# A decision-making architecture for automated driving without detailed prior maps

Antonio Artuñedo, Jorge Godoy and Jorge Villagra

*Abstract*— **Autonomous driving requires general methods to generalize unpredictable situations and reason in complex scenarios where safety is critical and the vehicle must react in a reliable manner. In this sense, digital maps are a crucial component for relating the location of the vehicle and identifying the different road features.**

**In this work, we present a decision-making architecture which does not require detailed prior maps. Instead, OSM is used to plan a global route and an automatically generate driving corridors, which are adapted using a proposed vision-based algorithm. Moreover, a grid-based approach is also applied to consider the localization uncertainty. Those self-generated driving corridors are used by the local planner to plan the trajectories the vehicle will follow. Our approach integrates global, local and HMI components to provide the required functionalities for autonomous driving in a general manner.**

## I. INTRODUCTION

Autonomous vehicles must operate in complex dynamic environments. In order to reach human-level abstract reasoning and react safely even in complex urban situations, autonomous driving requires methods to generalize unpredictable situations and reason in a timely manner [11]. To that end, two elements still need further substantial investigation: world modelling and decision-making in the uncertain [12].

This paper is inspired by previous works on autonomous driving applications without detailed prior maps. In [9], the authors focus on rural environments where it is hard to built and maintain highly detailed maps. Although detailed maps in urban environments might be more feasible, they are still not cost-efficient and without technical problems as minor changes appear constantly, making them not completely reliable. However, the information included in non-highly detailed maps remains more stable. Based on this idea, we propose a decision-making architecture for automated driving that uses information from *OpenStreetMap* (OSM) to obtain a global route and to automatically generate driving corridors, which are then adapted and bounded using different real-time mechanisms: (i) a vision-based lane detection algorithm, and (ii) a probabilistic grid-based corridor reduction considering localization uncertainty. Those self-generated driving corridors are used by a local planner to generate the trajectories the vehicle will follow.

Relying on this approach for local environment modelling, a generic decision-making architecture is proposed, integrating a local planner, a global router and Human-Machine Interface to orchestrate traded control. The ambition of this architecture is not only to provide an approximation for fully autonomous driving, but also to be eventually useful in SAE Levels 2-3, when the driving scene complexity might require human supervision or cooperation.

Contrarily to some previous works on probabilistic decision-making [6, 4], often intractable in real-time, the setting described in this paper inherits the uncertainty from world modelling and produce feasible and comfortable trajectories using cost-effective primitives evaluation and assuming simple patterns for targets motion prediction. The promising behaviour of this architecture will be showcased on a real automated vehicle driving in an urban-like test track.

This remainder of the paper is organized as follows: Section II introduces the proposed architecture. In section III, the road corridor generation from low-fidelity maps together with vision-based adaptation and localization uncertainty consideration are presented. Sections IV and V introduce the local and global planning components, respectively. In section VII the results of the experiments carried out are shown. Finally, in section VIII the conclusions and future work are discussed.

## II. ARCHITECTURE PROPOSAL

The proposed architecture is composed of a combination of modules with different functionalities that provide the whole system with capabilities for autonomous driving. These components are depicted in the functional diagram of Fig. 1, where dashed lines represent event-driven actions and continuous lines correspond to continuously applied actions. On the one hand, global planning and mapping modules address deliberative features such as the computation of the route to reach the destination. On the other hand, local planning modules deal with reactive decisions such as final trajectory generation and obstacle avoidance.

Instead of requiring high-definition maps, our approach uses low-fidelity map data to plan a global route and then automatically generate an extended data structure from OSM that enables the computation of driving corridors. One of the main advantages of OSM is that it is an openly accessible framework to integrate mapping data. However, this may provide a possible source of data inaccuracy. These inaccuracies,

*This work has been partially funded by the Spanish Ministry of Science, Innovation and Universities with National Project COGDRIVE (DPI2017-86915-C3-1-R), and by the European Commission through the Project PRYSTINE (ECSEL-783190-2).

Antonio Artuñedo, Jorge Godoy and Jorge Villagra are with the Centre for Automation and Robotics (CSIC-UPM), Ctra. M300 Campo Real, Km 0,200, Arganda del Rey - 28500 Madrid, Spain. antonio.artunedo, jorge.godoy, jorge.villagra at csic.es

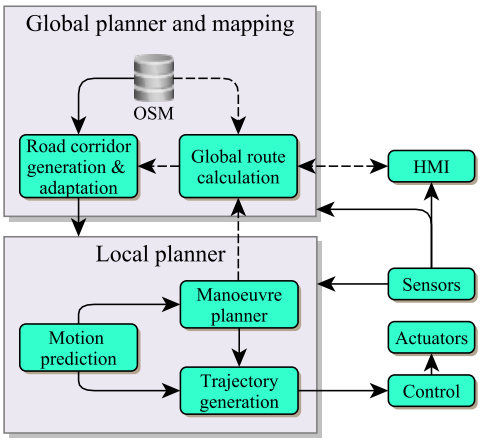



Fig. 1: Functional components of the architecture

together with localization measurement errors, are the main source of conflict when using OSM for automated driving. To mitigate these problems, the proposed architecture includes two complementary strategies employed in the road corridor generation & adaptation module of the architecture: a vision-based road corridor adaptation algorithm and a grid-based approach to consider the localization uncertainty.

