IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 8, 2021, accepted June 16, 2021, date of publication June 21, 2021, date of current version June 29, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3090907*

# Curriculum Learning for Vehicle Lateral Stability Estimations

**JIHWAN BAE, TAEKYUNG KIM, WONSUK LEE, AND INWOOK SHIM**
Ground Technology Research Institute, Agency for Defense Development, Daejeon 34186, Republic of Korea
Corresponding authors: Wonsuk Lee (wsblues@add.re.kr) and Inwook Shim (iwshim@add.re.kr)

This work was supported by the Agency for Defense Development (ADD).

**ABSTRACT** Precise estimations of the roll and sideslip angles of autonomous vehicles are essential for autonomous driving, which requires further information about the vehicle state. As such, novel deep learning approaches have been introduced for this purpose. However, the majority of deep learning works focusing on vehicle dynamics estimations have yet to delve into learning strategies specifically for this task. Here, we argue that simply applying an adequate learning strategy to the task can boost the estimation performance. In this paper, we propose a simple yet effective curriculum learning strategy for better estimations of the roll and sideslip angles simultaneously. In addition, we compare our curriculum using a self-taught scoring function with a curriculum sorted by prior human knowledge, demonstrating its superiority. The proposed method outperforms the non-curriculum method by a large margin (up to a 16.5% decrease for sideslip as validation and 3.7% on a test), especially with regard to cornering (up to a 4% decrease).

**INDEX TERMS** Curriculum learning, deep learning based estimator, roll angle, sensor fusion, sideslip angle, vehicle pose estimation.

## I. INTRODUCTION

Currently, autonomous vehicles are actively being developed by researchers worldwide, and numerous real road driving trials are underway. Thus, the importance of precise estimates of vehicle states is crucial, in order to design controllers that provide meticulous data for all driving conditions. It is well established in the literature that sideslip and roll angles are the most critical parameters of vehicle lateral stability, whereas the high cost of the sensors deployed to measure these values physically is not feasible when building autonomous vehicles. Alternatively, most research sidesteps this by estimating them. The Extended Kalman Filter (EKF) and H-infinity-based approaches to estimate vehicle lateral stability [1]–[3] have succeeded in mitigating the underlying non-linearity of the vehicle dynamics systems. Still, conventional approaches cannot fully address this type of non-linearity. The advent of deep learning has allowed researchers to introduce more robust outcomes from such non-linearity [4]–[8]. Neural networks used to estimate vehicle lateral stability are complex models, but the simple Fully Connected (FC) network has also proven to function well in estimating the sideslip and roll angles simultaneously [9], [10]. Diverse neural network architectures have been introduced to enhance the accuracy of the vehicle model, where adequate learning strategies are yet to be explored for vehicle lateral stability estimation tasks. Apart from choosing the most effective neural network architecture for the given task, an appropriate learning strategy is significant. By applying a suitable learning strategy, one can realize better performance from the same neural network backbone.

We observed that one of the major characteristics of vehicle state estimation problems is aggregation of diverse driving data from different settings. Specifically, these different settings differ by vehicle speed, maneuvers, road friction, and many other possible factors that affect the driving. This indicates the underlying congenital difficulty exists for each driving data, and we were highly motivated that if the data used for training the neural network is reordered in ideal curriculum sorted by adequate difficulty, the estimation performance of the neural network can be boosted. Humans and animals learn much better when the data is given in a meaningful order rather than being random shuffled; this type of a training strategy to learn the model is called curriculum learning [11], [12]. The major aspects of the superiority of curriculum learning [12], compared to other learning strategies, is that re-ordering the sequence of data with difficulty benefits the model by gradually capturing more complex concepts, thereby outperforming on the entire spectrum of the data.

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

The contribution of this work can be summarized as:

- We propose designing a curriculum learning strategy with task-specific scoring and pacing functions for vehicle lateral stability estimation.
- We attain robust estimation on nonlinear dynamics, *e.g.*, upon on large cornering movements
- We conduct empirical investigation of different curriculum conditions
- We analyze and evaluate our proposed curriculum learning strategy against human-prior curriculum and expert model, and show the improvement of the vehicle lateral stability in various driving scenarios.

This paper is organized as follows: Section II presents the related works from the literature regarding curriculum learning and vehicle stability estimation. Section III defines the theoretical formulation of the proposed method. Section IV introduces the dataset configuration along with the data logging system. Section V shows the experiment results and further investigations we conducted, followed by Appendix A and B. Finally, the recapitulation for the contributions of the work is summarized in Section VI.

## II. RELATED WORK

### A. CURRICULUM LEARNING

Curriculum learning [12] is a recent idea in the machine learning field, where a curriculum is configured based on ranking data with difficulty measures. These measurements are not likely to be found in the real world and are challenging to elicit from humans. Several approaches [13], [14] have proposed novel ideas to handle such issues. Self-paced learning, introduced by Kumar *et al*. [13], is a novel method in which sample selection appears during the training process to increase the level of difficulty. Nonetheless, the absence of a facilitator in self-paced learning does not guarantee the credibility of the difficulty measurements occurring during the training process. Hacohen and Weinshall [14] suggest two different strategies, knowledge transfer as in transfer learning for curriculum learning [15] and bootstrapping based on self-tutoring. Both methods show promising results, but the knowledge transfer method that uses another network pre-trained on different tasks to provide the ranking of the training data via the presumed difficulty is not ideal for our problem. Our task consists of diverse driving conditions that are compatible and comprehensive enough with other simulator-based tasks from research on vehicle dynamics stability. Therefore, it is unrealistic to acquire a new set of data with which to train a network that will be solely used for knowledge transfer based on difficulty measurements. The bootstrapping-based self-taught method trains the network without a curriculum and then uses the resulting network as a scorer to rank the training data to train the same network again, but with a curriculum. No additional resources are required for a self-taught scoring function. Thus, our scoring function is formulated based on the self-taught method.

### B. VEHICLE STABILITY ESTIMATION

There are numerous conventional approaches that focus on model-based estimators using a determined vehicle dynamics model [16], [17]. However, such approaches are vulnerable to predictions of the lateral states of vehicle driving on various friction surfaces and rough terrains, as the vehicle dynamics is innately nonlinear. More recent studies have applied an observer-based estimator to alleviate the nonlinear motions [1]–[3]. Meanwhile, artificial neural networks have been proven to perform better than observer-based approaches in vehicle lateral stability estimations with both linear and nonlinear features [9], [10]. Along with the development of high-fidelity dynamic simulators, it has become straightforward to acquire precise driving data. Accordingly, data-driven neural network estimators are becoming more actively studied.

Enlarging the neural network architecture and increasing the complexity of neural networks are common approaches to improve the accuracy of many other tasks, while these strategies are not applicable to vehicle stability estimations. It is unrealistic to deploy high computation resources in commercial vehicles. We believe that it is possible to improve the accuracy levels by simply applying a task-specific learning strategy instead.

### C. CURRICULUM LEARNING FOR MULTI-LAYER PERCEPTRON

In the field of imitation learning and policy gradient reinforcement learning, a fully-connected Multi-Layer Perceptron (MLP) is commonly employed as a policy model [18]–[20]. It has been demonstrated that training the models and agents in an ascending order of data difficulty can improve the overall performance in numerous works [21], [22]. Lee *et al.* [23] use an adaptive terrain curriculum while training the model with the knowledge from a teacher policy, updating the curriculum for every iteration of the policy. Moreover, robot navigation with deep reinforcement learning [24] increases the complexity of the environment in multiple stages. Moreover, autonomous overtaking in game simulator with curriculum reinforcement learning has already shown power of curriculum learning in simulation-based data [25].

We insist that curriculum learning in MLP-based vehicle states estimation tasks can also be beneficial because these tasks resemble the concepts in previous works. A neural network model of high-performance automated driving [9] was trained in a manner similar to curriculum learning. According to previous works that applied curriculum learning to similar tasks with vehicle stability estimation, we can expect promising results when adopting the curriculum learning strategy for our problem.

## III. APPROACH

Two main criteria must be considered when designing a curriculum: ranking the data in a proper order that reflects

difficulty or complexity (if the order is not well configured, the neural network cannot benefit from the idea that knowledge obtained while learning easy samples contributes to the learning of difficult samples), and the pacing rate at which the data is presented to the model; a moderate pacing rate is important because going over the simple data too rapidly may reduce the benefit, while the opposite may lead to overfitting the model to the simple data. In this work, we propose two separate but closely related functions for each criteria. First, a *scoring function*, determines the difficulty of the data and generates a curriculum that presents the easier data first to the network. Scoring is done based on a self-tutoring method [14]. The second function, termed a *pacing function*, determines the rate of data from different levels of difficulty being presented to the network.

