

Received 18 November 2022, accepted 11 December 2022, date of publication 14 December 2022,
date of current version 20 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3229067

RESEARCH ARTICLE

Chebyshev Transform-Based Robust Trajectory Prediction Using Recurrent Neural Network

SUJIN KWAG¹, BYEONGJU KANG¹, WONHEE KIM², (Member, IEEE),
AND YUNHYOUNG HWANG³

¹AI-Bigdata Research Center, Korea Automotive Technology Institute, Pangyo 13449, South Korea

²School of Energy Systems Engineering, Chung-Ang University, Seoul 06974, South Korea

³Department of Mechanical Engineering, Jeonbuk National University, Jeonju 54896, South Korea

Corresponding author: Yunhyoung Hwang (yhh@jbnu.ac.kr)

This work was supported in part by the Korea Institute of Police Technology (KIPoT) grant funded by the Korean Government [Korea National Police Agency (KNPA)] “Development of on-site control technology for road traffic network control in the event of an incident or disaster” under Grant 092021C29S02000, in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2022R1I1A3063354, and in part by the research funds for newly appointed professors at Jeonbuk National University, in 2021.

ABSTRACT Trajectory prediction is gaining attention as a form of situational awareness because it is an essential component of the support system of autonomous driving, particularly in urban areas. A promising application is cooperative driving automation, where the traffic scene is monitored by roadside sensors with undisrupted views. A critical problem is that these sensors are adversely affected by inclement weather, including drenching rain or large amounts of snow, in which case the reliability of the prediction results can be significantly compromised. To address these problems, this study proposes a framework for robust vehicle-trajectory predictions based on the Chebyshev transform. In the proposed framework, the original trajectory snippets (partial trajectories) are Chebyshev-transformed, and the resulting coefficients form new snippets. The LSTM (long-short term memory) encoder-decoder structure was trained and tested using these new coefficient snippets, which were extracted from a public vehicle trajectory dataset. The performance and robustness of the proposed framework were verified by emulating sensor data that were incomplete as a result of environmental factors. The proposed framework provides stable and accurate long-term trajectory prediction because the Chebyshev transform is robust to incomplete sensor data by virtue of its uniform nature.

INDEX TERMS Autonomous driving, Chebyshev polynomial, Chebyshev transform, long short-term memory, recurrent neural network, trajectory prediction.

I. INTRODUCTION

Autonomous driving is being raised to the next level by technical progress in the relevant software and hardware. However, the barriers are also becoming higher as the operational domain extends to urban areas. Removing these barriers is the key to advancing autonomous driving. One of these barriers is knowledge of the future trajectories of surrounding vehicles. In particular, it is necessary to exactly predict the future trajectories of non-autonomous vehicles because mixed traffic, consisting of both autonomous and

non-autonomous vehicles, can be expected to coexist for the foreseeable future until society becomes fully autonomous.

Trajectory prediction can be performed using information received from either the on-board sensors installed in autonomous vehicles or from roadside sensors for cooperative driving automation. Among them, trajectory prediction with the aid of roadside sensors is currently attracting considerable attention because roadside sensors usually have omniscient views; thus, the traffic scene can be monitored with less obstruction. Previous studies [1], [2], [3], [4] investigated this possibility in the next stage of autonomous driving.

The problem with traffic scene monitoring based on roadside sensors is that these sensors are continually exposed to

The associate editor coordinating the review of this manuscript and approving it for publication was Gerardo Flores ¹.

the environment. Thus, their reliable operation is negatively affected by adverse weather, such as heavy snowfalls, torrential rain, and strong wind, with camera sensors being particularly vulnerable to these conditions. Objects that should have been detected may be obstructed or may even remain undetected owing to raindrops or frost that covers the lens of sensors, as discussed in [5]. Accordingly, trajectory prediction would be negatively influenced by these conditions, degrading the reliability of autonomous driving.

Research on vehicle trajectory prediction has been gaining momentum, as this technology is required for a higher level of autonomous driving, and the widespread use of neural network-based approaches is now accelerating this trend. Althché and De la Fortelle [6] applied a long short-term memory (LSTM) network to vehicle trajectory prediction for a highway situation and validated the performance of the proposed model with the public NGSIM [7] dataset. Messaoud et al. [8] and Luo et al. [9] proposed an attention mechanism that combined different networks for trajectory prediction. Zyner et al. [10] proposed a mixture density network (MDN)-based framework for multimodal prediction in roundabouts. Ding and Shen [11] supplemented context information such as information about the construction site and speed regulation for elaborating the prediction accuracy. Their suggested model consisted of two levels of networks: The upper level classifies the driving policy, such as proceeding straight ahead, yielding, or turning, and the lower level generates the trajectory prediction based on an optimization method. The method proposed by Raipuria et al. [2] was also based on an LSTM encoder-decoder structure [12], and utilized a geometrical curvilinear coordinate system in which the road curvature was featured to elaborate the prediction accuracy. Similarly, Yu et al. [13] considered road geometries to improve the prediction accuracy in various road environments. Jiang et al. [14] focused on the temporal accuracy of trajectory prediction. Other researchers [15], [16], [17], [18] considered the surrounding nearby vehicles for an improved understanding of future vehicle trajectories. Deo and Trivedi [15] suggested a maneuver classification and trajectory prediction model using intervehicle interactions with an LSTM network. In addition, SCALE-NET, proposed by Jeon et al. [16], used an edge-enhanced graph convolutional neural network (EGCN) and LSTM, which features efficient computational performance regardless of the number of vehicles in the region of interest. Inter-vehicular interaction was applied to conflicting vehicles at an intersection [17]. The proposed model generates multiple hypothesis trajectories based on maneuver reasoning results. Deo and Trivedi [18] proposed a framework for vehicle trajectory prediction based on social-LSTM, which has been widely used for pedestrian trajectory prediction [19], [20]. Bock et al. [3] proposed an LSTM-based self-learning trajectory prediction framework with additional data from a new measurement.

