2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

Autonomous Predictive Driving for Blind Intersections

Yuki Yoshihara[1], Yoichi Morales[1], Naoki Akai[1], Eijiro Takeuchi[2] and Yoshiki Ninomiya [1]

***Abstract*— This paper presents a model for safe driving at blind intersections and its integration to a local planner based on a Frenet frame. The model predicts potential moving obstacles from blind intersections to proactively slow down to avoid potential collisions. The derivation of the model is described and its parameters are detailed. The local planner computes smooth trajectories with smooth velocity profiles so that the vehicle can follow the paths without jerk and sudden accelerations resulting in safe and comfortable navigation. Experimental results in simulation and in the real field with an autonomous car, show that the proposed predictive driving framework can reproduce human expert driver's trajectories and velocities when facing blind intersections.**

I. INTRODUCTION

Predicting potential hazardous situations is one of the main issues in global and local path plannings. Occluded pedestrians, bicyclists, or other cars may come out from blind intersections and corners. Many approaches exist which tried to enhance vehicles' sensing ability by installing long range sensors that can cover blind corners, or to utilize infrastructure sensors. However, in principal, it is not possible to eliminate occluded areas by any sensors. Also, it is not realistic to assume that all vehicles should have some networking devices inside, when communication itself could be unreliable. Therefore, the ego-vehicle has to somehow anticipate potential hazards and take decisions based on it.

Based on observations from expert driver data, it has been reported that they follow smooth trajectories without jerk and sudden accelerations [1]. When the expert driver faces a blind intersection, she/he anticipates potential moving obstacle coming out from the blind spots and smoothly slows down to avoid sudden braking and smoothly fully stop in case of necessary. If there are no moving objects coming the driver would accelerate and keep going. This is illustrated in Fig. 1. When the driver faces a blind intersection (left of Fig. 1), he anticipates potential moving objects coming from the blind spots and slows down (bottom right of Fig. 1). Once he has good visibility and confirms that there are no close moving objects he accelerates (top right of Fig. 1).

Previous studies on path planning have been mainly focused on visible potential dangers except some pioneering works [2] [3]. Until now the focus of driver assistant systems on urban roads centered in visible potential hazards. To the best of authors knowledge, there is no fully established path planner that can anticipate potential hazards. In this paper we propose a computational model of decision making for autonomous cars passing blind corner intersections, where hazards are uncertain and unpredictable. Our contributions in this paper are a model of speed limit for the ego-vehicle to safetly pass through a blind corner intersections (Section III), as well as a way to implement it into a path planner capable of computing smooth trajectories for real-sized car vehicles (Section IV). Performance is evaluated in simulations and experiments from two viewpoints: path computation and tracking capability (Section V) and proactive behavior which is compared towards expert human drivers (Section VI).

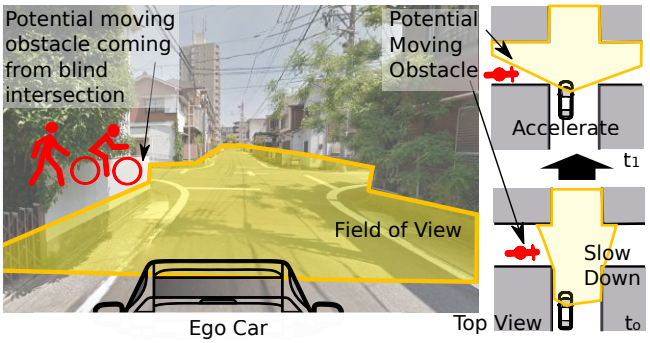

Fig. 1. Ego-vehicle facing a blind intersection. The vehicle anticipates potential moving obstacles coming from the blind area and slows down. When the visibility of the intersection improves the vehicle accelerates.

II. RELATED WORKS

Path planning is an active field in robotics and intelligent transportation systems research. From global path planning to the execution of the computed global path through local planners there is a large variety of planners [4].There are many obstacles and traffic participants in real road traffic, planners for real-sized autonomous cars must take into account risk factors simultaneously and/or hierarchically.

Risk assessment measures widely used are time to collision measures [5], due to its simplicity and low computational cost [6]. For multi obstacles, Monte-Carlo methods [7] has been proposed, which is based on random-sampling and model update. This results in a set of sampled points to form a predictive risk map [8], which can be extended for collision avoidance.However, it is not straightforward to apply them to a real-sized car, because of model uncertainty and high computational cost. Another approach, which we adopt, is more analytic; it tries to find explicit constraints where collisions can be avoided, such as a reachable region

*This work was supported by the Center of Innovation Program (Nagoya-COI) from Japan Science and Technology Agency.

[1]Yuki Yoshihara, Luis Yoichi Morales, Naoki Akai and Yoshiki Ninomiya are with the Institute of Innovation for Future Society (MIRAI), Nagoya University, Nagoya 464-8603, Japan {y-yuki, morales_yoichi, akai, ninomiya}@coi.nagoya-u.ac.jp

[2]Eijiro Takeuchi is with the Graduate School of Information Science, Nagoya University, Nagoya 464-8603, Japan takeuchi@coi.nagoya-u.ac.jp

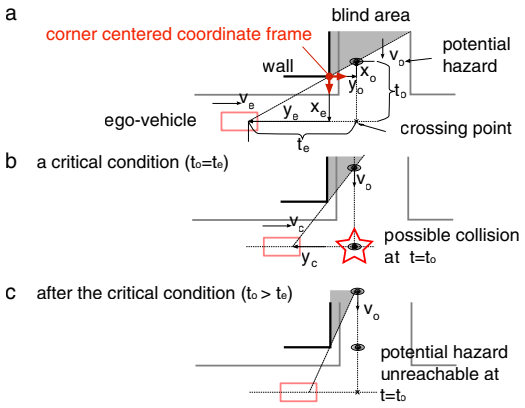

Fig. 2. A schematic description of the proposed model for computing a safe speed (9) to pass a blind intersection.

