



HAL
open science

The Tornado Project: An Automated Driving Demonstration in Peri-Urban and Rural Areas

Vicente Milanés, David González, Francisco Navas, Imane Mahtout, Alexandre Armand, Clement Zinoune, Arunkumar Ramaswamy, Farid Bekka, Nievsabel Molina, Emmanuel Battesti, et al.

► **To cite this version:**

Vicente Milanés, David González, Francisco Navas, Imane Mahtout, Alexandre Armand, et al.. The Tornado Project: An Automated Driving Demonstration in Peri-Urban and Rural Areas. IEEE Intelligent Transportation Systems Magazine, 2021, 10.1109/MITS.2021.3068067 . hal-03400571

HAL Id: hal-03400571

<https://hal.science/hal-03400571>

Submitted on 25 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TORNADO project: An automated driving demonstration in peri-urban and rural areas

Vicente Milanés¹, David González², Francisco Navas², Imane Mahtout¹, Alexandre Armand¹, Clement Zinoune¹, Arunkumar Ramaswamy¹, Farid Bekka¹, Nievsabel Molina², Emmanuel Battesti³, Yvon Kerdoncuff³, Carlos Guindel⁴, Jorge Beltrán⁴, Irene Cortés⁴, Alejandro Barrera⁴ and Fernando Garcia⁴

Abstract—This paper presents the results of a two-week robot-taxi service demonstration in peri-urban and rural areas. A fully robotized Renault ZOE was available for general public use in the Rambouillet Territory in France. The driving zone included several complex scenarios as two-way narrow road driving, a tunnel crossing with lane reduction from two-way road up to a single narrow lane or roundabouts, allowing to evaluate the maturity of the vehicle for such application. This paper describes the scientific and technical development especially from perception, navigation and control point of view to carry out such demonstration. Results indicate that even if the vehicle was able to autonomously navigate through peri-urban and rural areas, there are still some technical challenges that limit its integration with the transport system.

Index Terms—Intelligent vehicles, perception systems, trajectory planning, vehicle dynamics, intelligent transport systems.

I. INTRODUCTION

Automated vehicles progress has significantly advanced in the last years, moving from the first Advanced Driving Assistance Systems (ADAS) as lane departure warning [1] or cruise control systems [2] to highway lateral and longitudinal automated vehicle control (i.e. Tesla AutoPilot system¹). This pioneering production system has demonstrated the ability to introduce advanced supervised technologies in commercial vehicles. L2-based automated highway systems represent the first step toward automated vehicles in more complex traffic scenarios, demanding more intelligent systems.

Urban and rural areas remain as the most challenging scenarios because of the potential changing situations that can be found compared with highway scenarios [3]. There are just a few worldwide demonstrations dealing with these complex environments. VisLab’s BRAiVE autonomous car carried out a demonstration [4] through Parma streets in Italy mixing rural, highway and urban driving scenarios. Results highlighted that complex scenarios such as roundabouts are still an open research challenge. Also in Europe, a cooperation

¹V. Milanés, I. Mahtout, A. Armand, C. Zinoune, A. Ramaswamy and F. Bekka are with the Research Department, Renault SAS, Guyancourt, France {vicente.milanes, imane.mahtout, alexandre.armand, clement.c.zinoune, arunkumar.ramaswamy, farid.bekka}@renault.com

²D. González, F. Navas and N. Molina are with AKKA Technologies, Guyancourt, France. {david-enrique.gonzalez-bautista, francisco-martin.navas-matos, Nievsabel-Fireley.molina-ramos}@akka.eu

³E. Battesti and Y. Kerdoncuff are with ENSTA Paris, Palaiseau, France. {emmanuel.battesti, yvon.kerdoncuff}@ensta-paris.fr

⁴C. Guindel, J. Beltrán, I. Cortés, A. Barrera and F. García are with the Intelligent Systems Lab, Universidad Carlos III de Madrid, Leganés, Spain. {cguindel,jbeltran,irecorte,alebarre,fegarcia}@ing.uc3m.es

¹<https://www.tesla.com/support/autopilot>



Fig. 1. Tornado project key scenarios demonstration. Left graphic corresponds to a roundabout entrance. Centre graphic depicts a narrow two-way road with a gutter. Right graphic plots a non-visibility tunnel crossing with a reduction from two-way to a single lane.

between Daimler and Karlsruhe Institute of Technology (KIT) showed the ability to deal with complex situations in structured environments. Results of the autonomous trip from Mannheim to Pforzheim in Germany were presented in Ziegler et al. [5]. Similar to BRAiVE demonstration, they concluded that overall performance is significantly downgraded with respect to a human driver. Multiple private companies in the United States as Waymo, Zoox and Aptiv are testing their systems in limited well-structured urban environments but there are no scientific papers about these works.

Rural areas have received little attention [6] but they represent an excellent test bench for automated driving technologies. Conditions are usually degraded with respect to urban zones (i.e. poor or no lane markings available, patched pavement or narrowed roads among others) and they account for more than 97% of the United States’ land area [7], highlighting the importance of demonstrating (and validating) automated technologies on these zones. Additionally, autonomous mobility systems can be a potential solution for commuters in low-dense areas with a little or no public transport options.

This paper presents the results of an automated vehicle demonstration in Rambouillet (France) within the framework of the Tornado project, covering urban and rural areas. It describes the in-vehicle architecture for providing automated vehicle capabilities, focusing later on the development of perception, planning and control stages to overcome the technical challenges. Next subsection details the key driving scenarios encountered during the two-week demonstration.

II. TECHNICAL CHALLENGES

Peri-urban and rural areas present multiple challenging situations. Depending on the complexity, the associated challenges are divided as follows:

- **Nominal driving** corresponds to regular traffic situations. It includes standard width lanes, good lane-marking quality or well-identified pedestrian crossings. The way of handling these situations matches with the general

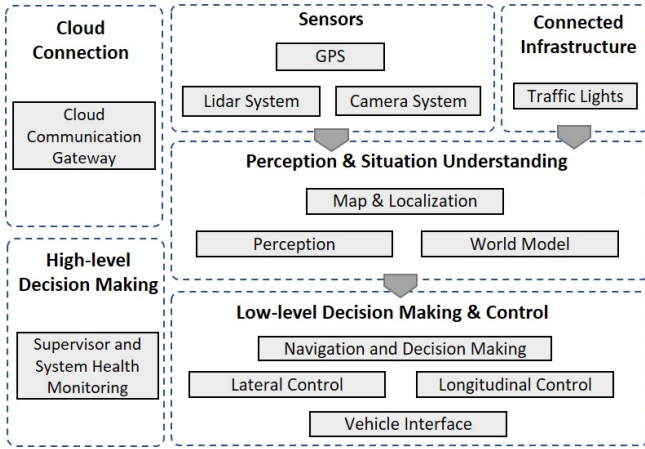


Fig. 2. Functional components of Tornado Vehicle

architecture description and it serves as starting point for all specific use case scenarios.

- **Roundabout driving** is getting more and more popular replacing crossroads because it increases traffic fluidity, reducing 37% overall collisions [8] (see left image in Fig. 1). However, there is no production system available to deal with these complex scenarios and just a few demonstrations have tackled this problem, recognizing its complexity [4]. A description of the perception-planning-control seamless integration to deal with these use cases is presented in the coming sections.
- **Narrow two-way road driving** represents a common scenario in rural areas. Highway lanes width are usually set at 3.5 meters in Europe with reductions up to 3m in urban areas to discourage high speeds in city centers [9]. When it comes to rural areas (see center image in Fig. 1), there are two main **constraints** to take into account: 1) standard vehicle width has increased in recent years from 1.8 to 2.2m (including side-view mirror); and 2) lane width is reduced up to 2.5m in rural areas, increasing the complexity when crossing vehicles in two-way roads.
- **Tunnel driving** constitutes a technical challenge that has been already faced in urban areas. However, additional factors may increase the complexity in these scenarios in rural areas. Specifically, the vehicle crosses a tunnel that includes a lane reduction from two-way road up to a single narrow lane with a sharp turn when exiting (see right image in Fig. 1), adding priority traffic management and control capabilities to classical localization problems in these situations.

