IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 September 2023, accepted 26 October 2023, date of publication 30 October 2023, date of current version 6 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3328628

RESEARCH ARTICLE

A Self-Driving Decision Making With Reachable Path Analysis and Interaction-Aware Speed Profiling

YUHO SONG[1], SANGWON HAN[2], AND KUNSOO HUH[1], (Member, IEEE)
[1]Department of Automotive Engineering, Hanyang University, Seoul 04763, South Korea
[2]Department of Automotive Engineering (Automotive-Computer Convergence), Hanyang University, Seoul 04763, South Korea

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

This work was supported by the Technology Innovation Program (Development on Automated Driving with Perceptual Prediction Based on T-Car/Vehicle Parts to Intelligent Control/System Integration For Assessment) funded by the Ministry of Trade, Industry & Energy (MOTIE), South Korea, under Grant 20018101.

ABSTRACT This paper proposes a behavior planning algorithm for self-driving vehicles to handle lane keeping, speed control considering inter-vehicle space, and collision avoidance under uncertainty. The behavior planning approach is structured as a hierarchically organized Markov Decision Process (MDP) comprising two components: the path planning MDP and the speed profiling MDP. The path planning MDP generates multiple path candidates using lane-change path data collected from human drivers. Evaluation of each path candidate is based on reward and penalty terms. The path planning MDP spans and updates considering the predicted simulation time of the path candidate. Subsequently, the speed profiling MDP determines the optimal sequence of speeds for the host vehicle on the planned path. Evaluation of the path and speed profile utility is performed using reward and penalty terms based on the current states of vehicles and road structure. A unique aspect of this approach is the incorporation of uncertainty-aware collision risk and interaction-aware gap penalty, which account for the uncertainty of perception and traffic motion. Various cut-in scenarios are presented in simulations to demonstrate the effectiveness of the proposed algorithm.

INDEX TERMS Autonomous vehicle, traffic model, decision-making algorithm, path planning, speed profiling, Markov decision process (MDP), self-driving formulation.

I. INTRODUCTION

The advancement in the development of the decision-making algorithms for self-driving cars has led to several companies initiating testing programs for vehicles with SAE Level 4 autonomous car. For instance, in August 2021, Toyota introduced the Level 4 self-driving service through its multi-functional vehicle, the e-Palette. Mercedes Benz, on the other hand, commercialized its Level 4 automated valet parking system through the second generation of its infotainment system, Mercedes-Benz User Experience (MBUX). Other major automobile manufacturers such as Honda and GM have also introduced their Level 4 self-driving services. Despite the progress made in this field, the incidence of accidents involving self-driving cars has risen considerably, as reported [1]. Thus, it is imperative to continue to develop and improve self-driving technology to ensure its safety and reliability [2].

The self-driving systems can encounter failures due to uncertainty present in the traffic scenario, such as unusual behavior of neighboring vehicles and perception errors. These uncertainties can result in incorrect predictions that can cause vehicle crashes or discomfort to passengers. As a result, it is crucial to consider uncertainty in the decision-making process of self-driving vehicles. One of the commonly acknowledged sources of uncertainty is perception error caused by sensor delay or noise, which can induce misjudgment in the decision-making process [3], [4], [5].

The associate editor coordinating the review of this manuscript and approving it for publication was P. Venkata Krishna.

Another factor contributing to the failure of self-driving systems is the unknown intention of surrounding vehicles, such as malicious cut-ins or unexpected interactions with other vehicles. These unknown intentions are linked with uncertainty in the decision-making algorithm of the self-driving car. If the motion of surrounding vehicles is incorrectly predicted, the self-driving vehicle can be put in a dangerous situation. To mitigate the risk posed by unknown intentions, it may be necessary for the self-driving decision to emulate the expected behavior of a human driver.

Path planning plays a crucial role in achieving human-like driving motion, as it is challenging to establish a standard trajectory for vehicles. This is due to the variability in driving styles and situational factors, such as a sudden lane change to avoid an imminent collision. However, the use of a large volume of real-traffic data can provide a nearly standardized vehicle path.

In this study, a novel approach to behavior planning that combines path-planning and speed profiling through the use of Markov Decision Process (MDP) is presented. To consider the behavior of the human-driven vehicle, the interaction associated with the speed and inter-vehicle gap is analyzed and designed into a Markov vehicle interaction model. Path and acceleration candidates are sampled from real-traffic data and the reachability of these candidates is evaluated based on the vehicle's dynamics and kinematics. The MDP tree for path-planning is designed to incorporate the expected driving time of each path segment as an action, and the utility of each MDP node is evaluated based on the different time periods of each path candidate. The uncertainty in traffic motion is also analyzed, and a collision risk assessment for the perturbed vehicle position is considered to account for the unpredictability in traffic conditions.

To ensure collision avoidance in uncertain conditions, the proposed algorithm dynamically determines the optimal path and speed for the host vehicle based on real-time observations of surrounding vehicles and the road structure. The specific values for path and speed profile are adjusted to maintain a safe distance from other vehicles and ensure stability during maneuvers. Additionally, the proposed approach considers various sources of uncertainty, including uncertain perception of surrounding vehicles, partially predictable driver behavior, and sensor measurement noise, to make proactive decisions and mitigate collision risks. The main contributions of this paper can be summarized as follows:

· The detection and trajectory prediction of cut-in vehicles are investigated based on real traffic data.

· A method is proposed for evaluating the utility of a tree node in the MDP, where the action is represented by a path segment.

· The reachability of a path unit is determined through consideration of the vehicle kinematics and mechanical constraints of the steering unit.

· A risk assessment algorithm is proposed considering the model uncertainty and vehicle-to-vehicle interaction.

II. RELATED WORK

In recent literature, numerous research efforts have been dedicated toward the development of decision-making algorithms for autonomous vehicles. This paper categorizes these algorithms into three distinct groups: rule-based, graph-search, and sampling-based methods.

The traffic rule serves as a crucial reference for the development of autonomous driving behavior. A rule-based approach is a common means of implementing these traffic rules in autonomous vehicles, with a finite state machine (FSM) being a widely employed method [6], [7], [8]. FSM involves defining the states associated with the target autonomous driving service and determining state transitions based on a flow chart. Despite its ease of implementation, FSM is not highly adaptable to varying road conditions and thus can be prone to errors.

To facilitate optimal decision-making in autonomous driving, graph-search and Markov Decision Process (MDP) are commonly employed approaches. The graph-search method, which aims to derive the optimal sequence of vehicle positions, is further classified into state lattice [9], elastic band [10], and A-star algorithm [11]. These methods result in a sequence of optimally selected positions on the road surface. McNaughton et al. [12] proposed a motion planning approach for autonomous driving using a conformal spatiotemporal lattice, which enables optimal trajectory planning for complex driving scenarios. Gu et al. [13] developed an automated tactical maneuver discovery, reasoning, and trajectory planning system for autonomous driving, which enhances decision-making and trajectory planning capabilities. Li et al. [14] described a real-time trajectory planning framework for autonomous urban driving, incorporating advanced algorithms and verification techniques to ensure safe and efficient navigation. Fernandes et al. [15] presented the CaRINA Intelligent Robotic Car, a comprehensive architectural design for an autonomous vehicle, addressing various aspects of autonomous driving systems. Hegedűs et al. [16]. suggest synthesis of the clustering-based, graph-based and dynamic-based methods for overtaking strategies.

On the other hand, MDP optimizes the action sequence considering uncertainty in traffic information, such as vehicle-to-vehicle interactions and perception errors. Lenz et al. [17] proposed a tactical cooperative planning approach for autonomous highway driving using Monte-Carlo Tree Search and uncertainty-aware vehicle prediction. Hubmann et al. [18] introduced a decision-making method for autonomous driving considering interaction and uncertain prediction of surrounding vehicles, utilizing a Partially Observable MDP (POMDP) maneuver planner. Hubmann et al. [19] further developed a POMDP maneuver planner to address occlusions in urban scenarios, enhancing the decision-making capabilities in challenging driving conditions. Avilés et al. [20] utilize Probabilistic Logic Markov Decision Processes to model driving behaviors in self-driving car. This approach integrates probabilistic reasoning to

address uncertainties and enhance the accuracy of capturing complex interactions and behaviors in uncertain environments.

To expedite rapid yet sub-optimal decision-making in self-driving, a sampling-based approach can be employed. The most prominent method in this category is the Rapidly-exploring Random Tree (RRT) algorithm [21]. While the original RRT algorithm samples target position candidates randomly, various modifications to the sampling method have been introduced to generate more suitable path candidates for autonomous driving. LaValle et al. [21] introduced the RRT algorithm as a novel tool for path planning, enabling efficient exploration of the state space and generating feasible trajectories for autonomous vehicles. Ma et al. [22] proposed an efficient sampling-based motion planning method tailored specifically for on-road autonomous driving, enhancing the decision-making capabilities and trajectory generation efficiency. Mashayekhi et al. [23] presented the Informed RRT*-Connect, an asymptotically optimal single-query path planning method, which significantly improves the quality and optimality of generated paths. Spanogiannopoulos et al. [24] developed a sampling-based non-holonomic path generation approach designed for self-driving cars, enabling effective trajectory planning in complex driving scenarios.

