



Trajectory optimization and state selection for urban automated driving

Keisuke Yoneda¹ · Toshiki Iida¹ · TaeHyon Kim¹ · Ryo Yanase¹ · Mohammad Aldibaja¹ · Naoki Suganuma¹

Received: 30 April 2018 / Accepted: 26 July 2018
© The Author(s) 2018

Abstract

The automated driving is an emerging technology in which a car performs recognition, decision making, and control. The decision-making system consists of route planning and trajectory planning. The route planning optimizes the shortest path to the destination like an automotive navigation system. According to static and dynamic obstacles around the vehicle, the trajectory planning generates lateral and longitudinal profiles for vehicle maneuver to drive the given path. This study is focused on the trajectory planning for vehicle maneuver in urban traffic scenes. This paper proposes a trajectory generation method that extends the existing method to generate more natural behavior with small acceleration and deceleration. This paper introduces an intermediate behavior to gradually switch from the velocity keeping to the distance keeping. The proposed method can generate smooth trajectory with small acceleration/deceleration. Numerical experiments show that the vehicle generates smooth behaviors according to surrounding vehicles.

Keywords Automated vehicle · Path planning · Frenet-frame

1 Introduction

Automated vehicle is recently developed in the field of automotive engineering, robotics, and artificial intelligence for the purpose of commercially available vehicle [1, 2]. The automated vehicle is expected to reduce traffic accidents, traffic jam and driver's burden. Nowadays, some driving assistant systems such as adaptive cruise control, lane keeping assist and emergency braking system, are available for expressway driving and emergency system.

To achieve urban automated driving, it is important to generate appropriate behaviors according to surrounding circumstances. The automated vehicles are generally equipped with various sensors including light detection and ranging (LiDAR), radar, camera, global navigation satellite system/inertial navigation system (GNSS/INS) as shown in Fig. 1a. These sensors data are processed to percept surrounding

environment, estimate a precise location, optimize trajectories and control the vehicle. To implement a robust vehicle system, one common approach is to achieve such perception and decision-making systems using a precise predefined map. These approaches generally refer to highly accurate landmark information (e.g., lane markings, intersections) from the predefined map based on the current vehicle pose as shown in Fig. 1b. This study is focused on a behavior planning for vehicle maneuver in urban traffic scenes.

The behavior planning in automated driving consists of route planning and trajectory planning. The route planning optimizes the minimum cost path from the current position to the destination. Generally, Dijkstra's and A* algorithms are able to find the shortest route in structured environments from digital map like an automotive navigation system. On the other hand, in unstructured environments, heuristic methods are able to optimize collision-free and smooth path in the free region [3–5]. For the obtained path, the trajectory planning is performed to optimize the minimum cost polynomial function [6–9]. In the trajectory planning, the 4th/5th order polynomial functions enable to generate continuous trajectories with the minimum acceleration/jerk. Primitive driving patterns such as distance keeping and velocity keeping, was described in [6]. The obtained behaviors were then evaluated qualitatively by numerical simulation. Although

This work was presented in part at the 2nd International Symposium on Swarm Behavior and Bio-Inspired Robotics, Kyoto, October 29–November 1, 2017.