Self-generated driving corridors are used by the local planner modules to finally plan the trajectories that the vehicle will follow. To that end, the local planner integrates three different elements: (i) a motion prediction component predicts the future motion of perceived objects, (ii) a manoeuvre planner is responsible for analysing the output of the motion prediction module by checking possible spatio-temporal collisions with the current planned trajectory, and consequently trigger a new trajectory planning request if needed. Based on the predicted motion of nearby objects and the manoeuvre request (iii) a trajectory generation module computes the final path and speed profile. For that purpose, the trajectory generator produces a large number of path candidates within the road corridor that are evaluated. Afterwards, the valid candidate that minimizes a given cost function is selected and a speed profile is computed.

Additionally, a HMI element is added to the architecture for the interaction between the whole system and the vehicle occupants.

## III. ROAD CORRIDOR GENERATION FROM LOW-FIDELITY MAPS

Within the architecture presented in this work, an automatic procedure that expands the OSM definition to generate a better approximation to the real shape of roads is included. To that end, the adjacency among the nodes is analysed for a given area, identifying and classifying all road junctions in order to generate an efficient, accurate polynomial-based map and $G^1$ continuous representation using Bézier curves [5].

The first task of the road corridor generation algorithm is to modify the map description, replacing the straight segments between two consecutive nodes by cubic Bézier curves and automatically adjusting their control points for fitting roads. This is carried out by following a two-stage procedure in which firstly (i) the information related to each node is expanded by analysing the geometric relationship of each node with its neighbours. After that, (ii) the Bézier curves adjustment is carried out to approximate the contour of the generated curves to the real roads shape [5].

After the corridor is generated, a vision-based algorithm is used to adapt the corridor to the current road perceived by the frontal camera. Finally, a grid-based approach to consider the localization uncertainty is applied. Both processes are described in the following subsections. Fig. 2 shows the whole process of the road corridor generation, detailing inputs and outputs of the process.

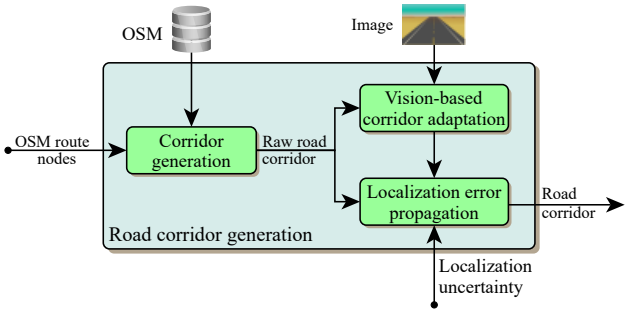



Fig. 2: Road corridor generation and adaptation process.

In Fig. 3 it can be seen how a road corridor is computed through the entrance and exit of a roundabout, where sections with different number of lanes are involved, obtaining a smooth shape.

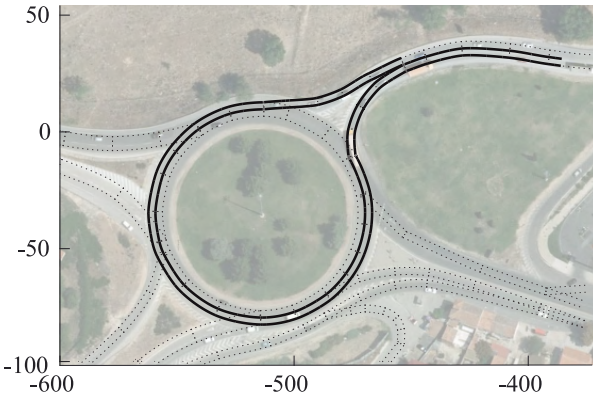



Fig. 3: Example of a generated road corridor

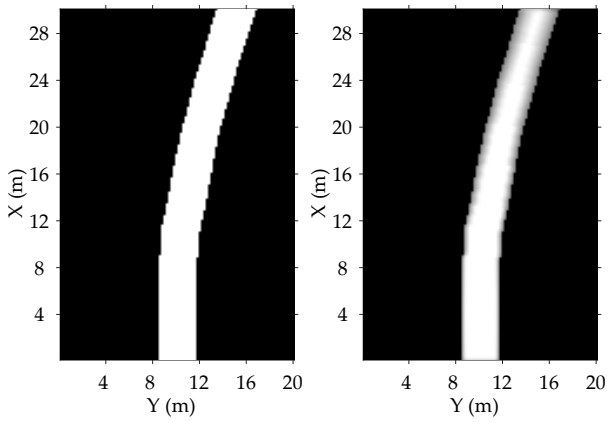
### A. UNCERTAINTY MANAGEMENT

In order to consider the localization uncertainty within the architecture, a similar approach to the one defined in [13] is used. This method requires the environment data to be represented in a probabilistic occupancy grid, which provides a way to represent information generated from different sensors measurements, taking into account their noise and uncertainty. Then, the vehicle pose uncertainty is propagated along the grid, obtaining the occupancy probability of each of the cells.