These vehicle trajectory prediction methods can be categorized into two groups based on traffic scene monitoring. In [2], [3], and [16], multiple roadside sensors were installed

TABLE 1. Comparative table of related studies.

Objective	Ref.	Base model	Type
Accuracy	[6], [8], [11], [12],	LSTM	Independent
	[13], [14], [16], [17]	LSTM	Interactive
Multi-modality	[9]	LSTM	Independent
	[10]	RNN	Independent
	[15], [18]	LSTM	Interactive
Sensor fault-tolerance	Proposed	LSTM	Independent

The term *independent* implies that only the feature data for the ego-vehicle are considered, whereas *interactive* implies that, in addition, the feature data for surrounding vehicles are considered.

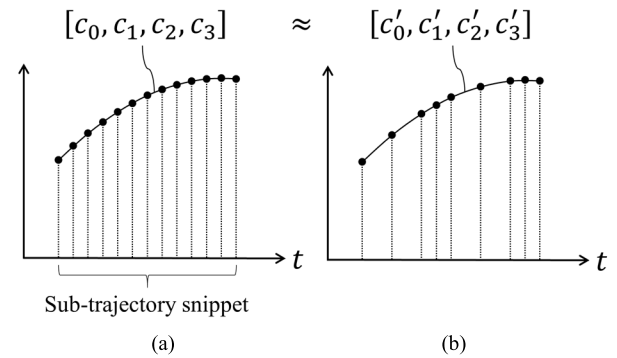


FIGURE 1. Illustrative example of the uniform nature of Chebyshev transform. The $c_{0:3}$ and $c'_{0:3}$ denote the transformed coefficients for (a) dense and (b) sparse sequences, respectively.

on a roadside unit, and trajectory predictions were provided to support cooperative driving automation. In [9] and [15], on the contrary, trajectory prediction was conducted on the side of the autonomous vehicle with on-board sensors and the predicted information was utilized in the decision process for autonomous driving. In addition to neural-network-based methods, various filter-based or stochastic methods were proposed [4], [21], [22], [23].

However, previous studies on trajectory prediction have typically focused on the improvement of the prediction performance, including accuracy and multimodality, as discussed in the literature survey, and problems that arise because of sensor degradation have not yet been addressed, although, practically, it is a critical issue. As discussed later, the conventional approach is highly vulnerable to incomplete sensor data. In this regard, a comprehensive discussion on sensor degradation issues is required when attempting to solve the trajectory prediction problem, considering that our study aims to provide robust long-term vehicle trajectory prediction, even with incomplete sensor data. Our findings are expected to make an important contribution to the realization of cooperative driving automation at a higher level. Table 1 presents a comparative table of the related studies and summarizes the model types and objectives of each study.

This study was motivated by the work of Wiest et al. [24] and was based on the Chebyshev transform [25], which transforms time-sequential physical data to coefficients for Chebyshev polynomial fitting. In the proposed framework,

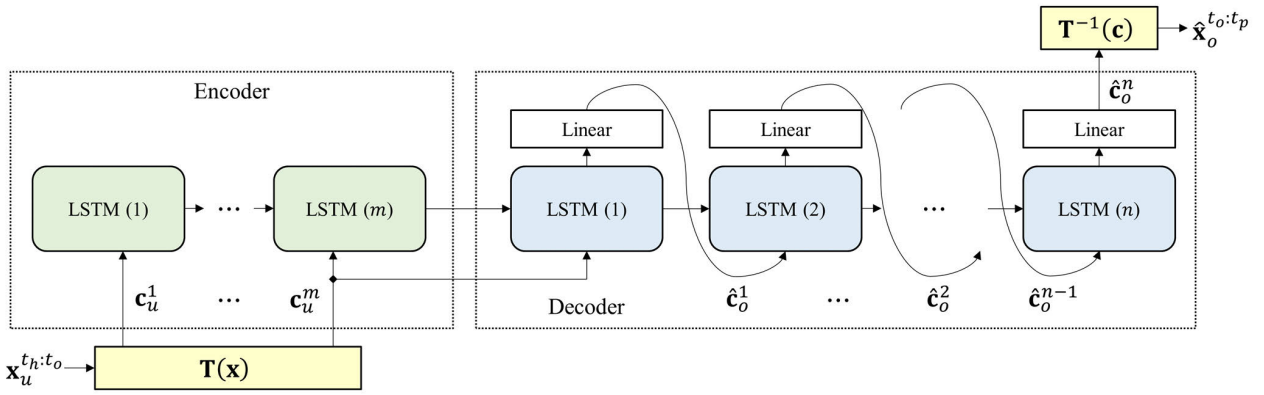


FIGURE 2. Proposed trajectory prediction framework. The original trajectory snippet is transformed into sequences of Chebyshev coefficient snippets (CCSs) and these CCS sequences are fed to the LSTM encoder. The subscripts u and o indicate the input and output sequences, respectively.

the original trajectory snippets, that is, partial trajectories, are Chebyshev-transformed, and the resulting Chebyshev coefficients form the new coefficient snippets, which are referred to as Chebyshev coefficient snippets (CCSs), for training and prediction. A strong advantage of the Chebyshev transform is that the resulting coefficients are uniform for a variable number of sequences in a fixed time interval [26]. Fig. 1 illustrates this aspect and suggests the possibility that the prediction results could be robust to partially incomplete sensor data. Another strong advantage is that the predictions can be in the form of a coefficient set, which is referred to as a *snapshot*, rather than the individual predicted positions upon which the predictions are made in the form of future positions. This aspect could be beneficial in vehicle-to-everything (V2X) communication applications with a limited bandwidth. For example, the only information transmitted via communication is a few Chebyshev coefficients rather than numerous time-sequential position values. The efficacy of this aspect increases as the number of vehicles for which the trajectory should be predicted increases.

The LSTM encoder-decoder was selected as a baseline structure because the family of recurrent neural networks (RNN) was shown to deliver good performance in the field of trajectory prediction as presented in the literature survey. The model was trained and tested with Chebyshev coefficient snippets in the proposed framework, and the prediction results were compared with those of the conventional approach, which is trained with time-sequential physical values in a conventional manner. The main contributions of this study are summarized as follows.

