IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 August 2025, accepted 15 September 2025,
date of publication 18 September 2025, date of current version 30 September 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3611825

RESEARCH ARTICLE

Real-Time Personalized Car-Following Control With Online Parameter Adaptation for Intelligent Regenerative Braking Systems

SEUNGYEON OAK[1], DAEKYEONG LEE[1], GYUBIN SIM[2], SOOYOUNG KIM[1,3], AND GISEO PARK[4]

[1]Department of Mechanical Systems Engineering, Sookmyung Women's University, Seoul 04312, Republic of Korea
[2]Hyundai Motor Company, Hwaseong 18280, Republic of Korea
[3]Institute of Advanced Materials and Systems, Sookmyung Women's University, Seoul 04312, Republic of Korea
[4]School of Mechanical Engineering, University of Ulsan, Ulsan 44610, Republic of Korea

Corresponding authors: Sooyoung Kim (syk@sookmyung.ac.kr) and Giseo Park (eloq123@ulsan.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Ministry of Science and ICT (MSIT) under Grant RS-2023-00279679 and Grant RS-2022-NR066631; and in part by the Regional Innovation System and Education (RISE) Program through Ulsan RISE Center, funded by the Ministry of Education (MOE) and Ulsan Metropolitan City, Republic of Korea, under Grant 2025-RISE-07-001.

ABSTRACT With the increasing integration of advanced driver assistance systems (ADAS) in modern vehicles, intelligent regenerative braking systems (IRBS) have emerged as an effective solution for energy-efficient car-following control. However, current IRBS implementations lack personalization and require drivers to select fixed braking levels, often causing discomfort by failing to reflect individual preferences. Existing personalization techniques rely on deep learning or offline-trained models, which are computationally intensive and unsuitable for real-time deployment on vehicle electronic control units (ECUs). To address these limitations, this study proposes a computationally efficient parameter adaptation algorithm based on the extended Kalman filter (EKF), enabling real-time personalized car-following control in IRBS. The proposed method adopts a dynamic relative speed (DRS) spacing policy to capture individual driving characteristics. A reference data generator generates real-time reference data during specific driver interventions, allowing the EKF algorithm to adapt control parameters accordingly. Simulation tests using a driver model demonstrated a significant reduction in driver interventions and maintained inter-vehicle distances aligned with the driver's preferred distance. Vehicle tests further confirmed the algorithm's real-time performance and low computational load, affirming its suitability for large-scale production. This study presents a practical and scalable method for personalizing IRBS-based car-following control without complex learning architectures, thereby enhancing driving comfort and expanding applicability to ADAS technologies such as adaptive cruise control (ACC).

INDEX TERMS Advanced driver assistance systems (ADAS), extended Kalman filter, spacing strategy, intelligent regenerative braking system (IRBS), personalized car-following control.

I. INTRODUCTION

With advancements in autonomous driving technology, advanced driver assistance systems (ADAS) have become fundamental components of modern vehicles [1], [2], [3]. ADAS includes multiple features to improve driving convenience and safety, such as adaptive cruise control (ACC) which enables car-following control through a spacing strategy, in contrast to conventional cruise control (CC) [4], [5], [6].

Advancements in vehicle electrification technology have given rise to intelligent regenerative braking systems (IRBS),

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

which facilitate car-following control using regenerative braking to manage vehicle spacing and deceleration [7]. However, existing IRBS implementations necessitate drivers to manually choose preset regenerative braking levels and do not autonomously adapt to individual driving behaviors or varying driving conditions. Consequently, drivers may feel discomfort during deceleration in car-following control while using the IRBS, since current production vehicles lack personalization features to mitigate this issue. To address this limitation, we introduce a personalized car-following control algorithm utilizing an extended Kalman filter (EKF) - based parameter adaptation framework. By dynamically adjusting parameters in real time, our method optimizes deceleration responses and enhances driving comfort. This algorithm advances conventional IRBS-based car-following control by integrating individual driving characteristics.

In recent years, extensive research has been conducted into personalization techniques for improving driving comfort, especially through various deep learning-based methods. Studies have utilized clustering techniques on driving data to categorize drivers into three distinct driving styles [8], [9]. In [9], researchers explored the personalization of ACC. However, the control parameters were not adjusted through real-time learning. Vehicle control relied on predefined target values for each classified driving style, which limited the ability of the system to adapt to individual driving behaviors.

References [10], [11] employed machine learning techniques, such as reinforcement learning (RL) to not only classify driving styles but also to learn and integrate individual driving patterns into vehicle control. Nevertheless, these RL-based models are computationally demanding and require extensive training time. Similar to [8] and [9], they face challenges in real-time learning and parameter adaptation, complicating implementation and validation in embedded vehicle systems. In [11], the model was tested in a controlled environment using a vehicle; however, validation was conducted with pre-trained offline models rather than an online learning approach, thereby limiting real-time adaptation and validation.

Studies [12], [13] integrated deep deterministic policy gradient (DDPG) with deep learning models to enhance the predictive performance of RL-based models and reduce learning time. However, this approach increases model complexity and demands a large volume of high-quality driving data encompassing diverse driving conditions. Moreover, the validation was primarily conducted in simulation environments, and although real-world driving data were utilized for learning and validation, the evaluation was not performed in an embedded vehicle system. Consequently, assessing whether the proposed method has undergone adequate vehicle testing to evaluate its feasibility for mass production is problematic. In contrast, the method proposed in this study achieves real-time adaptation with low computational cost, and has been validated in both simulation and real-world vehicle testing, making it more suitable for embedded system deployment than existing deep learning and RL-based approaches [10], [11], [12], [13].

In light of these limitations, researchers have recently focused on lowering computational costs and developing lightweight models for efficient integration into vehicle systems. However, these lightweight models frequently exhibit reduced performance compared to traditional models and still demand substantial CPU resources. Therefore, even if lightweight deep learning models are successfully integrated into vehicle controllers, significant obstacles persist in their incorporation into electronic control unit (ECU)-based vehicle control systems.

Beyond deep-learning based approaches, studies have investigated personalized car-following control to improve driving comfort [14], while others have examined modeling and control strategies for personalized braking systems that account for individual driving behaviors [15], [16]. In [15], a dynamic relative speed (DRS) spacing policy with fixed parameters was proposed and later expanded in [16] to implement a parameter adaptation algorithm using model predictive control (MPC). Nonetheless, the algorithm presented in [16] is computationally intensive, making real-time parameter adaptation challenging, and it has not been validated through vehicle testing.

The present study builds on the DRS spacing policy proposed in [15] and explores beyond simply classifying driving styles by enabling real-time parameter adaptation that reflects individual driving behavior without relying on complex learning models such as deep learning. This approach allows for a straightforward and transparent calibration process. In contrast, deep-learning and machine-learning models require retraining the entire model when performance issues emerge, making calibration difficult. Moreover, vehicle ECUs typically have limited memory and computational power, rendering them unsuitable for real- time learning in these models. The algorithm proposed in this study is computationally efficient and suitable for real- time implementation, enabling deployment on a vehicle ECU, where its performance has been successfully validated. Therefore, unlike existing deep learning, reinforcement learning, or MPC-based approaches, the proposed EKF-based framework enables real-time parameter adaptation with low computational cost and has been validated on an embedded ECU in a real vehicle, demonstrating its practicality for large-scale deployment.

The contributions of this paper are listed as follows:

1) We implemented a DRS spacing policy [15] and experimentally validated its effectiveness in characterizing the car-following control situation during IRBS operation. Additionally, we analyzed driver behavior related to spacing control using driving data from various car-following scenarios. Based on this analysis, we developed a Reference Data Generator that provides real-time reference data for parameter adaptation.

