Motion planning approach considering localization uncertainty

Antonio Artuñedo, Jorge Villagra, Jorge Godoy and Maria Dolores del Castillo

***Abstract*—Localization plays an important role in autonomous driving since a high level of accuracy in vehicle localization is indispensable for a safe navigation. Most of the motion planning approaches in the literature assume negligible uncertainty in vehicle localization. However, the accuracy of localization systems can be low by design or even can drop depending on the environment in some cases. In these situations, the localization uncertainty can be taken into consideration in motion planning to increase the system reliability. Accordingly, this work presents two main contributions: (i) a probabilistic occupancy grid-based approach for localization uncertainty propagation, and (ii) a motion planning strategy that relies on such occupancy grid. Thus, the proposed motion planning solution for automated driving is able to generate safe human-like trajectories in real time while considering the localization uncertainty, ego-vehicle constrains and obstacles. In order to validate the proposed algorithms, several experiments have been conducted in a real environment.**

***Index Terms*—Automated driving, Motion planning, Localization uncertainty.**

I. Introduction

AUTOMATED driving requires methods to generalize unpredictable situations and reason in a timely manner in order to react safely even in complex urban situations. Generally, the knowledge about the environment is incomplete and the associated uncertainty is high, which affects motion planning. In view of this, two elements still need further substantial investigation: world modelling and motion planning from uncertain information.

Among the decision-making tasks that an automated vehicle must carry out, motion planning is particularly relevant as it plays a key role in ensuring driving safety and comfort [1], [2] while producing safe, human-like and human-aware trajectories in a wide range of driving scenarios. The robotics community has been intensively working over the last 30 years in motion planning problems. Although many of the proposed algorithms are able to cope with a wide range of situations and contexts, they often demand computation-intensive algorithms with unbounded processing cycles, only feasible for low speed motion patterns. However, for on-road autonomous driving, determinism is necessary at high sampling rates. In this context, optimality can be slightly sacrificed at the expense of safe human-adapted paths. Nevertheless, these approaches involve some drawbacks: (i) these techniques often provide scenario-dependant solutions, which may cause wrong behaviours in general real driving on urban roads [3]; (ii) to guarantee reactivity, the trajectories need to be exhaustively sampled and evaluated in a high-dimensional space, which is computationally expensive. To cope with these limitations, some works (e.g. [4], [5]) propose a higher-level decision maker able to select the right cost set and sampling scale for different situations. Recent approaches are able to compute analytically both path and speed profile in real-time, taking into account kinematic and dynamic constraints of the vehicle [6].

The problem of finding an optimal path subject to holonomic constraints avoiding obstacles is known to be PSPACE-hard [7]. Significant research attention has been directed towards studying approximate methods or particular solutions of the general motion planning problem.

Since for most problems of interest in autonomous driving exact algorithms with practical computational complexity are unavailable [8], numerical methods are often used. These techniques generally do not find an exact answer to the problem, but attempt to find a satisfactory solution or a sequence of feasible solutions that converge to the optimal solution [9]. The performance of these approaches is typically quantified by the class of problems for which they are applicable as well as their guarantees for converging to an optimal solution. These approximate methods for path and trajectory planning can be divided in three main families [1]: variational methods [10]–[12], graph-based search methods [13]–[17], and incremental search methods [18], [19].

As the applicability of variational methods is limited by their convergence to only local minima, graph-search methods try to overcome the problem by performing global search in a discretized version of the path space, generated by motion primitives. In some specific situations, this fixed graph discretization may lead to wrong or suboptimal solutions, in which case incremental search techniques may be useful, providing a feasible path to any motion planning problem instance, if one exists. In exchange, the required computation time to verify this completeness property may be unacceptable for a real-time system.

The motion planning approaches reviewed above assume an ideal vehicle localization. However, different existing approaches take into account the localization uncertainty in the planning strategy. The following subsection gives an insight of these existing techniques that are proposed in the literature.

Antonio Artuñedo, Jorge Villagra, Jorge Godoy and Maria Dolores del Castillo are with the Centre for Automation and Robotics, Arganda del Rey, Madrid. (e-mails: {antonio.artunedo, jorge.villagra, jorge.godoy, md.delcastillo}@csic.es)

A. Considering localization uncertainty in navigation

Recent navigation systems rely on high-definition maps [20]. Assuming a high accuracy of the maps information, the localization with respect to the map plays an important role.

Although many guidance systems for automated vehicles use probabilistic algorithms to solve state estimation problems, path planning is often conducted without explicitly considering the uncertainty of the vehicle position. Ignoring positioning uncertainty during planning may be acceptable if the vehicle is precisely localized, but it can lead to sub-optimal navigation decisions if the uncertainty is large. Indeed, in some cases, the accuracy of localization systems can be low by design or can drop depending on the driving situation environment -often in urban environments. Two examples where this may occur are (i) GPS-based localization systems in cases in which there are reflections or satellites occlusions, or (ii) challenging weather conditions that could affect the positioning accuracy such as cloudy scenarios. In these situations, the localization uncertainty should be taken into consideration to increase the system reliability.

There is a number of works explicitly considering perception/prediction uncertainties in the motion planning process (e.g. [21]), referred typically to as planning in belief space. A general formalization for this type of problem is the Partially Observable Markov Decision Process. There are several works proposing solutions within this framework (e.g. [22], [23]), but most of them become quickly intractable for real-time automotive applications.

To reduce this complexity different strategies have been adopted: (i) to bound the uncertainties and base the decision making on the worst-case scenarios [24], (ii) to consider uncertainties with a moderated influence, but using an omnipresent residual threat level [25], (iii) to take into account multidimensional pose uncertainties due to erroneous measurement and object prediction through collision probabilities [26], [27].

Alternatively, [28] uses a Linear-Quadratic Gaussian framework to estimate the uncertainty when following a given candidate trajectory, given the noise characteristics of the localization and control. The authors incorporate the state estimation in the future instants as stochastic variables and the result is a distribution for the state of each candidate trajectory.

Another approach to deal with the localization uncertainty when using maps is the one proposed in [29]. Although this paper does not consider motion planning, it presents an interesting technique to encode lane and traffic information in grids. It represents environment data in a probabilistic occupancy grid, where the vehicle pose uncertainty is propagated in every grid cell.

