Received 10 October 2019; revised 2 December 2019; accepted 7 January 2020. Date of publication 13 January 2020; date of current version 13 February 2020.

Digital Object Identifier 10.1109/OJITS.2020.2965969

Surround Vehicle Motion Prediction Using LSTM-RNN for Motion Planning of Autonomous Vehicles at Multi-Lane Turn Intersections

YONGHWAN JEONG, SEONWOOK KIM, AND KYONGSU YI (Member, IEEE)

Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 08826, South Korea

CORRESPONDING AUTHOR: K. YI (e-mail: kyi@snu.ac.kr)

This work was supported in part by the Technology Innovation Program (Development and Evaluation of Automated Driving Systems for Motorways and City Roads) funded by the Korean Ministry of Trade, Industry and Energy under Grant 10079730, and in part by the Ministry of Land, Infrastructure, and Transport through Connected and Automated Public Transport Innovation (National R&D Project) under Project 18TLRP-B146733-01.

ABSTRACT This paper presents a surround vehicle motion prediction algorithm for multi-lane turn intersections using a Long Short-Term Memory (LSTM)-based Recurrent Neural Network (RNN). The motion predictor is trained using the states of subject and surrounding vehicles, which are collected by sensors mounted on an autonomous vehicle. Data on 484 vehicle trajectories were collected from real traffic situations at multi-lane turn intersections. 11,662 and 4,998 samples acquired from the vehicle trajectories were used to train and evaluate the networks, respectively. A motion planner based on Model Predictive Control (MPC) is designed to determine the longitudinal acceleration command based on the predicted states of surrounding vehicles. The future states of the subject vehicle derived by MPC is used as an input feature to reflect the interaction of subject and target vehicles in LSTM-RNN based motion predictor. The proposed algorithm was evaluated in terms of its accuracy and its effects on the motion planning algorithm based on the driving data sets. The improved prediction accuracy substantially increased safety by bounding the prediction error within the safety margin. The application results of the proposed predictor demonstrate the improved recognition timing of the preceding vehicle and the similarity of longitudinal acceleration with drivers.

INDEX TERMS Autonomous vehicle, intersection driving data, motion prediction, machine learning, recurrent neural network, long short-term memory, model predictive control.

I. INTRODUCTION

THE DEVELOPMENT of autonomous driving has been accelerated based on rapid improvements in sensors, actuators, processors, communications, and other technologies for autonomous vehicles (AVs). AVs improve road safety, convenience, and efficiency by reacting to potential risks. Three main challenges to achieving fully autonomous driving on urban roads are scene awareness, inferring other drivers' intentions, and predicting their future motions. Intersections on urban roads constitute especially complex environments that necessitate significant interactions with other drivers and predicting their movements [1]. Human drivers can predict the behavior of other drivers based on observed information and their experiences. Researchers have investigated developing prediction algorithms that can simulate a driver's intuition to increase safety when AVs and human drivers share roads.

Predicting driver behavior has been of broad interest for driver assistance systems and autonomous vehicles; researchers have formulated the problem in various ways and using numerous approaches. Motion prediction models can be classified into one of three categories: (1) physics-based; (2) maneuver-based; and (3) interaction-aware.

Researchers have utilized physics-based models for basic prediction in both constructed and unconstructed environments. These models predict future motion based on physics using kinematic or dynamic models starting from the given

The review of this article was arranged by Associate Editor Abdulla Hussein Al-Kaff.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

past states of prediction vehicles. Because simple models directly calculate future motion, physics-based models offer the advantages of easy modeling and minimal computational burden compared with other approaches. However, it is difficult to design kinematic or dynamic models that can adapt to driving situations and consider interactions between vehicles. Despite these limitations, researchers have utilized varied approaches, such as the Constant Velocity (CV) model [2], [3], the Constant Turn Rate and Velocity (CTRV) model [3], [4], the Constant Turn Rate and Acceleration (CTRA) model [3]–[6], the bicycle model [7], [8], the Intelligent Driver Model (IDM) [9], [10], and the path-following model [10]. Investigators have employed these models as a process update of various Kalman filters [10], Interacting Multiple Model (IMM) filters [3], and Monte-Carlo simulations [12], [13].

Maneuver-based models classify the driver's intention into a predefined set of groups, called maneuvers, examples of which include staying in the lane, changing lanes, and turning at intersections. After a driver's maneuver is determined, the future trajectory is predicted based on the models prepared to represent each maneuver. Therefore, maneuver-based models can consider not only the states of target vehicles, but also individual drivers' intentions, i.e., drivers can exhibit similar behaviors, but the system will infer different intentions according to the driving situation, which augments prediction accuracy. However, interactions between vehicles are not reflected in maneuver-based models. Based on these advantages, many approaches have been proposed to infer driver intention, such as the Gaussian mixture model [14], [15], multilayer perceptron [16], logistic regression [17], Bayesian networks [18], [19], Markov chain [20], Relevance Vector Machine (RVM), Support Vector Machine (SVM) [21], Hidden Markov Model (HMM) [22], [23], and Recurrent Neural Network (RNN) [24]–[26].

These physics- and maneuver-based models focus on predicting the future motion of individual targets, without considering interactions between neighboring vehicles. Therefore, a possibility exists of misunderstanding targets' behaviors when multiple vehicles are driving closely together. Interaction-aware models reflect interactions between surrounding vehicles, and predict future motions of detected vehicles simultaneously as a scene. These models can predict the most realistic behavior by reflecting interactions between vehicles. However, the prediction algorithm is more complex than other approaches, which increases the computational burden. Consequently, these models are often developed and evaluated in offline simulations. Coupled HMMs [27], double-layer HMMs [28], and rule-based models [29], [30] are used to reflect interactions in prediction models. This methodology is frequently employed in conjunction with the aforementioned physics- and maneuver-based models.

Researchers have widely studied maneuver- and interaction-based models for predicting vehicle intentions and behaviors. One proposed approach was an interaction-aware model, called the foresighted driver model, that uses the similarities of spatio-temporal trajectories [31]. In other research, maximum a posteriori probability was utilized to estimate driver-specific critical gaps, which constitutes an important parameter in decisions on crossing at unsignalized intersections, and evaluated in a driving simulator [32]. Researchers have investigated predicting the intended path at intersections based on HMM with real traffic data sets; this approach uses vehicle-to-vehicle communication to collect the data, which is composed of GPS and chassis sensor data [23]. Similar approaches entailed training an SVM and random decision forests [21], and an artificial neural network was applied to design a motion predictor. The influence network, which simultaneously considers agents and the environment, was trained and evaluated by driving data extracted from a surveillance camera at an intersection [33]. Investigators have also used Long Short-Term Memory (LSTM)-RNN to predict the future intention of targets at a roundabout [34]. A similar approach was employed to infer the intentions of targets approaching intersections based on GPS, Inertial Measurement Unit (IMU), and odometer data [35].

A careful review of the literature reveals less focus on predicting trajectories at intersections than on motorways. In addition, public data sets for analyzing driver behavior at intersections are largely insufficient. The NGSIM data set (from the Next Generation Simulation program), which researchers have frequently used, includes a small amount of data for intersections located along Lankershim Boulevard in Los Angeles, California, and Peachtree Street in Atlanta, Georgia [24].

However, it is highly challenging to develop an algorithm that can be applied to AVs because NGSIM data are extracted by applying the tracking algorithm to images collected from an overhead camera. This limitation in how data are acquired also applies to data from the Ko-PER project [36]. To overcome the limitations in public data sets, a number of researchers have collected data from vehicles equipped with sensors [10], [15], [21], [23], [25], [26], [33]–[35], but very few studies are based on data collected in real time and in real traffic conditions. Therefore, we focus on motion prediction at multi-lane turn intersections based on information from on-board sensors in autonomous vehicles.