2) Leveraging the validated spacing policy and reference data from the Reference Data Generator, we proposed an EKF-based parameter adaptation algorithm that enables personalized car-following control through real-time parameter adaptation reflecting individual driving characteristics.
3) Simulation-based validation, incorporating a driver model, demonstrated that the proposed algorithm effectively achieved personalization while minimizing driver intervention and discomfort during IRBS operation.
4) The proposed algorithm was embedded in a real vehicle control unit and validated through vehicle testing. The experimental results demonstrate that the algorithm functions reliably with low computational resources, attains real-time performance, and sustains stable operations, thereby validating the suitability of the algorithm for large-scale production.

The remainder of this paper is structured as follows. The DRS spacing policy utilized in this study is presented in Section II, and its effectiveness is verified in representing individual driving behaviors. The implementation of the EKF for real-time parameter adaptation to facilitate personalization is detailed in Section III. In Section VI, we discuss the simulation validation using the driver model along with CarSim - a vehicle simulation software. Thereafter, the experimental assessment of the proposed algorithm via vehicle testing is provided in Section V. Finally, the concluding remarks are summarized in Section VI.

II. IRBS SPACING POLICY

A. SPACING POLICY

The proposed algorithm identifies and adapts parameters that reflect individual driving characteristics, allowing for personalized inter-vehicle distance control during car-following scenarios. In such scenarios, each driver maintains a distinct preferred distance at which they feel comfortable. However, traditional IRBS-based inter-vehicle distance control overlooks these personal preferences, frequently resulting in driver discomfort. To overcome this limitation, an appropriate spacing policy is necessary to integrate individual driving preferences into the desired distance within the control mechanism. A widely used spacing policy in existing car-following control research is the constant time headway (CTH) spacing policy [17], which is defined using the headway time parameter, τ [18], [19], [20], [21]. The CTH spacing policy is formulated in (1), where τ represents the time headway and d_s denotes the minimum safe distance. This implies that the CTH spacing policy determines the desired distance proportional to the speed of the ego vehicle.

$$d_{des} = d_s + \tau v_{ego} \tag{1}$$

The string stability of the CTH spacing policy was previously demonstrated in [22]. In this study, we adopt the DRS spacing policy [15] over the CTH spacing policy to better integrate individual driving characteristics into car-following control. The proposed spacing policy for d_{des} is defined as follows:

$$d_{des} = d_s + \tau v_{ego} + b\Delta v^2 \tag{2}$$

where d_s represents the minimum safe distance, τ indicates the time headway, and b denotes the coefficient associated with Δv^2. Here, Δv denotes the relative speed, and d_{rel}, the relative distance corresponding to the desired distance for control, is defined as:

$$\Delta v = v_{ego} - v_{PV}, \quad d_{rel} = x_{PV} - x_{ego} \tag{3}$$

where x represents the position, v indicates the vehicle speed, and the subscripts ''ego'' and ''PV'' denote the ego vehicle and preceding vehicle, respectively.

The DRS spacing policy differs from the conventional CTH spacing policy by incorporating the relative speed term through parameter b, thereby enabling more stable control when the inter-vehicle distance decreases rapidly. Although this spacing policy was examined using driving data in [15], its ability to sufficiently reflect individual driver behaviors has not been clearly validated. Therefore, sufficient validation has not been conducted to determine whether driver behavior can be parameterized using the two parameters, τ and b. Therefore, this study aims to validate the DRS spacing policy.

B. EXPERIMENTAL VALIDATION OF THE DRS SPACING POLICY

To assess the suitability of the DRS spacing policy for the proposed algorithm, we analyzed experimental data from a test vehicle equipped with a radar system under highway driving conditions. In the analyzed scenario, the driver perceived the inter-vehicle distance maintained by the IRBS as too close, leading to manual intervention using the brake to achieve the preferred distance d_{rel}. This indicates that the driver actively controlled the inter-vehicle distance to their preferred distance rather than relying solely on the IRBS. We aimed to verify whether the d_{des} calculated using the DRS spacing policy in (2) accurately reflects the d_{rel} profile when the driver directly controls inter-vehicle distance during a car-following scenario with the IRBS. This experimental evaluation determines the extent to which the DRS spacing policy captures the driver's behavior in specific situations, thereby validating its effectiveness. The driving data were plotted and analyzed to examine the profile of the measured d_{rel}. Following the formulation of d_{des}, data analysis was conducted under the assumption that d_{rel} exhibits a quadratic relationship with Δv and a first-order linear relationship with v_{ego}. In other words, the data analysis was based on the premise that each driver exhibits a preferred distance relative to Δv and v_{ego}.

1) DATA PRE–PROCESSING

Experimental data for the inter-vehicle distance control, including relative distance (d_{rel}), relative speed (Δv), and

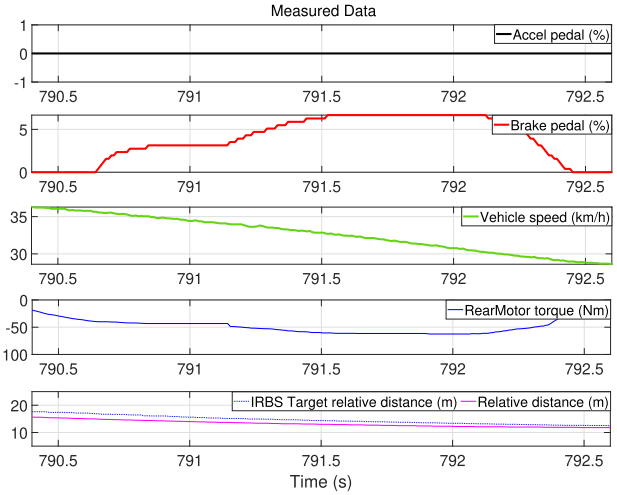

FIGURE 1. Example of measured data analysis.

ego vehicle speed (v_{ego}), are measured on the test vehicle, some of which are shown in Fig. 1. As initially assumed, we examined the profile of d_{rel} during instances in which the driver manually controlled the inter-vehicle distance during a car-following scenario with IRBS, considering both the Δv and (v_{ego}), criteria. The data preprocessing focused solely on scenarios where the driver's driving characteristics were distinctly represented in (d_{rel}), thereby facilitating the extraction of meaningful reference data that reflects the driver's preferred distance. To achieve this, relative distance, relative speed, and ego vehicle speed were employed to establish the preprocessing criteria (Fig. 1).

The preprocessing phase excluded instances where the ego vehicle speed was excessively low or the vehicle was stationary, as well as cases where no preceding vehicle was detected, making inter-vehicle distance control unfeasible. Furthermore, irrelevant situations were omitted from the analysis, including instances where the relative velocity was positive, signifying that inter-vehicle distance control was unnecessary. Consequently, we identified specific situations in which inter-vehicle distance control was initially managed solely by IRBS deceleration control. However, when the driver felt discomfort, they intervened by pressing the brake or accelerator pedals. In other words, we define the inter-vehicle distance d_{rel} in this context as d_{des}, which effectively represents the driver's driving behavior. An additional preprocessing criterion is introduced to identify situations in which both the brake stroke and acceleration remain constant, as illustrated in Fig. 2. These intervals more accurately reflect drivers' preferred inter-vehicle distances. After preprocessing the driving data, we analyzed whether the DRS spacing policy effectively incorporates the preferred distance d_{des}, thereby capturing drivers' driving characteristics.