The first step to consider the localization uncertainty is to represent the map over the grid. In our case, the information of the map is composed of a set of Béziers curves that define the left and right boundaries of the navigable space. In order to set the occupancy of each Béziers segment over the grid, an extension of the Bresenham algorithm for cubic Béziers [14] is applied. After that, the free space existing inside the road corridor is filled with null occupancy probability while the rest of the grid is set as occupied (see Fig. 4a).

The result of uncertainty propagation over a 20×30 m grid with a grid resolution of 20 cm is shown in Fig. 4. Fig. 4a shows the initial road corridor rasterization on the grid, while in Fig. 4b it can be seen the computed occupancy probabilities due to localization uncertainties. Note that the values of uncertainty used in this example ($\sigma_x = \sigma_y = 0.25\ m$ and $\sigma_\theta = 0.07\ rad.$) were deliberately high to showcase their impact on the occupancy grid.

The uncertainty propagation over the grid results in the narrowing of the road corridor due mainly to the heading uncertainty. The occupancy probability of a priori free cells becomes higher when the $Y$ coordinate is increased as can be seen in the example shown in Fig. 4b.




(a) Road corridor rasterization over the grid

(b) Occupancy probability after uncertainty propagation

Fig. 4: Localization uncertainty propagation over a 20x30 occupancy grid with a resolution of 20cm

### B. VISION-BASED ROAD CORRIDOR ADAPTATION

An algorithm for road corridor adaptation based on image processing is also included in the architecture. This algorithm is designed to run every time the camera captures a new image (assuming a camera acquisition frequency of 20 Hz). Once the image is captured, the detection of road lines is carried out using a variation of [1] to define the available navigable space. Finally, the results obtained are compared with the road corridor generated as described in section III, so that the accuracy of the road can be increased by adapting the road corridor with the detected lane. The proposed approach comprises several steps that can be grouped in two main stages: (i) image processing and (ii) mapping & validity checking. The validity checking step allows the system to recognize fault lane detections so that if the lane is not correctly detected, the road corridor is not adapted and the localization uncertainty is propagated along the whole grid. Nevertheless, if the lane is correctly detected, the section that is in the field of view of the camera is adapted and the localization uncertainty propagation is computed in the rest of the grid. An example of the application of this algorithm is shown in Fig. 5. Moreover, Fig. 6 shows a top view of a raw road corridor generated and its vision-based adaptation.

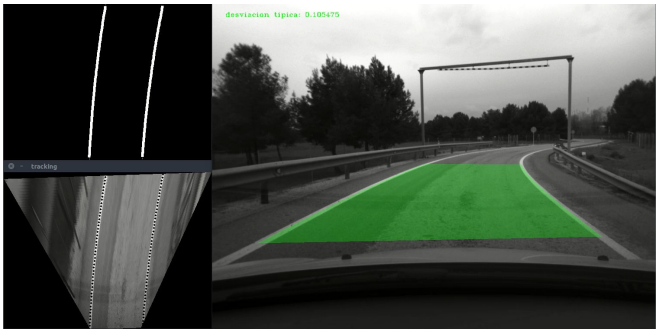


Fig. 5: Results of the vision-based adaptation

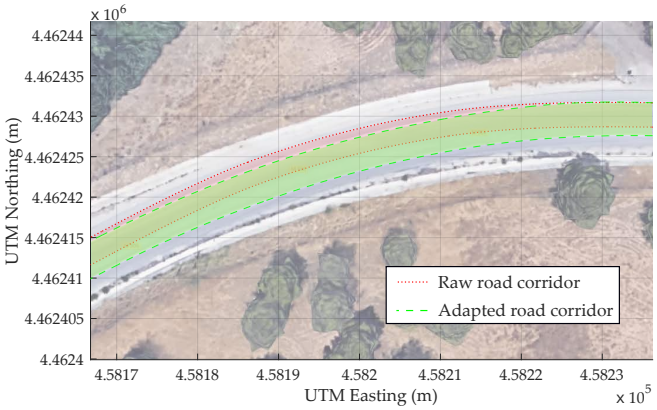



Fig. 6: Results of the vision-based adaptation

## IV. LOCAL PLANNER

The main goal of the local planner is to provide a feasible trajectory inside the previously generated road corridor. The local planner has been designed to generate smooth trajectories that guarantee a trade-off between comfort and safety inside the vehicle. To achieve that in an effective manner, the local planner involves three different modules that are described in the following subsections: manoeuvre planner, trajectory generation and motion prediction.