To ensure collision-free decision-making in autonomous vehicles, a crucial step involves the assessment of collision risk. The methodology for assessing collision risk varies depending on the algorithms used in conjunction. However, a common aspect is considering the position and shape of vehicles. First, the shape of the vehicle is often approximated as circular or elliptical. In this case, the algorithm checks whether two circles collide along their predicted paths [25], [26], [27], [28]. Alternatively, the vehicle's actual shape may be approximated as a rectangle that closely resembles its real configuration [29], [30]. The circular-based approach may exhibit false positive collision risks in scenarios where there is a vehicle passing adjacent to the subject vehicle in the lateral direction, leading to unnecessary risk detection. In cases where vehicles are approximated as rectangles, there may be insufficient margin for error due to the limited free space. Consequently, if there are inaccuracies in the perception system, it could lead to potential risks.

III. CUT-IN VEHICLE MOTION MODEL

The prediction of vehicle motion plays a crucial role in autonomous driving by reducing the area of interest to the most probable vehicle motion and subsequently reducing the search domain of the decision-making and control algorithms. Furthermore, the designed system can be validated using a vehicle model, allowing for the evaluation of potential risks through simulation. This chapter outlines a methodology for predicting the motion of surrounding vehicles, taking into account uncertainty and awareness of interactions.

Studies of the NHTSA [31] have shown that vehicle-to-vehicle collisions often occur during lane changes,

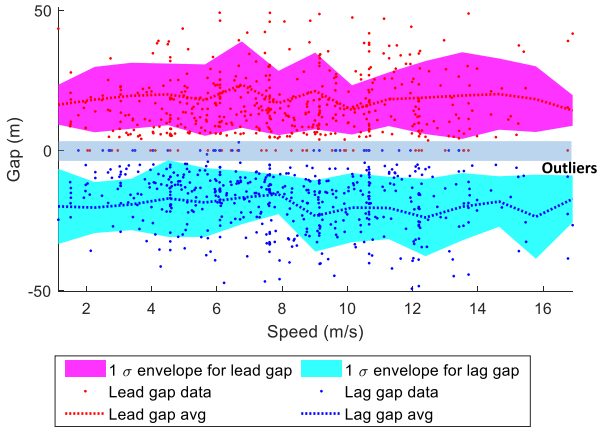

FIGURE 1. Lead/Lag gap analysis during the cut-in maneuvering.

highlighting the importance of motion prediction for lane-change vehicles. The prediction of the cut-in motion of a surrounding vehicle is dependent on the inter-vehicle space between the following vehicle and the leading vehicle. The longitudinal distance from the cut-in vehicle to the leading vehicle in the target lane is referred to as the lead gap, while the distance from the cut-in vehicle to the rear vehicle in the target lane is referred to as the lag gap [32]. Analysis of NGSIM data shows that the acceptance of cut-in intention is higher when the lag gap is greater than 10 meters (Figure 1), with no significant relationship observed between speed and gap acceptance. Consequently, if the inter-vehicle space between the host vehicle and the leading vehicle exceeds 20 meters, a vehicle in an adjacent lane may attempt a cut-in maneuver. To detect such cut-in vehicles, the cut-in candidate probability is defined as follows.

$$P(r_i|r_j, x_{ob}) = P(\Delta x_{lag} < \Delta x_{lag,avg} - \sigma_{x,lag}|r_j) \times P(\Delta x_{lead} > \Delta x_{lead,avg} + \sigma_{x,lead}|r_j) \quad (1)$$

with
$$P(\Delta x_{lag}) = \Phi_{\Delta x_{lag,avg},\sigma_{x,lag}}(\Delta x_{lag})$$
$$P(\Delta x_{lead}) = \Phi_{\Delta x_{lead,avg},\sigma_{x,lead}}(\Delta x_{lead})$$

where

r_i: i^{th} lane
r_j: j^{th} lane
x_{ob}: longitudinal position of the cut-in candidate
Δx_{lag}: lag gap (Distance from the cut-in candidate to the lag vehicle along route r_i)
$\sigma_{x,lag}$: variance of the lag gap
Δx_{lead}: lead gap (Distance from the cut-in candidate to the leading vehicle along route r_i)
$\sigma_{x,lead}$: variance of the lead gap

$\Phi_{\Delta x_{lag,avg},\sigma_{x,lag}}$ is cumulative distribution function of the lag gap and $\Phi_{\Delta x_{lead,avg},\sigma_{x,lead}}$ is cumulative distribution function of the lead gap. Then the cut-in motion is detected by the following cut-in motion detector.

$$P_{cut-in} = \mathrm{lgm}(q_{LC} \cdot \frac{v_d}{v_s} \cdot \Delta d_{\text{cut-in}}) \quad (2)$$

where

lgm: logistic sigmoid function used to model the cut-in probability
q_{LC}: stiffness term of the cut-in motion detector, which controls the sensitivity to lateral motion
v_s: longitudinal speed of the surrounding vehicle
v_d: lateral speed of the surrounding vehicle
$\Delta d_{\text{cut-in}}$: The lateral offset of the cut-in vehicle from the original lane

In this study, stiffness is a key parameter that controls how sensitive the reward and penalty terms are to variations in design variables. It influences the responsiveness of the reward and penalty terms to changes in the design variables.

In Equation (2), the probability of a cut-in event (P_{cut-in}) is determined using a logistic sigmoid function, which takes into account the stiffness term (q_{LC}) of the cut-in motion detector, the ratio of lateral speed to longitudinal speed ($\frac{v_d}{v_s}$) of the surrounding vehicle, and the lateral offset ($\Delta d_{\text{cut-in}}$) of the cut-in vehicle from the original lane. The logistic sigmoid function provides a smooth and bounded output, allowing for the modeling of the cut-in probability. The identification of the target lane is determined by the threshold value of P_{cut-in}. For instance, the lane that is closest to the vehicle in question is selected to be the target lane if P_{cut-in} value is greater than a threshold value of 0.7.

Once the cut-in motion of the surrounding vehicle is detected, it is necessary to define the lane change motion in order to predict the trajectory of the surrounding vehicle. The prediction of the longitudinal motion is initially described through the use of the constant velocity model, which is a straightforward algorithm and is suitable for short-term prediction based on the velocity and heading information of the dynamic object. The longitudinal motion of the surrounding vehicle is represented by the following equation derived from the constant velocity model.

$$s_{t+1} = s_t + dt \cdot v_{s,t} \tag{3}$$

where

s_t: longitudinal position at time step t
dt: time step in prediction
$v_{s,t}$: velocity at time step t

Then, the inter-vehicle space-keeping behavior can be modeled based on the adaptive cruise control model of [33].

$$a_{x,leadgap} = -\frac{1}{h}\left(-\Delta v_{x,lead} + \lambda(\Delta x_{lead,avg} + \epsilon - \Delta x_{lead})\right) \tag{4}$$

$$a_{x,laggap} = -\frac{1}{h}\left(-\Delta v_{x,lag} + \lambda(\Delta x_{lag,avg} + \epsilon - \Delta x_{lag})\right) \tag{5}$$

where

λ: time constant of space-keeping model
h: time gap parameter
Δv_x: relative velocity to the lead/lag vehicle
ϵ: uncertainty in the inter-vehicle space model

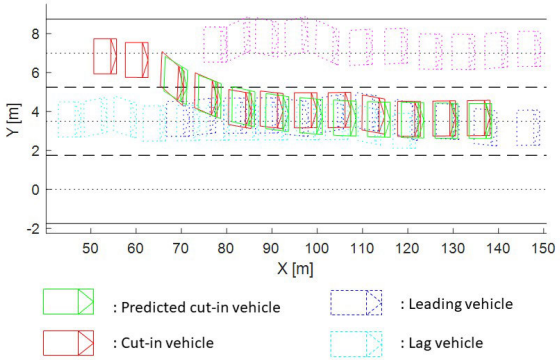

FIGURE 2. Detection and prediction of cut-in vehicle motion.

In order to approximately predict the cut-in trajectory of vehicles in the surrounding environment, it is necessary to model the motion that is correlated with road information. In this study, the lane-change motion is modeled by utilizing the 1^{st} order delay model as described below.

$$\tau_d \dot{d}_i + d_i = 0 \tag{6}$$

where

d_i: lateral distance from the i^{th} lane
τ_d: time constant for the lane-change motion

The obtained result of the cut-in motion is illustrated in Figure 2 where the predicted cut-in motion is compared to real-traffic data. The 1-σ uncertainty of the cut-in vehicle motion is calculated from Figure 1 and it is approximately 3 meters in the longitudinal direction and 0.2 meters in the lateral direction.