[9]. Although it needs assumptions on motion dynamics, the computational speed is high as it is explicitly represented as closed-form conditions. Risk uncertainty is handled by introducing the maximum value estimate. This is beneficial as we can avoid seeking for better model identification. Rather we need to estimate appropriate sets of small numbers of risk parameters.

Model predictive control (MPC) planners have been popular and used to follow given paths by satisfying given constraints [10] [11]. The work in [12] presents an approach to compute feasible paths and road planning. Those different planners succeeded in finding analytically optimal trajectory in terms of smoothness, safety, ride comfort [13], and even kindness to humans [14], which were generally applicable from real-sized cars to wheelchair-like robots [15].

On the other hand, more practical path planners were proposed, which generate several trajectories as 'candidates' and select one that makes some cost functionals optimal [16]. They are practical because they can avoid analytically solving optimization problems, as well as they leave a space to easily implement different constraints on demands. Among them, we chose to use a Frenet frame based path planner proposed by [17], as it incorporates several basic and important constraints for driving in public road. Simultaneous optimization of lateral and longitudinal motion differentiates this Frenet planner from other planners which explicitly separate lateral and longitudinal path design. With this framework, we will show our model for computing a safe speed for passing blind intersection would naturally be integrated, which will be described in next sections.

III. MODELING SAFE SPEED AT BLIND CORNERS

First, for the ego-vehicle to pass through a blind corner intersection safely, potential hazards must be modeled directly within an automated vehicle control framework. Potential hazards coming from blind sections are not known in advance and two physical attributes are crucial, i.e. a speed v_o and positions (x_o, y_o) of the unknown potential objects. Here, the subscription o denotes 'occluded', while e is the ego-vehicle. Given the ego vehicle positions as (x_e, y_e), geometric

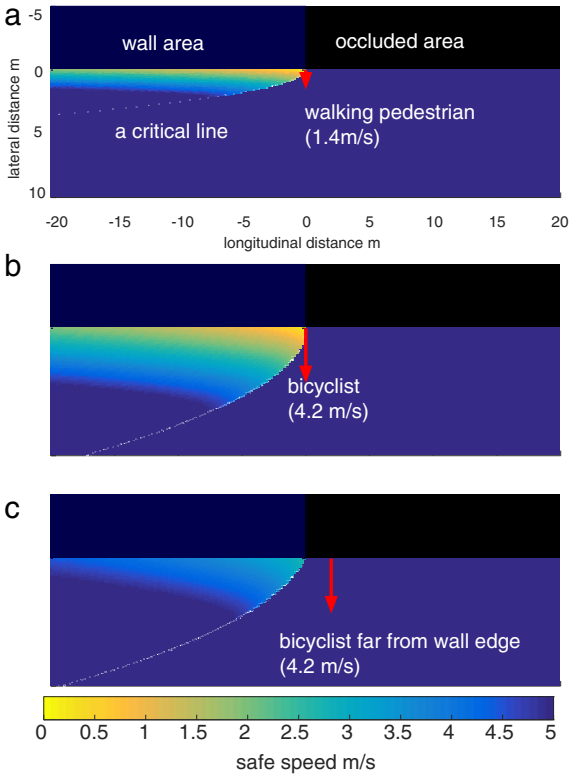

Fig. 3. Three different potential hazards: (a) a pedestrian coming from the wall edge, (b) a bicyclist coming from the wall edge, and (c) a bicyclist from 2 meters away from the wall. Red arrows show the positions and velocity of the potential hazards. Safe speeds (9) were computed and shown as heat maps. White critical lines show the position where the driver can decide to pass safely for the expected hazards.

consideration (Fig.2 a) leads to the following equation:

$$x_o/y_o = x_e/y_e, \tag{1}$$

because potential hazards that could give the maximum damage are ones that come out suddenly from the edge of blinded area. The equation (1) relates the ego-vehicle positions to the blind corner geometry. We should note that those positions were expressed in a coordinate frame the origin of which is at the corner of the wall, and its x direction aligns with the direction of edge line of the wall (shown as a red coordinate frame in Fig.2 a).

Second, assuming that v_o is constant and the ego-vehicle speed v_e is determined by a safety limited speed (see the next paragraph), the following equation holds at a critical point in which a crash will occur (Fig.2 b):

$$t_e = t_o, \tag{2}$$

where t_e, t_o are times to collision for the ego vehicle and the occluded traffic participant, which can be computed from following equations:

$$t_e = |y_e - y_o|/v_e, \tag{3}$$
$$t_o = |x_e - x_o|/v_o. \tag{4}$$

To avoid collisions, v_e must be below a certain safety limit $v_{e\max}$. That is, a critical condition holds as:

$$v_e = v_{e\max}. \tag{5}$$

Given the maximum deceleration of the ego vehicle, $a_{\max}$, the conservation law of energy will introduce the safety limited speed for the vehicle to stop as:

$$v_{e\max} = \sqrt{2a_{\max}|y_e - y_o|}. \tag{6}$$

The equation (6) defines the maximum admissible speed for the ego vehicle to stop before the crossing point (x_e, y_o) (shown as a cross in Fig.2 a).

Thirdly, by substituting the equation (3), (4), (5) and (6) into (2), and solve it for y_e, a critical position y_c can be computed as:

$$y_c = \left(-a_{\max}x_e^2 - \sqrt{a_{\max}^2 x_e^4 + 2a_{\max}x_e^2 y_o v_o^2}\right)/v_o^2. \tag{7}$$

Also, at the critical point y_c, the critical speed v_c can be introduced from the equation (6) as:

$$v_c = \sqrt{2a_{\max}|y_c - y_o|}. \tag{8}$$

The equations (7) and (8) are at critical because the other occluded traffic participant at the speed v_o can never reach the crossing point before the ego-vehicle passes through (Fig.2 c), if the ego-vehicle is at $y_e > y_c$, and vise versa for $y_e < y_c$. In other words, equations (7) and (8) define a decision point for controlling automated cars.