2) VALIDATION RESULTS

The analysis results are presented in Fig. 3 and Fig. 4. Consistent with our initial hypothesis, analysis of the plotted

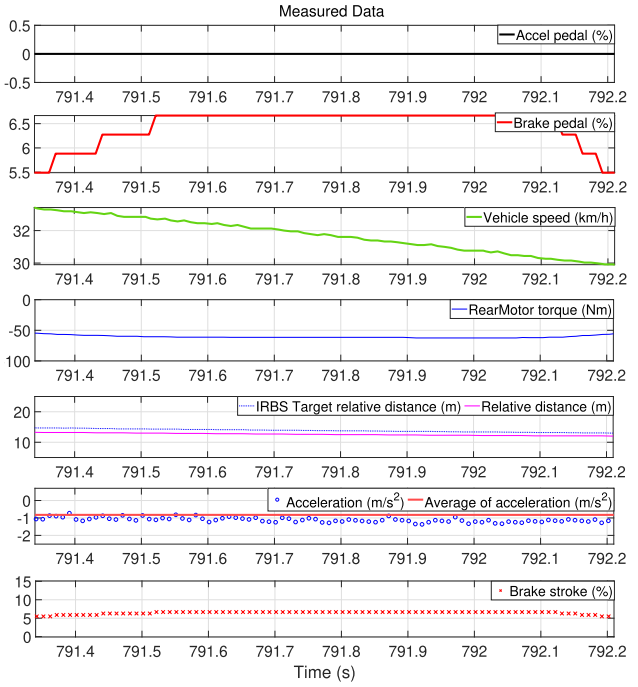

FIGURE 2. Example of a Case with Constant Brake Stroke and Acceleration.

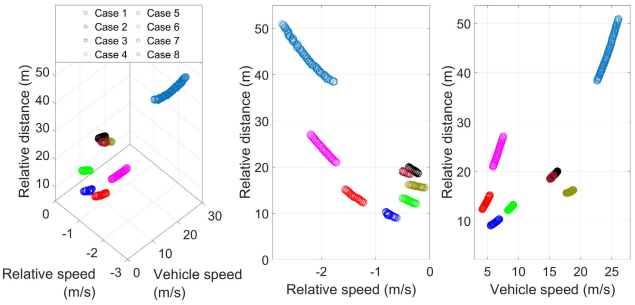

FIGURE 3. d_{rel} Profile analysis result 1.

data reveals that d_{rel} exhibits a quadratic relationship with respect to Δv and a first-order linear relationship with respect to v_{ego}. Therefore, when drivers directly control inter-vehicle distance during IRBS operations in car-following scenarios, the fitted results show that the proposed DRS spacing policy effectively describes the driver's control behavior. This verifies that the DRS spacing policy successfully represents the driver's preferred control distance. To determine whether the driver's driving behavior can be parameterized using parameters τ and b, we employed the least-squares method to identify these parameters [23]. We established the following matrix equation (4) using the previously measured driving data, specifically the values of v_{ego}, Δv and d_{des}.

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{4}$$

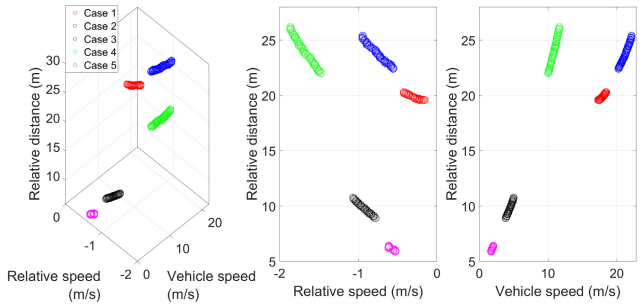

FIGURE 4. d_{rel} **Profile analysis result 2.**

where,

$$\mathbf{Y} = \begin{bmatrix} d_{des}^{(1)} \\ d_{des}^{(2)} \\ \vdots \\ d_{des}^{(n)} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} v_{ego}^{(1)} & (\Delta v^{(1)})^2 \\ v_{ego}^{(2)} & (\Delta v^{(2)})^2 \\ \vdots & \vdots \\ v_{ego}^{(n)} & (\Delta v^{(n)})^2 \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \tau \\ b \end{bmatrix} \tag{5}$$

where $\boldsymbol{Y}$ denotes the desired distance d_{des}, $\boldsymbol{X}$ indicates the matrix of independent variables (ego vehicle speed and squared relative velocity), $\boldsymbol{\epsilon}$ denotes the error term, and $\boldsymbol{\beta}$ presents the parameter vector to be estimated. The least-squares estimate of $\boldsymbol{\beta}$ is given by

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T Y \quad \text{or} \quad \begin{bmatrix} \hat{\tau} \\ \hat{b} \end{bmatrix} = (\mathbf{X^T X})^{-1} \mathbf{X^T Y} \tag{6}$$

By applying this equation to the measured data, parameters τ and b are identified for each case and presented in Table 1.

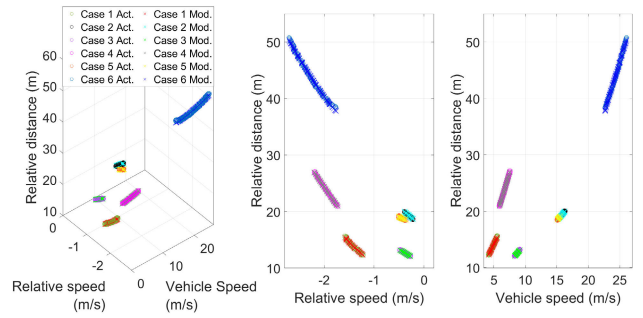

FIGURE 5. **Validation results of the DRS spacing policy.**

Using the parameter values of τ and b identified in Table 1, together with the measured data, d_{des} was calculated based on the DRS spacing policy (2). To evaluate whether this resulting d_{des} aligns with the measured relative distance d_{rel}, their profiles were compared with respect to both Δv and v_{ego}. As a result, the modeled data d_{des} —calculated using the τ and b values identified through the least-squares method—exhibited a profile consistent with the actual data d_{rel} (Fig. 5). Therefore, we validated that the DRS spacing policy effectively represents the driver's characteristics during inter-vehicle distance control, and these characteristics can be parameterized using τ and b.

TABLE 1. **Identified τ and b values.**

	Identified τ	Identified b
Case 1	1.6547	0.8855
Case 2	0.9476	3.7109
Case 3	0.9527	1.9396
Case 4	2.1712	1.3675
Case 5	2.1712	1.3675
Case 6	1.2131	2.0140

III. EKF-BASED PARAMETER ADAPTATION ALGORITHM

A. REFERENCE DATA GENERATION

In Section II-B, the DRS spacing policy was experimentally validated to effectively capture driver behavior in d_{des} through driving data analysis. This validation establishes that the d_{des} derived from the DRS spacing policy closely aligns with the driver's preferred inter-vehicle distance d_{rel}. Accordingly, the driver behavior can be parameterized using the two parameters, τ and b, and d_{des} serves as reference data. These parameters were subsequently integrated into the EKF-based parameter adaptation algorithm to personalize the inter-vehicle distance control of the IRBS.