- A trajectory prediction framework based on a neural network, which is robust to partially incomplete data, is proposed. The proposed framework integrates the special feature engineering functionality of the Chebyshev transform that encodes the original trajectory snippets to CCSs. The experimental results, produced with data from the public vehicle trajectory dataset inD [27], verified the robust trajectory prediction performance, even with incomplete sensor data. This achievement is crucial for the realization of cooperative driving

automation even under harsh conditions such as during adverse weather.

- A novel method based on feature engineering is proposed for trajectory prediction in the form of a snapshot, rather than in the form of a series of time-sequential values. The last output sequence is configured to be the set of Chebyshev coefficients that encapsulates a predicted trajectory of the entire prediction horizon in the proposed method. In other words, the only information transmitted via the communication line is a few coefficient values instead of a large number of time-sequential values.
- The architecture of the model can be simplified because the number of sequences for the RNN can be reduced and additional processes such as sensor data imputation are not required, and these aspects might further reduce the learning time.

This study focuses on robustness to the sensor degradation issue and does not consider interaction between traffic occupants or other contextual information. The remainder of this paper is organized as follows. Section II summarizes the basics of the Chebyshev transform, and Section III introduces the proposed trajectory prediction framework, which is based on the Chebyshev transform and LSTM encoder-decoder structure [12]. This section also presents a short description of the LSTM encoder-decoder structure and describes the way in which feature engineering is used to create a snapshot of the predicted trajectory for the entire prediction horizon. In Section IV, the proposed framework is verified with the inD [27] dataset by injecting sensor data faults in various patterns. Section V concludes the paper.

II. CHEBYSHEV TRANSFORM

The proposed trajectory prediction framework is based on the Chebyshev transform, from which the objective function is approximated. This approximation starts from the definition of the Chebyshev polynomial, which is defined by the following trigonometric function [25]:

$$P_v(t) = \cos(ncos^{-1}(t)) \quad (1)$$

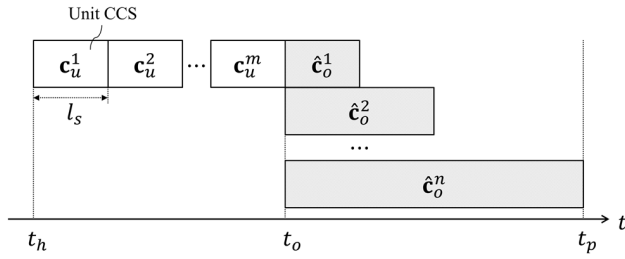


FIGURE 3. Configuration of Chebyshev coefficient snippets (CCSs) for input and output sequences. The subscripts u and o indicate the input and output features, respectively, and t_h , t_p , and t_o denote the history and prediction horizon, and the current time, respectively. The notation l_s denotes the time span for a unit CCS.

where v denotes the degree of the Chebyshev polynomial, and the values of this polynomial and domain variable t are bounded in $[-1, 1]$. The Chebyshev polynomial of an arbitrary degree is obtained from the recurrence relation, as follows:

$$\begin{aligned} P_0(t) &= 1 \\ P_1(t) &= t \\ P_{v+1}(t) &= 2tP_v(t) - P_{v-1}(t), \quad v \geq 1 \end{aligned} \quad (2)$$

From the properties of these orthogonal polynomials, the objective function $f(t)$ of the feature values can be approximated if the degree d is sufficiently large as follows:

$$f(t) \approx \left[\sum_{v=0}^{d-1} c_v P_v(t) \right] - \frac{1}{2} c_0, \quad t \in [-1, 1] \quad (3)$$

which is the truncated approximation for $d \leq N_d$, where N_d is an arbitrary number of data sample points. Note that the interval for the approximated function was normalized to $[-1, 1]$. Upon restoration of the original physical values using the inverse transform, the extent of the interval after the inverse transform can be arbitrarily extended to the designed horizon. The Chebyshev coefficient c_v in (3) can be obtained by the following Chebyshev transform:

$$c_v = \frac{2}{N_d} \sum_{i=1}^{N_d} f(t_i) P_v(t_i) \quad (4)$$

where t_i ($i = 1, \dots, N_d$) are the N_d zeros of $P_v(t)$, expressed as

$$t_i = \cos \left(\frac{\pi (i - 1/2)}{N} \right) \quad (5)$$

The transform in (4) is denoted for the set of coefficients:

$$[c_0^1, \dots, c_{d-1}^1, \dots, c_0^{N_f}, \dots, c_{d-1}^{N_f}] = \mathbf{T}(\mathbf{x}^{1:N_d}) \quad (6)$$

where $\mathbf{x}^{1:N_d}$ denotes N_d data samples, and N_f denotes the number of features in \mathbf{x} . Namely, the objective function for each feature is transformed, and the results are aggregated into one, as in (6).

In this study, the degree of approximation is set to $d = 4$, which is known as the cubic Chebyshev transform. As the

transformed coefficients in (6) form the feature data in the proposed framework, the degree of approximation determines the number of features; thus, the degree of approximation is determined by coordinating the model complexity and approximation quality. For the cubic Chebyshev transform, the approximated function for a single feature is represented as follows:

$$g(t) = c_0 + c_1 t + c_2 (2t^2 - 1) + c_3 (4t^3 - 3t) \quad (7)$$

where g is the approximated function of f .

III. TRAJECTORY PREDICTION FRAMEWORK

A. PROPOSED FRAMEWORK

The proposed framework is based on an LSTM encoder-decoder structure. Fig. 2 presents the architecture of the proposed trajectory prediction framework. The LSTM encoder-decoder, also known as *seq2seq*, is popular in the field of time-series prediction because it supports a flexible model structure with an arbitrary number of sequences on the side of both the input and output. The conventional prediction model based on physical values is represented as:

$$p(\hat{\mathbf{x}}_o^1, \dots, \hat{\mathbf{x}}_o^l | \mathbf{x}_u^1, \dots, \mathbf{x}_u^z) \quad (8)$$

where $\mathbf{x}_u^{1:z}$ and $\hat{\mathbf{x}}_o^{1:l}$ denote the input and predicted sequences, respectively, and $\mathbf{x}_u^{1:z}$ is from observations.

