IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 18, 2021, accepted November 1, 2021, date of publication November 4, 2021, date of current version November 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3125351

An Integrated Deep Ensemble-Unscented Kalman Filter for Sideslip Angle Estimation With Sensor Filtering Network

DONGCHAN KIM[1], GIHOON KIM[1], SEUNGWON CHOI[2], AND KUNSOO HUH[1], (Member, IEEE)

[1]Department of Automotive Engineering, Hanyang University, Seoul 04763, South Korea
[2]Department of Automotive Engineering (Automotive-Computer Convergence), Hanyang University, Seoul 04763, Republic of Korea

Corresponding author: Kunsoo Huh (khuh2@hanyang.ac.kr)

This work was supported by the Technology Innovation Program (Industrial Strategic technology development program, 20006862, Development of Automated Driving Systems and Evaluation) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

ABSTRACT An integration scheme for sideslip angle estimation is proposed where a deep neural network and a simple kinematics-based model are combined in an unscented Kalman filter. The deep neural network contains two modules: a sensor filtering network designed to overcome the limitations of the kinematics-based model and a deep ensemble network to estimate the sideslip angle and its uncertainty. Both networks use recurrent neural networks with long short-term memory to analyze sequential sensor data. The networks were trained using only input signal sets that can be obtained from on-board sensor measurements. The filtering network reduces the noise and bias of the input signals to match the model used for the unscented Kalman filter. Next, the initial estimate and its uncertainty obtained from the deep ensemble network are utilized as a new measurement in the unscented Kalman filter, inducing an adaptive measurement variance. The algorithm was verified through both simulation and experiment, and the results demonstrate the effectiveness of the proposed algorithm.

INDEX TERMS Sideslip angle estimation, kinematic model, unscented Kalman filter, deep ensemble, uncertainty, sensor filtering network.

I. INTRODUCTION

Modern vehicles are equipped with several components to improve their safety and stability [1]. Among these, many functions concern the lateral stability of the vehicle [2]–[5]. In particular, state variables related to vehicle motion cannot be easily measured by on-board sensors and their estimation has been the focus of intensive research in past decades, resulting in many works in the literature. These include a comparative study on the sideslip angle estimation [6], the longitudinal velocity estimation [7], and roll angle estimation [8]. The focus of this work considers the information about lateral state, especially the sideslip angle, β. To obtain a direct measurement of the sideslip angle, expensive and bulky sensors such as optical or RTK-GPS sensors are needed. Therefore, adequate strategies of indirect estimation are required to overcome this problem.

To address this problem, extensive studies have been conducted on methods for estimating the sideslip angle. For conventional methods, there are several model-based Kalman filtering (KF) approaches that utilize kinematics- and dynamics-based models. First, the kinematics-based model considers the movement of a vehicle without considering the forces [9]. The representative model is expressed considering the velocity, lateral acceleration, and yawrate [10]. However, the sideslip angle is in an integrated form in the sensor results. Thus, it is vulnerable to noise and bias, which accumulate over time, resulting in a large estimation error. Therefore, dynamics-based models with numerous degrees of freedom have been considered to enhance model accuracy and estimation performance. For example, two DOF single-track model and higher-order DOF vehicle models [11]–[15] have been utilized for estimator design. The advantage of using dynamics-based models is that the quality of models improves the estimation quality of the states. However, as the model complexity increases, the number of parameters to estimate also increases. In addition, the uncertainty and

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

non-linearity of the models make the estimation problem very difficult.

Recently, sideslip angle estimation methods [16]–[21] based on artificial neural networks (ANNs) and deep neural networks (DNNs) have seen dramatic progress. Melzi *et al.* designed a simple ANN using a single hidden layer to verify its performance [16]. Chindamo *et al.* also used the ANN to verify the estimation performance by means of the *CarSim* simulator [17]. Boada *et al.* designed an adaptive neuro-fuzzy inference system (ANFIS) [18]. This method improves the estimation performance in comparison to the simple ANN by combining the advantages of the steady performance of fuzzy logic and the generalization of ANN. Ghosh *et al.* used a multilayer long short-term memory (LSTM) network [19] to exploit the characteristic that previous inputs can be used to obtain the current state. Graeber *et al.* proposed a hybrid approach to sideslip angle estimation using recurrent neural network (RNN) and kinematic vehicle models [20]. They used the derivative of the sideslip angle, $\dot{\beta}$, obtained from the kinematic model, as a new input feature of the RNN to improve the performance. Bonfitto *et al.* employed a classification neural network to identify road conditions, and then designed three regression neural networks for use under different road conditions that were identified by the former network [21].

Nonetheless, there still exist limitations of learning-based models such that the estimation of learning-based methods is always considered to be reliable. Kim *et al.* considered uncertainty of learning-based method and designed various Kalman filter algorithms based on a dynamic model [22]. But, this approach requires significant computational burden due to the dynamic model. In addition, dynamic models that deal with forces and torques require considerable knowledge of parameters that can change over time.

In this paper, two independent approaches are combined to overcome these limitations. First, a sensor filtering network is proposed to overcome the limitation of the kinematics-based model which does not require vehicle parameters, tire-road friction coefficients, or a tire model. The sensor filtering network is utilized to match the calculated $\dot{\beta}$ from the filtered data to the actual value while reducing the noise and bias of the original sensor data. Next, the deep ensemble method is used to obtain a robust estimate and its uncertainty [23]. The information on uncertainty is valuable because the estimate can be less reliable when tested in an unfamiliar environment. To address this, the uncertainty is utilized as a measurement variance value of the adaptive unscented Kalman filer (UKF). This method can improve the reliability of the estimate when uncertainty is low, and otherwise, the model with the filtered sensor inputs is considered to be more reliable. The contributions of this study can be summarized as follows:

- A novel sideslip angle estimation method is proposed that combines the DNN and UKF with a simple kinematics-based model.
- The filtering network is designed to overcome the limitations of the kinematic-based model.
- The deep ensemble network with LSTM is utilized to analyze sequential sensor data, in addition to providing a robust estimation and the estimation uncertainty.
- The feasibility of the proposed method was validated under various road surface conditions using the commercial software *CarSim*, which uses high-dimensional vehicle dynamics. In addition, verification was carried out through real-car experiment in some limited environments.