III. IN-VEHICLE ARCHITECTURE

Autonomous driving is computationally intensive and requires integrating different functions such as perception, map service and localization, world model, navigation and decision-making, and vehicle control into a unified system that exhibits rational behavior. The arrangement and interaction of different functional components play a crucial role on the robustness and reliability of the vehicle operation. Figure 2 shows differ-

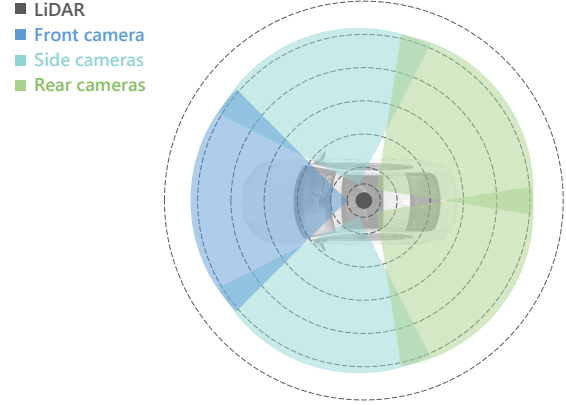


Fig. 3. Sensor setup in Tornado Vehicle

ent components of the vehicle. The following sections briefly explain each of these components:

Map and Localization: This component is responsible for gathering and centralizing all map related information, aiding to localize the vehicle within the available map. The map database stores the map of the operational area in a proprietary format that enables faster access (details are given in subsection IV-B). Communication with map service is implemented in query pattern. Requesting modules send a query to the map service component for receiving relevant information. The localization component provides position, heading, velocity and timing information, together with integrity information.

Cloud Support and Communications Links: This component provides a platform in which autonomous vehicles communicate with cloud server. The system mainly provides three functionalities: 1) Remote tracking and map visualization of all autonomous vehicles, providing them safety alerts and warnings such as, slippery zone or civil work in progress; 2) Remote control during critical situation like priority vehicle approaching; and 3) Connectivity between vehicles and cloud through cellular technology.

Supervisory System: This component monitors and controls all the components to ensure the safety and robustness of the operational modules. It selects the discrete behavior of the components, that is, to determine which behavior each of the functional components must have at each moment. It maintains different operational modes (e.g. autonomous, manual, standby) and sends signals to the appropriate operational components to switch their states accordingly.

Perception, World Model, Navigation and Decision Making and Vehicle Control components are detailed in the next sections.

IV. PERCEPTION AND WORLD MODEL

This section details the environment detection and understanding in two subsections (perception and world model respectively) accordingly to the in-vehicle architecture.

A. Perception

This section describes the sensor suite configuration and the algorithms designed to provide reliable detection of the

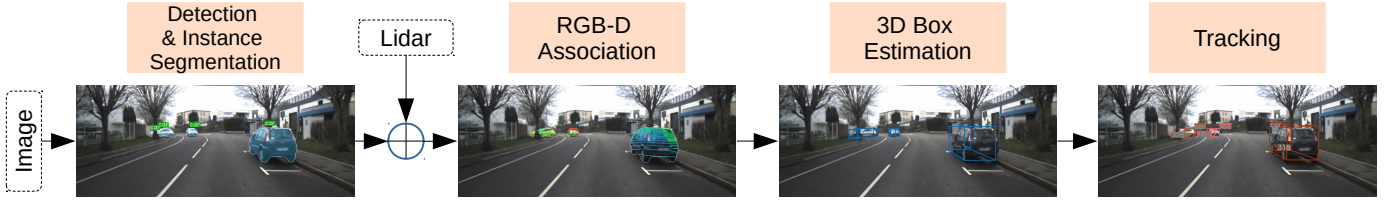


Fig. 4. Pipeline for 3D object detection. Objects are detected in the images from the cameras. Later, LiDAR data is included to infer the geometrical properties of the objects. Finally, all the detections are merged, and a tracking stage adds temporal consistency across consecutive frames.

obstacles in the surroundings of the vehicle. As depicted in Fig. 3, the system configuration is based on five evenly distributed cameras and a 32-layer LiDAR located in the center of the roof rack, providing a 360° field of view.

The perception solution stems from the one presented in [10], where an analysis of the performance of the complete pipeline is provided. In brief, the information provided by the sensors is fused through a low-level fusion approach for obstacle detection, classification, 3D box estimation, and tracking. The system obtains vision-based detections from a robust state-of-the-art framework and feeds them to a 3D box estimation method that makes use of LiDAR information to infer their size and location. Finally, the tracking algorithm exploits this spatial reasoning to add time consistency and enhance the reliability of the final detections. An overview of the pipeline is depicted in Fig. 4.

Vision-based detection and classification: Detection and classification are based on computer vision approaches to take advantage of the feature-rich appearance information delivered by the cameras. Due to the multi-camera setup, the system can identify agents in the entire range of interest, without blind spots. Furthermore, enhancing the detections with a pixel-wise semantic mask was proven useful to improve data association between both modalities (images and LiDAR data), provided that accurate extrinsic calibration is available. The Mask R-CNN framework [11] was adopted to perform both tasks jointly, as it combines the high detection accuracy featured by the well-known Faster R-CNN detector [12] with the ability to perform instance segmentation.

Mask R-CNN is a two-stage method that, unlike other similar approaches, can achieve real-time frame rates using deep convolutional networks. It accepts an RGB image as an input and feeds it through a set of stacked convolutional layers (backbone) responsible for extracting features that are shared for the subsequent tasks of detection, classification, and segmentation. Then, these features are used in the first stage to identify Regions of Interest (ROIs) in the image through an RPN (Region Proposal Network). Finally, in the second stage, features are pooled from these ROIs and propagated to successive dedicated layers to perform the final inference tasks. In this work, the ResNet-50 model [13] was adopted as the backbone because of its compelling performance and limited computational cost. It was also endowed with an FPN (Feature Pyramid Network) structure [14] to generate additional feature maps at different scales, allowing the identification of distant objects. The system’s speed-accuracy tradeoff is controlled through the downsampling of the input images, tuned to enable

proper identification of the objects in the intended peri-urban and rural scenarios.

The outcome provided by this stage is a set of 2D bounding boxes with information about the category (e.g., car, person, etc.), each containing a pixel-wise mask that defines the contour of the object. Note that a different set of detections is obtained for each camera.

3D box estimation: Once the objects are identified within the images, LiDAR data is included in the pipeline to provide geometrical information. The high accuracy featured by LiDAR scanners in measuring distances justifies the use of this modality at this stage.

The semantic masks from Mask R-CNN allow identifying the LiDAR data (represented as 3D points) that belong to each obstacle. At this point, accurate extrinsic calibration becomes critical to ensure proper data association. In this case, an evolution of [15] suitable for monocular devices was used to obtain the extrinsic parameters representing the relative position between sensors.

Frustum PointNet approach [16] was used to retrieve the geometrical structure behind the raw LiDAR representation of the obstacle. This approach is able to provide the size, location, and orientation of the obstacles using LiDAR information, and taking into account the missing parts due to occlusions or perspective. The use of the pixel-wise segmentation allows our approach to provide the 3D estimation stage with a more precise segmentation of the point cloud of every instance, minimizing the number of spurious points in the input data. The expected outcome of this stage is a set of positioned cuboids representing the real geometry and distance from the ego-vehicle to every obstacle.

LiDAR processing is performed as follows. In the first stage, outliers are removed through the use of an instance segmentation PointNet [17] that filters out the few remaining noisy points that do not belong to the obstacle. Later, the 3D bounding box of the obstacle is computed through a two-step phase. The first step provides an estimate of the center of the obstacle through the T-Net network (based on the PointNet architecture). Once obtained, the points belonging to the obstacle are translated into this new reference. In the final step, a new PointNet-like network is used for the computation of the final oriented 3D box of the obstacles.