✉ Keisuke Yoneda
k.yoneda@staff.kanazawa-u.ac.jp

¹ Kakuma-machi, Kanazawa, Ishikawa 920-1192, Japan

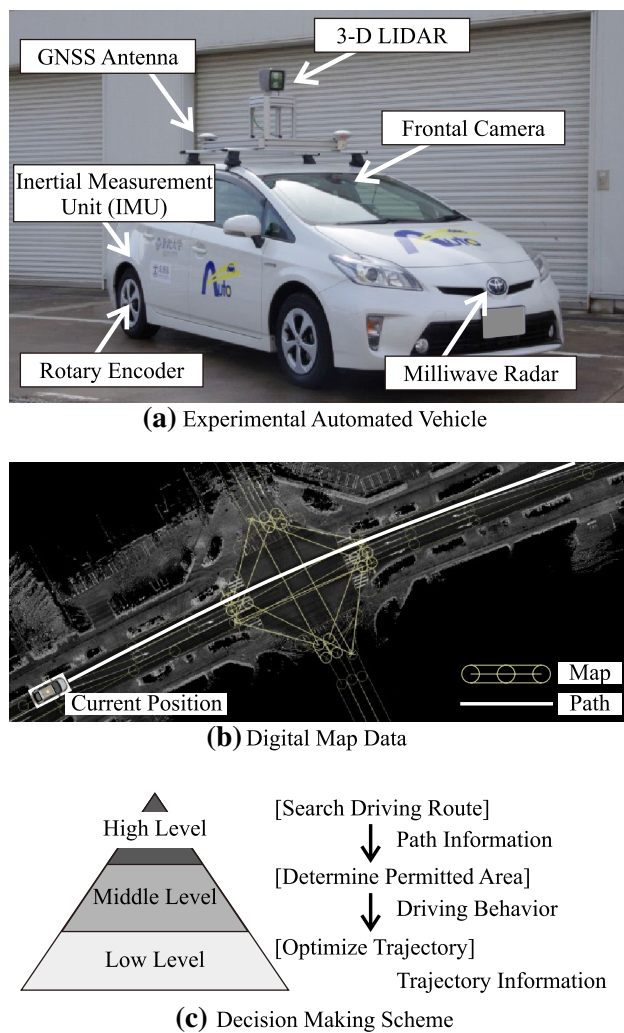


Fig. 1 Overview of experimental vehicle and decision making scheme. **a** Experimental automated vehicle, **b** digital map data, **c** decision making scheme

the primitive driving patterns could behave appropriately while smoothly avoiding obstacles in the surroundings, it was not a natural driving behavior because it was too aggressive. In other words, there is a problem that a large acceleration/deceleration occurs at the time of switching between the distance keeping and the velocity keeping. Driving with large acceleration/deceleration is an unnatural behavior. Therefore, this paper proposes a trajectory generation method that extends the existing method described in [6]. The proposed method can generate smooth trajectory with small acceleration/deceleration by introducing an intermediate behavior. Numerical experiments show that the vehicle generates smooth behaviors according to the surrounding vehicles.

The rest of this paper is composed as follows. Section 2 introduces the behavior model. Section 3 explains the proposed trajectory generation method. Section 4 describes

numerical experiments and Sect. 5 concludes with the obtained results.

2 Behavior model

2.1 Decision-making system

In our automated vehicle system, the decision-making system consists of multi-level of behavior planners as shown in Fig. 1c. It consists of a route navigation (high level), a driving permission planning according to traffic rules (middle level) and a trajectory planning (low level). The high-level planner has the role of the route planning using the digital map. The middle-level planner manages the traffic rules (e.g., velocity limit, permission of transiting intersection and passing lanes) for the obtained path. The low-level planner then generates a trajectory according to static and dynamic obstacles around the vehicle. Therefore, lateral and longitudinal profiles are optimized based on the obtained path from the high-level planner, and driving permissions from the middle-level planner.

2.2 Sensing data

The above-mentioned planners require sensing information about ego-vehicle and surrounding objects. GNSS/INS and the self-localization module provide precise position, orientation, velocity and acceleration for the ego-vehicle [10–12]. Surrounding objects are recognized using the onboard sensors such as LiDAR, radar and camera. For dynamic objects, information about position, orientation, rectangular size, velocity, and acceleration are estimated using LiDAR or radar (the size is only estimated by LiDAR). Static objects are estimated as an occupancy grid map by LiDAR.