Let $\mathbb{X} = \{\mathbb{X}_i\}_{i=1}^{N} = (x_i, y_i)_{i=1}^{N}$ denotes the data, where $x_i$ represents a single data point consisting of different sensor data; $y_i$ denotes its corresponding set of sideslip and roll angles, and $N$ denotes the total number of driving scenarios. Our curriculum learning setting is slightly different with curriculum learning works for classification tasks. Previous curriculum learning [14], [15] works sample each mini-batch $\mathbb{B}_i \subseteq \mathbb{X}$ uniformly from $\mathbb{X}$ and a list of these sampled mini-batches is used as the curriculum in every single epochs. Alternatively, we designed the curriculum learning to occur gradually during all training epochs. The MLP used in this work is a fully-connected layer which is deployed with fine regression. As a MLP is frequently used in reinforcement learning [26], [27], it is more likely to show feasible results when the curriculum settings from reinforcement learning are followed, where curriculum learning proceeds during all training epochs.

Our proposed algorithm for curriculum learning is presented here as Algorithm 1. In order to demonstrate the superiority of curriculum learning over a non-curriculum method, we devised two different pacing functions and show that both of them outperform a naive approach. Also, we defined a curriculum based on human prior knowledge and compared with our curriculum using a self-taught scoring function (see Section V-A3). Lastly, we examined an anti-curriculum condition, where samples ranked by scoring function $\mathbf{S}$ are sorted in descending order, and a random curriculum condition where samples are randomly sorted. These results are shown in Section V-B2.

### A. SCORING FUNCTION

The scoring function is defined as any function $\mathbf{S} : \mathbb{X} \rightarrow \mathbb{R}$, and it should have the following property.

$$D(x_i, y_i) > D(x_j, y_j) \;\; if \;\; \mathbf{S}(x_i, y_i) > \mathbf{S}(x_j, y_j), \tag{1}$$

where $D(x_i, y_i)$ denotes the ground-truth difficulty of the example $(x_i, y_i)$. In order to design such a scoring function, we initially train an identical network with the full set of data (self-teacher). For the loss function, we used the root mean square error (RMSE). The same loss function is used to train our neural network for curriculum learning. The loss function

**Algorithm 1**

**Input:** data $\mathbb{X}$, scoring function $\mathbf{S}$, pacing function $\mathbf{P}$, number of driving scenarios being added for each step $\theta$, number of driving scenarios $N$

sort $\mathbb{X} \rightarrow [\mathbb{X}'_1, \ldots, \mathbb{X}'_N]$ in ascending order based on $\mathbf{S}$

**for each epoch** $e$ **do**

$\mathbb{X}_e \leftarrow []$

$k = \min(\mathbf{P}_\theta(e), N)$

$\mathbb{X}_e \leftarrow [\mathbb{X}'_1, \ldots, \mathbb{X}'_k]$

train model with data subset $\mathbb{X}_e$

**end for**

is computed as follows:

$$Loss = \sqrt{\frac{1}{M}\sum_{i=0}^{M-1}\big(\hat{y}(i) - y(i)\big)^2}, \tag{2}$$

where $\hat{y}(i)$ and $y(i)$ are the estimated and the ground truth roll and sideslip angles, respectively, and $M$ is the batch size.

Our self-taught scoring function $\mathbf{S}(\mathbb{X}'_k)$ for each driving scenario $\mathbb{X}'_k$ is defined as follows:

$$\mathbf{S}(\mathbb{X}'_k) = \left\{(x(i), y(i)) \in \mathbb{X}'_k \mid \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(g(x(i)) - y(i))^2}\right\}, \tag{3}$$

where $n$ denotes the number of data in driving scenario $\mathbb{X}'_k$ and $g(x(i))$ denotes the inference result with the given input $x(i)$ of the trained teacher model, previously trained with the non-curriculum method. Thus, scores defined by scoring function $\mathbf{S}(\mathbb{X}'_k)$ encode the supervision from the self-teacher [14]. For each driving scenario $\mathbb{X}'_k$, the RMSE between the inference result of the teacher model given input $x(i)$ and ground-truth $y(i)$ where $(x(i), y(i)) \in \mathbb{X}'_k$ is defined as $\mathbf{S}(\mathbb{X}'_k)$. The smaller $\mathbf{S}(\mathbb{X}'_k)$ is, the better the teacher model predicts on $\mathbb{X}'_k$. This can be interpreted as meaning that if $\mathbf{S}(\mathbb{X}'_p) < \mathbf{S}(\mathbb{X}'_q)$, then driving scenario $\mathbb{X}'_p$ is more easier than $\mathbb{X}'_q$ in terms of the model.

### B. PACING FUNCTION

For the pacing function, we adopt single-step pacing to input data from easy driving scenarios during the early stages and gradually expand the range of the data to difficult driving scenarios in later stages.

The pacing function $\mathbf{P}$ controls the sequence of the driving data being input to the learner model for each epoch $e$. A sequence of subsets $[\mathbb{X}'_1, \ldots, \mathbb{X}'_k] \rightarrow \mathbb{X}_e$ is determined, where $k = \mathbf{P}_\theta(e)$. Thus, proposed pacing function $\mathbf{P}$ is given as:

$$\mathbf{P}_\theta(e) = \theta \times (1 + \left\lfloor \frac{e}{\theta} \right\rfloor), \tag{4}$$

where $\theta$ denotes number of driving scenarios to be added incrementally for the steps, $e$ is the current epoch and $\lfloor \frac{e}{\theta} \rfloor$ refers to the rounded down value of $\frac{e}{\theta}$. To elaborate, we can

say that the initial setting is given as $\theta = 2$. When training starts ($e = 0$), $\mathbf{P}_\theta(e)$ is equal to 2. Thus, the model starts training with data from the first two driving scenarios of $\mathbb{X}$ until $e = 1$. If $e$ becomes six, training data is brought from first eight driving scenarios of $\mathbb{X}$. Eventually, $\mathbf{P}_\theta(e)$ will be larger than the total number of driving scenarios $N$ and will be saturated as $N$ in Algorithm1. After saturation, model will be trained on the $\mathbb{X}$ overall during the remaining epochs. The effect of varying $\theta$ in the training models is described in Section V-A4.

### C. BACKBONE ARCHITECTURE

Our task aims to train the model with the data obtained from the simulator and achieve decent performance during real-world driving, often referred to as 'Sim2Real' in the literature. Physical sensor data on an actual real platform are noisy enough to produce non-negligible errors; thus, neural networks that recurrently process data for predictions are not adequate for our task [28]. Additional modules such as an observer based on filtering combined with a neural network [3] show robust performance on vehicle roll angle predictions. However, filter-based approaches are usually associated with a phase delay, meaning that it is undesirable to attach extra filters onto a neural network because real-time estimations are required on the vehicle.

Considering the characteristics of our task, a MLP is used in this work. The MLP has shown reliable results in several works that perform accurate predictions [29], [30] and does not accumulate errors derived from the previous estimations. Figure 1 illustrates our FC model, which is referenced from a backbone neural network in a deep learning model for mass production vehicles [10]. We denote $a_x$ and $a_y$ as the longitudinal and lateral acceleration variables, $\dot{\phi}$ as the roll rate, $\dot{\psi}$ as the yaw rate, $\delta$ as the steering angle and $v_x$ as the longitudinal velocity of the vehicle body. The selected input states are essential values for the neural network to derive the roll and sideslip angle, and they are able to be accurately measured on low-cost sensors, including IMU and GPS, in real time.

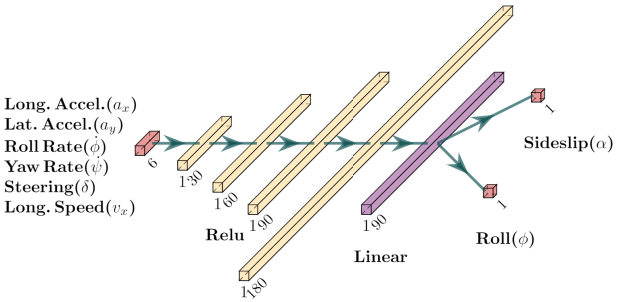



**FIGURE 1.** Backbone architecture.

## IV. DATASETS

### A. DATA LOGGING SYSTEM

The training dataset should include diverse driving scenarios that could represent comprehensive vehicle conditions during driving. It is difficult to obtain data from aggressive driving in an actual vehicle because drifting on a road with low friction is dangerous, while such training data are crucial for estimating vehicle states under unstable conditions. Therefore, the dataset should be obtained from a vehicle simulator, where such unrealistic driving data can be acquired. In this work, we use CarMaker, which solves nonlinear equations based on high-fidelity vehicle mathematical models in real-time. This approach allows users to construct the environments and modify the roads and maneuvering conditions quickly.

The simulator provides an interface by which to gain access to the simulated vehicle states. We built a logging system based on a Robot Operating System (ROS) [31] during a real-time simulation (see Figure 2). We employed ROS2 in this work. We manually drove the simulated vehicle with Logitech G29 racing wheel and pedals to obtain validation data. The data were logged in the CSV format at a frequency of 20 $Hz$.