As with the detection part, a different set of 3D boxes is obtained for each camera. At the end of this stage, all of them are expressed in a common coordinate system and merged through a greedy NMS (Non-Maximum Suppression) procedure that prevents eventual duplications by unifying

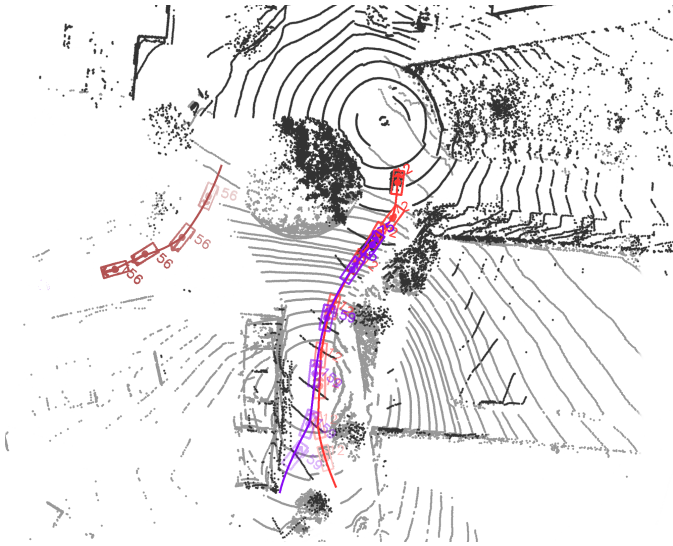


Fig. 5. Bird's eye view representation of a roundabout scenario according to the information provided by the perception system. This example depicts three different obstacles and their trajectories over several consecutive frames, drawn over raw LiDAR data in different colors and with a unique ID each. Note that the trajectories overlap as they correspond to different frames.

objects that overlap in a top-down view of the environment.

Tracking: When the whole set of 3D detections is available and expressed in the same frame, the next step is to add consistency to the detection through the tracking stage. The tracking algorithm used in the perception pipeline is composed of three stages. The first one is the movement estimation, based on an Unscented Kalman Filter [18] with different movement models for pedestrians and vehicles. The second step is the ego-vehicle movement compensation, which is required to correct the misalignments on the subsequent detections due to the movement of the vehicle. This is done through the use of the GPS/INS system available in the vehicle. Finally, the last stage, data association, correlates the detection in the current cycle with the set of already tracked agents, adding new instances whenever it is necessary.

As a result of the tracking stage, the temporal coherence between consecutive frames can be exploited, avoiding inconsistencies and smoothing out any eventual deviation. As an example, Fig. 5 shows the output of the perception pipeline over time in a situation with three agents being tracked in consecutive frames. Experimental results in [10] prove the adequacy of the solution, which can populate the World Model with stable detections that are accurately localized.

B. World Model

The perception system provides information about the surrounding obstacles but without considering context information. By combining perception output with information stored in the digital map, the World Model system can add scene understanding capabilities (see Fig. 2).

High Definition Vector Map: The operation area is described through a high definition vector map that was created specially for the experimentation. This map is a general representation of the road network and includes topological, geometrical and

semantic information. It comprises details about the interactions that may exist between roads at a lane level detail. The center and the boundaries of each lane are also stored in the map with centimeter accuracy (for more details see §2.4.5 in [19]). In addition, semantics are associated with each lane as attributes representing information such as speed limit, marking types, driving directions, etc. In France, and especially in rural zones, markings and signs related to intersections may be challenging to perceive and understand. All information about intersection type, shapes or priority order is stored in the map. Traffic lights location and association to lanes leading to intersections are also stored in the map. All this knowledge is compiled as a geographical database within the vehicle. The Map Service system manages this database and makes it available to other vehicles' systems upon request (see Fig. 2). The use of this map is threefold: 1) it allows computing the itinerary to reach the desired destination; 2) it supports the navigation module for real time computation of the trajectory while operating in autonomous mode; and 3) it provides context in which the vehicle is navigating.

Obstacles Contextualization: The map is used to contextualize the perception system output. This contextualization aims to filter the perceived objects by labelling them as *pertinent* or *not pertinent*. There is no additional obstacle tracking performed within the World Model. This is a part of the situation understanding process as it is defined in [20]. For this purpose, pedestrians and vehicles are considered differently. All pedestrians are automatically labelled as *pertinent*, wherever they can be located within the navigable space. On the contrary, vehicles and other obstacles are considered as *not pertinent* if they are not located on a space that is navigable by the ego vehicle.

Connected Traffic Lights: For this experimentation, the detection of traffic lights only relies on Vehicle to Infrastructure (V2I) communication as the embedded perception sensors do not return traffic lights information. A V2I setup available on the market was used; it is composed of a pair of construction site traffic lights equipped with a Road Side Unit (RSU) which transmit the state of the lights every 100ms. In addition, the vehicle is equipped with an On Board Unit (OBU) to receive the information. The V2X devices use the 802.11p communication standard [21]. The state of the traffic lights is sent by the infrastructure through a Signal Phase and Timing (SPaT) standard message [22]. This message stores for each light information about its state, the time before state change, the ID, etc. When approaching an area equipped with V2I, the vehicle receives information about several traffic lights. After matching the perceived traffic lights with the map (through the traffic light ID, available in the SPaT message and in the map), it is straightforward to identify which traffic light is *pertinent* for the vehicle decision making and navigation module.

V. TRAJECTORY GENERATION

The trajectory generation process takes into account a set of high level requirements related to regulatory rules, comfort and mission purposes. It aims at providing a safe and comfortable trajectory in real-time to the control while navigating to the

user-defined destination. The process is divided into two steps to limit computational effort, namely decision making in which the appropriate maneuver is chosen; and trajectory planning in which an optimized short term trajectory is computed and transmitted to the control system. Moreover, the autonomous vehicle presented in this work aims at providing a mobility service within an operational area. When starting a new mission, a road level plan is computed to determine the shortest route to reach the requested destination and an evaluation of the best *a priori* trajectory is performed. Decision making and trajectory planning are then computed in real time based on this initial driving plan and detected obstacles.

A. Decision Making

General concept: The decision making process consists of evaluating the consequence of each obstacle and traffic light perceived at a given process time onto the initial driving plan. This driving plan is then modified according to the expected interaction between the ego vehicle and the detected obstacle in order to maintain a safe and comfortable behaviour. The expected interaction is formulated based on the relative obstacle position (e.g. same lane as ego vehicle, opposite lane, partially on ego lane, side walk), direction (i.e., moving along and backward or forward, moving perpendicular and to or away from the driving plan) and type (e.g. pedestrian, vehicle). The maneuver associated with each perceived obstacle is established in an expert system manner. By doing so, we keep computational complexity low and get a deterministic behaviour. Every maneuver adds constraints on the longitudinal (slow down, follow, stop) and lateral (avoid) dimensions which are taken into account in the trajectory planning process described in section V-B.

Intersection management: Intersections require a special treatment because of their complexity. Having this in mind, an extra set of constraints is designed to handle intersection crossing. They consider not only the intersection detection thanks to the digital map but also their priority rules according to the interactions with other vehicles.

A lane-level description of the road network is provided by the map embedded in the vehicle. In case of intersecting or merging lanes, the map also provides the legal right-of-way between lanes. Finally, the type of intersection is also provided by the map (e.g. yield, stop). Every lane involved in the driving plan is studied by the decision making module to detect future intersection and apply the corresponding maneuver sequence. If a stop or yield intersection is detected along the path, an interest zone containing oncoming lanes is computed. The ego vehicle will therefore stop and wait if an obstacle is detected inside this interest zone.

The list hereafter describes the decision sequence to enter a roundabout as implemented in this work:

- Approaching: A yield-type intersection is detected on the driving plan in a 100 m range. The speed profile is adapted accordingly before reaching the yield sign.
- Waiting (Yielding): Fig. 6 illustrates this and in this example, vehicles are detected in the interest zone. The decision to stop is taken and is kept until the interest zone is free of obstacles.

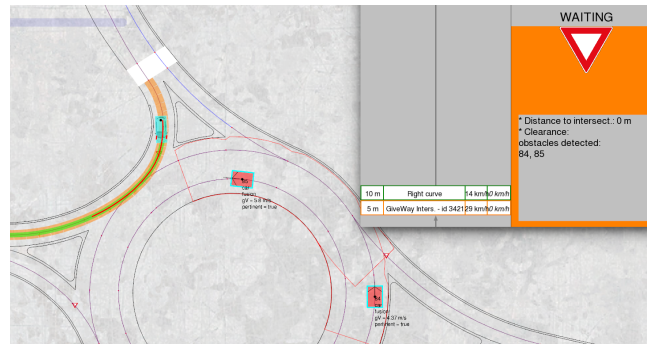


Fig. 6. Intersection management, example of entering a roundabout. Ego vehicle is in blue and detected vehicles are in red. The initial driving plan is represented in green and the area with right-of-way is formed by red polygons. The current status of decision making is shown in the top right window.