Given the different nature of the approaches found in the literature to consider the localization uncertainty in the motion planning strategy, it is very difficult to compare in a quantitative manner the method proposed in this paper with the related work. Nevertheless, a qualitative analysis of the key features of the main approaches proposed in the literature is summarized in Table I.

These approaches are compared in terms of computational cost, their dependency on the scenario and whether they have been tested only in simulation or not. As can be seen, our approach exhibits lower computational cost as it does not require to solve dynamic programming problems (as in [28]) or solve POMDPs (as needed in [30]). In the case of [24], authors do not provide computation times. In contrast to other approaches, which have only been tested in simulation, our method has been validated in an experimental platform in real environments. In addition, the proposed method is not dependent on the applied scenario, while the other analysed strategies have only been tested in specific traffic situations.

Approach \ Feature	Computational cost	Scenario dependent	Only tested in simulation
POMDP-based [30]	High	Yes	Yes
LQG-based [28]	Medium	Yes	Yes
IGHLC [24]	—	Yes	Yes
Our approach	Low	No	No

TABLE I
QUALITATIVE COMPARISON OF RELATED WORK.

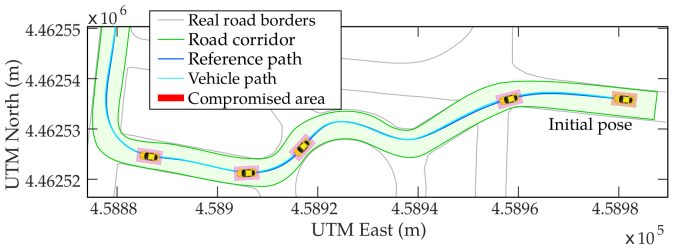
Contrarily to some previous works on probabilistic motion planning [30], [31], often intractable for real-time applications, the setting presented in this work produces feasible and comfortable trajectories that introduces an explicit dependence of localization uncertainty in a grid-based world model. To deal with positioning inaccuracies in motion planning when map data is used, a localization uncertainty propagation algorithm is proposed. By means of this approach the occupancy probability of each cell in a local grid map is obtained, thus providing a more adapted information about the navigability of the nearby environment than merely a road corridor, as considered in the majority of related works.

This article presents two main contributions: (i) the grid-based approach for localization uncertainty propagation, and (ii) a motion planning approach that relies on a probabilistic occupancy grid. Thus, the proposed motion planning solution for automated driving is able to generate safe human-like trajectories in a real time setting, while considering the localization uncertainty, ego-vehicle constraints and obstacles. The promising results are validated on a variety of driving situations using an automated vehicle.

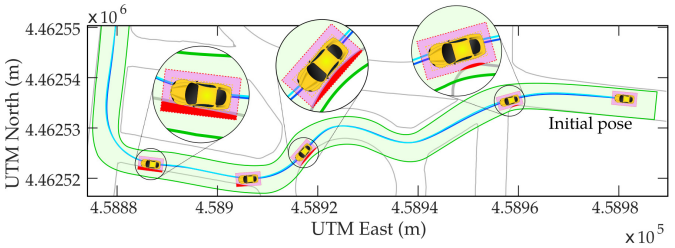
The remainder of this paper is structured as follows: In section II, the proposed localization uncertainty propagation algorithm is described. Section III introduces the motion planning approach based on an occupancy grid. The results on a real vehicle are presented and discussed in section IV. Finally, section V summarises the conclusions and future work.

II. LOCALIZATION UNCERTAINTY INFLUENCE

Localization plays an important role in autonomous driving. In particular, when maps are used as a part of the environment understanding, a good localization with respect to the map becomes critical. Note that the localization uncertainty propagation essentially affects the map and not the perceived objects. In the ideal case where the localization of the vehicle is perfectly known, i.e. there are no localization uncertainties ($\sigma_x = \sigma_y = \sigma_\theta = 0$), a planned trajectory taking into account information from maps (road corridors) will not compromise



(a) Ideal case: $\sigma_x = \sigma_y = \sigma_\theta = 0$



(b) Realistic case with small localization errors: $x_e = y_e = 0.05\ m$, $\theta_e = 1°$

Fig. 1. Problem introduction

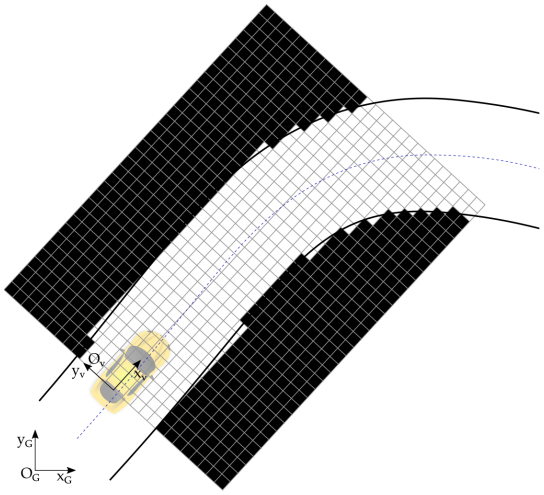


Fig. 2. Road corridor rasterization over the grid

safety, assuming the map is accurate enough. Nonetheless, in realistic scenarios, errors in both position and orientation can cause the motion planner to compute trajectories that take the vehicle off the real edges of the road. To illustrate this, Fig. 1 shows a comparison between both cases in the same scenario. In the ideal case, shown in Fig. 1a, the road corridor fits to the real borders of the road. However, in a more realistic case (Fig. 1b), the road corridor does not fit the real edges of the road due to the ego-vehicle localization errors ($x_e = y_e = 0.05\ m$, $\theta_e = 1°$) along the route. As can be seen in Fig. 1b, the safety margin of the vehicle (depicted with a light red rectangle) is firstly compromised 20 m ahead the initial position. Moreover, the drift of the road corridor increases the greater the distance from the initial position. While the localization error represents the difference between a localization measurement and the true localization, the localization uncertainty is the range of values within which the true localization is guaranteed to be found. Accordingly, the uncertainty is related to error to the extent that it represents an estimation about the error in the measured quantity. Henceforth, location uncertainty is used to state the proposed propagation method and thus take it into account in the motion planning strategy.