This study focused on improvement of in-lane target recognition and achieving human-like acceleration decisions at multi-lane turn intersections by introducing the learning-based target motion predictor and prediction-based motion predictor. A data-driven approach for predicting trajectory and velocity of surrounding vehicles on urban roads at multi-lane turn intersections is described. LSTM architecture, a specific kind of RNN capable of learning long-term dependencies, is designed to manage complex vehicle motions in multi-lane turn intersections. The proposed LSTM-based RNN is trained using data collected from surrounding

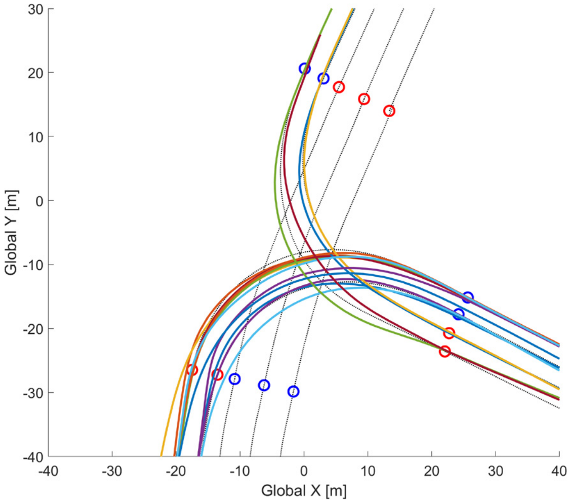


FIGURE 1. Examples of driving trajectories of human drivers turning in traffic at a multi-lane turn intersection.

vehicles obtained by sensors on an AV. Fig. 1 shows several examples of the surrounding vehicle tracks measured while the data collection vehicle was driving with traffic participants. These trajectories show the various atypical trajectories of vehicles passing through a multi-lane turn intersection, which is hard to model as a single prediction model by a conventional approach. Real data captured on urban roads in Seoul are used to evaluate the accuracy of the motion predictor and the improvement of motion planning performance in multi-lane turn intersections.

The main contributions of this work are as follows:

1) a LSTM-RNN based single model is defined to predict the various motions of surrounding targets at multi-lane turn intersections;
2) the future states of the subject vehicle derived by the MPC-based motion planner are employed as an input feature of LSTM-RNN to reflect the interaction of subject and target vehicles in a prediction horizon;
3) the states of only subject and surrounding vehicles are used to exclude dependency on road structure or digital maps, and
4) a motion predictor is designed based on real traffic data obtained using AV-mounted sensors.

II. OVERVIEW OF THE LSTM-RNN BASED MOTION PREDICTOR

This study focused on a motion predictor for surrounding vehicles in multi-lane turn intersections, which is a topic not yet addressed in the extant literature. Research with intersections concentrated on maneuver-level prediction of activities, such as cross, stop, or making a left turn. Trajectory-level prediction, which was the goal of this study, was commonly covered in structured environments that enforce classifiable behavior, such as lane-keeping or lane-changing. However, trajectory-level prediction is more important in multi-lane turn intersections because different drivers will have varied driving behaviors as they travel through the intersections, as shown in Fig. 1. Based on these analyses, the problem can be summarized as shown in Fig. 2. As shown in the figure, future trajectories of the surrounding vehicles that travel with the subject vehicle are predicted based on the states of subject and surrounding vehicles.

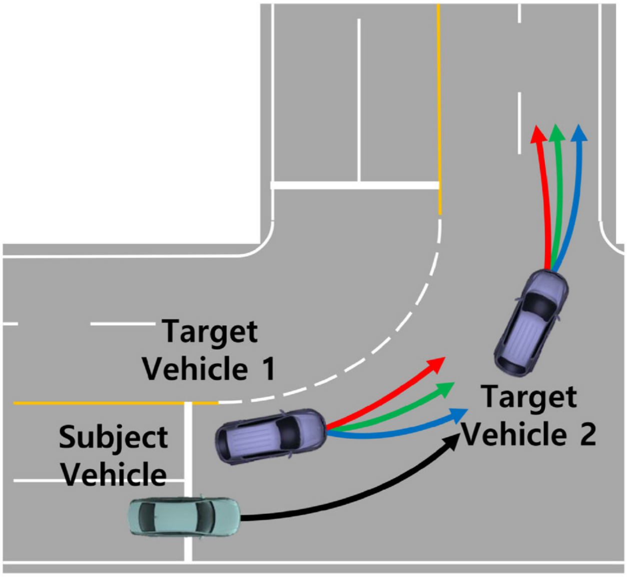


FIGURE 2. Defining the problem of predicting the motions of surrounding vehicles in multi-lane turn intersections.

Fig. 3 depicts the architecture of the surrounding target trajectory predictor. The proposed algorithm comprises three modules: (1) a data encoder; (2) an LSTM-based RNN; and (3) a data decoder. As previously mentioned, the proposed architecture uses the surrounding vehicles' states estimated by the perception algorithm, which relies on six ibeo Lux laser scanners. The output of the proposed predictor, which predicted surrounding vehicles' states, is utilized to determine the desired longitudinal acceleration in real traffic at intersections.

III. LSTM-RNN BASED MOTION PREDICTOR

A. DATA SET

Multi-lane turn intersections were chosen as the target roads for this study. These intersections are substantially more complex than highways or other structured environments where typical maneuvers include lane-keeping or lane-changing. Maneuvers at multi-lane turn intersections are complex, and it is difficult to classify specific maneuvers. We collected data that were appropriate for considering the behaviors of vehicles in intersections; to avoid over-fitting of the learning-based motion predictor, we chose multiple intersections. Moreover, the data collection vehicle drove with real traffic to reflect the characteristics of real driving conditions. The details of the data set will be discussed in the following section.

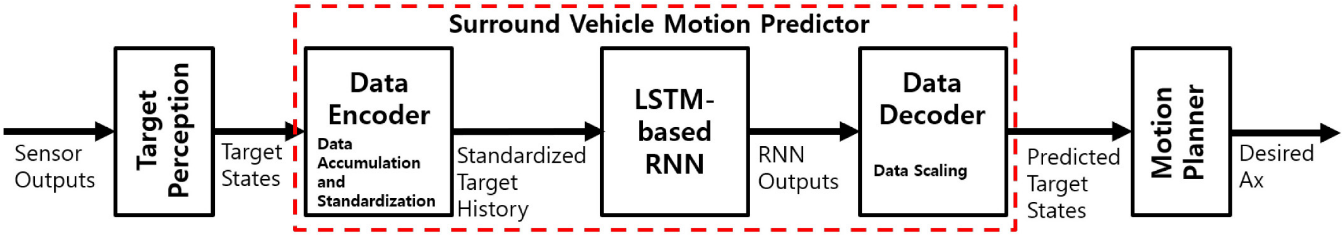


FIGURE 3. Overall architecture of the proposed surrounding target trajectory predictor.

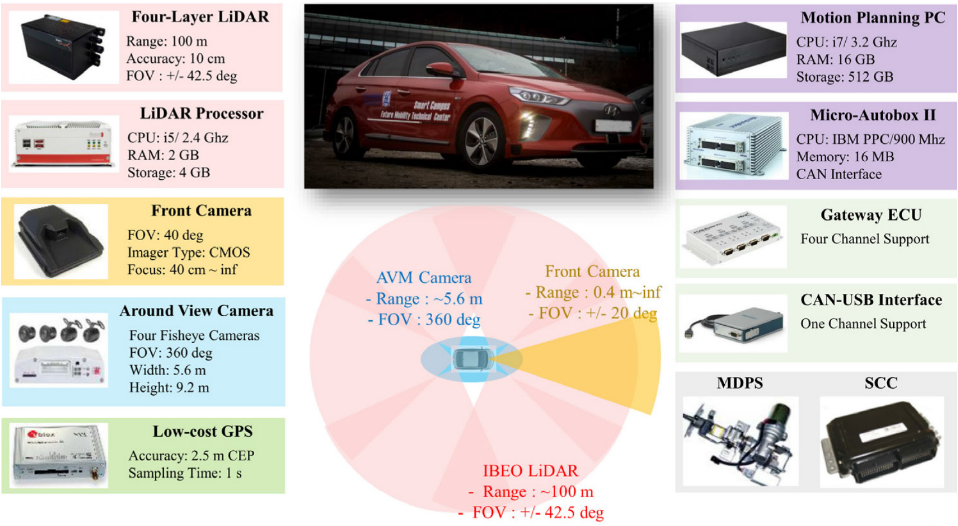


FIGURE 4. Configuration of data collection vehicle.

1) DATA COLLECTION VEHICLE

The data for this study were collected from an AV equipped with multiple sensors, as shown in Fig. 4. In this work, the human-driven AV is used to collect data on the motions of the surrounding vehicles in real traffic flow. Specifically, we used six ibeo LUX sensors with ibeo.HAD Feature Fusion, which operates at 25 Hz and detects traffic participants at a range of up to 100 m, to detect surrounding vehicle motion [37]. This LiDAR system provides relative position, heading, velocity, and box size in local coordinates of the data collection vehicle with classification information. In addition, a front camera, Around-View Monitoring (AVM), and low-cost GPS were employed to acquire the lanes, road markers, and global position of the subject vehicle. A gateway engine control unit obtained the outputs of the chassis sensors, and this information was fused with a digital map to acquire the vehicle's precise global position in urban road environments. All sensor data were synchronized and stored on an industrial PC. The Micro-Autobox II (dSPACE, Inc., Wixom, MI, U.S.A.) and a motor-driven power steering/smart cruise control module were utilized to control and actuate the subject vehicle in autonomous driving.