- Entering: Arriving at the yield line, the interest zone is (or becomes) free of any detected obstacle. The decision to slowly enter the roundabout is taken. The speed remains slow to be able to stop again if a new obstacle is detected in the interest zone.
- Leaving: The vehicle has passed the yield line. The decision to accelerate is taken in order to reach a comfortable speed inside the roundabout.

This approach of decision making coupled with *a priori* driving plan provided encouraging results through the different scenarios. A transition state machine guarantees that no illegal decision is taken, managing successive intersections and providing flexibility for handling perception limitations. A transition from "entering" to "waiting" and *vice versa* make possible to stop again if a vehicle lately appears in the interest zone. This situation occurred when traffic was partially occluded, when position measurement and type classification errors filtered out the obstacle; and when vehicles were coming at high speed.

B. Trajectory planning

The short-term trajectory of the ego vehicle is computed in real time, including the geometric description of the desired path, lateral error tolerance and associated speed profile.

Space definition: High definition vector map provides a geometric description of each lane which is used to create lateral boundaries (i.e., a driving corridor) limits to the vehicle. A longitudinal boundary is set with respect to the closest obstacle for which the decision system chooses a stop maneuver. Spatio-temporal footprint of each obstacle associated with an avoidance maneuver is computed based on estimated position and speed. The available driving space is therefore reduced according to the set of perceived obstacles.

Path optimization: This component is in charge of optimizing the driving space in the short term horizon. The driving space is sampled into a set of vertices along regularly spaced transverse lines. They are candidates for constructing the path using a set of connected splines as illustrated by Fig. 7. Splines combinations are convenient to compute a path complying with second order geometric continuity [23]. Moreover, closed

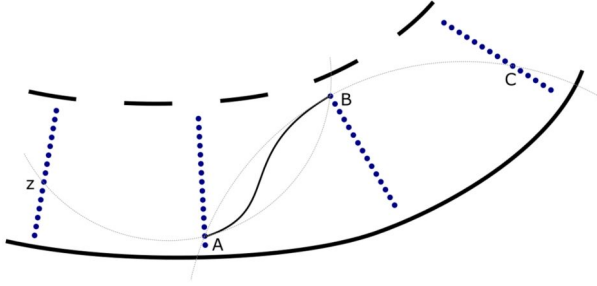


Fig. 7. Sampling of driving space with vertices (blue dots) arranged in layers. An example of a circle spline that connects second and third layer is shown in solid black. Continuity with first and fourth layer are shown in dotted black lines.

form expressions of each circle spline can be found which makes efficient the final sampling of the resulting path.

The full circle spline combination set is computed in order to create a graph of possible solutions for the current computation step. This table demands high computational resources. However, it happens only at the first time step since it is simply updated while the vehicle progresses on its trajectory.

Finally the optimal circle spline compilation is searched using a Dijkstra algorithm [24]. Eq. (1) introduces the costs function for a candidate path \mathbf{P} .

$$J(\mathbf{P}) = w_p p(\mathbf{P}) + w_d d(\mathbf{P}) + w_o \frac{1}{o(\mathbf{P}) + \varepsilon} \quad (1)$$

The cost function is composed of three terms in order to take into account:

- Distance to initial driving plan $p(\mathbf{P})$. Cost increases with area between candidate and initial driving plan.
- Derivative similarity $d(\mathbf{P})$. Cost increases with the difference between candidate's derivative and initial driving plan derivative. This provides high cost to oscillating candidates.
- Distance to driving space boundaries $o(\mathbf{P})$. High costs are associated with candidates that make the vehicle getting close to the boundaries. The physical vehicle's dimensions are taken into account. ε stands for a small positive number (set to 10^{-6} here).

w_p , w_d and w_o are scalar parameters to adapt the relative impact of each term on the total candidate cost and were set to 0.1, 1 and 25 respectively for this experimentation.

Dijkstra search algorithm explores in priority the less cost solution. This results in fast solution convergence if the initial driving plan is still convenient at the current time step. However, more candidates are evaluated if current constraints make the initial driving plan infeasible which increases significantly the computation time. If no solution is found, the previous solution is kept but the associated speed profile sharply falls to make the vehicle brake promptly.

Speed profile: The final step of trajectory planning consists in associating a speed profile to the optimal path considering road geometry and vehicle dynamic constrains (i.e. maximum allowed speed or desired lateral acceleration in turns). Initial driving plan provides a first skeleton on which maneuvers associated with obstacles apply extra constraints as illustrated

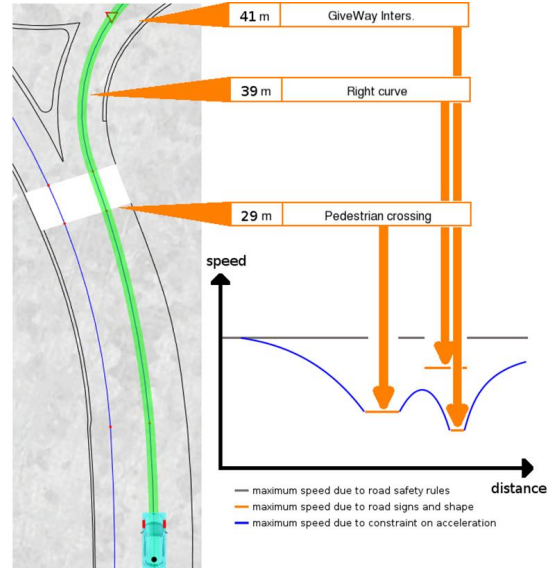


Fig. 8. Speed profile computation. Static and dynamic speed constraints are combined for the current ego vehicle's position.

by Fig. 8. The final speed profile is obtained by filtering and smoothing. The consolidated trajectory is finally sent to the control system as a set of way-points with information about vehicle position, heading, speed, acceleration and road curvature within a speed-based horizon.

VI. VEHICLE DYNAMIC CONTROL

This section describes both lateral and longitudinal controllers. They use as main inputs the positioning and navigation systems' information.

A. Lateral control

First Tornado project demonstration in 2018 [25] proposed two different lateral controllers: 1) a path-tracker controller able to provide a zero tracking error, increasing the safety feeling on-board; and 2) a predicted controller minimizing the control effort whereas keeping the vehicle in the lane (mimicking human behavior). Both lateral controllers are based on a look-ahead dependent minimization of the desired yaw rate [26]. The controller is a second order transfer function with a gain K , and a variable look-ahead distance d . Its objective is to determine the steering wheel angle δ_c according to a yaw rate error w_r calculated as the difference between desired w_d and measured w_v yaw rates. The desired yaw rate w_d is equivalent to the vehicle longitudinal speed v divided by the road curvature ρ in the target point (set at a look-ahead distance d). Notice how K and d were tuned in [25] to achieve different behavior for the tracker K_1 and the predicted controllers K_2 . K_1 controller was optimized to perfectly follow the navigation path, providing a safety feeling to the passenger in terms of precision. However, it exhibited an aggressive behavior in roundabouts and lane changes. The main reason behind this undesirable behavior was that the path tracker controller was tuned to be a fast controller. On the contrary, K_2 was a relaxed version of the path tracker

controller, taking advantage of the driving corridor provided by the navigation system. It was optimized to be slower, less precise, but exhibiting less overshoot in lane changes and roundabouts. The predicted controller was more comfortable but the tracking feeling was downgraded with respect to K_1 .

Both controllers are here implemented in a single control structure based on users' feedback during the 2018 demonstration (see [25] for details). This control structure is based on the Youla-Kucera parametrization [27].