To mitigate the effect of ego-localization uncertainty, the method proposed in this work, which is inspired by the aforementioned work [29], takes into consideration the localization uncertainty estimation by propagating it along an occupancy grid. Thus, in situations where the localization accuracy is low, the localization uncertainty can be taken into consideration in the motion planning in order to increase the reliability of the navigation system.

The first step is to represent the available map over the grid. In our case, the information of the map is self-generated from OpenStreetMap by applying the algorithm presented in [32]. It is composed of a set of Bézier curves that define the left and right boundaries of the navigable space. In order to set the occupancy of each Bézier segment over the grid, an extension of the Bresenham algorithm for cubic Béziers [33] 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. 2). In this work, the map is assumed to be highly reliable. Nevertheless, the unreliability of the maps can be also considered within the proposed framework.

Finally, the uncertainty of the vehicle pose is propagated at all the initially free cells using the general approach defined in [29]. Let Cx_i and Cy_i denote the coordinates of the cell i of the grid in the frame O_v, and Vx_{O_G}, Vy_{O_G} and $V\theta_{O_G}$ denote the vehicle position and heading in the frame O_G. Then, it is transformed into the frame O_G as follows:

$$\begin{pmatrix} x_{O_G} \\ y_{O_G} \end{pmatrix} = T_f = R_{vG} \begin{pmatrix} Cx_i \\ Cy_i \end{pmatrix} + \begin{pmatrix} Vx_{O_G} \\ Vy_{O_G} \end{pmatrix} \tag{1}$$

$$R_{vG} = \begin{pmatrix} \cos(V\theta_{O_G}) & -\sin(V\theta_{O_G}) \\ \sin(V\theta_{O_G}) & \cos(V\theta_{O_G}) \end{pmatrix} \tag{2}$$

where R_{vG}, in equation (1), is the rotation matrix from the vehicle frame O_v to the global frame O_G. Note that the uncertainty of the position of each cell in the global frame $(x_{O_G},\ y_{O_G})$ comes from the uncertainty in the vehicle pose in the global frame $(Vx_{O_G},\ Vy_{O_G})$ as the position of the cells in the vehicle frame (Cx_i, Cy_i) is known.

The covariance matrix g_i in each cell i can be calculated from the estimated pose uncertainty as follows:

$$g_i(x_i, y_i) = \left(\frac{\delta T_f}{\delta V_{O_G}}\right) Q_V \left(\frac{\delta T_f}{\delta V_{O_G}}\right)^T \tag{3}$$

where $T_f(Vx_{O_G}, Vy_{O_G}, V\theta_{O_G})$ denotes the transformation in equation (1), Q_V represents the covariance matrix of the current pose $V_{O_G}(Vx_{O_G}, Vy_{O_G}, V\theta_{O_G})$ in the global frame and $\frac{\delta T_f}{\delta V_{O_G}}$ is the Jacobian, expressed as follows:

$$\left(\frac{\delta T_f}{\delta V_{O_G}}\right) = \begin{pmatrix} \frac{\delta T_f}{\delta V x_{O_G}} \\ \frac{\delta T_f}{\delta V y_{O_G}} \\ \frac{\delta T_f}{\delta V \theta_{O_G}} \end{pmatrix}^T \tag{4}$$

Thus, the resulting covariance matrix can be expressed as:

$$g_i(x_i, y_i) = \begin{pmatrix} \sigma_{xi}^2 & \rho_i \sigma_{xi} \sigma_{yi} \\ \rho_i \sigma_{xi} \sigma_{yi} & \sigma_{yi}^2 \end{pmatrix} = \begin{pmatrix} \sigma_x^2 + \sigma_\theta^2 \; u(V\theta_{O_G}) & \sigma_\theta^2 \; t(V\theta_{O_G}) \\ \sigma_\theta^2 \; t(V\theta_{O_G}) & \sigma_y^2 + \sigma_\theta^2 \; v(V\theta_{O_G}) \end{pmatrix} \tag{5}$$

where:

$$\begin{aligned} u(V\theta_{O_G}) &= (-\sin(V\theta_{O_G}) \; Cx_i - \cos(V\theta_{O_G}) \; Cy_i)^2 \\ v(V\theta_{O_G}) &= (\cos(V\theta_{O_G}) \; Cx_i - \sin(V\theta_{O_G}) \; Cy_i)^2 \\ t(V\theta_{O_G}) &= \sin(V\theta_{O_G}) \cos(V\theta_{O_G})(Cx_i^2 - Cy_i^2) + \\ & \quad Cx_i Cy_i (\sin(V\theta_{O_G})^2 - \cos(V\theta_{O_G})^2) \\ \rho_i &= \frac{\sigma_\theta^2 \; t(V\theta_{O_G})}{\sigma_{xi} \; \sigma_{yi}} \end{aligned}$$

being σ_x, σ_y and σ_θ the given pose uncertainties of the ego-vehicle localization.

Once the covariance matrix g_i is known, a bivariate Gaussian distribution f_i is applied to compute the probability distribution for each cell:

$$f_i(x_j, y_j) = \frac{1}{2\pi \sigma_{xi} \sigma_{yi} \sqrt{1-\rho_i^2}} \cdot e^{-\frac{1}{2(1-\rho_i^2)} \left[\left(\frac{Cx_j - Cx_i}{\sigma_{xi}} \right)^2 - 2\rho_i \left(\frac{Cx_j - Cx_i}{\sigma_{xi}} \right) \left(\frac{Cy_j - Cy_i}{\sigma_{yi}} \right) + \left(\frac{Cy_j - Cy_i}{\sigma_{yi}} \right)^2 \right]} \tag{6}$$

To compute final occupancy probability of each cell, a 95% confidence ellipse (χ^2 = 5.991) is defined from the computed covariances. Then, the final occupancy probability of each cell is obtained by dividing the sum of the expected values of the cells that fall within the confidence ellipse ($\sum_{j \in I_i} f_i(x_j, y_j) \cdot F_j$) by the sum of their probabilities ($\sum_{j \in I_i} f_i(x_j, y_j)$).

$$P(x_i, y_i) = \frac{\sum_{j \in I_i} f_i(x_j, y_j) \cdot F_j}{\sum_{j \in I_i} f_i(x_j, y_j)} \tag{7}$$

where I_i is the set of cells that falls within the ellipse generated for cell i, j is the index of the cells inside I_i, S is the size of set I_i, $f_i(x_j, y_j)$ is the probability in cell j obtained by the Gaussian distribution in equation (6), generated for the cell i, and $F_j \in \{0, 1\}$ is the initial occupancy value of the cell j.

