IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 6, 2020, accepted June 24, 2020, date of publication July 10, 2020, date of current version July 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3008374

A Two-Stage Real-Time Path Planning: Application to the Overtaking Manuever

FERNANDO GARRIDO[1], LEONARDO GONZÁLEZ[2,5], VICENTE MILANÉS[3], JOSHUÉ PÉREZ RASTELLI[2], (Member, IEEE), AND FAWZI NASHASHIBI[4], (Member, IEEE)

[1] Driving Assistance Research (DAR) Team, Valeo, 93000 Bobigny, France
[2] Automotive Area/Industry and Transport Division, Tecnalia Research and Innovation, 48160 Derio, Spain
[3] Renault SAS, 78280 Guyancourt, France
[4] RITS Team, INRIA, 75012 Paris, France
[5] Faculty of Engineering of Bilbao, University of the Basque Country (UPV/EHU), 48940 Leioa, Spain

Corresponding author: Fernando Garrido (fernando.garrido@valeo.com)

This work was supported in part by the INRIA and VEDECOM Institutes under the Ph.D. Grant, and in part by the Electronic Components and Systems for European Leadership (ECSEL) Project AutoDrive under Agreement 737469.

ABSTRACT This paper proposes a two-stage local path planning approach to deal with all kinds of scenarios (i.e. intersections, turns, roundabouts). The first stage carries out an off-line optimization, considering vehicle kinematics and road constraints. The second stage includes all dynamic obstacles in the scene, generating a continuous path in real-time. Human-like driving style is provided by evaluating the sharpness of the road bends and the available space among them, optimizing the drivable area. The proposed approach is validated on overtaking scenarios where real-time path planning generation plays a key role. Simulation and real results on an experimental automated platform provide encouraging results, generating real-time collision-free paths while maintaining the defined smoothness criteria.

INDEX TERMS Automated driving, path planning, overtaking, intelligent transportation systems.

I. INTRODUCTION

Urban environments are the most complex environment for the development of automated vehicles due to the multiple road users interactions (i.e. pedestrians, bikes, motorcycles) compared with highway roads. Urban road layout is also more complex than in its highway counterpart, presenting many more configurations in a tighter space, including straight stretches, intersections such as T-turns, 4-way turns, U-turns, roundabouts, etc. Both challenges demand a fast response from the automated vehicle side to properly respond to any situation, adapting to the motion and intention of other road users.

Current motion planning solutions have been focused on treating each road layout configuration separately or for a specific planning purpose or urban use case. A survey of motion prediction and risk assessment method was presented in [1], specifically studying the maneuver intention at road intersections. A motion planning approach dealing with yielding maneuvers both at intersections and roundabouts was presented in [2]. Similarly, De Beaucorps *et al.* [3] presented a decision making algorithm covering unsignalized intersections and roundabouts in presence of other vehicles. Roundabouts were approached by [4] with parametric curves from a comfort perspective. In the same way, [5] used parametric curves at roundabouts for comfort and safety reasons. Unsignalized roundabouts were treated in [6] developing an interaction-aware behavioral planner with special interest in the safety of the merging maneuver. Overtaking maneuvers represent one of the most dangerous use cases in urban driving scenarios [7], [8]. The main reason for overtaking accidents is associated with a failure to recognize hazards by the driver [9]: in particular, failing to leave enough space with the overtaken vehicle.

The main concern when performing maneuvers involving other vehicles is to correctly recognize the risks in the environment, in the case of overtaking, the intention and estimation of the vehicle to overtake. From an automated driving perspective, path planning generation plays a key role in this maneuver. Research teams have put their efforts in defining and developing algorithms to perform safer overtaking and obstacle avoidance maneuvers. The role of the path

The associate editor coordinating the review of this manuscript and approving it for publication was Malik Jahan Khan.

planning in these maneuvers is essential, not only for designing collision-free trajectories, but also to provide comfort to the occupants of the vehicle.

The paper proposes a two-staged planning method to tackle the overtaking maneuver in urban environments. It presents a two-staged algorithm where the pre-planning stage precomputes optimal curves for all possible scenarios, considering vehicle kinematic and road information. Later on, the real-time planning stage uses road topology information and the path itinerary to optimize consecutively pairs of curves loaded from the pre-planning stage, generating a smooth and continuous path. This approach is extended to scenarios with obstacles thanks to the generation of a virtual lane, allowing to transform the dynamic problem into a static path following scenario. A prediction of the obstacle's and ego-vehicle's motion are considered there to generate a safe path into the virtual lane. Finally, a solution is proposed for scenarios where an obstacle blocking the lane is detected while performing the overtaking maneuver using a fallback virtual lane to return safely to the original lane.

The rest of the paper is structured as follows: Section II presents a review for motion planning for the overtaking maneuver. Our methodology for path planning in static and dynamic environments is thoroughly explained in Section III and Section IV respectively. Results in static and dynamic scenarios are presented in Section V. Finally, Section VI presents the conclusions and discussion for future works.

II. RELATED WORK

The obstacle avoidance maneuver in automated vehicles involves complex decision making, considering obstacle information and trajectory generation. In the late 90s, Shiller and Sundar [10] determined the sharpest feasible maneuver for a vehicle using the minimum distance from which a static obstacle cannot be avoided at a given speed. The generation of collision avoidance maneuvers in emergency scenarios was also treated in [11], where a model predictive controller (MPC) was presented, considering stabilization and collision avoidance as a single problem.

Optimization approaches have been very popular to determine optimal trajectories for the overtaking problem. Shamir [12] found that the shape and time of the trajectory in a lane change does not depend on the velocity of the obstacle. They formulated an optimization problem for the lane change trajectory considering maximal acceleration during the maneuver as the only dynamic constraint, minimizing jerk. Murgovski and Sjöberg [13] also addressed the overtaking problem from an optimization point of view minimizing the error on the reference velocity and position trajectory to plan the entire maneuver in one optimization step. Karlsson *et al.* [14] used a model predictive controller in an overtaking maneuver for slow leading vehicle, with oncoming traffic in the opposing lane. Andersen *et al.* [15] proposed a visibility maximization method by providing a blind spot definition. This allows the system to determine whether its overtaking trajectory is safe, based on its perception in real-time.

The geometric characteristics of the trajectories defined for the overtaking also have been subject to study, with several curves taking precedence in the field. Xu *et al.* [16] use quartic and cubic polynomials for trajectory generation, applying an iterative path and speed optimization. Polynomials were also used in [17], this time of cubic order, to define the lane change trajectory with good results. Different security lateral distances between overtaking vehicles and obstacles in the lane change maneuver were defined in [7] using obstacle dimensions.

Quintic polynomials produced a smoother curve than quartic and cubic approaches, and ensure C^2 continuity as found by González *et al.* [18]. This study used quintic bézier curves also to generate a continuous speed profile, where the quintic curves provide a smooth jerk and acceleration. Artuñedo *et al.* [19] also used bézier curves in a real environment along with collision avoidance. Its method calculated several trajectories in the space ahead and selected the best candidate based on a cost function in a dynamic environment. Qian *et al.* [20] combined a sublex optimization of the quintic-curves path with a reference speed profile optimization using an MPC that considers dynamic and energy consumption constraints. You *et al.* [21] also used quintic curves in a collaborative strategy where they apply the infinite dynamic circles approach for detecting possible collisions.