### A. MANOEUVRE PLANNER

The manoeuvre planner is in charge of analyzing the predicted behaviour of the perceived environment and deciding how the vehicle should react to the current situation. Four different planning modes have been stipulated to address all possible situations when analyzing the predicted motion of

nearby objects together with the current planned trajectory of the ego-vehicle: (i) re-plan from current pose, (ii) extend current trajectory, (iii) avoid static obstacle and (iv) avoid dynamic obstacle. Fig. 7 shows how these planning modes make easier the trigger of the final trajectory generation based on the current perceived situation. In this figure, $min_{pl}$ refers to the minimum threshold established for the remaining path length so that if the current path length is lower than $min_{pl}$, a trajectory extension is requested (planning mode 1).

In some cases, the current state of the road could impede to continue driving through the current road corridor due to road works, accidents, etc. To that end, the manoeuvre planner includes an interaction mechanism with the global planner to communicate the need of a new road corridor when needed (see Fig. 1 for more details).

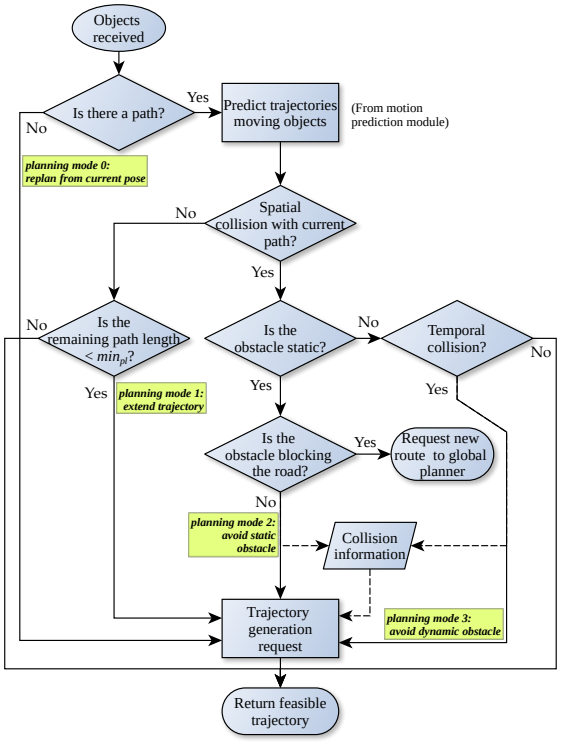



Fig. 7: Flow diagram of the manoeuvre planner module.

## B. TRAJECTORY GENERATION

The main goal of this module is to provide a new trajectory when requested by the manoeuvre planner. This trajectory must meet a set of requirements:

- To ensure comfort inside the vehicle, steering behaviour must be smooth and continuous. Moreover, lateral and longitudinal accelerations should not exceed specified maximum values along the trajectory.
- The trajectory generator must be able to provide feasible trajectories to avoid static or dynamic objects.
- The trajectory must be computed in a reasonable amount of time in order to be reactive enough to avoid collisions in dangerous situations.