Vehicle lateral model G and both controllers K_1 and K_2 are factorized as the product of stable left and right coprime transfer functions (see [28] for more details) :

$$\begin{aligned} G &= NM^{-1} = \tilde{M}^{-1}\tilde{N} \\ K_1 &= UV^{-1} = \tilde{V}^{-1}\tilde{U} \\ K_2 &= U'V'^{-1} = \tilde{V}'^{-1}\tilde{U}' \end{aligned} \quad (2)$$

The Youla-Kucera parameter Q permits to describe the class of all stabilizing controllers; but it is here used for connection of controllers K_1 and K_2 : $Q = \tilde{V}'(K_2 - K_1)V$. The final control structure is expressed as follows:

$$K(\gamma Q) = (U + \gamma QM)(V + \gamma QN)^{-1} \quad (3)$$

where $\gamma \in [0, 1]$ is computed online with respect to the corridor width provided by the navigation system and the lateral error. Different values of γ activate different level of controllers K_1 and K_2 ($\gamma = 0$ is K_1 , $\gamma = 1$ is K_2 , and $\gamma \in [0, 1]$ is a mixed behavior between both controllers). Figure 9 clarifies how the γ -based decision-making system works together with the Youla-Kucera based control structure. The supervisor module checks first the corridor width, and then the lateral error, providing the value of γ that activates the corresponding controller:

- In the roundabout entrance and exit, the predicted controller is activated since the driving corridor is wide, allowing significant lateral errors. This leads to smoother control efforts, having a more comfortable control response.
- In the narrow two-way road and the tunnel, the tracker controller is activated. This is caused by the reduction of the driving corridor, demanding a higher tracking precision.
- In the other areas γ changes gradually between $[0, 1]$ according to the current lateral error [29]. It activates the adequate proportion of each controller and performs a smooth transition between the two previous areas.

B. Longitudinal control

The experimental platform is already equipped with a low-level longitudinal controller. This low-level controller receives speed reference from the navigation system, actuating accordingly over throttle and brake pedals. Reference/output behavior are in blue and red solid lines respectively in Fig. 11. Notice how discontinuities appear in the speed reference (around second 22) and how the low-level controller takes around 400 ms to reach the desired speed.

To deal with such discontinuities, a high level longitudinal controller with better tracking capabilities and robustness was

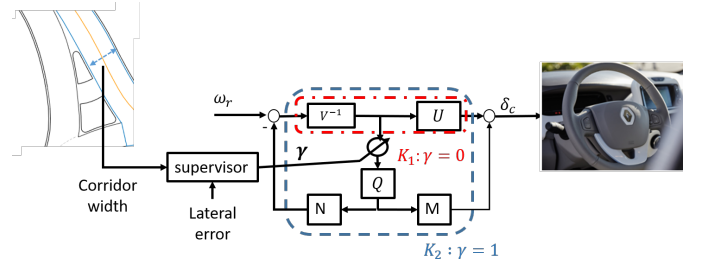


Fig. 9. Vehicle lateral controller

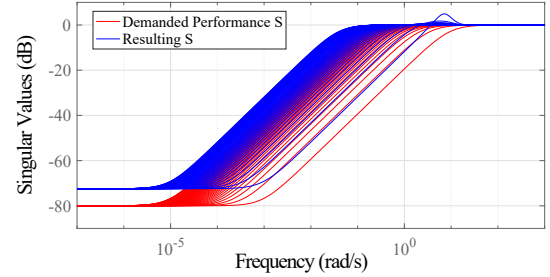


Fig. 10. Sensitivity transfer functions for high level LPV longitudinal controller.

designed. A Linear Parameter Varying (LPV) controller is proposed, having as scheduling parameter the speed reference difference from one sample to the other. Specifically, the low-level controller behavior is considered as an open-loop system, together with a LPV weighting function based on the desired sensitivity of the closed-loop (CL) system. Sensitivity transfer function bandwidth changes with the scheduling parameter, allowing a faster tracking capability when the speed reference change is small (less than $\pm 2 \text{ m/s}^2$), and slowing down the response when the difference is greater. A gridding-based LPV synthesis approach [30] [31] is used for solving corresponding inequalities in a specific scheduling parameter range. Reference and resulting sensitivities are in red and blue in Fig. 10 for the whole scheduling parameter range. Resulting LPV controller behavior is in solid yellow line in Fig. 11. The designed high level controller permits a faster response (200 ms) with better tracking capabilities, filtering out discontinuities in navigation speed reference, and maintaining a comfort acceleration in the desired range ($\pm 2 \text{ m/s}^2$).

VII. EXPERIMENTAL RESULTS

This section describes the experimental tests conducted during two weeks at Rambouillet where 155 people experienced the vehicle that drove more than 560km in autonomous mode. The prototype was a Renault ZOE vehicle that travelled 7.5km per itinerary. Along these kilometers, three scenarios were the main technical challenges in the demonstration: A roundabout, a narrow two-way road and a tunnel (see Fig. 1 for details). Figure 12 presents the vehicle performance zoom-in for the three specific scenarios. Road boundaries are plotted in red, autonomous vehicle position with time mark in yellow and obstacles' position at a given instant in dark red. Figure 13 shows experimental results on the full itinerary whereas

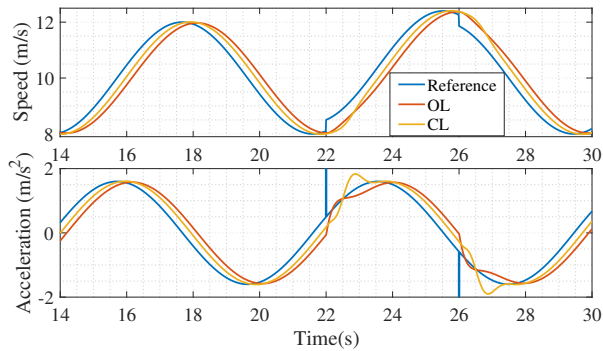


Fig. 11. Longitudinal dynamics comparison. Longitudinal low-level controller (OL) and designed high-level LPV controller (CL).

orange, pink, blue and grey zones represent the roundabout driving, the narrow S-shaped, the tunnel crossing and the U-turn (in manual mode) respectively. The top graph shows the vehicle lateral error. The next one depicts the vehicle steering wheel angle. The following graph points to the steering wheel angle rate. The bottom graph shows the speed reference from the navigation (dotted magenta line) and the current vehicle speed (solid blue line). These graphics correspond to a given test but vehicle performance was highly repetitive in each test.

A. Scenario analysis

Results related to the roundabout scenario are depicted in Fig 12a. The vehicle crosses this area twice: first between seconds 50 and 60 while entering the north roundabout access and driving right to the west exit; and then between seconds 630 and 670, where the vehicle enters the roundabout through the west access and goes north to continue its way back. The performance of the vehicle is visibly good, accurately tracking the planned trajectory (see lateral error in the orange zone on the top graph of Fig. 13), adjusting its speed (second 50) or even stopping when other vehicles are present in the roundabout (second 630) when approaching the entrance (see bottom graph of Fig. 13). The steering angle and the steering wheel rate are kept within the vehicle's limits.

The two-way narrow road driving is shown in a S-shaped section (see Fig 12b). It depicts one of the most difficult scenarios with little tracking error margin. The vehicle travels through this section twice: 1) Between seconds 70 and 90 from right to left in the figure, where the tire is always close to a gutter on the right side; and 2) Between seconds 590 and 610 where a curb is present on the right. The tracking performance can be seen in the top graph of Fig. 13 in the magenta areas. The vehicle lateral error never goes larger than 0.2 m precision, depicting a proper behavior even when turning and maneuvering in narrow two-way roads, keeping steering angle and steering rate vehicle response within vehicle boundaries (see second and third graphs from the top in Fig. 13 respectively). The speed profile (see bottom graph in Fig. 13) indicates how the vehicle adapts its speed before reaching the S-shaped (second 70), decoupling lateral and longitudinal accelerations as requested by users during 2018 demonstration

[25]. Interestingly, the vehicle is following a bicycle (second 600 s) in the way back, modifying in real-time the speed profile accordingly.

Figure 12c presents a bird view of the tunnel scenario. The vehicle also crosses this area twice: first between seconds 260 to 300; and then between seconds 400 to 430. The scenario depicts a one way tunnel where crossing priority is assigned via two coordinated V2I traffic lights at each side of the tunnel. Tunnel boundaries are depicted by the vehicle positions at second 280 on the outward journey and vehicle position at second 420 on the way back. This is one of the most challenging scenarios for the vehicle since it has to first manage the traffic light, then passes through a narrow tunnel, reducing its speed since the positioning system is degraded (i.e. GPS/INS system information quality is reduced, moving to a close-loop model-based positioning system estimator). This is visible in the bottom plot in Fig. 13 between seconds 280 and 420 (blue marked areas), when the speed is significantly reduced whereas the positioning system is recovering its precision, autonomously driving in a degraded mode. Scenario complexity is highlighted in second 290 where the vehicle is exiting the tunnel. There are two stopped vehicle in the opposite direction waiting in the traffic light that have to be perfectly positioned, specially in heading. Additionally, there is a preceding vehicle to follow when traffic light turned green. The top graph in Fig. 13 shows tracking error (including positioning system inaccuracy), where the vehicle exhibits a good performance, keeping the vehicle in the lane. During this time, K_2 controller is fully activated to provide smooth steering wheel changes (see second and third graph in Fig. 13), increasing the comfort on-board. Obstacles are depicted at the exit of the tunnel (second 290), showing two stopped vehicles at the second traffic light and a vehicle in front which passed the tunnel before the ego vehicle. Time between seconds 235 and 260 when the vehicle has zero velocity represents the moment when it's stopped in front of the traffic light. The steering angle remains static at the desired angle and the steering rate goes to zero until traffic light is green and the vehicle goes through the tunnel. The steering signal is continuous, giving a smooth and comfortable vehicle behavior even in full stop situations.