Fu *et al.* [22] combined the circle navigation method with cubic splines to analyze the risk of collision and generate collision-free trajectories. A road course estimation on a grid-based representation of the environment is proposed in [23], this approach benefits from offline maps to generate collision-free paths combining A* and RRT graph-based algorithms. Hundelshaussen *et al.* presented an ego-centered occupancy grid-based method for drivability [24], combining it with the tentacles that are used for evaluating it and to perform the motion, integrating the method on the finalist Team AnnieWAY's vehicle of the DARPA Urban Challenge [25].

Mouhagir *et al.* proposed a method to deals with the uncertainty in the perception of the environment by combining Belief Functions to build an evidential occupancy grid and clothoid tentacles for trajectory planning on dynamic environments [26]. The evidential grid considers the safety distances between the automated vehicle and the obstacles, and they enhance the binary occupancy grid by including the road limit and the longitudinal expansion of the dynamic obstacles.

Most of the strategies described above address the lane change problem from a spatial point of view. From a time-based approach point of view, Nilsson *et al.* [27] evaluated the appropriate inter-vehicle traffic gap and time instance to perform a lane change maneuver. There, reachability analysis is done to ensure safe margins and satisfying the physical limitations of the road.

From a vehicle-modeling perspective, Petrov and Nashashibi [17] proposed kinematic modeling of the relative

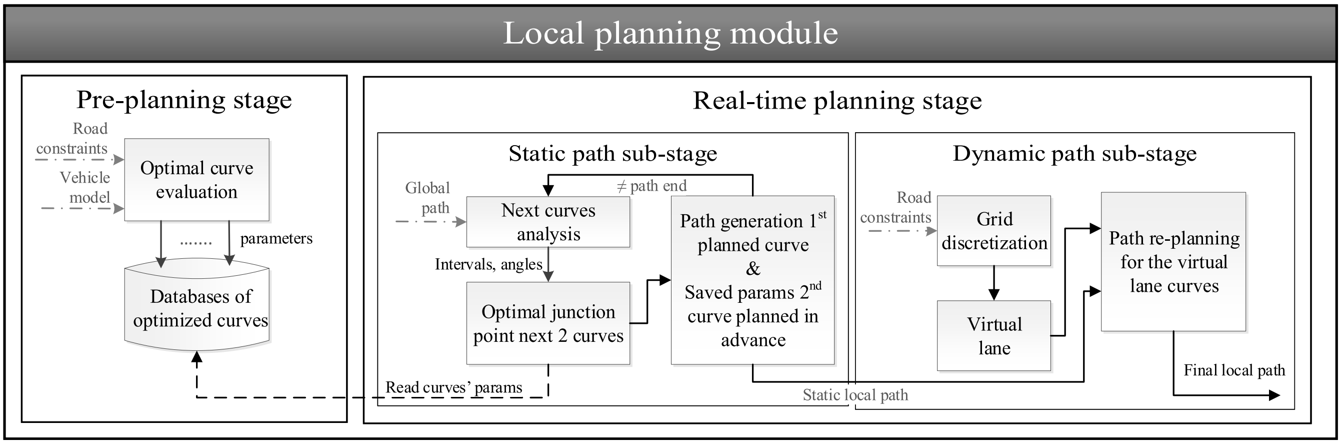


FIGURE 1. Path planning system architecture.

inter-vehicle kinematics during the overtaking maneuver with no communications, and adaptable to multiple-vehicle scenarios. Ji *et al.* [28] took into consideration the road geometry and obstacles dynamic constraints using trigonometric and exponential functions, respectively. For the collision avoidance path planning a 3D potential field was used. Then, it uses a multi-constrained MPC for path-tracking, trying to minimize the risk through evasive maneuvering. Dixit *et al.* also used an MPC framework, this time a robust implementation using reachability analysis alongside potential fields to compute safe areas to perform the overtaking maneuver in highway scenarios [29]. Finally, Wei *et al.* used a non-linear MPC for a collision-free corridor. It uses driver data in risk scenarios to study hazards and account for them in different scenarios, including overtaking a slow leading vehicle [30].

Recently an emphasis on data-driven and end-to-end solutions has gained weight in the field. Ngai and Yung [31] proposed a multiple-goal reinforcement learning (MGRL) framework in a simulation environment with multiple agents for the overtaking maneuver. Kaushik *et al.* [32] and Xie *et al.* [33] proposed deep learning approaches. The first one used reinforcement to train its agents in highway environments. While the second, used long short-term memory (LSTM) networks to predict agents' actions and decide upon lane changes.

Authors in the literature review target dynamic environments as specific scenarios: emergency scenarios in [11], overtaking as a singular scenario in [12], [13] from a dynamics point of view, in [14], [15] as a predictive point of view with other obstacles, overtaking as a geometric problem in [16]–[25], focusing on the occupancy of the space in [27], from a vehicle-modeling point of view [17], [28] or from the use of data-driven data [29]–[33], among others.

Meanwhile, the proposed approach aims to propose a general method to adapt the static path to fit with any dynamic scenario by generating a virtual lane overlying the original one. It comprises the points covered by the other authors: emergency situations, consideration of vehicle and surrounding obstacles dynamics, the geometry of the path, discretization of the space and use of a priori data (static information) such as the kinematics of the vehicle for the pre-planning stage.

A. WORK CONTRIBUTIONS

This work presents a path planning solution based on two-stages: pre-planning and real time-planning. Figure 1 shows the proposed path planning architecture. The pre-planning stage [34] (left part of Fig. 1) generates the optimal curve for each feasible single-turn scenario, considering the static information of the road and the vehicle kinematics. The real-time stage (right part of Fig. 1) generates a continuous path by loading the optimal curves from the previous stage that fit better with the road layout [35], considering a planning horizon of two curves. The generation of virtual drivable lanes significantly reduces the computational load, being able to generate optimal trajectories to manage dynamic scenarios.

The main contributions of this work are the following:

- Modeling all kind of road layout as a set of single-turn scenarios
- Off-line curve optimization for each feasible single-turn scenario considering static information of road and vehicle kinematics.
- Human-like turning behavior, cutting consecutive curves in S-shape turns or opening them in U-shape turns, generating smoother paths.
- Real-time analysis of sharpness and space between turns to reduce curvature profile and ease path tracking.
- Real-time planning of two consecutive turns with curves loaded from pre-planning stage.
- Virtual lane generation to handle overtaking maneuvers in dynamic environments
- System validation on a real platform, including a comparison with other planning methods.

III. PATH PLANNER FOR STATIC ENVIRONMENTS

The local planner for static paths is divided in a pre-planning (left part of Fig. 1), and real-time stage (middle part of Fig. 1).

In the pre-planning stage, a series of primitives curves are defined, called single-turn scenarios (optimal curve evaluation sub-stage in Fig. 1), using three parameters (angle of turn, and distance available before and after the turn). Then, the optimal parametric curve for these single-turn scenarios is pre-computed and saved into databases.

For the static path sub-stage in the real-time planning, the single-turn scenarios that better fit with the upcoming road layout configuration is selected from the databases as shown in Fig. 1. That way, any road layout configuration is modeled as a series of turns. A planning horizon of two curves is chosen, where the algorithm optimizes the upcoming pair of curves for the corresponding turns in the itinerary, according to smoothness criteria. This local planner finally generates a smooth continuous path that is later tracked by the vehicle.

A. ASSUMPTIONS AND CONSTRAINTS

The algorithm considers a series of assumptions and constraints concerning the road, vehicle kinematics and execution times, as follows:

(i) Concerning to the road: Information about the road and obstacles ahead is coming from the perception stage. The global planner is assumed to be accurate, as well as the vehicle positioning. This accurate global planner provides a series of way-points to the local planner, which form the itinerary to follow, along with the lane width. Since lane width in urban roads is usually between 3 and 3.7 meters, a 3 meters lane width is assumed for our purposes.