IV. SELF-DRIVING MDP

The self-driving MDP generates path candidates through sampling and uses these candidates to construct an MDP tree. The path candidates are generated by taking into account the characteristics of the driving data of human-driven vehicles. The speed profiling MDP is implemented using a discrete longitudinal acceleration space, and a method to mitigate the impact of chattering caused by the discrete action space is presented. Additionally, a self-driving formulation that incorporates human interaction is included.

A. REACHABLE PATH CANDIDATES

The driving path of autonomous vehicles should be both safe and comfortable, characterized by a traceable path with minimized jerk. To generate such paths, a path planning method that takes into account the dynamic characteristics of the vehicle is needed. While previous studies have proposed path planning methods that utilize smooth paths such as the Dubins curve [34] and the Reeds-Shepp curve [35], these methods have limitations for autonomous driving because they do not consider the mechanical constraints of the vehicle.

While tracking the planned path, the yaw rate of the vehicle changes continuously, leading to an actual driving path that cannot be described as a constant turning rate trajectory.

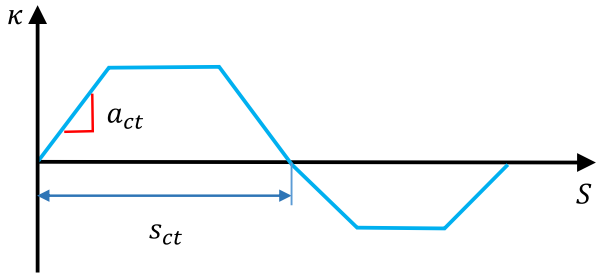

FIGURE 3. Curvature of a PLC curve.

To accurately describe the varying yaw rate, the reachable clothoid curve is implemented using a piece-wise linear clothoid (PLC) curve [36]. As illustrated in Figure 3, the PLC curve is parameterized as the slope of the side, a_{ct}, and the arc length, S_{ct}, to complete the path from the starting point to the target point. in particular, a trapezoidal-shaped PLC curve is frequently used for minimizing lateral acceleration when driving at a constant speed. To calculate the PLC curve based on the starting and target points, the clothoid curve must be transformed from the station-curvature coordinates to the Euclidean coordinates [37].

To ensure safe and comfortable driving, it is important to restrict the lateral acceleration within a controllable bound. If excessive acceleration is applied, the lateral force exerted on the tire is not linearly proportional to tire slip, leading to unpredictable vehicle motion. Thus, the curvature, κ, of the path candidates must be bounded as follows.

$$|\kappa| \leq \frac{a_{y,al}}{v_{ego}^2} \tag{7}$$

where v_{ego} is the longitudinal velocity of the host vehicle. $a_{y,al}$ is the lateral acceleration limit for autonomous driving and, for instance, it is selected as 80% of the maximum lateral acceleration allowed. Besides, by considering the maximum steering angle, the bound for the curvature can be expressed as follows.

$$|\kappa| \leq min\left(\frac{tan\delta_{f,max}}{L_{wb}}, \frac{a_{y,al}}{v_{ego}^2}\right) \tag{8}$$

where L_{wb}: wheelbase of the host vehicle
$\delta_{f,max}$: maximum steering angle

The curvature of a path candidate needs to consider the maximum slew rate of the steering angle, $\dot{\delta}_{f,max}$. When the vehicle is driven along the path such as a PLC curve, the steering angle varies linearly during the transition and the turning rate is determined by the slope of the PLC curvature, indicated by a_{ct} in Figure 3.

$$a_{ct} = \frac{d^2\psi}{ds^2} \cong \frac{1}{v_{ego}^2}\frac{d^2\psi}{dt^2} \leq \frac{a_x tan\delta_f + v_{ego} sec^2\delta_f \dot{\delta}_{f,max}}{v_{ego}^2 L_{wb}} \tag{9}$$

In the numerator of the above equation, the size of the first term is much smaller than that of the second term

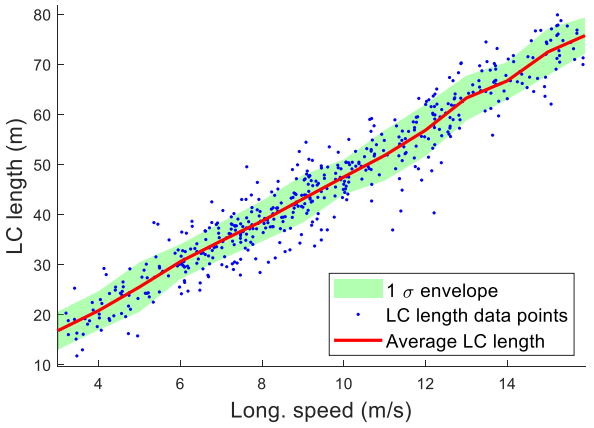

FIGURE 4. Lane change length of human drivers.

in autonomous driving. Besides, by considering the normal steering angle, the boundary of a_{ct} can be simplified as follows.

$$|a_{ct}| \leq \frac{\dot{\delta}_{f,max}}{v_{ego} L_{wb}} \tag{10}$$

Finally, real data collected from human drivers is considered in the selection of path candidates. For example, based on real-road lane-change data from the NGSIM-I80 in Figure 4, the relation between the lane change length and speed is utilized to design the lane change scenario such as the target point and arc length.

B. SELF-DRIVING BEHAVIOR PLANNING MDP

The state representation in the Markov Decision Process (MDP) framework consists of the current position and velocity of the surrounding vehicles. This study utilizes the methodology suggested by Kurniawati [38] for the implementation of the MDP model. The two-stage structure for autonomous vehicle behavior planning is adopted: path planning MDP and speed profiling MDP. The reasons for this approach are twofold: firstly, performing simultaneous path planning and speed profiling would require an excessive amount of computation time; and secondly, it is not advisable to frequently update the driving route, as sudden changes in the path can cause misunderstandings with other drivers and result in miscommunication. Therefore, if the path is planned through behavior planning, the autonomous vehicle should adhere to the pre-determined path.

For the path planning MDP, an autonomous vehicle is assumed to be driven on the PLC curve. The path-planning MDP tree is spanned with respect to PLC route candidates. For this approach, simulation time is variant according to the path candidates. Therefore, the path-planning MDP tree should be spanned considering this property. To implement asynchronously spanning path-planning MDP, a node contains the simulation step information. During the simulation, the reward for the PLC path is calculated considering the

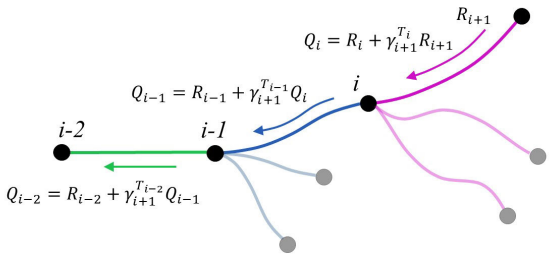

FIGURE 5. The extent of the tree structure in path-planning MDP.

discount factor, γ, as follows.

$$R_i = \sum_{t=1}^{T} \gamma^{t-1} r_t \tag{11}$$

where R_i integrated reward for PLC curve of the i^{th} MDP node
r_t: total reward at the time step t
T: total time step

The value of r_t is obtained by conducting a simulation of the driving scenario, wherein the cut-in model from the previous section is utilized to predict the cut-in and longitudinal motion of the surrounding vehicles. The value of r_t is ascertained through a straightforward summation of both the reward and penalty terms outlined in the subsequent section.

The utility of the PLC is evaluated optimally as the MDP tree is spanned. In order to consider the simulation steps of the PLC path, the Q-value of the parent node is updated as follows.

$$Q_{i-1} = R_{i-1} + \gamma^{T_{i-1}} max\left\{R_i | i \in I_{ch,i-1}\right\} \tag{12}$$

where $I_{ch,i-1}$ is the index set of the children nodes of the node $i-1$. Then, the Q-value of the ancestor nodes is updated as illustrated in Figure 5.

The optimal path is determined through the path-planning MDP, while the optimal speed is planned using the conventional MDP method that spans the MDP tree through the utilization of the state and action variables. In this study, the action variable is defined as the longitudinal acceleration, while the state is comprised of the global positions (X, Y, ψ), vehicle speed v_s and relevant target route information.

C. SELF-DRIVING FORMULATION

In the formulation of the reward for autonomous driving, the fundamental capabilities of self-driving vehicles are taken into consideration, including speed regulation and lane keeping. To account for the multiple-lane position, a reward function for multi-lane keeping is formulated.

$$r_{lane,i} = \begin{cases} W_{targ} exp(-q_{lane} d_i^2), \text{ if lane } i \text{ is target lane} \\ W_{lane} exp(-q_{lane} d_i^2), \text{ else} \end{cases} \tag{13}$$

where d_i: lateral distance from the i^{th} lane
q_{lane}: stiffness of the lane keeping reward
W_{targ}: reward for keeping the target lane i
W_{lane}: lane keeping reward for the non-target lane

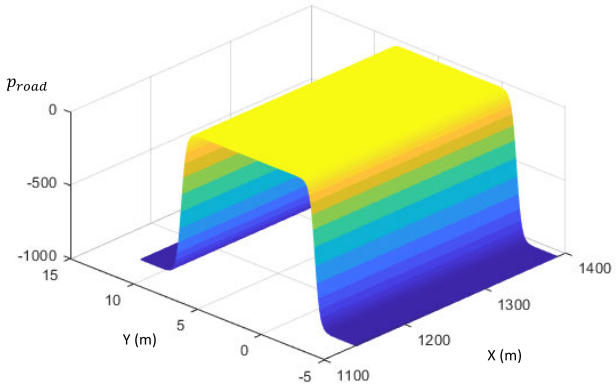

FIGURE 6. Illustration of the reward and penalty term of the road structure.