The remainder of this paper is organized as follows. In Section 2, the deep learning models comprising the sensor filtering and deep ensemble networks are described. Next, Section 3 explains the integration of both deep learning and kinematics-based models with the UKF in detail. In Section 4, the proposed algorithm is verified in simulation. Then, the proposed algorithm is verified in experiment in Section 5.

II. DEEP LEARNING MODELS

A. PRELIMINARIES

In this study, RNN and deep ensemble methods are used for precise sideslip angle estimation. First, an RNN is utilized to analyze the sequential data. A basic RNN has a vanishing gradient problem; therefore an LSTM network is used instead [24]. Next, a deep ensemble is used in this study as it uses multiple models to derive outcomes. This method improves the estimation performance and makes the estimate more robust. In our work, the deep ensemble network developed by Lakshminarayanan *et al.* [23] is used to estimate the sideslip angle and its uncertainty. From each network of the entire ensemble model, the estimate (μ) and standard deviation (σ) of the estimate are obtained given the sensor input (S). Then, the final estimate (μ_f) and standard deviation (σ_f) are calculated as follows:

$$\mu_f = \frac{\sum_i \mu_i(S_i)}{N} \tag{1}$$

$$\sigma_f^2 = \frac{\sum_i (\sigma_i^2(S_i) + \mu_i^2(S_i))}{N} - \mu_f^2 \tag{2}$$

where N is the number of networks and subscript i indicates the index of a single network. As the standard deviation should always have a positive value, the softplus function is applied as follows.

$$softplus(\sigma_f) = \log(1 + \exp(\sigma_f)) \tag{3}$$

B. DEEP LEARNING ARCHITECTURE

The deep learning models are divided into two modules. The first module filters sensor data to make it available for the kinematics-based model. The filtering network is composed of denoising and debiasing networks as shown in Fig. 1; both have one LSTM layer with a 32-dimensional cell state. The output of each denoising LSTM cell is connected to a fully-connected layer with 32 hidden units, whereas in the case of the debiasing LSTM, only the end of LSTM is connected to a fully-connected layer. The sensor data input for the filtering network, S_{SF}, includes velocity (v_x), yawrate ($\dot{\psi}$) and lateral

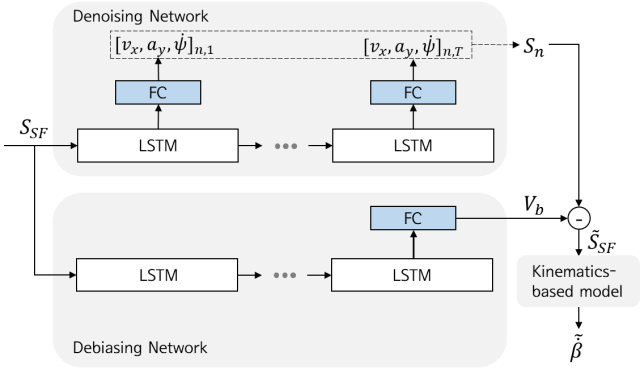

FIGURE 1. Sensor filtering network.

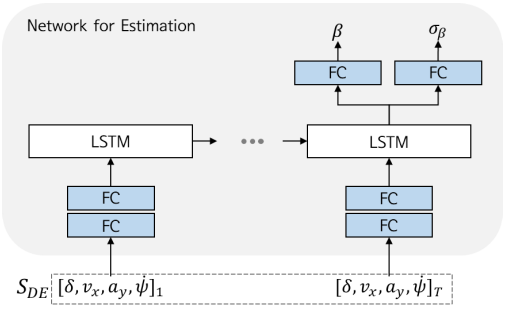

FIGURE 2. Network for sideslip angle and its uncertainty estimation.

acceleration (a_y) for 10 steps in the interval of 0.1 s. The noisy and biased measurements at timestamp n are modeled as follows [25]:

$$v_{x,n} = v_{x,n}^{true} + w_n^{v_x} \tag{4}$$

$$\dot{\psi}_n = \dot{\psi}_n^{true} + b_n^{\dot{\psi}} + w_n^{\dot{\psi}} \tag{5}$$

$$a_{y,n} = a_{y,n}^{true} + b_n^{a_y} + w_n^{a_y} \tag{6}$$

where $b_n^{\dot{\psi}}$ and $b_n^{a_y}$ are quasi-constant biases and $w_n^{v_x}$, $w_n^{\dot{\psi}}$, and $w_n^{a_y}$ are zero-mean Gaussian noises.

The sensor data with noise and bias are input to the denoising LSTM which provides denoised sensor signals, S_n, with biases. The output of the debiasing LSTM, V_b, is utilized to obtain the final filtered sensor signals, $\tilde{S}_{SF}$, which are used to calculate the derivative of the sideslip angle using (9) through the kinematics-based model.

To train the sensor filtering network, the loss function is composed of the following three terms:

$$L_{SF} = c_1|S_n - S_{SF}| + c_2|\dot{\beta}_{true} - \tilde{\dot{\beta}}| + c_3 \sum_{i=1}^{bs} |V_{b,i} - V_{b,i-1}| \tag{7}$$

where $\dot{\beta}_{true}$ and $\dot{\beta}$ indicate the true value of the derivative of the sideslip angle and the estimated value from the sensor filtering network, respectively, and bs is the batch size. c_1, c_2, and c_3 are weights for each loss.

The first loss term is used to reduce the difference between the initial unfiltered sensor values and denoised sensor values. This loss prevents the output of the denoising network from deviating significantly from the original value. The second term reduces the difference between the true and calculated value from the sensor signals obtained from the network. The last term restricts the bias value from fluctuating because quasi-constant biases are assumed in (5) and (6). Through these processes, the sensor signals are trained to be modified to match well to the kinematics-based model.

The second module is used to estimate the sideslip angle and its uncertainty. The module consists of fully connected layers and an LSTM. Input features, S_{DE}, are a combination of the steering wheel angle (δ), velocity (v_x), yawrate ($\dot{\psi}$) and lateral acceleration (a_y). A total of 10 sequential data points, with an interval of 0.1 s, is used as the input of each fully-connected layer as shown in Fig. 2. Each input feature passes through two fully-connected layers with 128 hidden units, and the result is connected to the one-layer LSTM with a 256-dimensional cell state. Next, by passing the output of the LSTM to a fully-connected layer, an estimate and its uncertainty are obtained. Finally, by incorporating the results from each ensemble network using (1) and (2), the final sideslip angle estimate and standard deviation are calculated.