A schematic example of the result is shown in Fig. 3, where cell colours vary from white (free cells) to black (occupied cells) passing through grey shades representing different occupancy probability values.

A summary of the localization propagation algorithm over the road corridor is shown in algorithm 1.

The uncertainty propagation over the grid results in the narrowing of the road corridor, being this effect particularly pronounced when the heading uncertainty is high. The occupancy probability of a priori free cells becomes higher when the x_v coordinate is slightly increased as can be seen in 3.

Fig. 4 shows the results of uncertainty propagation using different values of σ_x, σ_y and σ_θ. In all cases the grid size is 20×30 meters with a grid resolution of 20 cm.

Fig. 4a shows the initial road corridor rasterization over the grid. In this first case, only fully occupied (black) and free (white) cells are obtained. The other three figures (4b, 4c and

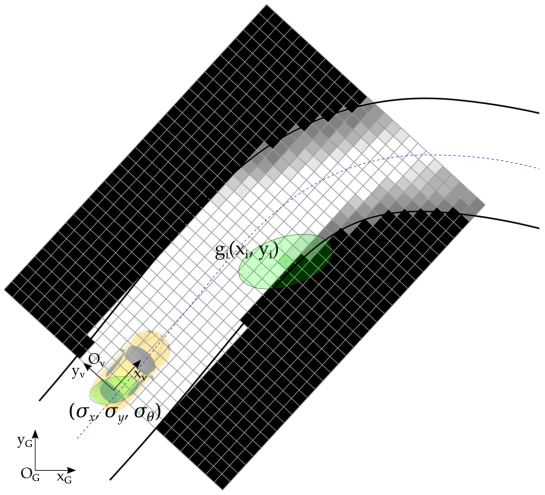
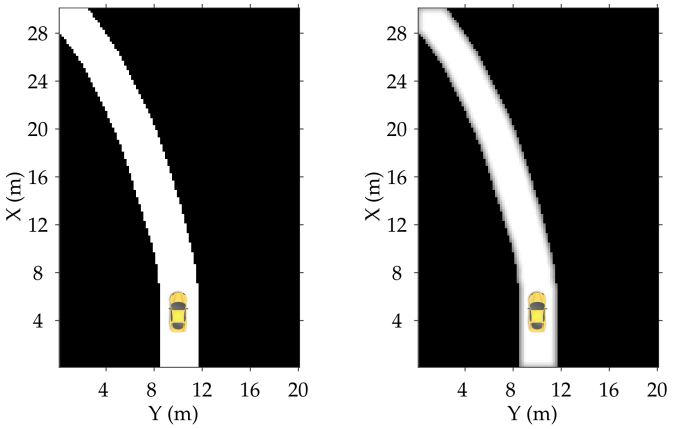


Fig. 3. Localization uncertainty propagation over the grid



(a) Road corridor rasterization over the grid

(b) $\sigma_x = 0.5\ m$, $\sigma_y = 0.5\ m$ and $\sigma_\theta = 0\ rad$

(c) $\sigma_x = 0.5\ m$, $\sigma_y = 0.5\ m$ and $\sigma_\theta = 0.07\ rad$

(d) $\sigma_x = 0.5\ m$, $\sigma_y = 0.5\ m$ and $\sigma_\theta = 0.3\ rad$

Fig. 4. Examples of localization uncertainty propagation over a $20m \times 30m$ occupancy grid

Input: Pose uncertainty (σ_x, σ_y, σ_θ), Road corridor
Output: Occupancy probability in each cell
$grid \leftarrow$ Left and right boundaries of the road corridor ;
$grid$(cells inside road corridor) $\leftarrow 0$ (empty) ;
$grid$(cells outside road corridor) $\leftarrow 1$ (occupied) ;
foreach $i \leftarrow 1$ ***to*** n **do**
  compute covariance matrix [equation (3)]
  compute bivariate Gaussian distribution [equation (6)]
  $grid(i) \leftarrow \frac{\sum_{j \in I_i} f_i(x_j,y_j) \cdot F_j}{\sum_{j \in I_i} f_i(x_j,y_j)} [equation\ (7)]$
end
return $grid$;

Algorithm 1: Localization uncertainty propagation over an occupancy grid

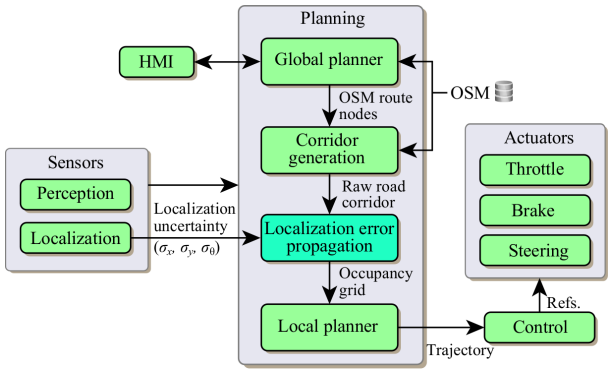


Fig. 5. Functional diagram of the architecture

4d) show the occupancy probabilities resulting from different values of σ_x, σ_y and σ_θ. As can be observed in Fig. 4b, where $\sigma_x = \sigma_y = 0.5\ m$ and $\sigma_\theta = 0\ rad$, most of the cells within the road corridor are completely free even with high uncertainties in X and Y axis. In this case, it can be noticed that the occupancy probability is increased in the proximity to the corridor edges. Furthermore, the case represented in Fig. 4c adds a small orientation uncertainty. This leads to the narrowing of the road corridor in the farthest cells from the vehicle. These examples depict the high influence of the orientation uncertainty in comparison with X and Y ones. Finally, a case with high longitudinal, lateral and angular uncertainties is shown in Fig. 4d, where only the cells closest to the vehicle are guaranteed to be unoccupied.