Lastly, we relate the above equations to introduce a safe speed for the ego vehicle at (x_e, y_e) as:

$$v_{\text{safe}} = v_c + \sqrt{2a_p|y_e - y_c|}, \quad y_e < y_c, \tag{9}$$

where a_p is a parameter defining a preferred deceleration.

Equation (7) implies that the farther the ego-vehicle from the wall edge, the smaller the critical longitudinal position y_c is. This will be followed by an increase of the safe speed limit (9). Representative safe speeds were computed and shown in Fig.3, where we have tested different lateral positions x_e. Intuitively, the risk maps show that areas near the wall edge is hazardous, because the vehicle cannot decide until the last moment whether or not potential hazards could arrive.

Note that the equation (9) is not the only relation between the safe speed and the critical point. Rather, it is one of simple expressions for automated cars to make decisions. Given a_p, $a_{\max}$, (x_e, y_e), v_o, and x_o, for example, the equation (9) will be a help for a decision making module to filter out unsafe paths if they contain a speed above the safe speed limit. And we should also note ways to compute the coordinates defined on walls (Fig.3) in a real vehicle. To do this, we have two options; a high definition vector map or a vehicle mounted laser scanner. The former contains geometric information of real road including geometries of walls and fences(Section V-A), while the latter would provide 3-d data of wall positions. In this paper, we chose the first option because we have constructed the map already, however, we should note that it is not the only solution for getting the coordinates. For unknown environments, the latter option could be more practical. We have proposed a way to extract visibility from laser scanner data in [18]. In the next section, we will describe on the application of the safe speed model (9) to integrate to our path planning module.

IV. PATH PLANNING WITH POTENTIAL HAZARDS RISK MANAGEMENT

A. General framework

We chose a Frenet coordinate planner as it compute smooth lateral and velocity local trajectories to follow. In Frenet frame optimization, the optimal path will be selected by balancing achievements of desired lateral positions and a desired longitudinal speed. This is important for our application as it is an automated vehicle driving in residential area road, where the vehicle should control its speed frequently depending on lateral steering controls. From a viewpoint of path planner field of research, our contribution in this paper is to extend the original framework, to more inherently danger situations, i.e. residential area road. From technical point of view, it should improve its mobility as the vehicle often needs to curve sharp corners which would be rare in highway streets. And, the most important thing is that it needs to clear blind corner intersections safely without making automated drive too much conservative. Thus, our safe speed model at blind corner intersections (Section III) comes into play to be integrated into a Frenet frame path planner, which we will describe in detail in the following sections.

B. Trajectory generation in a Frenet coordinate frame

In this section, we will describe mainly a trajectory generation module in our control structure (Fig.4), in detail step by step, with some basics of Frenet frame computation proposed by Werling et. al [17].

First, as the highest command, a referential path to be followed should be defined. We implemented this as a set of discrete way points $\mathbf{r}(s)$ in a global state space, which constitute a road lane. s is an individual variable expressing a longitudinal travel distance along the referential path. Combining a lateral displacement variable d, a state (s,d) constitutes a point in a Frenet frame coordinate.

Second, given current positions of ego-vehicle $\mathbf{x}_0 = (x_{g0}, y_{g0})$ in the global coordinate, they are converted to a starting point in a Frenet frame coordinate (Fig.5), which are referred to as (s_0, d_0). The subscription 0 denotes that the point is a starting point of a trajectory to be generated. In general, the referential path does not have an analytical expression. Thus, s_0 should be numerically found as:

$$s_0 = \underset{s}{\operatorname{argmin}}(\mathbf{x}_0 - \mathbf{r}(s)). \tag{10}$$

Then, d_0 can be computed as:

$$d_0 = (\mathbf{x}_0 - \mathbf{r}(s_0))^T \mathbf{n}_r(s_0), \tag{11}$$

where $\mathbf{n}_r(s)$ is a normal vector at s of the referential path.

Third, given an ego-vehicle's current heading angle θ_{x0}, a longitudinal speed v_{x0}, a curvature k_{x0}, and an acceleration

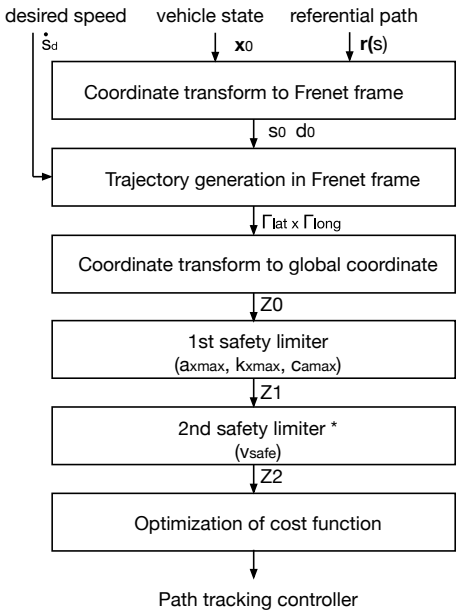

Fig. 4. A block diagram of proposed controller. *In the second limiter, coordinate transformation to wall centered coordinate should be done (see texts for detail).