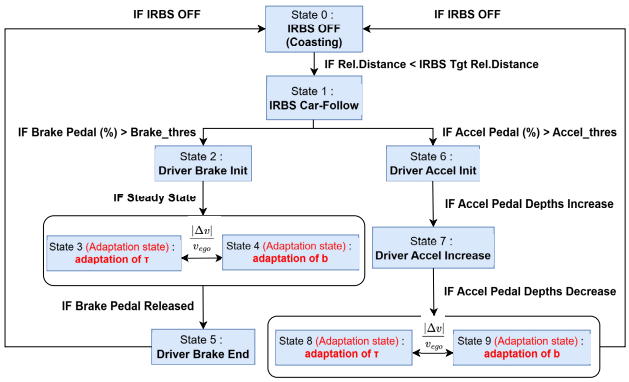

FIGURE 6. **Flowchart of the reference data generator.**

Utilizing the preprocessing criteria outlined in Section II-B2, system states were designed to determine the adaptation state of the Reference Data Generator. Parameter adaptation occurs within driving scenarios that most accurately represent driver behavior, enabling real-time generation of reference data. The reference data generator operates across nine states: State 0 denotes IRBS OFF; State 1 corresponds to IRBS-based car-following; State 2 signals the initiation of driver brake intervention; State 3 is activated when the driver's brake input remains constant, allowing adaptation of τ; and State 4 is activated under the same conditions, enabling adaptation of b. In this framework, the parameters to be adapted—τ (State 3) and b (State 4)—are determined based on $\frac{|\Delta v|}{v_{ego}}$.

When the relative velocity component is dominant compared with the ego vehicle speed, the parameter b (coefficient of the relative velocity term) is adapted. Conversely, when the effect of Δv is relatively minor, τ is adapted. The threshold for selecting adaptation parameters was empirically established

through data analysis. State 5 signifies the termination of driver-initiated brake intervention. To accommodate accelerator intervention scenarios, additional states (States 6–9) were incorporated into the Reference Data Generator. These states depict instances where the driver perceives that the IRBS maintains an inter-vehicle distance longer than preferred and manually intervenes by pressing the accelerator. For accelerator intervention instances (States 6–9), the preprocessing criteria were based on the assumption that the driver releases the accelerator upon achieving the desired inter-vehicle distance. This framework facilitates the bidirectional adaptation of τ and b, allowing them to adapt in both increasing and decreasing directions in response to driver intervention. State 6 represents the initiation of the driver's accelerator input, whereas State 7 corresponds to the phase where the input becomes more pronounced. State 8 is triggered upon the release of the accelerator pedal, permitting τ to adapt. Similarly, State 9 is activated under identical conditions, allowing the adaptation of b. Specifically, the Reference Data Generator was designed to generate reference data corresponding to driver-initiated brake inputs in States 3 and 4, and accelerator inputs in States 8 and 9. The overall structure of the reference data generator is depicted in Fig. 6.

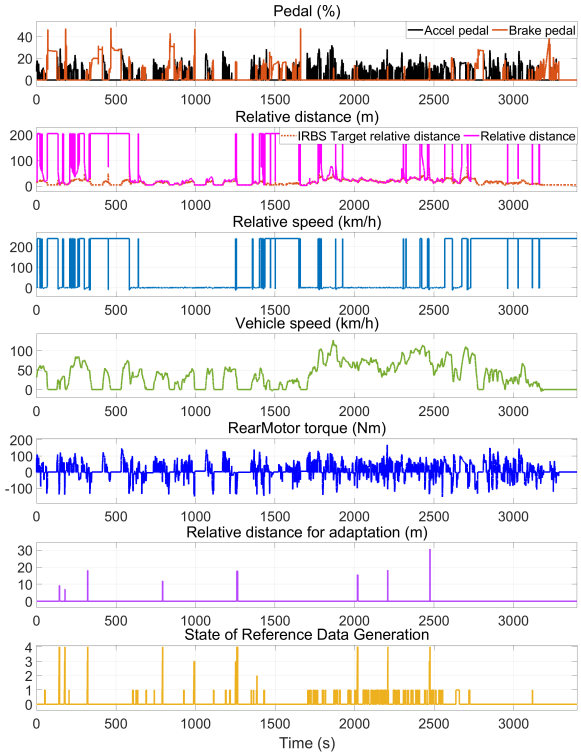

FIGURE 7. Results of reference data derivation.

Fig. 7 illustrates the output generated by the Reference Data Generator, utilizing the driving data (Δv, v_{ego}, d_{rel}) analyzed in Section II-B. In the driving data shown in Fig. 7, instances where the relative speed (Δv) or relative distance (d_{rel}) exceeds 200 indicate moments when the preceding vehicle was not detected. Since this driving data encompasses scenarios involving driver brake intervention, the Reference Data Generator is expected to reach State 3 or 4 when parameter adaptation is required. As portrayed in Fig 7, States 3 and 4 are triggered exclusively in valid brake intervention scenarios that meet the preprocessing criteria, thereby validating the intended design. Consequently, the implemented Reference Data Generator is capable of generating reference data in real-time during scenarios requiring parameter adaptation, facilitating the adaptation of parameters τ and b. The reference data generated by the proposed framework were subsequently applied to the EKF-based parameter adaptation algorithm to integrate driver behavior into the inter-vehicle distance control of the IRBS.

B. IMPLEMENTATION OF EKF-BASED ADAPTATION

Section III-B presents the implementation of a parameter adaptation algorithm. In this process, parameter adaptation utilizes the reference data generated by the Reference Data Generator implemented in Section III-A. The algorithm computes the error between this reference data d_{des} and the current relative distance d_{rel} controlled by the IRBS, and adapts the parameters accordingly to minimize the error. An Extended Kalman filter (EKF) was employed as the algorithm for parameter adaptation. The EKF-based parameter adaptation algorithm estimates and adapts the parameters (τ, b) in real time to reflect the driver's individual driving behavior, using the reference data d_{des}. The algorithm is computationally efficient and easy to calibrate due to the small number of parameters, which makes it well-suited for implementation in vehicle controllers.

1) EKF DESIGN FOR PARAMETER ADAPTATION

To implement the EKF-based parameter adaptation algorithm, a nonlinear DRS spacing policy is adopted as a model to describe the desired inter-vehicle distance. This policy is defined as follows.

$$d_{\text{des}} = d_s + \tau v_{\text{ego}} + b\Delta v^2 = f(\tau, b), \tag{7}$$

The driving behavior parameters τ and b are represented in vector form as:

$$\boldsymbol{\theta}(k) \triangleq \left[\tau(k)\ b(k)\right]^T. \tag{8}$$

The linearized model equation utilized in the EKF was derived by partially differentiating (7) with respect to each parameter.

$$\mathbf{F}(k) = \left[\frac{\partial f}{\partial \tau}\ \frac{\partial f}{\partial b}\right] = \left[v_{\text{ego}}\ \Delta v^2\right]. \tag{9}$$

The EKF-based parameter adaptation is characterized by the following set of equations [24]:

$$\mathbf{P}^{-}(k) = \Lambda^{-1}\mathbf{P}^{+}(k-1)\Lambda^{-1} + \mathbf{Q}(k-1), \tag{10}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}. \tag{11}$$

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{K}(k)\left[y(k) - f\left(\hat{\boldsymbol{\theta}}(k-1), \boldsymbol{u}, k\right)\right], \tag{12}$$

$$\mathbf{K}(k) = \mathbf{P}^{-}(k)\mathbf{F}^{T}(k)\left[\mathbf{F}(k)\mathbf{P}^{-}(k)\mathbf{F}^{T}(k) + \mathbf{R}(k)\right]^{-1}, \tag{13}$$

$$\mathbf{P}^{+}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{F}(k)]\,\mathbf{P}^{-}(k), \tag{14}$$

In (10)–(14), λ_1 and λ_2 denote the forgetting factors that indicate the rate of change of the parameters τ and b, respectively. The variable k denotes the discrete time index indicating the current estimation or update step. $\boldsymbol{K}(k)$, $\boldsymbol{P}(k)$, and $\boldsymbol{F}(k)$ represent the Kalman gain, the estimation error covariance, and the linearized model equations, respectively. $\boldsymbol{u}(k)$ represents the control input vector at time step k, $\boldsymbol{R}(k)$ denotes the measurement noise covariance matrix, and $\boldsymbol{I}$ is the identity matrix with appropriate dimensions. As illustrated in (12), the EKF adapts the parameters by accounting for the error value between the reference inter-vehicle distance and the predicted value based on the current spacing policy.