2) TARGET ROADS

The data on surrounding vehicle tracks were collected from an AV driving on the urban roads of Gwanak-gu, Seoul, South Korea. Fig. 5 presents a satellite map of the study road with the driving route for the data collection highlighted in red color, and this route includes two signalized multi-lane left turn intersections. As mentioned in Section III-A, the sensors collected the surrounding vehicle data based on the local coordinates of the subject vehicle. To generate appropriate data for learning the motion patterns of surrounding vehicles, all data should be in fixed global coordinates with time indices. In this study, we used the estimated global

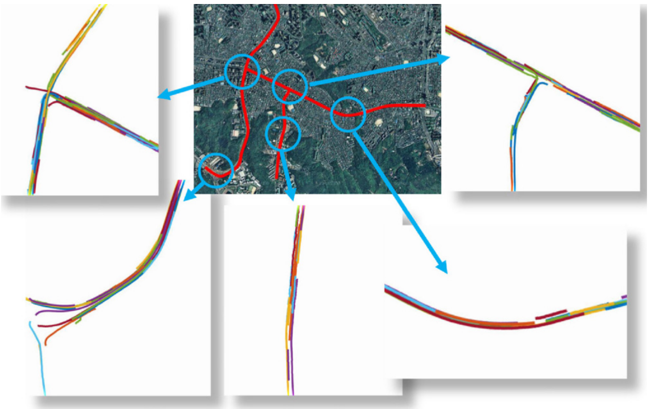


FIGURE 5. Data collection roads and surrounding vehicle tracks accumulated in global coordinates.

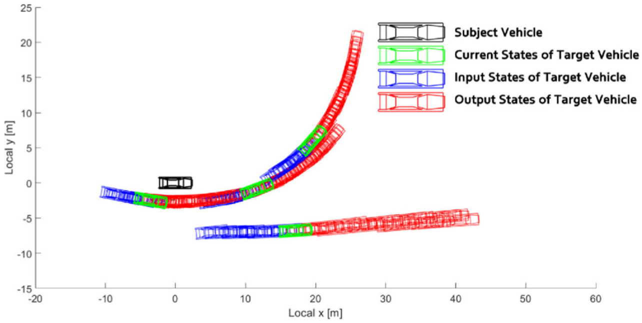


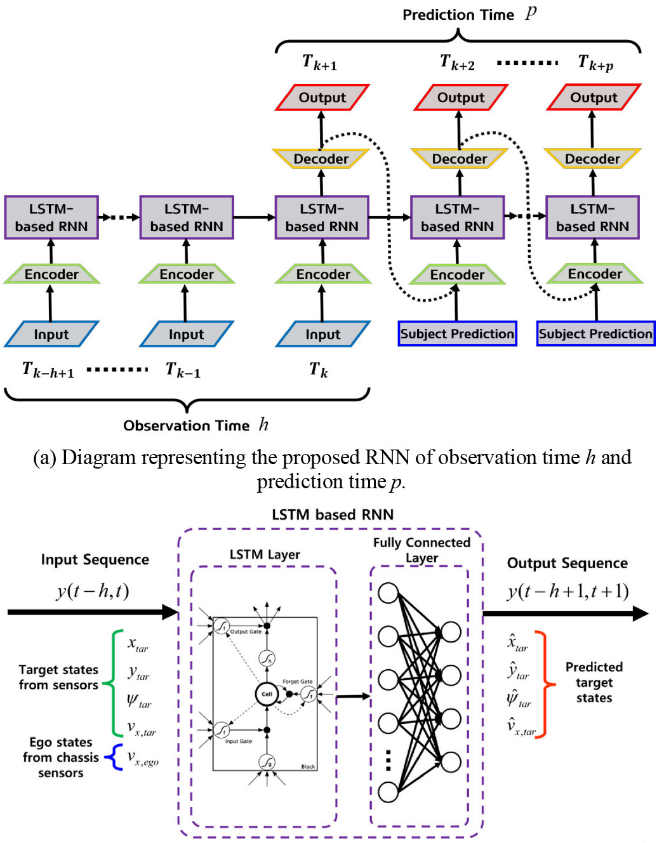
FIGURE 6. Example of data extraction from accumulated vehicle trajectories.

position and heading of the subject vehicle to transform the position and heading of the surrounding vehicles from local to global coordinates. In order to acquire high accuracy positioning results, outputs of low-cost GPS were fused with lane mark information from the AVM image, which is introduced in [38]. By using the GPS/Lane mark fused system, we stabilized the positioning error to a suitable level of under 0.5 m Circular Error Probable (CEP) in urban situations. Through this data processing, vehicle trajectories for a specific period can be reconstructed in the local coordinate system. Fig. 5 also displays example trajectories of the accumulated data collected from five sections of the data collection roads.

3) DATASET SELECTION

The real traffic data collected, as shown in Fig. 5, contained 4,312 trajectories of vehicles driving around the subject vehicle. Among these trajectories, 484 data points were collected while passing the intersection, which are used to train and evaluate the proposed motion predictor.

The training data set was generated using the 484 trajectories collected by driving in real traffic. As mentioned in Section IV-B, each trajectory includes a time index. Therefore, we could extract the previous and subsequent states of surrounding vehicles accumulated in global coordinates at specific moments. Fig. 6 presents samples of extracted data, in which the subject and surrounding target vehicles are in black and green color, respectively. The previous and subsequent states of surrounding target vehicles with respect to this moment are represented in blue and red color, respectively, and these sequences of blue and red vehicles are used as input and output sequences of the proposed motion predictor. After postprocessing of the collected data, a total of 16,660 data samples were generated, that were divided into 11,662 data samples for training and 4,998 for evaluation.



(a) Diagram representing the proposed RNN of observation time h and prediction time p.

(b) Conceptual diagram of the single step of the LSTM-RNN predictor.

FIGURE 7. Diagram of the proposed LSTM-RNN based motion predictor.

B. MOTION PREDICTOR

Vehicle motion is continuous based on vehicle dynamics; in other words, the future motion of the vehicle depends on the sequential previous motion. In addition, driver intention is also important in predicting vehicles' future motions. Previous motion can be measured using sensors on AVs, but it is difficult to infer driver intention based on rules. Particularly on roads where irregular behavior occurs frequently, such as at intersections and in roundabouts, it is highly challenging to apply conventional maneuver-based approaches to infer driver intention and predict future motion. Therefore, we propose a data-driven approach to predict future motions of surrounding vehicles based on their previous motions. The motion predictor based on LSTM-RNN architecture that we propose in this work used only information collected from the sensors on an autonomous vehicle. The contribution of the network architecture of this study is that the subject vehicle's future states are used as an input feature of prediction horizon as shown in Fig. 7 (a). The future states of the subject vehicle are determined by MPC-based motion planner which will be discussed in Section IV. Therefore, this feedback of the outputs of motion planner makes the LSTM-RNN based motion predictor can reflect the interaction of subject and surrounding targets occurring during the prediction horizon.

1) NETWORK ARCHITECTURE

An RNN is an artificial neural network that is appropriate for utilization with sequential data, such as speech or text recognition written in natural language. RNNs can also

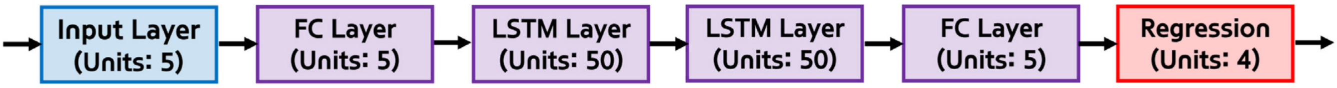


FIGURE 8. Depiction of the individual layers of the LSTM-RNN based motion predictor.

be used with time series data, in which the pattern of the data depends on time flow. Recurrence in RNNs allows for modeling the correlations between consecutive data points in a sequence. This feature is realized by having the same network for each time step and passing activations to a successor [39].

The RNN can contain feedback loops, which permits activations to flow interactively in the loop. This feature enables processing sequences of inputs by persisting the activations over multiple steps. In other words, the network can memorize previous information and predict the future after specific steps by applying the same network iteratively. Fig. 7 (a) illustrates the unrolled structure of the RNN used in this study for an observation horizon h and prediction horizon p. As the figure shows, the activations in each step are passed to the same network of the next time step and updated with new input data. This means that one set of weights of the RNN is repeated over the prediction horizon by matching the formats of the input and output sequences. The details of the encoder and decoder will be discussed in subsequent sections of this paper.

The single prediction step using the proposed RNN is conceptually expressed in Fig. 7 (b), with LSTM used as a network cell. LSTM can avoid the vanishing gradient problem by making the error flow backward through unlimited numbers of virtual layers. This property prevents the error from increasing or decaying over time, which would make the network train inappropriately [40].

