Hybrid Trajectory Planning for Autonomous Driving in On-Road Dynamic Scenarios

Wonteak Lim, *Member, IEEE*, Seongjin Lee, *Student Member, IEEE*, Myoungho Sunwoo, *Member, IEEE*, and Kichun Jo, *Member, IEEE*

***Abstract*— A safe trajectory planning for on-road autonomous driving is a challenging problem owing to the variety and complexity of driving environments. The problem should involve the consideration of numerous aspects such as road geometry, lane-structured roads, traffic regulations, traffic participants, and vehicle physical limitations. Furthermore, dynamically changing movements of surrounding vehicles make the problem more challenging. It requires the planner's ability to react to the changes in the driving environments in real time. To solve this problem, sampling and numerical optimization-based trajectory planners were introduced. However, these methods have their own limitations in generating a safe trajectory in these dynamic scenarios. To overcome these issues, this paper proposes a hybrid trajectory planning scheme to integrate the strength of the sampling and optimization methods. With the sampling method for a lateral movement, the planner can deal with various trajectories with multiple maneuvers. This helps the planner to generate a reactive trajectory in a dynamically changing environment. The numerical optimization of a longitudinal movement enables the planner to adapt to diverse situations without restriction of predefined patterns for specific driving purposes. The proposed method was implemented with an embedded optimization coder and C++ environment. Based on this, its performance was evaluated through simulation and real driving tests in various on-road dynamic scenarios.**

***Index Terms*— Planning, trajectory planning, motion planning, trajectory generation, path planning, autonomous driving, dynamic scenario, lane change, hybrid, sampling, and optimization.**

I. Introduction

OVER the last few decades, autonomous driving has been a key technology for improving driving safety and convenience in road traffic. Although various research activities, from the DARPA Urban challenge [1] to demonstrations of autonomous driving [2]–[5], have been conducted, several challenges remain for fully autonomous driving. One of the challenging tasks is trajectory planning [6]–[8] in time-critical street scenarios such as lane change in a dynamically changing traffic flow, as described in Fig. 1. The vehicle must perform lane change when the vehicle wants to exit the highway or make a right turn at the next intersection. In addition, the trajectory planning becomes more complicated because of the various driving situations due to different types of roads, traffic regulations, barriers, and traffic participants. To cope with the complex and time-critical scenarios, the planner should produce safe and dynamically feasible trajectories for a wide range of driving situations. Furthermore, this planner requires real-time capabilities to ensure a rapid reaction to the change in the driving environment.

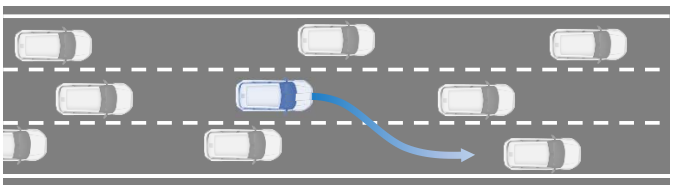

Fig. 1. Lane change in dense traffic flow.

To solve these dynamic problems, various approaches for trajectory planning have been studied recently, which can be organized into two main categories: sampling and optimization-based methods. The sampling-based method [9]–[14] generates the best trajectory by selecting the optimal one among the collision-free trajectory candidates. These candidates are spread in each iteration based on the predefined patterns. The well-known patterns for discretizing and formulating the planning problems are pan-like [10], [11], lattice [9], and terminal manifold [12]. An advantage of these methods is its multi-maneuver capability to manage the combinatorial aspects of on-road trajectory planning. The planner generates vast candidates with different maneuvers such as lane keeping, lane changing, merging, and split. With the maneuver diversity, the generated trajectory is not limited to a specific maneuver selected by the behavior decision but can take into account a wide range of driving possibilities. In addition, a computation effort can be optimized by customizing the number of candidates. Against these strengths, the sampling-based planner delivers only suboptimal solutions depending on the discretization of the planning problem.

Manuscript received December 2, 2018; revised June 29, 2019; accepted November 21, 2019. Date of publication December 13, 2019; date of current version December 24, 2020. This work was supported in part by the BK21 Plus Program through the Ministry of Education, Republic of Korea, under Grant 22A20130000045, in part by the Industrial Strategy Technology Development Program under Grant 10039673, Grant 10060068, Grant 10042633, Grant 10079961, and Grant 10080284, in part by the International Collaborative Research and Development Program through the Ministry of Trade, Industry and Energy (MOTIE Korea), under Grant N0001992, and in part by the National Research Foundation of Korea (NRF) Grant Funded by the Korean Government under Grant 2011-0017495 and Grant 2019R1G1A1099806. The Associate Editor for this article was C. Guo. *(Corresponding author: Kichun Jo.)*

W. Lim, S. Lee, and M. Sunwoo are with the Department of Automotive Engineering, Hanyang University, Seoul 133-791, South Korea.

K. Jo is with the Department of Smart Vehicle Engineering, Konkuk University, Seoul 05029, South Korea (e-mail: kichun.jo@gmail.com).

Digital Object Identifier 10.1109/TITS.2019.2957797

The literature [12] provides efficient patterns to formulate planning problems as a simplified curve model. This model can generate reasonable trajectories in simple traffic scenarios. However, the performance degrades in complex scenarios such as lane change in dense traffic. This method only selects the safe one among the trajectory candidates generated based on the curve patterns. Since these patterns are predefined and do not change in real time, the resultant trajectory induces a passive action, which indicates that the planner cannot find a solution if the candidates do not contain it. For this reason, this method struggles to generate the lane-changing trajectory in dense traffic.

The other planner is based on a continuous optimization method [15]–[17]. This method makes a mathematical model with a cost function and constraints. The constrained optimization problem can be solved by linear or nonlinear programming according to the complexity of the cost and constraint functions. Contrary to the sampling approach, the numerical method offers the advantage that the optimal trajectory is not restricted by the predefined pattern. The cost and constraint functions adjust to the driving environment with road geometry, boundaries, and obstacles. This adaptation leads the final trajectory to have high flexibility on the change in the environment. It helps to make the active action to overcome the complex scenarios in contrast to the sampling approach. Unfortunately, this process requires a high computational resource to solve the space-time critical planning problem with dynamic obstacles because the problem is formulated by the nonlinear optimization problem. The stronger the nonlinear characteristics is, the more difficult it is for the solver to find the optimal solution. Moreover, the nonlinear programming requires pre-processes to determine the desired behavior and select an initial state suitable for the behavior. Depending on the choice of the behavior (or maneuver), the cost and constraint functions are different even in the same environment. Occasionally, the planner fails to find a safe trajectory for the selected behavior although feasible solutions exist for other behaviors. To overcome this failure, the planner should prepare an alternative method to generate a backup trajectory [17].

Recently, some authors have proposed a hybrid scheme of the sampling and optimization methods to combine their characteristics. The author [18] integrated multiple-phase state sampling for path generation and the convex optimization for the speed profile. Since this method focuses on a highly constrained environment with static obstacles, it is not suitable for on-road trajectory planning in dynamic scenarios. The hybrid discrete-parametric optimization [19] uses the sampling method to select the appropriate initial point. Based on the initialization, the final trajectory is generated by the continuous optimization [15]. This approach provides an effective idea to overcome the initial point problem of the nonlinear programming. However, the final solution is found by the optimization method, which means that this method still needs a backup solution for failure. Furthermore, the hierarchical concept for efficient trajectory planning [20] defines not only an initial point for the optimization problem but also space and time boundaries by the behavior trajectory layer. This method minimize the gap between the behavior planning layer and local planning layer in motion planning, and helps to maximize the performance of the numerical optimization for local trajectory generation in dynamic scenarios. However, its performance relies on the convergence of the optimization problem. For this reason, an additional algorithm is required to consider multiple maneuvers in preparation of the failure in the selected behavior. As described above, the previous hybrid methods have their own strong points; however, they are not suitable to generate a safe trajectory in highly dynamic scenarios due to the following reason. They still need a backup trajectory planner for a failure-safe function or generate only a passive action in complex situations such as the sampling method.

To overcome these limitations, this paper proposes a safe and efficient trajectory planning scheme for dynamic scenarios by merging the strengths of the sampling and numerical optimization methods. The trajectory planning problem is divided into two steps: sampling for lateral movement and numerical optimization for longitudinal movement. The lateral trajectories are generated by the sampling method [12] for multiple maneuvers. Each lateral candidate is modeled by a quintic polynomial. This is an efficient way to represent lateral movements for the various maneuvers. To fulfill each maneuver, its longitudinal movement is numerically optimized by linear Model Predictive Control (MPC) [17]. Owing to its convex formulation with a quadratic form, the optimal solution can be found with low computation. By using the MPC scheme, it is a strong advantage that we do not need to design the complex sampling pattern for the longitudinal candidate. Moreover, the longitudinal trajectory can be actively adapted to the various environment because the cost and constraint functions are designed by the obstacles and maneuvers in each planning step. Finally, the method selects the best one among the trajectory candidates including parametric-lateral and discrete-longitudinal movements. The selection scheme gives a failure-safe function in which the alternative trajectories are always prepared, although the trajectory for the specific maneuver is infeasible. Therefore, generation of a smooth and collision-free trajectory can be guaranteed in dynamic street scenarios with low computation. This performance is evaluated with simulation and real-driving experiments in time-critical scenarios.

The rest of this paper is organized as follows. Section II defines the trajectory planning problem. Section III explains the overall architecture of the proposed method. Furthermore, this section describes the main steps for generating the optimal trajectory based on the hybrid method. This method is evaluated by simulation and real driving tests in Section IV and V. Finally, Section VI concludes this paper.