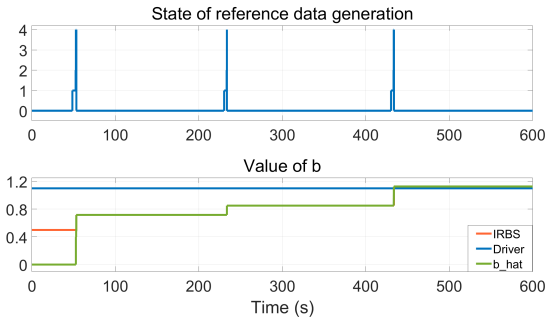

FIGURE 8. Adaptation state and adapted value of *b*.

2) EXPERIMENTAL VALIDATION OF EKF ADAPTATION ALGORITHM

When parameter adaptation is activated (in states 3, 4, 8, or 9), the reference data d_{des} are generated, and the parameters τ and b are adapted to synchronize the IRBS-controlled distance with the driver's preferred distance d_{des}. Accordingly, simulations were conducted to evaluate whether the implemented EKF adaptation algorithm effectively adapts the estimated parameters $\hat{\tau}$ and $\hat{b}$ upon reaching the parameter adaptation state. Here, $\hat{\tau}$ and $\hat{b}$ represent the estimated values of the parameters τ and b, respectively.

As depicted in Fig. 8, parameter b is adapted when the system reaches State 4. The IRBS control parameter b was initially set to 0.5, while the reference value for b was set to 1.1. Following three adaptation steps, $\hat{b}$ and the IRBS control parameter b converged to 1.12672, indicating successful alignment with the reference value.

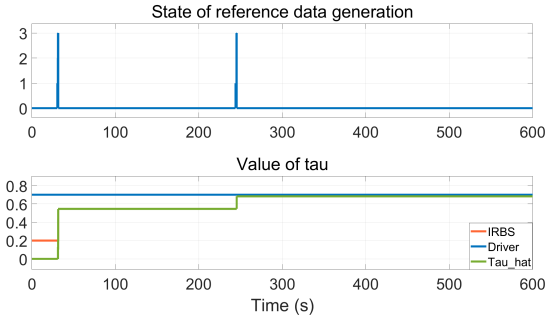

FIGURE 9. Adaptation state and adapted value of τ.

Similarly, as illustrated in Fig. 9, parameter τ is adapted upon the system reaching State 3. In this case, IRBS control parameter τ was initialized to 0.2, and the reference value for τ was set to 0.7. After two adaptation steps, both $\hat{\tau}$ and τ converged to 0.683. These results validate that the EKF-based adaptation algorithm reliably estimates the reference data and effectively adapted parameters under the adaptation-enabled conditions.

IV. VALIDATION VIA DRIVER MODEL-BASED SIMULATION

A. IMPLEMENTATION OF THE DRIVER MODEL

A simulation-based validation was performed to assess the efficacy of the reference data generator and the parameter adaptation algorithm described in Section II-B2. To improve the validation's clarity and reliability, a driver model with predefined reference values for τ and b was implemented. This driver model—acting as a virtual driver with reference data—determined whether to intervene by applying the brake or accelerator pedal inputs during IRBS operation in a car-following scenario. The performance of the improved IRBS was quantitatively evaluated based on the intervention level of the driver model.

The IRBS controller was initially deployed in a car-following scenario. A driver model was developed to capture individual driving characteristics, with the preferred inter-vehicle distance (reference data) varying according to driving conditions. The model intervenes when the IRBS-controlled distance diverges from the preferred distance (d_{des}). Different values were assigned to the parameters and b in the reference data of the driver model ($d_{\mathrm{des}} = d_s + \tau v_{\mathrm{ego}} + b\Delta v^2$) and to the initial IRBS control parameters. Vehicle states such as v_{ego} and Δv were obtained from a valid vehicle model implemented in CarSim. Upon driver intervention, reference data for parameter adaptation is generated, and the adapted parameters ($\hat{\tau}$, $\hat{b}$) are integrated into the IRBS inter-vehicle distance control. The Fig. 10 illustrates the overall validation flowchart of the adaptation algorithm.

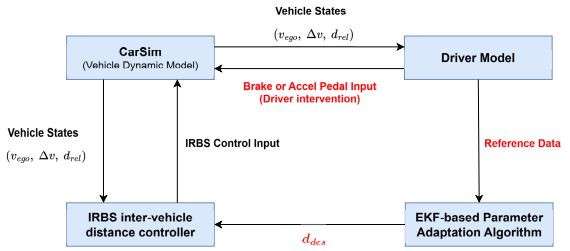

FIGURE 10. Flowchart for driver model-based adaptation algorithm validation.

The validation scenarios involving the driver model are as follows:

In the brake intervention scenario, the IRBS-controlled distance was initially set smaller than d_{des} prompting the driver model to apply the brake and reduce the distance to the preferred inter-vehicle distance when the gap to the preceding vehicle decreased. In the acceleration intervention scenario, the IRBS-controlled inter-vehicle distance was set longer than d_{des}, leading the driver model to apply the accelerator pedal to decrease the distance to the preferred distance. The adaptation algorithm utilizes the driver model's reference data to adapt the IRBS control parameters ($\hat{\tau}$ or $\hat{b}$) A subsequent drive in the same car-following scenario was conducted to evaluate whether the braking and acceleration interventions of the driver model were reduced after parameter adaptation. The EKF-based adaptation algorithm was validated by integrating the driver model and scenarios into CarSim, comparing intervention levels before and after adaptation.

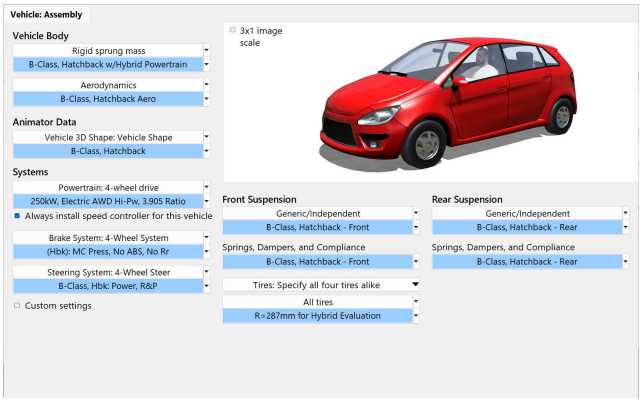

FIGURE 11. The simulation vehicle model of CarSim software.

B. SIMULATION RESULTS

1) SIMULATION SET-UPS

To validate this, a CarSim simulation environment, as described in Section IV-A, was constructed to closely approximate real-world driving conditions. A comparative analysis of the adapted parameters and the driver model intervention levels was performed using the simulation results. The proposed adaptation algorithm was considered effective when the adapted parameter approximates the reference data. A reduced level of driver model intervention after parameter adaptation indicates better incorporation of individual driving characteristics. Based on these criteria, a simulation environment was constructed, and the driver model outcomes of the driver model were systematically analyzed.