To train the deep ensemble network, negative log likelihood (NLL) loss is used as follows [26]:

$$L_{nll} = \frac{\log(\sigma^2)}{2} + \frac{(y-\mu)^2}{2\sigma^2} + \frac{1}{2}\log(2\pi) \tag{8}$$

where y is the true value, μ is the estimated value and σ is the standard deviation of the estimate. As the NLL loss captures predictive uncertainty, if the unseen input data is given, the deep ensemble provides the estimate with large uncertainty accordingly.

Two networks are trained independently and the standard ADAM optimizer [27] is used. For the filtering network, the learning rate is set to 0.0001 with a batch size of 32 and 10000 training iterations. Then, the learning rate is set to 0.001 for the deep ensemble network with a batch size of 256 and 20000 training iterations.

III. INTEGRATION OF DEEP LEARNING-BASED AND MODEL-BASED SIDESLIP ANGLE ESTIMATION

A. KINEMATICS-BASED MODEL

A kinematics-based model considers only the motion of vehicle without any information regarding tire-road friction or other dynamic properties of the vehicle. A typical kinematics-based model can be expressed as a numerical integration of sensor signals comprising yawrate ($\dot{\psi}$), vehicle velocity (v_x) and lateral acceleration (a_y) as follows:

$$\dot{\beta} = \frac{a_y}{v_x} - \dot{\psi} \tag{9}$$

However, there is a critical drawback of using a kinematics-based model. As sensor signals are always accompanied by noise or bias, the kinematics-based model cannot be perfectly accurate, resulting in poor estimation performance [9]. This

problem is investigated using a sensor filtering network in this study.

B. INTEGRATED METHOD

The extend Kalman filter (EKF) uses a linearization process; thus, the performance worsens when the non-linearity gets stronger. In contrast, the UKF uses a non-linear function without a linearization process. First, the system state, x_t, and measurement, z_t, are expressed using the state transition function, g, and measurement model, h, as follows:

$$x_t = g(u_t, x_{t-1}) + w_t \tag{10}$$
$$z_t = h(x_t) + \epsilon_t \tag{11}$$

where u_t is the input, and w_t and ϵ_t are the process and measurement noise, respectively. The non-linear function g in our work uses the kinematic model in (9). The measurement model h is the linear function which simply assigns the input given to itself. The state vector becomes $x = [\beta]$, the input vector becomes $u = [v_x, \dot{\psi}, a_y]^T$, and the measurement vector becomes $z = [\beta_{DE}]$ which is the sideslip angle estimate from the deep ensemble network. These vectors are utilized in Algorithm 1 to estimate the sideslip angle.

Then, the UKF uses sigma points that are symmetrically distributed near the estimated state. For n-dimensional Gaussian with the mean value (μ) and covariance (Σ), the resulting $2n + 1$ sigma points ($\mathcal{X}^{[i]}$) are chosen as follows:

$$\mathcal{X}^{[0]} = \mu \tag{12}$$
$$\mathcal{X}^{[i]} = \mu + (\sqrt{(n+\lambda)\Sigma})_i, \quad i = 1, \ldots, n \tag{13}$$
$$\mathcal{X}^{[i]} = \mu - (\sqrt{(n+\lambda)\Sigma})_i, \quad i = n+1, \ldots, 2n \tag{14}$$

where $\lambda = \alpha^2(n + \kappa) - n$, α and κ are scaling parameters.

Then, to recover the mean and covariance of the Gaussian distribution, each sigma point is given a weight as follows:

$$w_m^{[0]} = \frac{\lambda}{n+\lambda} \tag{15}$$
$$w_c^{[0]} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \zeta) \tag{16}$$
$$w_m^{[i]} = w_c^{[i]} = \frac{1}{2(n+\lambda)}, \quad i = 1, \ldots, 2n \tag{17}$$

where ζ contains prior information of the distribution. w_m is used to compute the mean and w_c is used to recover the covariance of the Gaussian.

The mean (μ') and covariance (Σ') of the Gaussian distribution can be recovered from the sigma points explained above and are shown in (18) and (19):

$$\mu' = \sum_{i=0}^{2n} w_m^{[i]} \xi^{[i]} \tag{18}$$
$$\Sigma' = \sum_{i=0}^{2n} w_c^{[i]} (\xi^{[i]} - \mu')(\xi^{[i]} - \mu')^T \tag{19}$$

where $\xi^{[i]} = g(\mathcal{X}^{[i]})$.

Algorithm 1 Deep Ensemble and Sensor Filtering Network Based Unscented Kalman Filter

1: **function** DEEP_UKF(μ_{t-1}, Σ_{t-1}, μ_t, z_t, $S_{DE,t}$, $S_{SF,t}$)
2: $\beta_{DE,t}, \sigma_{\beta_{DE},t} = f_{DE}(S_{DE,t})$
3: $\tilde{S}_{SF,t} = f_{SF}(S_{SF,t})$
4: $\tilde{z}_t = \begin{pmatrix} z_t \\ \beta_{DE,t} \end{pmatrix}$
5: $\tilde{Q}_t = \begin{pmatrix} Q_t & 0 \\ 0 & w_\beta \sigma^2{}_{\beta_{DE},t} \end{pmatrix}$
6: $\mathcal{X}_{t-1} = (\mu_{t-1}, \mu_{t-1} + \gamma\sqrt{\Sigma_{t-1}}, \mu_{t-1} - \gamma\sqrt{\Sigma_{t-1}})$
7: $\bar{\mathcal{X}}_t^* = g(\tilde{S}_{SF,t}, \mathcal{X}_{t-1})$
8: $\bar{\mu}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{X}}_t^{*[i]}$
9: $\bar{\Sigma}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)(\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)^T + R_t$
10: $\bar{\mathcal{X}}_t = (\bar{\mu}_t, \bar{\mu}_t + \gamma\sqrt{\bar{\Sigma}_t}, \bar{\mu}_t - \gamma\sqrt{\bar{\Sigma}_t})$
11: $\bar{Z}_t = h(\bar{\mathcal{X}}_t)$
12: $\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{Z}_t^{[i]}$
13: $S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{Z}_t^{[i]} - \hat{z}_t)(\bar{Z}_t^{[i]} - \hat{z}_t)^T + \tilde{Q}_t$
14: $\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t)^T$
15: $K_t = \bar{\Sigma}_t^{x,z} S_t^{-1}$
16: $\mu_t = \bar{\mu}_t + K_t(\tilde{z}_t - \hat{z}_t)$
17: $\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$
18: **return** μ_t, Σ_t
19: **end function**

All processes of the standard UKF are outlined in Lines 6-17 of the Algorithm 1 if $\tilde{Q}_t$, $\tilde{z}_t$ and $\tilde{S}_{t,SF}$ are changed to Q_t, z_t and u_t, respectively [28].