The proposed strategy is generic so there is no specific working range for the localization uncertainty values. As a result, in extreme cases where the localization information is not available, the localization uncertainty would increase to very high values. In these situations, the proposed method would provide an occupancy grid with high occupancy probability values in the initially free cells. Consequently, the motion planning algorithm would not provide trajectories that pass through that probably occupied space.

III. Occupancy grid-based motion planning

The proposed strategy for motion planning runs within the functional architecture shown in Fig. 5. As can be observed, the *Planning* block includes both global and local planning functionalities.

On the one hand, based on a destination coming from the human-machine interface (HMI) and OpenStreepMap data (OSM), the *global planner* is able to obtain a global route represented by a list of nodes that are later used to compute a road corridor [32]. Then, this corridor is used by the algorithm proposed in this paper (blue block) to propagate the localization uncertainties (σ_x, σ_y and σ_θ) over the occupancy grid.

On the other hand, the *local planner* block is in charge of computing the final trajectories that the control module will use to generate the final control actions. The proposed local motion planning algorithm uses the path candidates generation method presented in [34], [35]. This trajectory generation relies on safe, human-like and comfortable trajectories computed evaluating cost-effective primitives, based on quintic Bézier curves. Using this type of primitive, it is possible to generate paths with continuous and smooth curvatures. Furthermore, the applied method for speed planning imposes maximum speed and maximum lateral and longitudinal accelerations to provide speed profiles similar to what a human driver would do. In this regard, by human-like safe trajectories we refer to those that, on the one hand, are safe, i.e. remain within the navigation corridor and do not collide with any obstacle, and on the other hand, are similar to those that a human driver would perform in the same driving situation, both in geometric terms of the path and in the computed speed profile. Moreover, the method is able to consider the kinodynamic constrains of the vehicle while reactively handling dynamic environments in presence of both static and moving obstacles. Nevertheless, note that the strategy proposed in this work for considering the localization uncertainty is not dependent on the path primitive used. This motion planning approach comprises the following stages:

1) **Motion planning problem initialization**: At this first stage, the motion planning solver defines the search space to explore depending on the current driving situation (i.e. whether static/dynamic obstacles must be avoided or not). From the stated search space, a set of continuous-curvature path candidates based on quintic Bézier curves is generated.
2) **Candidates evaluation**: At this stage all the path candidates are evaluated by checking their validity and calculating their costs based on previously defined cost functions.
3) **Candidate selection**: Among the valid evaluated candidates, one is selected based on its cost value.
4) **Final trajectory calculation**: Once the best candidate is chosen, the speed profile is calculated taking into account the maximum speed and accelerations to ensure comfort inside the vehicle.

Each of these stages are detailed in the following subsections.

A. Motion planning problem initialization

In order to use the occupancy probability information in the local planner, the stages **problem initialization** and **candidates evaluation** have been extended with respect to the

approach in [35]. The first stage of the trajectory generator is in charge of defining the search space to create the path candidates. Taking into account the size of the occupancy grid, the **problem initialization** states the path candidates from the reference points that falls within the grid.

The size of the occupancy grid is updated every time it is computed. This is carried out to cover a specified amount of meters ahead the current vehicle position measured over the centreline of the road corridor. In this context, the road corridor ahead the vehicle is used to calculate the width and height of the occupancy grid ensuring that both left and right boundaries from the vehicle position until the last considered point centreline point fall within the grid. Moreover, the occupancy grid is always aligned with the vehicle orientation.

1) Path candidates generation: In this first stage (motion planning problem initialization), the set of path candidates to be evaluated are stated. The generation of path candidates relies on the algorithm proposed in [34], [35], which focuses on ensuring comfort inside the vehicle while providing safe trajectories to avoid collisions with obstacles. For further details about the collision avoidance strategy for both static and moving obstacles, please refer to [35]. This approach is based on self-generated road corridors as described in [32], whose centrelines are used to generated a set of reference points to create the path candidates from the vehicle pose. To choose the path primitive, the extensive comparison presented in [36] has been taken into account. A quintic Bézier curve is used to generate the path candidates due to its controllability and the possibility to impose the curvature at its extremes. The analytic expression is obtained as follows:

$$C_{d_b}(t) = \sum_{i=0}^{d_b} P_i^j B_{i,d_b}(t),\ t \in [0,1] \tag{8}$$

being $B_{i,d_b}(t) = \binom{d_b}{i} t^i (1-t)^{d_b - 1}$ the Bernstein polynomials, P_i the control points of Bézier curve, and d_b the degree of the Bézier curve. In the case of a quintic curve ($d_b = 5$), it can be expressed explicitly as:

$$\begin{aligned} C_5(t) = {} & (1-t)^5 \mathbf{P}_0 + 5t(1-t)^4 \mathbf{P}_1 \\ & + 10t^2(1-t)^3 \mathbf{P}_2 + 10t^3(1-t)^2 \mathbf{P}_3 \\ & + 5t^4(1-t)\mathbf{P}_4 + t^5 \mathbf{P}_5,\ t \in [0,1] \end{aligned} \tag{9}$$

By using this primitive, the motion planning approach is able to generate continuous curvature paths, i.e. maintaining G^2 continuity. Moreover, this approach aims to achieve the best trade-off between optimality and computing time.

B. Candidates evaluation

In this stage, all the path candidates that have been selected in the problem initialization are evaluated.

The main difference when using the occupancy grid for motion planning with respect to the approach presented in [34], [35] is found in this stage. The drivable space boundaries will be replaced by the occupancy grid to evaluate the validity of each path candidate and the cost related to its smoothness.

The validity is determined by checking three conditions are fulfilled: firstly (i) the maximum curvature of the path candidate must be lower than the maximum curvature feasible by the vehicle ($\kappa_{max}^{p_c} < \kappa_{max}^{v}$); (ii) the path must be inside the road corridor and (iii) there must not be a collision with any obstacle.

One of the main advantages of the occupancy grid approach is that it allows to fuse environmental information coming from different sources. In this case, both the road corridor and the obstacles perceived by exteroceptive sensors are rasterized into the grid. This method allows to abstract the motion planning algorithm to the different information sources, focusing only in one occupancy grid. That means that points (ii) and (iii) mentioned in the paragraph above (i.e. collision checking) can be verified at once. Therefore, instead of using collision checking algorithms to sequentially verify that the path is inside the road corridor and there are no collisions with obstacles, the occupancy values of the grid cell crossed by the path candidate are used to accomplish this task.