(ii) Concerning to the automated vehicle: The vehicle model is a well-known non-holonomic system, whose limitations are defined by its physical parameters such as width, length, and the maximum steering angle, constraining the generated curves to its maximum curvature.

(iii) Concerning the execution time: real-time capabilities play a key role in path planning implementation, especially in emergency situations. Quartic Bézier curves are chosen as primitive for the path generation, ensuring path continuity by introducing the curvature constraint and presenting a low computational cost.

B. PRE-PLANNING STAGE

This first stage pre-computes the optimal curve of the vehicle. It considers the aforementioned constraints, saving the parameters of the optimal curves into databases.

Fig. 2 illustrates the pre-planning stage optimization. Each path generation is carried out in a given distance-based horizon where three singular points are extracted from the global planner: 1) the initial point G_{n-1} (the closest singular point to the vehicle position); 2) the ending point G_{n+1} (end of distance-based horizon optimization); and 3) the turning point (G_n), which is the singular connecting the initial and ending points of the single-turn model.

Based on this global planner discretization, the off-line local planner defines the single-turn scenarios with three parameters, allowing to model the road layout as a set of turns:

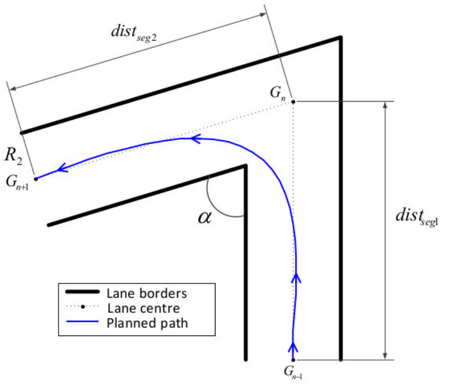


FIGURE 2. Single-turn scenario with optimal curve search [34].

- The angle of the turn (α)
- The distance from the initial point to the turning point ($dist_{seg_1}$)
- The distance to exit the turn ($dist_{seg_2}$)

To cover the maximum number of single-turn scenarios, the algorithm iterate over the three parameters. Since these parameters are the indexes of the databases containing the optimal curves, a trade-off between database resolution and database size is needed to find the minimum variation step for these parameters. First, the turn angle is defined into the following range:

- The minimum α considered is 40° being the maximum steering angle of the platforms.
- The maximum α is 180°, i.e. a straight line.
- The variation step for α is 5°, since it is the minimum value that ensures a bounded lateral error of $20cm$ in the worst possible turn, which is the one whose angle is the maximum turning angle of the vehicle (around 40°), with a maximum heading angle error of 2.5°, that maintains database access time and size feasible.

Second, the distances $dist_{seg_1}$ and $dist_{seg_2}$ are defined in the following range: $2m \leq dist_{seg_1} | dist_{seg_2} \leq 40m$ by $1m$

- The minimum distance for $dist_{seg_1}$ and $dist_{seg_2}$ is 2m since it is the minimum distance allowing the vehicle to perform the sharpest feasible turn, considering the kinematic model of the vehicle: $D = k_{max} = tan(\phi)/L = 1/R$, where ϕ is the maximum steering of the platforms (minimum value is 38.5°), L is the vehicle length (being 1.25m for small vehicles), k_{max} is the maximum curvature ($0.63m^{-1}$) and D is the minimum distance for this maximum curvature, which would be $1.58m$, value rounded to $2m$.
- The maximum distance considered for $dist_{seg_1}$ and $dist_{seg_2}$ is $40m$, being the perception range for the detection of obtacles, defining the maximum distance of the following singular point from the global planner.

- The variation step for changing these distances is 1m, to guarantee real-time evaluation in the set of databases [34].

Optimal curves for right-turns are computed geometrically by symmetry of the equivalent left single-turn. A quartic Bézier curve is generated for each single-turn iteration using Equation 1, here $n = 4$ is the degree of the polynomial equation and P_i are the control points defining the curve. The five control points P_i are moved both longitudinally and laterally to find the optimal curve. Equation 1).

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i P_i \quad t \in [0, 1] \tag{1}$$

All generated curves must ensure continuity with the maximum curvature constraint and the curvature at the initial and final segment. 4th degree Bézier curves are chosen because they are simple to manipulate as they are defined by control points. Additionally, the Convex Hull property of Bézier curves constrains path generation to remain within the lane width. These curves present a low computational load since they are defined from the equations of the analytical method [36]

The optimal curve is chosen using the cost function Q shown in Equation 2, which minimizes the curvature (k) and its derivative ($\dot{k}$) in all the points of the curve.

$$Q = \sum\nolimits_{i=0}^{n} |k_i| + |\dot{k}_i| \tag{2}$$

Four different databases of optimal curves are generated, where the initial and final position of the ego-vehicle in the lane is as follows:

1) DB_1: Initial position at lane center, final position at the lane center.
2) DB_2: Initial position at lane center, final position displaced to the lane border.
3) DB_3: Initial position displaced to the lane border, final position at the lane center.
4) DB_4: Final position displaced to the lane border, final position displaced to the lane border.

This feature emulates the behavior of the human drivers, who use the entire width of the lane depending on the road layout for reducing the curvature profile. These curves pre-generated are available in the module Databases of optimized curves, presented in the bottom part of the pre-planning stage (Figure 1).