II. Problem Statement

A. Planning Space

A trajectory planner for on-road scenarios monitors and manages various surrounding environments such as vehicles, bikes, pedestrian, road shape, lane boundaries, traffic light, and traffic sign. To deal with these effectively, this paper

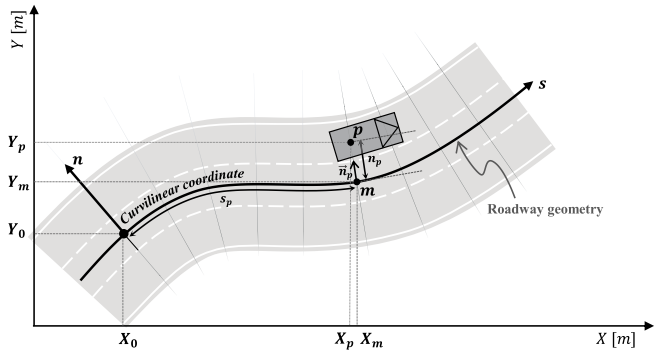

Fig. 2. Coordinate conversion between the Cartesian and the Curvilinear coordinates.

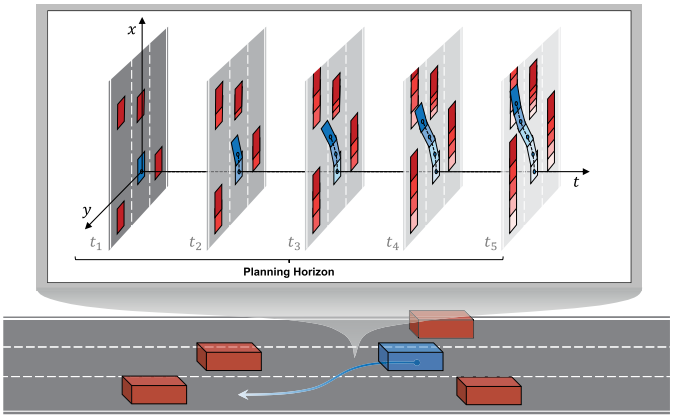

Fig. 3. A trajectory planning problem is to find a safe trajectory on three dimensions of space (x and y) and time (t), and its result should be collision-free within a planning horizon.

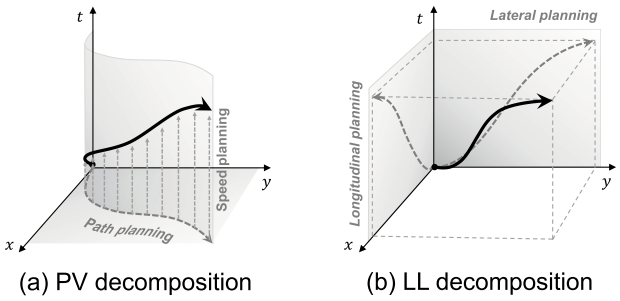

Fig. 4. Trajectory decomposition methods: Path-Velocity decomposition (PV) and Longitudinal-Lateral trajectory decomposition (LL).

defines the planning problem based on the planning space [20]. The planning space refers to a data abstraction layer to represent the driving environment surrounding the ego vehicle. Its advantage lies in the fact that the planner can manage the various elements with common interface regardless of the sensor configuration of the autonomous system.

The planning space organizes the environment for trajectory planning by using three components: object, Traffic Control Device (TCD), and route. The object means a spatial obstacle to be avoided and contains static and dynamic obstacles depending on their moving characteristics. Each obstacle is modeled by an occupancy grid map [21] and a bounding box model [22]. TCD is a device used to present the traffic flow and regulations including traffic light and sign. They can be represented by a state machine. However, this element is not mentioned in detail in this paper because it is beyond this study. The final element is the route to abstract a road for trajectory planning. The road information is a key element to help the planner to determine where the ego vehicle will drive. To fulfill the objective, the route is modeled by a hierarchical structure consisting of the global and local route. The global route indicates sequential road connections from the start to the destination, and contains road segments that the vehicle will follow. Based on the global route, the local route is an autonomous driving horizon to represent the more near-sighted and detailed environment. It contains road geometry, road boundary, and lane information surrounding the vehicle. To model the local route, various models can be used. Among these, this study utilizes the B-spline representation [20], [23].

B. Curvilinear Coordinate

In a lane-structured road, lane geometry is the main clue for helping human drivers to plan their trajectories. For example, they drive following the lane geometry except when they change a lane. Even when they change a lane, the humans determine their trajectories based on their lateral movement relative to the lane geometry rather than the absolute ground.

To achieve this human-like driving, this paper formulates the trajectory planning problem on Curvilinear coordinate system, as shown in Fig. 2. As opposed to the Cartesian coordinate (X, Y), the Curvilinear coordinate (s, n) is established relative to the roadway geometry obtained from the digital map [20]. The state (x_p, y_p) of the point p on the Cartesian coordinate can be converted into the state (s_p, n_p) on the Curvilinear coordinate with the root point m. s_p indicates the arc length from the origin point (X_0, Y_0) to the point m. n_p refers to the lateral offset toward the direction $\vec{n}_p$ from the point m to the point p. Its direction is orthogonal to the s-direction $\vec{s}_p$.

Based on this coordinate conversion, all states of the planning space on the Cartesian coordinate are converted into the states on the Curvilinear coordinate system. In other words, the trajectory planning problem can be formulated by the Curvilinear coordinate system. Through the conversion, the planner can plan the vehicle's trajectory relative to the lanes, like human drivers. It is a more intuitive representation for on-road planning than the Cartesian coordinate system. Furthermore, the planner can easily adapt curve scenarios by using the Curvilinear coordinate system.

In the final step, the resultant trajectory returns to the Cartesian coordinate because its coordinate system is more suitable for vehicle control. The Curvilinear coordinate is used only inside trajectory planning. The details about the transformations between these coordinate systems are summarized in the literature [11], [12].

C. Space-Time Coupling Problem

Based on the Curvilinear coordinate system, the on-road planning problem aims to generate a safe and smooth trajectory to avoid collisions with dynamic obstacles, as shown

in Fig. 3. Since each state of the obstacles changes dynamically with time, a trajectory planner should find the optimal trajectory by considering not only space but also time. In other words, this problem is called the space-time coupled problem, and is so complex and time-consuming that it is difficult to be solved in real-time.

To overcome the coupling problem, the literature [12], [24] proposed two types of decomposition methods: Path-Velocity (PV) and Longitudinal-Lateral direction (LL) decomposition, as shown in Fig. 4. The PV method is the most famous method for separating the 3D problem into a combination of path planning and velocity planning, as described in Fig. 4 (a). The path planner finds a spatial path with a static environment. Along the path, the velocity planner optimizes the vehicle's velocity to avoid collisions with the dynamic obstacles. It is known as an intuitive solution because the problem is converted into two parts: where to go and when to go. In a static environment with sparse dynamic obstacles, it is an efficient solution to find a safe trajectory. However, it is not suitable for on-road dynamic scenarios because the method cannot provide a safe solution in highly dynamic situations. As opposed to PV, LL decomposition divides the trajectory planning problem into longitudinal and lateral trajectory planning, as shown in Fig. 4 (b). Since each direction's trajectory can consider its movement by time, this method can generate a more dynamic trajectory than PV. Moreover, LL is more efficient method than PV to represent the on-road trajectory planning because a human driver makes an on-road driving strategy with two separate directions, such as longitudinal and lateral movements relative to a lane. To cover the dynamic street scenarios, this paper utilizes the LL method on the Curvilinear coordinate.

III. Hybrid Trajectory Planning

Motion planning is a core module to determine the vehicle's behavior and trajectory based on the perception information as shown in Fig 5. The motion planning has three steps: planning space generation, behavior planning, and hybrid trajectory planning. The planning space represents driving environment as a common interface with route, object, and Traffic Control Device (TCD). Based on the information, a behavior planner determines a long-term maneuver for safe driving. The trajectory planner generates a motion-level trajectory to obey the determined maneuver within the vehicle's physical limitations. To consider various maneuvers and dynamic scenarios, this paper proposes the hybrid scheme for the trajectory planning. The overall structure of the method obeys the general sampling method's steps: candidate generation and optimal trajectory selection. Contrary to the previous methods, this method generates trajectory candidates with the hybrid process of the sampling and optimization methods. The terminal-state sampling method spreads the lateral trajectory candidates considering the type of road, number of lanes, and possible lateral maneuvers including lane keeping and changing. Based on these, the Linear Model Predictive Control (LMPC) optimizes their longitudinal trajectories considering the maneuvers and potential collisions with obstacles. Among these, infeasible trajectories are filtered out through consideration of vehicle dynamics, safety constraints, and collision. In the final step, this algorithm selects a safe and optimal trajectory considering the degree of driving comfort, safety, and the preferred maneuver selected by the behavior planning. A detailed description for each module will be expressed as follows.

A. Sampling for Lateral Movement

1) Modeling of Lateral Movement: On a lane-structured road, the ideal behavior of an autonomous vehicle is to move along the road if there are no exceptions: lane change and an obstacle avoidance maneuver. For these exceptional maneuvers, the vehicle needs to make a certain lateral offset to the center of the desired lane. The movement by time was successfully designed and implemented by [12]. Similar to this study, the proposed method uses a quintic polynomial on the curvilinear coordinate to represent a lateral movement. The quintic polynomial requires six parameters as follows:

$$n(t) = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0. \tag{1}$$

As the jerk-optimal connection [25], these parameters can be found from three types of components: a initial state $N_0 = [n_0, \dot{n}_0, \ddot{n}_0]$, a terminal state $N_T = [n_T, \dot{n}_T, \ddot{n}_T]$ and a time interval T as follows:

$$\begin{aligned} n(0) &= n_0, \quad n(T) = n_T \\ \dot{n}(0) &= \dot{n}_0, \quad \dot{n}(T) = \dot{n}_T \\ \ddot{n}(0) &= \ddot{n}_0, \quad \ddot{n}(T) = \ddot{n}_T. \end{aligned} \tag{2}$$

Based on the equations (1) and (2), six corresponding equations are derived as

$$\begin{aligned} n_0 &= c_0, \\ \dot{n}_0 &= c_1, \\ \ddot{n}_0 &= 2c_2, \\ n_T &= c_5 T^5 + c_4 T^4 + c_3 T^3 + c_2 T^2 + c_1 T + c_0, \\ \dot{n}_T &= 5c_5 T^4 + 4c_4 T^3 + 3c_3 T^2 + 2c_2 T + c_1, \\ \ddot{n}_T &= 20c_5 T^3 + 12c_4 T^2 + 6c_3 T + 2c_2. \end{aligned} \tag{3}$$