The deep learning-based and model-based methods are combined as shown in Fig. 3. From the deep ensemble network, the sideslip angle estimate, β_{DE}, and standard deviation, $\sigma_{\beta_{DE}}$, are extracted. This process can be seen in Line 2 of Algorithm 1, where $f_{DE}(\cdot)$ denotes the deep ensemble model. The obtained values are utilized as the measure and noise variance values. The standard deviation, $\sigma_{\beta_{DE}}$, provides valuable information to the model such that the reliability of the new measure can be adaptively considered in the UKF.

Next, the sensor filtering procedure is explained in Line 3, where $f_{SF}(\cdot)$ denotes the sensor filtering network. The output of the network is the input vector for the UKF system. This process overcomes the limitations of the kinematics-based model by filtering out the original biased and noisy sensor data.

IV. SIMULATION

To verify the feasibility of the proposed estimation method, *CarSim*, *MATLAB/Simulink* and Tensorflow [29] are utilized. From *CarSim*, the dataset needed for training the network is obtained by constructing the simulation environment. Then, the UKF algorithm is designed using *MATLAB/Simulink*. Finally, deep learning models are designed and trained in the Tensorflow environment. The test scenarios include different driving maneuvers or different surface condition compared to the dataset used in the training phase.

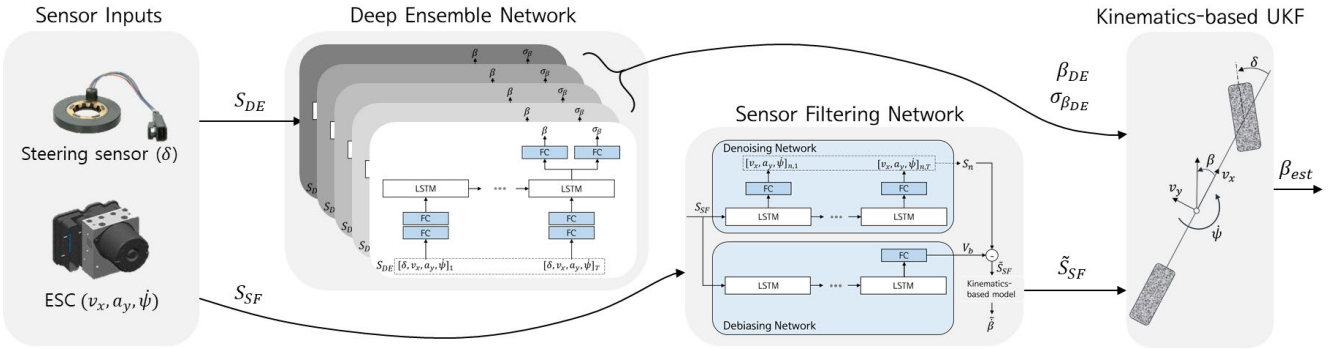

FIGURE 3. Overall architecture of the proposed method.

A. TRAINING DATASET

Collecting a dataset is an important process. A high-quality dataset can be used to train a model to deal with a wide range of situations. In our training dataset, two road surfaces are included, i.e., a dry asphalt road with tire-road friction (μ) of 0.85, and a slippery road with tire-road friction of 0.3. Sine wave steering, swept steering, and ramp steering maneuvers are considered to construct the dataset. The velocity is set between 20 and 120 kph, and the steering angle range is between $-120°$ and $120°$. For the sine wave steering maneuver, the frequency of the steering wheel angle is set to 0.2 or 0.5 Hz. In the swept steering scenario, the frequency is set to be varied between 0.2-0.5 Hz.

To emulate sensor data from on-board sensors, Gaussian noise with variances of $0.04°$, $0.03\ m/s$, $0.04\ °/s$, and $0.01\ m/s^2$ are added to steering wheel angle, velocity, yawrate and lateral acceleration, respectively. In addition, biases of $0.1\ °/s$ and $0.2\ m/s^2$ are added to yawrate and lateral acceleration, respectively.

B. TEST DATASET

The scenarios for the test dataset to verify the performance of the algorithm are listed in Table. 1. The first scenario includes a double lane change (DLC) maneuver at $\mu = 0.3$. In the next two scenarios, sine wave steering is performed at $\mu = 0.85$ and $\mu = 0.3$. In the 2^{nd} and 3^{rd} scenarios, the frequency of the steering wheel angle is set to 0.25 Hz, which is different from that in the training dataset. The velocity is accelerated at $1\ m/s^2$ in both scenarios. The last scenario includes a new surface condition ($\mu = 0.2$), and step steering is applied with a maneuver that is not used in the training phase.

C. RESULTS OF FILTERING NETWORK

Figs. 4 and 5 compare the results of $\dot{\beta}$ with and without the filtering network in all test scenarios. The plots include real $\dot{\beta}$ value from *CarSim* (red solid), $\dot{\beta}$ estimate without the filtering network (gray dashed) and $\dot{\beta}$ estimate with the filtering network (blue dash-single dotted). The result using the filtering network matches better with the ground truth than

TABLE 1. Simulation scenarios for test dataset.

	Scenario description	Velocity [kph]
1	DLC on slippery road (μ=0.3)	120
2	Sine steering on a dry asphalt road (μ=0.85)	70→120
3	Sine steering on a slippery road (μ=0.3)	70→90
4	Step steering on a new road surface (μ=0.2)	90

TABLE 2. Filtering network results for $\dot{\beta}$ for test scenarios in simulation.