A vehicle model with specifications similar to a real-world vehicle was set up in CarSim, and a car-following scenario was implemented where the preceding vehicle decelerated. For simulation, the minimum safe distance in (2) was set to 10 m. The initial speed of the ego vehicle was defined as 0 km/h before entering the car-following scenario. The scenario involved only longitudinal control on a straight road, with the leading vehicle reducing speed from 60 to 5 km/h over 100 s. The deceleration scenario was repeated four times, with parameters adapted up to three times during the process.

Fig. 11 illustrates setups of the vehicle model, including the car body model, tire model, steering system model, brake system model, and suspension model. The vehicle model equipped with a regenerative braking system is CarSim B-class hatchback with a hybrid powertrain, which weighs 2100 kg. The sampling times were configured as 0.0005 s in CarSim and 0.01 s in Simulink, corresponding to the real ECU sampling period. For validation, the vehicle dynamics model and driving scenarios in CarSim were integrated with the driver model presented in Section IV-A using Simulink blocks.

2) VALIDATION RESULTS

The parameter adaptation results and the corresponding levels of driver model intervention are summarized below.

The final adapted values of the IRBS parameter for the brake intervention scenario of the driver model are presented in Table 2. Similarly, the reference data for both the driver model and the IRBS control parameters were arbitrarily assigned, as detailed in Section IV-A. Consequently, across Cases 1–4, the adapted parameter closely aligned with the reference value specified in the driver model.

TABLE 2. Parameter b adaptation results by EKF adaptation algorithm $brake intervention).

Case 1	τ	b	**Case 2**	τ	b
IRBS (Initial Value)	0.5	0.6	IRBS (Initial Value)	0.5	0.5
Driver Model	0.5	**1**	Driver Model	0.5	**1.1**
Adapted Value	×	**1.015**	**Adapted Value**	×	**1.087**
Case 3	τ	b	**Case 4**	τ	b
IRBS (Initial Value)	0.5	0.7	IRBS (Initial Value)	0.5	0.7
Driver Model	0.5	**0.9**	Driver Model	0.5	**1.2**
Adapted Value	×	**0.949**	**Adapted Value**	×	**1.226**

Fig. 12 illustrates the brake interventions of the driver model before and after adapting the parameter to the IRBS. The before and after adaptation values in Fig. 12 correspond to those in Case 2, as presented in Table 2. The findings revealed that adapting the parameter b resulted in an average 50% reduction in driver brake interventions. In this context, the average reduction in driver intervention was quantitatively

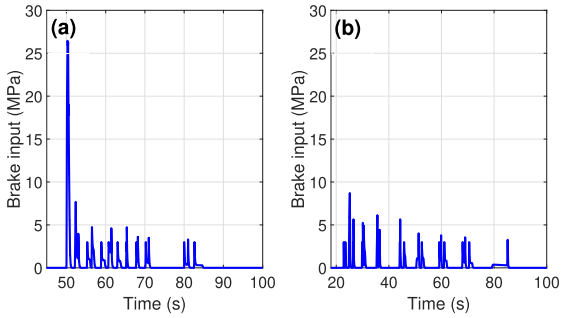

FIGURE 12. Brake pressure by Driver Model Intervention (a) Before, and (b) After *b* adaptation.

computed using valid intervention data, excluding brake intervention (brake pressure) instances that did not meet the data preprocessing criteria. All subsequent average values were derived using the same criteria and methodology.

Table 3 summarizes the final adapted values of parameter for the same brake intervention scenario. Similar to the b adaptation, the reference data for the driver model and IRBS control parameters were arbitrarily defined. As demonstrated in Table 3 (Cases 1 - 4), the adapted parameter τ closely matches the corresponding reference values.

TABLE 3. Parameter τ adaptation results by EKF adaptation algorithm $brake intervention).

Case 1	τ	b	Case 2	τ	b
IRBS (Initial Value)	0.2	1	IRBS (Initial Value)	0.3	1
Driver Model	**0.5**	1	Driver Model	**0.8**	1
Adapted Value	**0.488**	×	**Adapted Value**	**0.757**	×
Case 3	τ	b	**Case 4**	τ	b
IRBS (Initial Value)	0.2	1	IRBS (Initial Value)	0.25	1
Driver Model	**0.65**	1	Driver Model	**0.5**	1
Adapted Value	**0.620**	×	**Adapted Value**	**0.542**	×

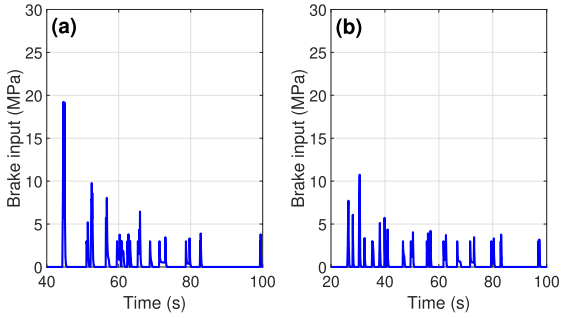

FIGURE 13. Brake pressure by driver model intervention (a) Before, and (b) after τ adaptation.

Fig. 13 depicts the driver model's brake intervention before and after τ adaptation within the IRBS control. The before-and-after adaptation values in Fig. 13 correspond to those in Case 2, as listed in Table 3. The results indicate that adapting parameter τ led to an average reduction of approximately 35% in driver brake intervention.

The final adapted values of parameter for the acceleration intervention scenario of the driver model along-side the corresponding reference values are comparatively outlined in Table 4. In both Cases 1 and 2, the adapted parameter closely matches the reference values. Using the values of

TABLE 4. Parameter τ adaptation results by EKF adaptation algorithm $accel intervention).

Case 1	τ	b	Case 2	τ	b
IRBS (Initial Value)	0.5	0.6	IRBS (Initial Value)	1	0.5
Driver Model	**0.3**	**1**	Driver Model	**0.5**	**1.1**
Adapted Value	**0.324**	×	**Adapted Value**	**0.497**	×

parameter from Table 4 for Cases 1 and 2, the resulting accelerator intervention outcomes are illustrated in Fig. 14 and Fig. 15, respectively. These figures show the driver model's accelerator control input interventions of the driver model before and after adaptation within the IRBS. Specifically, Fig. 14 demonstrates that τ adaptation reduced the driver's accelerator intervention by approximately 50%, whereas Fig. 15 indicates a reduction of approximately 88%.

Simulation-based driving with the driver model confirmed that the proposed parameter adaptation algorithm effectively reduces driver brake and accelerator interventions and accurately calibrated the parameters to the reference values. Therefore, the proposed algorithm more effectively reflects driver behavior than existing IRBS.

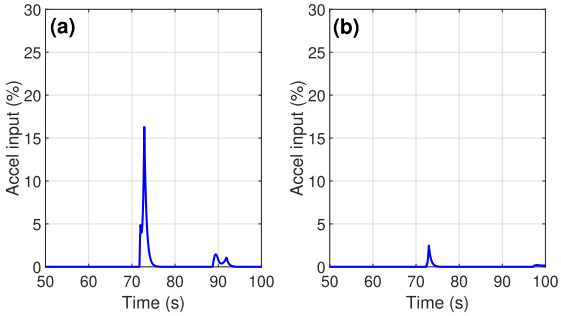

FIGURE 14. Accelerator Control Input in Case 1 by Driver Model Intervention (a) Before and, (b) After τ adaptation.