B. Overall performance

Performance out of the challenging zones (see Fig. 13 for details) exhibits also good results and confirms a proper behavior of the vehicle throughout the itinerary. The top graph of Fig. 13 shows that the lateral error is always lower than 0.1 m when out of the painted areas. The steering wheel angle rate (see third graph in Fig. 13) is kept under $200^\circ/s$ (maximum system capabilities) even in challenging conditions as roundabouts or tunnel scenarios. The speed shows a proper tracking (in blue) of the speed trajectory (in red), even when following cyclists (see between seconds 560 and 640 in the bottom part of Fig. 13).

Finally, Fig. 14 presents the acceleration GG diagram. The accelerations were limited to 0.2 g for the lateral axis of the vehicle, 0.2 g for the longitudinal acceleration and

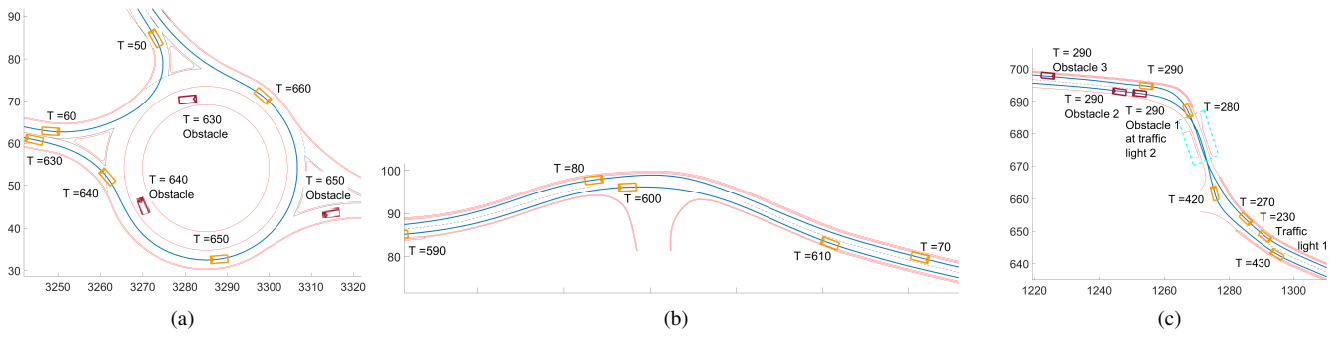


Fig. 12. The vehicle effective path is shown in blue, vehicle position with time mark in yellow, road boundaries are depicted in red, obstacles at a given interaction instant in dark red and road lines are plotted in black. (a) Vehicle performance in a roundabout. (b) Vehicle performance in a narrow “S-shaped” road section. (c) Vehicle performance in the tunnel scenario (the tunnel area is depicted by the teal dashed lines; the traffic lights are located at the front of the ego-vehicle in $T=230$ for the first traffic light and in front of Obstacle 1 in $T=290$ for the second traffic light).

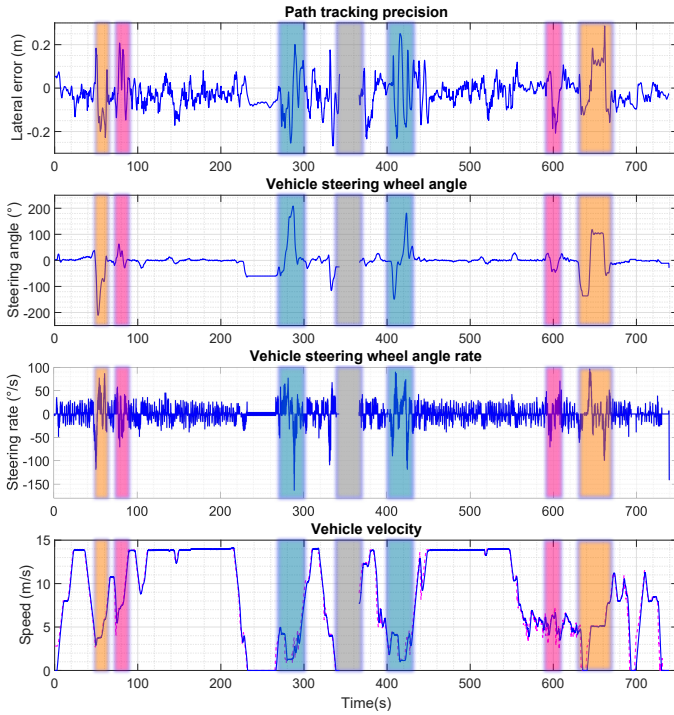


Fig. 13. Vehicle performance for the round trip proposed experience. Orange, pink, blue and grey zones represent the roundabout, the narrow S-shaped, the tunnel crossing and the U-turn (in manual mode) respectively

0.25 g for braking accordingly to users’ feedback in 2018 demonstration [25]. The correspondent acceleration ellipse is depicted in orange in the diagram. The acceleration profile of the vehicle is presented in blue and the acceleration points out of the ellipse are presented in magenta. Interestingly, the vehicle acceleration points are 99.25% inside the circle, thus respecting the acceleration constraints, showing that most situations are properly anticipated by the navigation system.

C. Processing time analysis

One of the challenges in such implementations lies in the capacity for the autonomous vehicle to react correctly to real traffic conditions. Despite their complexity, algorithms involved in autonomous driving shall run with reasonably low cycle times. This section provides some insights about the two

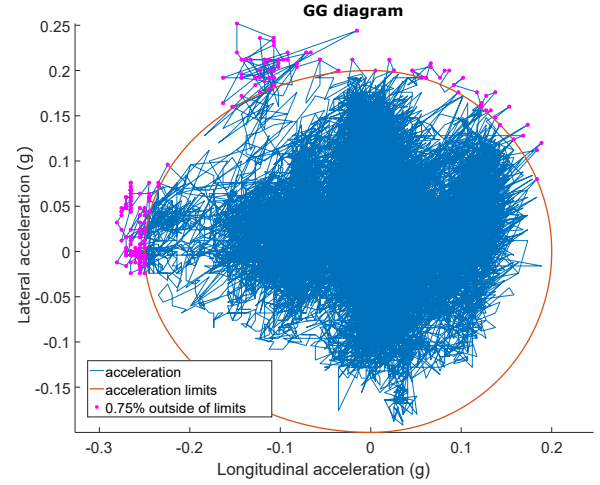


Fig. 14. Acceleration GG diagram throughout the itinerary. The orange circle depicts the acceleration limits for the demonstration.

main processing time consumers, namely Perception system and Navigation and Decision Making system (see Fig. 2). Figure 15 shows the breakdown of the computational time for each module according to their pipelines.

Perception: The configuration of the perception pipeline was focused on efficiency, targeting rates above 10 FPS. The most computationally intensive processes (i.e., object detection and 3D box estimation) are carried out in parallel for all the cameras. As shown in Fig. 15, measurements made during a journey yield an average run time of 63.0 ms (with 6.8 ms of standard deviation) for the detection stage and 21.3 ms (with 6.8 ms of standard deviation) for the 3D box estimation part. The run time of the RGB-D association step, which is in between the previous two, is almost negligible, as is the case with the final tracking stage (they take around 1.5 ms each). The latter requires, however, a previous procedure to express the detections from all the cameras in the LiDAR reference frame and perform NMS that takes 6.9 ms (3.3 ms of standard deviation).