		Scen. 1	Scen. 2	Scen. 3	Scen. 4
W/o FN	RMSE [deg]	**0.26**	0.45	0.46	0.36
	MAE [deg]	0.24	0.41	0.43	0.36
W/ FN	RMSE [deg]	**0.26**	**0.30**	**0.21**	**0.06**
	MAE [deg]	**0.20**	**0.24**	**0.16**	**0.04**

RMSE: root mean square error; MAE: mean absolute error; FN: filtering network.

the result without the filtering network, indicating a better accuracy of sensor inputs for the kinematics-based model used in our work. The performance of the filtering network is summarized in Table. 2. Metrics for validation are selected as the root mean square error (RMSE) and mean absolute error (MAE). The results with the proposed network show better performance in all test scenarios.

D. RESULTS OF SIDESLIP ANGLE ESTIMATION

In this subsection, the results of the proposed method are explained in detail. Figs. 6, 7, 8 and 9 show the results of the sideslip angle estimation. In each figure, the top graph shows the real β value obtained from *CarSim* (red solid), the estimate from the deep ensemble (black dashed), β_{DE}, and the estimate with uncertainty values (green solid), $\beta_{DE} \pm \sigma_{\beta_{DE}}$. The graph in the middle shows the β estimation results without the filtering network (blue dash-single dotted) and with the filtering network (green solid). As can be seen in the bottom graph, the estimation performance improves when the filtering network is utilized and this indicates that the filtering network improves the suitability of sensor inputs for

TABLE 3. Simulation results for β for test scenarios.

		Scen. 1	Scen. 2	Scen. 3	Scen. 4
Chindamo et al. [17]	RMSE [deg]	0.95	1.13	1.74	0.62
	MAE [deg]	0.72	0.89	1.16	0.41
Ghosh et al. [19]	RMSE [deg]	0.55	0.70	0.96	0.31
	MAE [deg]	0.43	0.62	0.52	0.26
Deep Ensemble (DE)	RMSE [deg]	0.38	0.38	1.01	0.19
	MAE [deg]	0.28	0.31	0.56	0.10
DE + UKF w/o FN	RMSE [deg]	0.25	0.22	0.30	0.25
	MAE [deg]	0.21	0.19	0.24	0.19
DE + UKF w/ FN	RMSE [deg]	**0.07**	**0.18**	**0.08**	**0.11**
	MAE [deg]	**0.06**	**0.15**	**0.06**	**0.08**

use with the kinematic model. Furthermore, the uncertainty of deep ensemble network is high at about 6 s in Fig. 9. The performance of the proposed method is improved by giving a greater weight to the UKF result. The final estimation results in terms of RMSE and MAE are summarized in Table. 3. The proposed algorithm is also compared in simulation with the other algorithms [17], [19] reported in the related literature. In general, the integrated deep ensemble and UKF provides robust estimation results for the test scenarios than various artificial networks only approaches. In addition, the results using the filtering network show better performance than the others in terms of both metrics.

V. EXPERIMENT

For experimental verification, the test vehicle is equipped with a Correvit S-Motion optical sensor from Kistler as shown in Fig. 10. The optical sensor determines the longitudinal and lateral velocities to calculate the sideslip angle precisely. All the data from the on-board sensors of steering wheel angle, velocity, yawrate and lateral acceleration is logged through the Controller Area Network (CAN) interface.

A. DATASET

For the data collection using the test vehicle, various maneuvers in a velocity range of 40 $\sim$ 130 kph and a steering wheel angle range of $-100 \sim 100°$ are conducted. For the maneuvers, lane change, sine wave steering and random steering in various speed ranges are selected. The training and test datasets consist of different scenarios. The experimental scenarios for test dataset are listed in Table. 4. The first scenario includes the random steering maneuver on dry asphalt road with the velocity range of 40 $\sim$ 60 kph. In the second scenario, DLC (Double Lange Change) maneuver is performed with steering wheel angle of $-60 \sim 60°$ at the speed of 75 $\sim$ 100 kph. For the last scenario, the slalom maneuver is performed with steering wheel angle between $-60 \sim 60°$ at the speed of 40 $\sim$ 70 kph.

B. RESULTS OF FILTERING NETWORK

The same filtering network architecture which is used in the simulation section is used for training. Fig. 11 compares the result of $\dot{\beta}$ with and without the filtering network in test scenarios 1. The result with the filtering network shows better performance than the result without the filtering network. The experimental results of the filtering network are summarized in Table. 5 in terms of RMSE and MAE. The results with the proposed network show better performance in all test scenarios.

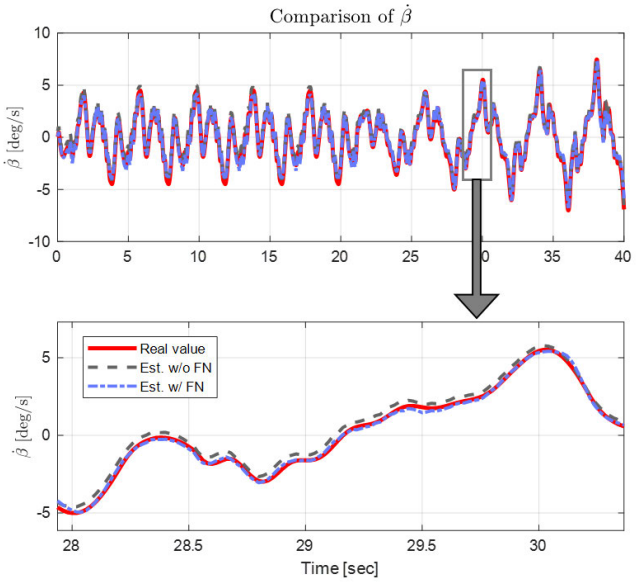

FIGURE 4. Filtering network results for $\dot{\beta}$ for test scenario 2 in simulation.

