normalized RMSE is defined as the ratio of the absolute RMSE to the maximum absolute value of the sideslip angle, and the normalized maximum error follows the same definition. The arrows indicate the direction of improved performance (lower RMSE and Max Error), and the dashed ellipses group results by dataset for comparison. From the previous results, it is evident that the overall estimation accuracy of the dynamics-based estimator is poor, so this method is not included in the figure for consistency in presentation. It is clear that the proposed RL-fusion estimator demonstrates the best estimation performance, achieving the smallest normalized RMSE and the lowest maximum error among the four estimators under transient maneuvers and steady-state conditions. It effectively integrates kinematics-based and dynamics-based estimators, dynamically adjusting weights based on driving conditions to ensure high accuracy and stability under various driving conditions.

To evaluate the practical applicability of the proposed method, its compatibility with production-level ECUs was examined. The proposed estimator relies solely on standard on-board signals, which are accessible via common vehicle protocols. On a desktop CPU, it achieves an inference time of approximately 4.47 ms per step, averaged over ten independent trials across five datasets. The proposed estimator also demonstrated stable performance across repeated trials, confirming its strong potential for deployment in production vehicles.

In summary, the above results demonstrate the excellent performance of GRU-based neural network with self-attention in sideslip angle estimation. The reinforcement learning-based estimator fusion framework with soft actor–critic provides accurate and stable sideslip angle estimation on different real-vehicle platforms under diverse driving conditions. The generalization and robustness of

the proposed fusion framework are validated on both self-collected and public real-vehicle datasets. The engineering significance of the proposed estimator's high potential for practical deployment in production vehicles is analyzed.

4. Conclusions

This paper proposes a reinforcement learning-based fusion framework with soft actor–critic for vehicle sideslip angle estimation, integrating dynamically and kinematically guided neural network estimators to improve estimation accuracy and robustness under diverse driving conditions across different vehicle configurations. The main contributions of this paper are summarized as follows:

(1) A novel physically guided neural network estimator is proposed for vehicle sideslip angle estimation. The physical knowledge of vehicle dynamics and kinematics guides the input configurations of the neural network estimator. The architecture of the GRU-based neural network estimator with self-attention excels in capturing both transient and long-term dependencies.
(2) A new reinforcement learning-based estimator fusion framework with soft actor–critic is proposed, along with a sound mathematical implementation. The estimator fusion process is formulated as a Markov decision process, considering the state transition's independence from agent actions, thereby extending reinforcement learning to estimation tasks beyond control problems.
(3) An innovative vehicle sideslip angle estimator within the reinforcement learning-based fusion framework, integrating dynamically and kinematically guided neural network estimators, is presented. The accuracy and robustness of sideslip angle estimation, as well as the generalizability and adaptability of the reinforcement learning-based estimator fusion framework, were validated on self-collected and public real vehicle datasets under normal and extreme driving conditions . The high potential deployability of the proposed estimator in production vehicles is also discussed.

This paper has thoroughly discussed vehicle sideslip angle estimation and the estimator fusion method in general applications, while future research could focus on real vehicle experimental validation on wet and icy road conditions and the accurate estimation of tire forces. Additionally, it is essential to validate the effectiveness of the proposed reinforcement learning-based estimator fusion framework with soft actor–critic in broader critical fields, such as industrial instrumentation and aerospace. Moreover, we also plan to reduce the complexity of the proposed estimator, including optimizing the tuning process and pruning, to facilitate its deployment in real-world embedded systems in production vehicles.

CRediT authorship contribution statement

Chaofan Gong: Writing – original draft, Methodology, Software. **Yan Kong:** Writing – review & editing, Methodology. **Dong Zhang:** Funding acquisition, Methodology, Writing – review & editing, Project administration. **Yao Ma:** Investigation, Methodology, Writing – review & editing. **Bo Lu:** Investigation, Project administration. **Shuyong Xing:** Investigation. **Changfu Zong:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Development of tire models considering temperature and road conditions (Grant No. 12893) and the China Scholarship Council program (Grant No. 202306170146).