In the proposed framework, the original trajectory snippet for input sequences $\mathbf{x}_u^{t_h:t_o}$ is divided into m sub-trajectory snippets, where t_h and t_o denote the history horizon and current time, respectively, and the original physical data \mathbf{x} are composed of two features consisting of the x - and y -axis coordinates as

$$\mathbf{x} \in \mathbb{R}^2 = [x, y] \quad (9)$$

That is, $\mathbf{x}_u^{t_h:t_o}$ represents the past positions of the vehicle in the interval from t_h to t_o . The subtrajectory snippet for each feature in (9) is cubic-Chebyshev-transformed, and the resulting coefficients are aggregated into one from (6) as:

$$\mathbf{c} \in \mathbb{R}^{2d} = [c_0^1, \dots, c_3^1, c_0^2, \dots, c_3^2] \quad (10)$$

where \mathbf{c} represents the new feature data and is termed the Chebyshev coefficient snippet (CCS). In (10), the first four and next four elements are from the subtrajectory snippets for the x - and y -positions, respectively. It should be noted that the number of sequences was reduced to one for each subtrajectory snippet. Consequently, there exists a total of m input sequences of CCS, which are denoted as $\mathbf{c}_u^{1:m}$, and they are fed to the encoder part. In terms of the output sequences, a total of n sequences of CCS are predicted by the proposed framework and denoted as $\hat{\mathbf{c}}_o^{1:n}$. Finally, the prediction model is transformed into the proposed framework as follows:

$$p(\hat{\mathbf{c}}_o^1, \dots, \hat{\mathbf{c}}_o^n | \mathbf{c}_u^1, \dots, \mathbf{c}_u^m) \quad (11)$$



FIGURE 4. Road setting in Neuköllner Strasse, Aachen, Germany, where data for the inD vehicle trajectories dataset were collected. The yellow line with the notation L_s indicates the imaginary start line of the vehicle trajectories for the experiment.

B. CONFIGURATION OF CHEBYSHEV COEFFICIENT SNIPPETS

Fig. 3 shows the configuration of the CCSs for the input and output sequences in the proposed framework. For the input sequences, the time span $[t_h, t_o]$ is divided into m equal sub-time spans, as depicted in Fig. 3, such that the input trajectory snippet $\mathbf{x}_u^{t_h:t_o}$ is divided into m subtrajectory snippets with the same time span. Subsequently, each subtrajectory snippet is transformed into a CCS, resulting in a total of m CCSs as

$$\mathbf{c}_u^{1:m} = \left[T \left(\mathbf{x}_u^{t_h:t_o - (m-1)N_s T} \right), \dots, T \left(\mathbf{x}_u^{t_o - kT:t_o} \right) \right] \quad (12)$$

where $T(\cdot)$ denotes the transform for each subtrajectory snippet and $\mathbf{c}_u^{1:m}$ denotes the set of input CCS for equally divided time slots, and k and T denote the number of samples in the subtime span and sample time, respectively. The history horizon t_h can be represented as:

$$t_h = t_o - mkT \quad (13)$$

For the output sequences, the time span for a sequence is increased in multiples of the time span for a unit CCS to the prediction horizon t_p as depicted on the right in Fig. 3, where t_p can be represented as:

$$t_p = t_o + nkT \quad (14)$$

In Fig. 1, $\hat{\mathbf{c}}_o^i$ denotes the predicted output sequence of the i -th index, and the output has a total of n sequences. Note that the last output sequence $\hat{\mathbf{c}}_o^n$ covers the entire time span up to the prediction horizon; thus, it becomes the final prediction result in this configuration. This means that the prediction result exists in the form of the coefficient set, which is referred to as a *snapshot*, rather than a series of numerous time-sequential values. This is the optimal configuration that ensures the best prediction results and was empirically determined by conducting a substantial amount of validation using various types of configurations.

The predicted trajectory can be restored from the last sequence $\hat{\mathbf{c}}_o^n$ in the form of predicted positions using an

TABLE 2. Hyper-parameters Set-up.

Parameters	Values / Types
Hidden size	200
Batch size	100
Number of layers in encoder	3
Number of layers in decoder	3
Optimizer	Adam
Weight-decay	0.00001
Loss function	MSE
History horizon (t_h) @ $t_o = 0$ s	3 s
Prediction horizon (t_p) @ $t_o = 0$ s	5 s

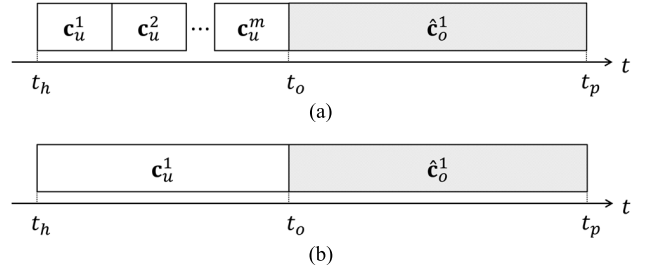


FIGURE 5. Various configurations of CCSs for the comparative analysis. Configuration for the (a) M2O and (b) O2O models, respectively.

inverse transform, such as

$$\hat{\mathbf{x}}_o^{t_o:t_p} = T^{-1}(\hat{\mathbf{c}}_o^n) \quad (15)$$

A great advantage of this restoration process is that the sample time for the restoration can be set regardless of the original sample time in the data. This feature can be useful for path planning on the side of the receiving vehicle. Moreover, because the noise is removed by filtration during the transform, higher-order physical values, such as the velocity and acceleration, can be restored without the effect of noise.