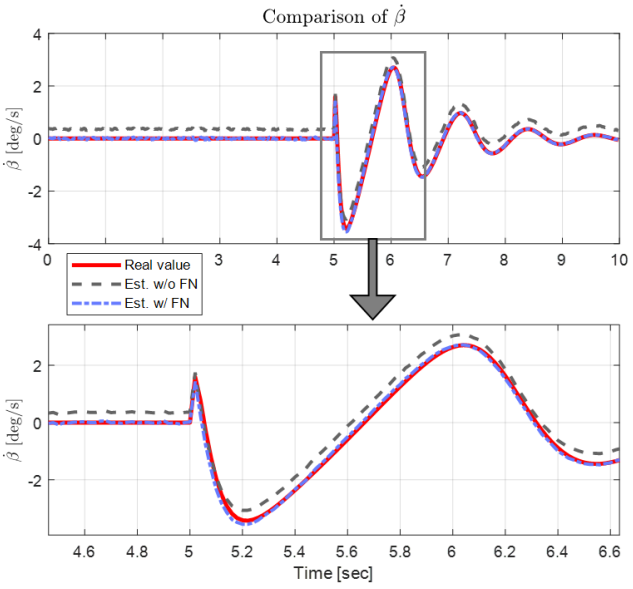

FIGURE 5. Filtering network results for $\dot{\beta}$ for test scenario 4 in simulation.

C. RESULTS OF SIDESLIP ANGLE ESTIMATION

In this subsection, the experimental result of the proposed method is explained in detail. Fig. 12 shows the result of the sideslip angle estimation. As seen in the bottom enlarged graph, the estimation performance improves when

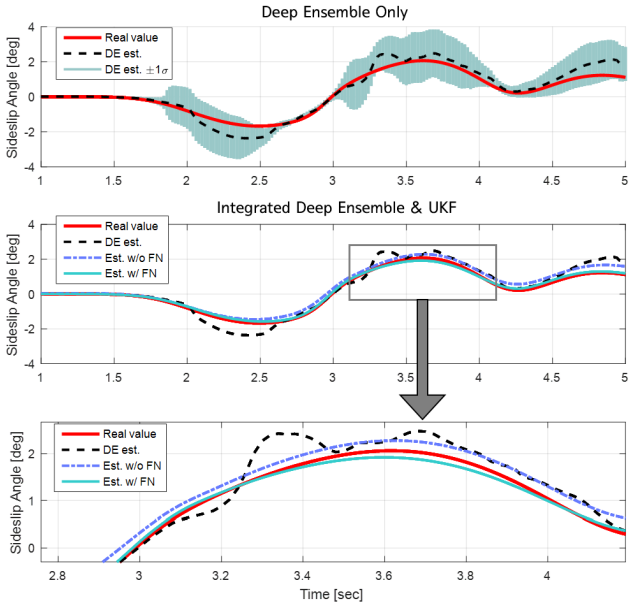

FIGURE 6. Simulation results for sideslip angle estimation for test scenario 1.

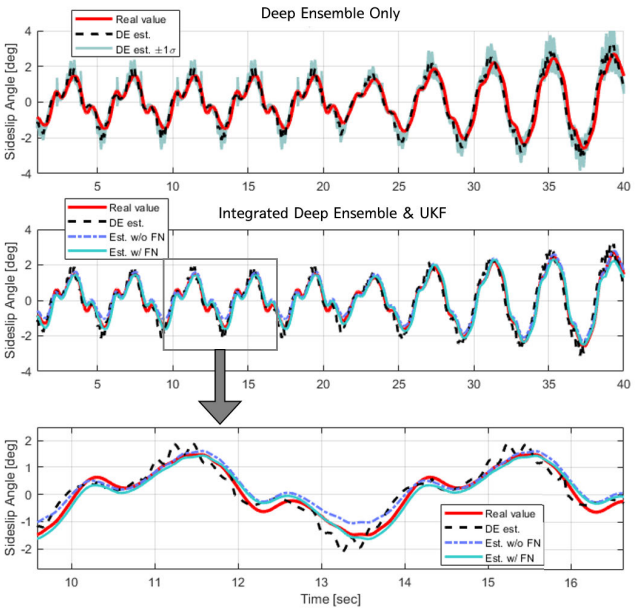

FIGURE 7. Simulation results for sideslip angle estimation for test scenario 2.

TABLE 4. Experimental scenarios for test dataset.

	Scenario description	Velocity [kph]
1	Random steering (-60∼80°) on dry asphalt road	40∼60
2	DLC (-60∼60°) on dry asphalt road	75∼100
3	Slalom maneuver (-60∼60°) on dry asphalt road	40∼70

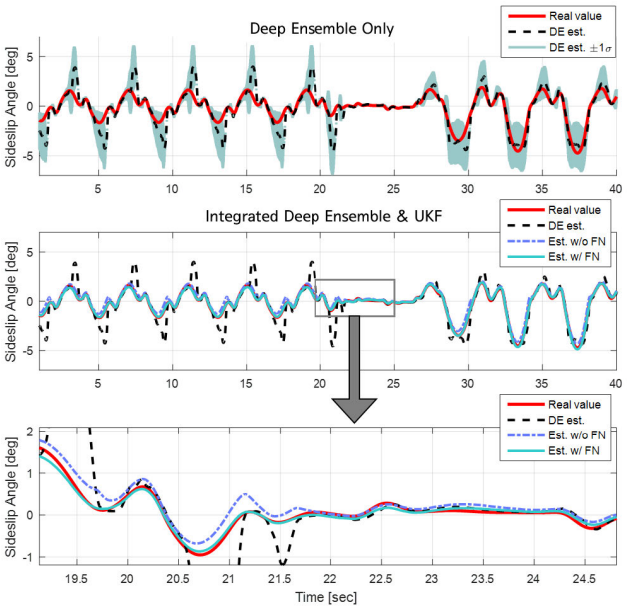

FIGURE 8. Simulation results for sideslip angle estimation for test scenario 3.

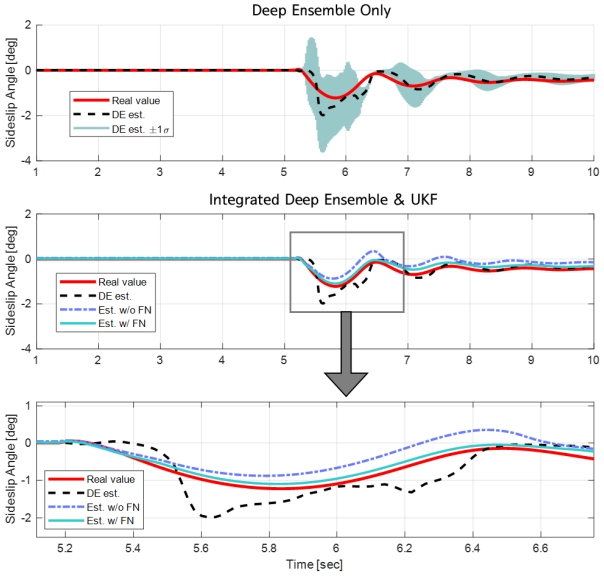

FIGURE 9. Simulation results for sideslip angle estimation for test scenario 4.