The six independent equations are a sufficient condition to determine the six unknown variables of c_0, c_1, c_2, c_3, c_4, and c_5. As a result, these parameters of the polynomial are summarized by

$$\begin{bmatrix} c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} \ddot{n}_0/2 \\ \dot{n}_0 \\ n_0 \end{bmatrix}, \tag{4}$$

$$\begin{bmatrix} c_5 \\ c_4 \\ c_3 \end{bmatrix} = \begin{bmatrix} T^5 & T^4 & T^3 \\ 5T^4 & 4T^3 & 3T^2 \\ 20T^3 & 12T^2 & 6T \end{bmatrix}^{-1} \begin{bmatrix} n_T - (c_2 T^2 + c_1 T + c_0) \\ \dot{n}_T - (2c_2 T + c_1) \\ \ddot{n}_T - (2c_2) \end{bmatrix}. \tag{5}$$

Since the initial state is predetermined by the current vehicle state, this curve for the lateral movement is determined by the terminal state and time interval. This means the convergence state and convergence time of the lateral trajectory. By choosing appropriate parameters, this method easily generates the jerk-optimal lateral trajectory. However, it is difficult to select these elements suitable for the

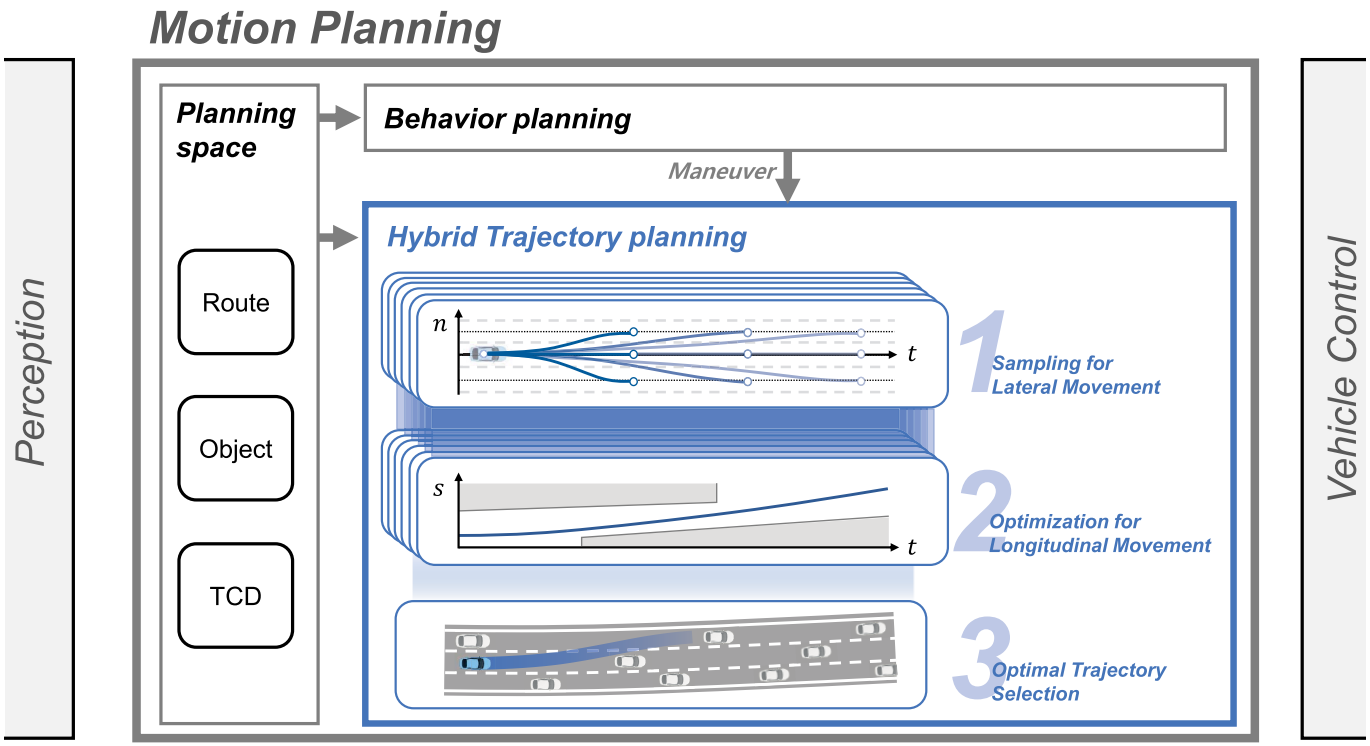

Fig. 5. Architecture of motion planning for autonomous driving.

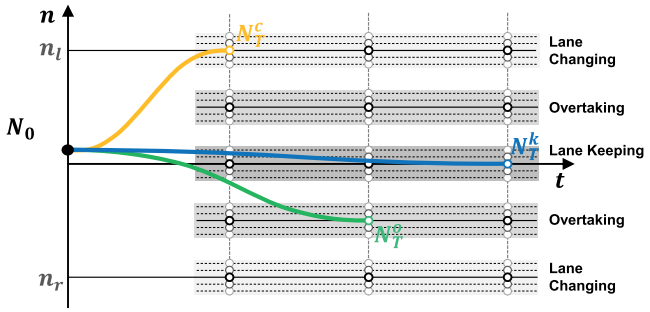

Fig. 6. Lateral candidates are generated as three groups: lane keeping, overtaking, and lane changing.

surrounding environment. To determine them efficiently, this paper uses the state sampling method. The details are described in the following section.

2) Candidate Generation: The lateral movement of on-road driving can be categorized as the following three maneuvers: lane keeping, lane changing, and overtaking. For example, a car tries to follow the center of the road for the lane keeping maneuver. Contrary to the maneuver, lane changing and overtaking need the lateral offset to the center of the ego lane. Following this strategy, the samples of the terminal states are generated as three groups: samples for lane keeping, samples for lane changing, and samples for overtaking as described in Fig. 6.

The terminal states N_T^k for lane keeping (blue) have a common lateral offset with zero. However, we diversify these candidates with different interval times. Owing to their variety, we can adjust the convergence time to the desired lane. This time should be adjusted considering the trade-off between the convergence speed and driving comfort. A detailed approach to select the optimal time will be described in section VI.

To cover the overtaking trajectory (green), the terminal samples N_T^o are spread in a grid shape. Similar to the overtaking, the samples N_T^c for lane changing (yellow) also have certain lateral offsets, which are determined by the lateral offset n_l for a left lane or n_r for a right lane.

For a feasible lateral movement, these candidates should be restricted within a safe acceleration limit as follows:

$$|a_{lat}| \leq a_{lat}^{limit} \tag{6}$$

where the lateral acceleration limit a_{lat}^{limit} is a user-defined parameter determined by the physical limit and driving comfort. To prune out the infeasible candidates, the maximum acceleration $a_{lat,i}^{max}$ and minimum acceleration $a_{lat,i}^{min}$ of the i^{th} candidate are calculated by

$$\begin{aligned} a_{lat,i}^{max} &= max\ \ddot{n}(t) \quad 0 \leq t \leq T_i, \\ a_{lat,i}^{min} &= min\ \ddot{n}(t) \quad 0 \leq t \leq T_i. \end{aligned} \tag{7}$$

In summary, infeasible candidates with exceeding acceleration are removed from the lateral movement candidates by the following feasibility conditions:

$$(-a_{lat}^{limit} \leq a_{lat,i}^{min}) \quad and \quad (a_{lat,i}^{max} \leq a_{lat}^{limit}) \tag{8}$$

B. Optimization for Longitudinal Movement

Unlike the lateral movement, a longitudinal movement is strongly affected by the motions of surrounding traffic participants. If there are no surrounding participants, a car tries

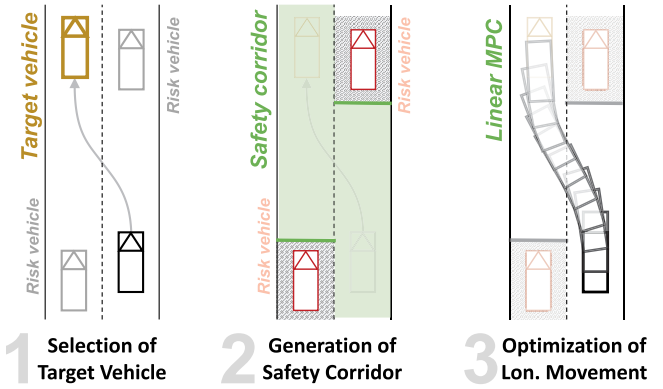

Fig. 7. A longitudinal movement is optimized by the following three steps: selection of a target vehicle, generation of a safety corridor, and numerical optimization for the movement.

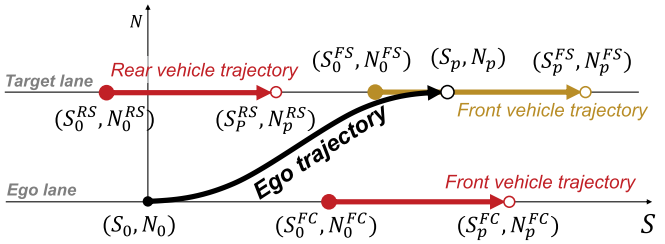

Fig. 8. A target vehicle is selected based the lateral ego's trajectory (a black line) and surrounding vehicle's trajectories (red lines).