IV. EXPERIMENT

A. DATASET

The proposed framework was verified using the inD [27] dataset, which is a large-scale dataset of naturalistic vehicle trajectories at urban intersections. The data were collected at 25 Hz using camera-equipped drones and a typical position error of less than 10 cm was guaranteed. The experiment was conducted with data collected in Neuköllner Strasse, Aachen, Germany. In particular, 143 vehicle trajectories for left turns (depicted in Fig. 4) were utilized in our experiment because they feature highly dynamic motions, which enabled the verification to be conducted under severe conditions. As shown in Fig. 4, the vehicle trajectories for the experiment are widely distributed, which increases the uncertainty in the prediction problems. In this configuration, a total of 23,419 trajectory snippet pairs for the input and target sequences were generated, and they were transformed into CCSs according to the discussion in Section III. The data were randomly split into learning and test sets in an 80:20 ratio.

TABLE 3. Prediction errors [-] for Chebyshev coefficients.

Model	No missing value	f_m : 1 Hz			f_m : 2.5 Hz			f_m : 5 Hz		
		N_m : 1	N_m : 2	N_m : 3	N_m : 1	N_m : 2	N_m : 3	N_m : 1	N_m : 2	N_m : 3
M2M1	0.0015	0.0017	0.0039	0.0092	0.0437	0.0144	0.1582	0.0923	0.4829	0.5555
M2M2	0.0025	0.0024	0.0037	0.0071	0.0055	0.0141	0.0276	0.0463	0.0768	0.1065

f_m denotes the frequency at which missing values occur and N_m denotes the number of consecutive missing values.

TABLE 4. Comparative study I: prediction errors [m] for future positions.

Model	No missing value	f_m : 1 Hz			f_m : 2.5 Hz			f_m : 5 Hz		
		N_m : 1	N_m : 2	N_m : 3	N_m : 1	N_m : 2	N_m : 3	N_m : 1	N_m : 2	N_m : 3
M2M1	2.299	2.334	2.832	3.674	2.582	3.803	3.929	2.562	3.936	5.087
M2M2	2.456	2.453	2.453	2.454	2.506	2.501	2.491	2.451	2.470	2.460
Baseline	2.192	3.311	3.905	4.282	4.700	5.930	7.034	5.081	9.577	12.597

B. EXPERIMENTAL SET-UP

The experiment was set up on the PyTorch platform using the hyperparameters specified in Table 2 in our model. NVIDIA GeForce RTX 2060, 16 GB RAM, and Intel Core i7-8750 CPU @2.21 GHz were used in the experiments. The mean squared error (MSE) was used as a loss function, and this metric evaluates the average squared Euclidean distance between the target coefficient vector as in (10) and the predicted coefficient vector as

$$\ell = \frac{1}{nN_b} \sum_{k=1}^{N_b} \sum_{j=1}^n w_i \sum_{i=0}^{d-1} \left[(e_{ijk}^x)^2 + (e_{ijk}^y)^2 \right] \quad (16)$$

where $e_{ijk}^x = \hat{c}_{ijk}^x - c_{ijk}^x$ and $e_{ijk}^y = \hat{c}_{ijk}^y - c_{ijk}^y$ denote the prediction errors for each element in (9). N_b in (16) denotes the number of data points in a batch. As denoted in (16), the loss function is customized to set the weights w_i to focus on learning for the specific output sequences. This is because the final prediction results were obtained from the final output sequence. In this regard, the weight w_i was assigned the value of 1 for sequences that correspond to the time horizon from 3 to 5 s, and set to 0 for short-term sequences. The Adam optimizer was used with a weight decay of 0.00001.

The objective of this study is to construct a robust trajectory prediction framework that is robust to sensor-degradation problems. One of the representative and critical types of incomplete sensor data is missing data, for which various factors, including adverse weather and hardware failures, could be responsible. Thus, incomplete sensor data were emulated by injecting the missing periodic data. The original physical input data were zero padded at the frequency of 1/2.5/5 Hz, and the number of consecutive missing data was set to 1/2/3 samples, resulting in a total of 9 patterns of missing data. For a comparative analysis between the proposed and conventional approaches and between various CCS configurations in the proposed approach, the following alternatives were examined:

- Baseline: The LSTM encoder–decoder model without any feature engineering. The input and output feature data consist of time-sequential position values in a conventional approach.

- M2M1: The LSTM encoder–decoder model with multiple input and output CCS sequences, as shown in Fig. 3. The time span for input unit CCS l_s in Fig. 3 was set to 0.5 s. The time span for the output sequences is increased in multiples of l_s to the prediction horizon t_p in Table 2, as described in the previous section.
- M2M2: Same as the M2M1 model, except that the time span for unit CCS l_s was set to 1.0 s, which was longer than that of M2M1.
- M2O: The LSTM encoder–decoder model with multiple input CCS sequences and only one output CCS sequence. The l_s for the input sequence is set to 0.5 s, and only the output sequence covers the entire time span, up to the prediction horizon t_p in Table 2. Fig. 5(a) shows the M2O configuration.
- O2O: The LSTM encoder–decoder model with one input and output CCS sequence. For this model, the input sequence from the observations covers the entire time span down to the historical horizon t_h in Table 2, and the output sequence covers the entire time span up to the prediction horizon t_p in Table 2. Fig. 5(b) shows the O2O configuration.

A comparative study shows that the proposed framework with Chebyshev transform-based feature engineering renders the trajectory predictions even more robust than the conventional approach.

Moreover, a comparative study between various CCS configurations presents the most suitable configuration for the trajectory prediction problems with the proposed framework. For each configuration of the CCSs for the various models presented in Fig. 3 and Fig 5, the input CCSs sequences are fed to the proposed framework, as shown in Fig. 2, which outputs the predicted CCS sequences according to the configurations described above.