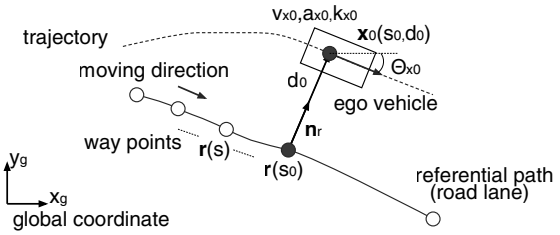

Fig. 5. A Frenet coordinate frame proposed in [17] with a slight modification for our application.

a_{x0}, the other differential states in the Frenet frame can be computed. Combined with equations (10) and (11), we express this transformation as:

$$\mathbf{q}_0 = \phi(\mathbf{p}_0), \tag{12}$$

$$\mathbf{q}_0 \equiv (s_0, \dot{s}_0, \ddot{s}_0, d_0, \dot{d}_0, \ddot{d}_0, d_0', d_0'')^T, \tag{13}$$

$$\mathbf{p}_0 \equiv (\mathbf{x}_0, \theta_{0x}, k_{0x}, v_{0x}, a_{0x})^T. \tag{14}$$

d' and d'' are the first and second derivatives w.r.t. s; $d' = \frac{d}{ds}(d)$, $d'' = \frac{d}{ds}(d')$. Here, we will skip too much details of this transformation ϕ (see [17] for its complete introduction), and proceed to the last trajectory generation process.

A candidate trajectory is one connecting the initial state $\mathbf{q}_0$ to a desired final state $\mathbf{q}_f$. In general $\mathbf{q}_f$ is described as:

$$\mathbf{q}_f = (s_f, \dot{s}_f, \ddot{s}_f, d_f, \dot{d}_f, \ddot{d}_f, d_f', d_f'')^T \tag{15}$$

As we will focus on a speed control for our vehicle in this paper (Section V and Section VI), and refrain from using an extremely low speed, $\mathbf{q}_0$ and $\mathbf{q}_f$ can be much simpler as:

$$\mathbf{q}_0 = (s_0, \dot{s}_0, \ddot{s}_0, d_0, \dot{d}_0, \ddot{d}_0)^T \tag{16}$$

$$\mathbf{q}_f = (\dot{s}_f, \ddot{s}_f, d_f, \dot{d}_f, \ddot{d}_f)^T \tag{17}$$

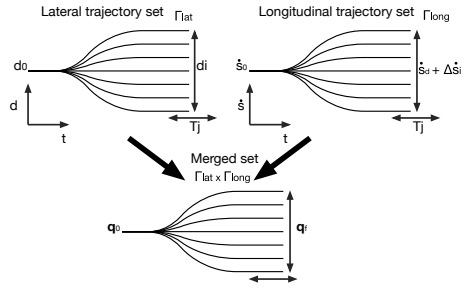

Fig. 6. Candidate trajectory generation in a Frenet planner

For a longitudinal trajectory set, we will extend $(\dot{s}_f, \ddot{s}_f)$ in (17) as $[\dot{s}_f, \ddot{s}_f, T]_{ij}$, where T is a time to reach the final state. Let $\dot{s}_d$ be a desired speed, $\Delta\dot{s}_i$ a change added to the desired speed, and assume that the final longitudinal acceleration should be zero, we obtain:

$$[\dot{s}_f, \ddot{s}_f, T]_{ij} = [\dot{s}_d + \Delta\dot{s}_i, 0, T_j] \tag{18}$$

The equation (18) is a boundary condition for a longitudinal trajectory (Fig.6). Note that the initial condition $[s_0, \dot{s}_0, \ddot{s}_0]$ contains three components, a total of five conditions require a fourth order polynomial function to be connected. Similarity for a set of lateral trajectories, the initial and final conditions are $[d_0, \dot{d}_0, \ddot{d}_0]$ and $[d_i, 0, 0, T_j]$, respectively. This requires a fifth order polynomial trajectory.

Now merging the above two sets of longitudinal and lateral trajectories, Γ_{long}, Γ_{lat}, we get merged sets, $\Gamma_{long} \times \Gamma_{lat}$ (Fig.6). Those are candidate trajectories and need to be tested according to some safety measures, the process of which we call limiters. We implemented our model of a safe speed for passing blind intersections (Section III) as one of such limiters, which will enhance the vehicle safety under uncertainty and will be a focus in the next section.

C. Limiting unsafe trajectories

For removing unsafe candidate trajectories from $\Gamma_{long} \times \Gamma_{lat}$, we introduced two types of limiters. The first limiter manage several limitations as follows:

1) the admissible longitudinal acceleration $a_{x\max}$,
2) the admissible curvature $k_{x\max}$,
3) the admissible centrifugal acceleration, $c_{a\max}$ and
4) the safe speed for passing blind intersections, v_{safe}.

A trajectory exceeding one of the above limitations will be rejected (Fig.7). The centrifugal acceleration is to evaluate the speed with respect to the curvature:

$$c_a = k_x v_x^2. \tag{19}$$

It should also note that we applied coordinate transformations for the original set $\Gamma_{long} \times \Gamma_{lat}$, as it was expressed in the Frenet coordinate frame while the above parameters were expressed in different coordinates. A closed form transformation was again proposed in [17], which we applied:

$$\mathbf{p}_t = \phi^{-1}(\mathbf{q}_t), \tag{20}$$

where $\mathbf{q}_t$ denotes each point in the trajectory set $\Gamma_{long} \times \Gamma_{lat}$. Let Z_0 a set of all candidate trajectories expressed in the