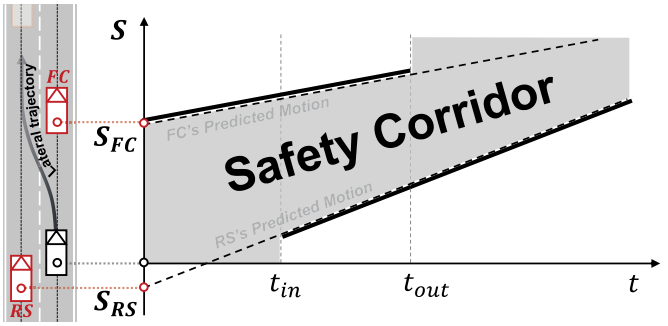

Fig. 9. Longitudinal safety corridor indicates a collision-free space between the risk vehicles.

to maintain a constant speed as long as the curvature of the road is not too high. However, the car's speed should be dynamically changed to avoid collision with others when the surrounding obstacles exit on the ego's driving path. These obstacles can be divided into two groups: a target and a risk obstacle. The target obstacle is also called the preceding, front, and following vehicle in the literature [26], [27]. The target indicates the object that an ego follows in a longitudinal direction. In a case of Adaptive Cruise Control (ACC), the ego vehicle maintains a constant time gap to the target vehicle for safety and stability. The risk obstacle refers to an object that the ego vehicle should avoid for safe driving.

Following this strategy, the longitudinal planner generates an optimal speed profile in three steps: selecting a target vehicle, generating safe corridors, and optimizing a speed profile based on a linear Model Predictive Control (MPC) scheme. These steps are described in Fig. 7. The first step chooses the target and risk obstacles based on the trajectory of each lateral candidate. The risk obstacles give information related to a free space by time. The space defines a safety corridor. Based on the target obstacle and safety corridors, we can define the longitudinal planning problem as a MPC problem with a quadratic cost function and boundary conditions. This can be easily solved by using a Quadratic Programming (QP) solver.

1) Selection of a Target Vehicle: A target vehicle is an object that the ego vehicle follows and keeps a safe distance from. It is a major factor for determining the ego's longitudinal movement. In other words, the ego's longitudinal movement can vary depending on which one the target vehicle is. When the ego keeps a lane, it is obvious that the target is a front vehicle in the same lane. However, the target selection at the time of starting a lane change varies depending on when or how fast the ego approaches the changing lane.

To select a reasonable target, this paper proposes the target selection method by using the lateral movement candidates. Based on maneuver of each candidate, the target selection is divided into two sub-algorithms: a target for keeping a lane and that for changing a lane. For lane keeping, the target vehicle is the vehicle driving on the same lane as the ego one and existing in front of it. For the lane changing maneuver, this algorithm selects a target vehicle based on the lateral candidate. As described in Fig. 8, the surrounding vehicles' trajectories from the current state (S_0^x, N_0^x) to the terminate state (S_p^x, N_p^x) can be predicted by the motion prediction module [28] for the prediction time p. This is equal to the convergence time T of the lateral candidate. The ego's trajectory can be predicted by combining the lateral and longitudinal movements. However, the longitudinal movement is unknown. To select the best target vehicle that minimizes the change in the ego's speed during lane changing, we assume that its speed is constant for the prediction time p. Through the prediction process, the method chooses the front-closest vehicle (S_p^2, N_p^2) at the time p as the target vehicle.

2) Generation of a Safety Corridor: From the safe driving viewpoint, the surrounding vehicles are categorized into two types: a target and a risk vehicle. Unlike the target vehicle described in the previous section, the risk vehicle refers to an object that the ego vehicle should avoid for safe driving. In on-road driving, the remainders, except for the target vehicle, are chosen as the risk vehicles shown in Fig. 7.

To avoid a collision with these obstacles, this paper defines a safety corridor as described in Fig. 9. The safety corridor indicates a collision-free space where the ego vehicle can drive toward the longitudinal direction without any collision with these moving obstacles. The corridor is estimated by the ego's lateral trajectory and predicted motions of the front-center (FC) and rear-side (RS) vehicles. The lateral trajectory is a given condition from the candidate of the lateral trajectories in Section IV. The obstacles' trajectories can be predicted by the motion prediction models [28]. Based on them, the final safety corridor is represented as the grey area.

Depending on the ego's lateral positions n_{in} and n_{out} and corresponding times t_{in} and t_{out}, this corridor is divided into

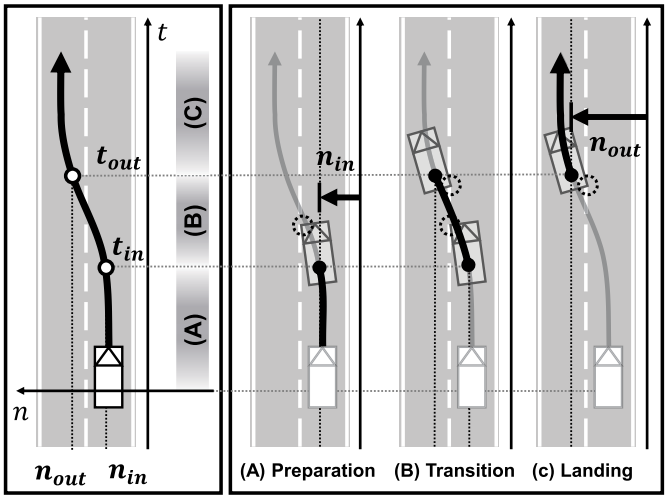

Fig. 10. Longitudinal safety corridor contains three regions: (a) preparation, (b) transition, and (c) landing.

three parts: preparation (A), transition (B), and landing (C) as shown in Fig. 10. The preparation from 0 to t_{in} is the period for which the ego exists in the current lane. For this reason, its motion is affected only by the Front Center (FC) obstacle. In the transition part, the safe region is bounded by trajectories of both the Rear Side (RS) and FC obstacles because the ego straddles two lanes. The landing part refers the convergence region in which the ego's volume is out of the current lane. In this region, the corridor has only a lower limit by the RS's trajectory. The times t_{in} and t_{out} are determined by the shape of the lateral trajectory.

Within the safety corridor, generation of the safe longitudinal trajectory is described in the following section. However, if the upper and lower boundaries are crossing, it means that the ego's lateral movement is too slow to penetrate the gap between the FC and RS obstacles. In other words, the planner cannot generate a collision-free trajectory with following the selected lateral trajectory. For this reason, this lateral trajectory with the crossing boundaries is removed from the set of candidates.

3) Optimization of a Speed Profile Based on a Linear MPC Scheme: The longitudinal planning problem aims to generate an optimal speed profile considering the target obstacle and risk obstacles. The profile leads to the autonomous car maintaining a certain distance from the target obstacle and avoiding collisions with the risk obstacles. If there are no target and risk obstacles, we want the ego vehicle to maintain the desired speed. These various intents can be generalized into four problems: Speed keeping without corridors (SK), Speed keeping with corridors (SKC), Following the target without corridors (FT), and Following the target with corridors (FTC). The proper one is selected by the existence of target obstacle and safety corridors, as shown in Table I.

To find optimal solutions for these problems, this paper uses a MPC scheme, which is an optimal control method for predicting its future state by the system model and finding the optimal inputs by minimizing the predictive cost. The general

TABLE I
TYPES OF QUADRATIC PROGRAMS (QPS)

Target	Corridor	Type of Quadratic Programs
X	X	Speed Keeping (SK)
X	O	Speed Keeping with Corridors (SKC)
O	X	Target Following (TF)
O	O	Target Following with Corridors (TFC)

MPC model is

$$\begin{aligned} &minimize\ f = \sum_{t=0}^{N} (e_t^T Q e_t + u_t^T R_1 u_t + \Delta u^T R_2 \Delta u) \\ &subject\ to\ X_{t+1} = AX_t + Bu_t, \quad t = 0, \ldots, N \\ &\qquad u_t \geq u_{min}\ and\ u_t \leq u_{max}, \quad t = 0, \ldots, N. \end{aligned} \tag{9}$$

The cost function f is a quadratic form for representing the magnitude of the weighted sum of the tracking error e_t, the control effort u_t, and the input change δu_t. Their weights are Q, R_1, and R_2, respectively. The equality constraint, including the system matrix A and the input matrix B, estimates the future state X_{t+1} based on the current state x_t and the input u_t during the prediction horizon N. By minimizing the cost function with following the constraints, the optimization solver finds the optimal control input between u_{min} and u_{max}.

The MPC models for the four longitudinal problems are modeled in Table II. In this table, the error state e_t is defined as $(X - X_r)$. The SK model is designed with the constant acceleration model with a speed v and acceleration a. To follow the desired speed v_{ref}, the error state e_t is the gap between the predictive speed v_t and the desired one v_{ref}. The desired speed v_{ref} is determined by the regulation speed of a road and the maximum speed by the road curvature. Contrary to the SK model, the TF model wants to follow the target obstacle and maintain a safe distance from the target. For this reason, the state X of this model contains a position difference Δs and a speed difference Δv between the ego and the target obstacle. The error state e_t for the TF is defined as $\Delta s - d_r$ and Δv. Each element of the error is weighted by q_{df} and $q_v f$ for the optimization. This MPC model has been widely used for Adaptive Cruise Control (ACC) in the literature [27]. Based on the SK and FT models, the SKC and FTC models have additional state variables and constraints for the safety corridors. The related parameters are shown in Table II. The constraints contain the upper and lower limits of the longitudinal position as follows:

$$s_t \geq s_t^{min}\ \ and\ \ s_t \leq s_t^{max}, \qquad t = 0, ..., T+1. \tag{10}$$

As shown in the equations (9) and (10), the MPC model has a quadratic cost function with linear constraints. Because of the positive definite Q, R_1, and R_2, the quadratic problem can be solved in polynomial time [29]. The quadratic programming can be solved with variable methods of interior point [30],

TABLE II
MPC MODELS FOR LONGITUDINAL PLANNING

	X	X_r	A	B	Q
SK	$[v]$	$[v_r]$	$[1]$	$[\Delta t]$	$[q_{vk}]$
SKC	$\begin{bmatrix} s \\ v \end{bmatrix}$	$\begin{bmatrix} 0 \\ v_r \end{bmatrix}$	$\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & q_{vk} \end{bmatrix}$
TF	$\begin{bmatrix} \Delta s \\ \Delta v \end{bmatrix}$	$\begin{bmatrix} d_r \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -0.5\Delta t^2 \\ -\Delta t \end{bmatrix}$	$\begin{bmatrix} q_{df} & 0 \\ 0 & q_{vf} \end{bmatrix}$
TFC	$\begin{bmatrix} s \\ v \\ \Delta s \\ \Delta v \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ d_r \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \\ -0.5\Delta t^2 \\ -\Delta t \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q_{df} & 0 \\ 0 & 0 & 0 & q_{vf} \end{bmatrix}$

active set [31], augmented Lagrangian [31], conjugated gradient [32], and gradient project [33]. Among them, this paper solves the problem with the interior point method [34]. For real time performance, the algorithm should set the maximum computational time T_{opt}. This bounds the worst time to find an optimal value.