Additionally, the potential dangers posed by road structures such as guardrails and curbs should be incorporated into the self-driving problem to mitigate the risk of collisions. Based on the prior research [29], the road structure risk is formulated as follows.

$$p_{road} = W_{rb}\left(1 + \text{lgm}(q_{rb} d_{rb,left}) - \text{lgm}(q_{rb} d_{rb,right})\right) \tag{14}$$

where
q_{rb}: stiffness of the penalty for the road boundary
$d_{rb,left}$: lateral distance from the ego-vehicle to the left road boundary
$d_{rb,right}$: lateral distance from the ego-vehicle to right road boundary
W_{rb}: penalty value for the road boundary risk

In order to avoid the collision with the road boundary, W_{rb} is set to be much higher than that of the reward terms, e.g: $W_{rb} = 1000$.

Maintaining the desired speed while keeping the lane is a concurrent task that must be considered in the formulation of the reward function. To account for the simultaneous nature of these events, the synchronous characteristics of speed regulation and lane keeping are formulated as a reward term.

$$r_{speed} = exp\left(-q_v (v_{ego} - v_{targ})^2\right) \sum_{i=1}^{n_{lane}} r_{lane,i} \tag{15}$$

where q_v is the stiffness of the speed tracking reward. v_{targ} is the target speed while keeping the lane. The representation of the penalty associated with road structures is shown in Figure 6.

The requirement to suppress aggressive maneuvers necessitates the implementation of a penalty mechanism. The resulting expression for penalizing aggressive acceleration can be formulated as follows.

$$p_{as} = -W_{as} a_x^2 - W_{ad}\left(\kappa v_{ego}^2\right)^2 \tag{16}$$

where W_{as}: weight for the longitudinal acceleration
W_{ad}: weight for the lateral acceleration

Excessive lane-change behavior is also considered as an undesirable driving behavior. If this behavior is not appropriately penalized, the host vehicle is prone to making unnecessary lane changes, which can be particularly dangerous with multiple lane changes. In light of this, the following expression represents the penalty for lane-change behavior.

$$p_{LC} = -W_{LC}\left(\frac{d_{route,ego}}{w_{lane}}\right)^{10} \tag{17}$$

where $d_{route,ego}$: lateral distance from the present lane to the ego-vehicle
W_{LC}: weight for the lane-change penalty
w_{lane}: lane width

To incorporate a penalty for the possibility of collision with an object, the previously established simplified Minkowski algorithm [39] is modified to account for perception uncertainty. Positions and dimension of the vehicles for the Minkowski collision check algorithm are illustrated in Figure 7 where collision between two rectangles is detected by the following algorithm.

$$R_{col} = \begin{cases} R_{col} = 1, \text{ if } \{c_1 \cap c_2 \cap c_3 \cap c_4 \cap c_5 \cap c_6\} \\ 0, \text{ else} \end{cases} \tag{18}$$

where:

$$c_1 = \{(x, y, \psi)|x_1 \le x \le x_5\}$$
$$c_2 = \{(x, y, \psi)|cos\psi_{ob}(x - x_2) + sin\psi_{ob}(y - y_2) \ge 0\}$$
$$c_3 = \{(x, y, \psi)|cos\psi_{ob}(x - x_5) + sin\psi_{ob}(y - y_5) \le 0\}$$
$$c_4 = \{(x, y, \psi)|y_3 \le y \le y_6\}$$
$$c_5 = \{(x, y, \psi)| - sin\psi_{ob}(x - x_1) + cos\psi_{ob}(y - y_1) \le 0\}$$
$$c_6 = \{(x, y, \psi)| - sin\psi_{ob}(x - x_4) + cos\psi_{ob}(y - y_4) \ge 0\}$$
$$x_1 = x_2 = x_{ob} - \frac{L_{ob}}{2}cos\psi_{ob} - \frac{W_{ob}}{2}sin\psi_{ob} - \frac{L_{ego}}{2}$$
$$y_1 = y_2 + w_{ego} = y_{ob} - \frac{L_{ob}}{2}sin\psi_{ob} + \frac{W_{ob}}{2}cos\psi_{ob} + \frac{W_{ego}}{2}$$
$$x_3 = x_{ob} - \frac{L_{ob}}{2}cos\psi_{ob} + \frac{W_{ob}}{2}sin\psi_{ob} - \frac{L_{ego}}{2}$$
$$y_3 = y_{ob} - \frac{L_{ob}}{2}sin\psi_{ob} - \frac{W_{ob}}{2}cos\psi_{ob} - \frac{W_{ego}}{2}$$
$$x_4 = x_5 = x_{ob} + \frac{L_{ob}}{2}cos\psi_{ob} + \frac{W_{ob}}{2}sin\psi_{ob} + \frac{L_{ego}}{2}$$
$$y_4 = y_5 - w_{ego} = y_{ob} + \frac{L_{ob}}{2}sin\psi_{ob} - \frac{W_{ob}}{2}cos\psi_{ob} - \frac{W_{ego}}{2}$$
$$x_6 = x_{ob} + \frac{L_{ob}}{2}cos\psi_{ob} - \frac{W_{ob}}{2}sin\psi_{ob} + \frac{L_{ego}}{2}$$
$$y_6 = y_{ob} + \frac{L_{ob}}{2}sin\psi_{ob} + \frac{W_{ob}}{2}cos\psi_{ob} + \frac{W_{ego}}{2}$$

In this study, the uncertainty in the perception of the surrounding vehicles is modeled as Gaussian distributions, where the longitudinal error (Δs) is represented as $\Delta s \sim N(0, \sigma_s^2)$ and the lateral error (Δd) as $\Delta d \sim N(0, \sigma_d^2)$. Then, the penalty for the collision is expressed as follows.

$$p_{col} = W_{col}\psi\left(\Delta S \cdot Rot \cdot \Sigma_{s,d}^{-1} Rot^T \cdot \Delta S\right) \otimes R_{col} \tag{19}$$

$$\text{with } Rot = \begin{bmatrix} cos\psi_{ob} & -sin\psi_{ob} \\ sin\psi_{ob} & cos\psi_{ob} \end{bmatrix}, \Sigma_{s,d} = \begin{bmatrix} \sigma_s & 0 \\ 0 & \sigma_d \end{bmatrix}$$
$$\Delta S = [x_{ob} - x_{ego}, y_{ob} - y_{ego}]^T$$

where:

$x_{ego}, y_{ego}, \psi_{ego}$: longitudinal position, lateral position, heading angle of the host vehicle

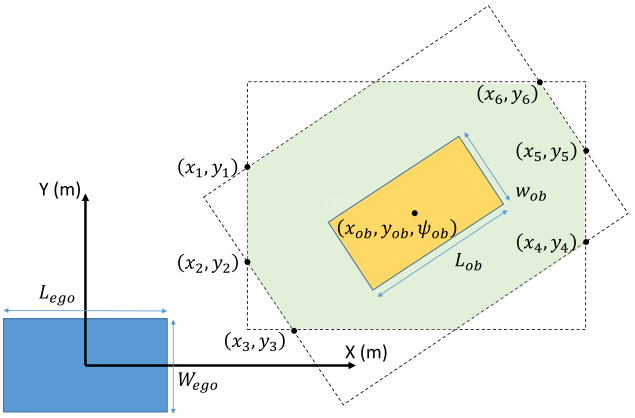

FIGURE 7. Geometry for Minkowski collision check algorithm.

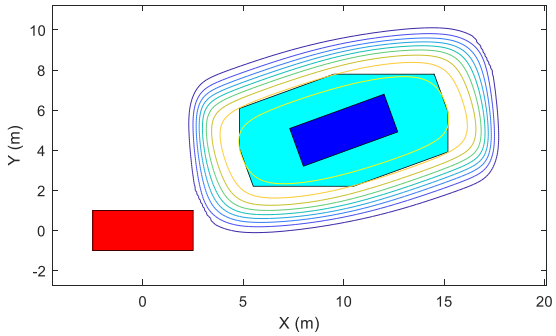

FIGURE 8. Two-dimensional contour plot of the collision risk.