Appendix. Detailed formulations for RL-based estimator fusion framework with SAC

The soft Bellman operator $\mathcal{T}^{\pi}$ in RL-based fusion estimation framework is defined as:

$$\mathcal{T}^{\pi}Q(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}\sim P_p(\cdot|s_t)}\left[V(s_{t+1})\right], \tag{41}$$

where

$$V(s_{t+1}) = \mathbb{E}_{a_{t+1}\sim\pi(\cdot|s_{t+1})}\left[Q(s_{t+1}, a_{t+1}) - \alpha \log \pi(a_{t+1} \mid s_{t+1})\right]. \tag{42}$$

This operator is a contraction under a suitable norm due to the discount factor $\gamma < 1$. As a result, the fixed-point iteration $Q_{k+1} = \mathcal{T}^{\pi}Q_k$ converges to the unique fixed point Q^{π}, representing the soft Q-function of the policy π.

The soft Q-function quantifies the expected return and is represented using a neural network in practice, with Q_θ parameterized by θ.

The soft Bellman backup target y is defined as:

$$y = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}\sim P_p(\cdot|s_t)}\left[V_{\hat{\theta}}(s_{t+1})\right], \tag{43}$$

where

$$V_{\bar{\theta}}(s_{t+1}) = \mathbb{E}_{a_{t+1}\sim\pi_\phi(\cdot|s_{t+1})}\left[Q_{\bar{\theta}}(s_{t+1}, a_{t+1}) - \alpha \log \pi_\phi(a_{t+1} \mid s_{t+1})\right]. \tag{44}$$

The policy update in SAC is guided by minimizing the Kullback–Leibler (KL) divergence between the current policy π_ϕ and the Boltzmann distribution π_B; the latter is described as the target action probabilities for a given state s_t:

$$\pi_B(a \mid s_t) = \frac{\exp\left(\frac{1}{\alpha} Q_\pi(s_t, a)\right)}{Z_\pi(s_t)}, \tag{45}$$

where

$$Z_\pi(s_t) = \int \exp\left(\frac{1}{\alpha} Q_\pi(s_t, a)\right) da. \tag{46}$$

The policy objective based on minimizing the KL divergence is expressed as:

$$\begin{aligned}
\pi_{\text{new}} &= \arg\min_{\pi'\in\Pi} \mathbb{D}_{\text{KL}}\left(\pi'(\cdot \mid s_t) \,\middle\|\, \pi_B(a \mid s_t)\right) \\
&= \arg\min_{\pi'\in\Pi} \mathbb{E}_{a_t\sim\pi'(\cdot|s_t)}\left[\log \pi'(a_t \mid s_t) - \left(\frac{1}{\alpha} Q_\pi(s_t, a_t) - \log Z_\pi(s_t)\right)\right] \\
&= \arg\min_{\pi'\in\Pi} \mathbb{E}_{a_t\sim\pi'(\cdot|s_t)}\left[\log \pi'(a_t \mid s_t) - \frac{1}{\alpha} Q_\pi(s_t, a_t)\right],
\end{aligned} \tag{47}$$

where the constant term $\log Z_\pi(s_t)$ is independent of π' and can be ignored during optimization, π_{new} represents the optimized policy, π' denotes a candidate policy within the policy space Π.

The gradient of $J_\pi(\varphi)$ can be derived through the chain rule, as shown below:

$$\begin{aligned}
\hat{\nabla}_\varphi J_\pi(\varphi) &= \mathbb{E}_{s_t\sim D, \epsilon_t\sim\mathcal{N}}\left[\nabla_\varphi\left(\alpha \log \pi_\varphi(f_\varphi(\epsilon_t; s_t) \mid s_t)\right) - \nabla_\varphi Q_\theta(s_t, f_\varphi(\epsilon_t; s_t))\right] \\
&= \mathbb{E}_{s_t\sim D, \epsilon_t\sim\mathcal{N}}\left[\nabla_\varphi\left(\alpha \log \pi_\varphi(f_\varphi(\epsilon_t; s_t) \mid s_t)\right) - \left(\nabla_{a_t} Q_\theta(s_t, a_t) \nabla_\varphi f_\phi(\epsilon_t; s_t)\right)\right] \\
&= \mathbb{E}_{s_t\sim D, \epsilon_t\sim\mathcal{N}}\left[\nabla_\varphi\left(\alpha \log \pi_\varphi(f_\varphi(\epsilon_t; s_t) \mid s_t)\right) + \left(\nabla_{a_t} \alpha \log \pi_\varphi(a_t \mid s_t) - \nabla_{a_t} Q_\theta(s_t, a_t)\right) \nabla_\varphi f_\varphi(\epsilon_t; s_t)\right].
\end{aligned} \tag{48}$$

Data availability

Data will be made available on request.