C. Optimal Trajectory Selection

Through the hybrid scheme of the sampling and optimization methods, each trajectory candidate has its own safe moving plan for a specific maneuver. First, this algorithm checks if the maximum speed of each trajectory is within the speed boundary which is defined by the regulation speed and the road's curvature. Among the various feasible candidates, the trajectory planner should select the optimal one considering safety, comfort, efficiency, and a long-term objective. To evaluate them, this method calculates the performance index of each candidate based on the cost function

$$f = \omega_m C_m + \omega_{lat} C_{lat} + \omega_{lon} C_{lon} \tag{11}$$

which has the following three types of weighted costs: the maneuver cost C_m, the lateral cost C_{lat}, and the longitudinal cost C_{lon} with their own weight ω.

A behavior planner gives the desired maneuver a trajectory planner considering a long-term objective of autonomous driving. The trajectory planner should obey the maneuver, and generate a proper trajectory. To perform this, the maneuver cost C_m is a key element. For the simplification, it is assumed that the best maneuver m_d among lane keeping, overtaking, and lane changing is predetermined by the behavior planner. To prefer the trajectory candidate with the desired maneuver m_d, the maneuver cost C_m is calculated as follows:

$$C_m = \begin{cases} C_d^0 & for \quad m_d = m_i \\ C_p^0 & for \quad m_d \neq m_i \quad \text{AND} \quad m_i = m_p \end{cases} \tag{12}$$

where the i^{th} candidate's maneuver m_i is one of the predesigned maneuvers m_p containing lane keeping ($p = k$), overtaking ($p = o$), and lane changing ($p = c$). C_d^0, C_k^0, C_o^0, and C_c^0 are configuration parameters to make a preference for the specific maneuver. These parameters are configured as

$$C_d^0 \ll C_k^0 < C_c^0 < C_o^0. \tag{13}$$

Here, C_d^0 is set as the lowest value because it gives the trajectory planner's preference to the trajectory with the desired maneuver. The remainders indicate the maneuver priority for an alternative solution when there is no feasible trajectory with the desired maneuver. Based on this order, the planner tries to select an alternative trajectory in the sequence of lane keeping, lane changing, and overtaking.

The lateral cost C_{lat} is established by following three lateral elements: as the convergence distance, convergence time, and lateral ride comfort.

$$C_{lat} = \alpha_n (\Delta n)^2 + \alpha_T T_i + \alpha_j \int_0^T \frac{1}{T} \left(\frac{d^3}{dt^3} n(t) \right)^2 dt \tag{14}$$

where α_n, α_T, and α_j are the related weights. The convergence distance Δn is the difference between the target lateral position n_t and the lateral terminal position n_T of the candidate. If Δn is zero, the candidate converges the target lane in the convergence time (or terminal time) T_i. The shorter the convergence distance Δn and convergence time T_i are, the higher is the convergence speed. However, it degrades the ride comfort. Based on the trade-off, this cost helps the planner select the optimal trajectory from the perspective of lateral movement.

Similar to the lateral cost C_{lat}, the longitudinal cost C_{lon} consists of two terms of speed error and ride comfort in s direction, as follows:

$$C_{lon} = \beta_s \sum_{i=1}^{N} \frac{1}{N} (v_i - v_r)^2 + \beta_j \sum_{i=1}^{N} \frac{1}{N} \left(\frac{d^3}{dt^3} s(t_i) \right)^2 \tag{15}$$

where β_s and β_j are the weights of the average difference between the planned speed v_i and the reference speed v_r and the average integral jerk, respectively. The speed difference measures how close the candidate follows the reference speed and the other term evaluates the ride comfort in the longitudinal direction. With these, the cost C_{lon} makes the planer

choose the optimal candidate with a lower speed error and better ride comfort.

In summary, the proposed planner selects the optimal trajectory with a comprehensive consideration of the maneuver, lateral movement, and longitudinal movement. An effect of each element can be configured by the related weights ω_m, ω_{lat}, and ω_{lon}.

IV. Simulation

The proposed algorithm of the hybrid trajectory planner was implemented in C/C++ environment. For an efficient calculation of the QP problem for the longitudinal movement, we used a CVXGEN [34] that generates the embedded C code to solve QP-representable convex optimization problems. All computations run on a embedded PC with i5 Intel core. Every 100 ms, the planner generates an updated trajectory with keeping the continuity [35] with the previous trajectory produced at the last time. The resultant trajectory was represented as 31 nodes with 3 s prediction time and 0.1 s resolution.

To evaluate the performance of the proposed method, we prepared two types of dynamic scenarios: lane change in a traffic flow and backup trajectory generation to abort a lane change maneuver. Furthermore, these tests were performed on the road map modeled by the public roads to validate the generality of the trajectory planning on various road geometry.

A. Simulation I: Lane Change in a Traffic Flow

The dynamic scenarios indicate the continuously changing situations over time by moving traffic participants such as cars, bikes, bicycles, and pedestrians. In the challenging situations, a trajectory planner for autonomous driving requires various driving maneuvers including changing a lane, merging and splitting a lane, and passing through a roundabout. However, the lane change is a fundamental maneuver among the others because the planner can conduct the other maneuvers based on the lane-changing trajectory generation with some modification. For this reason, this section evaluates the performance of the proposed method for *lane change in a traffic flow*, as shown in Fig. 1.

This scenario exhibits a structured road with two lanes and four surrounding vehicles, as described in Fig. 11. The black and red boxes refer to an ego and surrounding vehicles, respectively. The dotted lines indicate the center line of each lane. Furthermore, the blue dotted line is a desired lane that should be determined by behavior planning in real time. However, to focus the performance of the hybrid trajectory planning module, this scenario predefined the desired lane before the simulation. In this figure, each snap shot illustrates a generated trajectory by the proposed method during lane changing.

At the first time (32th frame), the ego vehicle was surrounded by four vehicles in two lanes. The desired lane was the same as the current lane on which the ego vehicle drove. In this situation, the proposed planner generated eight trajectory candidates for keeping and changing a lane. Among these, the planner selected the lane keeping trajectory

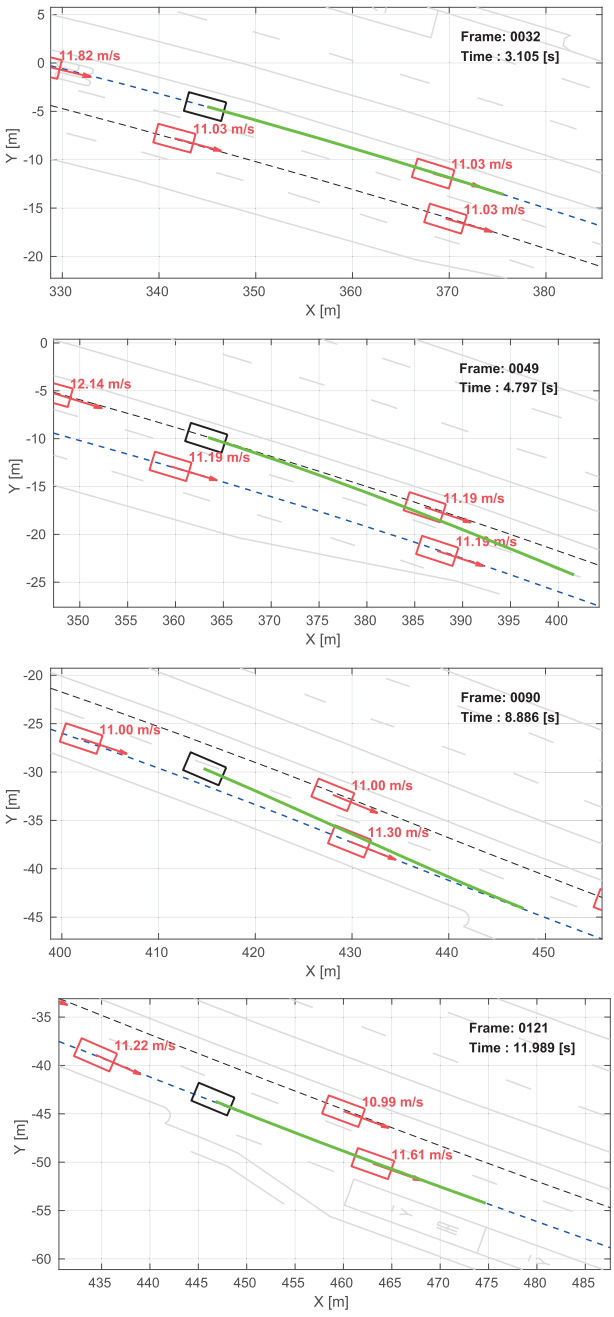

Fig. 11. Hybrid planner succeeded to generate a safe trajectory during changing a lane in a dense traffic because the longitudinal optimization helped the ego vehicle accelerate the speed when the lane change starts.

following a preceding vehicle that was generated by the Target Following (TF) strategy.

At the 49th frame, the desired lane was changed to the right lane. This meant a lane change request by the behavior planner. Based on the request, the proposed planner optimized a safe trajectory for changing a lane without any collision. The green line represents the resultant path generated by the proposed method. The longitudinal trajectory is illustrated in Fig. 12. The red lines indicate the boundary conditions of position, speed, and acceleration for longitudinal trajectory optimization. In particular, the area surrounded by the boundaries in s position refer the *safety corridor* described in Section V.B.