C. PREDICTION RESULTS–CHEBYSHEV COEFFICIENTS

The original trajectory snippets were transformed into CCSs, which were subsequently used to train the model. Thus, the prediction performance was first verified in the form of the Chebyshev coefficients. The proposed framework was

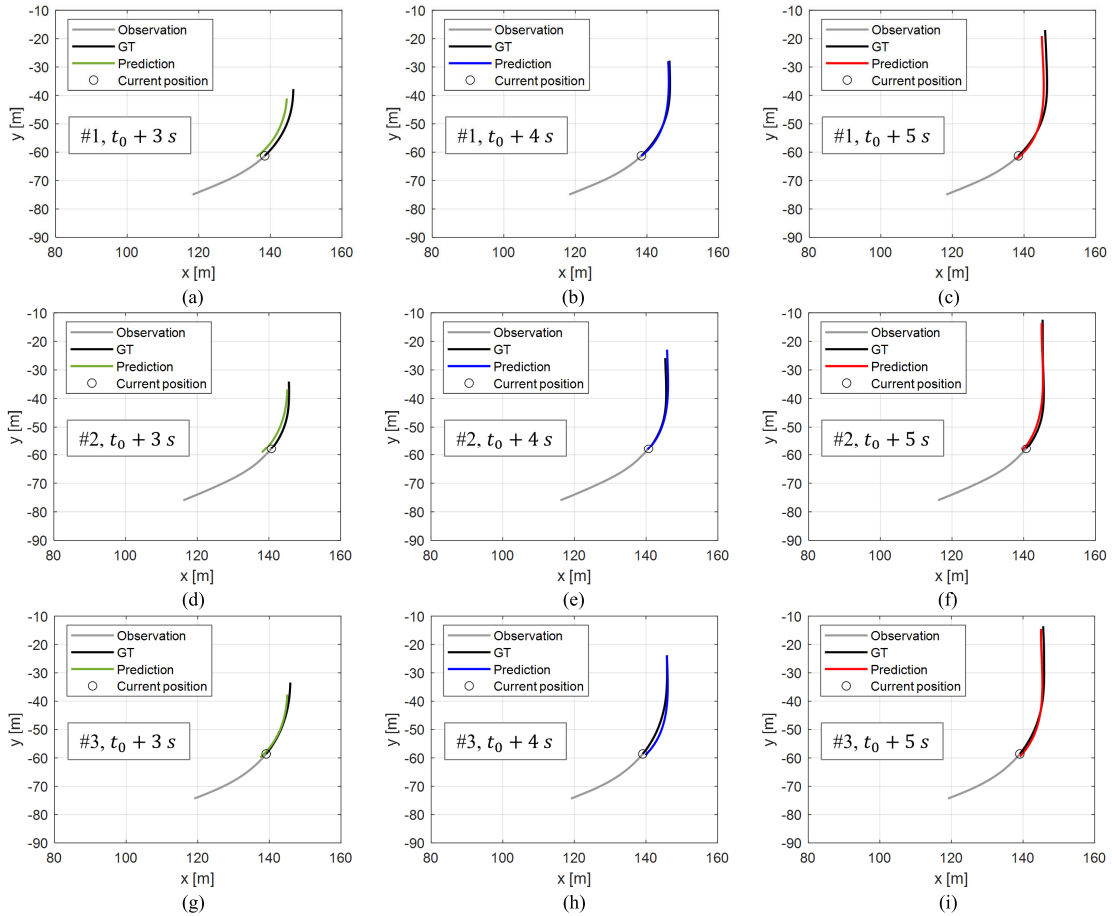


FIGURE 6. Trajectory prediction results for three samples using the M2M1 model for $f_m : 1$ Hz and $N_m : 1$: prediction results from the predicted sequences that correspond to the time horizon of (a), (d), (g) 3, (b), (e), (h) 4, and (c), (f), (i) 5 s, respectively.

optimally trained with losses from the learning and test data, but the prediction accuracy was calculated from a specific instance of the trajectory during the left turn. This instance was set to $t_l + 2$ s, where t_l indicates the time the vehicles crossed the imaginary start line L_s . This configuration generates verification data for the most dynamic instances during the turns. Recall that the final prediction results are obtained from the last output CCS sequence. The prediction accuracy was measured as the average of the Euclidean distances between the target coefficient vector, as in (10), and the predicted coefficient vector from the last output CCS sequence, as

$$e_c = \frac{1}{N_v} \sum_{j=1}^{N_v} \sqrt{\sum_{i=0}^{d-1} \left[\left(\hat{c}_{ij}^{*1} - c_{ij}^{*1} \right)^2 + \left(\hat{c}_{ij}^{*2} - c_{ij}^{*2} \right)^2 \right]} \quad (17)$$

where N_v denotes the number of verification data points, and the asterisk indicates that the value is from the last output sequence. Table 3 presents the experimental results of the trajectory prediction for the M2M1 and M2M2 models in the form of Chebyshev coefficients, and the values were calculated from the metric in (17). As shown in Table 3, the lowest error was recorded for the M2M1 model with a shorter CCS time span, and it seems that the higher the number

of input sequences, the more accurate the results become. However, the M2M2 model, with a longer CCS time span, produced superior prediction results when the input data were periodically missing. Moreover, the results for the M2M1 model deteriorated when data were more frequently missing, whereas the results for M2M2 were robust to the frequency of missing data. This could be attributed to the longer CCS time span because the results become insensitive to missing data owing to the uniform nature of the Chebyshev transform.

D. PREDICTION RESULTS-POSITIONS

The last output CCS sequence was inverse-transformed by (15) for all verification data and the accuracy was calculated using the mean absolute error (MAE) as:

$$e_p = \frac{1}{N_v N_p} \sum_{j=1}^{N_v} \sum_{i=1}^{N_p} \sqrt{\left(\hat{x}_{ij}^* - x_{ij}^* \right)^2 + \left(\hat{y}_{ij}^* - y_{ij}^* \right)^2} \quad (18)$$

where \hat{x}_{ij}^* and \hat{y}_{ij}^* denote the predicted x - and y -position inverse-transformed from the last output CCS sequence, respectively; and N_p denotes the number of predicted position sequences, where $N_p = t_p/T$. The error for each individually predicted point was calculated using the Euclidean distance between the two points, using (18). Table 4 presents the comparative experimental results of the trajectory prediction

TABLE 5. Comparative study II: prediction errors [m] for various configurations.