References

[1] W. Liu, M. Hua, Z. Deng, Z. Meng, Y. Huang, C. Hu, S. Song, L. Gao, C. Liu, B. Shuai, A. Khajepour, L. Xiong, X. Xia, A systematic survey of control techniques and applications in connected and automated vehicles, IEEE Internet Things J. 10 (24) (2023) 21892–21916.

[2] Q. Li, B. Zhang, H. He, Y. Wang, D. He, S. Mo, A hybrid physics-data driven approach for vehicle dynamics state estimation, Mech. Syst. Signal Process. 225 (2025).

[3] Y. Wang, F. Hu, C. Tian, P. Li, H. Huang, G. Yin, C. Huang, FTEKFNet: Hybridizing physical and data-driven estimation algorithms for vehicle state, IEEE Trans. Intell. Veh. (2024).

[4] W. Liu, X. Xia, L. Xiong, Y. Lu, L. Gao, Z. Yu, Automated vehicle sideslip angle estimation considering signal measurement characteristic, IEEE Sensors J. 21 (2021) 21675–21687.

[5] G.H. Lee, D.H. Kim, J.M. Pak, C.K. Ahn, Vehicle sideslip angle estimation using finite memory estimation and dynamics/kinematics model fusion based on neural networks, IEEE Trans. Intell. Transp. Syst. (2024).

[6] J. Liu, Z. Wang, L. Zhang, P. Walker, Sideslip angle estimation of ground vehicles: A comparative study, IET Control Theory Appl. 14 (2020) 3490–3505.

[7] D. Qi, J. Feng, W. Wan, B. Song, A novel maximum correntropy adaptive extended Kalman filter for vehicle state estimation under non-Gaussian noise, Meas. Sci. Technol. 34 (2023).

[8] D. Jeong, G. Ko, S.B. Choi, Estimation of sideslip angle and cornering stiffness of an articulated vehicle using a constrained lateral dynamics model, Mechatronics 85 (2022).

[9] Q. Chen, B. Yu, H. Pang, C. Zhong, D. You, Z. Jiang, Distributed drive electric vehicle sideslip angle estimation based on the AVOA-MCSCKF algorithm, Proc. Inst. Mech. Eng. Part E: J. Process. Mech. Eng. (2024).

[10] Y. Wang, K. Geng, L. Xu, Y. Ren, H. Dong, G. Yin, Estimation of sideslip angle and tire cornering stiffness using fuzzy adaptive robust Cubature Kalman filter, IEEE Trans. Syst. Man Cybern.: Syst. 52 (2022) 1451–1462.

[11] D. Selmanaj, M. Corno, G. Panzani, S.M. Savaresi, Vehicle sideslip estimation: A kinematic based approach, Control Eng. Pract. 67 (2017) 1–12.

[12] X. Xia, L. Xiong, Y. Lu, L. Gao, Z. Yu, Vehicle sideslip angle estimation by fusing inertial measurement unit and global navigation satellite system with heading alignment, Mech. Syst. Signal Process. 150 (2021).

[13] X. Xia, E. Hashemi, L. Xiong, A. Khajepour, N. Xu, Autonomous vehicles sideslip angle estimation: Single antenna GNSS/IMU fusion with observability analysis, IEEE Internet Things J. 8 (2021) 14845–14859.

[14] X. Ding, Z. Wang, L. Zhang, Event-triggered vehicle sideslip angle estimation based on low-cost sensors, IEEE Trans. Ind. Informat. 18 (2022) 4466–4476.

[15] G. Park, Vehicle sideslip angle estimation based on interacting multiple model Kalman filter using low-cost sensor fusion, IEEE Trans. Veh. Technol. 71 (2022) 6088–6099.

[16] S. Zhong, Y. Zhao, L. Ge, Z. Shan, F. Ma, Vehicle state and bias estimation based on unscented Kalman filter with vehicle hybrid kinematics and dynamics models, Automot. Innov. 6 (2023) 571–585.

[17] F. Wang, M. Zhao, T. Shen, G. Cai, J. Liang, Y. Wang, G. Yin, A robust adaptive fault-tolerant estimator for sideslip angle and tire cornering stiffness with multiple missing data, IEEE/ASME Trans. Mechatronics (2024).