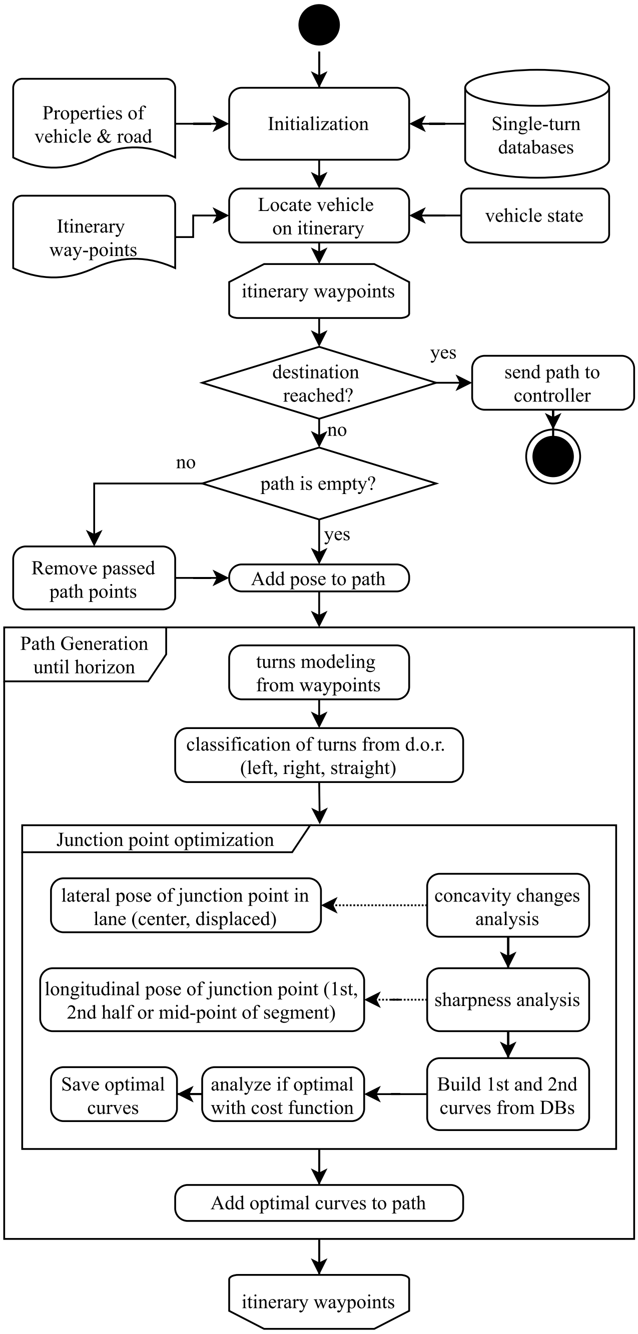

Properties of vehicle & road
Initialization
Single-turn databases
Itinerary way-points
Locate vehicle on itinerary
vehicle state
itinerary waypoints
destination reached?
yes
send path to controller
no
no
path is empty?
yes
Remove passed path points
Add pose to path
Path Generation until horizon
turns modeling from waypoints
classification of turns from d.o.r. (left, right, straight)
Junction point optimization
lateral pose of junction point in lane (center, displaced)
concavity changes analysis
longitudinal pose of junction point (1st, 2nd half or mid-point of segment)
sharpness analysis
Save optimal curves
analyze if optimal with cost function
Build 1st and 2nd curves from DBs
Add optimal curves to path
itinerary waypoints

FIGURE 3. Real-time path planning flowchart.

C. RT PLANNING: THE STATIC PATH SUB-STAGE

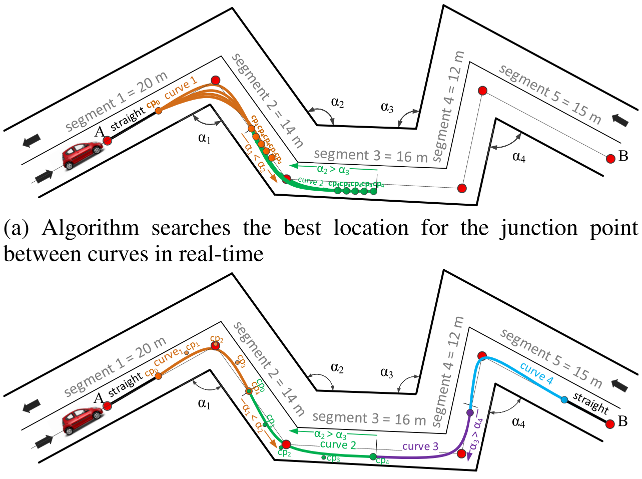
Flowchart 3 explains the process to plan the complete trajectory in real-time for static paths. As shown in the middle part of Fig. 1, this phase can be divided into three fundamental tasks:

1) **Next curves analysis**: It analyses the turns and straights stretch of the road. It determines the available space among them and the sharpness of the turns.
2) **Optimal junction**: A real-time evaluation seeks the optimal location for the junction point between curves, evaluating turns in pairs. It loads the better curves from the databases, based on the previously defined cost function Q.
3) **Add planned curves to path**: Once the junction point is found, the algorithm assembles the optimal curve for the upcoming turn with the parameters loaded from the database. In addition, it keeps the optimal curve parameters for the following turn, which has been planned in advance.

Finally, after repeating the process for all the turns on the itinerary, the complete static path is fed to the dynamic path component, which modifies it according to the obstacles present in its path (explained in the next section).

This approach can compute the distance among different road elements such as turns, intersections, and roundabouts. For every turn, the real-time algorithm plans the curve that better fits by considering a pair of turns: the upcoming and the following one.

An example of this behavior is shown in Fig. 4, consisting of a two-lane road where some consecutive turns. In the figure (4a), point A shows the departure and point B the destination. The red points represent the way-points defining the itinerary for the path-following scenario, received from the global planner.



(a) Algorithm searches the best location for the junction point between curves in real-time

(b) Planned path after repeating the process for every curve where the best junction points are maintained

FIGURE 4. Real-time path planning process: from loading the curves from the databases to get the whole path.

Using this itinerary the algorithm computes the angles of the turns that will be found in the itinerary ($\alpha_1, \alpha_2, \alpha_3$ and α_4), and its corresponding inter-distance (segment$_1$, segment$_2$, segment$_3$ and segment$_4$). The figure (4a) shows the process to find the best junction point between the turns α_1 and α_2 considering the sharpness of both.

First, a criteria is defined to limit the longitudinal position of the junction point in the segment between both turn center points, by evaluating the sharpness of the two turns (see table 1).

TABLE 1. Criteria for longitudinal position optimization of junction points.

Sharpness comparison	$\alpha_1 < \alpha_2$	$\alpha_1 = \alpha_2$	$\alpha_1 > \alpha_2$
Junction point.long pose	2nd half	mid-point	1st half

In the case of Fig. 4, as α_1 is sharper, the algorithm searches the junction in the half of the segment$_2$ which is closer to α_2. That way, the algorithm gives more possibilities to plan the curve for the sharper turn, which is more difficult to track.

Additionally, the curves direction of rotation (d.o.r), or concavity change, is considered to decide whether to start and end the curve, at the center of the lane or, close to the lane boundary. For this purpose, another criteria for the lateral position in the lane of the junction point is defined in Table 2.

TABLE 2. Criteria for lateral position in lane of junction points.

Turn	Direction of rotation			
α_1	left	right	left	right
α_2	left	right	right	left
junction point.lat pose in lane	Lane border		Lane center	

Then, the algorithm can determine which databases are loaded since the lateral position in the lane for the two upcoming turns has been evaluated before. Table 3 shows the different possibilities for loading the optimal curves passing trough the turns layout.

TABLE 3. Databases selection for consecutive optimal curves.

Databases to load	Initial point α_1	Final point α_1 Initial point α_2	Final point α_2
$DB_1 - DB_1$	lane center	lane center	lane center
$DB_1 - DB_2$	lane center	lane center	lane border
$DB_2 - DB_3$	lane center	lane border	lane center
$DB_2 - DB_4$	lane center	lane border	lane corder
$DB_3 - DB_1$	lane border	lane center	lane center
$DB_3 - DB_2$	lane border	lane center	lane border
$DB_4 - DB_3$	lane border	lane border	lane center
$DB_4 - DB_4$	lane border	lane border	lane border

Coming back to the example of Fig. 4, as α_1 is a right turn and α_2 is a left turn, there is a concavity change. In this case, the junction point is placed at the center of the lane, due to the previously described smoothness criteria.

Therefore, the algorithm generates the optimal pair of curves for both turns by loading them from the databases for each position of the junction point. The localization of both curves is minimized considering the best junction. The parameters α_1 and α_2 are saved for the next iteration. The process is repeated for the following turns, getting the whole static path as depicted in the figure (4b). In the next iterations, the difference lies in the upcoming turn will plan the following curve.

IV. DYNAMIC PATH SUB-STAGE: PLANNER OBSTACLE MANAGEMENT

This section describes how the dynamic obstacles are managed for the path planner (right part of the figure 1), creating collision-free paths [37]. For the sake of clarity, the dynamic planner is introduced using an overtaking maneuver as example, generating four consecutive curves (see Fig. 5): two for the first lane change and another two for returning to the original lane once the obstacle is overtaken.