$x_{ob}, y_{ob}, \psi_{ob}$: longitudinal position, lateral position, heading angle of the object vehicle
L_{ego}: overall length of the host vehicle
w_{ego}: overall width of the host vehicle
L_{ob}: overall length of the surrounding vehicle
w_{ob}: overall width of the surrounding vehicle
W_{col}: weight for the vehicle-to-vehicle collision risk
$\otimes$: convolution operator
σ_s: variance of the longitudinal traffic motion
σ_d: variance of the lateral traffic motion

Figure 8 shows an example of a collision penalty plot from equation (19). The reward and penalty formulations of self-driving vehicles are described in (13) ∼ (19). In addition, the interactive component must be considered such as maintaining a specified distance from the vehicle in front or behind. If the gap interaction is not incorporated into the self-driving MDP, the host vehicle may exhibit dangerous behavior such as tailgating. To discourage tailgating and maintain a certain distance such as in ACC (Adaptive Cruise Control), a gap penalty, p_{gap}, is formulated as follows.

$$p_{gap} = \begin{cases} -W_{targ}, \text{ if } \begin{array}{c} \Delta x_{lead} < \Delta x_{lead,avg} - \nu\sigma_{x,lead} \\ \text{or} \\ \Delta x_{lag} > \Delta x_{lag,avg} + \nu\sigma_{x,lag} \end{array} \\ 0, \text{ else} \end{cases} \tag{20}$$

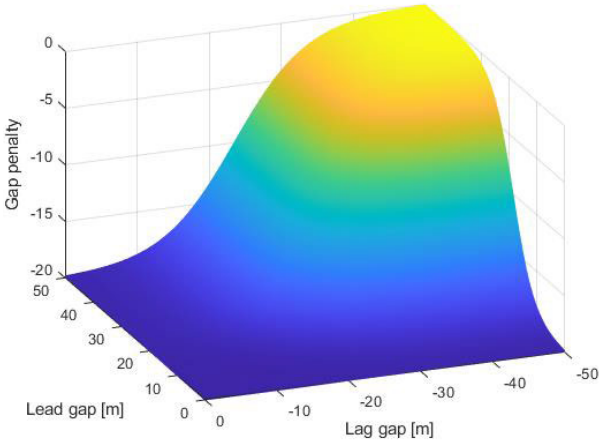

FIGURE 9. Gap penalty by Cumulative Distribution Function (CDF).

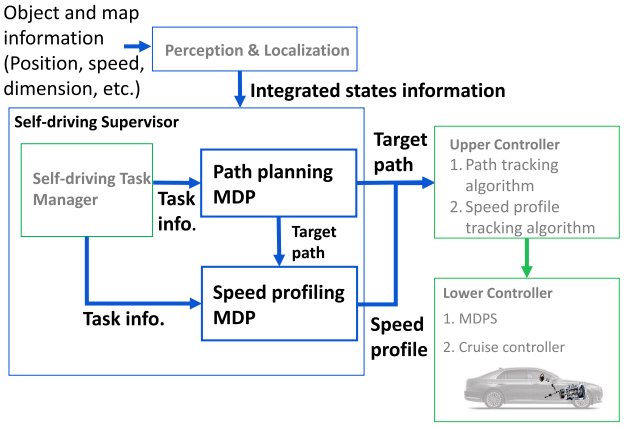

FIGURE 10. Architecture for the Self-driving behavior planning algorithm.

ν represents the gap-condition factor used to assess the degree of uncertainty associated with the inter-vehicle gap distance.

The magnitude of the gap penalty is determined by its ability to counteract the lane-keeping reward if the host vehicle drives too close to either the leading or trailing vehicle. This approach also prevents unintended lane changes in situations where there is insufficient space in the target lane. The gap condition can be represented by a probability model, such as a logistic sigmoid function or a cumulative distribution function (CDF). Figure 9 shows a visualization example of the CDF when the parameters of the gap penalty (mean and variance) are obtained based on the cut-in data in Figure 1. As previously indicated, the overall reward r_t specified in equation (11) is obtained through a straightforward summation of both the reward and penalty terms.

$$r_t = \sum_{i=1}^{n_{lane}} r_{lane,i,t} + r_{speed,t} + p_{road,t} + p_{as,t} + p_{LC,t} + p_{col,t} + p_{gap,t} \tag{21}$$

V. SYSTEM ARCHITECTURE

The hierarchically organized structure of Markov Decision Process (MDP) for self-driving is illustrated in Figure 10. It is assumed that the necessary information regarding the position, velocity, and map data of the host vehicle and its surrounding vehicles can be obtained through perception and localization techniques. Additionally, the self-driving task is assigned by the task manager, which provides information such as the target area, speed limit, and road type.

Although controller design for autonomous driving is beyond the scope covered in this paper, a basic controller was constructed to verify the proposed path planning algorithm. The upper controller has two functions: path tracking and speed control. The path tracking control utilizes a kinematic model-based pure pursuit algorithm [40] as depicted in Figure 11. The steering angle, δ_f, is calculated as

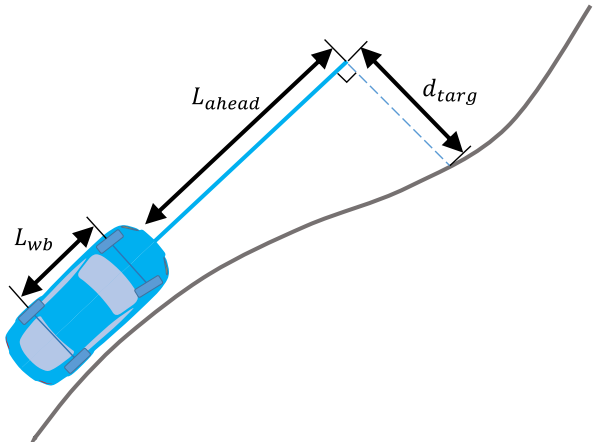

FIGURE 11. Geometry for the pure pursuit algorithm.

follows:

$$\delta_f = tan^{-1}\left(\frac{2L_{wb}d_{targ}}{L_{ahead}^2 + d_{targ}^2}\right) \tag{22}$$

where L_{wb}: wheelbase of the host vehicle
L_{ahead}: look-ahead distance
d_{targ}: lateral offset to the target path at the look-ahead distance

For the speed controller, the longitudinal dynamics of the cruise-controlled system are modeled as a first-order delay system and the target acceleration, $a_{x,targ}$, for the cruise controller is calculated as the control input:

$$a_{x,targ} = \frac{1}{\tau_{cc}}(-v_x + v_{x,targ}) \tag{23}$$

where τ_{cc}: time constant of the cruise-controlled system
$v_{x,targ}$: target speed from the speed-profiling MDP

The path planning and speed-profiling MDP are updated every 0.5 seconds in case of normal operation. To effectively

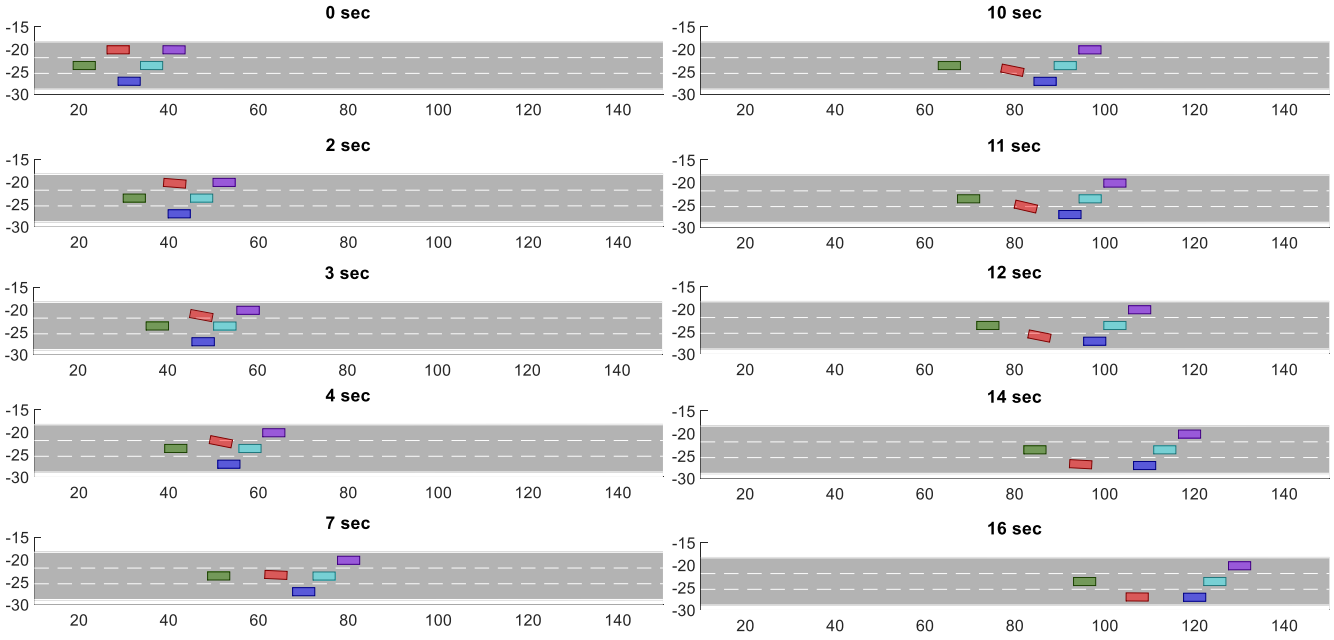

FIGURE 12. Scenario 1, Target area: 3rd lane, red: host vehicle, purple: leading vehicle in the 1st lane, Cyan: leading vehicle in the 2nd lane, green: lagged vehicle in the 2nd lane, red: leading vehicle in the 3rd lane.