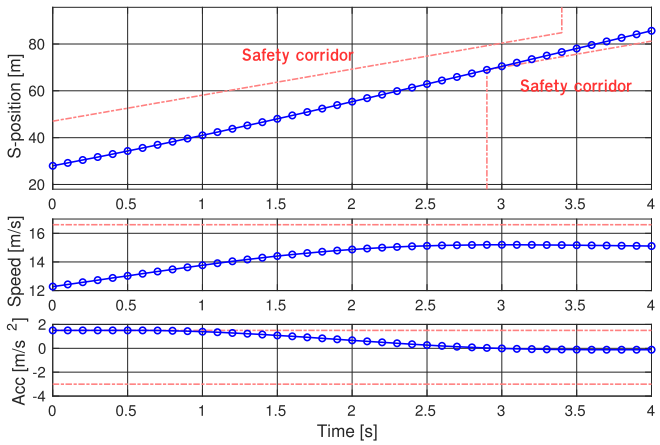

Fig. 12. This figure shows the optimization result of the longitudinal movement within the safety corridor, speed limit, and acceleration limit.

Within these boundaries, the proposed method succeeded in generating a safe longitudinal movement, as shown by the blue lines. If we used the sampling method in this case, countless candidates were required for the dynamically changed environment. However, owing to the optimization scheme, the planner can generate flexible longitudinal movements to an driving environment's change without the effort to tune the sampling patterns suitable for each scenario. This plays an essential role in overcoming the dynamic driving scenarios.

Through the same process, the planner generated a safe trajectory for changing a lane in each step. Furthermore, the planner kept a safe distance from the preceding vehicle after the lane change. Eventually, the proposed planner led the ego vehicle to successfully make a safe and smooth lane change in a time-critical scenario.

B. Simulation II: Backup Trajectory Generation to Abort a Lane-Change Maneuver

The lane change in dynamic street situations is a challenging task due to dimensional complexity of a time- and space-coupled problem and unpredictable behaviors of traffic participants. Since the autonomous vehicle occupies the space and has physical limits to accelerate and decelerate, the lane change is not always a possible operation in all situations. For this reason, the trajectory planner should select a proper maneuver and generate a safe trajectory even if the behavior planner gives an improper request. In other words, the autonomous vehicle should be able to build a backup trajectory to abort a lane-changing maneuver for safe driving.

This scenario evaluated the performance of the proposed planner to generate a backup trajectory in this conflicting scenario. The backup trajectories and ego's positions are represented in Fig. 13. Since the preceding vehicle was very slow in a traffic flow, the behavior planner changed the desired lane into the left lane at the 48th frame. At this time, the ego vehicle tried to change a lane like the green trajectory. However, the left-rear vehicle accelerated when the ego's lane change started. If the ego kept the lane-changing maneuver despite the change in the environment, a collision with the vehicle was inevitable.

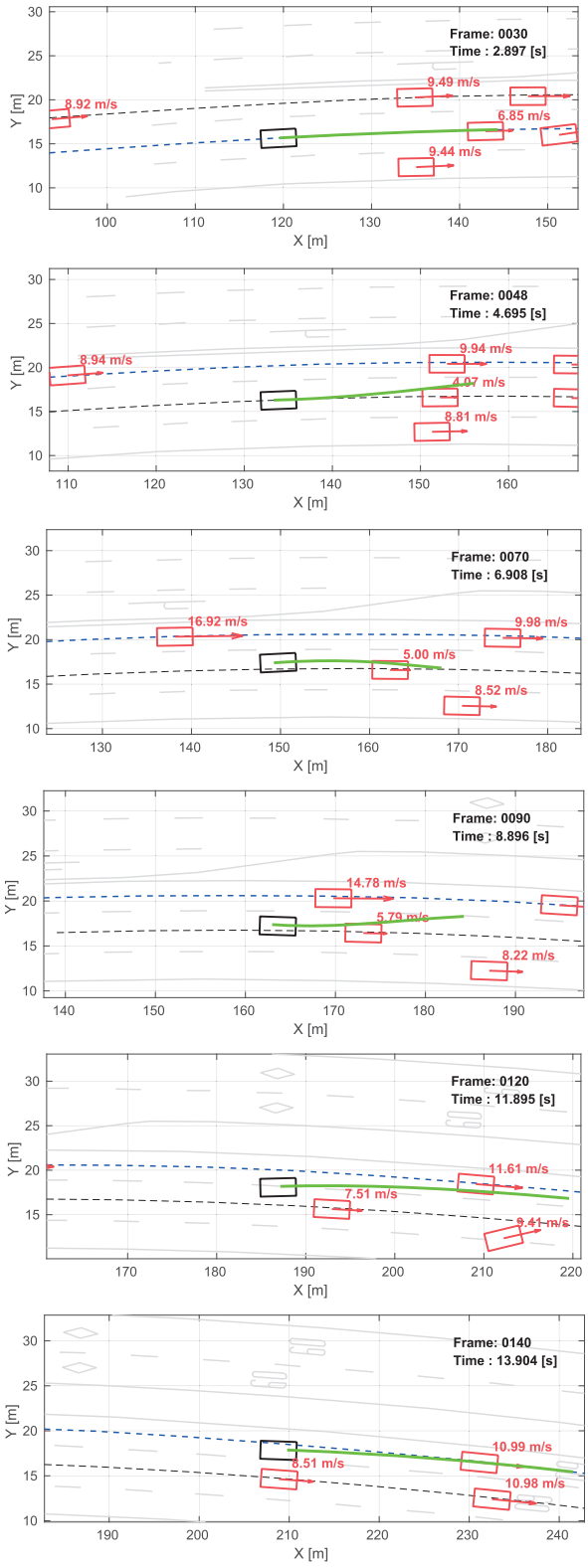

Fig. 13. Based on the sampling method, the hybrid method had an ability to generate a backup trajectory if the planner failed to find a safe lane-change trajectory during changing a lane.

To avoid the collision with the left-rear vehicle, the proposed planner built a backup trajectory to abort a lane-change maneuver at the 70th frame. The hybrid planner generated various trajectory candidates according to different lateral maneuvers

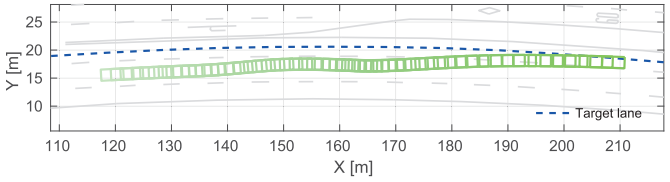

Fig. 14. Swerve motion trajectory to avoid the collision with the unexpected vehicle during lane change.

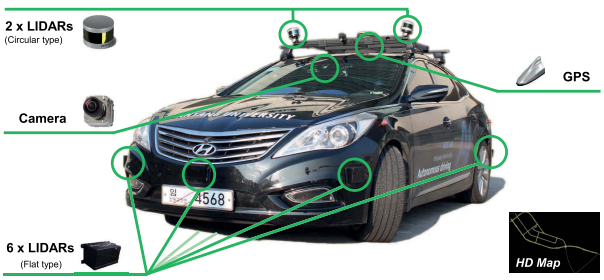

Fig. 15. Test vehicle A1 was equipped with camera, LIDARs, GPS, and HD map for evaluating the trajectory planning.

by using the sampling scheme. Although each candidate was optimized for no collision in the longitudinal direction, all trajectories were not drivable due to the ego's physical limits. After pruning the infeasible candidates, the planner selected the best trajectory without any collision. Based on the characteristics of the sampling scheme, the proposed method can provide multiple maneuver trajectories to overcome the mistaken decision of the behavior planner.

After the side vehicle passed through the ego and the planner checked the possibility of lane change to the left lane, it again attempted to perform the lane change at the 90th frame. The snap shot at the 120th frame illustrated that the ego kept performing lane change with avoiding the slow preceding vehicle. Finally, the lane change was successfully performed at the 140th frame. The whole trajectory in the unexpected situation is described as a swerve motion in Fig. 14.

V. Experimental Evaluation

The simulation results showed the feasibility of the proposed method in dynamic scenarios. Based on the hybrid strategy, this planner can effectively handle both the flexibility in the environmental change and the capability of multiple maneuvering trajectories with the optimization method for the longitudinal movement and the sampling method for the lateral movement.

In this section, we verified its ability in a real driving environment. For the environment, we used the autonomous car *A*1 [36] of Hanyang University in Fig. 15. This vehicle was equipped with LIDARs, cameras, GPS, and a High Definition (HD) map. These sensors provided fundamental information for the *planning space* which was described in Section II.A. The LIDARs and camera detected the *obstacles* and recognized their state and classification such as a car, bike, bicycle, and pedestrian. To perceive the precise states of the obstacles in 360 degree, we used synchronized scan data from six flat-type LIDARs placed on the front and rear bumpers and two circular-type LIDARs placed on the roof of the vehicle. Furthermore, the point cloud data from the LIDARs was utilized for precise position estimation with GPS and the HD map. Through the system, we obtained a robust estimation of the ego position in various driving environments such as highway and urban roads. The position and map provided road information including road network, road geometry, lanes, lane boundaries, traffic signs, and traffic lights, which were essential information for generating the *route* and *TCD* for autonomous driving on structured roads.

The proposed algorithm was implemented in the same computing environment as the simulation. Furthermore, all parameters for the planner were set as the identical ones used in the simulation. To trace the generated trajectory, the system had a trajectory tracking controller that controlled the steering angle and acceleration based on Model Predictive Control. For a rapid reaction, this module was executed at 100 Hz.

A. Scenario I: Lane Change

This was a basic scenario for the planner to overcome in a street road. An ego vehicle and side vehicle on the right lane drove side by side, as illustrated in Fig. 16. Since both speeds were similar, the ego vehicle should accelerate or decelerate for lane change to the right lane in this situation. This scenario was prepared to test the coping ability of the proposed method.

In this scenario, the desired speed was set as 30 km/h (8.33 m/s). At the 30th frame, the target lane (blue dotted line) was same as the ego's driving lane. For this reason, the planner generated a lane-keeping trajectory with cruising the desired speed. When the lane-change request was engaged at the 50th frame, the planner updated the trajectory for lane change to the target lane. The terminal state of lateral movement was moved to the target lane. Based on the lateral trajectory, the safe longitudinal trajectory was optimized within the *safe corridor*. In other words, the numerical optimization of the proposed method provided an efficient solution to find a safe trajectory in a challenging situation conflicting the surrounding vehicle. As a result, the planner generated the lane change trajectory with acceleration to avoid collision with the side vehicle. After passing the obstacle, the planner continued to the lane change trajectory, which contained the deceleration speed profile toward the desired speed. Following the trajectory, the ego vehicle completed the lane change without collisions at the 90th frame.