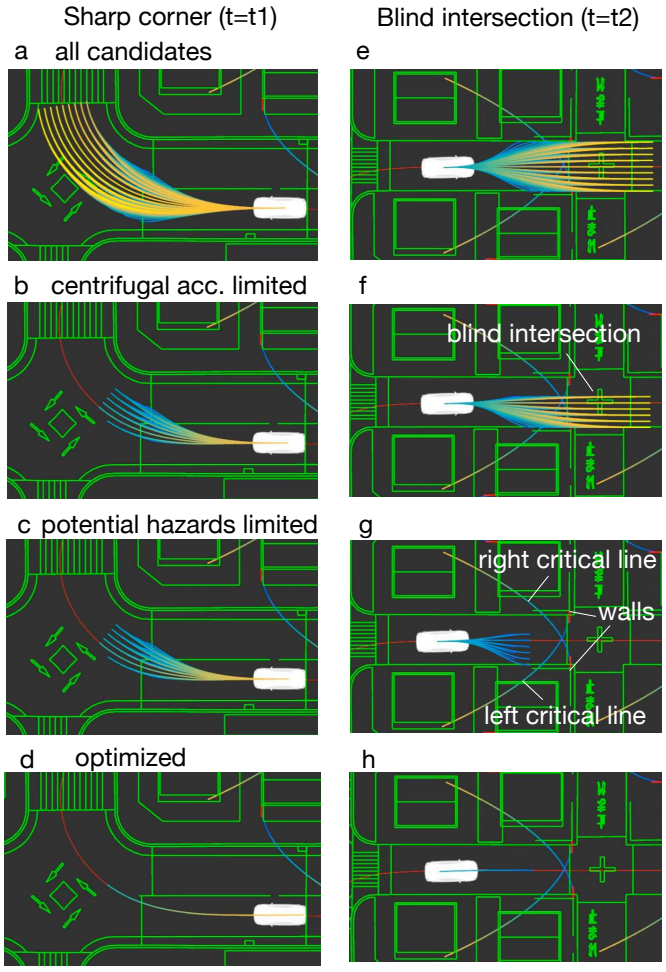

Fig. 7. Examples of limiting unsafe trajectories. The centrifugal acceleration limiter is effective for removing unsafe trajectories in curved scene (b), while the potential hazards limiter removes unsafe trajectories at the bind intersection (g). Note that the colors express the speed at each point of the trajectories, and thus inverted to the colors of Fig.3.

global coordinate, the first limiter checks the above limitations and pass only trajectories satisfying the limits, to result in a safe set Z_1. Similarly, the second limiter checks safe speed values for each points in Z_1, and passes through the safe ones, Z_2. It should note that for the safe speed limit v_{safe}, an additional transformation to the wall centered coordinate (shown in red arrows in Fig.2) was necessary, which could be easily done by ordinary coordinate transformations.

Now as the set, Z_2, is satisfying the safety limits, it can be optimized to find a optimal trajectory. A cost function should deal with other important factors for driving, such as efficiency or smoothness. Such costs can be written as:

$$C_{total} = k_{lat}C_{lat} + k_{long}C_{long}, \quad (21)$$

$$C_{lat} = k_{Jlat}J_{lat} + k_{tlat}T_f + k_{plat}d_f, \quad (22)$$

$$C_{long} = k_{Jlong}J_{long} + k_{tlong}T_f + k_{plong}(\dot{s}_f - \dot{s}_d)^2, \quad (23)$$

where, T_f, d_f, s_f are respectively the end time, the final lateral displacement, and the final speed of the trajectory, which would express efficiency of the movement. J_{lat} and

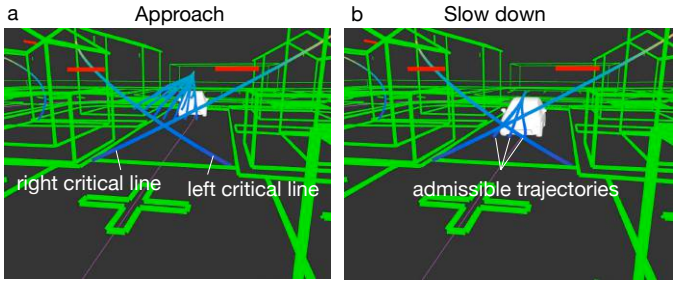

Fig. 8. (a) 3-d snapshots of approaching to a blind intersection. (b) Only trajectories below safe speed limits are admissible. Note that only critical lines are illustrated so as not to make the figure clutter. As critical lines (Eq. (7)) express the minimum speeds, thus it can be considered as limiters for candidate trajectories (see supplementary video for details).

J_{long} are sum of squared jerk, which we computed as:

$$J_{lat} = \sum_{j=0}^{n-1} \dddot{d}_j^{\,2}/n, \quad (24)$$

$$J_{long} = \sum_{j=0}^{n-1} \dddot{s}_j^{\,2}/n, \quad (25)$$

where n is a number of points in the trajectory. This would express the smoothness of the trajectory. k_{lat}, k_{long}, k_{Jlat}, k_{tlat}, k_{plat}, k_{Jlong}, k_{tlong}, k_{plong} are weights of those costs.

As a summary, the proposed path planner would set the first priority to the safety requirements, as shown in Fig.4. Candidate trajectories, generated by the Frenet planner, would be tested via the two safety limiters and optimal cost functions. Compared to the original framework, we implemented the centrifugal acceleration in the first limiter, intend to make it to improve the ride-comfort as well mobility in curved corners. Also, the second filter would manage to keep safety at blind intersection. In the next section, we are going to test the performance of the proposed system as a whole.

V. PATH FOLLOWING SIMULATIONS AND EXPERIMENTS

A. Experimental setup

Our experimental vehicle is a right handed Prius, equipped with a laser scanner (Velodyne 64-S2) and an encoder inside of the rear wheel (Fig.10 a). We fused those sensors to get stable and the accurate pose of the vehicle. We confirmed in our previous study that the positional accuracy was below 0.1 m with a 100 Hz updating rate (see [19] [1] for more details). As low level controllers, we implemented a PID-controller for vehicle speeds, and a PI controller for steering angles. Commands of those controllers were computed by a micro controller with 1kHz sampling frequency and sent to the vehicle area network (CAN). The micro controller also receives inputs via a wired ether network connection from a main desktop PC, in which all computations including the localization and the path planning are done. The update rate of the control modules in the main PC is 10 Hz. The main PC also has a direct connection to the CAN, enabling us to collect operational data of drivers. Those different sensors are synchronized by Robot Operating System (ROS).