3 Trajectory planning

3.1 Frenet-frame coordinate

Traveling path is defined by parametric curves such as Bezier curve and B-Spline curve. To generate trajectories along the shape of the given path, the Frenet-frame is commonly used in [6]. On the Frenet coordinate system, normal trajectory $d(t)$ and tangential trajectory $s(t)$ are defined for the given curve as shown in Fig. 2a. Let $d(t)$ and $s(t)$ be a offset pattern and a distance pattern, respectively. The offset pattern corresponds to the amount of lateral movement within the traveling lane. On the other hand, the distance pattern corresponds to the acceleration/deceleration profile. Therefore, the vehicle trajectory is formulated in the following equation.

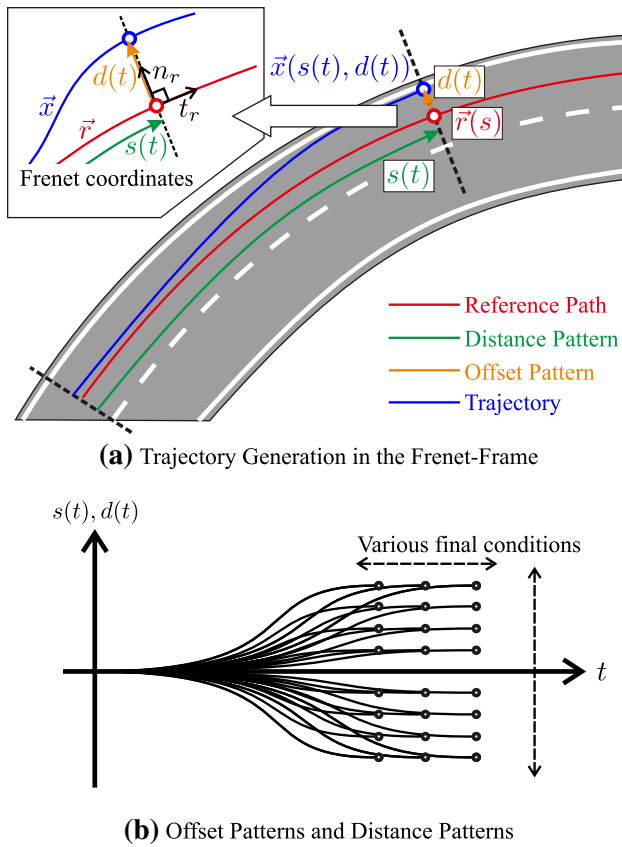


Fig. 2 Trajectory generation. **a** Trajectory generation in the Frenet-frame, **b** offset patterns and distance patterns

$$\vec{x}(s(t), d(t)) = \vec{r}(s(t)) + d(t)\vec{n}_r(s(t)). \quad (1)$$

In practical use, the optimum trajectory is searched while adjusting the various types of terminal condition for each pattern as shown in Fig. 2b.

3.2 Polynomial trajectory

The vehicle trajectory is a combination of an offset pattern $d(t)$ and a distance pattern $s(t)$. Both patterns are generally formulated as a 4th/5th order polynomial function. The 4th order function enables to generate the minimum acceleration trajectory. The 5th order function enables to generate the minimum jerk trajectory. The offset pattern $d(t)$ and the distance pattern $s(t)$ are defined as follows.

$$d(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5, \quad (2)$$

$$s(t) = b_0 + b_1t + b_2t^2 + b_3t^3 + b_4t^4 + b_5t^5, \quad (3)$$

where variable a_i and b_i are computed according to the initial conditions $[d_0, \dot{d}_0, \ddot{d}_0]/[s_0, \dot{s}_0, \ddot{s}_0]$ and terminal conditions $[d_1, \dot{d}_1, \ddot{d}_1, \Delta T_d]/[s_1, \dot{s}_1, \ddot{s}_1, \Delta T_s]$ at time

t_1 ($t_1 = t_0 + \Delta T_d$ or ΔT_s). Let ΔT_* be a preview time ($\Delta T_{\min} < \Delta T_* < \Delta T_{\max}$, $\Delta T_{\min} = 2.0[s]$, $\Delta T_{\max} = 6.0[s]$).