B. Scenario II: Lane Change in Traffic Flow

This scenario is similar to scenario I; however, three surrounding vehicles were added to a traffic flow. Unlike the previous scenario, an ego vehicle should keep a safe distance from the preceding vehicle. Figure 17 shows the process of lane change in a traffic flow. In spite of noisy environment in real driving conditions unlike the simulation, the planner provided safe trajectories for the lane change in a complex situation. In particular, the planner generated an evasive trajectory between the preceding and right-side vehicles to conduct

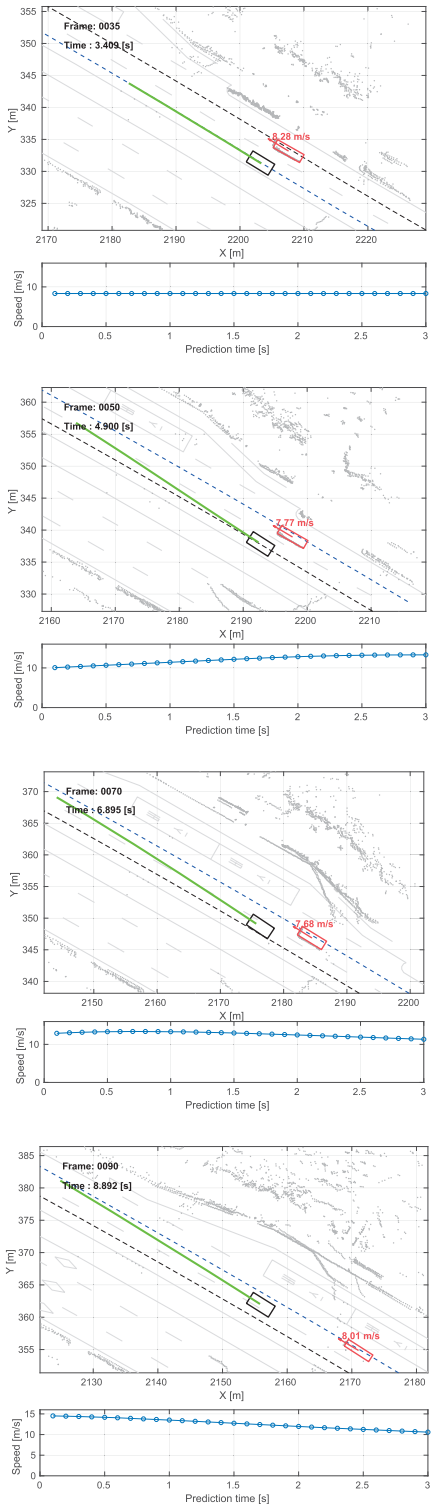

Fig. 16. The proposed planner led the ego vehicle perform a safe lane change, which contained acceleration for overtaking the side vehicle at the 50th frame and deceleration for maintaining the desired speed at 90th frame.

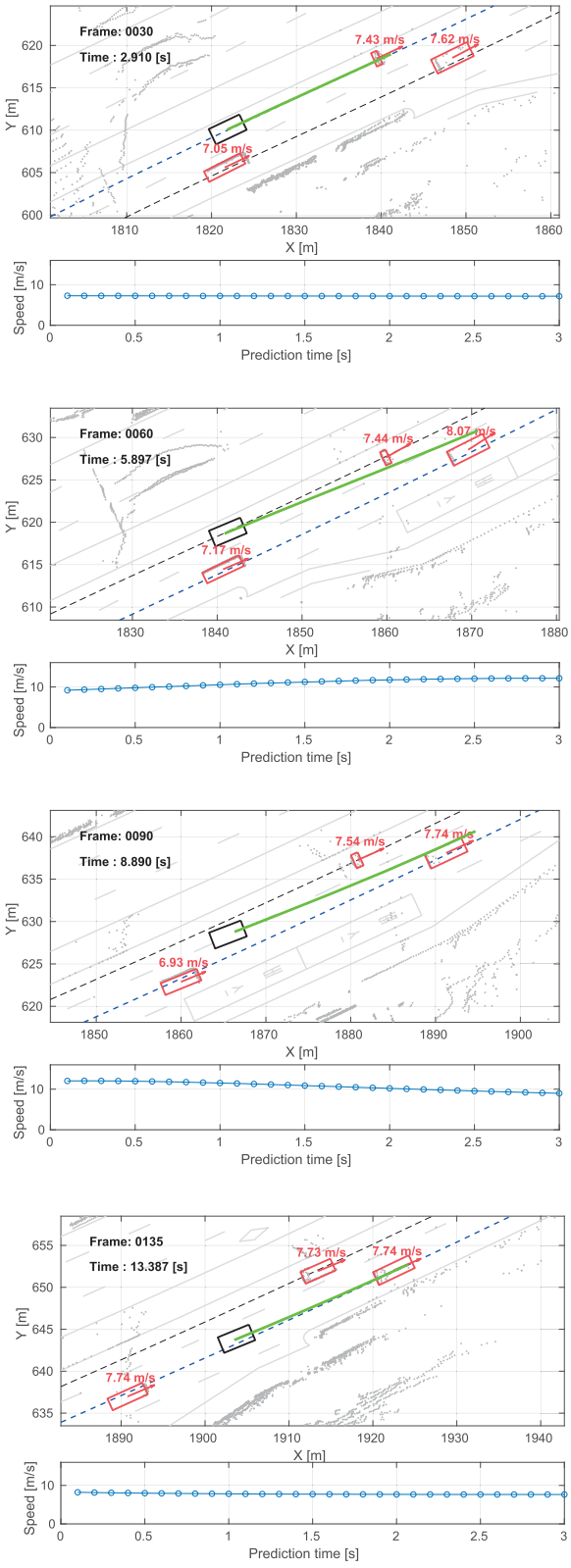

Fig. 17. The proposed planner generated safe lane-change trajectories in dense traffic flow based on the integration of the sampling and optimization methods.

the lane change toward the right lane. This was a difficult situation to overcome with a sampling-based planner alone.

C. Scenario III: Backup Trajectory Generation for Safe Driving

As described in simulation II, backup trajectory generation is an essential ability for safe lane change in dynamically changed environments. To focus on the performance of the backup solution, the scenario was simplified. It had one conflicting vehicle (CV) on the right lane of the ego's lane.

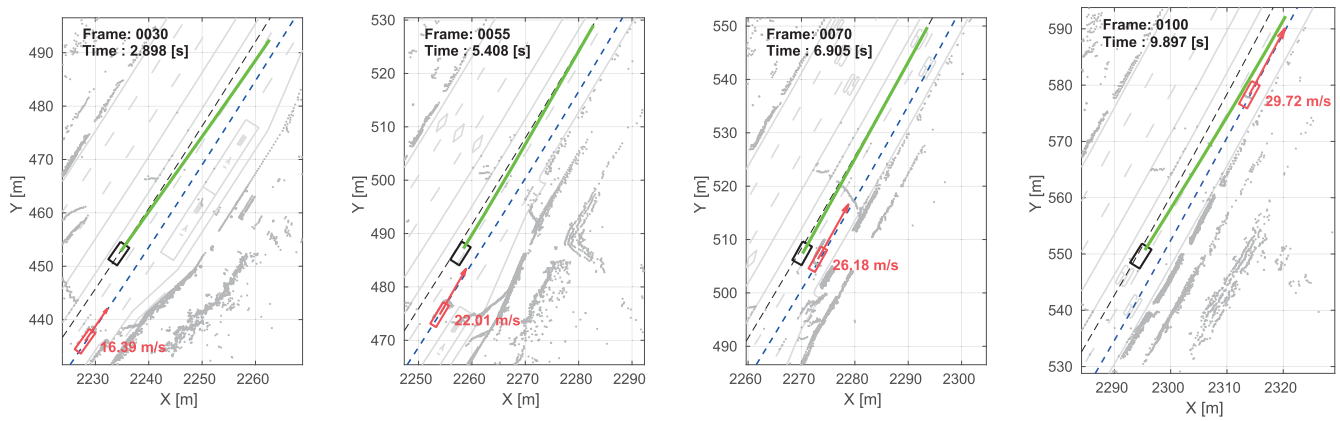

Fig. 18. With the backup trajectory generation, the planner succeeded to overcome the unexpected situation due to the rapid acceleration of the rear-right vehicle.

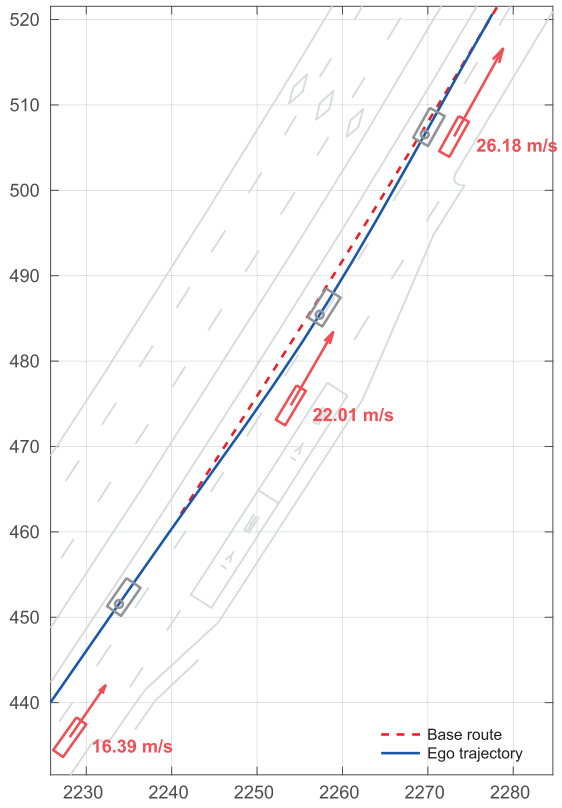

Fig. 19. The ego's movements during the period of backup trajectory generation from the 30th frame to the 70th frame.