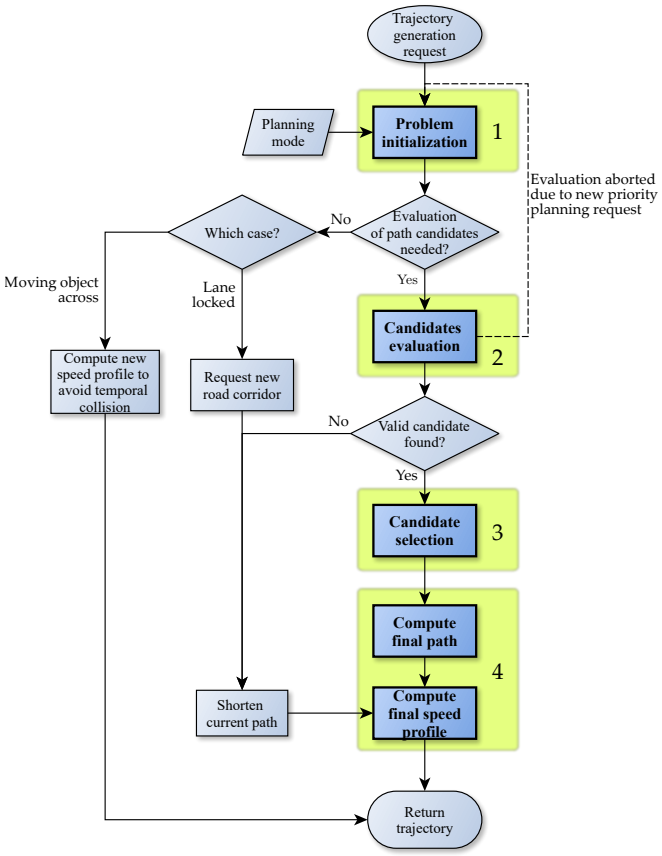


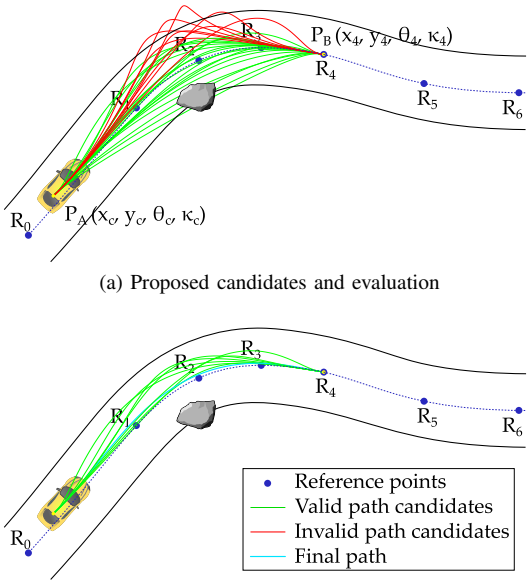

Fig. 8: General flow diagram of motion planning algorithm

The proposed strategy to solve the motion planning problem comprises several steps as depicted in the flow diagram of Fig. 8, where the four main stages are shaded in green and are described below:

1) **Motion planning problem initialization**: The initialization is the first task performed when a new trajectory is requested. Its main function is to set up the rest of the planning process based on the current vehicle state and the planning mode requested. At this first stage, the motion planning solver defines the search space to explore depending on the planning mode that has been set.
2) **Candidates evaluation**: At this stage all the path candidates that are stated in the problem initialization are computed and evaluated.
   Based on the above requirements, the primitive used for path planning must be able to generate a continuous curvature path. In geometric terms, that means that $G^2$ continuity must be guaranteed. Furthermore, the path must be computed as fast as possible to allow a large number of path candidates be evaluated. Taking into account the extensive comparison presented in [2] and the algorithm proposed in [3], a quintic Bézier curve is chosen to generate the final path sections.
   Some of the main advantages of fifth order Bézier curves over cubic ones are the higher control of the

curve shape and the possibility to impose curvature at both extremes of the curve. As a result of the above, it is possible to concatenate quintic Bézier sections to achieve curvature continuity along the path, thus complying with comfort requirements.
The first step of the evaluation is the validity checking. To consider as valid a path candidate, (i) its maximum curvature must be lower than the maximum curvature feasible by the vehicle, (ii) the space occupied by the vehicle while driving on the path candidate must be inside the road corridor and (iii) this space must not spatio-temporally collide with any static object or with the predicted trajectory of dynamic objects. Once the validity of candidates is checked, the associated costs are calculated for the valid ones based on a previously defined cost function related to the curve smoothness. Fig. 9a shows a simplified example of the candidates corresponding to one of the reference point ($R_4$). This step is the most computationally expensive of the motion planning algorithm.

3) **Candidate selection**: Among all valid candidates evaluated in the previous stage, the candidate with minimum cost is selected as the path for the final trajectory. Fig. 9b shows the valid candidates in green and the final selected path in cyan.
4) **Final trajectory calculation**: In order to obtain the final trajectory, firstly the Bézier curve of the selected path candidate is re-computed to obtain equidistant points with a small separation ($\approx$ 10cm). The final step is to compute its associated speed profile taking into account the maximum accelerations and speed to ensure comfort inside the vehicle. To compute the speed profile it is assumed motion with constant acceleration between two consecutive points of the path. First of all, a maximum speed is calculated by applying the centripetal force equation using the path curvature and the maximum desired lateral acceleration. Then, the speed profile is computed in a similar way to that proposed in [7]: the speed profile is traversed forwards and backwards to limit the longitudinal accelerations (positive and negative) and compute a new value for the speed if they are exceeded .




(a) Proposed candidates and evaluation

(b) Valid paths and final path selection

Fig. 9: Path planning process

## C. MOTION PREDICTION AND COLLISION AVOIDANCE

The motion prediction component of the architecture is in charge of determining the future motion of perceived obstacles. This component is the main input of the manoeuvre planner as the decisions it makes are closely dependent on the predicted behaviour of other nearby traffic participants or obstacles.

Firstly, the absolute speeds of the detected objects are computed to classify them as static or dynamic. In case an static object collides with the current trajectory of the ego-vehicle, a planning request that includes the relevant information about the detected collision (object ID, collision point, time to collision, etc.) is sent to the trajectory generator. Then, it is evaluated whether the object can be avoided or not by taking into account its position, dimensions and the road corridor width at its location. If the static obstacle can be avoided, the trajectory generator states a set of reference points within the free space in the normal direction of the road corridor at the location of the object, as depicted in Fig. 10. These reference points are used to generate the path candidates in this case.

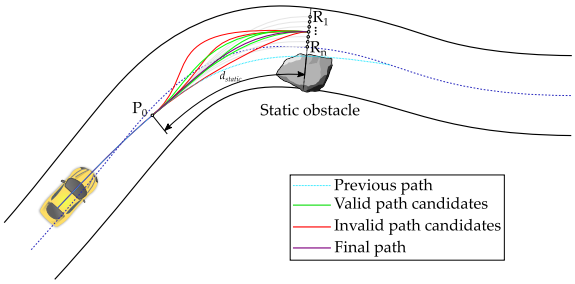



Fig. 10: Static obstacle avoiding

Regarding dynamic objects, their trajectories are computed assuming a constant velocity vector. Since this step is carried out when a new perception measurement is received (at a frequency of 12.5 Hz), this simplistic assumption allows to obtain a trajectory estimation that is reliable enough to safely check the collision with the ego-vehicle current trajectory. A schematic example is shown in Fig. 11, where a pedestrian is crossing the road corridor. Since there is spatial collision, the temporal collision must be checked using the inequality $\Delta T_{coll} > th_{coll}$, where $\Delta T_{coll}$ is the difference between the time at which the ego-vehicle ($T_v$) and the object ($T_p$) will pass through the collision point, and $th_{coll}$ is the safety temporal threshold (in seconds) to determine whether there is collision or not.