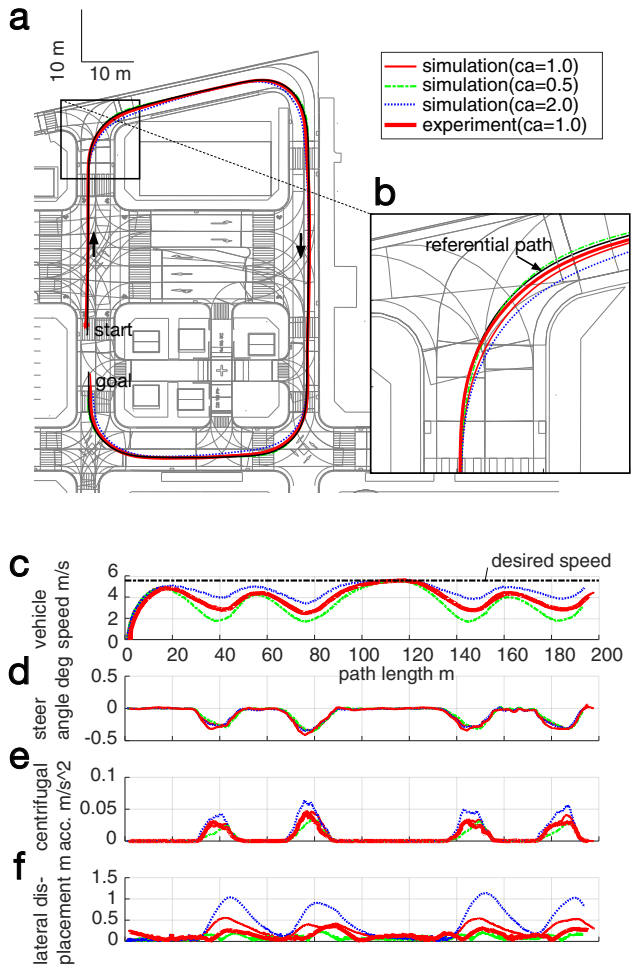

Fig. 9. Experimental validation on path tracking ability in a model city.

Before conducting expeiment, we constructed a structural map, which referred to as the high-definition vector map (Fig.10 b), using a mobile mapping system (MMS; 25 Mitsubishi Electric Corp.). The absolute accuracy of the MMS is 0.1 m. The high-definition vector map contains geometric information such as start and end points of each white line, pole, center point of each intersection, and most importantly wall and fence geometries. Those geometric features are used subsequently for data analysis and control.

B. Evaluation on path following capability

We evaluated the basic performance of our path planner by testing its path following capability with experiments. To do this, the second limiter for the potential hazards risk management (Fig.4) was turned off. Thus, paths allowed by the first limiter were sent directly to the low level tracking controller. Then our experimental vehicle was commanded to follow a referential path (Fig.9 b) at a desired speed of 5.6 m/s (20 kmph). We conducted the experiment in an educational facility for learning traffic skills in Toyota city, which provides a controlled environment with no residents but with houses and blind corners (Fig.9 a). This would be a verification of the first limiter on improving the vehicle's mobility, especially at curved corners. Clearing safely on curved corners should be a fundamental requirement for autonomous cars working in residential area road.

Overall, the vehicle achieved to clear straight lines and four corners, two of which are almost right angled. The maximum lateral displacement was below 0.3 meters, which was acceptable. Specifically we have found decreases of vehicle speed around entries of four corners and a characteristic increase at the longest straight road (around 100 meters ahead from the start point in Fig.9 a), which almost achieved the desired speed. We should note that gains of our low level speed controller was tuned well to follow a commanded speed. Thus, the result in Fig.9 c should mean that the speed was adjusted appropriately depending on shape of the referential path to follow. Among the parameters (the maximum acceleration, curvature, and centrifugal acceleration), we found that the maximum centrifugal acceleration seemed to influence the curving performance the most. Then we simulated further by changing $c_{a\max}$ from 0.5, 1.0 and 2.0 m/s^2. Simulation results clearly showed that the larger $c_{a\max}$ was followed by the faster vehicle speed and the larger lateral displacement from the referential path (Fig.9 e, f).

From those experiments and simulations, we conclude that the first limiter, especially the centrifugal acceleration limiter improved the vehicle mobility at curved corners. It should also contribute to improvement of ride-comfort. In the next section, we will focus on studying effects of the safe speed limiter for passing a blind intersection, based on the parameters tuned in this section.

VI. VALIDATION

A. Autonomous vehicle control at a blind intersection

Now the referential path was changed to one in residential area in the same test sight (Fig.10 a). The parameters were not changed except a desired speed decreased to 4.2 m/s (15 kmph). For simplicity, we started to consider three walls, illustrated by red circles in Fig.10 b. For computing safe speed limits(9), each corner of the walls, as well as each direction, was depicted from the high-definition map (Fig.10 b), resulting in three coordinate frames (one of which is illustrated in red arrows in Fig.10 b). We computed coordinate transformations to the vehicle position for each of the frames. As the result, the positions were converted to each of the frames-centered positions. Then, a safe speed limit $v_{\text{safe}}^{(i)}$, for each one of the frames i was computed. Among them, the minimum was selected for the speed limiter:

$$v_{\text{safe}} = \min\{v_{\text{safe}}^{(i)}, i = 1,2,3\}. \tag{26}$$

This experiment would be a validation of our entire system, in the controlled test sight which mimics public residential areas. It would also be a test in a situation with multiple risks, in which obstacles that may come from left and right directions from several blind areas. Now the second limiter (Fig.4) was turned on. The vehicle commanded to go straight while passing the blind intersection, slowed down