The offset pattern $d(t)$ is defined using a quintic function to control the lateral position precisely. The distance pattern $s(t)$ is defined using a quartic or quintic functions for velocity keeping and distance keeping, respectively. To control a velocity smoothly, the distance pattern is generated by solving a quartic function (in this case, $b_5 = 0.0$). On the other hand, to control a distance to the leading vehicle precisely, a quintic function provides the distance pattern.

3.3 Offset pattern generation

To generate collision-free trajectories for static and dynamic objects, trajectories of several terminal conditions are generated as candidates according to the lane with. The trajectory is defined between the initial condition $[d_0, \dot{d}_0, \ddot{d}_0]$ at time t_0 and the terminal condition $[d_1, \dot{d}_1, \ddot{d}_1, \Delta T_d]$ at time t_1 using a quintic function Eq. (2). Considering a comfortable driving, velocity and acceleration for lateral direction should be minimized. Therefore, the terminal condition is $[d_1, \dot{d}_1, \ddot{d}_1, \Delta T_d] = [\Delta d, 0, 0, \Delta T_d]$. Several patterns of Δd and ΔT_d are selected to generate candidate trajectories.

3.4 Distance pattern generation

The distance pattern has the role of velocity profile according to the surrounding objects. Distance keeping and velocity keeping behaviors are generated depending on the distance to the leading vehicle and so on. In [6], track mode, stop mode and cruise mode was proposed as primitive behavior patterns. The track and stop mode are basic distance keeping behaviors which precisely adjust vehicular gap to the leading vehicle and distance to the stop line using quintic functions in Eq. (3). In the distance keeping, it is necessary to consider continuity of position, velocity and acceleration. The initial condition $[s_0, \dot{s}_0, \ddot{s}_0]$ at time t_0 and terminal condition $[s_1 + \Delta s, \dot{s}_1, \ddot{s}_1, \Delta T_s]$ at time t_1 are then determined to generate trajectories. Let Δs be a small value of displacement around the terminal condition.

On the other hand, the cruise mode is defined as a velocity keeping behavior. In the velocity keeping, it is not necessary to adjust the terminal position precisely. Because it leads to quick acceleration and deceleration. Therefore, when there is no preceding vehicle nearly, the distance pattern is generated using a quartic function in Eq. (3) under $b_5 = 0.0$. The initial condition $[s_0, \dot{s}_0, \ddot{s}_0]$ at time t_0 and terminal condition $[\dot{s}_1 + \Delta \dot{s}, \ddot{s}_1, \Delta T_s]$ at time t_1 are then determined to generate trajectories.

Werlind et al. [6] demonstrated the collision-free maneuver by numerical simulation. However, the obtained behavior was not a natural driving behavior because a large acceleration and deceleration occurs at the time of switching from the velocity

3.4.2 Distance keeping: stop mode

To park the vehicle at the stop line or the destination, the stop model generates a deceleration profile as shown in Fig. 3b. s_{stop} is a distance to the parking place which is given from the middle-level planner.

$$\begin{cases} s_1 = s_{\text{stop}} \\ \dot{s}_1 = \ddot{s}_1 = 0.0. \end{cases} \quad (5)$$

3.4.3 Velocity keeping: **cruise mode**



The cruise mode keeps the ego-velocity to the given velocity \dot{s}_{tgt} as shown in Fig. 3c. \dot{s}_{tgt} is determined from the speed limit.

$$\begin{cases} \dot{s}_1 = \dot{s}_{\text{tgt}} \\ \ddot{s}_1 = (\dot{s}_1 - \dot{s}_0)/\Delta T_s \end{cases} \quad (6)$$