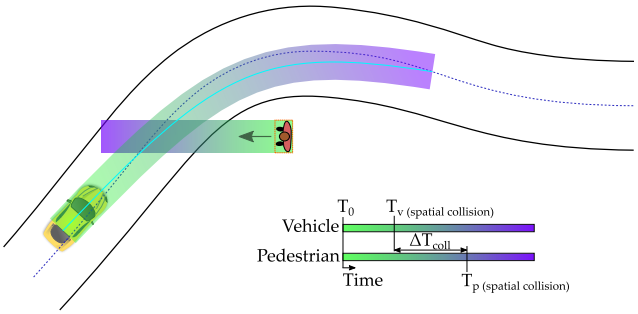



Fig. 11: Dynamic obstacle trajectory prediction and collision detection

## V. GLOBAL PLANNER

The automatic road corridor generation is integrated with a global planner which provides a global route to a selected destination.

The map representation employed by the global planner is extracted from OSM. Moreover, the data structure is extended to add additional information related to the nodes and ways allowing the modification of travelling costs of each way. This enables the global planner to take into account updated information about the state of the road that comes from other architecture components or even external information services using V2X communications.

The global planner is based on *Open Source Routing Machine* (OSRM) [8], which is used to obtain the route from the vehicle pose to the destination point by minimizing the travelled distance. In order to be able to update the map data structure with information coming from the local planner (blocked road, etc.), a modified instance of OSRM has been developed.

## VI. HUMAN MACHINE INTERFACE

The HMI component enables a bidirectional communication channel between the vehicle occupants and the autonomous driving decision architecture. On the one hand, the occupants can set the destination point from a map view represented in a friendly interface, and eventually select the possibility to take control. On the other hand, any relevant information about the decisions that the vehicle makes is shown (including its current state or warning information about perceived objects). This provides a practical way to keep aware the occupants of the current actions and future intentions of the ego-vehicle.

The implementation of the HMI is based on the open source project *OsmAnd* [10]. This interface has been modified to add the communication functionalities required for the interaction with the other architecture modules. Fig. 12 shows a snapshot of the HMI at the moment that a static obstacle has been detected.

## VII. EXPERIMENTAL RESULTS

The proposed decision-making architecture has been validated by performing different experiments using one of the instrumented vehicles of the AUTOPIA Program (see

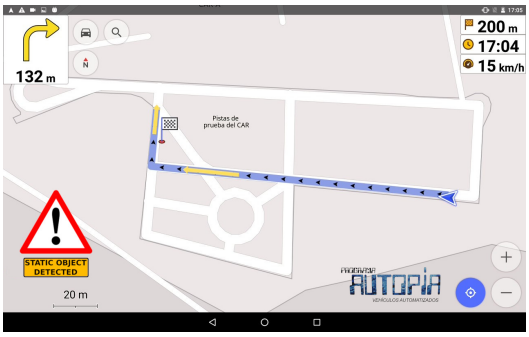



Fig. 12: HMI snapshot.


Fig. 13: Instrumented vehicle used in the experiments

Fig. 13) at the test track of the Centre for Automation and Robotics (CSIC-UPM). The localization of the vehicle used in these tests relies on a RTK DGPS receiver. To perceive the environment, the vehicle is equipped with a stereo camera and a four-layers LiDAR installed at the front of the vehicle. In these tests only the LiDAR was used to perceive the environment. The vehicle also includes an onboard computer with an Intel Core i7-3610QE and 8 Gb RAM. Furthermore, an Android-based tablet is also included as HMI for the vehicle occupants.

Two different trials were conducted in the real testing platform. In both experiments, the trajectory generator was set up to compute a total of 4500 paths in each planning request.

In the first experiment, static and dynamic obstacles are placed at different places of the route that the vehicle is following to reach the destination point, which was set a few meters behind the initial point as can be seen in Fig. 14. To improve the readability of this figure, only the valid path candidates of five representative planning requests (*1, 6, 16, 40, 42*) during the trial are plotted. To complement the understanding of this plot, table I shows relevant information about the representative planning requests of Fig. 14. In this table the planning modes correspondence is *0* - re-plan from current pose, *1* - extend current trajectory, *2* - avoid static obstacle, and *3* - avoid dynamic obstacle. Moreover, the percentage of valid candidates and the processing time in milliseconds at each request are also shown in this table.

The average planning time for the whole trial was 37 ms with a standard deviation of 25 ms.