The approach for collision checking uses the bounding rectangle of the vehicle. The method is based on the generation of a polygon that represents the space that the vehicle would take while driving along the computed path. These occupancy polygons are referred as path-polygons in the remaining of the paper and are used to check the validity of each path candidate.

1) Occupancy polygon computation: Once it has been verified that the maximum curvature constraint is fulfilled, the path-polygons of the candidate is computed. In order to obtain the occupancy polygon, the dimensions of the vehicle and the path generated as explained in section III-A1 are needed. Taking advantage of the fact that the path is a Bézier curve or a concatenation of them, the tangent vector and the curvature can be obtained analytically at each point of the path.

Based on the path, the right and left bounds of the area occupied by the vehicle can be calculated as follows: the right bound will be composed of the points of right extreme of the front of the vehicle when the vehicle is turning left and of the points of right extreme of the rear axle when turning right. The left bound is calculated analogously: it is composed of the points of left extreme of the front of the vehicle when the vehicle is turning right and of the points of left extreme of the rear axle when turning left.

It is worth to mention that a safety margin is added around the vehicle (d_{sm}). To determine if vehicle is turning right or left, the sign of the curvature is used. Finally, the polygon is conformed by joining the points of right and left sides to obtain a closed shape. Fig. 6 shows an example of the path-polygon for a given path, where l_{tw} is the vehicle axle track, l_{la} is the distance from the rear axle to the front bumper and l_{lb} is the distance from rear bumper to the rear axle.

In point B of Fig. 6 it can be seen how the vehicle is turning right and the extreme left point of the front belongs to the left bound of the polygon, while the right extreme point of the rear axle y used for the right bound.

Once the path path-polygon has been obtained, the grid cells that the candidate occupies are identified. To do that, firstly the candidate polygon is rasterized on an empty grid with the

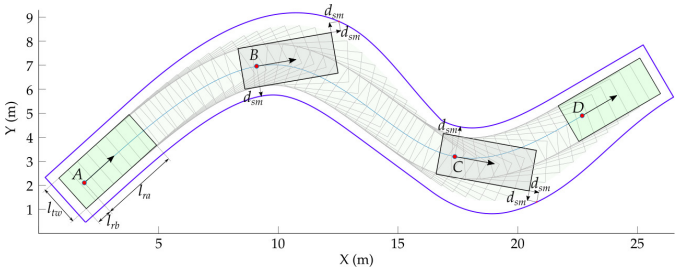
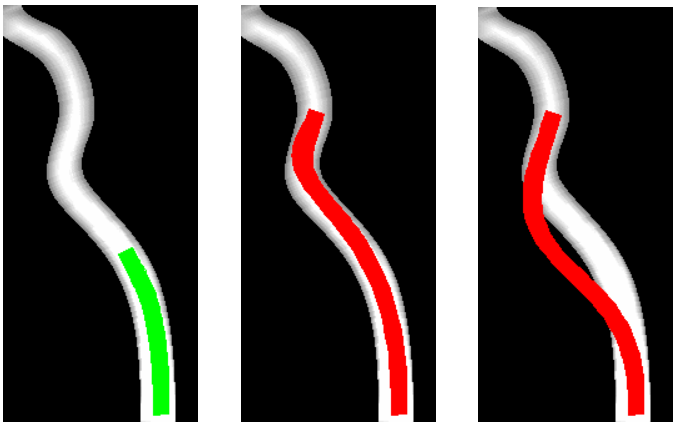


Fig. 6. Example of the path-polygon calculation for collision checking



(a) Valid candidate. (b) Invalid candidate. (c) Invalid candidate.

Fig. 7. Path-polygon rasterization and validation on an occupancy grid.

same size that the occupancy grid and the scanline flood fill algorithm [37] is used to find the cells that fall within the polygon. Thus, a list of all cells occupied by the candidate is obtained and can be used to extract occupancy data from the occupancy grid to verify its validity and compute its cost.

To determine the validity of the candidate, the occupancy probability of each cell traversed by the candidate must be below a given occupancy probability threshold. Thus, if the path candidate falls on cells where the occupancy probability is greater than Pv_{th}, the candidate is discarded. In this regard, to consider as invalid the path candidates that pass through a probably occupied area, the threshold Pv_{th} has been set to 50%. The candidate is valid if the following equation is verified:

$$P^c_{max} < Pv_{th} \tag{10}$$

where P^c_{max} is the maximum probability occupancy of all cells crossed by the candidate.

Some examples of the rasterization of path-polygons are shown in Fig. 7. Notice that in the case depicted in Fig. 7b, the path is inside the road corridor, unlike the case in Fig. 7c. Nevertheless, it is also considered as invalid as the maximum occupancy probability (P^c_{max}) exceeds the established threshold (Pv_{th}).

2) Candidate cost computation: Once the validity of the candidates is checked, the cost of the valid ones is computed. This cost function comprises different components: On the one hand, first and second derivatives of the curvature reflect the smoothness of the path along the curve. A weight ($w_{\ddot{\kappa}}$) has been added to balance the value of both derivatives. Moreover, the length of the path (L_p) is used to normalize its result. The motivation for adding a weight to L_p (w_{L_p}) is that in case that $w_{L_p} = 1$ (i.e. this weight is not considered), the path that minimizes the cost function tends to be straight and short in spite of being in curved road sections, thus obtaining almost straight paths in cases in which it should not. In order to avoid that, w_{L_p} is used with values greater than 1. In addition to that, another component has been added to the function in order to represent the mean occupancy probability over the path candidate ($\overline{P}_{pc}$), that is weighted by using w_{og}. Note that this last term of the cost function ($\overline{P}_{pc}$) allows the candidate's mean occupancy probability to be weighted along with the other terms of such function in order to avoid driving through areas with high occupancy probabilities.

$$J_p = w_{og}\overline{P}_{pc} + \frac{1}{w_{L_p}L_p}\int_{s_0}^{s_f} \dot{\kappa}(s)^2 + w_{\ddot{\kappa}}\ddot{\kappa}(s)^2\, ds \tag{11}$$

C. Best candidate selection and final trajectory calculation

Among all valid candidates evaluated in the previous stage, the candidate with minimum cost is selected as the path for the final trajectory. Then, the Bézier curve of the selected path candidate is evaluated to obtain equidistant points.