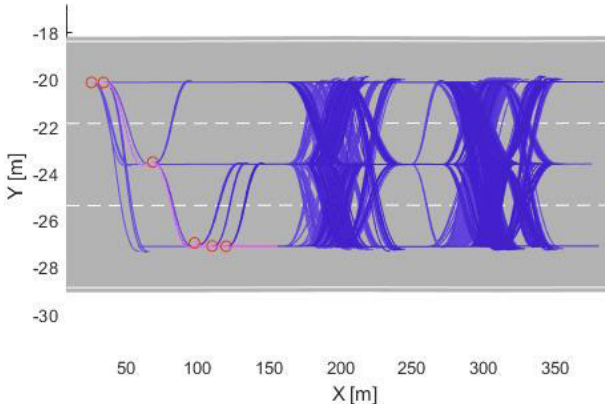

FIGURE 13. Trajectory planning results of scenario blue line: path candidates, magenta line: path-planning MDP.

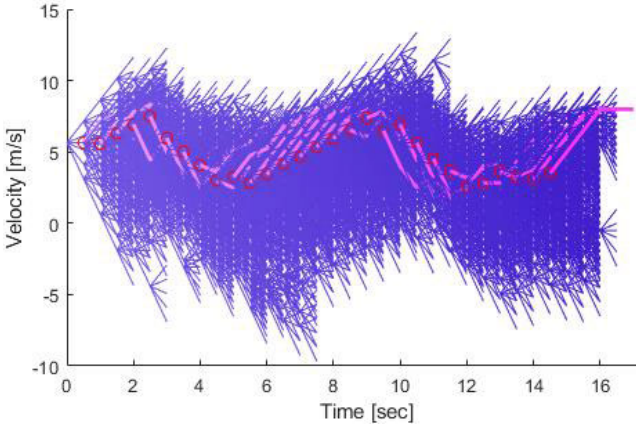

FIGURE 14. Speed planning results of scenario 1. blue line: speed candidates, magenta line: speed-profiling MDP.

address abrupt and substantial changes in driving circumstances, such as collision avoidance or sudden lane changes, the dynamic states and localization of vehicles are continuously updated at a frequency of 50 milliseconds. If the perceived state deviates from the predicted state beyond a predefined confidence boundary, the path-planning and speed profiling MDP are performed again promptly. The deviation is checked by the following equation.

$$q_{update} = \frac{1}{n_{veh}} \sum_{i=1}^{n_{veh}} \left(\begin{bmatrix} X_i \\ Y_i \end{bmatrix}_p - \begin{bmatrix} X_i \\ Y_i \end{bmatrix}_m \right)^T Q_u \left(\begin{bmatrix} X_i \\ Y_i \end{bmatrix}_p - \begin{bmatrix} X_i \\ Y_i \end{bmatrix}_m \right) \tag{24}$$

where

q_{update}: update cost, which quantifies the discrepancy between the predicted and measured state vectors for all vehicles (i)

$[X_i, Y_i]_p^T$: predicted state vector for vehicle i, representing its position (X_i, Y_i) based on the current estimation

$[X_i, Y_i]_m^T$: measured state vector for vehicle i, representing its actual position (X_i, Y_i) obtained from sensor measurements.

Q_u: weighting matrix, which determines the relative importance of each state variable (X, Y) in the cost function. This matrix controls the influence

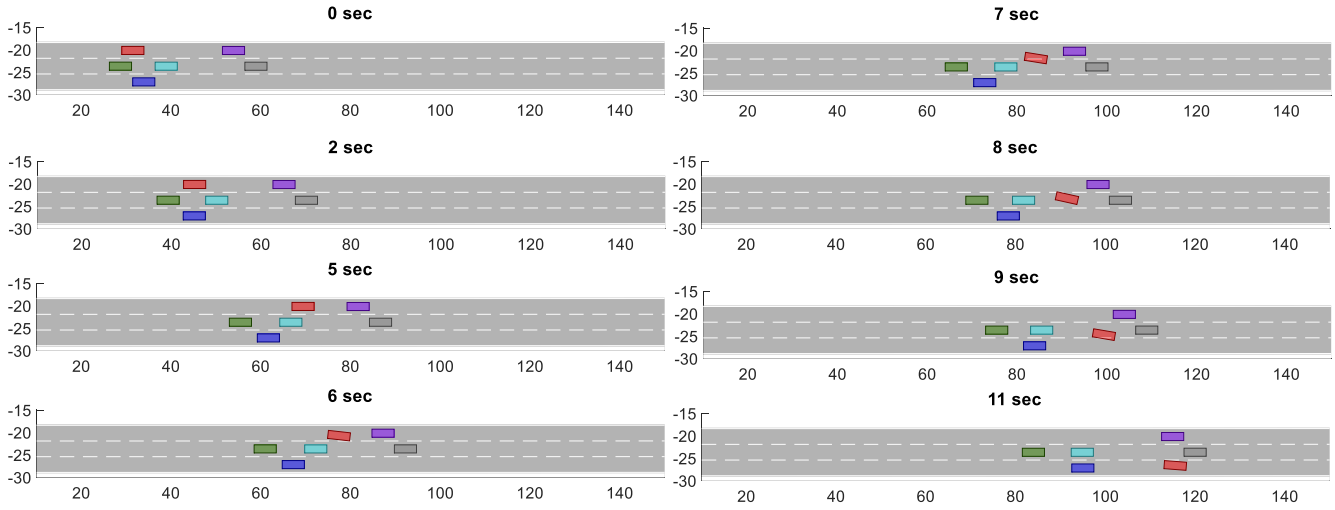

FIGURE 15. Scenario 2, Target area: 3^{rd} lane, red: host vehicle, purple: leading vehicle in the 1^{st} lane, Cyan: leading vehicle in the 2^{nd} lane, green: lagged vehicle in the 2^{nd} lane, blue: leading vehicle in the 3^{rd} lane.

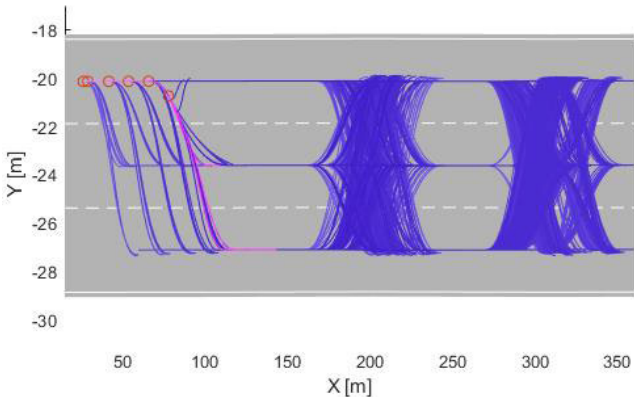

FIGURE 16. Trajectory planning results of scenario 2 blue line: path candidates, magenta line: path-planning MDP.

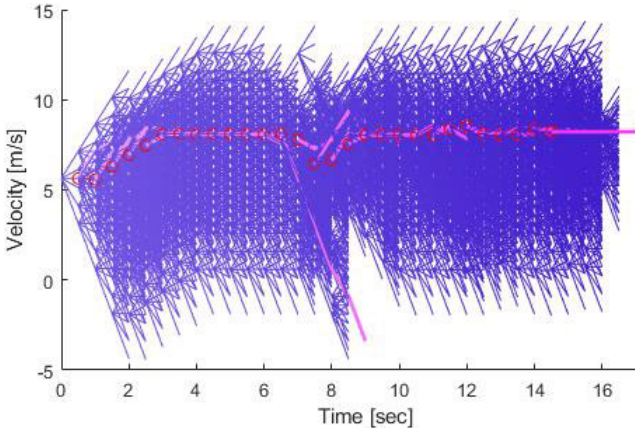

FIGURE 17. Speed planning results of scenario 2 blue line: speed candidates, magenta line: speed-profiling MDP.

of individual state variables on the overall update cost. In this paper, $Q_u = diag([0.25, 4])$ is utilized.

VI. SIMULATION

The proposed self-driving MDP has been verified in various scenarios through a co-simulation of MATLAB/Simulink and high-fidelity vehicle dynamics software. The relevant parameters of the vehicle and road geometry are listed in the Appendix. The entire system is executed on an Intel(R) Core(TM) i7-4790K CPU.