Fig. 8 presents the individual layers of the LSTM-RNN with the number of units in each layer. This structure is determined by comparing the accuracy of 72 RNNs, which consisted of combinations of four input sets and 18 network configurations.

2) INPUT AND OUTPUT FEATURES

Researchers collected the data for these approaches using infrastructure sensors, such as observation cameras and data collection vehicles parked on target roads. However, because these approaches do not consider driving situations around autonomous vehicles, it is difficult to guarantee their performance in AVs in real traffic. In order to apply the motion predictor to driving AV, the velocity of the data collection vehicle is added to the input sequences. As shown in Fig. 7 (b), the input sequence consists of relative X/Y position, relative heading angle, velocity of surrounding target vehicles, and velocity of the data collection vehicle. The output sequence is the same as the input sequence, such as relative position, heading, and velocity.

3) ENCODER AND DECODER

The perception algorithm estimates the target vehicle's states, relative X/Y position, relative heading angle, and relative velocity with track ID. The encoder accumulates the multi-target states with respect to track ID to construct the input sequences. After target states were accumulated, the neural network input data should be preprocessed to improve stability and performance. Moreover, the data are prepared based on certain techniques, such as normalization and standardization, to scale the input and output data to train and use the neural network model. In this study, we introduce an encoder and decoder to process the input from the sensors and output from the RNN, respectively. This encoder standardizes each component of input data to rescale the data to a mean of 0 and a standard deviation of 1, and the decoder destandardizes the output data to scale back to real-world units using the same parameters as those in the encoder. Parameters μ and σ were determined using the 11,662 training data samples only, and stored for reuse in validating the proposed algorithm and applying it to the AV. The input to the network is standardized as follows:

$$\bar{x}_{t,n} = \frac{x_{t,n} - \mu_n}{\sigma_n} \tag{1}$$

where $x_{t,n}$ is the n-th component of the input data, such as position or heading at time t. In addition, $\bar{x}_{t,n}$ is the standardized input of $x_{t,n}$, and μ_n and σ_n are the mean and standard deviation of the n-th component, respectively. Therefore, a total of 5 μ_n and σ_n were prepared based on the training data set. Since the network is trained by standardized data using the encoder, the outputs of the neural network are standardized. Therefore, the decoder destandardizes the output data to rescale back to the physical scale using the same parameters as those in the encoder. The decoder makes the network output as actual prediction results, which are usable for the motion planner.

4) SEQUENCE LENGTH

The sequence lengths of the RNN input and output are another important factor in improving prediction performance. For this study, we trained the network architecture depicted in Fig. 8 using several candidate input sequence lengths to identify the optimal length. Sequences of 5, 10, 15, 20, 25, and 30 steps with a sampling time of 100 ms were compared, and 15 steps showed relatively accurate results even though the observation time is short among the candidates; thus, the input sequence of 15 steps, 1.5 s, is used. Meanwhile, the length of the output sequence should be determined considering the design of the algorithms that use the prediction results. For example, if the motion planner determines an output with a prediction horizon of 2 s, the target motion predictor should provide 2-s prediction results for the surrounding vehicles. In this study, a motion planner is designed to determine the desired acceleration based on the current state of the subject vehicle and the prediction results for the surrounding targets. As mentioned in Section III-B and shown in Fig. 7 (a), the prediction horizon p can be adjusted flexibly by changing the number of iteration loops. Therefore, the proposed architecture can be applied to a variety of motion planners and controllers with different prediction horizons.

IV. MOTION PLANNING BASED ON SURROUNDING VEHICLE MOTION PREDICTION

The key objectives for motion predictors for autonomous driving are simulating the behaviors of human drivers and improving safety. In everyday driving, experienced drivers predict possible risks based on their observations of surrounding vehicles and ensure safety by modifying their behaviors before risks arise. To realize the human-like motion planning, a prediction-based autonomous vehicle motion planner with consideration of the human driver's future behavior is designed based on Model Predictive Control (MPC) approaches. In order to reflect the predicted behavior of surrounding human-driven vehicles, the reference states and constraints of the MPC are defined based on the prediction results from the LSTM-RNN based motion predictor. The cost function of the motion planner is determined as follows:

$$J = \sum_{k=1}^{N_p} \left(x(k|t) - x_{ref}(k|t)\right)^T Q\left(x(k|t) - x_{ref}(k|t)\right) + R \sum_{k=0}^{N_p-1} u(k|t)^2 + R_{\Delta u} \sum_{k=0}^{N_p-2} \left(u(k+1|t) - u(k|t)\right)^2 \quad (2)$$

where k and t are the prediction step index and time index, respectively; $x(k|t)$ and $x_{ref}(k|t)$ are the states and reference of the MPC problem, respectively; $x(k|t)$ is composed of travel distance p_x and longitudinal velocity v_x; $x_{ref}(k|t)$ consists of reference travel distance $p_{x,ref}$ and reference longitudinal velocity $v_{x,ref}$; $u(k|t)$ is the control input, which is the longitudinal acceleration command; N_p is the prediction horizon, which is determined by dividing prediction time T_p by sampling time dt; T_p of 2 s and dt of 100 ms are used to configure the motion predictor, and therefore N_p of 20 to define the model predictive controller; and Q, R, and $R_{\Delta u}$ are the weight matrices for states, input, and input derivative, respectively, and these weight matrices were tuned to obtain control inputs from the proposed controller that were as similar as possible to those of human-driven vehicles. As mentioned later, two sets of weight matrices were used for the controller, for the proposed and conventional motion predictors.

The constraints were defined to consider dynamics, actuator limits, ride comfort, and safety. First, the simplified first-order delay model was used to define the dynamic constraints for the controller. Since the autonomous driving platform is based on an electric vehicle (i.e., Hyundai IONIQ Electric 16MY) and the actuator of the smart cruise control is used to control the vehicle, we conducted the characteristic analysis of the longitudinal behavior. Based on the analysis results, the first- order delay model well simulated the behavior of the electric vehicle, even though it was a simple model. Therefore, the dynamic constraints for the controller were defined as follows:

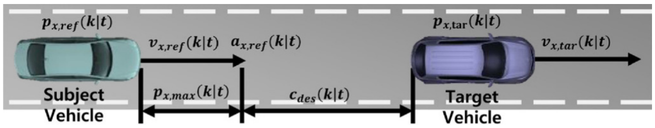


FIGURE 9. State and parameter description of MPC-based motion planner.

$$\begin{bmatrix} p_x(k+1|t) \\ v_x(k+1|t) \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 - dt/\tau \end{bmatrix} \begin{bmatrix} p_x(k|t) \\ v_x(k|t) \end{bmatrix} + \begin{bmatrix} 0 \\ dt/\tau \end{bmatrix} u(k|t) \quad (3)$$

where τ is the actuator delay, which is defined by vehicle characteristics. The constraints for the control input are defined as follows for magnitude and jerk, respectively:

$$u_{\min} \leq u(k|t) \leq u_{\max} \quad (4)$$

$$\|u(k+1|t) - u(k|t)\| \leq S \quad (5)$$

where u_{min}, u_{max}, and S are the minimum/maximum control input and maximum slew rate of input, respectively. For this study, u_{min} of -3m/s^2, u_{max} of 1m/s^2, and S of 5m/s^2 are used.

The predicted states of the surrounding target vehicle were used to define $x_{ref}(k|t)$ and the boundary conditions of p_x and v_x. We summarize how we determined the reference state in Fig. 9. The initial reference states were defined using the subject vehicle's current state as follows:

$$p_{x,ref}(0|t) = 0 \quad (6)$$

$$v_{x,ref}(0|t) = v_{x,sub}(t) \quad (7)$$

From this initial condition, the reference states were updated iteratively using the following equations:

$$c_{des}(k|t) = t_{gap} \cdot v_{x,ref}(k|t) + c_{\min} \quad (8)$$

$$u_{ref}(k+1|t) = k_1 \cdot \left(v_{x,tar}(k|t) - v_{x,ref}(k|t)\right) + k_2 \cdot \left(c_{des}(k|t) - c(k|t)\right) \quad (9)$$

$$v_{x,ref}(k+1|t) = v_{x,ref}(k|t) + u_{ref}(k+1|t) \cdot dt \quad (10)$$

$$p_{x,ref}(k+1|t) = p_{x,ref}(k|t) + v_{x,ref}(k|t) \cdot dt + 0.5 \cdot u_{ref}(k+1|t) \cdot dt \quad (11)$$

where t_{gap} and c_{min} are the time gap and minimum clearance to the in-lane target vehicle, respectively. In this study, t_{gap} of 1.2 s, c_{min} of 3 m, k_1 of 0.4, and k_2 of 1.0 were used to determine the reference input u_{ref}.