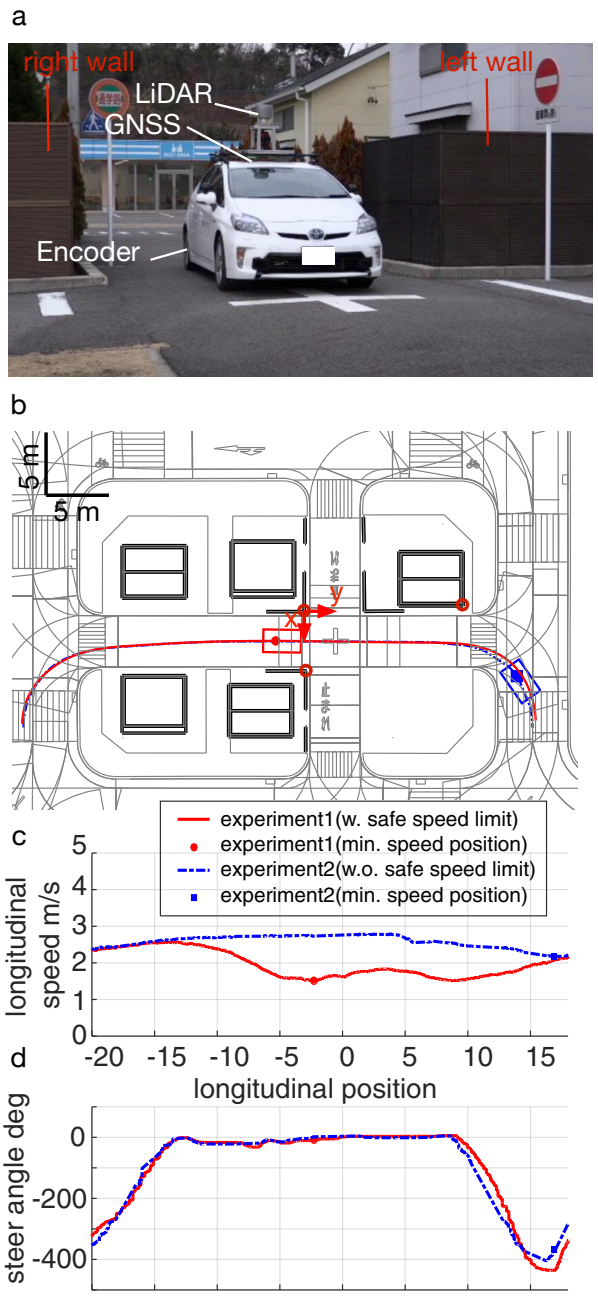

Fig. 10. Autonomous vehicle experiment at a blind intersection. Red circles indicate corners for computing safe speeds.

its speed while approaching to the intersection. Compared to the case without the second speed limiter, the minimum speed decreased to 57 % (Fig.10 c). It occurred when the front nose of the vehicle almost aligned to the line of the left wall edge (Fig.10 b). After the minimum speed, the vehicle slightly accelerated and then decelerated again to the second minimum speed, where the second left wall would have decreased visibility from the vehicle. The speed profile showed characteristic two peaks.

This would imply that the safe speed limiter worked appropriately to potential hazards from two different blind areas. In case a pedestrian or a bicyclist might come out from the

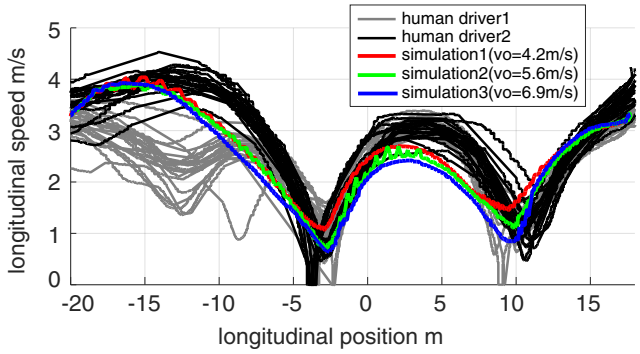

Fig. 11. Validation by comparison to expert human drivers.

wall edges, the safe limiter will help the vehicle to stop faster. Also it would be a help for those possible traffic participants to inform that the ego-vehicle is approaching and prepare for possible crashes. From the definition (9), our model is capable on setting speeds and positions of a potential hazard, as the paramters v_o and x_e. In this experiment, we did not pay too much care on those parameters as we did not have a referential profiles. We would go on further with this topic in the next section, collecting experts' driving data as such references and do direct comparison between our models and expert data, with fully a calibrated set of parameters.

B. Comparisons to expert drivers behavior

We evaluated the performance by comparing trajectories to those by human drivers. Two drivers, who had been licensed as 'fully skilled' by a safety committee regarding automated vehicle experiments in Nagoya university were recruited. The experiments were conducted in the same test sight of the previous section (Fig.10 b). Before starting, the drivers were instructed to safely drive as usual on a specified route, and informed that our extra pedestrian might come out from the blind intersection. Our extra pedestrian sometimes came out from the occluded areas, forcing the driver to stop the vehicle (some cases are shown in Fig.11 where speeds were almost decreased to zero). The other cases, where pedestrians did not come out, they drove carefully, slowing down when they passed before the intersection.

Then, we compared the paths to the ones by simulations. We get the best match (Fig.11) with the parameters as shown in Table I. Among them, potential hazards' speed v_o affected shape of the speed profiles (some examples were shown as red, green and blue lines in Fig.11). For each of different values of v_o, positions of the minimum speed shifted. The smaller the value of v_o, the smaller the minimum speed and the longer the minimum speed position was. It would mean that the vehicle speed was appropriately adjusted depending on v_o. A large/small v_o would make an autonomous driving conservative/aggressive. A slight difference between our model and the human drivers can be seen for the second minimum speed, where the drivers selected slightly smaller speed. This might be because the crossing road was wider and posed the drivers to take much attention.