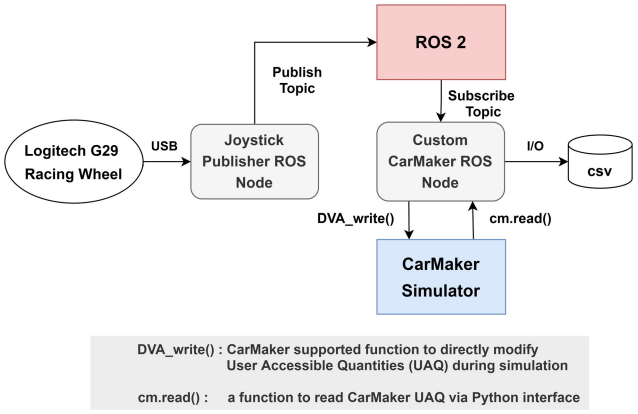



**FIGURE 2.** CarMaker data logging system with ROS2.

### B. TRAIN & TEST DATASET

We used the default demo car model, and the scheduled maneuvers were driven using the built-in controller of CarMaker. Three maneuver scenarios are employed in simulation: lane change, J-turn, and slalom maneuvers, all of which are commonly used in experiments on vehicle dynamics [4], [5], [32]. The lane change scenario contains three double lane-change maneuvers. The driven road length during each lane change is varied, with values used here of 25 $m$, 20 $m$, and 15 $m$. For the J-Turn, the vehicle turns left after accelerating to the target speed and performs a successive right turns after recovering its stability. Two slalom driving maneuvers with 18 $m$ and 36 $m$ interval lengths are employed to enhance the instability severity.

The road friction coefficient is varied from 0.4 to 1.0, and the target speed is varied from 20 $km/h$ to 100 $km/h$. Given that the training and testing data must be carefully managed to avoid unintended bias, some of the logged data is discarded. Constant vehicle states from exceptionally slow driving in some maneuvers are meaningless. Such data will account for

**TABLE 1. Specific dataset composition.**

| $\mu$ | Maneuvers | Detail Description | Speed [km/h] | Steering Angle [deg] |
|---|---|---|---|---|
| 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 | Lane change | distance between each lane change point: $150m$ | 30, 40, 50, 60 | - |
| | | distance between each lane change point: $250m$ | 70, 80, 90, 100 | - |
| | J-turn | - | 20, 30, 40, 50, 60, 70, 80 | 20, 40, 60, 80, 100, 120 |
| | Slalom 18 m | vehicle starts $250m$ away from the origin | 20, 30, 40 | - |
| | | vehicle starts $100m$ away from the origin | 50, 60, 70 | - |
| | Slalom 36 m | vehicle starts $300m$ away from the origin | 30 | - |
| | | vehicle starts $250m$ away from the origin | 40, 50, 60 | - |
| | | vehicle starts $200m$ away from the origin | 70, 80 | - |

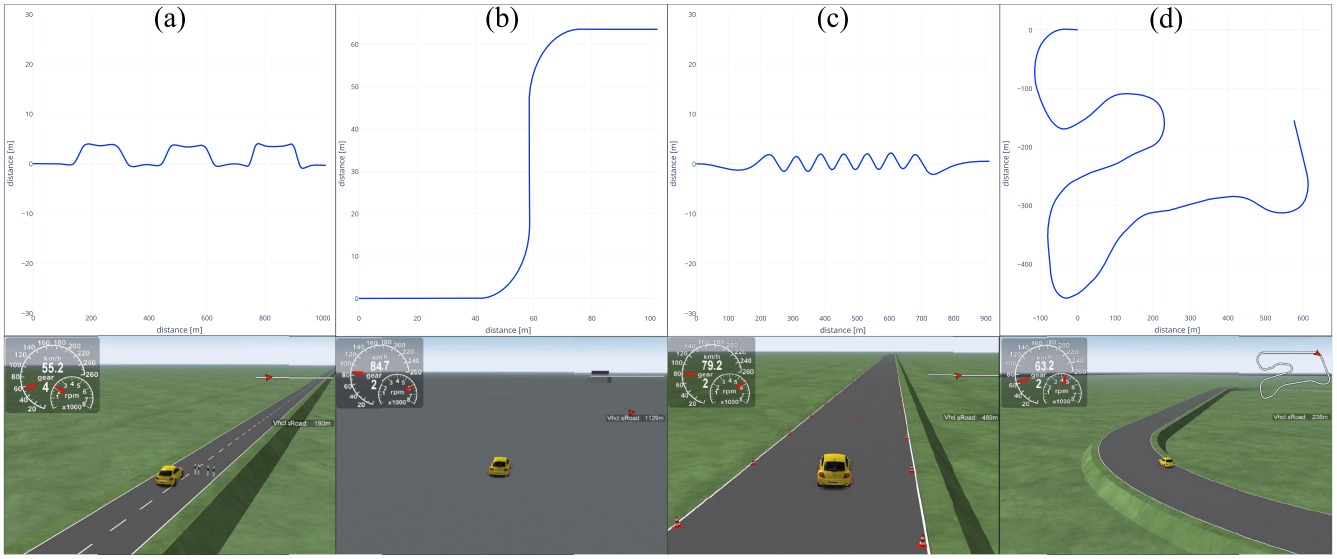



**FIGURE 3. Dataset acquisition using CarMaker. (a) Double lane change maneuvers. (b) J-turn maneuvers. (c) 36m slalom maneuvers. (d) A race track for the validation dataset.**

a large portion of all data because the vehicle needs more time to finish the course at a slow speed as compared to a faster speed. If these data logged from slow driving are not manually discarded, it leads to the overfitting of the neural network. For example, 20 $km/h$ maneuvers during a lane change were discarded. Consequently, 20 $km/h$ maneuvers were not included in the 36 $m$ length slalom scenario. We also adjusted the start point of each maneuver in order to reduce straight driving at a constant speed. Some extreme cases where the car entirely lost its stability, such as fishtailing, were also removed, as they are not reliable data to train the network. Table 1 shows the specific information of our dataset.

The variables listed in Table 1 are parameterized in the simulator, and all of those maneuvers were automatically logged by reserving the range of the parameters. We simulated a total of 434 maneuvers, 335.5 minutes of driving, and 406, 392 data.

### C. VALIDATION DATASET

The validation dataset was obtained in a different environment in CarMaker. We chose a racing track for validation data which is exclusive to the training and testing dataset. The starting and finishing points are also properly adjusted to avoid long straight driving at a constant speed, as such redundant data harms the reliability of the validation performance. Validating the trained neural network on an immense amount of easy data shows good performance regardless of the knowledge network learned during the training. We also chose the new road friction coefficients to evaluate the neural network with the unseen data. The same vehicle model was used, but it was driven by two different drivers: the built-in controller in the simulator and a skilled human driver. The skilled human driver attempted to make as large a sideslip angle as possible while maintaining the desired speed at the corners. This ensures the diversity of unsteady states in the validation dataset, allowing the trained neural networks to be evaluated in comprehensive cases. The driven courses of the training and validation data are shown in Figure 3. The Table 2 shows the detailed information of the validation

**TABLE 2. Specific validation dataset composition. We denote clockwise as CW and counter-clockwise as CCW.**

| $\mu$ | Driver | Orientation | Speed [km/h] |
|---|---|---|---|
| 0.45, 0.65, 0.85 | Built-in controller | CW | 40, 60, 80 |
| | | CCW | 40, 60, 80 |
| | Human | CW | 40, 60, 80 |
| | | CCW | 40, 60, 80 |

dataset. We simulated 36 maneuvers, 81.6 minutes of driving, and 99, 356 data in total.

## V. RESULTS AND DISCUSSION

### A. EXPERIMENTAL SETTINGS

Our experiment setting for neural network training is quite simple. The neural network was trained with a batch size of 32 and a learning rate of 0.0001. It was initialized with the He initializer [33]. The RMSE loss is used as the loss function, with the rectified Linear Unit (ReLu) [34] for activation in hidden layers.

#### 1) DATASET CONFIGURATION

With regard to vehicle lateral stability, especially for the sideslip angle, road surface friction is one of the most dominant factors. The tire cornering stiffness is an important parameter that determines how much the vehicle can resist lateral force during cornering. It is decreased in a low tire-road friction coefficient environment, meaning that the vehicle has a larger sideslip angle with the same lateral tire force than in a high friction environment [35]. In the case of a vehicle driving aggressively on a low-friction road conditions such as wet asphalt or dirt, it will tend to drift with nonlinear dynamics more easily, making it more difficult to estimate the stability [36]. Chen *et al.* [6] experimentally showed that a slight variation in the friction coefficient has a major impact on the sideslip angle. Accordingly, we divide the acquired dataset into 14 driving scenarios on the basis of the road friction and the vehicle speed. In our setting, speed greater than or equal to 50 $km/h$ is classified as fast, with lower speeds considered to be slow. The threshold was empirically determined in order to maintain a data balance between classes [14]. Then, data in each class were split into 70% for training and 30% for testing.