Since equidistant points can not be directly computed in a Bézier curve, the curve polynomials are evaluated using a fine discretization of the parameter t to approximate the relationship between t and the distance over the curve (s). Finally, the values of t to obtain equidistant points are calculated by interpolation.

Once a path with equidistant points is obtained, the speed profile is computed using the method introduced in [35]. This speed profile calculation method allows to impose maximum lateral ($a_{max,lat}$) and longitudinal accelerations ($a_{max,acc}, a_{max,dec}$) as well as maximum speed (v_{max}) for an arbitrary path whose curvature can be accurately obtained.

IV. Results and Discussion

This section presents the experiments carried out in an experimental vehicle at the Centre for Automation and Robotics (CAR). A set of different trials have been conducted for the proposed motion planning approach.

A. Experimental platform

The proposed motion planning algorithm has been validated by performing different experiments using one of the instrumented vehicles of the AUTOPIA Program (see Fig. 8) at the test track of the Centre for Automation and Robotics (CSIC-UPM) in Arganda del Rey, Spain.

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 on-board computer with an Intel Core i7-3610QE and 8Gb RAM.

Fig. 8. Experimental platform

Parameter	Description	Value
d_{sm}	Safety distance around vehicle (m)	0.4
n_{rp}	Reference points used to create candidates	15
v_{max}	Max. allowed speed (km/h)	20
$a_{max,lat}$	Max. lateral acceleration (m/s^2)	1.0
$a_{max,acc}$	Max. positive longitudinal acceleration (m/s^2)	0.4
$a_{max,dec}$	Max. negative longitudinal acceleration (m/s^2)	0.7

TABLE II
PATH AND SPEED PLANNING CONFIGURATION

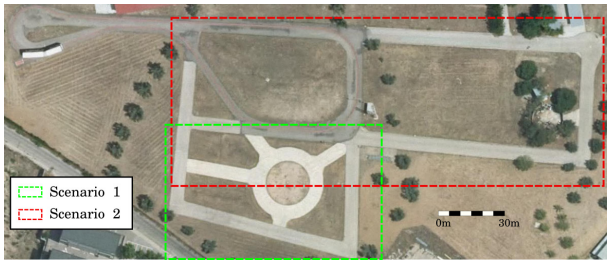


Fig. 9. Test track and scenarios location at Centre for Automation and Robotics (CSIC-UPM) in Arganda del Rey, Spain

Despite the fact that automated vehicle control involves coupled longitudinal and lateral dynamics, they can be considered as almost entirely decoupled when the intended applications comprise slight control actions [38]. In highways (high speed with low curvatures) or urban environments (low speed with high curvatures) lateral and longitudinal vehicle control systems are very often designed independently (e.g. [39]). Thus, the trajectory tracking used in this work is designed and behaves in a decoupled manner.

Both lateral and longitudinal controllers for trajectory tracking rely on fuzzy logic: On the one hand, the lateral controller uses both lateral and angular errors measured from the ego-vehicle pose with respect to the trajectory, to computed the steering wheel position. On the other hand, the longitudinal controller computes the positions of throttle and brake pedals from the reference speed (taken from the trajectory speed profile) and speed error. Further details about the trajectory tracking can be found in [40].

B. Motion planning results

Different trials have been carried out to test the performance of the proposed motion planning method. To perform them, the trajectory generator has been integrated in the architecture proposed in [34].

The proposed motion planning approach requires the specification of some parameters. On the one hand, those used to define the occupancy grid properties: distance ahead = 60 m and grid cell size 0.2 m. Although the algorithm has been designed to consider the localization uncertainties provded by the localization system at each instant, the following fixed values have been considered in the results shown below: $\sigma_x = \sigma_y = 0.02$ m and $\sigma_\theta = 0.05$ rad. The motivation for such high values is to make the proposed motion planning algorithm work under the worst conditions a vehicle equipped with RTK-DGPS can typically navigate. On the other hand, the parameters of the trajectory generation algorithm. They have been selected empirically in order to find a balance between computing time, grid accuracy and distance ahead the vehicle. In order to balance the different aspects represented by the cost function in (11), values of the weights are: $w_{og} = 10^{-5}$, $w_{L_p} = 10^3$ and $w_{\ddot{\kappa}} = 60$. In the experiments carried out, the trajectory generator was set up to compute a total of 4500 path candidates in each planning request. The most relevant parameters and values used in these experiments are shown in table II.

The proposed algorithm has been evaluated in two different driving scenes in a real environment (see Fig. 9): (i) an urban-like scenario with sharp curves and (ii) an urban-like scenario where obstacles must be avoided.

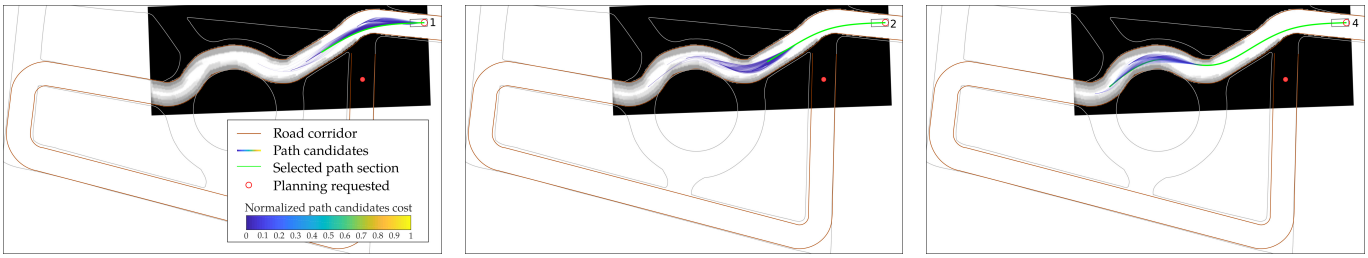
1) Scenario 1: Urban-like route through tight curves: This scenario consists of an urban-like layout where the vehicle has to drive through consecutive tight curves.