3.4.4 Velocity keeping: adjust mode

When the preceding vehicle is at a long distance, the adjust mode gradually reduces the vehicular gap to $D_{\text{des}}(t_1)$ as shown in Fig. 3d. It searches the maximum velocity \dot{s}_{adj} with the maximum preview time ΔT_s under $D_{\text{des}}(t_1) < D(t_1)$.

$$\begin{cases} \dot{s}_1 = \dot{s}_{\text{adj}} \\ \ddot{s}_1 = 0.0 \\ T_{\text{min}} = t_0 + \Delta T_{\text{min}} \\ T_{\text{max}} = t_0 + \Delta T_{\text{max}}. \end{cases} \quad (7)$$

3.5 Collision detection

For all generated trajectory set, it is determined whether or not each trajectory is drivable. For dynamic objects, it is confirmed the overlapping of bounding boxes at each time step as shown in Fig. 5a. When the rectangular boxes of the ego-vehicle and the moving object overlap, it is judged that a collision has occurred. In a similar way, a collision detection with static object is performed using the occupancy grid map. Each bounding box of the ego-trajectory is confirmed a collision with the obstacle grid cell of the occupancy grid map as shown in Fig. 5b.

3.6 Optimal trajectory selection

Collision-free trajectory set are generated based on the above-mentioned procedures. The minimum cost trajectory

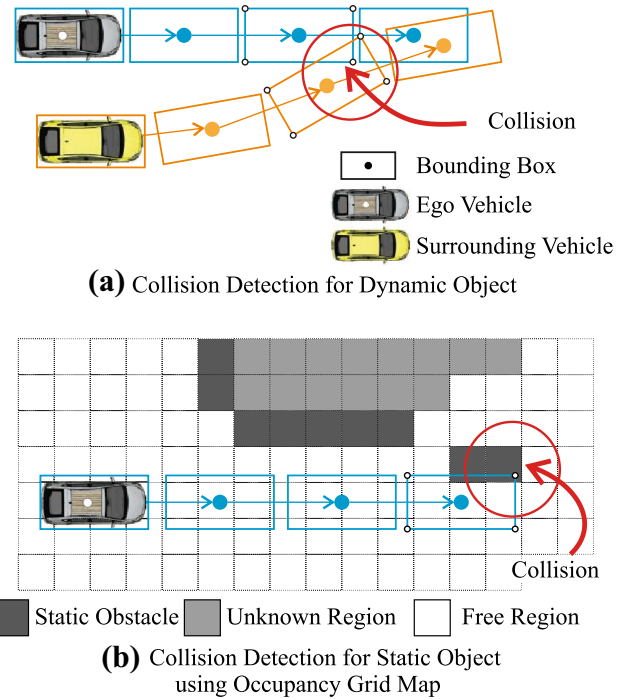


Fig. 5 Collision detection. **a** Collision detection for dynamic object, **b** collision detection for static object using occupancy grid map

is selected from these trajectories [13]. The cost function C is defined in Eqs. (8)–(10).

$$C = k_{\text{path}} C_{\text{path}}^i + k_d C_d + k_s C_s, \quad (8)$$

$$C_d = k_j^d \int_{t_0}^{t_1} \ddot{d}^2(t) dt + k_x^d \Delta T_d + k_y^d \Delta d^2, \quad (9)$$

$$C_s = \begin{cases} k_j^s \int_{t_0}^{t_1} \ddot{s}^2(t) dt + k_x^s \Delta T_s + k_y^s \Delta s^2 & (\text{if distance keeping}) \\ k_j^s \int_{t_0}^{t_1} \ddot{s}^2(t) dt + k_x^s \Delta T_s + k_y^s \Delta \dot{s}^2 & (\text{if velocity keeping}), \end{cases} \quad (10)$$

where, C_d and C_s are costs about the offset and the distance patterns, respectively. Both terms evaluate integral of jerks, preview times and displacements of terminal condition. C_{path}^i is a weight of drivable lanes for the i -th lane. C_{path}^i is determined from the middle-level planner to enable to change lanes for a multiple-lane road. In case of $C_{\text{path}}^0 < C_{\text{path}}^1$, the trajectory traveling in 0-th lane tends to be selected as shown in Fig. 6. On the other hand, the trajectory to change the lane is likely to be selected in case of $C_{\text{path}}^0 > C_{\text{path}}^1$.