#### 2) ERROR METRIC

We use two error metrics for the performance evaluation: $\mathbf{E}_{RMS}$ and $\mathbf{E}_{max}$.

$$\mathbf{E}_{RMS} = \sqrt{\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\big(\hat{y}(i,j) - y(i,j)\big)}, \tag{5}$$

$$\mathbf{E}_{max} = \{i \in (0,1,\ldots,M-1), j \in (0,1,\ldots,N-1) \mid \max\big(\big|\hat{y}(i,j) - y(i,j)\big|\big)\}, \tag{6}$$

where $\hat{y}(i,j)$ and $y(i,j)$ are the estimated and the ground truth roll and sideslip angles, respectively, and $M$ and $N$ are the number of batches and the mini-batch size. $\mathbf{E}_{RMS}$ and $\mathbf{E}_{max}$ are commonly used error metrics in vehicle lateral stability estimation tasks [2], [10].

#### 3) SCORING FUNCTION

The scoring function is based on Equation 3. The self-taught scoring function measures the difficulty of each driving scenario. However, it is also possible for a human to empirically rank the driving scenarios according to the level of difficulty with their prior knowledge of vehicle dynamics and real-world driving experience. The two possible candidates are shown in Table 3. In general cases, the vehicle becomes unstable when the friction of the road surface is low and the speed of the vehicle is high. According to common sense, it is natural to assume that unstable vehicle states are more difficult to estimate than stable ones. Therefore, we can come up with two possible candidates in terms of a human considering two main criteria applicable to vehicle instability. The first candidate assumes the effect of the friction coefficient to be greater than the vehicle speed in driving, whereas the second candidate assumes the opposite. The curriculum generated by our scoring function **S** ranks driving scenarios similarly with the second candidate but is slightly different.

**TABLE 3. The two possible curriculum candidates according to human prior knowledge, and a curriculum devised by the scoring function S.**

| Difficulty | Candidate 1 | Candidate 2 | Measured by S |
|---|---|---|---|
| ↑ most difficult | $\mu$: 0.4, Fast | $\mu$: 0.4, Fast | $\mu$: 0.6, Fast |
| | $\mu$: 0.4, Slow | $\mu$: 0.5, Fast | $\mu$: 0.7, Fast |
| | $\mu$: 0.5, Fast | $\mu$: 0.6, Fast | $\mu$: 0.8, Fast |
| | $\mu$: 0.5, Slow | $\mu$: 0.7, Fast | $\mu$: 0.4, Fast |
| | $\mu$: 0.6, Fast | $\mu$: 0.8, Fast | $\mu$: 0.5, Fast |
| | $\mu$: 0.6, Slow | $\mu$: 0.9, Fast | $\mu$: 1.0, Fast |
| | $\mu$: 0.7, Fast | $\mu$: 1.0, Fast | $\mu$: 0.4, Slow |
| | $\mu$: 0.7, Slow | $\mu$: 0.4, Slow | $\mu$: 0.9, Fast |
| | $\mu$: 0.8, Fast | $\mu$: 0.5, Slow | $\mu$: 0.5, Slow |
| | $\mu$: 0.8, Slow | $\mu$: 0.6, Slow | $\mu$: 1.0, Slow |
| | $\mu$: 0.9, Fast | $\mu$: 0.7, Slow | $\mu$: 0.8, Slow |
| | $\mu$: 0.9, Slow | $\mu$: 0.8, Slow | $\mu$: 0.6, Slow |
| | $\mu$: 1.0, Fast | $\mu$: 0.9, Slow | $\mu$: 0.7, Slow |
| ↓ least difficult | $\mu$: 1.0, Slow | $\mu$: 1.0, Slow | $\mu$: 0.9, Slow |

The curriculum devised by our scoring function shows several irregular patterns, such as a driving scenario with a friction coefficient $\mu$ of 0.6 and a rapid speed being ranked as more difficult data than that with a friction coefficient $\mu$ of 0.4 and the same rapid speed. Such a reversal occurs due to the conservative behavior of the built-in controller of the simulator. The controller tries to complete the reserved maneuvers while maintaining the desired speed, but it will slow down the speed during the large drifts because the controller is fundamentally designed not to push the vehicle out of the course. Thus, the vehicle states driven at lower friction levels may be more stable than those with higher friction. The difficulty order ranked by the self-taught scoring function could capture these ambiguous features of the given data while human scoring could only rely on the exterior information such as prior knowledge. Therefore, we can conclude that our scoring function is expected to be more appropriate for designing a curriculum for the vehicle dynamics estimation task than a scoring function based on human prior knowledge.

The estimation results by the curriculum based on our scoring function are evaluated in Section V-B1. We train the model with the two above human-prior-knowledge-based curriculum candidates described above and compared these with a curriculum based on our scoring function in Section V-B3 for further investigation.

#### 4) PACING FUNCTION

The pacing function is based on Equation 4. The parameter $\theta$ in pacing function $\mathbf{P}_\theta$ determines the rate of the curriculum being updated. Different rates of curriculum updates in terms of $\theta$ are visualized in Figure 4. The non-curriculum method is constantly trained with a universal set of the training data. When the $\theta$ is small, *i.e.*, 1 or 2, the training data is slowly increases with the epochs from the easier to the more difficult driving scenarios. If $\theta$ is large, *i.e.*, 12 or 13, the training data is nearly identical to that of the non-curriculum method. In terms of the neural network, the difficulty of the knowledge it should digest for each epoch increases as $\theta$ increases, thereby mitigating the effect of the gradual increase in the difficulty. In other words, it lacks the benefit of curriculum learning where knowledge obtained from learning easier data contributes to the learning of more difficult data if $\theta$ is large. Table 4 shows the results of a neural networks trained with $\theta = 1, 2, 4$, and 8. The estimation performance declines as $\theta$ increases, empirically proving that a small value of $\theta$ must be employed for a proper curriculum learning strategy. Therefore, we chose two different pacing functions with small values of $\theta$, 1 and 2. The comparison between the non-curriculum method and the curriculum methods is described in Section V-B1.

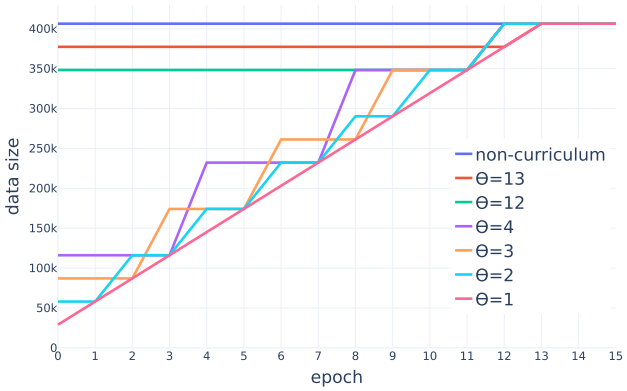



**FIGURE 4.** Increasing data size with respect to epochs on different $\theta$ in pacing function $\mathbf{P}_\theta$.

**TABLE 4.** $\mathbf{E}_{RMS}$ of different $\mathbf{P}_\theta$ for the test and the validation dataset.

| | $\theta = 1$ | | $\theta = 2$ | | $\theta = 4$ | | $\theta = 8$ | |
|---|---|---|---|---|---|---|---|---|
| | **Roll** | **Slip** | **Roll** | **Slip** | **Roll** | **Slip** | **Roll** | **Slip** |
| **Test** | **0.03902** | 0.2601 | 0.0470 | **0.2595** | 0.0508 | 0.2725 | 0.0426 | 0.2867 |
| **Val.** | **0.0582** | **0.5078** | 0.0615 | 0.5649 | 0.0616 | 0.5211 | 0.0630 | 0.5431 |

### B. EVALUATIONS

In this section, first we evaluate our method by comparing it with the non-curriculum method, including a detailed analysis. Secondly, we show the result of an alternative curriculum compared to our proposed method. Next, we collated the curriculum based on human prior knowledge with ours. Lastly, we compare our curriculum method with a domain expert model, which is trained on data with a narrower range of friction coefficients. This is described in Appendix B.

#### 1) CURRICULUM METHODS

In this work, *the improvement achieved in the sideslip angle estimation should be focused on*, as it is much more difficult to estimate the sideslip angle than roll angle on various friction surfaces [6].

We compared the estimation results of the non-curriculum method over the curriculum method with two pacing functions. Table 5 shows the curriculum method performs better in most cases compared to the non-curriculum method. The errors in the sideslip angle have a much larger magnitude than errors in the roll angle, and the performance increases or decreases for the roll angle are so small so as to be negligible. The sideslip angle $\mathbf{E}_{RMS}$ of the test data of the curriculum method with $\theta = 1$ shows more than a **16.5%** decrease compated to the non-curriculum method, which is a notable advance. There is also a 3.7% decrease in the validation data. The curriculum method with $\theta = 1$ has the best overall performance on both the test and validation data. Appendix A shows error graphs of the roll angle and sideslip angle during 50 epochs and reports the errors for all driving scenarios.