FIGURE 10. Experimental setup and acquisition system.

the filtering network is utilized. This indicates that the filtering network improves the suitability of sensor inputs for use in the kinematic model in the real-car experiment. The uncertainty of the deep ensemble is large along with an inaccurate estimate at the time about 19.7 s. The proposed method uses the uncertainty information into the adaptive

TABLE 5. Filtering network results for $\dot{\beta}$ for test scenarios in experiment.

		Scen. 1	Scen. 2	Scen. 3
W/o FN	RMSE [deg]	2.51	4.30	1.83
	MAE [deg]	1.91	3.23	1.47
W/ FN	RMSE [deg]	**0.85**	**0.87**	**0.65**
	MAE [deg]	**0.62**	**0.67**	**0.51**

TABLE 6. Experimental results for β for test scenarios.

		Scen. 1	Scen. 2	Scen. 3
Chindamo et al. [17]	RMSE [deg]	0.49	0.38	0.36
	MAE [deg]	0.41	0.27	0.31
Ghosh et al. [19]	RMSE [deg]	0.22	0.27	0.17
	MAE [deg]	0.17	0.20	0.13
Deep Ensemble	RMSE [deg]	0.25	0.28	0.37
(DE)	MAE [deg]	0.16	0.21	0.11
DE + UKF	RMSE [deg]	0.39	0.42	0.16
w/o FN	MAE [deg]	0.29	0.33	0.13
DE + UKF	RMSE [deg]	**0.19**	**0.25**	**0.12**
w/ FN	MAE [deg]	**0.15**	**0.18**	**0.09**

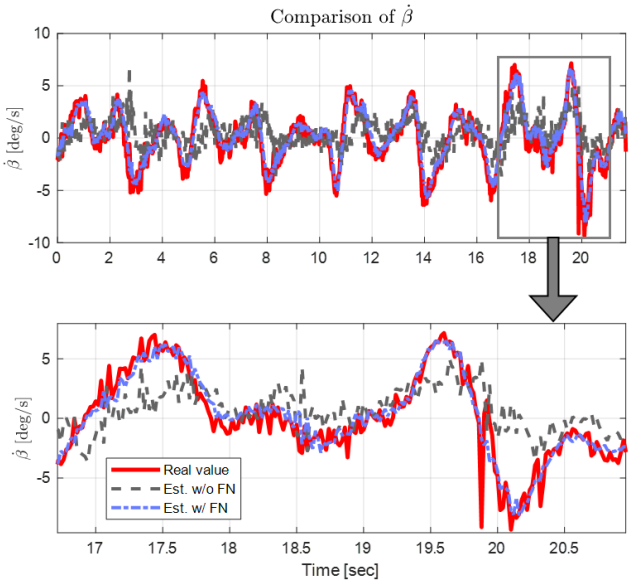

FIGURE 11. Filtering network results for $\dot{\beta}$ for test scenario 1 in experiment.

UKF, resulting in better performance. The final estimates for the sideslip angle are summarized in Table. 6. The proposed algorithm is also compared in experiment with the other algorithms [17], [19]. The experimental results demonstrate that the proposed approach integrating deep ensemble with UKF provides least errors in the sideslip angle estimation. The estimation performance without the filtering network is the worst in most cases compared to the methods using the deep ensemble or filtering network. This indicates that using the raw sensor data to a simple kinematic model in the UKF provides large error in the estimation result.

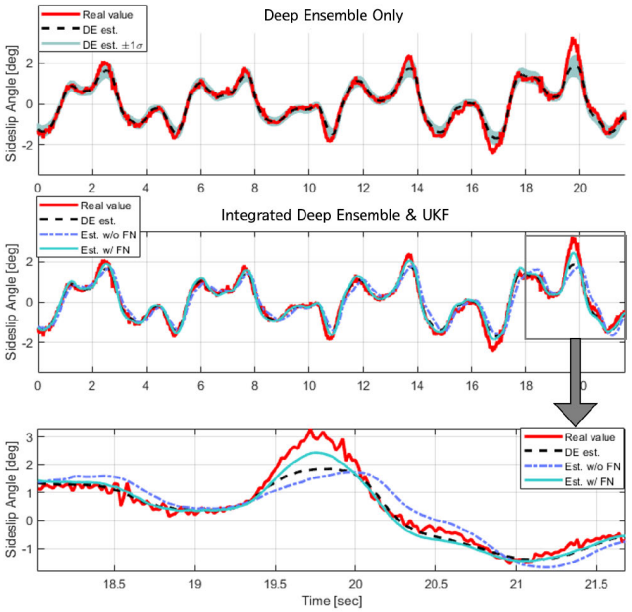

FIGURE 12. Experimental results for sideslip angle estimation for test scenario 1.

The proposed method is run with an average computation time of 15 ms on a PC (Intel(R) Core(TM) i7 at 4.00 GHz, 16.00-GB memory).

VI. CONCLUSION

This paper presents a novel framework integrating deep learning-based and model-based methods to estimate the vehicle sideslip angle. A sensor filtering network is designed to ensure that the input sensor signals work well for the kinematics-based model. Then, a deep ensemble network provides a sideslip angle estimate and its uncertainty. The estimate and its uncertainty from the network are utilized as measurement inputs of the UKF. The proposed algorithm is first validated in a simulation by means of *CarSim*. Then, a real-car experiment is also carried out using an optical sensor for validation purpose. Both simulation and experimental results demonstrate that the proposed approach shows better performance in sideslip angle estimation than deep ensemble only and integrated method without the filtering network. Future work will validate this method under more diverse driving maneuvers and road conditions.