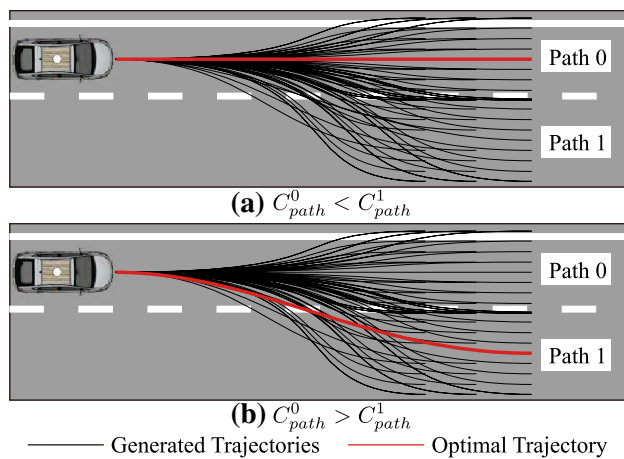


Fig. 6 Multiple lane driving

4 Simulation and evaluation

4.1 Conditions

Two types of experiments are carried out to evaluate the effectiveness of the proposed method. The first experiment quantitatively evaluates the maximum value of acceleration/deceleration depending on whether or not the proposed intermediate mode (adjust mode) is introduced. As one of the scenes where large deceleration occurs, the scene approaching the stopped vehicle while traveling at the maximum speed of the general road (60 km/h) is evaluated as shown in Fig. 7a. The second evaluation qualitatively observes the behavior for typical driving scenes. The obtained behaviors are observed with typical driving maneuvers of following and overtaking scenarios as shown in Fig. 8a, b.

4.2 Results

Figure 7b, c indicate experimental results for the approaching driving. In the case where the adjust mode is not used, it can be confirmed that a large deceleration occurs immediately after switching from the velocity keeping to the distance keeping. In Fig. 7b, the maximum deceleration value was -3.94 m/s^2 . On the other hand, by introducing the adjust mode, it was confirmed that deceleration started from a few seconds before and the maximum deceleration value was suppressed to -1.71 m/s^2 as shown in Fig. 7c. Therefore, the results showed that the introduction of the proposed method is possible to reduce the maximum deceleration value to 43.4% of the conventional one.

Figure 8c, d shows experimental results for typical driving maneuvers. In the following scenario, the preceding vehicle starts to decelerate after 10 s and stops as shown in Fig. 8c. While the gap is increasing, the ego-vehicle drives the cruise mode to keep the target velocity. When the