TABLE 1. Parameters for MPC-based motion planner.

Parameters	Description	Values
T_p	prediction horizon in seconds	2 s
dt	sampling time	0.1 s
N_p	prediction horizon in step	20
τ	actuator delay	1 s
u_{min}	minimum input magnitude	-3 m/s^2
u_{max}	maximum input magnitude	1 m/s^2
S	maximum input jerk	5 m/s^2
t_{gap}	desired time gap	1.2 s
c_{min}	minimum clearance	3 m
k_1	gain for relative velocity	0.4
k_2	gain for clearance	1.0

The position and velocity boundaries were determined based on the predicted states as follows:

$$p_{x,\max}(k|t) = p_{x,tar}(k|t) - c_{des}(k|t) \tag{12}$$

$$p_{x,\min}(k|t) = 0 \tag{13}$$

$$v_{x,\max}(k|t) = \min\left(v_{x,ret}(k|t),\ \ v_{x,limit}\right) \tag{14}$$

$$v_{x,\min}(k|t) = 0 \tag{15}$$

where $v_{x,\text{limit}}$ is the velocity limit of the subject vehicle, defined by driver input or the speed limit on the driving road. The parameters for the MPC-based motion planner are summarized in Table 1.

V. PREDICTION PERFORMANCE ANALYSIS AND APPLICATION TO MOTION PLANNING

The proposed motion predictor was evaluated through driving data-based simulation. We used 4,998 of the data samples as the evaluation data set to compare prediction accuracy between the base and proposed algorithms in Section V-A. The results for applying the motion predictor to the motion planner were described in Section V-B.

The prediction results from the proposed algorithm were compared with the results from three base algorithms. Specifically, the path-following model with constant velocity (CV/Path) and the path-following model with traffic flow (V$_{flow}$/Path) [41] were used to predict the surrounding vehicles' motion within intersections. The path-following model, which is used for those two predictors, assumed that vehicles would be following the roads, with the roads to follow being defined by the nearest path on a digital map. This constituted a reasonable assumption because there are no lane markers in intersections, and this feature of intersections makes it difficult to predict the surrounding vehicles' motion based on lane information from vision sensors. Therefore, it is appropriate to integrate the digital map with surrounding vehicles' states for intersections.

In addition, the prediction performance of a CTRV model was compared with that of the proposed algorithm; CTRV models are frequently used for target tracking and motion prediction [3], [4], [42]. When choosing the base algorithms, we excluded models based on acceleration because it is highly challenging to estimate longitudinal and yaw acceleration of surrounding vehicles to the level applicable to prediction models using only on-board sensors.

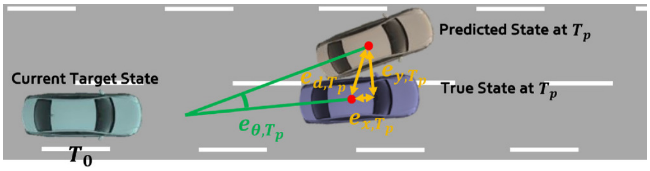


FIGURE 10. Definition of prediction error at prediction time T_p.

TABLE 2. Prediction errors of the proposed motion predictor.

Error	Proposed			CV/Path model		
	Mean	STD	RMSE	Mean	STD	RMSE
$e_x\ (m)$	0.53	1.98	2.23	1.49	4.09	4.35
$e_y\ (m)$	-0.02	0.37	0.45	-2.25	2.13	3.10
$e_\theta\ (°)$	-0.56	1.87	2.97	-4.60	6.61	8.05
$e_v\ (km/h)$	-0.16	0.65	0.67	0.80	1.23	1.46

A. PREDICTION ACCURACY ANALYSIS

The prediction error was defined to compare prediction accuracy between the true and predicted states. All 4,998 evaluation samples were processed to compare prediction error between the proposed and base algorithms. The x position error $e_{x,Tp}$, y position error $e_{y,Tp}$, heading error $e_{\theta,Tp}$, and velocity error $e_{v,Tp}$ were defined as follows:

$$e_{x,Tp} = p_{x,Tp} - \hat{p}_{x,Tp} \tag{16}$$

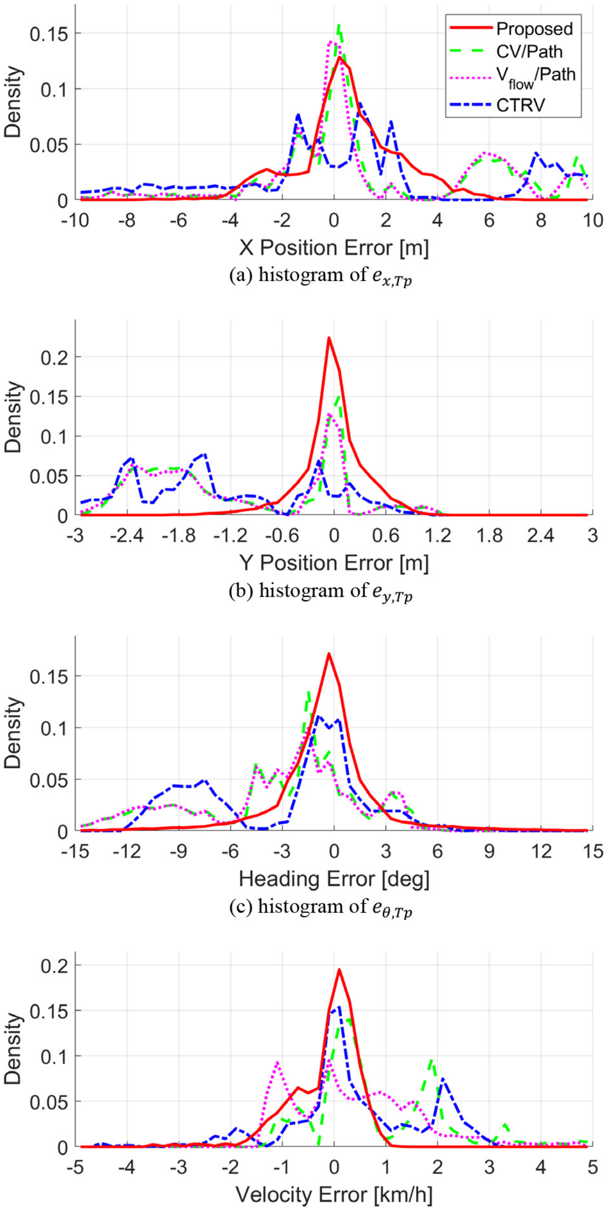
$$e_{y,Tp} = p_{y,Tp} - \hat{p}_{y,Tp} \tag{17}$$

$$e_{\theta,Tp} = \theta_{Tp} - \hat{\theta}_{Tp} \tag{18}$$

$$e_{v,Tp} = v_{Tp} - \hat{v}_{Tp} \tag{19}$$

Among the prediction errors, $e_{x,Tp}$ and $e_{y,Tp}$ were defined as local coordinates that originated in the true state at T_p, as shown in Fig. 10. This error definition prevented misinterpretation of the prediction results due to coordinate changes when the vehicle rotates in the intersection.

The analysis results of the prediction error for proposed and base algorithms are depicted in Fig. 11. The means and standard deviations of proposed algorithm and CV/Path model are summarized in Table 2. The proposed algorithm shows significantly reduced prediction errors compared to the base algorithms in terms of mean, standard deviation (STD), and root mean square error (RMSE). First, the proposed algorithm exhibits a bell curve with a near zero mean, which means that the algorithm precisely predicted the intended direction of the human drivers. Meanwhile, the x and y position errors of CV/Path and V$_{\text{flow}}$/Path show two bell curves because the base algorithm chose the following path among the predefined paths on the digital map. The shifted bell curve was created on only one side because it was possible to make multi-lane left turns at the intersections where the evaluation data were acquired; if multi-lane right turn cases are added, another curve will be created on the other side of the proposed algorithm's results. The CTRV model showed more distributed results because the yaw rate change within the prediction horizon was not considered, and the yaw rate estimation error is directly related to prediction.



(a) histogram of $e_{x,Tp}$

(b) histogram of $e_{y,Tp}$

(c) histogram of $e_{\theta,Tp}$

(d) histogram of $e_{v,Tp}$

FIGURE 11. Comparison of prediction error between proposed and base algorithm.