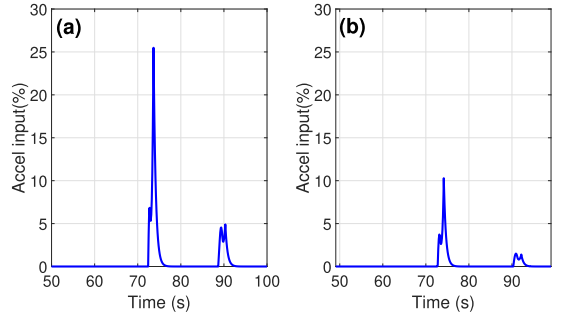

FIGURE 15. Accelerator Control Input in Case 2 by Driver Model Intervention (a) Before and, (b) After τ adaptation.

V. VEHICLE TESTS

A. VEHICLE TEST OVERVIEW

The overall framework of the proposed EKF-based parameter adaptation algorithm implemented for real-vehicle testing is illustrated in Fig. 16. The algorithm was embedded in a vehicle control unit and evaluated through vehicle testing to verify its performance. To ensure that the driver's reference data d_{des} were generated under predefined conditions, driving data from multiple real-time vehicle tests in a car-following scenario were analyzed. The analysis also examined whether

TABLE 5. τ Adaptation results by EKF in vehicle test (brake intervention).

	IRBS (Initial Value)	Adapted Value
τ	0.89025	1.79688

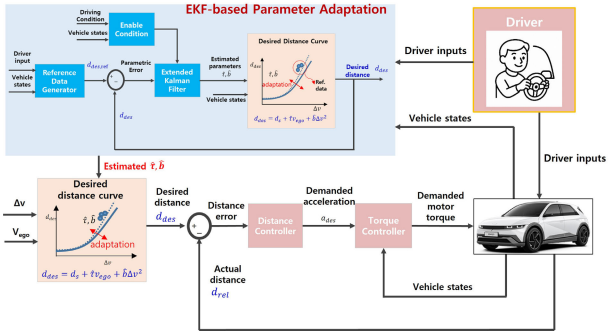

FIGURE 16. EKF-Based Parameter Adaptation Framework for Personalized Car-Following in IRBS With Real-Vehicle Validation.

the IRBS effectively incorporated d_{des} into controlling the inter-vehicle distance via parameter adaptation. Driving data were collected according to the driver brake intervention scenario, and the analysis followed the plotting method detailed in Section II-B. In a blind test involving five drivers, all participants reported increased comfort in inter-vehicle distance control when using the IRBS with the proposed algorithm.

B. VEHICLE TEST RESULTS

The initial IRBS value before parameter adaptation and the adapted value of τ obtained from the vehicle test are summarized in Table 5.

The reference data generation process d_{des} for τ adaptation, as outlined in Table 2, is depicted in Fig. 17 (a) and (b). The measured data in Fig. 17 (a) confirm that the conditions for generating reference data are satisfied, illustrating a car-following scenario where the vehicle decelerates under IRBS control and the driver intervenes by braking to achieve the desired inter-vehicle distance. Fig. 17 (b) depicts that the reference data generator reaches State 3, during which the reference data d_{des} for adaptation of τ is generated in real time. Subsequently, τ adaptation performed based on this generated data. Fig. 17 (c) presents the performance of the EKF-based parameter adaptation algorithm through data analysis. The IRBS configured with initial values prior to parameter adaptation failed to reflect the driver's driving characteristics, resulting in an inter-vehicle distance error exceeding 15 m from the desired d_{des}. After parameter adaptation, the IRBS achieved a distance error of less than 1 m relative to the desired d_{des}, demonstrating that the inter-vehicle distance control more effectively reflects the driver's driving characteristics.

These results validate that the EKF-based parameter adaptation algorithm ensures stable spacing control performance by incorporating the driver's behavior into the inter-vehicle distance control of the IRBS. Additionally, a vehicle driving test demonstrates that the proposed algorithm operates

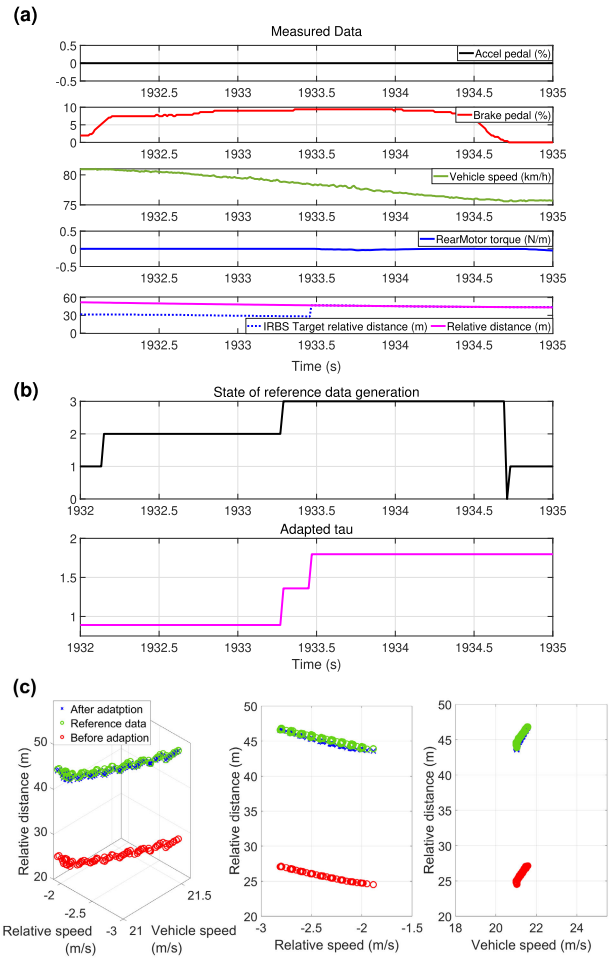

FIGURE 17. Evaluation of EKF adaptation algorithms through Vehicle Test(a) Measured data of vehicle test, (b) adaptation state and adapted τ value,(c) performance of parameter adaptation.

reliably with low computational resources, indicating its effectiveness for real-time applications and suitability for mass production.

VI. CONCLUSION

This study introduces an EKF-based parameter adaptation algorithm for an IRBS that enables personalized car-following control by reflecting individual driving behaviors in real time. Unlike conventional approaches that rely on deep or offline learning, the proposed algorithm achieves computationally efficient adaptation, making it suitable for real-time implementation in embedded vehicle systems. To effectively parameterize driver behavior, a DRS spacing policy was adopted and experimentally validated. Based on the analysis of driving data, preprocessing criteria were established to identify scenarios where the driver's preferred inter-vehicle distance d_{des} was clearly exhibited. A Reference Data Generator was developed to generate reference data in real time when a driver's behavior was distinctly observed. Using this reference data, the algorithm estimates and adapts parameters in real time to effectively reflect the individual driving characteristics.

Simulation results with a driver model demonstrated that the algorithm reduced driver intervention and aligned well with the reference parameters, confirming its effectiveness in personalizing IRBS inter-vehicle distance control. Vehicle testing validated the computational efficiency and suitability of the proposed algorithm for large-scale production.

Overall, this study introduced a practical and scalable framework for personalized vehicle control, integrating individual driver characteristics without relying on complex learning architectures. The method enhances driving comfort and is compatible with a broader range of ADAS technologies, including ACC.

Nonetheless, this study adapted the parameters τ and b independently based on specific criteria. Future research should focus on developing an integrated algorithm that enables the simultaneous adaptation of both parameters. Furthermore, the proposed method primarily addresses longitudinal vehicle control, future studies may consider extending it to a comprehensive ADAS framework that also accounts for lateral control. Additionally, extending the algorithm to reflect car-following behavior in cornering scenarios may serve as an important direction for future research.