The local path planner for dynamic environments follows three stages: 1) grid discretization; 2) virtual lane generation; and 3) path re-planning (see Fig. 1). For the adaptation of the static to dynamic path generation, a virtual lane is proposed. To accomplish this generation, a grid discretization of the road was used, where the obstacle information is perceived by the sensors. This virtual lane overlays the current lane

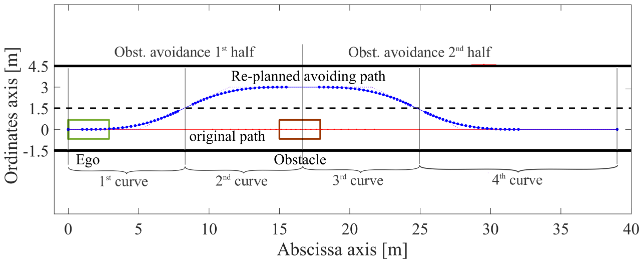


FIGURE 5. Trajectory planned with the proposed algorithm for obstacle avoidance.

and modifies the way-points of the itinerary, re-planning only the driving area where interaction with other traffic agents occurs.

A. GRID DISCRETIZATION

The purpose of using grids is to obtain a better description of the road representing the space as free or occupied by means of cells, making a discretization of the road space allowing the system to consider the uncertainty of the sensors, and as a result, the dynamism of the road [38]. In case any obstacle is detected in front of the ego-vehicle, the dynamic path planner will receive its Oriented Bounding Box (OBB). In our system, it receives the OBB of the obstacle as 2D coordinates of the rear part of the vehicle in front, i.e. the distance from the automated vehicle to such obstacle [39].

The OBB only provides information about the obstacle width for a front obstacle, leaving the vehicle length unknown or not accurate enough. Then, an obstacle classification is performed to determine the security distance according to the vehicle type, a similar classification is done in [7].

TABLE 4. Obstacle types classification based on the width.

Road user dimension	Road users classification according to their dimensions			
	Vulnerable	Cybercar	Car	Bus/truck
Width [m]	[0.0 - 1.0)	[1.0 - 1.4)	[1.4 - 2.1)	[2.1 - 3)
Length max [m]	3	2.9	5.5	18
Security dist. [m]	1.5	width/2	width/2	width/2

This classification is performed using Table 4. There, different vehicle types are defined based on the European Euro Ncap Structural category and the American US EPA Size Class regulations. For each vehicle type, both the width-range and its equivalent length have been determined by analyzing the dimensions of their reference vehicles. It provides obstacle length estimation for each obstacle, which in turn allows the lateral and longitudinal safety distances to be properly determined, to plan the avoiding maneuver.

The space discretization is performed by representing the space occupied by the obstacle after the length prediction in the grid matrix. This area corresponds to the axis-aligned Bounding Box (BB), depicted with the red rectangle in Fig. 6.

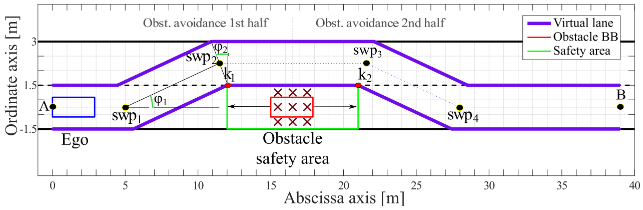


FIGURE 6. Virtual lane generation for obstacle avoidance.

This discretization allows easier recognition of the free space on the road, where space occupied the BB by in the discretization, is represented with the red crosses. Once the obstacle's BB is formed, lateral and longitudinal safety distances are added (external green rectangle). On the one hand, a longitudinal distance of the ego-vehicle length is considered both on the front and the rear, based on previous works in the literature [12], [40]. On the other hand, a safety lateral distance is chosen from Table 4, according to the obstacle type.

B. VIRTUAL LANE MODULE

Based on this grid discretization, a virtual lane is built to avoid obstacles on the path. All obstacles are classified, and the safety area is defined, as shown in purple lanes for the overtaking maneuver (Fig. 6).

The aim of the virtual lane is to modify the itinerary (A, B) by adding some supplementary way-points (swp_n) to the global path, resulting in $(A, swp_1, swp_2, swp_3, swp_4, B)$. These points define the center of the new virtual lane, from where its limits are computed geometrically.

The generation of the virtual lane starts from the internal points of the obstacle's safe area k_1 and k_2. From these points are placed the supplementary way-points swp_2 and swp_3 on the left lane. It constitutes the last way-point for the first lane-change and the first way-point for the second lane-change, respectively. Finally, the points swp_1 and swp_4 for the initial and final phases of the lane-changes are computed, placing them in the departing right lane.

The angles φ_1 is used to find the least slope on the virtual lane, and φ_2 (Fig. 6) is used to determine the location of the point swp_2. The last step moves laterally the way-points of the new global itinerary that form the center of the virtual lane, with a positive and a negative distance of half of the road width.

Both angels (φ_1 and φ_2) are modified from 0° degrees to the steepest (the worse case is 45 °), getting the position of these two way-points. Then, the real-time local planner finds the optimal junction point between the two curves for the lane-change, as explained in the static path planner. Finally, the optimal virtual road is generated (in purple in Fig. 6). This way, just by modifying the global path we use the static local planner for dynamic behavior, only searching in real-time the least feasible slope for the lane-change.

For moving obstacles, the grid is modified considering the obstacle speed to predict the space in front. In the virtual lane module, only the worse case is considered, it means, using maximum speed and acceleration, as described in Equation 3, based on [41],

$$x_{obst_{pred.}} = x_{obst} + vMax_{obst} * vMax_{obst} / accMax_{obst} \quad (3)$$

where $x_{obst_{pred}}$ represents the prediction for the obstacle's occupied space, x_{obst} is the current position, and $vMax_{obst}$ and $accMax_{obst}$ are the maximum speed and acceleration of the obstacle, respectively. The new supplementary way-points for the new virtual lane are calculated in the same way as shown before. The resulting new virtual lane for moving obstacles is depicted in Fig. 7.

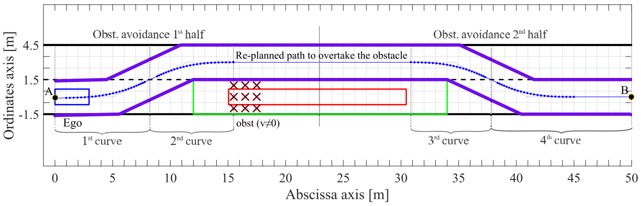


FIGURE 7. Virtual lane generation for moving obstacles considering the obstacle's occupied space prediction.

C. RE-PLANNING METHOD

The virtual lane module gives the boundaries used in the re-planning module. Our real-time re-planning approach considers the speed and acceleration of both ego-vehicle and obstacles. It re-computes the trajectory if there is a gap between the predicted and traveled distances. The re-planning method is presented in the Algorithm 1.