Second, the three-sigma ranges of $e_{x,Tp}$, $e_{y,Tp}$, and $e_{v,Tp}$ are bounded within reasonable levels for the motion planner. In other words, prediction result-based path generation can guarantee the safety of AVs based on the prediction results because the three-sigma range of $e_{y,Tp}$ is -1.09m to 1.13m, which is within the margin of the lane width. For longitudinal motion planning, the error levels of $e_{x,Tp}$ and $e_{v,Tp}$ were also within the control margin of the motion planner. Therefore, motion planning for AVs based on the proposed prediction algorithm can increase the safety and passenger acceptance of autonomous driving. The details of applying the motion planner of the proposed motion predictor will be discussed in the following section.

B. MOTION PLANNING APPLICATION

The results of applying the proposed predictor to a motion planning algorithm were summarized into results for a representative case and the results from analyzing the entire evaluation data set for multi-lane turn intersections.

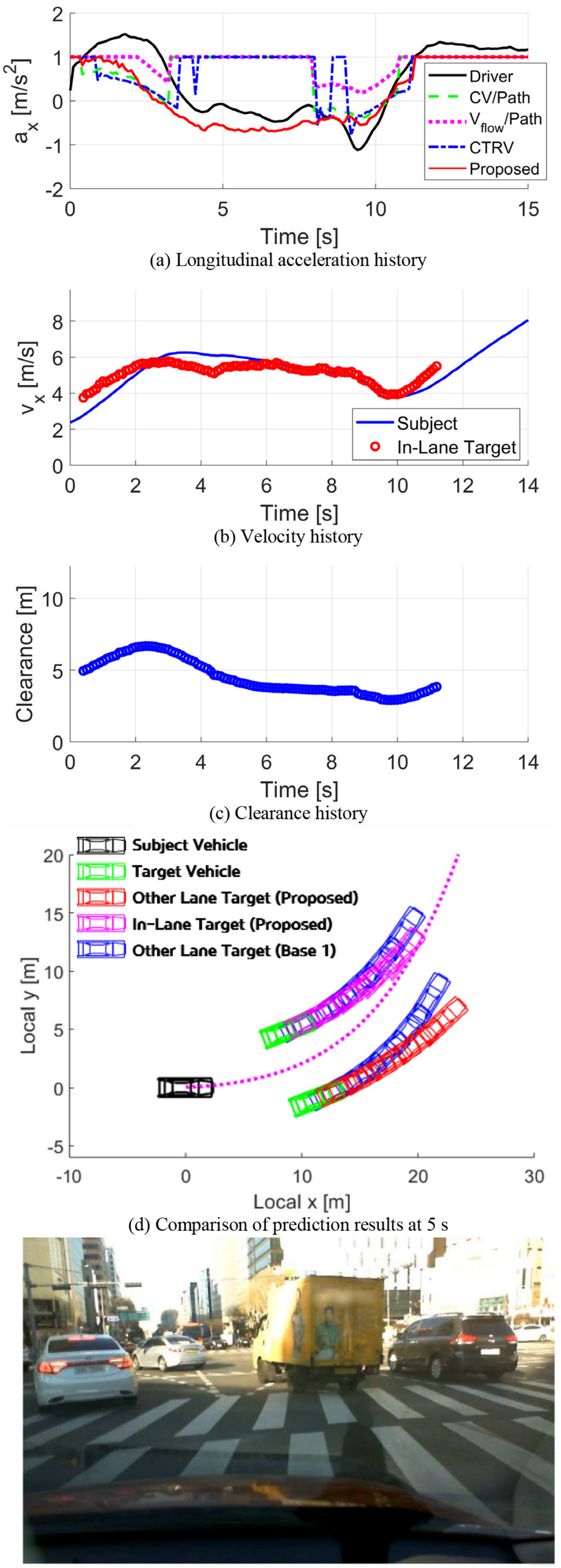
1) CASE STUDY OF A MULTI-LANE LEFT TURN SCENARIO

Fig. 12 (a) to (c) presents the results for the motion planner with the proposed algorithm and the base algorithms as longitudinal acceleration, velocity, and clearance history. Fig. 12 (a) shows the driver acceleration history and the motion planner command output with the proposed and base algorithms. In this example, the proposed motion predictor improved the longitudinal motion planning of all three algorithms in a multi-lane turn intersection. The first scenario is the turning scenario in the inner lane of two turning lanes.

First, the in-lane target decision performance improved when the vehicle traveled through the intersections. Even when the in-lane target vehicle was not following the intersection guide line, the proposed motion predictor precisely predicted the future trajectory of the target vehicle and accurately classified the target as an in-lane target. The trajectory prediction results of proposed and base algorithms are depicted in Fig. 12 (d), with the corresponding dashboard camera image in Fig. 12 (e). As shown in Fig. 12 (e), the white sedan, which is the in-lane target, deviated from the path beyond the lane width by turning left inward. In this case, the three base algorithms determined that this vehicle had exited the driving lane, whereas the proposed algorithm predicted the vehicle's returning maneuver and determined that it was an in-lane target. This precise prediction achieves human-like in-lane target decisions and acceleration commands.

Second, the acceleration command of the proposed algorithm simulates that of a human driver, as shown in Fig. 12 (a), except with less jerking movement than with human drivers. The motion planner based on prediction shows similar behaviors to those of human drivers because the proposed controller simulates a human driver's decision-making process, observing vehicles, predicting their motion, and determining the desired response motion. However, if the prediction results are inaccurate, there may be no significant improvement of motion planning performance over the performance of the controller using the current states only. The proposed algorithm showed more similar acceleration to that of human drivers than did the base algorithms, which showed less accurate prediction results.

Finally, the proposed algorithm reacted to the acceleration and deceleration of the in-lane target much more quickly than did the base algorithms. When the in-lane target accelerated at t = 0 to 2 s, CV/Path and CTRV showed more conservative acceleration than did the other two algorithms. However, V_{flow}/Path determined acceleration to be risky when the in-lane target accelerated at t = 2 to 3 s because V_{flow}/Path predicts the longitudinal behavior of targets based on the



(a) Longitudinal acceleration history

(b) Velocity history

(c) Clearance history

(d) Comparison of prediction results at 5 s

(e) Dash cam image at 5 s

FIGURE 12. Comparison of motion planning results for inner lane case.

average traffic flow speed. In other words, if a small number of vehicles change their velocity to the opposite direction of traffic flow, the prediction accuracy of these vehicles decreases. However, in these two cases, the proposed algorithm showed more appropriate reactions than those from the three base algorithms, and this improvement also appeared at t = 8 to 11 s when the in-lane target reaccelerated after braking.

Other results of the turning scenario in the outer lane are depicted in Fig. 13. The preceding vehicle in the outer lane prior to entering the intersection was turning inward because the guide line is not drawn for the outer lane. The prediction results are depicted in Fig. 13 (d), with the corresponding dashboard camera capture in Fig. 13 (e). As shown in Fig. 13 (d), the preceding vehicle deviated from the path defined on the map, which was judged to be the inner lane target by the CV/Path model. This misjudgment of in-lane target also occurred by other base algorithms, V$_{\text{flow}}$/Path and CTRV model. Based on the prediction results, the proposed algorithm determined the precise in-lane target decision with human-like acceleration commands.