Models	No missing value	f_m		
		1 Hz	2.5 Hz	5 Hz
M2M1	2.299	2.334	2.582	2.562
M2M2	2.456	2.453	2.506	2.451
M2O	2.709	2.709	2.716	2.903
O2O	2.827	2.848	2.858	2.884

in the form of the inverse-transformed positions. To verify the efficacy of the proposed framework, the results of the baseline model in the conventional approach are presented in Table 4. The input data of the conventional approach features the x - and y -position, velocity, and heading, whereas the target data features only the x - and y -position. The conventional approach produced a total of 76 and 125 sequences for the time horizon in Table 2, and other hyperparameters were assigned the same values as in the proposed method. Note that the number of sequences was larger than that of the proposed framework, thus the training time was much longer in the case of the conventional approach, as will be discussed in the following paragraph.

In the case of the proposed framework, the results were generally consistent with those in Table 3. The prediction accuracy was higher in the case of the shorter CCS time span (M2M1), but the results for the longer CCS time span (M2M2) were more robust to the various patterns of missing data. Fig. 6 presents the trajectory prediction results from three samples with the M2M1 model. As shown in the figure, the prediction results became more stable as the sequence progressed. This aspect is one of the advantages of the proposed framework because the last predicted sequence in Fig. 6(c), (f), and (i) are the final prediction results for the prediction horizon. This appears to be a remedy for the chronic problem of LSTM, namely the inconsistency in the early sequences, as reported in [28].

The differences in the prediction accuracy between the models were not significant when no values were missing. However, in the case of the conventional approach, the prediction accuracy dropped dramatically even for the pattern in which missing data occurred most rarely. The experimental results in Table 4 verify that the proposed framework is significantly robust to incomplete sensor data.

Moreover, the number of variables in the final prediction results for the proposed framework was only $2d = 8$, whereas that for the conventional approach was $2 \times 125 = 250$. As mentioned previously, this aspect is likely to be beneficial for V2X-based applications in cooperative driving automation.

Furthermore, Table 5 presents the trajectory prediction results for other CS configurations. In the table, all configurations listed were robust to the various missing data patterns, but the best results were obtained with the M2M model depicted in Fig. 3. These results prove that the RNN structure is highly appropriate for the trajectory prediction

problems. Although not listed in Table 5, numerous configurations were tested to determine the optimal configuration for the proposed framework.

E. TRAINING COST

Under the experimental set-up of this study, the training costs of M2M1 and M2M2 models were 1h:18m:40s and 1h:4m:6s in time and 574 and 626 epochs, respectively. These results were comparable to those of the baseline, that is, 3h:5m:10s for 705 epochs, and attributed to the significant reduction in the number of sequences owing to the proposed Chebyshev transform-based feature engineering. For example, the lengths of the sequences were merely 6 and 10 for the input and output, respectively, in the M2M1 configuration, whereas it was 76 and 125 for the baseline.

V. CONCLUSION

Cooperative driving automation for a higher level of autonomy requires situation awareness to be robust to harsh environments, including adverse weather, because roadside sensors are exposed to the external environment. This study proposed a robust trajectory prediction framework based on a recurrent neural network. The robustness of the framework was proved by analyzing experimental data from a public vehicle trajectory dataset. Moreover, the proposed framework establishes the possibility of efficient communication between the connected vehicles and roadside units. This efficiency was attributed to the fact that the proposed framework requires only a few coefficients, rather than numerous time-sequential physical values, to be transmitted. This aspect is expected to play an important role in the realization of cooperative driving automation.

The present study has a limitation in that the feature data only include the trajectories of the ego vehicle for simplicity. Moreover, the implementation of fault injection was limited to the representative form of the missing data. Because the future trajectory can also be affected by the behavior of surrounding vehicles, in the future, we will include a discussion on the effect of various forms of incompleteness in the information about ego and surrounding vehicles in trajectory prediction, as well as on the relevant countermeasures. Notably, the sophisticated feature engineering proposed in this study can be integrated with other models.

REFERENCES

- [1] M. Goldhammer, E. Strigel, D. Meissner, U. Brunsmann, K. Doll, and K. Dietmayer, "Cooperative multi sensor network for traffic safety applications at intersections," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Anchorage, AK, USA, Sep. 2012, pp. 1178–1183.
- [2] G. Raipuria, F. Gaisser, and P. P. Jonker, "Road infrastructure indicators for trajectory prediction," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 537–543.
- [3] J. Bock, T. Beemelmans, M. Klösges, and J. Kotte, "Self-learning trajectory prediction with recurrent neural networks at intelligent intersections," in *Proc. 3rd Int. Conf. Vehicle Technol. Intell. Transp. Syst.*, Porto, Portugal, 2017, pp. 346–351.
- [4] Y. Hwang and S. B. Choi, "Awareness on present and future trajectory of vehicle using multiple hypotheses in the mixed traffic of intersection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 1–14, Oct. 2022, doi: 10.1109/TITS.2022.3169030.