The first scenario involves the host vehicle and surrounding vehicles that are traveling at an initial velocity of 5.55m/s. The host vehicle is positioned in the first lane and the surrounding vehicles are driven according to the traffic model as described in equations (3) $\sim$ (5). The goal of this scenario is to drive the host vehicle to the third lane while considering interaction and collision avoidance. As illustrated in Figure 12, the host vehicle executes lane changes step by step into the second and third lanes where the vehicle behind slows down to accommodate the cut-in motion. The target speed is 8.33m/s. The lane change from the first to the second lane begins at 2 seconds while the host vehicle follows the leading vehicle. Upon completion of the lane change to the second lane, the host vehicle maintains a desired inter-vehicle distance from the leading vehicle. Then, the host vehicle executes another lane change and begins following the leading vehicle in the third lane.

Figure 13 is the visualization of the path-planning MDP tree. The position where the path-planning MDP is updated is described by magenta circles. As illustrated in Figure 13, the MDP is updated when a better route is discovered, such as a lane change to the target lane (second and third circle), or when the predicted motion of the vehicle differs from the measured motion (the other circles). Given the importance

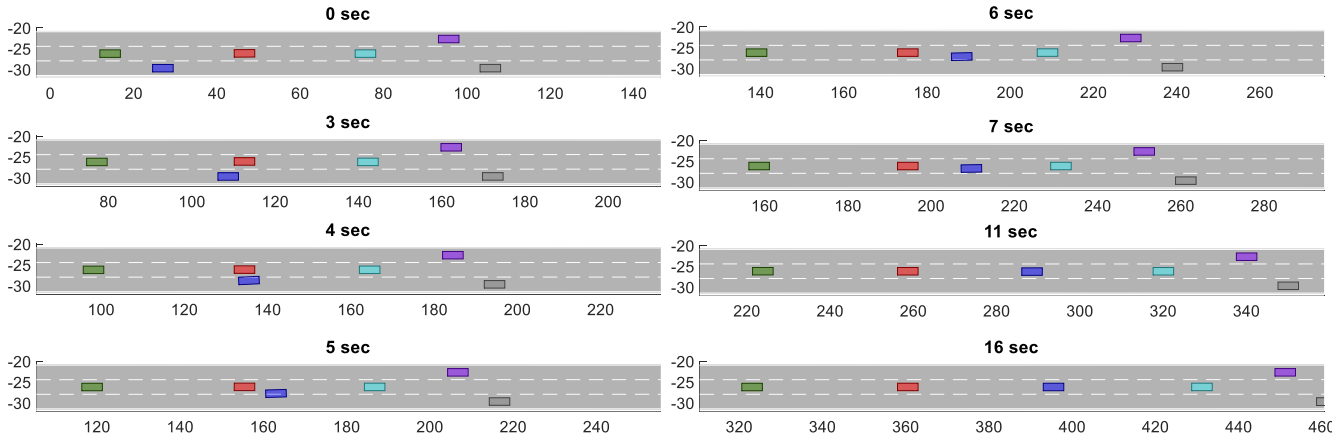

FIGURE 18. Scenario 3, Target area: 2rd lane, red: host vehicle, purple: leading vehicle in the 1st lane, Cyan: leading vehicle in the 2nd lane, green: lagged vehicle in the 2nd lane, blue: Cut-in vehicle.

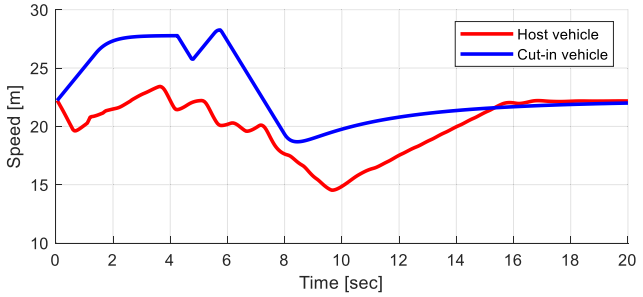

FIGURE 19. Velocity comparison between the host vehicle and cut-in vehicle.

of safe planning and control, it is necessary to consider the convergence of the decision-making algorithm in determining an optimal path. Consequently, the path-planning MDP explores a large number of potential paths for a prolonged duration to identify an optimal path while accounting for the convergence of the decision-making algorithm. In the results of path-planning MDP, the presence of overlapping paths can be observed at approximately 200 meters and 300 meters, which can be attributed to an extensive search. In this scenario, the vehicles behave similarly to the traffic model, resulting in the path-planning MDP being updated when the host vehicle begins a lane change. The speed profiling result in Figure 14 shows that the host vehicle slows down during the lane change from the first to the second lane because the vehicles in the second lane are slower than the host vehicle. Then, the host vehicle decelerates to maintain inter-vehicle space from the blue vehicle in the target lane. As the predicted speed profiling is represented by the magenta color, there is some error due to unmodeled dynamics that affects the performance of the speed-profiling MDP. This error is compensated by the speed-profiling MDP, which operates in a manner similar to receding horizon control.

In the second scenario depicted in Figure 15, the conditions remain the same as in the first scenario, except for the fact that the green and cyan vehicles in the second lane do not facilitate the cut-in vehicle's maneuver. This requires the decision-making algorithm to handle the uncertainty in traffic behavior prediction. The gap penalty is substantial when the path candidates are drawn towards the second lane, where the host vehicle is located at 30m in the X-position. The results of the path-planning, as shown in Figure 16, reveal that the MDP tree is updated six times during a time frame of 0 to 8 seconds. This increased frequency of updates is a result of the anomalous behavior displayed by the green vehicle. At 2 seconds, the host vehicle aborts the cut-in path to the second lane and plans another cut-in behind the grey vehicle at 6 seconds. However, the large error in the MDP state at 8 seconds leads to an update of the path-planning MDP tree, resulting in a new path drawn directly to the target lane. This change is necessitated by the higher predicted collision risk compared to the multiple-lane change penalty. Subsequently, the host vehicle successfully proceeds to the third lane. In the speed-profiling MDP results shown in Figure 17, a significant variance in speed is observed when the host vehicle initiates the lane change from the first to the second lane. This is due to the deviation of the green vehicle's behavior from the modeled traffic motion described in equations (3) to (5). However, these errors can be compensated for by updating the behavior planning MDP, leading to a new optimal path and speed profile, and allowing the host vehicle to reach its desired location and speed.

The third scenario shown in Figure 18 focuses on the performance of the speed profiling MDP. Figure 19 displays the velocity graph of the host vehicle and the cut-in vehicle. The simulation involves the agents being driven at 22.2m/s to assess the high-speed performance. The cut-in vehicle in the third lane, represented in blue, initially lags 20m behind the host vehicle and accelerates to 27.8m/s before initiating

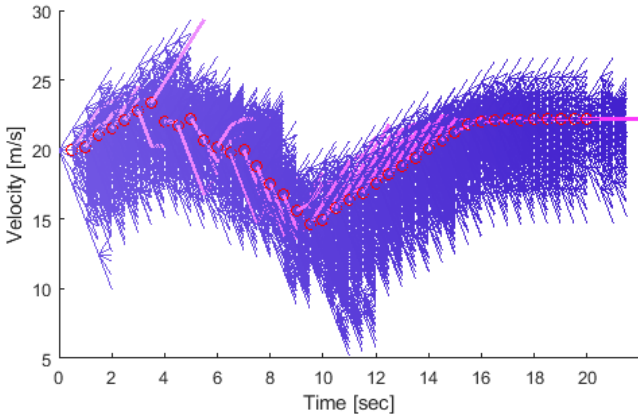

FIGURE 20. Speed planning results of scenario 3. Blue line: speed candidates, magenta line: speed-profiling MDP.

a cut into the second lane at 4 seconds. The speed profiling results in Figure 20 show that the host vehicle's speed profile is designed to decelerate significantly in response to the cut-in vehicle's maneuver. This deceleration occurs before the blue vehicle cuts into the second lane to avoid a collision. Subsequently, the host vehicle decelerates to avoid the imposition of the gap penalty, which starts at approximately 35m of the lead gap. Finally, the host vehicle accelerates to reach the target speed.

VII. CONCLUSION

In this paper, a hierarchical behavior planning approach is presented, consisting of path-planning MDP and speed-profiling MDP. The cut-in motion of the surrounding vehicle is modeled by utilizing the distance and speed-keeping model. The desired speed and inter-vehicle space are obtained from open-source highway traffic data to analyze traffic behavior. To implement the path-planning MDP, the lane-change behavior of human drivers is parameterized using a piecewise linear clothoid (PLC) curve. Additionally, the parameter conditions for the PLC path are established with considerations of vehicle kinematics and steering limitations. A tree structure for the path-planning MDP is constructed by considering the variation in driving time associated with the PLC curve. This structure also includes methods for determining new node rewards and updating the node utility value of the existing nodes. The formulation of self-driving takes into account typical driving behaviors including lane keeping and speed tracking, as well as inter-vehicle behavior and collision avoidance. To account for the uncertainty in predicting the behavior of surrounding vehicles, a collision risk metric is proposed considering the uncertainty of the cut-in prediction model. To enable human-like longitudinal control, an interaction-aware gap penalty is also included. The proposed behavior planning algorithm is evaluated in various scenarios, including cut-in maneuvers in congested traffic and counter-maneuvers against threatening vehicles. For real-time applications, crucial areas that demand attention can be computation time, sensor integration, calibration, uncertain and dynamic environments, as well as safety and regulatory compliance. Thus, the future work will outline a strategic approach to address these challenges, encompassing optimization of computation time, rigorous sensor calibration, enhancement of algorithm robustness, and adherence to safety and regulatory standards.