Due to the design of the inference frameworks, these run times depend on the number of objects found in the environment, although this dependency is reasonably limited (as evidenced by the low standard deviations). Values reported

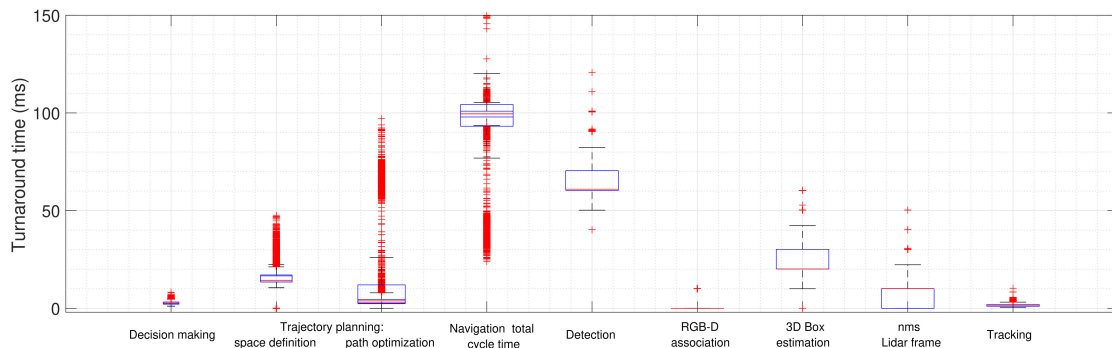


Fig. 15. Breakdown of computation times in the navigation and perception pipeline. The first four boxplot are part of the Navigation and decision making block. The other five boxplots depict the Perception module performance.

above correspond to a relatively crowded scenario with 5.7 obstacles per camera; in the extreme case where 24 agents are found per camera, the detection stage reaches a maximum of 84.9 ms, and the 3D box estimation one, of 45.7 ms. Even though the sum of times for each stage in a cycle is often above 100 ms, several frames can be processed at the same time in different parts of the pipeline so that a new cycle can start before the end of the previous one. Therefore, the perception pipeline can sustain average rates of around 10 Hz.

Navigation and Decision Making: As introduced in section V, real time operations of this system is composed of two main steps: Decision Making and Trajectory planning. Figure 15 presents the computation time breakdown at each cycle for a complete user journey. The smallest part of the time budget is the decision making process since it takes an average time of 2.8 ms (with 0.92 ms standard deviation).

The second time consumer is for the initial step of trajectory planning: space definition. This takes an average time of 16.5 ms (with 6.3 ms of standard deviation). This process consists in geometric construction of the navigable space so a large number of operations is required. Moreover, when interactions with other vehicles occur, the process is more complex so computation time increases.

Then, path optimisation takes a similar average time than space definitions (17.9 ms) but has larger variability with a standard deviation of 26.1 ms (see third boxplot in Fig. 15). The computation complexity significantly increases when the vehicle has driven a sufficient distance to create a new vertex layer and all the associated candidate splines (see Fig. 7). These spikes are therefore more rare when the vehicle drives at low speed.

Finally, the complete trajectory is sent to Control system with an average period of 100 ms but with a quite high variability (41.3 ms standard deviation) due to path optimisation as shown in the fourth boxplot of Fig. 15.

VIII. LESSONS LEARNED

The main idea behind the trials and the two-week demonstration with different passengers on-board was to evaluate the need for novel mobility systems in peri-urban and rural areas. A questionnaire designed using a modified version of the Unified Theory of Acceptance and Use of Technology

(UTAUT) model was completed for more than 150 users. Interestingly, people found the system good enough to either replace or cohabit with current transport systems but there were still some remarks on the way the vehicle negotiate some specific interactions. The environment understanding and the decision associated with it remain as the two main challenges to technically overcome in the near future.

Driving rules violation constitutes one of the main blocking points for the system deployment. As example, some drivers did not respect the traffic light installed in the tunnel to deal with visibility problems (one way tunnel and sharp turn when exiting). Even if during informal talks with Rambouillet residents they pointed out that the traffic light in the tunnel was helping to remove an old road safety problem; it created some critical situations with the automated prototype (i.e., both vehicles at the same time in the middle of the tunnel) from people that crossed in red. This kind of situation that for two human-driven vehicles are easy to negotiate demonstrates the complexity when it comes to putting autonomous vehicles on the road.

A clear improvement point (also mentioned in previous autonomous driving demonstrations [4], [5]) was the ability to properly manage roundabouts. Especially in rural areas, two-lane roundabouts allow cutting both lanes, describing almost a straight line for drivers that are used to the roads. This means that a considerably longer detection range is required to handle such drivers, pointing out to the adoption of V2X technologies in those situations to improve vehicle performance towards a more natural response.

Finally, it is worth to mention that perception system was conceived for urban environments (i.e. high interaction with other road agents and moderate speeds). This is clearly linked to the Operation Design Domain (ODD) concept, meaning that the specific sensor set-up should be enlarged to deal with higher speeds (i.e. highway driving).

IX. CONCLUSIONS

The development and deployment of automated transport systems in low-dense areas represents a realistic application domain for autonomous vehicle: less interaction with other road agents (i.e., vehicles, pedestrians or bikes); limited and repetitive driving areas that significantly help in terms of

having an up-to-date digital map with high precision; and a realistic use-case where the autonomous vehicle can replace a classical public transport system or even create it with less constraints. [A final project demo showing the system capabilities can be found in https://www.youtube.com/watch?v=rjkk6R-SKYg](https://www.youtube.com/watch?v=rjkk6R-SKYg)

Next steps will be focused on removing the current technical barriers (i.e. the ability to deal with unexpected circumstances as other vehicles passing the traffic light in red) as well as enlarging the testing area towards a more complete mobility service. These objectives match with users' expectations collected via a questionnaire after experience the ride, demanding even a longer itinerary with more connections.

ACKNOWLEDGMENT

Authors thank ANRT FUI Tornado project to partially fund this work. They also want to especially thank Carole Forte and Rambouillet Territoires for its support on organizing the testing experiments as well as all users that provide their feedback after test.

This paper reflects solely the views of the authors and not necessarily the views of the company they belong to.

REFERENCES

- [1] I. Besselink, S. Achrifi, and H. Nijmeijer, "Lateral vehicle dynamics on rutted roads," in *The IAVSD International Symposium on Dynamics of Vehicles on Roads and Tracks*. Springer, 2019, pp. 1242–1251.
- [2] L. Xiao and F. Gao, "A comprehensive review of the development of adaptive cruise control systems," *Vehicle system dynamics*, vol. 48, no. 10, pp. 1167–1192, 2010.
- [3] T. Luettel, M. Himmelsbach, and H.-J. Wuensche, "Autonomous ground vehicles—concepts and a path to the future," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1831–1839, 2012.
- [4] A. Broggi, P. Cerri, S. Debattisti, M. Laghi, and P. Medici, "Proud—public road urban driverless-car test," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3508 – 3519, 2015.
- [5] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, and E. Kaus, "Making Bertha drive—An autonomous journey on a historic route," *IEEE Intelligent transportation systems magazine*, vol. 6, no. 2, pp. 8–20, 2014.
- [6] A. Soteropoulos, M. Berger, and F. Ciari, "Impacts of automated vehicles on travel behaviour and land use: an international review of modelling studies," *Transport reviews*, vol. 39, no. 1, pp. 29–49, 2019.
- [7] C. Chmielewski, "Self-driving cars and rural areas: The potential for a symbiotic relationship," *Journal of Law & Commerce*, vol. 37, no. 1, pp. 57 – 81, 2019.
- [8] P. Hammond, "2007-2026 highway system plan: High benefit low cost," Washington State Department of Transportation, Tech. Rep., December 2007.
- [9] "Décret n°2019-1082 du 23 octobre 2019 - art. 2: Article r110-2," Sécurité Routière France, Tech. Rep., October 2019.
- [10] J. Beltrán, C. Guindel, I. Cortés, A. Barrera, A. Astudillo, J. Urdiales, M. Álvarez, F. Bekka, V. Milanés, and F. García, "Towards autonomous driving: a multi-modal 360° perception proposal," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.
- [11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.
- [12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2117–2125.
- [15] C. Guindel, J. Beltrán, D. Martín, and F. García, "Automatic extrinsic calibration for lidar-stereo vehicle sensor setups," in *Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 674–679.
- [16] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 918–927.
- [17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.
- [18] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.
- [19] F. Li, "Lane-level vehicle localization with integrity monitoring for data aggregation," Ph.D. dissertation, Compiègne, 2018.
- [20] S. Ulbrich, T. Menzel, A. Reschka, F. Schudt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 982–988.
- [21] D. Jiang and L. Delgrossi, "Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments," in *VTC Spring 2008-IEEE Vehicular Technology Conference*. IEEE, 2008, pp. 2036–2040.
- [22] *ISO/TS 19091:2019. Intelligent transport systems— Cooperative ITS— Using V2I and I2V communications for applications related to signalized intersections*. Standard, International Organization for Standardization. Geneva, CH, Jun. 2019, vol. 2019.
- [23] A. Piazzzi, C. L. Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic G²-Splines for the iterative steering of vision-based autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 27–36, 2002.
- [24] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [25] V. Milanés, F. Navas, D. Gonzalez, and I. Mahtout, "On the passenger acceptance of driverless shuttles," *IEEE Intelligent Transportation Systems Magazine*, InPress 2020.
- [26] H.-S. Tan and J. Huang, "Design of a high-performance automatic steering controller for bus revenue service based on how drivers steer," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1137–1147, 2014.
- [27] H. Niemann and J. Stoustrup, "An architecture for implementation of multivariable controllers," in *American Control Conference, 1999. Proceedings of the 1999*, vol. 6. IEEE, 1999, pp. 4029–4033.
- [28] K. Zhou, J. Doyle, and K. Glover., *Robust and Optimal Control*, P. Hall, Ed., 1996.
- [29] I. Mahtout, F. Navas, D. Gonzalez, V. Milanés, and F. Nashashibi, "Youla-kucera based lateral controller for autonomous vehicle," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3281–3286.
- [30] F. Wu, *Control of linear parameter varying systems*. University of California, Berkeley, 1995.
- [31] P. Apkarian and R. J. Adams, "Advanced gain-scheduling techniques for uncertain systems," in *Advances in linear matrix inequality methods in control*. SIAM, 2000, pp. 209–228.