[18] L. Gao, Q. Wu, Y. He, K. Wu, P. Shao, Robust estimation of sideslip angle for heavy-duty vehicles under payload conditions using a series-connected structure estimator, IEEE Trans. Intell. Veh. (2024).
[19] X. Xia, P. Hang, N. Xu, Y. Huang, L. Xiong, Z. Yu, Advancing estimation accuracy of sideslip angle by fusing vehicle kinematics and dynamics information with fuzzy logic, IEEE Trans. Veh. Technol. 70 (2021) 6577–6590.
[20] T. Chen, Y. Cai, L. Chen, X. Xu, Sideslip angle fusion estimation method of three-axis autonomous vehicle based on composite model and adaptive Cubature Kalman filter, IEEE Trans. Transp. Electrification 10 (2024) 316–330.
[21] Y. Sun, Y. Pan, I. Kawsar, G. Wang, L. Hou, Combined recurrent neural networks and particle-swarm optimization for sideslip-angle estimation based on a vehicle multibody dynamics model, Multibody Syst. Dyn. (2024).
[22] D. Pölzleitner, J. Ruggaber, J. Brembeck, Feature and extrapolation aware uncertainty quantification for AI-based state estimation in automated driving, in: IEEE Intelligent Vehicles Symposium, Proceedings, Institute of Electrical and Electronics Engineers Inc. 2024, pp. 2756–2762.
[23] G. Novotny, Y. Liu, W. Morales-Alvarez, W. Wöber, C. Olaverri-Monreal, Vehicle side-slip angle estimation under snowy conditions using machine learning, Integr. Comput.-Aided Eng. 31 (2024) 117–137.
[24] B. Yang, R. Fu, Q. Sun, S. Jiang, C. Wang, State estimation of buses: A hybrid algorithm of deep neural network and unscented Kalman filter considering mass identification, Mech. Syst. Signal Process. 213 (2024).
[25] Y. Zhang, Y. Huang, K. Deng, B. Shi, X. Wang, L. Li, J. Song, Vehicle dynamics estimator utilizing LSTM-ensembled adaptive Kalman filter, IEEE Trans. Ind. Electron. (2024).
[26] A. Bertipaglia, M. Alirezaei, R. Happee, B. Shyrokau, An unscented Kalman filter-informed neural network for vehicle sideslip angle estimation, IEEE Trans. Veh. Technol. (2024).
[27] M.D. Lio, M. Piccinini, F. Biral, Robust and sample-efficient estimation of vehicle lateral velocity using neural networks with explainable structure informed by kinematic principles, IEEE Trans. Intell. Transp. Syst. 24 (2023) 13670–13684.
[28] T. Lin, Z. Ren, L. Zhu, Y. Zhu, K. Feng, W. Ding, K. Yan, M. Beer, A systematic review of multi-sensor information fusion for equipment fault diagnosis, IEEE Trans. Instrum. Meas. (2025) 1–1.
[29] X. Zhang, C. Wang, W. Zhou, J. Xu, T. Han, Trustworthy diagnostics with out-of-distribution detection: A novel max-consistency and min-similarity guided deep ensembles for uncertainty estimation, IEEE Internet Things J. 11 (2024) 23055–23067.
[30] L. Kamyabi, T.T. Lie, S. Madanian, S. Marshall, A comprehensive review of hybrid state estimation in power systems: Challenges, opportunities and prospects, Energies 17 (2024) 4806.
[31] T. Alsuwian, S. Ansari, M.A.A.M. Zainuri, A. Ayob, A. Hussain, M.H. Lipu, A.R. Alhawari, A. Almawgani, S. Almasabi, A.T. Hindi, A review of expert hybrid and co-estimation techniques for SOH and rul estimation in battery management system with electric vehicle application, Expert Syst. Appl. 246 (2024) 123123.
[32] M. Lin, S. Chen, J. Meng, W. Wang, J. Wu, Instantaneous energy consumption estimation for electric buses with a multi-model fusion method, IEEE Trans. Intell. Transp. Syst. 26 (2025) 371–381.
[33] Z. Wei, Z. Duan, U.D. Hanebeck, Distributed fusion of multiple model estimators using minimum forward Kullback–Leibler divergence sum, IEEE Trans. Aerosp. Electron. Syst. 60 (2024) 2934–2947.
[34] M.S.H. Lipu, M.S.A. Rahman, M. Mansor, S. Ansari, S.T. Meraj, M.A. Hannan, Hybrid and combined states estimation approaches for lithium-ion battery management system: Advancement, challenges and future directions, J. Energy Storage 92 (2024) 112107.