To observe the individual improvements for each driving scenario realized by our curriculum method compared to the non-curriculum method, we analyze a heat map of the estimation differences between the non-curriculum and curriculum methods, as depicted in Figure 7. Each row represents the error for each driving scenario, with the arrangement going from top to bottom according to the level of difficulty computed by the scoring function $\mathbf{S}$. The two sections of $\mathbf{E}_{RMS}$ and $\mathbf{E}_{max}$ are respectively divided by the largest absolute value in each section. Therefore, the more each element approaches $-1$ *i.e.*, dark blue, the greater the estimation improvement. The heat map shows that the benefits from curriculum learning are mainly achieved in difficult driving scenarios.

We plotted the inference results of the two neural networks trained with the non-curriculum method and our proposed curriculum method with $\theta = 1$. The validation data obtained by the controller of the simulator (Figure 5) and by a proficient human driver (Figure 6) were used for the estimation. The estimation result for test data is shown in Appendix A from the randomly selected driving scenario. It is demonstrated that the curriculum method outperforms the non-curriculum method, especially on the peak values where significant cornering occurs. We calculated the errors of the peak values with the entire validation dataset. The local maxima and minima were selected as the peak values by a simple comparison of 200 horizontal neighboring sideslip angle values. The corresponding sideslip angle mean absolute errors (MAE) of the non-curriculum method and curriculum method were 0.3009° and 0.2889°, showing a 4% of improvement.

**TABLE 5.** Estimation results for the test and the validation dataset of the non-curriculum method and our curriculum methods. The pacing function is set with the parameters $\theta = 1$ and $\theta = 2$.

| | Non-curriculum Method | | | | Curriculum Method($\theta = 1$) | | | | Curriculum Method($\theta = 2$) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Test | | Validation | | Test | | Validation | | Test | | Validation | |
| | Roll | Slip | Roll | Slip | Roll | Slip | Roll | Slip | Roll | Slip | Roll | Slip |
| $\mathbf{E}_{RMS}$ [deg] | 0.0432 | 0.3115 | **0.0549** | 0.5272 | **0.03902** | 0.2601(16.5% ↓) | 0.0582 | **0.5078**(3.7% ↓) | 0.0470 | **0.2595**(16.7% ↓) | 0.0615 | 0.5649(7.2% ↑) |
| $\mathbf{E}_{max}$ [deg] | 1.5631 | 8.1047 | 1.8342 | 17.6414 | 2.1768 | 8.5516 | 1.6537 | 16.1136 | 1.2619 | 9.9335 | 1.5887 | 15.9792 |


**FIGURE 5.** Inference results of the non-curriculum(original) and curriculum methods on a maneuver at a race track. These validation data were acquired on a road with a friction coefficient of 0.45 and a desired speed of 40 *km*/*h*: (a) Roll angle predictions graph, and (b) sideslip angle predictions graph.

**TABLE 6.** Estimation results for the test and the validation datasets. The pacing function is set with a parameter of $\theta = 1$.

| | Curriculum Method | | | | Anti-Curriculum Method | | | | Random Curriculum Method | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Test | | Validation | | Test | | Validation | | Test | | Validation | |
| | Roll | Slip | Roll | Slip | Roll | Slip | Roll | Slip | Roll | Slip | Roll | Slip |
| $\mathbf{E}_{RMS}$ [deg] | 0.03902 | **0.2601** | 0.0582 | **0.5078** | **0.0388** | 0.2995 | 0.0532 | 0.5505 | 0.0494 | 0.3037 | **0.0494** | 0.5475 |
| $\mathbf{E}_{max}$ [deg] | 2.1768 | 8.5516 | 1.6537 | 16.1136 | 2.5488 | 14.2054 | 1.9022 | 15.8884 | 1.5828 | 22.4276 | 1.0520 | 17.2514 |

### 2) ANTI-CURRICULUM AND RANDOM CURRICULUM

To demonstrate that our self-taught scoring feature helps curriculum learning, we conducted further experiments. Two different methods were designed, as described in Section III. The anti-curriculum method used the scoring function that sorts the data in reverse order of the curriculum method. The random curriculum method used a scoring function in which the driving scenarios are randomly scored. The performances are shown in Table 6. The results indicate that the anti-curriculum and random curriculum methods have inference accuracy levels similar to that of the non-curriculum method. The test and validation $\mathbf{E}_{RMS}$ of the sideslip angle by our curriculum method showed corresponding improvements of 16.5% and 3.7%(see Table 5), whereas the anti-curriculum


**FIGURE 6.** Inference results of non-curriculum and curriculum methods on a maneuver at a race track by the skilled human driver. The validation data ware acquired on a road with a friction coefficient of 0.85 and a desired vehicle speed of 80 $km/h$. (a) Roll angle predictions graph, and (b) sideslip angle predictions graph.

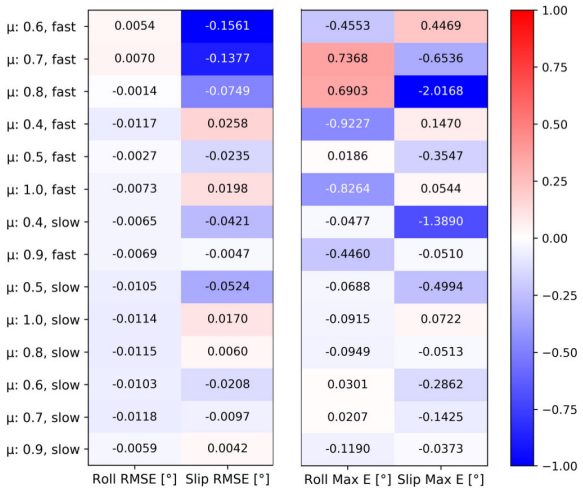



**FIGURE 7.** Performance differences of the non-curriculum and the curriculum methods with respect to each driving scenario.

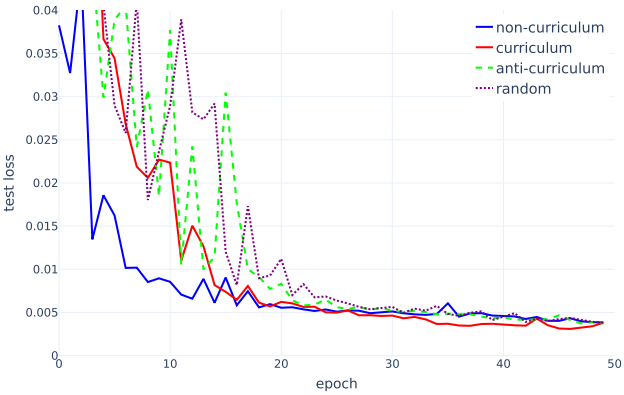



**FIGURE 8.** Results of compared test losses among the different methods during training. The pacing function of the curriculum method is $\mathbf{P}_{\theta=1}$. The curriculum method (in blue) has a lower test loss outcome than the others.

and random curriculum outcomes only achieved improvements of 3.8% and 2.5% on the test $\mathbf{E}_{RMS}$. Moreover, the respective validation $\mathbf{E}_{RMS}$ outcomes even increased by 4.4% and 3.8%.

We compiled a test loss graph of the four different methods; non-curriculum, our curriculum, anti-curriculum, and