Algorithm 1 Re-Planning Alg. for Dynamic Path Generation

```
1:  **Receive** global path and static local path
2:  **Check** if there is any obstacle in the path
3:  **Check** if ego-vehicle is faster than obstacle
4:  **Compute** accelerations of ego and obstacle
5:  **Compute** distance covered by ego and obstacle on $\Delta T$
6:  **if** dist. covered by ego or obst. in $\Delta T >$ threshold **then**
7:      Compute Time-to-collision (TTC)
8:      **if** TTC $\leq$ 6 s **then**
9:          **Predict** distance covered by obstacle on TTC
10:         **Predict** distance covered by obstacle on next $\Delta T$
11:         **Project** obstacle position
12:         **Compute** dist. traveled by ego and obst. on $\Delta T$
13:         **Compute** obstacle BB from obstacle width
14:         **Compute** obstacle's safe zone
15:         **Compute** new global path and new virtual lane
16: **Update** previous speeds and accelerations
```

This process consists of the following steps: 1) The algorithm receives from the static planning stage both the global and the local path. 2) it checks if there is any obstacle in the itinerary. 3) If it is faster than the ego-vehicle, which could create a risk of collision. 4) In that case, the acceleration of both ego-vehicle and obstacle is computed for predicting their motion. 5) Besides, the algorithm computes the real distance covered by both ego-vehicle and the obstacle in the last ΔT. This ΔT is the parameter for the planning period, which is under 0.1s to ensure a real-time performance. If this distance is non-negligible, being higher than the desired threshold (i.e. 0.5 meters for urban environments), then the computation of the new virtual lane begins. 6) Time To Collision (TTC) is computed. Then the algorithm checks if the collision would occur in the range of accuracy of the vision system (i.e. <6 seconds for available perception). 7) Once the TTC is computed, the algorithm can update the predicted distance that both ego-vehicle and obstacle will cover in this time, projecting the position of the obstacle ahead. 8) In the same way, the prediction of the distances covered in ΔT is done. 9) Then, the Bounding Box of the obstacle can be recomputed from its projection, as well as the safety area for the avoidance maneuver. 10) Finally, the supplementary way-points corresponding to the new itinerary can be recomputed from the safety area, following the virtual lane method, described in the previous section.

D. FALLBACK RETURN-TO LANE MANEUVER

Figure 8 shows an example of the emergency scenario described, where the fallback corridor is depicted in dark green. This fallback trajectory is generated to cancel the overtaking manuever after the ego vehicle's perception system detects a second obstacle "Obst 2" on its way. Then, the system switches between the original global path to the new one by adding only two new supplementary waypoints, $nswp_3$ and $nswp_4$ depicted in green in Fig. 8, and removing the supplementary waypoints corresponding to the curve for returning to the lane after avoiding the first obstacle, i.e. red swp_3 and swp_4. Finally, the system switches between the original (red) to fallback (green) itinerary, computing the fallback trajectory for returning safely to the original lane.

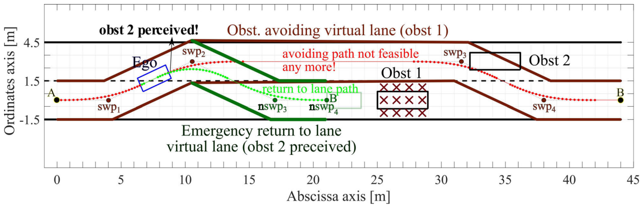


FIGURE 8. Emergency return to the lane maneuver under unexpected vehicle in path.

V. EXPERIMENTS

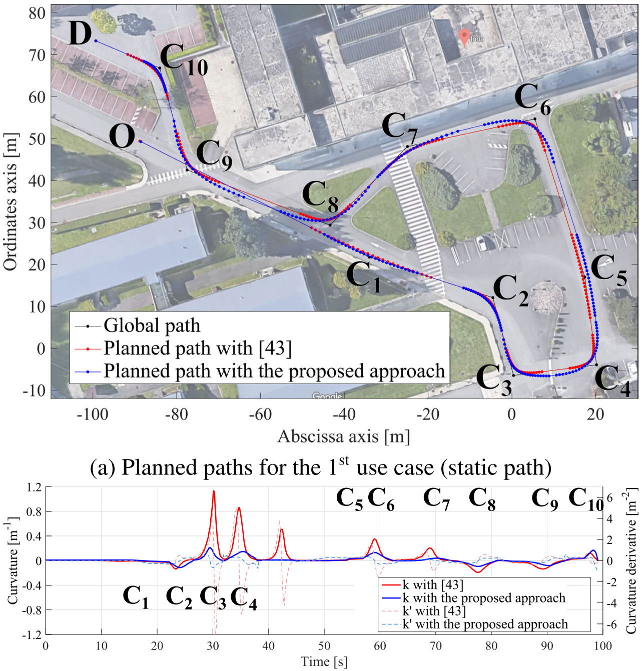
This section evaluates the two-stage planning performance. It was tested both in simulations and in an automated platform. The simulation environment combines both Pro-Sivic[1] and RTMaps[2] software. Pro-SiVIC is used to simulate the

[1] www.civitec.com
[2] www.intempora.com

driving environment, providing different car models. The path planning algorithm is generated using RTMaps that sends vehicle actuators' commands to Pro-SiVIC environment.The experimental platform is a Cybercar. It is a low-speed vehicle designed to operate on urban roads, both for passengers and goods transport [42]. It is equipped with a Real-Timne Kinematic Differential Global Positioning System (RTK-DGPS), an Inertial Measurement Unit (IMU) and two front 2D Lidar sensors.

A. RESULTS FOR THE STATIC PATH PLANNER

First use case evaluates vehicle performance when the vehicle is driven without any other traffic agent interaction. The test is carried out in real vehicles using the Cybecar platform. The proposed planning approach is compared with a Bézier-based path generation without offline stage [43].



(a) Planned paths for the 1st use case (static path)

(b) Curvature and curvature derivative of the planned path for the 1st use case (static path)

FIGURE 9. Planned path and metrics for evaluating the proposed approach in a static environment.

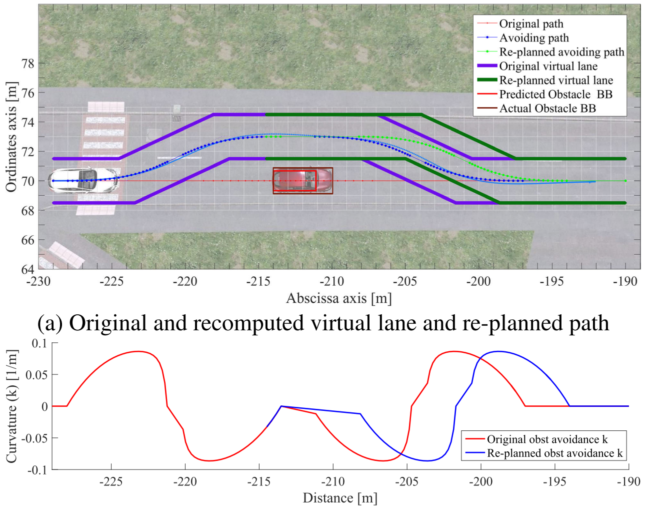
Figure 9a shows the experimental area. The vehicle travels from point O to point D, while the curves C_1 to C_{10} represents singular points in the path. The black line shows the itinerary or global path, the dark and light blue line represent the planned path with the proposed approach, whereas the red line represent the planned path with the entirely real-time approach, respectively.

Fig. 9b depicts the curvature (solid line) and the curvature derivative (dashed line) for both planners. Red color is used for the proposed planning whereas blue color plots the performance of the algorithm presented in [43]. One can appreciate how both are significantly reduced in the proposed approach. For example, the U-turn formed by curves $C_2 - C_5$, corresponding to time 22-50s, the maximum curvature for the proposed approach is under $0.2m^{-1}$ whereas it is over $1.1m^{-1}$ for [43]. Additionally, a difference of the maximum curvature derivative can be highlighted, being $0.75m^{-2}$ for the proposed approach and $5m^{-2}$ for [43].

The proposed planning is able to generate a real-time planning that presents a significantly smoother path, increasing the ride comfort thanks to less abrupt changes in the curvature profile.

B. RESULTS FOR THE DYNAMIC PATH PLANNER

This section evaluates the two-stage architecture when dealing with traffic agent interactions. Specifically, an overtaking maneuver scenario is used, evaluating both: 1) the generation of the overtaking path in real-time; and 2) the ability to provide a new trajectory in case of the maneuver is cancelled.