[35] G. Xu, Y. Qiao, X. Chen, T. Peng, C. Zhao, Enhanced vehicle sideslip angle estimation through multi-source information fusion, in: 2023 7th CAA International Conference on Vehicular Control and Intelligence, CVCI, IEEE, 2023, pp. 1–6.
[36] R. Song, Y. Fang, Vehicle state estimation for INS/GPS aided by sensors fusion and SCKF-based algorithm, Mech. Syst. Signal Process. 150 (2021).
[37] R. Wang, B. Chen, Z. Hu, L. Yu, Distributed event-triggered nonlinear fusion estimation under resource constraints, IEEE Trans. Aerosp. Electron. Syst. 59 (2023) 1–35.
[38] L. Li, D. Zhao, Y. Xia, Indoor positioning systems based on a modified matrix-weighted fusion estimator with multipath and NLOS mitigation, IEEE Internet Things J. 11 (2024) 40041–40050.
[39] Y. Zhang, F. Liu, Z. Lu, Y. Wei, H. Wang, Multi-anemometer optimal layout and weighted fusion method for estimation of ship surface steady-state wind parameters, Ocean Eng. 266 (2022) 112793.
[40] N. He, C. Qian, C. Shen, Y. Huangfu, A fusion framework for lithium-ion batteries state of health estimation using compressed sensing and entropy weight method, ISA Trans. 135 (2023) 585–604.
[41] F. Liu, D. Yu, W. Su, F. Bu, Multi-state joint estimation of series battery pack based on multi-model fusion, Electrochim. Acta 443 (2023) 141964.
[42] Y.J. Ma, X.L. Zhou, M.P. Ran, Estimation of sideslip angle based on the combination of dynamic and kinematic methods, Int. J. Automot. Technol. (2024).
[43] T. Chen, Y. Cai, L. Chen, X. Xu, Sideslip angle fusion estimation method of three-axis autonomous vehicle based on composite model and adaptive Cubature Kalman filter, IEEE Trans. Transp. Electrification 10 (2024) 316–330.
[44] H. Hu, P. Wang, F. Xin, L. Li, Failure probability function estimation in augmented sample space combined active learning kriging and adaptive sampling by voronoi cells, Mech. Syst. Signal Process. 206 (2024) 110897.
[45] G.H. Lee, D.-H. Kim, J.M. Pak, C.K. Ahn, Vehicle sideslip angle estimation using finite memory estimation and dynamics/kinematics model fusion based on neural networks, IEEE Trans. Intell. Transp. Syst. 26 (2025) 2157–2168.
[46] G. Sharma, A. Singh, S. Jain, DeepEvap: Deep reinforcement learning based ensemble approach for estimating reference evapotranspiration, Appl. Soft Comput. 125 (2022) 109113.
[47] C.-C. Wong, H.-M. Feng, K.-L. Kuo, Multi-sensor fusion simultaneous localization mapping based on deep reinforcement learning and multi-model adaptive estimation, Sensors 24 (2023) 48.
[48] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv preprint.
[49] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, arXiv preprint.
[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017, arXiv preprint.
[51] M.V. Otterlo, Markov Decision Processes: Concepts and Algorithms, Tech. rep., Springer, 2012.
[52] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018, arXiv preprint.
[53] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, S. Levine, Soft actor-critic algorithms and applications, 2018, arXiv preprint.
[54] Applus IDIADA, Test tracks China, 2025.
[55] J. C., L.K. Harbott, J.C.K. Gerdes, 2014 targa sixty-six. Stanford digital repository, 2016.
[56] J.C. Kegelman, L.K. Harbott, J.C. Gerdes, Insights into vehicle trajectories at the handling limits: analysing open data from race car drivers, Veh. Syst. Dyn. 55 (2017) 191–207.
[57] J. Heer, Fast & accurate Gaussian kernel density estimation, in: 2021 IEEE Visualization Conference, VIS, 2021, pp. 11–15.
[58] A. Leanza, G. Reina, J.L. Blanco-Claraco, A factor-graph-based approach to vehicle sideslip angle estimation, Sensors 21 (2021).
[59] M. Tristano, B. Lenzo, A parametric interpolation-based approach to sideslip angle estimation, in: Lecture Notes in Mechanical Engineering, Springer Science and Business Media Deutschland GmbH, 2024, pp. 279–285.
[60] D. Chicco, M.J. Warrens, G. Jurman, The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation, PeerJ Comput. Sci. 7 (2021) 1–24.