Vicente Milanés received his Ph.D. degree in electronic engineering from 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. Then, he was awarded with a two-years Fulbright fellowship at California PATH, UC Berkeley. In 2014, he joined the RITS team at INRIA, France. Since 2016, he is with the Research Department at Renault, France. His research interests cover multiple aspects in the autonomous vehicle field.



David González received the Ph.D. degree in real-time computer science, robotics and automation from MINES ParisTech / PSL Research University, France, in 2017. Since then, he has been working with the Intelligent Systems department at AKKA Technologies. His research interest include vehicle automation, motion planning, real-time trajectory generation, vehicle modeling, vehicle control and human-machine shared control systems.



Arunkumar Ramaswamy received his PhD degree in computer science and robotics from ENSTA-ParisTech, France in 2017. Currently he is working as an architect for Autonomous driving projects in research department at Renault. His research interests include mobile robots, robot architectures, computer vision, human-robot interaction and systems engineering. Arunkumar received his dual Master degree in Robotics from Ecole Centrale de Nantes in France (2012) and Warsaw University of Technology in Poland (2011).



Francisco Navas received the Ph.D. degree in real-time computer science, robotics and automation from MINES ParisTech, PSL Research University, Paris, France, in 2018. He is currently working with the Intelligent Systems department at AKKA Technologies, France. His research interests include control of autonomous vehicles, Youla-Kucera parameterization, vehicle modeling, and vehicle-infrastructure cooperation.



Farid Bekka received Engineer Degree from Ecole nationale Supérieure des Telecommunications Paris France in 1984. After few years in developing Expert System and industrial and banking networks, he joined Renault 30 years ago. He has been involved in numerous projects, in manufacturing, sales, engineering, information technology, and connected car. The last six years his works covered autonomous vehicle simulation, and part of 2 autonomous vehicle projects at Research department.



Imane Mahtout received the Ph.D. degree in real-time computer science, robotics and automation from MINES ParisTech, PSL Research University, Paris, France, in 2020. Her research interest include switching control, vehicle dynamics and intelligent transportation systems.



Nievesabel Molina received the B.Sc. degree in Electronics Engineering, with specialty in control systems and mechatronics at Simon Bolívar University - Venezuela, in 2018. She did an internship in autonomous vehicles control with the RITS team at INRIA Paris, France. Currently she is working with the Intelligent Systems department at AKKA Technologies, France. Her research interests includes autonomous vehicles, modeling and vehicle dynamic control and user experience for autonomous system technology.



Alexandre Armand received his Ph.D. degree in computer science and robotics from ENSTA Paristech, France, in 2016. He made his research with the ADAS department at Renault, focusing on risk assessment. Then he joined the Research department at Renault, to work on autonomous driving technologies, with a particular interest for world modeling and situation understanding.



Emmanuel Battesti is an engineer working in the U2IS laboratory of ENSTA (Ecole Nationale Supérieure de Techniques Avancées) since 2010. He holds a master's degree in "photonics and image" from the Louis Pasteur University of Strasbourg, and an engineering degree from the ENSPS (Ecole Nationale Supérieure de Physique de Strasbourg). After a career in the private sector started in 2000, he joined the academic world in 2008 to work on different projects: data fusion for pedestrian tracking, navigation and visual slam, using topometric maps, for indoor robots, etc. Since 2012, he has been working in partnership with Renault on the development of autonomous cars on different aspects: navigation, decision making, teleoperation and simulation.



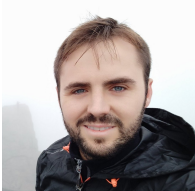
Clément Zinoune obtained his Ph.D. degree in Computer Science and Engineering at the Université de Technologie de Compiègne (UTC), France in 2014. He graduated as a mechanical engineer in 2010 in the UTC. He obtained a master of science on autonomous vehicles dynamics and control in Cranfield University, United Kingdom in 2010. In 2014, he joined the team in charge of the development of autonomous vehicle in Renault S.A., France.



Yvon Kerdoncuff graduated from ENSTA Paris (Institut Polytechnique de Paris) in 2016 with major "Robotics and embedded systems". He has worked as an intern on swarm robotics at IRIDIA (AI lab at Université Libre de Bruxelles) and in the automotive industry in the private sector. In 2017, he joined U2IS lab at ENSTA Paris as robot platform engineer and took part to European Robotics League robotics search and rescue challenge. Then, he joined Emmanuel Battesti to develop software on autonomous car research projects in partnership with Renault. He is now working in embedded systems on sun tracking photo-voltaic power plants.



Carlos Guindel received his Ph.D. degree in Electrical Engineering, Electronics and Automation from Universidad Carlos III de Madrid (Spain) in 2019. He has been with the Intelligent Systems Laboratory (LSI) research group at that university since 2011. His research interests lie in the application of deep learning techniques to the autonomous driving field, including topics such as object detection, pose estimation, and sensor fusion.



Jorge Beltrán received his M.Sc degree in Robotics and Automation from Universidad Carlos III de Madrid, Spain, in 2016. Since 2014, he is part of the Intelligent Systems Laboratory at the same University, where he is currently pursuing his Ph.D. in Electrical, Electronic and Automatic Engineering. His research interests include computer vision, sensor fusion, and 3D object detection for autonomous vehicles.



Irene Cortés received her M.Sc degree in Robotics and Automation from Universidad Carlos III de Madrid, Spain, in 2018. She is currently pursuing her PhD degree in Electrical, Electronic and Automatic Engineering. Since 2016, she has been working with the Intelligent Systems Laboratory at Universidad Carlos III of Madrid. Her research interests include perception systems, object detection, and neural networks.



Alejandro Barrera received his M.Sc. Degree in Robotics and Automation from Universidad Carlos III de Madrid, Spain, in 2018. He is currently pursuing his PhD degree in Electrical, Electronic and Automatic Engineering at the Intelligent Systems Laboratory at Universidad Carlos III de Madrid, within the Department of Systems and Automation Engineering. His research interests include perception systems, deep learning and intelligent transport systems.



Fernando Garcia received his Ph.D. degree in Electrical, Electronic and Automatic Engineering from Universidad Carlos III de Madrid in 2012 where he works as Associate Professor. His research interests are perception and data fusion, mainly applied to vehicles and robotics. He is member of the Board of Governors of the IEEE-ITS Society since 2017 and chair of the Spanish chapter for the period 2019-2020.