Furthermore, Fig. 15 shows the speed profile and vehicle speed (left ordinate axis) together with the planning requests during the experiment (right ordinate axis). In this figure it can be seen how and when the planning requests are triggered with different planning modes depending on the situation: at the beginning, the first trajectory is planned from the current vehicle pose (mode *0*). At instant t=31.47 s, a dynamic obstacle is detected and consequently a new planning request is carried out (request ID 10). Later, at instant t=45.54 s, the first static obstacle is detected and a planning request of mode *2* is sent (request ID 16). Afterwards, successive planning requests are closely triggered (request IDs 17-42). This is caused by the small shape and localization changes of the perceived obstacles at different consecutive instants, what leads to launch the candidates evaluation procedure at a higher frequency. Note that the trajectory is quickly corrected to avoid the obstacles satisfactorily even with highly noisy perception information. It is also to be noticed that although the speed profile is smooth and continuous almost at every moment, there are two instants (planning requests 10 and 11), where the sudden incursion of the pedestrian in the road forces to reduce the target speed so that the vehicle keeps in the safe envelope.

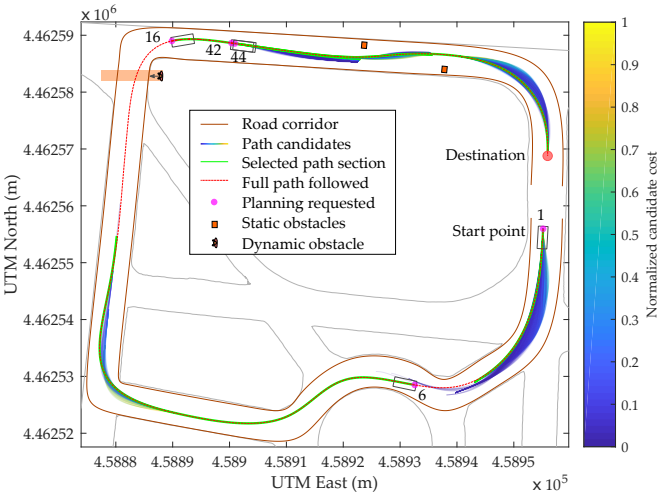



Fig. 14: Results of the first test

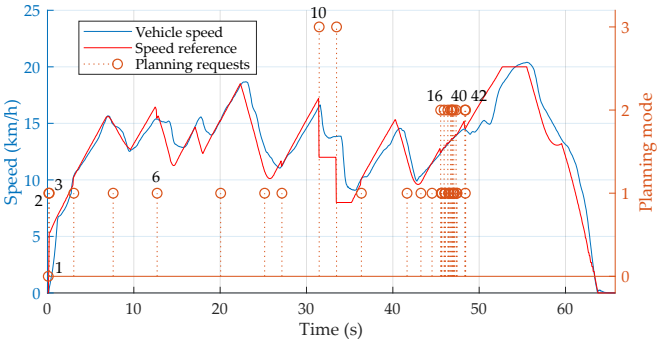



Fig. 15: Resulting speed profile and planning requests.

In order to analyze the comfort inside the vehicle during the test, Fig. 16 shows a density map to represent the measured longitudinal and lateral accelerations. In these tests, the maximum positive longitudinal, negative longitudinal and lateral accelerations were set to $0.4\ m/s^2$, $0.7\ m/s^2$ and $1.0\ m/s^2$, respectively.

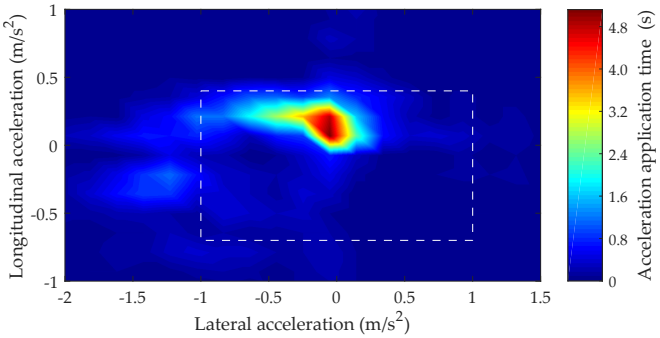



Fig. 16: Density graph of measured acceleration in the vehicle during the test

Fig. 16 shows how most of the measured acceleration values fall within the limits (marked with a dashed white rectangle in Fig. 16) established in the planning. However, some values are outside mainly due to the joint effect of vibrations induced by road imperfections and road and vehicle pitching and rolling. It can also be noticed that since the vehicle must go through more right turns than left ones to reach the destination, more negative (acceleration applied in left direction) than positive lateral acceleration values are measured.

| Planning request ID | Request time-stamp (s) | Planning mode | Planning time (ms) | Valid candidates (%) |
|---|---|---|---|---|
| 1 | 0.092 | 0 | 54.45 | 30.18 |
| 6 | 12.70 | 1 | 36.66 | 33.22 |
| 10 | 31.47 | 3 | 1.01 | - |
| 16 | 45.54 | 2 | 67.23 | 34.24 |
| 40 | 48.34 | 2 | 70.40 | 27.56 |
| 42 | 48.43 | 2 | 14.49 | 41.47 |

TABLE I: Additional information related to relevant planning requests.