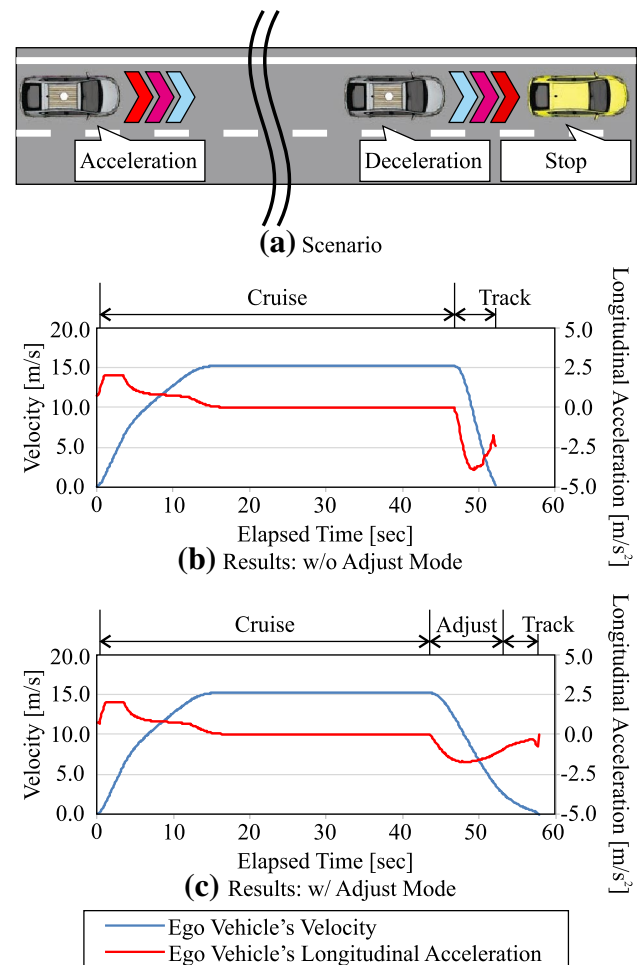


Fig. 7 Experimental results. **a** Scenario, **b** results: w/o adjust mode, **c** results: w/ adjust mode

preceding vehicle starts decelerating, the ego-vehicle gradually reduces the gap using the adjust mode. The track mode then accurately controls the gap. In a series of state transitions, it is observed that the ego-velocity and the gap are smoothly updating. On the other hand, the preceding vehicle parks by the side of the road in the overtaking scenario. As shown in Fig. 8d, the ego-vehicle smoothly controls the lateral offset and velocity during overtaking. These results show that the proposed planner controls the vehicle properly according to surroundings.

5 Conclusion

A method of trajectory optimization is proposed for vehicle maneuvers in urban road. Polynomial curves generate the minimum jerk curve in given conditions. According to the static and dynamic objects, the ego-vehicle switches motion pattern smoothly. The proposed method introduces