(a) Original and recomputed virtual lane and re-planned path

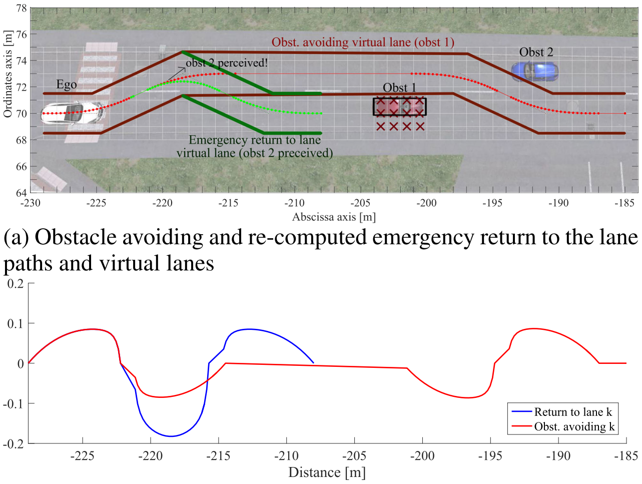
(b) Curvature of the path

FIGURE 10. Virtual lane re-computation and path re-planning applied to a wrong prediction of the obstacle dimensions.

Fig. 10a shows the initial scenario for the simulations. The purple line represents the generated virtual lane for overtaking the obstacle whereas the green line depicts the re-planning carries out during the maneuver based on the time-to-collision. With the new constrains, the static local planner computes the new path depicted in light green fitting into the new virtual lane, overtaking the obstacle safely.

The figure 10b shows that the smoothness of the re-planning path. Solid red line represents the original path. Solid blue line depicts the re-planning path. One can appreciate how the proposed approach avoids curvature peaks or discontinuities, being the curvature peak below $0.1m^{-1}$.

A fallback path generation during an overtaking maneuver is evaluated in Fig. 11a. A new obstacle is detected when performing the overtaking (Obst 2), being able to cancel the maneuver and returning the vehicle to its lane. The fallback path is generated by the creation of a new virtual lane, shown in green. The last way-points of the original avoiding virtual lane are replaced by new points in the right lane while



(a) Obstacle avoiding and re-computed emergency return to the lane paths and virtual lanes

(b) Curvature and curvature derivative of the path

FIGURE 11. Emergency return to the lane re-computation of the virtual lane, aborting the original avoiding maneuver.

leaving enough distance respect to the first obstacle (Obst 1). Curvature evaluation is presented in Fig. 11b. Despite the emergency maneuver, the fallback path still has a low-curvature profile (below 0.2 m^{-1}), validating the proposed approach.

VI. CONCLUSIONS AND FUTURE WORK

This paper proposes a local path planning algorithm divided into two sub-stages: pre-planning and real-time planning. The former optimize offline every possible single-turn scenario considering road layout and vehicle model as constrains. The latter creates a continuous path by optimizing two consecutive curves using the first stage. Concavity changes of the turns are used to create four databases, providing a human-like behavior by using the full lane width. Thanks to the pre-planning stage, the real-time stage only needs to optimize the junction point between curves, reducing the computational load. Experiments provide encouraging results, providing low curvature-profile paths even in emergency situations.

The proposed planner is evaluated in dynamic environments in an overtaking maneuver. A virtual lane modifies the original itinerary, easily adapting the trajectory. The algorithm uses a grid-based discretization of the space to classify the obstacle and predict its length, being able to either extend the path to finish the maneuver or generate a fallback trajectory in case the maneuver is cancelled.

Future works will be focused on the decision-making stage, extending the current work to deal with scenarios where more simultaneous traffic agent interactions occur.

REFERENCES

[1] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech J.*, vol. 1, no. 1, pp. 1–4, Dec. 2014.

[2] J. Nilsson, M. Brannstrom, 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.

[3] P. de Beaucorps, T. Streubel, A. Verroust-Blondet, F. Nashashibi, B. Bradai, and P. Resende, "Decision-making for automated vehicles at intersections adapting human-like behavior," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 212–217.

[4] D. Gonzáleza, J. Pérezb, and V. Milanés, "Parametric-based path generation for automated vehicles at roundabouts," *Expert Syst. Appl.*, vol. 71, pp. 332–341, Apr. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417416306583

[5] R. Lattarulo, L. González, E. Martí, J. Matute, M. Marcano, and J. Pérez, "Urban motion planning framework based on N-Bézier curves considering comfort and safety," *J. Adv. Transp.*, vol. 2018, pp. 1–13, Jul. 2018.

[6] M. Rodrigues, A. McGordon, G. Gest, and J. Marco, "Autonomous navigation in interaction-based environments—A case of non-signalized roundabouts," *IEEE Trans. Intell. Veh.*, vol. 3, no. 4, pp. 425–438, Dec. 2018.

[7] V. Milanés, D. F. Llorca, J. Villagrá, J. Pérez, C. Fernández, I. Parra, C. González, and M. A. Sotelo, "Intelligent automatic overtaking system using vision for vehicle detection," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3362–3373, Feb. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417411013339

[8] E. Thorn, "A framework for automated driving system testable cases and scenarios," Dept. Transp., National Highway Traffic Safety, Washington, DC, USA, Tech. Rep. DOT HS 812 623, 2018.

[9] G. M. Fitch, S. E. Lee, S. Klauer, J. Hankey, J. Sudweeks, and T. Dingus, "Analysis of lane-change crashes and near-crashes," U.S. DOT Nhtsa, U.S. Dept. Transp., Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Tech. Rep. DOT HS 811 147, 2009.

[10] Z. Shiller and S. Sundar, "Emergency lane-change maneuvers of autonomous vehicles," *J. Dyn. Syst., Meas., Control*, vol. 120, no. 1, pp. 37–44, Mar. 1998.

[11] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1204–1216, Jul. 2017.

[12] T. Shamir, "How should an autonomous vehicle overtake a slower moving vehicle: Design and analysis of an optimal trajectory," *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 607–610, Apr. 2004.

[13] N. Murgovski and J. Sjöberg, "Predictive cruise control with autonomous overtaking," in *2015 54th IEEE Conf. Decision Control (CDC)*, Dec. 2015, pp. 644–649.

[14] J. Karlsson, N. Murgovski, and J. Sjoberg, "Computationally efficient autonomous overtaking on highways," *IEEE Trans. Intell. Transp. Syst.*, early access, Jul. 31, 2019, doi: 10.1109/TITS.2019.2929963.

[15] H. Andersen, W. Schwarting, F. Naser, Y. H. Eng, M. H. Ang, D. Rus, and J. Alonso-Mora, "Trajectory optimization for autonomous overtaking with visibility maximization," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.

[16] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 2061–2067.

[17] P. Petrov and F. Nashashibi, "Modeling and nonlinear adaptive control for autonomous vehicle overtaking," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1643–1656, Aug. 2014.

[18] D. Gonzalez, V. Milanes, J. Perez, and F. Nashashibi, "Speed profile generation based on quintic Bézier curves for enhanced passenger comfort," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 814–819.

[19] A. Artu nedo, J. Villagra, and J. Godoy, "Real-time motion planning approach for automated driving in urban environments," *IEEE Access*, vol. 7, pp. 180039–180053, 2019.

[20] X. Qian, I. Navarro, A. de La Fortelle, and F. Moutarde, "Motion planning for urban autonomous driving using Bézier curves and MPC," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 826–833.