APPENDIX:

A. PARAMETERS OF THE VEHICLE & BEHAVIOR PLANNING

Symbol	Description [Unit]	Value
dt	Sampling time [sec]	0.5
h	Time gap [sec]	0.3
λ	Time constant for distance keeping [sec]	0.5
L_{wb}	Wheel base of the host vehicle [m]	3
$\delta_{f,max}$	Maximum front-wheel steered angle [rad]	$\pi/6$
$a_{y,al}$	Maximum lateral acceleration	2
$\dot{\delta}_{f,max}$	Slew rate of the MDPS	3π
γ	Discount factor	0.0371
W_{targ}	Target lane keeping reward	20
W_{lane}	Non-target lane keeping reward	10
q_{lane}	Stiffness of the lane keeping reward	0.2
W_{rb}	Penalty for the road boundary	-1000
q_v	Stiffness of the speed-tracking reward	0.001
W_{as}	weight for the longitudinal acceleration	0.0001
W_{ad}	Weight for the lateral acceleration	0.001
W_{LC}	Weight for the lane-change penalty	2
w_{lane}	Lane width [m]	3.5
W_{col}	Penalty for the collision	-20000
τ_{cc}	Time const. of cruise-controlled system [sec]	0.5

REFERENCES

[1] *Summary Report: Standing General Order on Crash Reporting for Level 2 Advanced Driver Assistance Systems*, National Highway Traffic Safety Administration, U.S. Department of Transportation, Washington, DC, USA, 2022, pp. 1–9.

[2] T. Litman, "Autonomous vehicle implementation predictions: Implications for transport planning," TRIS ITRD Database, Tech. Rep. 01727624, 2020.

[3] S.-K. Weng, C.-M. Kuo, and S.-K. Tu, "Video object tracking using adaptive Kalman filter," *J. Vis. Commun. Image Represent.*, vol. 17, no. 6, pp. 1190–1208, Dec. 2006.

[4] A. Haselhoff and A. Kummert, "A vehicle detection system based on Haar and triangle features," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2009, pp. 261–266.

[5] X. Zhang, H. Gao, C. Xue, J. Zhao, and Y. Liu, "Real-time vehicle detection and tracking using improved histogram of gradient features and Kalman filters," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 1, Jan. 2018, Art. no. 172988141774994.

[6] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, vol. 56. Cham, Switzerland: Springer, 2009.

[7] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Trans. Ind. Electron.*, vol. 62, no. 8, pp. 5119–5132, Aug. 2015.

[8] R. Guidolini, L. G. Scart, L. F. R. Jesus, V. B. Cardoso, C. Badue, and T. Oliveira-Santos, "Handling pedestrians in crosswalks using deep neural networks in the IARA autonomous car," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[9] J. G. Mooney and E. N. Johnson, "A comparison of automatic nap-of-the-Earth guidance strategies for helicopters," *J. Field Robot.*, vol. 26, no. 3, pp. 1–17, 2014.

[10] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1993, pp. 802–807.

[11] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[12] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4889–4895.
[13] T. Gu, J. M. Dolan, and J.-W. Lee, "Automated tactical maneuver discovery, reasoning and trajectory planning for autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 5474–5480.
[14] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 2, pp. 740–753, Apr. 2016.
[15] L. C. Fernandes, J. R. Souza, G. Pessin, P. Y. Shinzato, D. Sales, C. Mendes, M. Prado, R. Klaser, A. C. Magalhães, A. Hata, D. Pigatto, K. C. Branco, V. Grassi, F. S. Osorio, and D. F. Wolf, "CaRINA intelligent robotic car: Architectural design and applications," *J. Syst. Archit.*, vol. 60, no. 4, pp. 372–392, Apr. 2014.
[16] T. Hegedűs, B. Németh, and P. Gáspár, "Challenges and possibilities of overtaking strategies for autonomous vehicles," *Periodica Polytechnica Transp. Eng.*, vol. 48, no. 4, pp. 320–326, 2020.
[17] D. Lenz, T. Kessler, and A. Knoll, "Tactical cooperative planning for autonomous highway driving using Monte-Carlo tree search," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 447–453.
[18] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1671–1678.
[19] C. Hubmann, N. Quetschlich, J. Schulz, J. Bernhard, D. Althoff, and C. Stiller, "A POMDP maneuver planner for occlusions in urban scenarios," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 2172–2179.
[20] H. Avilés, M. Negrete, R. Machucho, K. Rivera, D. Trejo, and H. Vargas, "Probabilistic logic Markov decision processes for modeling driving behaviors in self-driving cars," in *Proc. Ibero-Amer. Conf. Artif. Intell.* Cham, Switzerland: Springer, 2022, pp. 366–377.
[21] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," MIT Press, Annu. Res. Rep., Jan. 1998.
[22] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1961–1976, Aug. 2015.
[23] R. Mashayekhi, M. Y. I. Idris, M. H. Anisi, I. Ahmedy, and I. Ali, "Informed RRT*-connect: An asymptotically optimal single-query path planning method," *IEEE Access*, vol. 8, pp. 19842–19852, 2020.
[24] S. Spanogiannopoulos, Y. Zweiri, and L. Seneviratne, "Sampling-based non-holonomic path generation for self-driving cars," *J. Intell. Robotic Syst.*, vol. 104, no. 1, pp. 1–17, Jan. 2022.
[25] C. Pek and M. Althoff, "Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1447–1454.
[26] P. Wang, S. Gao, L. Li, B. Sun, and S. Cheng, "Obstacle avoidance path planning design for autonomous driving vehicles based on an improved artificial potential field algorithm," *Energies*, vol. 12, no. 12, p. 2342, Jun. 2019.
[27] Z. Liu, J. Chen, F. Lan, and H. Xia, "Methodology of hierarchical collision avoidance for high-speed self-driving vehicle based on motion-decoupled extraction of scenarios," *IET Intell. Transp. Syst.*, vol. 14, no. 3, pp. 172–181, Mar. 2020.
[28] N. S. Manikandan, G. Kaliyaperumal, and Y. Wang, "Ad hoc-obstacle avoidance-based navigation system using deep reinforcement learning for self-driving vehicles," *IEEE Access*, vol. 11, pp. 92285–92297, 2023.
[29] Y. Song and K. Huh, "Driving and steering collision avoidance system of autonomous vehicle with model predictive control based on non-convex optimization," *Adv. Mech. Eng.*, vol. 13, no. 6, Jun. 2021, Art. no. 168781402110276.
[30] F. Pusse and M. Klusch, "Hybrid online POMDP planning and deep reinforcement learning for safer self-driving cars," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1013–1020.
[31] *2015 Motor Vehicle Crashes: Overview. Traffic Safety Facts: Research Note*, National Highway Traffic Safety Administration, Washington, DC, USA, 2016, pp. 1–9.
[32] C. Dong, Y. Zhang, and J. M. Dolan, "Lane-change social behavior generator for autonomous driving car by non-parametric regression in reproducing kernel Hilbert space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 4489–4494.
[33] R. Rajamani, *Vehicle Dynamics and Control*. Cham, Switzerland: Springer, 2011.
[34] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, no. 3, pp. 497–516, 1957.
[35] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393, Oct. 1990.
[36] G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*. Amsterdam, The Netherlands: Elsevier, 2014.
[37] J. McCrae and K. Singh, "Sketching piecewise clothoid curves," *Comput. Graph.*, vol. 33, no. 4, pp. 452–461, Aug. 2009.
[38] H. Kurniawati and V. Yadav, "An online POMDP solver for uncertainty planning in dynamic environment," in *Robotics Research*. Cham, Switzerland: Springer, 2016, pp. 611–629.
[39] E. Deza, M. M. Deza, M. M. Deza, and E. Deza, *Encyclopedia of Distances*. Cham, Switzerland: Springer, 2009.
[40] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Robot. INST, Carnegie-Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-92-0, 1992.

YUHO SONG received the bachelor's degree in automotive engineering and the integrated master's and Ph.D. degree in automotive engineering from Hanyang University, in 2015 and 2022, respectively. His current research interests include the domains of action planning for advanced driver assistance systems (ADAS), autonomous driving, and the dynamic behavioral control of vehicle-type robots.

SANGWON HAN received the B.S. degree in automotive engineering from Hanyang University, Seoul, South Korea, in 2018. He is currently pursuing the Ph.D. degree with the Department of Automotive Engineering (Automotive-Computer Convergence). His research interests include the planning and control of commercial vehicles, which is path planning considering surrounding vehicles or obstacles in large vehicles and developing optimal control systems through analysis of the dynamic characteristics of vehicles.

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