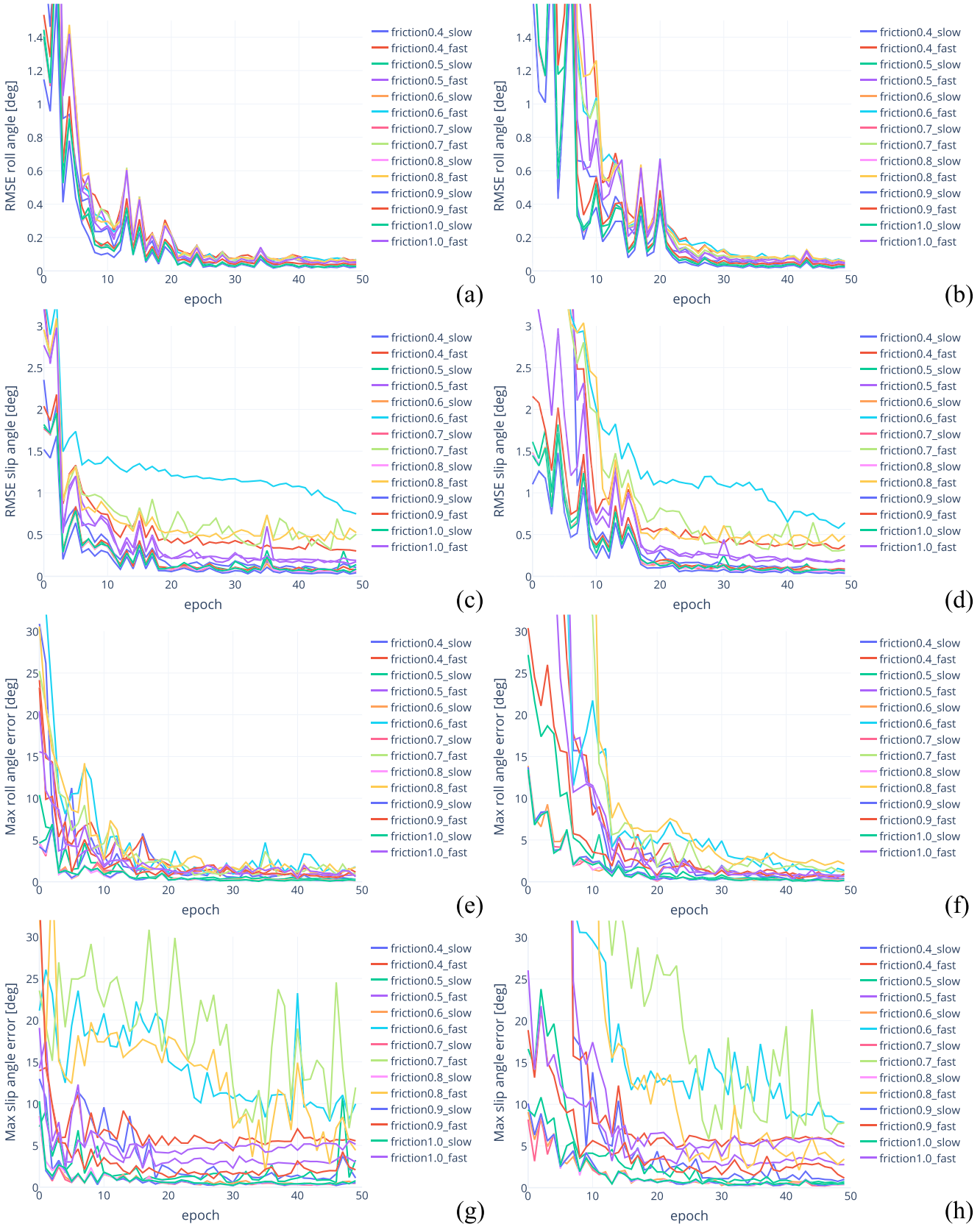



**FIGURE 9.** Error graph comparison of the non-curriculum method and the curriculum method (with $\theta = 1$) during training. The graphs in the left column are for the non-curriculum method and the graphs in the right column are for the curriculum method. (a)-(b) RMSE of the roll angle. (c)-(d) RMSE of the sideslip angle. (e)-(f) Max errors of the roll angle. (g)-(h) Max errors of the sideslip angle.

**TABLE 7.** Error measurements for the test and the validation datasets. The pacing functions of the three curriculum methods use the parameter $\theta = 1$.

| | Candidate 1 by a Human | | | | Candidate 2 by a Human | | | | Measured by a Scoring Function | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Test | | Validation | | Test | | Validation | | Test | | Validation | |
| | Roll | Slip | Roll | Slip | Roll | Slip | Roll | Slip | Roll | Slip | Roll | Slip |
| $\mathbf{E}_{RMS}$ [deg] | 0.0512 | 0.2804 | 0.0561 | 0.5255 | 0.0403 | 0.2763 | **0.0559** | 0.5535 | **0.03902** | **0.2601** | 0.0582 | **0.5078** |
| $\mathbf{E}_{max}$ [deg] | 2.3659 | 22.6417 | 1.3524 | 17.2104 | 2.0390 | 19.7553 | 1.1425 | 15.7273 | 2.1768 | 8.5516 | 1.6537 | 16.1136 |

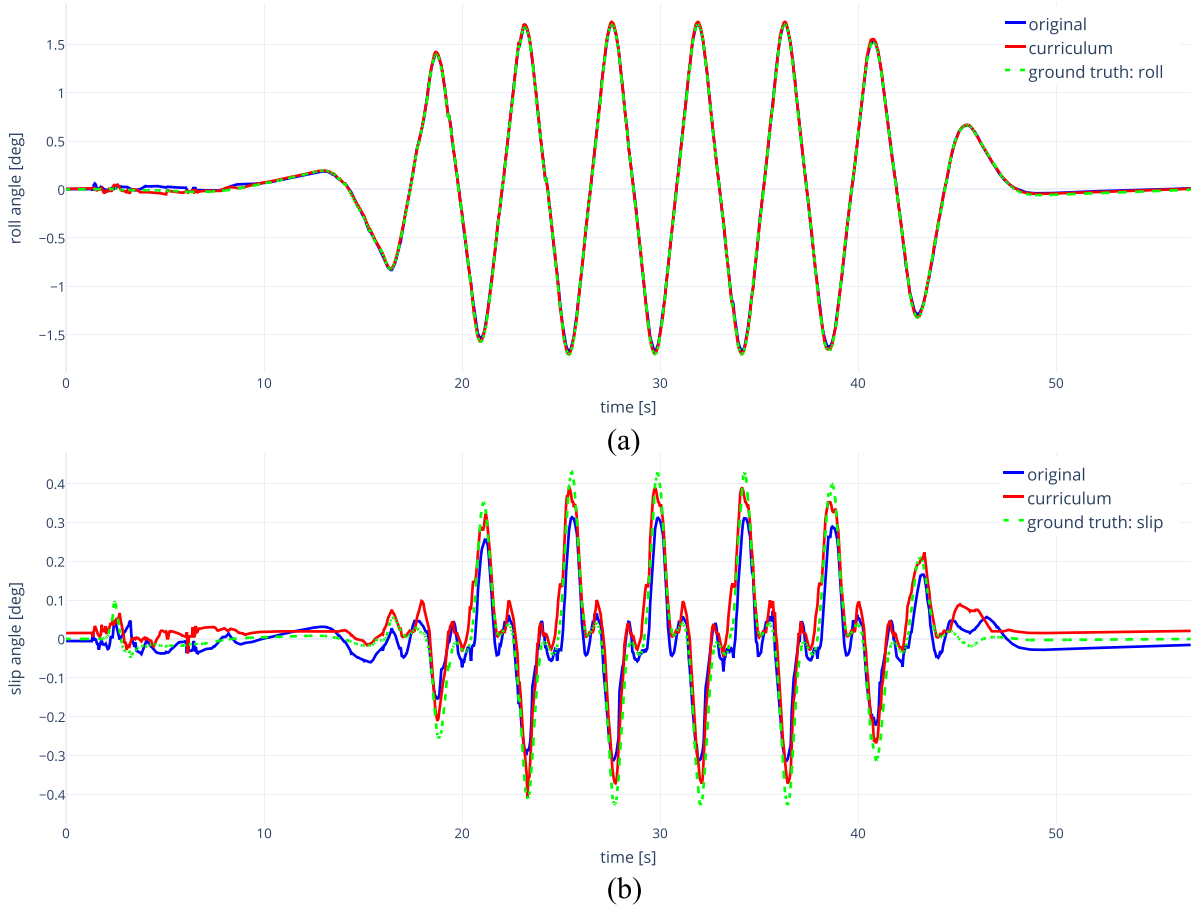



**FIGURE 10.** Inference results of non-curriculum and curriculum methods on a slalom maneuver included in the test data. The data ware acquired on a road with a friction coefficient of 0.6 and a desired vehicle speed of 60 *km*/*h*. (a) Roll angle predictions graph, and (b) sideslip angle predictions graph.

random curriculum method for a comparison, as shown in Figure 8. The curriculum method exhibits the lowest test loss, which is the same as the results of curriculum learning for the training of deep learning networks [14]. Another notable finding is that the test loss of the curriculum method decreases continuously, while that of the anti-curriculum and random curriculum methods fluctuate greatly. It can be interpreted that the knowledge model acquired from the previous data is highly appreciated in learning data with proximal difficulty in the near future steps.

### 3) HUMAN SCORED DIFFICULTY ORDER

We also demonstrated that the well-defined self-taught scoring function contributes to curriculum learning more than just the heuristically determined data difficulty order. The two difficulty order candidates are described in Table 3 with pacing function $\mathbf{P}_{\theta=1}$. Table 7 shows the self-taught scoring function has much better performances than those with human-scored orders. The test $\mathbf{E}_{RMS}$ of the sideslip angle of the former has a 16.5% improvement. However, the candidate 1 and 2 only have 9.9% and 11.3% improvements, respectively. This indicates that our self-taught scoring function is thoroughly designed and sufficient to outperform scoring based on human prior knowledge.