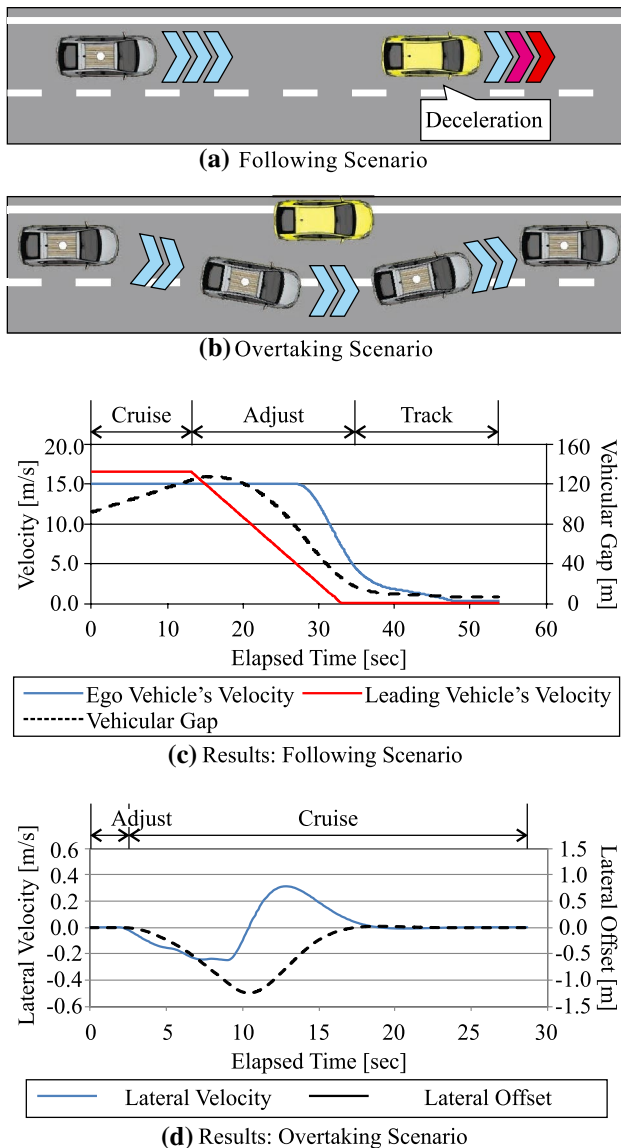


Fig. 8 Experimental results. **a** Following scenario, **b** overtaking scenario, **c** results: following scenario, **d** results: overtaking scenario

the intermediate behavior to smoothly switch from the velocity keeping behavior to the distance keeping behavior. Experimental results showed that by introducing the proposed method, it is possible to halve the deceleration value that occurs when the ego-vehicle is approaching a stationary vehicle.

Additionally, the proposed trajectory planner has been installed in our automated vehicle and conducted a self-driving test in real traffic scenes. Further quantitative evaluations in dense traffic scenes in actual automated driving are future works.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Franke U, Pfeiffer D, Rabe C, Knoeppel C, Enzweiler M, Stein F, Herrtwich RG (2013) Making Bertha see. Proceedings of ICCV workshop on computer vision for autonomous driving
2. Kato S, Takeuchi E, Ishiguro Y, Ninomiya Y, Takeda K, Hamada T (2015) An open approach to autonomous vehicles. *IEEE Micro* 35(6):60–69
3. Dolgov D, Thrun S, Montemerlo M, Diebel J (2008) Practical search techniques in path planning for autonomous driving. Proceedings of the first international symposium on search techniques in artificial intelligence and robotics
4. Kuwata Y, Fiore GA, Teo J, Frazzoli E, How JP (2008) Motion planning for urban driving using RRT. Proceedings of IEEE/RSJ international conference on intelligent robots and systems, pp 1681–1686
5. Huy Do Q, Tehrani H, Yoneda K, Ryohei S, Mita S (2013) Vehicle path planning with maximizing safe margin for driving using Lagrange multipliers. Proceedings of 2013 IEEE intelligent vehicles symposium, pp 171–176
6. Werlind M, Ziegler J, Kammel S, Thrun S (2010) Optimal trajectory generation for dynamic street scenarios in a Frenet frame. Proceedings of international conference on robotics and automation, pp 987–993
7. Ziegler J, Bender P, Dan T, Stiller C (2015) Trajectory planning for Bertha: a local, continuous method. Proceedings of 2013 IEEE intelligent vehicles symposium, pp 611–616
8. Huy Do Q, Mita S, Tehrani H, Egawa M, Muto K, Yoneda K (2017) Human drivers based active-passive model for automated lane change. *IEEE Intell Transp Syst Magaz* 9(1):42–56
9. Tehrani H, Shimizu M, Ogawa T (2013) Adaptive lane change and lane keeping for safe and comfortable driving. Proceedings of second international symposium on future active safety technology toward zero-traffic accident
10. Urmson C, Anhalt J, Bagnell D, Baker C, Bittner R, Clark MN, Dolan J, Duggins D, Galatali T, Geyer C et al (2008) Autonomous driving in urban environments: boss and the urban challenges. *J Field Robot* 25(8):425–466
11. Aldibaja M, Suganuma N, Yoneda K (2017) Robust intensity based localization method for autonomous driving on snow-wet road surface. *IEEE Trans Ind Inf* 13(5):2369–2378
12. Yoneda K, Yanase R, Aldibaja M, Suganuma N, Sato K (2018) Mono-camera based vehicle localization using Lidar intensity map. Proceedings of the 23rd international symposium on artificial life and robotics
13. Levinson J, Askeland J, Dolson J, et al (2011) Toward fully autonomous driving: systems and algorithms. Proceedings of 2011 IEEE intelligent vehicles symposium, pp 163–168