Finding an appropriate v_o is challenging issue as it is an uncertain parameter. One simple way would be using the legal speed limit of crossing road, which would be obtained from a car-mounted navigation system. Another method would be building a model based on driver data. From our recent studies [20], deceleration profiles by expert drivers at several different blind intersections were mostly understandable by numerable environmental factors. Appropriate computational models would be considerable that could relate environmental factors to v_o, which will be an open question for our future studies.

TABLE I

CALIBRATED PARAMETERS OF THE PROPOSED CONTROLLER

Description		Notation	Value	Unit
Desired states	time horizon	T_j	2.5	s
	des. lat. pos.	d_i	0.0	m
		d_i min	-0.25	
		d_i max	0.25	
		d_i step	0.05	
	des. long. speed	s_d	5.6	m/s^2
		$\Delta\dot{s}_i$ min	-5.0	
		$\Delta\dot{s}_i$ max	0.0	
		$\Delta\dot{s}_i$ step	0.56	
Weights of the lat. costs	total weight	k_{lat}	1.0	–
	smoothness	k_{Jlat}	1.0	–
	lateral position	k_{plat}	100.0	–
	final time	k_{tlat}	1.0	–
Weights of the long. costs	total weight	k_{long}	1.0	–
	smoothness	k_{Jlong}	1.0	–
	long. speed	k_{plong}	200.0	–
	final time	k_{tlong}	1.0	–
Params. of the first limiter	adm. acc.	$a_{x\max}$	2.0	m/s^2
	adm. curvature	$k_{x\max}$	20.0	m^{-1}
	adm. centrifugal acc.	$c_{a\max}$	2.0	m/s^2
Params. of the second limiter	max. acc.	$a_{\max}$	0.8	m/s^2
	preferred acc.	a_p	0.5	m/s^2
	pot. hazard speed	v_o	4.2/5.6/6.9	m/s
	pot. hazard pos.	x_e	0.0	m

VII. CONCLUSIONS

In this paper we proposed an integrated path planner for an autonomous car to safely pass through blind intersections and sharp corners. To do this, we have proposed a model that can compute safe speed limits for blind intersections. The model was implemented into a Frenet frame planner capable of computing smooth trajectories with their respective velocity profiles. Experimental results in simulation and with an autonomous car in the field demonstrated the ability to generate a variety of safe and smooth predictive driving trajectories. The proposed framework can decelerate before curved corners and blind intersections, avoiding unsmooth paths, and sudden acceleration, which we consider are fundamental requirements for comfortable autonomous driving and driver assistant systems. Our future perspective would be more comprehensive validation by performing experiments in complex environment that resembles particular situation in real-life, such as blind intersection with visible and invisible pedestrians, bicycles, and other cars.

REFERENCES

[1] Y. Yoshihara, E. Takeuchi, and Y. Ninomiya, "Accurate analysis of expert and elderly driving at blind corners for proactive advanced driving assistance systems," in *Transportation Research Board 95th Annual Meeting*, no. 16-1992, 2016.

[2] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299–310, 2009.

[3] R. Hayashi, J. Isogai, P. Raksincharoensak, and M. Nagai, "Autonomous collision avoidance system by combined control of steering and braking using geometrically optimised vehicular trajectory," *Vehicle system dynamics*, vol. 50, no. sup1, pp. 151–168, 2012.

[4] S. LaValle, "Motion planning: The essentials," *Robotics Automation Magazine, IEEE*, vol. 18, no. 1, pp. 79–89, 2011.

[5] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, *et al.*, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, 2008.

[6] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech Journal*, vol. 1, no. 1, p. 1, 2014.

[7] A. Broadhurst, S. Baker, and T. Kanade, "Monte carlo road safety reasoning," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*. IEEE, 2005, pp. 319–324.

[8] F. Damerow and J. Eggert, "Balancing risk against utility: Behavior planning using predictive risk maps," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 857–864.

[9] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[10] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," in *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007, pp. 2980–2985.

[11] B. Gutjahr, L. Groll, and M. Werling, "Lateral vehicle trajectory optimization using constrained linear time-varying mpc," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–10, 2016.

[12] D. Ferguson , T. Howard, and M. Likhachev , "Motion planning in urban environments: Part i," in *Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems*, September 2008.

[13] Y. Morales, N. Kallakuri, K. Shinozawa, T. Miyashita, and N. Hagita, "Human-comfortable navigation for an autonomous robotic wheelchair," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2737–2743.

[14] Y. Morales, A. Watanabe, F. Ferreri, J. Even, T. Ikeda, K. Shinozawa, T. Miyashita, and N. Hagita, "Including human factors for planning comfortable paths," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 6153–6159.

[15] J. J. Park and B. Kuipers, "A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 4896–4902.

[16] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *J. Field Robot.*, vol. 25, no. 9, pp. 640–673, Sept. 2008. [Online]. Available: http://dx.doi.org/10.1002/rob.v25:9

[17] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 987–993.

[18] E. Takeuchi, Y. Yoshihara, and N. Yoshiki, "Blind area traffic prediction using high definition maps and lidar for safe driving assist," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 2311–2316.

[19] E. Takeuchi and T. Tsubouchi, "A 3-d scan matching using improved 3-d normal distributions transform for mobile robotic mapping," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 3068–3073.

[20] Y. Morales, Y. Yoshihara, N. Akai, E. Takeuchi, and Y. Ninomiya, "Proactive driving modeling in blind intersections based on expert driver data," in *2017 IEEE Intelligent Vehicles Symposium*, Jun 2017, pp. 901–907.