- [5] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 103–111, Jun. 2019, doi: [10.1109/MVT.2019.2892497](https://doi.org/10.1109/MVT.2019.2892497).
- [6] F. Althe and A. de La Fortelle, "An LSTM network for highway trajectory prediction," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Yokohama, Japan, Oct. 2017, pp. 353–359.
- [7] *Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data*, US Dept. Transp., Federal Highway Admin., Washington, DC, USA, 2016, doi: [10.21949/1504477](https://doi.org/10.21949/1504477).
- [8] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Attention based vehicle trajectory prediction," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 1, pp. 175–185, May 2021, doi: [10.1109/TIV.2020.2991952](https://doi.org/10.1109/TIV.2020.2991952).
- [9] C. Luo, L. Sun, D. Dabiri, and A. Yuille, "Probabilistic multi-modal trajectory prediction with lane attention for autonomous vehicles," in *Proc. IEEE/RSS Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Jan. 2021, pp. 2370–2376.
- [10] A. Zyner, S. Worrall, and E. Nebot, "Naturalistic driver intention and path prediction using recurrent neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1584–1594, Apr. 2020, doi: [10.1109/TITS.2019.2913166](https://doi.org/10.1109/TITS.2019.2913166).
- [11] W. Ding and S. Shen, "Online vehicle trajectory prediction using policy anticipation network and optimization-based context reasoning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 9610–9616.
- [12] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*.
- [13] D. Yu, H. Lee, T. Kim, and S.-H. Hwang, "Vehicle trajectory prediction with lane stream attention-based LSTMs and road geometry linearization," *Sensors*, vol. 21, no. 23, p. 8152, Dec. 2021, doi: [10.3390/s21238152](https://doi.org/10.3390/s21238152).
- [14] R. Jiang, H. Xu, G. Gong, Y. Kuang, and Z. Liu, "Spatial-temporal attentive LSTM for vehicle-trajectory prediction," *ISPRS Int. J. Geo-Information*, vol. 11, no. 7, p. 354, Jun. 2022, doi: [10.3390/ijgi11070354](https://doi.org/10.3390/ijgi11070354).
- [15] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Changshu, China, Jun. 2018, pp. 1179–1184.
- [16] H. Jeon, J. Choi, and D. Kum, "SCALE-net: Scalable vehicle trajectory prediction network under random number of interacting vehicles via edge-enhanced graph convolutional neural network," in *Proc. IEEE/RSS Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 2095–2102.
- [17] C. Choi, S. Malla, A. Patil, and J. H. Choi, "DROGON: A trajectory prediction model based on intention-conditioned behavior reasoning," 2019, *arXiv:1908.00024*.
- [18] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Salt Lake City, UT, USA, Jun. 2018, pp. 1468–1476.
- [19] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 961–971.
- [20] Y. Zhou, H. Wu, H. Cheng, K. Qi, K. Hu, C. Kang, and J. Zheng, "Social graph convolutional LSTM for pedestrian trajectory prediction," *IET Intell. Transp. Syst.*, vol. 15, no. 3, pp. 396–405, Mar. 2021, doi: [10.1049/itr2.12033](https://doi.org/10.1049/itr2.12033).
- [21] C. Barrios and Y. Motai, "Improving estimation of vehicle's trajectory using the latest global positioning system with Kalman filtering," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 12, pp. 3747–3755, Dec. 2011, doi: [10.1109/TIM.2011.2147670](https://doi.org/10.1109/TIM.2011.2147670).
- [22] J. Wiest, C. Wohler, K. Schenk, and F. Kummert, "Incorporating categorical information for enhanced probabilistic trajectory prediction," in *Proc. ICMLA*, Miami, FL, USA, vol. 1, 2013, pp. 402–407.
- [23] C. Hermes, C. Wohler, K. Schenk, and F. Kummert, "Long-term vehicle motion prediction," in *Proc. IEEE Intell. Vehicles Symp.*, Xi'an, China, Jun. 2009, pp. 65–652.
- [24] J. Wiest, M. Hoffken, U. Kresel, and K. Dietmayer, "Probabilistic trajectory prediction with Gaussian mixture models," in *Proc. IEEE Intell. Vehicles Symp.*, Spain, Jun. 2012, pp. 141–146.
- [25] W. H. Press, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 1992, pp. 190–197.

- [26] I. N. Kovalenko, A. N. Nakonechnyi, and A. B. Romanov, "Chebyshev approximation of the distribution function of nonnegative random variables," *Cybern. Syst. Anal.*, vol. 32, no. 2, pp. 211–218, 1996, doi: [10.1007/BF02366534](https://doi.org/10.1007/BF02366534).
- [27] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Las Vegas, NV, USA, Oct. 2020, pp. 1929–1934.
- [28] F. Leon and M. Gavrilescu, "A review of tracking and trajectory prediction methods for autonomous driving," *Mathematics*, vol. 9, no. 6, p. 660, Mar. 2021, doi: [10.3390/math9060660](https://doi.org/10.3390/math9060660).



SUJIN KWAG received the M.S. and Ph.D. degrees in electrical engineering from Hanyang University, Seoul, South Korea, in 2002 and 2009, respectively. He is currently the Director with the AI-Bigdata Research Centre, Korea Automotive Technology Institute. His current research interest includes decision and control for autonomous vehicles. He is a member of the Korean Society of Automotive Engineers.



BYEONGJU KANG received the B.S. and M.S. degrees in mechanical engineering from Kunsan National University, Gunsan, South Korea, in 2014 and 2017, respectively. He is currently a Researcher with the AI-Bigdata Research Centre, Korea Automotive Technology Institute. His current research interests include situation awareness and control of autonomous vehicle.



WONHEE KIM (Member, IEEE) received the B.S. and M.S. degrees in electrical and computer engineering and the Ph.D. degree in electrical engineering from Hanyang University, Seoul, South Korea, in 2003, 2005, and 2012, respectively. From 2005 to 2007, he was with Samsung Electronics Company, Suwon, South Korea. In 2012, he was with the Power and Industrial Systems Research and Development Center, Hyosung Corporation, Seoul. In 2013, he joined the Institute of Nano Science and Technology, Hanyang University, Seoul, as a Post-doctoral Researcher and became a Visiting Scholar with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA, USA. From 2014 to 2016, he was with the Department of Electrical Engineering, Dong-A University, Busan, South Korea. He is currently an Associate Professor with the School of Energy Systems Engineering, Chung-Ang University, Seoul. His current research interests include nonlinear control and nonlinear observers, and their industrial applications. He has served as an Associate Editor for IEEE ACCESS, IEEE/ASME TRANSACTIONS ON MECHATRONICS, and the *Journal of Electrical Engineering and Technology*.



YUNHYOUNG HWANG received the B.S. degree in electrical engineering from Hanyang University, Seoul, South Korea, the M.S. degree from the Graduate School of Automotive Technology, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, and the Ph.D. degree in mechanical engineering from KAIST, in 2021. From 2008 to 2021, he was with the AI-Bigdata Research Centre, Korea Automotive Technology Institute. He is currently an Assistant Professor with the Department of Mechanical Engineering, Jeonbuk National University. His current research interest includes decision and control for autonomous vehicles. He is a member of the Korean Society of Automotive Engineers.

...