## VI. CONCLUSION

We present effective curriculum learning scheme for vehicle dynamics estimation along with self-taught scoring and pacing functions, which are devised thoroughly by considering the nature of the problem. The curriculum method shows

**TABLE 8.** Error measurements of the test data for each driving scenarios.

| | Non-curriculum Method | | | | Curriculum Method ($\theta = 1$) | | | |
|---|---|---|---|---|---|---|---|---|
| | $E_{RMS}$ [deg] | | $E_{max}$ [deg] | | $E_{RMS}$ [deg] | | $E_{max}$ [deg] | |
| | Roll | Slip | Roll | Slip | Roll | Slip | Roll | Slip |
| $\mu$: 0.4, Slow | 0.0344 | 0.1219 | 0.3796 | 2.3385 | **0.0280** | **0.0798** | **0.3319** | **0.9495** |
| $\mu$: 0.4, Fast | 0.0647 | **0.3159** | 1.5631 | **5.7585** | **0.0530** | 0.3417 | **0.6403** | 5.9055 |
| $\mu$: 0.5, Slow | 0.0313 | 0.1036 | 0.3436 | 1.1723 | **0.0208** | **0.0512** | **0.2747** | **0.6728** |
| $\mu$: 0.5, Fast | 0.0438 | 0.1963 | **0.7786** | 3.4373 | **0.0411** | **0.1728** | 0.7973 | **3.0826** |
| $\mu$: 0.6, Slow | 0.0291 | 0.0603 | **0.2326** | 0.6985 | **0.0188** | **0.0396** | 0.2627 | **0.4123** |
| $\mu$: 0.6, Fast | **0.0558** | 0.7740 | 1.4697 | **8.1047** | 0.0612 | **0.6179** | **1.0144** | 8.5516 |
| $\mu$: 0.7, Slow | 0.0295 | 0.0468 | **0.1884** | 0.4562 | **0.0177** | **0.0370** | 0.2091 | **0.3137** |
| $\mu$: 0.7, Fast | **0.0532** | 0.4432 | **0.9851** | 7.0702 | 0.0602 | **0.3055** | 1.7219 | **6.4166** |
| $\mu$: 0.8, Slow | 0.0290 | **0.0412** | 0.2299 | 0.4002 | **0.0175** | 0.0472 | **0.1349** | **0.3489** |
| $\mu$: 0.8, Fast | 0.0602 | 0.5750 | **1.4866** | 5.5292 | **0.0587** | **0.5001** | 2.1768 | **3.5124** |
| $\mu$: 0.9, Slow | 0.0199 | **0.0287** | 0.2370 | 0.4472 | **0.0140** | 0.0329 | **0.1180** | **0.4099** |
| $\mu$: 0.9, Fast | 0.0380 | 0.0839 | 1.0880 | 2.4185 | **0.0310** | **0.0791** | **0.6420** | **2.3676** |
| $\mu$: 1.0, Slow | 0.0292 | **0.0473** | 0.2307 | **0.4240** | **0.0178** | 0.0643 | **0.1392** | 0.4962 |
| $\mu$: 1.0, Fast | 0.0514 | **0.1486** | 1.2710 | **5.5396** | **0.0441** | 0.1684 | **0.4446** | 5.5940 |

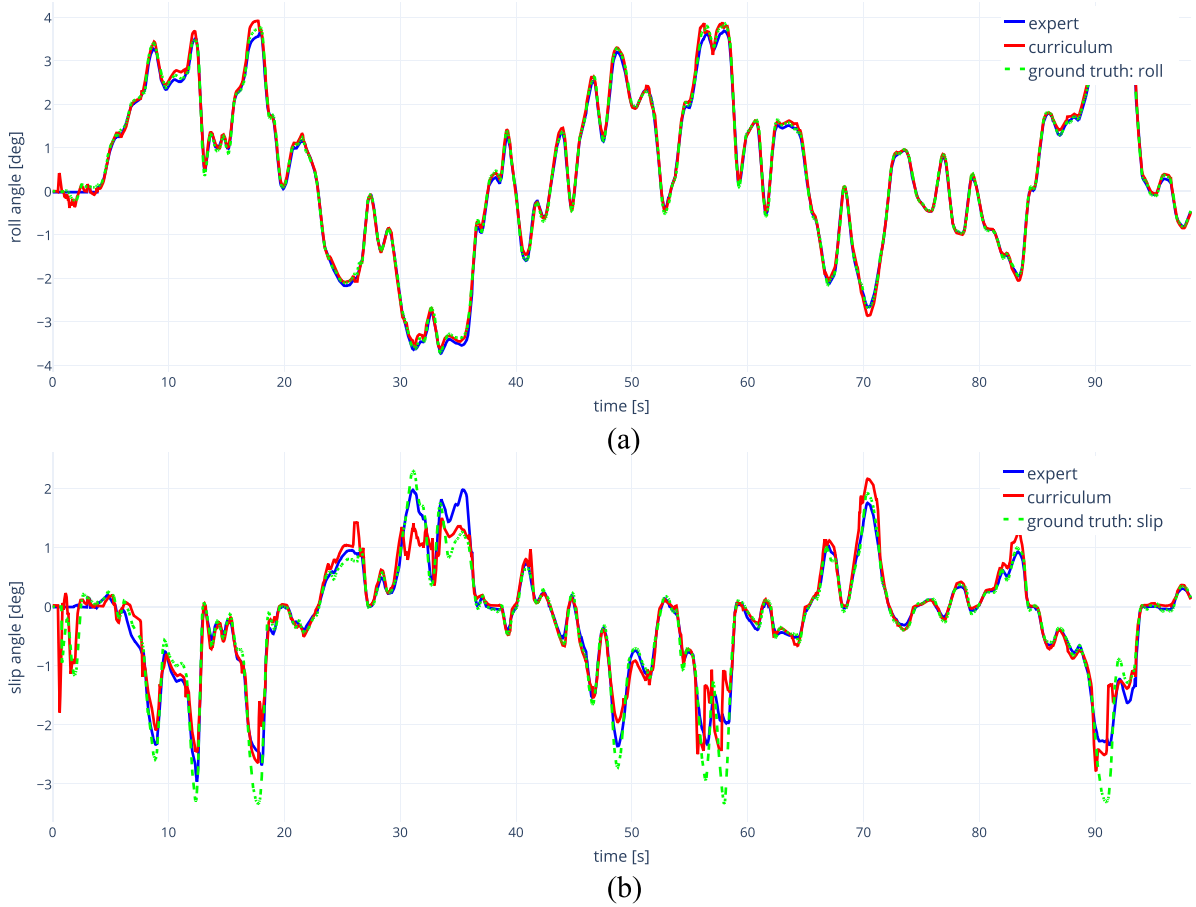



**FIGURE 11.** Inference results of our model and the expert model. This validation data ware acquired from a human driver on a road with a friction coefficient of 0.85 and a desired vehicle speed of 80 *km*/*h*. (a) Roll angle predictions graph, and (b) sideslip angle predictions graph.

reasonable improvements in most of the driving scenarios compared to the non-curriculum method, especially during significant cornering motions. Moreover, our curriculum outperforms the curriculum defined by human prior knowledge, indicating the distinction of our self-taught scoring function.

In addition, we also examined the performance between an expert model and the proposed method. Despite the slight difference in the performance, considering the inference speed and computational budget our method is reasonably comparable.

## APPENDIX A
## ESTIMATION RESULTS OF CURRICULUM LEARNING

The detailed versions of test errors measured for each driving scenario with the non-curriculum and curriculum methods ($\theta = 1$) are shown in Table 8. The visualized error graphs of the roll angle and sideslip angle during 50 epochs are shown in Figure 9. The estimation result using an arbitrarily selected course from the test data is shown in Figure 10. The compared models are trained with the non-curriculum and the curriculum method with the pacing function $\mathbf{P}_{\theta=1}$.

## APPENDIX B
## COMPARISON WITH THE EXPERT MODEL

There are numerous studies focused on two-stage estimation methods to determine vehicle sideslip angles [6], [7]. The main idea is to clarify the correlation between the human prior knowledge and the stability of the vehicle for the neural network. The second-stage estimator will make predictions on vehicle stability according to the prior information predicted by the first-stage estimator. Similarly, the mixture-of-experts method can be introduced for this task. We can design a model in which an input-dependent gate makes a soft prediction about the driving environment, with the weighted summation of the individual expert opinions based on the gate for the final result [37]. In such a situation, the expert model is expected to have a better prediction in a certain driving environments than a single model trained to fit all data acquired from various environments. However, the mixture-of-experts model has more network parameters and greater complexity than a single model and therefore requires more inference time. Because software runtime is very important on a commercial vehicle, there is a tradeoff between the accuracy and inference speed.

We simply compared our model with the single expert model. We assumed that the road surface friction most critically affects the lateral stability of the vehicle. Then, an expert model was trained only with the data acquired from environments with road friction coefficients of 0.8 and 0.9 road friction coefficient. This expert model was not trained while coupled with the gate, but it can be considered as an empirically generated expert to regress the classified data subset more accurately. The expert model employed an identical network, as shown in Figure 1. Finally, we compare the prediction results of our model and the expert model (see Figure 11). The data used for validation are identical to the data shown in Figure 6.

The results show that the expert model trained with the confined road friction range performs slightly better than our model with proposed the curriculum learning approach. However, embedded single board computers in actual vehicles have limited parallel computing capabilities; thus, the estimator should be guaranteed for real-time performance. Considering that the single model runs much faster than the multiple expert models, the difference between the two models is reasonably small.