Before the ego vehicle started to move to the right lane, the CV drove far away from the ego vehicle. Since the CV's speed was similar to the ego's speed, there was no collision risk if both speeds were not changed. However, the CV increased its speed as soon as the ego's turn signal to the right direction turned on. This was a realistic scenario faced in a public road.

Figure 18 shows the sequence scenes during the lane change operation. At the 30th frame, the side and ego vehicles' speeds were similar around 60 km/h (16.6 m/s). the planner decided to perform a lane change to the right lane because the collision risk was very low. At that time, the CV speed up to 26 m/s. The planner generated a backup trajectory to abort the lane change due to unexpected acceleration of the dangerous vehicle at the 55th frame. The planner waited until the dangerous vehicle passed. When the space and time gap were enough for the ego vehicle to conduct a lane change, the planner regenerated the lane change trajectory at the 100th frame. The ego's trajectory during the backup period is illustrated as a blue line in Fig. 19, where their positions and speeds at the 30th, 55th, and 100th frame were marked, respectively.

The proposed method can consider various possibilities of the movement with multiple maneuvering trajectories, which was realized by the lateral sampling scheme of the proposed method. Among the various maneuvering trajectories, the planner selected the best one considering safety, comfort, and behavior preferences. This ability assigns a key role to the trajectory planner to generate a safe trajectory within the performance limitation of the behavior planner.

VI. Conclusion

This paper proposes a hybrid trajectory planning for autonomous driving in dynamic street scenarios. These challenging scenarios require the planner's capability of adaptation in various situations and agile reaction in dynamically changing environments. To overcome the difficulties, this study combines a sampling and a numerical optimization approach. With the maneuver-based sampling method, the planner generates multiple lateral movements based on a quintic polynomial. Their longitudinal movements are optimized by the MPC scheme. At the final step, the planner chooses the optimal trajectory among the candidates through the cost-based evaluation.

The proposed method maximizes the complementary effects of these two methods for trajectory planning in highly dynamic situations. The sampling scheme improves the planner's ability to cover multiple maneuvers in complex situations. Based on this, the planner can generate a promptly reactive motion according to the change in situations. The optimization scheme helps the planner to find an optimal trajectory without the predefined sampling patterns. It enhances the ability of adaptation to the various driving situations that an autonomous vehicle faces.

To evaluate the performance of the hybrid method, we provided the test results in simulation and real-driving environments. The simulation results demonstrated that the system can overcome the dynamic street scenarios in an ideal environment. Based on this, the experimental results represented the system's real-time performance in a real driving environment. In all dynamic scenarios, the proposed method succeeded in generating a safe and comfortable trajectory in real time. This proved the capability of the proposed method in dynamic street scenarios.

In this study, we assumed that a proper maneuver was provided from the behavior planning layer. However, how to select the proper maneuver or human-like maneuver is another challenging problem to generate a safe and efficient trajectory in urban scenarios. In the future, we will apply the deep learning technique to determine human-like behavior decision making and improve the planner's performance.

REFERENCES

[1] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Berlin, Germany: Springer-Verlag, 2009.

[2] J. Ziegler *et al.*, "Making bertha drive—An autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Apr. 2014.

[3] M. Aeberhard *et al.*, "Experience, results and lessons learned from automated driving on Germany's highways," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 1, pp. 42–57, Jan. 2015.

[4] *Tesla Autopilot*. Accessed: Oct. 15, 2018. [Online]. Available: https://www.tesla.com/autopilot

[5] *Waymo*. Accessed: Oct. 15, 2018. [Online]. Available: https://waymo.com/

[6] S. M. LaValle *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[7] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[8] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 416–442, Nov. 2015.

[9] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.

[10] S. Thrun *et al.*, "Stanley: The robot that won the DARPA grand challenge," *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, Sep. 2006.

[11] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1599–1616, Dec. 2012.

[12] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *Int. J. Robot. Res.*, vol. 31, no. 3, pp. 346–359, Mar. 2012.

[13] J. Kim, K. Jo, K. Chu, and M. Sunwoo, "Road-model-based and graph-structure-based hierarchical path-planning approach for autonomous vehicles," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 228, no. 8, pp. 909–928, Jul. 2014.

[14] J. Kim, K. Jo, W. Lim, and M. Sunwoo, "A probabilistic optimization approach for motion planning of autonomous vehicles," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 232, no. 5, pp. 632–650, Apr. 2018.

[15] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for bertha—A local, continuous method," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, no. 4, pp. 450–457.

[16] B. Gutjahr, L. Groll, and M. Werling, "Lateral vehicle trajectory optimization using constrained linear time-varying MPC," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1586–1595, Jun. 2017.

[17] J. Nilsson, M. Brännström, J. Fredriksson, and E. Coelingh, "Longitudinal and lateral control for automated yielding maneuvers," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1404–1414, May 2016.

[18] Y. Zhang, H. Chen, S. L. Waslander, J. Gong, G. Xiong, T. Yang, and K. Liu, "Hybrid trajectory planning for autonomous driving in highly constrained environments," *IEEE Access*, vol. 6, pp. 32800–32819, 2018.

[19] F. Kunz and K. Dietmayer, "Hybrid discrete-parametric optimization for trajectory planning in on-road driving scenarios," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 802–807.

[20] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 613–626, Feb. 2018.

[21] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 763–770. [Online]. Available: http://ieeexplore.ieee.org/document/6629559/

[22] D. Ferguson, M. Darms, C. Urmson, and S. Kolski, "Detection, prediction, and avoidance of dynamic obstacles in urban environments," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 1149–1154. [Online]. Available: http://ieeexplore.ieee.org/document/4621214/

[23] K. Jo and M. Sunwoo, "Generation of a precise roadway map for autonomous cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 925–937, Jun. 2014.

[24] T. Fraichard and C. Laugier, "Path-velocity decomposition revisited and applied to dynamic trajectory planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1993, pp. 40–45.

[25] A. Takahashi, T. Hongo, Y. Ninomiya, and G. Sugimoto, "Local path planning and motion control for AGV in positioning," in *Proc. IEEE/RSJ Int. Workshop Intell. Robots Syst. (IROS)*, Sep. 1989, pp. 392–397.

[26] R. Rajamani, *Vehicle Dynamics and Control*. Boston, MA, USA: Springer, 2012.

[27] S. Li, K. Li, R. Rajamani, and J. Wang, "Model predictive multi-objective vehicular adaptive cruise control," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 3, pp. 556–566, May 2011.

[28] S. Lefevre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech J.*, vol. 1, p. 1, Jul. 2014.

[29] M. K. Kozlov, S. P. Tarasov, and L. G. Khachiyan, "The polynomial solvability of convex quadratic programming," *USSR Comput. Math. Math. Phys.*, vol. 20, no. 5, pp. 223–228, 1980.

[30] J. Mattingley and S. Boyd, "Real-time convex optimization in signal processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 50–61, May 2010.

[31] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer-Verlag, 2006.

[32] M. Hazewinkel, *Encyclopaedia Mathematics*. Amsterdam, The Netherlands: Springer, 2002.

[33] J. B. Rosen, "The gradient projection method for nonlinear programming. Part I. Linear constraints," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 1, pp. 181–217, 1960.

[34] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization," *Optim. Eng.*, vol. 13, no. 1, pp. 1–27, 2012.

[35] N. Nagasaka and M. Harada, "Towards safe, smooth, and stable path planning for on-road autonomous driving under uncertainty," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*. Nov. 2016, pp. 795–801.

[36] 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.

Wonteak Lim (S'14–M'19) received the B.S. degree in mechanical engineering from Hanyang University in 2013, and the Ph.D. degree in automotive engineering from Hanyang University, Seoul, South Korea, in 2019. Since 2013, he has been conducting autonomous driving research in the Automotive Control and Electronic Laboratory (ACE Lab), Department of Automotive Engineering, Hanyang University. His research interests include decision making, trajectory planning, vehicle control, and real-time embedded systems for highly automated driving.

Seongjin Lee (S'16) received the B.S. degree in automotive engineering from Hanyang University, Seoul, South Korea, in 2015. Since 2015, he has been with the Automotive Control and Electronics Laboratory (ACE Lab), Department of Automotive Engineering, Hanyang University, conducting research on the implementation of autonomous cars. His work focuses on behavioral decisions, trajectory planning, and vehicle motion control for highly automated vehicles.

Myoungho Sunwoo (M'81) received the B.S. degree in electrical engineering from Hanyang University in 1979, the M.S. degree in electrical engineering from the University of Texas at Austin in 1983, and the Ph.D. degree in system engineering from Oakland University in 1990.

He joined the General Motors Research (GMR) Laboratories, Warren, MI, USA, in 1985. He has worked in the area of automotive electronics and control for 30 years. During his nine-year tenure at GMR, he worked on the design and development of various electronic control systems for powertrains and chassis. Since 1993, he has led research activities as a Professor with the Department of Automotive Engineering, Hanyang University. His work has focused on automotive electronics and controls, such as modeling and control of internal combustion engines, design of automotive distributed real-time control systems, intelligent autonomous vehicles, and automotive education programs.

Kichun Jo (S'10–M'14) received the B.S. degree in mechanical engineering and the Ph.D. degree in automotive engineering from Hanyang University, Seoul, South Korea, in 2008 and 2014, respectively. From 2014 to 2015, he was with the ACE Laboratory, Department of Automotive Engineering, Hanyang University, doing research on system design and implementation of autonomous cars. From 2015 to 2018, he was with the Valeo Driving Assistance Research, Bobigny, France, working on the highly automated driving. He is currently an Assistant Professor with the Department of Smart Vehicle Engineering, Konkuk University, Seoul. His current research interests include localization and mapping, objects tracking, information fusion, vehicle state estimation, behavior planning, and vehicle motion control for highly automated vehicles.