[21] F. You, R. Zhang, G. Lie, H. Wang, H. Wen, and J. Xu, "Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system," *Expert Syst. Appl.*, vol. 42, no. 14, pp. 5932–5946, Aug. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095741741500216X

[22] M. Fu, K. Zhang, Y. Yang, H. Zhu, and M. Wang, "Collision-free and kinematically feasible path planning along a reference path for autonomous vehicle," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 907–912.

[23] G. Tanzmeister, D. Wollherr, and M. Buss, "Grid-based Multi-Road-Course estimation using motion planning," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 1924–1935, Apr. 2016.

[24] F. V. Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, *Driving with Tentacles—Integral Structures for Sensing and Motion*. Berlin, Germany: Springer, 2009, pp. 393–440, doi: 10.1007/978-3-642-03991-1_10.

[25] S. Kammel, ''Team AnnieWAY's autonomous system for the 2007 DARPA urban challenge,'' *J. Field Robot.*, vol. 25, no. 9, pp. 615–639, 2008, doi: 10.1002/rob.20252.

[26] H. Mouhagir, V. Cherfaoui, R. Talj, F. Aioun, and F. Guillemard, ''Using evidential occupancy grid for vehicle trajectory planning under uncertainty with tentacles,'' in *Proc. IEEE 20th Int. Conf. Intell. Transp.*, Yokohama, Japan, Oct. 2017, pp. 1–7. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01576974

[27] J. Nilsson, M. Brannstrom, E. Coelingh, and J. Fredriksson, ''Lane change maneuvers for automated vehicles,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1087–1096, May 2017.

[28] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, ''Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints,'' *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 952–964, Feb. 2017.

[29] S. Dixit, U. Montanaro, M. Dianati, D. Oxtoby, T. Mizutani, A. Mouzakitis, and S. Fallah, ''Trajectory planning for autonomous high-speed overtaking in structured environments using robust MPC,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2310–2323, Jun. 2020.

[30] C. Wei, R. Romano, N. Merat, Y. Wang, C. Hu, H. Taghavifar, F. Hajiseyedjavadi, and E. R. Boer, ''Risk-based autonomous vehicle motion control with considering human driver's behaviour,'' *Transp. Res. C, Emerg. Technol.*, vol. 107, pp. 1–14, Oct. 2019.

[31] D. C. K. Ngai and N. H. C. Yung, ''A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 509–522, Jun. 2011.

[32] M. Kaushik, V. Prasad, K. M. Krishna, and B. Ravindran, ''Overtaking maneuvers in simulated highway driving using deep reinforcement learning,'' in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2018, pp. 1885–1890.

[33] D.-F. Xie, Z.-Z. Fang, B. Jia, and Z. He, ''A data-driven lane-changing model based on deep learning,'' *Transp. Res. C, Emerg. Technol.*, vol. 106, pp. 41–60, Sep. 2019.

[34] F. Garrido, D. González, V. Milanés, J. Pérez, and F. Nashashibi, ''Optimized trajectory planning for cybernetic transportation systems,'' in *Proc. 9th IFAC Symp. Intell. Auton. Veh. (IAV)*, 2016, pp. 1–6.

[35] F. Garrido, D. Gonzalez, V. Milanes, J. Perez, and F. Nashashibi, ''Real-time planning for adjacent consecutive intersections,'' in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1108–1113, [Online]. Available: https://hal.inria.fr/hal-01356706

[36] D. J. Walton, D. S. Meek, and J. M. Ali, ''Planar G2 transition curves composed of cubic Bézier spiral segments,'' *J. Comput. Appl. Math.*, vol. 157, no. 2, pp. 453–476, Aug. 2003. http://www.sciencedirect.com/science/article/pii/S0377042703004357

[37] D. Ferguson, T. M. Howard, and M. Likhachev, *Motion Planning Urban Environments*. Berlin, Germany: Springer, 2009, pp. 61–89, doi: 10.1007/978-3-642-03991-1_2.

[38] A. Elfes, ''Using occupancy grids for mobile robot perception and navigation,'' *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.

[39] P. Merdrignac, E. Pollard, and F. Nashashibi, ''2D laser based road obstacle classification for road safety improvement,'' in *Proc. IEEE Int. Workshop Adv. Robot. Social Impacts (ARSO)*, Jun. 2015, pp. 1–6.

[40] J. E. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro, ''Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 438–450, Sep. 2008.

[41] G. Yang, H. Xu, Z. Tian, and Z. Wang, ''Vehicle speed and acceleration profile study for metered on-ramps in california,'' *J. Transp. Eng.*, vol. 142, no. 2, Feb. 2016, Art. no. 04015046.

[42] L. Roldao, J. Perez, D. Gonzalez, and V. Milanes, ''Description and technical specifications of cybernetic transportation systems: An urban transportation concept,'' in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Nov. 2015, pp. 176–181.

[43] D. Gonzalez, J. Perez, R. Lattarulo, V. Milanes, and F. Nashashibi, ''Continuous curvature planning with obstacle avoidance capabilities in urban scenarios,'' in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1430–1435.

FERNANDO GARRIDO received the Ph.D. degree in automation and real-time systems from Mines ParisTech PSL, France, in 2018. His Ph.D. work was done at Inria and Vedecom research institutes, France. Since 2018, he has been with the Driving Assistance Research Team, Valeo, France. His research interests include motion planning and decision making systems.

LEONARDO GONZÁLEZ received the B.Sc. degree in electronic engineering from the Universidad Simón Bolívar University, Caracas, Venezuela, in 2014. He is currently pursuing the Ph.D. degree in robotics, automation, and control with the University of the Basque Country, Bilbao, Spain. He joined the Automated Driving Team, Tecnalia Research and Innovation, in 2017. His research interests include fuzzy logic, decision making and planning in automated vehicles, and risk assessment.

VICENTE MILANÉS received the Ph.D. degree in electronic engineering from the University of Alcalá, Madrid, Spain, in 2010. He was with the AUTOPIA Program at the Center for Automation and Robotics (UPM-CSIC), Spain, from 2006 to 2011. In 2014, he joined the RITS Team, INRIA, France. Since 2016, he has been with the Research Department, Renault, France. His research interest includes multiple aspects in the autonomous vehicle field. Then, he was a recipient of the two-years Fulbright Fellowship at California PATH, UC Berkeley.

JOSHUÉ PÉREZ RASTELLI (Member, IEEE) received the B.E. degree in electronic engineering from Simon Bolıvar University, Venezuela, in 2007, and the M.E. and Ph.D. degrees from the University Complutense of Madrid, in 2009 and 2012, respectively. He is currently a Research Leader on Automated Driving at Tecnalia Research and Innovation, since 2015. He has more than 13 years of experience in the intelligent transportation system field, and more than 130 publications related to automated driving and ADAS.

FAWZI NASHASHIBI (Member, IEEE) received the master's degree in automation, industrial engineering, and signal processing from the Laboratory for Analysis and Architecture of Systems/Centre Nationnal de la Recherche Scientifique (LAAS/CNRS), Toulouse, France, the Ph.D. degree in robotics from the LAAS/CNRS Laboratory, Toulouse University, and the HDR Diploma (accreditation to research supervision) degree from Pierre and Marie Curie University (Paris VI), Paris, France. Since 1994, he has been a Senior Researcher and the Program Manager with the Robotics Center, Mines ParisTech, Paris. Since 2010, he has been a Senior Researcher and the Program Manager with the RITS Team, Inria, Paris-Rocquencourt, France.