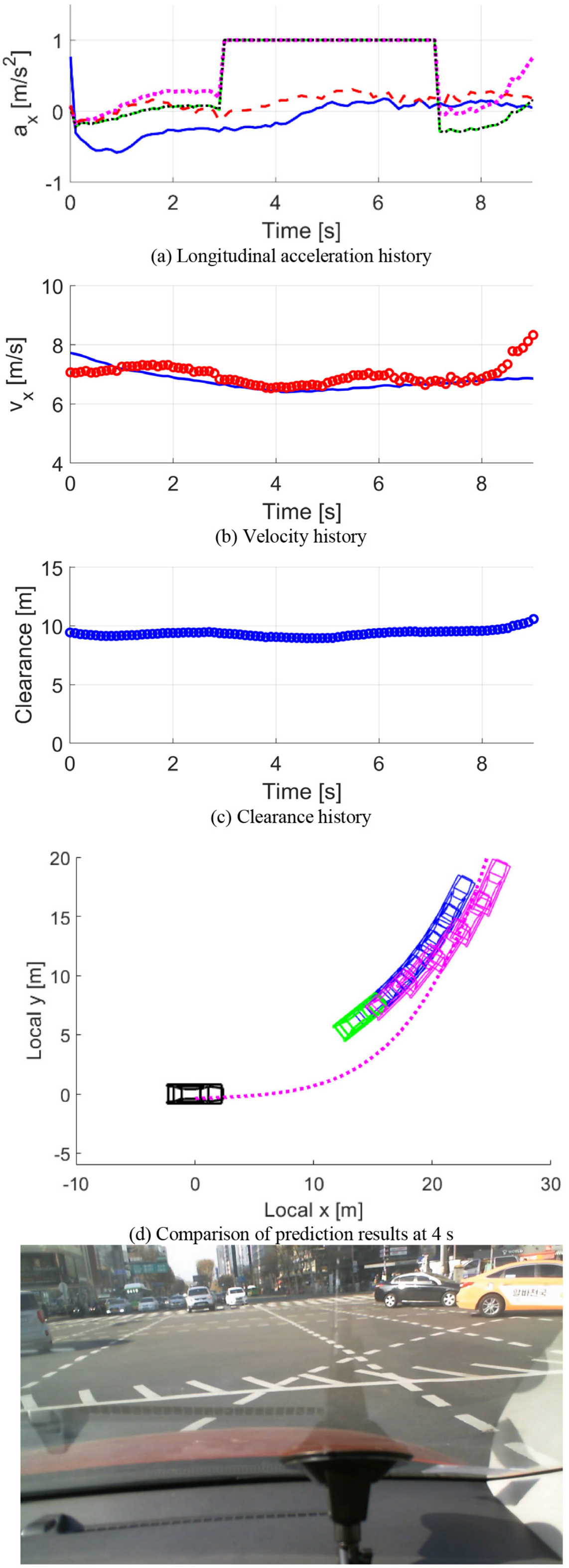
2) STATISTICAL ANALYSIS OF MOTION PLANNING APPLICATION RESULTS

As mentioned in Section VI, 484 data samples for the AV were collected by traveling in multi-lane turn intersections in real traffic. We analyzed these data from two perspectives: (1) the time to recognize the in-lane target; and (2) the similarity to human driver commands.

Among the 484 pieces of data, 252 cases exhibited irregular behavior, which made it difficult to recognize the in-lane target.

For example, some vehicles turned toward or away from the guidelines inside of intersections and lane changes at intersections. We compared the difference in recognition timing between the proposed and CV/Path motion predictor because the CV/Path and V$_{\text{flow}}$/Path algorithms use the same assumptions for lateral motion prediction as those of the path-following model, meaning that the two base algorithms showed nearly the same recognition timing results. Fig. 14 clearly presents the results for comparing the proposed and CV/Path algorithms. The figure shows that, in 135 of 252 cases, the AV recognized the in-lane target earlier than the base algorithm by up to 5 s. In 91 cases, the proposed and base algorithms showed the same recognition timing; however, in 26 cases, the AV classified the surrounding target vehicle as the in-lane target later by up to 1.2 s. This is what occurs when predictions are made before the newly detected target within the intersection has completed the input sequence observation length. Therefore, the proposed algorithm only recognized cases later than the base algorithm did when the surrounding target vehicles first appeared beyond the sensors' region of interest boundaries. This means that these cases took place sufficiently beyond the safety distance, and had little influence on determining the behavior of the subject vehicle.

The similarities between the results from the proposed algorithm and human driving decisions were analyzed by comparing the longitudinal acceleration command and



(a) Longitudinal acceleration history

(b) Velocity history

(c) Clearance history

(d) Comparison of prediction results at 4 s

(e) Dash cam image at 4 s

FIGURE 13. Comparison of motion planning results for outer lane case.

human driving history. Acceleration error was defined as follows:

$$a_{x,error} = a_{x,human} - a_{x,cmd} \tag{20}$$

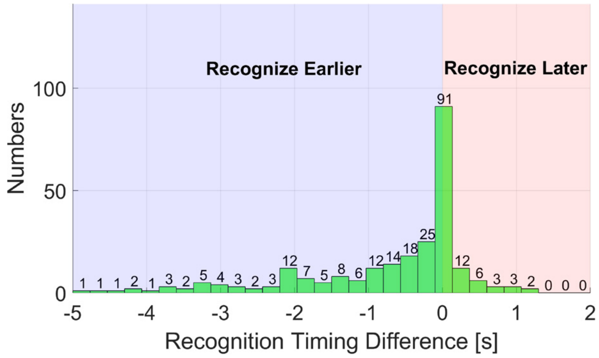


FIGURE 14. In-lane target recognition timing differences between the proposed and base algorithms.

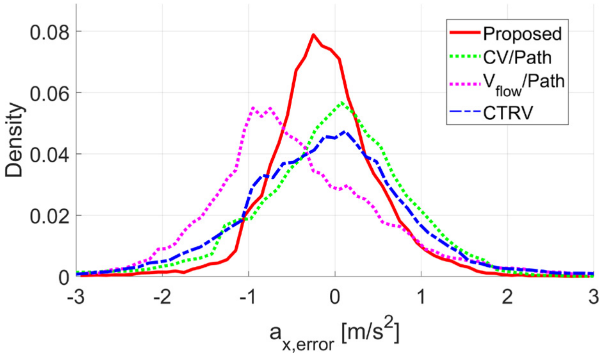


FIGURE 15. Longitudinal acceleration error between human driver and proposed base algorithms.

where $a_{x,human}$ and $a_{x,cmd}$ are the human driver's acceleration history and the command from the proposed algorithm, respectively. In this study, $a_{x,cmd}$ of the proposed and base algorithms were used to evaluate the $a_{x,error}$, the results of which are presented in Fig. 15. As shown in the figure, the proposed algorithm showed more similar results to human drivers' decisions than did the base algorithms. Two-thirds of $a_{x,error}$, 67.36%, was distributed in the $\pm 0.5 \mathrm{m/s}^2$ region, and 91.97% was in the $1.0 \mathrm{m/s}^2$ region. Meanwhile, the $a_{x,error}$ distributions for the CV/Path and CTRV models were wider than the distribution for the proposed algorithm. In particular, the CV/Path and CTRV models could not predict the target's acceleration motion, which broadens the distribution in both directions. However, $a_{x,error}$ for $\mathrm{V}_{\mathrm{flow}}$/Path was biased on the negative plane because the $\mathrm{V}_{\mathrm{flow}}$/Path model's predictions of the target motion were under the assumption of following the traffic flow. In other words, this model possesses limited ability to respond to different in-lane target behaviors in traffic flow. In particular, the $a_{x,error}$ distribution of $\mathrm{V}_{\mathrm{flow}}$/Path was biased in the negative plane because the magnitude of deceleration is generally larger than that of acceleration in normal driving conditions. In short, the proposed algorithm responded to the surrounding target vehicles in similar ways to human drivers when the AV traveled in multi-lane turn intersections.

VI. CONCLUSION

A surrounding vehicle motion predictor based on a Long Short Term Memory (LSTM)-Recurrent Neural Network (RNN) at multi-lane turn intersections was developed, and its application in an autonomous vehicle was evaluated. The LSTM-RNN based motion predictor was utilized to predict irregular behaviors of surrounding vehicles based on motion history. The proposed network was trained using 11,662 data samples collected by on-vehicle sensors on an AV driving in real traffic. These data were processed with an encoder and a decoder to standardize each component of the input data and to scale back to real-world units using the same encoder parameters. The motion planner based on the predicted states of surrounding targets was designed using Model Predictive Control (MPC). The proposed algorithm was evaluated based on 4,998 data samples, which were not used to train the network. Then, the results from the proposed algorithm were compared with three base algorithms: (1) models for path-following with constant velocity; (2) path-following with traffic flow; and (3) constant turn rate and velocity.

The evaluation results showed precise prediction accuracy, which confirmed the safety of the autonomous vehicle. In addition, when the proposed motion predictor was applied to a motion planner, the time to recognize in-lane targets within the intersection improved significantly over the performance of the base algorithms. Furthermore, the proposed algorithm was compared with human driving data, and it showed similar longitudinal acceleration. The proposed motion predictor can be applied to path planners when AVs travel in unconstructed environments, such as multi-lane turn intersections. We expect that a motion planner with an integrated motion predictor can achieve passenger safety and acceptance of autonomous vehicles. Future works in motion prediction of surrounding vehicles can be summarized in four aspects. The first is developing trajectory prediction algorithms using other machine learning algorithms, such as attention-aware neural networks. The second is applying the machine learning-based approach to infer lane change intention at motorways and main roads of urban environments. The third is extending the target road of the trajectory predictor, such as roundabouts or uncontrolled intersections, to infer yield intention. The final research direction constitutes learning the behavior of surrounding vehicles in real time while automated vehicles drive with real traffic. Exploration of these topics is expected to substantially increase the safety and acceptance of autonomous vehicles to traffic participants on urban roads.

REFERENCES

[1] E. Choi, *Crash Factors in Intersection-Related Crashes: An On-Scene Perspective (No. Dot HS 811 366)*, U.S. DOT Nat. Highway Traffic Safety Admin., Washington, DC, USA, 2010.

[2] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in *Proc. IEEE 5th Int. Conf. Intell. Comput. Commun. Process.*, Cluj-Napoca, Romania, 2009, pp. 417–422.