REFERENCES

[1] L. Yue, M. A. Abdel-Aty, Y. Wu, and A. Farid, "The practical effectiveness of advanced driver assistance systems at different roadway facilities: System limitation, adoption, and usage," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3859–3870, Sep. 2020.

[2] F. J. Belmonte, S. Martín, E. Sancristobal, J. A. Ruipérez-Valiente, and M. Castro, "Overview of embedded systems to build reliable and safe ADAS and AD systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 4, pp. 239–250, Winter. 2021.

[3] K. Damsara and A. R. D. S. Barros, "A systematic review on user acceptance of advanced driver assistance systems (ADAS)," *Transp. Res. Proc.*, vol. 82, pp. 3472–3482, Jan. 2025.

[4] Y. Wang, "Research on adaptive cruise control strategy of distributed drive intelligent electric vehicle," in *Proc. Asia–Pacific Conf. Image Process., Electron. Comput. (IPEC)*, Apr. 2023, pp. 130–134.

[5] C. Yang and B. Liu, "Research on adaptive cruise control system for electric vehicles," in *Proc. 5th Int. Conf. Electron. Commun. Artif. Intell. (ICECAI)*, May 2024, pp. 645–648.

[6] S. Vasebi, Y. M. Hayeri, and A. M. Saghiri, "A literature review of energy optimal adaptive cruise control algorithms," *IEEE Access*, vol. 11, pp. 13636–13646, 2023.

[7] J. Guo, W. Li, J. Wang, Y. Luo, and K. Li, "Safe and energy-efficient car-following control strategy for intelligent electric vehicles considering regenerative braking," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7070–7081, Jul. 2022.

[8] W. Wang and J. Xi, "A rapid pattern-recognition method for driving styles using clustering-based support vector machines," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 5270–5275.

[9] B. Gao, K. Cai, T. Qu, Y. Hu, and H. Chen, "Personalized adaptive cruise control based on online driving style recognition technology and model predictive control," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12482–12496, Nov. 2020.

[10] H. Chu, L. Guo, Y. Yan, B. Gao, and H. Chen, "Self-learning optimal cruise control based on individual car-following style," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6622–6633, Oct. 2021.

[11] Z. Zhao, Z. Wang, K. Han, R. Gupta, P. Tiwari, G. Wu, and M. J. Barth, "Personalized car following for autonomous driving with inverse reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2891–2897.

[12] F. Islam, J. E. Ball, and C. T. Goodin, "Enhancing longitudinal velocity control with attention mechanism-based deep deterministic policy gradient (DDPG) for safety and comfort," *IEEE Access*, vol. 12, pp. 30765–30780, 2024.

[13] F. Islam, M. M. Nabi, J. E. Ball, and C. T. Goodin, "Optimizing longitudinal velocity control via self-supervised learning and deep deterministic policy gradient," *IEEE Access*, vol. 12, pp. 128963–128978, 2024.

[14] W. Lim, S. Lee, J. Yang, M. Sunwoo, Y. Na, and K. Jo, "Automatic weight determination in model predictive control for personalized car-following control," *IEEE Access*, vol. 10, pp. 19812–19824, 2022.

[15] K. H. Kwak, Y. He, Y. Kim, Y. M. Chen, S. Fan, J. Holmer, and J. H. Lee, "Desired relative distance model-based personalized braking algorithm for one-pedal driving of electric vehicles," *IFAC-PapersOnLine*, vol. 55, no. 37, pp. 62–67, 2022.

[16] Y. He, S. Fan, K. Hyun Kwak, and Y. Kim, "Personalized automated braking for one-pedal driving in electric vehicles using model predictive control," *IEEE Trans. Transport. Electrific.*, vol. 11, no. 1, pp. 3215–3228, Feb. 2025.

[17] C. Wu, Z. Xu, Y. Liu, C. Fu, K. Li, and M. Hu, "Spacing policies for adaptive cruise control: A survey," *IEEE Access*, vol. 8, pp. 50149–50162, 2020.

[18] L. Yang, J. Mao, K. Liu, J. Du, and J. Liu, "An adaptive cruise control method based on improved variable time headway strategy and particle swarm optimization algorithm," *IEEE Access*, vol. 8, pp. 168333–168343, 2020.

[19] Y. He, M. Montanino, K. Mattas, V. Punzo, and B. Ciuffo, "Physics-augmented models to simulate commercial adaptive cruise control (ACC) systems," *Transp. Res. C, Emerg. Technol.*, vol. 139, Jun. 2022, Art. no. 103692.

[20] W. B. Qin, "A nonlinear car-following controller design inspired by human-driving behaviors to increase comfort and enhance safety," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8212–8224, Aug. 2022.

[21] K. Wang, Y. Yang, S. Wang, and Z. Shi, "Research on car-following model considering driving style," *Math. Problems Eng.*, vol. 2022, pp. 1–9, Feb. 2022.

[22] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed., New York, NY, USA: Springer, 2011.

[23] Å. Björck, *Numerical Methods for Least Squares Problems*. Philadelphia, PA, USA: SIAM, 2024.

[24] S. Kim, H. Lee, J. Kim, and G. Park, "Online adaptive identification of clutch torque transmissibility for the drivability consistency of high-performance production vehicles," *Control Eng. Pract.*, vol. 147, Jun. 2024, Art. no. 105926.

SEUNGYEON OAK received the B.S. degree in mechanical systems engineering from Sookmyung Women's University, Seoul, South Korea, in 2025. From 2023 to 2025, she was with the E-mobility Control Laboratory (EMCLab), Sookmyung Women's University to conduct research on vehicle control for advanced driver assistance systems. Her research interests include vehicle control systems, data-driven modeling and control, and electrified and autonomous vehicles.

DAEKYEONG LEE received the B.S. degree in mechanical systems engineering from Sookmyung Women's University, Seoul, South Korea, in 2024. From 2022 to 2023, she was with the Machine Perception and Intelligence Laboratory (MPIL), Daegu Gyeongbuk Institute of Science and Technology, conducting research on visual-inertial odometry and naviagation. From 2023 to 2024, she was with the E-mobility Control Laboratory (EMCLab), Sookmyung Women's University to conduct research on vehicle control for advanced driver assistance systems. Her research interests include localization, mapping, and real-time control systems for autonomous vehicles.

GYUBIN SIM received the M.S. degree in automotive electronics control engineering from Hanyang University, Seoul, South Korea, in 2020. Since 2020, he has been a Software Engineer with Hyundai Motor Company, where he is involved in the development of control software for electric vehicle drivetrain systems. His research interests include embedded systems, vehicle control, and automotive software engineering.

SOOYOUNG KIM received the B.S., M.S., and Ph.D. degrees in mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2012, 2014, and 2018, respectively. From 2018 to 2022, he was with Hyundai Motor Company, Hwasung, South Korea, where he was involved in the control and state estimation of electrified powertrains for mass-produced vehicles. Since 2022, he has been a Faculty Member with the Department of Mechanical Systems Engineering, Sookmyung Women's University, South Korea. His current research interests include data-driven modeling and the model-based control of electrified autonomous vehicle systems.

GISEO PARK received the B.S. degree in mechanical engineering from Hanyang University, Seoul, South Korea, in 2014, and the M.S. and Ph.D. degrees in mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST), in 2016 and 2020, respectively. He was a Senior Engineer with Hyundai Motor Company, Hwasung, South Korea. He is currently an Assistant Professor with the School of Mechanical Engineering, University of Ulsan, Ulsan, South Korea. His research interests include vehicle dynamics, control theory, active safety systems, and autonomous driving systems.