## ACKNOWLEDGMENT

*(Jihwan Bae and Taekyung Kim contributed equally to this work.)*

## REFERENCES

[1] G. Baffet, A. Charara, and D. Lechner, "Estimation of vehicle sideslip, tire force and wheel cornering stiffness," *Control Eng. Pract.*, vol. 17, no. 11, pp. 1255–1264, Nov. 2009.

[2] C. Zhang, Q. Chen, and J. Qiu, "Robust H∞ filtering for vehicle sideslip angle estimation with sampled-data measurements," *Trans. Inst. Meas. Control*, vol. 39, no. 7, pp. 1059–1070, 2017.

[3] B. L. Boada, M. J. L. Boada, L. Vargas-Melendez, and V. Diaz, "A robust observer based on H∞ filtering with parameter uncertainties combined with neural networks for estimation of vehicle roll angle," *Mech. Syst. Signal Process.*, vol. 99, pp. 611–623, Jan. 2018.

[4] L. Vargas-Meléndez, B. Boada, M. Boada, A. Gauchía, and V. Díaz, "A sensor fusion method based on an integrated neural network and Kalman filter for vehicle roll angle estimation," *Sensors*, vol. 16, no. 9, p. 1400, Aug. 2016.

[5] L. Xiong, X. Xia, Y. Lu, W. Liu, L. Gao, S. Song, Y. Han, and Z. Yu, "IMU-based automated vehicle slip angle and attitude estimation aided by vehicle dynamics," *Sensors*, vol. 19, no. 8, p. 1930, Apr. 2019.

[6] W. Chen, D. Tan, and L. Zhao, "Vehicle sideslip angle and road friction estimation using online gradient descent algorithm," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 11475–11485, Dec. 2018.

[7] F. Naets, S. van Aalst, B. Boulkroune, N. E. Ghouti, and W. Desmet, "Design and experimental validation of a stable two-stage estimator for automotive sideslip angle and tire parameters," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 9727–9742, Nov. 2017.

[8] 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.

[9] N. A. Spielberg, M. Brown, N. R. Kapania, J. C. Kegelman, and J. C. Gerdes, "Neural network vehicle models for high-performance automated driving," *Sci. Robot.*, vol. 4, no. 28, Mar. 2019, Art. no. eaaw1975.

[10] L. P. González, S. S. Sánchez, J. Garcia-Guzman, M. J. L. Boada, and B. L. Boada, "Simultaneous estimation of vehicle roll and sideslip angles through a deep learning approach," *Sensors*, vol. 20, no. 13, p. 3679, Jun. 2020.

[11] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, no. 1, pp. 71–99, Jul. 1993.

[12] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. ICML*, 2009, pp. 41–48.

[13] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. NIPS*, 2010, p. 2.

[14] G. Hacohen and D. Weinshall, "On the power of curriculum learning in training deep networks," in *Proc. ICML*, 2019, pp. 2535–2544.

[15] D. Weinshall and G. Cohen, "Curriculum learning by transfer learning: Theory and experiments with deep networks," in *Proc. ICML*, 2018, pp. 5238–5246.

[16] J. Farrelly and P. Wellstead, "Estimation of vehicle lateral velocity," in *Proc. CCA*, 1996, pp. 552–557.

[17] A. Y. Ungoren, H. Peng, and H. Tseng, "A study on lateral speed estimation methods," *Int. J. Vehicle Auton. Syst.*, vol. 2, nos. 1–2, pp. 126–144, 2004.

[18] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *Found. Trends Robot.*, vol. 7, nos. 1–2, pp. 1–179, 2018.

[19] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*. [Online]. Available: https://arxiv.org/abs/1509.02971

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*. [Online]. Available: http://arxiv.org/abs/1707.06347

[21] M. Cutler, T. J. Walsh, and J. P. How, "Reinforcement learning with multi-fidelity simulators," in *Proc. ICRA*, 2014, pp. 3888–3895.

[22] S. Forestier, R. Portelas, Y. Mollard, and P.-Y. Oudeyer, "Intrinsically motivated goal exploration processes with automatic curriculum learning," 2017, *arXiv:1708.02190*. [Online]. Available: http://arxiv.org/abs/1708.02190

[23] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Sci. Robot.*, vol. 5, no. 47, Oct. 2020, Art. no. eabc5986.
[24] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *Proc. IROS*, 2020, pp. 1–7.
[25] Y. Song, H. Lin, E. Kaufmann, P. Dürr, and D. Scaramuzza, "Autonomous overtaking in Gran Turismo sport using curriculum reinforcement learning," in *Proc. ICRA*, 2021, pp. 1–7.
[26] R. Furuta, N. Inoue, and T. Yamasaki, "Fully convolutional network with multi-step reinforcement learning for image processing," in *Proc. AAAI*, 2019, pp. 3598–3605.
[27] S. Woo, J. Yeon, M. Ji, I.-C. Moon, and J. Park, "Deep reinforcement learning with fully convolutional neural network to solve an earthwork scheduling problem," in *Proc. SMC*, 2018, pp. 4236–4242.
[28] Y. Zhou, Z. Li, S. Xiao, C. He, Z. Huang, and H. Li, "Auto-conditioned recurrent networks for extended complex human motion synthesis," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.
[29] Y. Li, G. Tang, J. Du, N. Zhou, Y. Zhao, and T. Wu, "Multilayer perceptron method to estimate real-world fuel consumption rate of light duty vehicles," *IEEE Access*, vol. 7, pp. 63395–63402, 2019.
[30] S. Yoon and D. Kum, "The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles," in *Proc. IV*, 2016, pp. 1307–1312.
[31] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, 2009, p. 5.
[32] X. Ji, Y. Liu, X. He, K. Yang, X. Na, C. Lv, and Y. Liu, "Interactive control paradigm-based robust lateral stability controller design for autonomous automobile path tracking with uncertain disturbance: A dynamic game approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 6906–6920, Aug. 2018.
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. ICCV*, 2015, pp. 1026–1034.
[34] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 1–8.
[35] K. Han, M. Choi, and S. B. Choi, "Estimation of the tire cornering stiffness as a road surface classification indicator using understeering characteristics," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 6851–6860, Aug. 2018.
[36] R. Rajamani, *Vehicle Dynamics and Control*. Springer, 2011.
[37] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1177–1193, Aug. 2012.

**JIHWAN BAE** was born in Changwon, South Korea, in September 1997. He received the degree from the Busan Science High School, Busan, South Korea, in 2016, and the B.S. degree in electrical engineering and computer science (EECS) from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2020.

Since 2020, he has been a Researcher with the Ground Technology Research Institute, Agency for Defense Development, Daejeon, Republic of Korea. His research interests include the theoretical machine learning and deep learning, high-level computer vision, and studies of perception for autonomous vehicles.

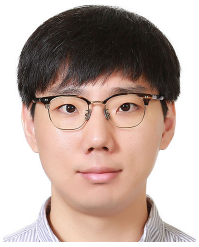
Mr. Bae's awards and honors include the Caltech Summer Undergraduate Research Fellow (SURF) Award, in 2019, and the Best Paper Award from the GIST EECS Department, in 2019.

**TAEKYUNG KIM** was born in Daejeon, Republic of Korea, in 1997. He received the B.S. degree from the College of Transdisciplinary Studies, Daegu Gyeongbuk Institute of Science and Technology (DGIST), in 2020.

Since 2020, he has been a Researcher with the Ground Technology Research Institute, Agency for Defense Development, Daejeon. His research interests include the development of mobile robot navigation systems, robot planning using deep learning and studies of path planning, control, perception, and integration for autonomous vehicles.

Mr. Kim's awards and honors include the Talent Award of Korea (Deputy Prime Minister of the Republic of Korea) and First Prize from the SOSCON 2019 Robot Open Source Laboratory (Competition at Samsung Open Source Conference), the SOSCON Robot Cleaner Autonomous Path Planning Algorithm Hackathon (Competition at the Samsung Open Source Conference), and the Autonomous Vehicle Technical Report in, 2018 and 2019 International Student Green Car Competitions.

**WONSUK LEE** received the B.S. degree in aerospace engineering from Korea Aerospace University, in 2005, and the M.S. and Ph.D. degrees in aerospace engineering from KAIST, in 2007 and 2011, respectively. He is currently a Research Scientist with Agency for Defense Development. His research interests include planning and control methods for autonomous systems and deep reinforcement learning. He received the Sage Best Paper Award, in 2011.



**INWOOK SHIM** received the B.S. degree in computer science from Hanyang University, in 2009, the M.S. degree in Robotics Program in electrical engineering from KAIST, in 2011, and the Ph.D. degree in electrical engineering from the Division of Future Vehicles, KAIST, in 2017. He is currently a Research Scientist with Agency for Defense Development. His research interests include robot vision for autonomous systems and deep learning. He was a member of Team KAIST, which took first place at the DARPA Robotics Challenge Finals, in 2015. He was a recipient of the Qualcomm Innovation Award and a NI Finalist at the NI Student Design Showcase. He received the KAIST Achievement Award of Robotics and the Creativity and Challenge Award from KAIST.