[3] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. J. Dietmayer, "IMM object tracking for high dynamic driving maneuvers," in *Proc. IEEE Intell. Veh. Symp.*, Parma, Italy, 2004, pp. 825–830.

[4] A. Polychronopoulos, M. Tsogas, A. J. Amditis, and L. Andreone, "Sensor fusion for predicting vehicles' path for collision avoidance systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 549–562, Sep. 2007.

[5] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?" in *Proc. IEEE Intell. Veh. Symp.*, Eindhoven, The Netherlands, 2008, pp. 1068–1073.

[6] P. Lytrivis, G. Thomaidis, and A. Amditis, "Cooperative path prediction in vehicular environments," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Beijing, China, 2008, pp. 803–808.

[7] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Seoul, South Korea, 2015, pp. 1094–1099.

[8] J. Huang and H.-S. Tan, "Vehicle future trajectory prediction with a DGPS/INS-based positioning system," in *Proc. Amer. Control Conf.*, Minneapolis, MN, USA, 2006, pp. 5831–5836.

[9] S. Hoermann, D. Stumper, and K. Dietmayer, "Probabilistic long-term prediction for autonomous vehicles," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Los Angeles, CA, USA, 2017, pp. 237–243.

[10] M. Liebner, M. Baumann, F. Klanner, and C. Stiller, "Driver intent inference at urban intersections using the intelligent driver model," in *Proc. IEEE Intell. Veh. Symp.*, Alcala de Henares, Spain, 2012, pp. 1162–1167.

[11] B. Kim and K. Yi, "Probabilistic and holistic prediction of vehicle states using sensor fusion for application to integrated vehicle safety systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2178–2190, Oct. 2014.

[12] A. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo road safety reasoning," in *Proc. IEEE Intell. Veh. Symp.*, Las Vegas, NV, USA, 2005, pp. 319–324.

[13] M. Althoff and A. Mergel, "Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1237–1247, Dec. 2011.

[14] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with Gaussian mixture models," in *Proc. IEEE Intell. Veh. Symp.*, Alcala de Henares, Spain, 2012, pp. 141–146.

[15] R. Huang, H. Liang, P. Zhao, B. Yu, and X. Geng, "Intent-estimation- and motion-model-based collision avoidance method for autonomous vehicles in urban environments," *Appl. Sci.*, vol. 7, no. 5, p. 457, Apr. 2017.

[16] M. G. Ortiz, J. Fritsch, F. Kummert, and A. Gepperth, "Behavior prediction at multiple time-scales in inner-city scenarios," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Baden-Baden, Germany, 2011, pp. 1068–1073.

[17] S. Klingelschmitt, M. Platho, H.-M. Groß, V. Willert, and J. Eggert, "Combining behavior and situation information for reliably estimating multiple intentions," in *Proc. IEEE Intell. Veh. Symp.*, Dearborn, MI, USA, 2014, pp. 388–393.

[18] T. Gindele, S. Brechtel, and R. Dillmann, "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Funchal, Portugal, 2010, pp. 1625–1631.

[19] M. Schreier, V. Willert, and J. Adamy, "Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, 2014, pp. 334–341.

[20] M. Althoff, O. Stursberg, and M. Buss, "Safety assessment of driving behavior in multi-lane traffic for autonomous vehicles," in *Proc. IEEE Intell. Veh. Symp.*, Xi'an, China, 2009, pp. 893–900.

[21] F. Gross, J. Jordan, F. Weninger, F. Klanner, and B. Schuller, "Route and stopping intent prediction at intersections from car fleet data," *IEEE Trans. Intell. Veh.*, vol. 1, no. 2, pp. 177–186, Jun. 2016.

[22] P. Liu, A. Kurt, and Ü. Özgüner, "Trajectory prediction of a lane changing vehicle based on driver behavior estimation and classification," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, 2014, pp. 942–947.

[23] T. Streubel and K. H. Hoffmann, "Prediction of driver intended path at intersections," in *Proc. IEEE Intell. Veh. Symp.*, Dearborn, MI, USA, 2014, pp. 134–139.

[24] D. J. Phillips, T. A. Wheeler, and M. J. Kochenderfer, "Generalizable intention prediction of human drivers at intersections," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Los Angeles, CA, USA, 2017, pp. 1665–1670.

[25] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Yokohama, Japan, 2017, pp. 399–404.

[26] A. Khosroshahi, E. Ohn-Bar, and M. M. Trivedi, "Surround vehicles trajectory analysis with recurrent neural networks," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Rio de Janeiro, Brazil, 2016, pp. 2267–2272.

[27] W. Liu, S.-W. Kim, K. Marczuk, and M. H. Ang, "Vehicle motion intention reasoning using cooperative perception on urban road," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, 2014, pp. 424–430.

[28] L. He, C.-F. Zong, and C. Wang. "Driving intention recognition and behaviour prediction based on a double-layer hidden Markov model," *J. Zhejiang Univ. Sci. C*, vol. 13, no. 3 pp. 208–217, 2012.

[29] K. Driggs-Campbell, V. Govindarajan, and R. Bajcsy, "Integrating intuitive driver models in autonomous planning for interactive maneuvers," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3461–3472, Dec. 2017.

[30] A. Lawitzky, D. Althoff, C. F. Passenberg, G. Tanzmeister, D. Wollherr, and M. Buss, "Interactive scene prediction for automotive applications," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Gold Coast, QLD, Australia, 2013, pp. 1028–1033.

[31] F. Damerow, S. Klingelschmitt, and J. Eggert, "Spatio-temporal trajectory similarity and its application to predicting lack of interaction in traffic situations," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Rio de Janeiro, Brazil, 2016, pp. 2512–2519.

[32] D. Orth, D. Kolossa, and M. Heckmann, "Predicting driver left-turn behavior from few training samples using a maximum a posteriori method," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Yokohama, Japan, 2017, pp. 1–6.

[33] A. Sarkar, K. Czarnecki, M. Angus, C. Li, and S. Waslander, "Trajectory prediction of traffic agents at urban intersections through learned interactions," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, 2017, pp. 1–8.

[34] A. Zyner, S. Worrall, and E. Nebot, "A recurrent neural network solution for predicting driver intention at unsignalized intersections," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1759–1764, Jul. 2018.

[35] A. Zyner, S. Worrall, J. Ward, and E. Nebot, "Long short term memory for driver intent prediction," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Los Angeles, CA, USA, 2017, pp. 1484–1489.

[36] E. Strigel, D. Meissner, F. Seeliger, B. Wilking, and K. Dietmayer, "The Ko-PER intersection laserscanner and video dataset," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, 2014, pp. 1900–1901.

[37] *Ibeo Automotive Systems GmBH*. Accessed: May 18, 2016. [Online]. Available: www.ibeo-as.de

[38] H. Lee, S. Kim, S. Park, Y. Jeong, H. Lee, and K. Yi, "AVM / LiDAR sensor based lane marking detection method for automated driving on complex urban roads," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Los Angeles, CA, USA, 2017, pp. 1434–1439.

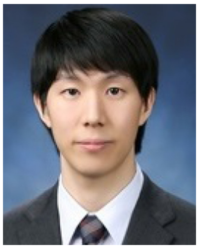
[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[41] J. Suh, H. Chae, and K. Yi, "Stochastic model-predictive control for lane change decision of automated driving vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 4771–4782, Jun. 2018.

[42] A. Barth and U. Franke, "Tracking oncoming and turning vehicles at intersections," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Funchal, Portugal, 2010, pp. 861–868.

YONGHWAN JEONG received the B.S. degree in mechanical engineering from Seoul National University, Seoul, South Korea, in 2014, where he is currently pursuing the Ph.D. degree in mechanical and aerospace engineering. His research interests include sensor fusion, risk assessment, automated vehicle motion planning in urban roads, automated vehicle control, intersection assistance systems, driver intention inference, and state prediction.



SEONWOOK KIM received the B.S. and M.S. degrees in mechanical engineering from Seoul National University, Seoul, South Korea, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in mechanical and aerospace engineering. His research interests include tracking moving objects, and managing object information for automated driving vehicles via fusion of sensors, such as LiDAR, radar, and vision.

KYONGSU YI (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from Seoul National University, Seoul, South Korea, in 1985 and 1987, respectively, and the Ph.D. degree in mechanical engineering from the University of California, Berkeley, in 1992. He is a Professor with the School of Mechanical and Aerospace Engineering, Seoul National University. His research interests include control systems, driver assistant systems, active safety systems, and automated driving of ground vehicles. He currently serves as a member of the editorial board of *Mechatronics* and the Chair of KSME IT Fusion Technology Division.