REFERENCES

[1] M. Corno, G. Panzani, and S. M. Savaresi, "Single-track vehicle dynamics control: State of the art and perspective," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 4, pp. 1521–1532, Aug. 2015.
[2] Y. Suzuki and M. Takeda, "An overview on vehicle lateral dynamics and yaw stability control systems," *J. Adv. Vehicle Eng.*, vol. 2, no. 4, pp. 182–190, 2016.
[3] L. Li, Y. Lu, R. Wang, and J. Chen, "A three-dimensional dynamics control framework of vehicle lateral stability and rollover prevention via active braking with MPC," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 3389–3401, Apr. 2016.
[4] J.-S. Jo, S.-H. You, J. Y. Joeng, K. I. Lee, and K. Yi, "Vehicle stability control system for enhancing steerabilty, lateral stability, and roll stability," *Int. J. Automot. Technol.*, vol. 9, no. 5, p. 571, 2008.

[5] D. Piyabongkarn, R. Rajamani, J. A. Grogg, and J. Y. Lew, "Development and experimental evaluation of a slip angle estimator for vehicle stability control," *IEEE Trans. control Syst. Technol.*, vol. 17, no. 1, pp. 78–88, Jan. 2008.
[6] J. Liu, Z. Wang, L. Zhang, and P. Walker, "Sideslip angle estimation of ground vehicles: A comparative study," *IET Control Theory Appl.*, vol. 14, no. 20, pp. 3490–3505, Dec. 2020.
[7] R. Song and Y. Fang, "Vehicle state estimation for INS/GPS aided by sensors fusion and SCKF-based algorithm," *Mech. Syst. Signal Process.*, vol. 150, Mar. 2021, Art. no. 107315.
[8] M. K. Mazhar, M. J. Khan, A. I. Bhatti, and N. Naseer, "A novel roll and pitch estimation approach for a ground vehicle stability improvement using a low cost IMU," *Sensors*, vol. 20, no. 2, p. 340, Jan. 2020.
[9] H. Guo, D. Cao, H. Chen, C. Lv, H. Wang, and S. Yang, "Vehicle dynamic state estimation: State of the art schemes and perspectives," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 2, pp. 418–431, 2018.
[10] R. M. Brach and R. M. Brach, "Modeling combined braking and steering tire forces," SAE, Tech. Paper 2000-01-0357, 2000.
[11] H. Du, N. Zhang, and G. Dong, "Stabilizing vehicle lateral dynamics with considerations of parameter uncertainties and control saturation through robust yaw control," *IEEE Trans. Veh. Technol.*, vol. 59, no. 5, pp. 2593–2597, Jun. 2010.
[12] D.-H. Lee, I.-K. Kim, and K.-S. Huh, "Tire lateral force estimation system using nonlinear Kalman filter," *Trans. Korean SoC. Automot. Eng.*, vol. 20, no. 6, pp. 126–131, Nov. 2012.
[13] J. Kim, "Effect of vehicle model on the estimation of lateral vehicle dynamics," *Int. J. Automot. Technol.*, vol. 11, no. 3, pp. 331–337, Jun. 2010.
[14] K. B. Singh, "Vehicle sideslip angle estimation based on tire model adaptation," *Electronics*, vol. 8, no. 2, p. 199, Feb. 2019.
[15] F. D. Biase, B. Lenzo, and F. Timpone, "Vehicle sideslip angle estimation for a heavy-duty vehicle via extended Kalman filter using a rational tyre model," *IEEE Access*, vol. 8, pp. 142120–142130, 2020.
[16] S. Melzi and E. Sabbioni, "On the vehicle sideslip angle estimation through neural networks: Numerical and experimental results," *Mech. Syst. Signal Process.*, vol. 25, no. 6, pp. 2005–2019, Aug. 2011.
[17] D. Chindamo and M. Gadola, "Estimation of vehicle side-slip angle using an artificial neural network," in *Proc. MATEC Web Conf.*, vol. 166, 2018, p. 02001.
[18] B. L. Boada, M. J. L. Boada, A. Gauchía, E. Olmeda, and V. Díaz, "Sideslip angle estimator based on ANFIS for vehicle handling and stability," *J. Mech. Sci. Technol.*, vol. 29, no. 4, pp. 1473–1481, Apr. 2015.
[19] J. Ghosh, A. Tonoli, and N. Amati, "A deep learning based virtual sensor for vehicle sideslip angle estimation: Experimental results," SAE, Tech. Paper 2018-01-1089, 2018.
[20] T. Graber, S. Lupberger, M. Unterreiner, and D. Schramm, "A hybrid approach to side-slip angle estimation with recurrent neural networks and kinematic vehicle models," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 1, pp. 39–47, Mar. 2019.
[21] A. Bonfitto, S. Feraco, A. Tonoli, and N. Amati, "Combined regression and classification artificial neural networks for sideslip angle estimation and road condition identification," *Vehicle Syst. Dyn.*, vol. 58, no. 11, pp. 1766–1787, Nov. 2020.
[22] D. Kim, K. Min, H. Kim, and K. Huh, "Vehicle sideslip angle estimation using deep ensemble-based adaptive Kalman filter," *Mech. Syst. Signal Process.*, vol. 144, Oct. 2020, Art. no. 106862.
[23] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6402–6413.
[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[25] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," 2017, *arXiv:1704.06053*.
[26] D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 1, Jul. 1994, pp. 55–60.
[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[28] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
[29] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://tensorflow.org/

DONGCHAN KIM received the B.S. degree in automotive engineering from Hanyang University, Seoul, Republic of Korea, in 2015, where he is currently pursuing the Ph.D. degree with the Department of Automotive Engineering. His research interests include motion planning, trajectory prediction, vehicle state estimation, reinforcement learning, and deep learning.

GIHOON KIM received the B.S. degree in automotive engineering from Hanyang University, Seoul, Republic of Korea, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Automotive Engineering. His research interests include interaction-aware driving, which is planning the path considering effects to other vehicles from ego vehicle's movement and the development of control system for collision avoidance using aggressive-maneuver driving like drift control.

SEUNGWON CHOI received the B.S. degree in automotive engineering from Hanyang University, Seoul, Republic of Korea, in 2019, where he is currently pursuing the Ph.D. degree with the Department of Automotive Engineering. His research interests include trajectory prediction, motion planning, localization, and deep learning.

KUNSOO HUH (Member, IEEE) received the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 1992. He is currently a Professor with the Department of Automotive Engineering, Hanyang University, Seoul, South Korea. His research interests include machine monitoring and control, with emphasis on their applications to vehicular systems. His current research interests include sensor-based active safety systems, V2X-based safety systems, autonomous vehicle control, and AI applications in autonomous vehicle.