In the second trial, in addition to avoidable objects, the vehicle finds static obstacles that are blocking the road corridor. In this case, a new route and consequently new road corridors are automatically requested to the global planner.

Figs. 17 and 18 show two snapshots of the front camera of the vehicle. At the bottom right corner of both figures, the HMI image is shown. In Fig. 17, it can be seen that the vehicle detects a big static obstacle that is blocking the lane. Immediately, the manoeuvre planner request a new route to the global planner and the trajectory generator shorten the current trajectory to avoid the collision with the blocking object while waiting for the new road corridor, as described in diagram of Fig. 8. Finally, the driving corridor is received by the local planner (see Fig. 18) and the vehicle continue the ride to the destination.

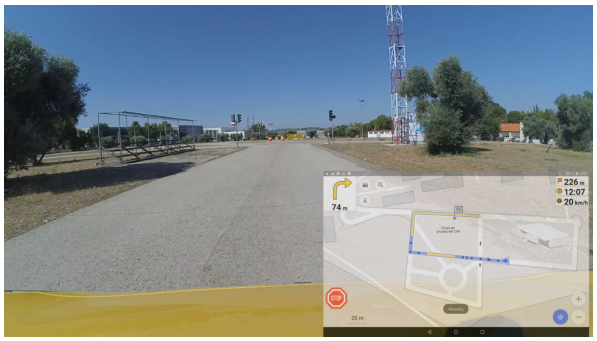


Fig. 17: Detection of lane block. A new route request to the global planner.

Fig. 18: The new road corridor is received.

Similar trials were carried out in the *Autonomous Driving Events* section of the *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS 2018) that was held at the Centre for Automation and Robotic (CSIC-UPM) facilities. A complete video of the whole demonstrator can be accessed in the following link: `https://youtu.be/SGVnmsUS06k`, where three different demonstrations that used the proposed decision-making architecture in this work were carried out: static obstacle avoidance, dynamic object avoidance and re-routing due to road blockage.

## VIII. CONCLUSIONS

In this work, a decision-making architecture to provide autonomous driving capabilities without detailed prior maps is proposed. The proposed architecture addresses both deliberative and reactive characteristics as well as the integration with an HMI. The road corridor generation module includes two different components to cover the lack of accuracy of such maps: a vision-based road corridor adaptation algorithm and a grid-based approach to propagate the localization uncertainty in the corridors. Finally, local planner components use the self-generated driving corridors to plan the trajectories that the vehicle should follow to reach its destination point safely while keeping comfort and predictability. Future work will be devoted to introduce interaction-aware motion prediction of the perceived dynamic objects, aiming at testing the system in crowded urban environments.

## REFERENCES


[1] M. Aly. "Real time detection of lane markers in urban streets". In: *Intelligent Vehicles Symposium, 2008 IEEE*. 2008, pp. 7–12.

[2] A. Artunedo, J. Godoy, and J. Villagra. "A primitive comparison for traffic-free path planning". In: *IEEE Access* (2018).

[3] A. Artuñedo, J. Godoy, and J. Villagra. "Smooth path planning for urban autonomous driving using OpenStreetMaps". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 837–842.

[4] S. Brechtel, T. Gindele, and R. Dillmann. "Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. 2014, pp. 392–399.

[5] J. Godoy, A. Artuñedo, and J. Villagrá. "Self-generated OSM-based driving corridors". 2019.

[6] C. Hubmann et al. "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 1671–1678.

[7] B. Lau, C. Sprunk, and W. Burgard. "Kinodynamic motion planning for mobile robots using splines". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 2427–2433. ISBN: 978-1-4244-3803-7. DOI: 10.1109/IROS.2009.5354805.

[8] D. Luxen and C. Vetter. "Real-time routing with OpenStreetMap data". In: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '11. Chicago, Illinois: ACM, 2011, pp. 513–516.

[9] T. Ort, L. Paull, and D. Rus. "Autonomous Vehicle Navigation in Rural Environments without Detailed Prior Maps". In: *International Conference of Robotics and Automation*. 2018.

[10] *OSMAnd. Openstreetmap automated navigation direction.* URL: `http://osmand.net/`.

[11] W. Schwarting, J. Alonso-Mora, and D. Rus. "Planning and Decision-Making for Autonomous Vehicles". In: *Annual Review of Control, Robotics, and Autonomous Systems* 1.1 (2018), pp. 187–210.

[12] D. Watzenig and M. Horn. *Automated driving: safer and more efficient future driving*. Springer, 2016.

[13] C. Yu, V. Cherfaoui, and P. Bonnifait. "Semantic evidential lane grids with prior maps for autonomous navigation". In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. 2016.

[14] A. Zingl. "A Rasterizing Algorithm for Drawing Curves". In: *Multimedia und Softwareentwicklung* (2012).