Fig. 10 shows 6 different instants of the trial when a new trajectory is requested. Each of them includes the occupancy grid, the valid candidates being evaluated and the selected path. In this figure, each candidate is represented by its path instead of its path-polygon for a better visualization. In general, it can be noticed that the space occupied by the selected path candidates (in green) comprise a very low occupancy probability. For example, in Fig. 10a, even if there are longer valid candidates, the selected candidate falls in an area that has less occupancy probability than the longer ones.

Fig. 11 shows the concatenation of all paths followed by the vehicle and the real vehicle path during the performed trial. The resulting path presents a smooth shape along the whole trial.

In Fig. 12, results of the trajectory tracking during the full trial in scenario 1 are shown. It can be seen that both lateral and angular errors used in the lateral control present a smooth behaviour, even though the trajectory is updated 19 times during the trial. Moreover, in Fig. 12b can be noticed that the curvature remains continuous along the whole path. Note that the higher the grid size, the longer the grid computing time. Finally, Fig. 12c shows the reference speed and measured vehicle speed during this trial.

Regarding the planning time, the mean of the processing time per planning request is 112.99 ms with a maximum of



(a) Planning request 1. Instant t=0.813 s (b) Planning request 2. Instant t=1.355 s (c) Planning request 4. Instant t=1.982 s

(d) Planning request 10. Instant t=18.044 s (e) Planning request 18. Instant t=41.662 s (f) Planning request 19. Instant t=44.223 s

Fig. 10. Valid and selected candidates at some of the planning requests during the trial in scenario 1

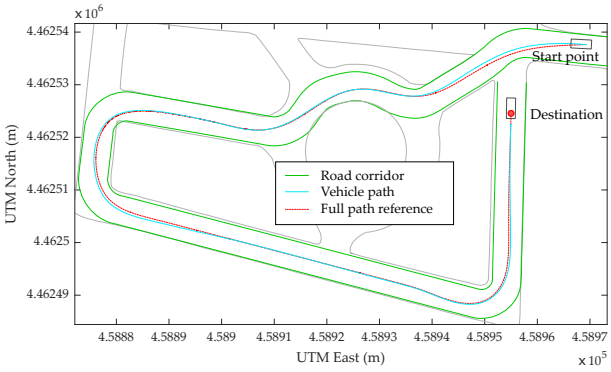


Fig. 11. Final reference path and vehicle path in the scenario 1

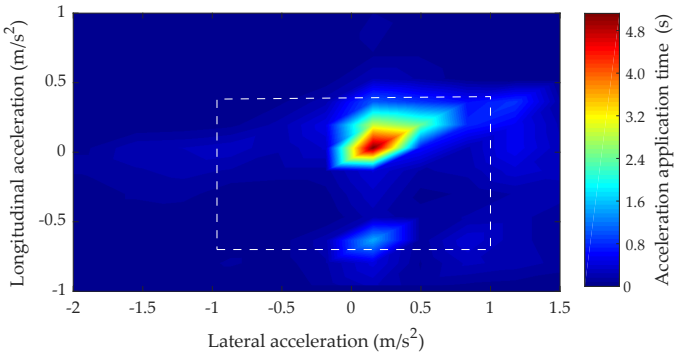
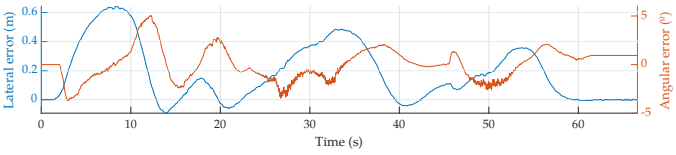
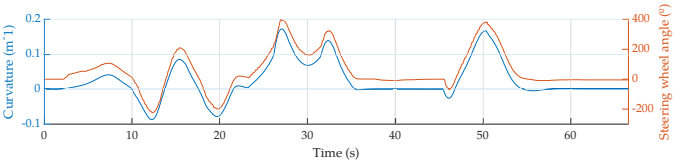


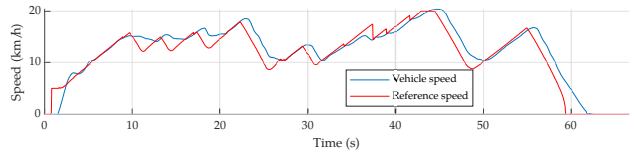
Fig. 13. Acceleration measured in the vehicle during the trial in scenario 1



(a) Lateral and angular control errors during the trial.



(b) Path curvature and steering wheel angle during the trial.



(c) Reference and vehicle speed during the trial.

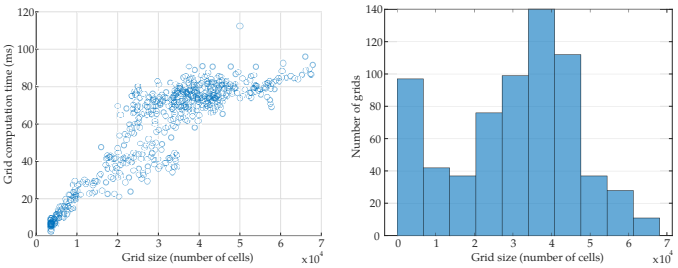
Fig. 12. Trajectory tracking in scenario 1

218.45 ms, showing a reasonable computing time.

To analyze the resulting behaviour of the vehicle in terms of occupant comfort, Fig. 13 shows a density plot of the real longitudinal and lateral accelerations to measured along the trial. It shows that most of the acceleration measurements fall within the dashed white rectangle that represent the stated acceleration limits.

Fig. 14a and 14b show the computation time and the number of grids of grids generated during this experiment, respectively. Note that the variability in grid sizes is due to variations in road geometry over the entire trial. The histogram in Fig. 14b depicts the distribution of the number of computed grids during the test (60 s approximately), according to their sizes. As can be seen, the size of most grids is between 20000 and 50000 cells in this case. Fig. 14a shows the computation time of localization uncertainty propagation on the grids with respect to their size of all the grids computed. In this case, the computation times of most grids concentrates between 60-90 ms, what is a reasonable value.

2) Scenario 2: Obstacles avoidance: This scenario includes two obstacles located at different places of the route that the



(a) Computation time of occupancy grid regarding the grid size.

(b) Histogram with respect of grid size.

Fig. 14. Occupancy grid results in scenario 1

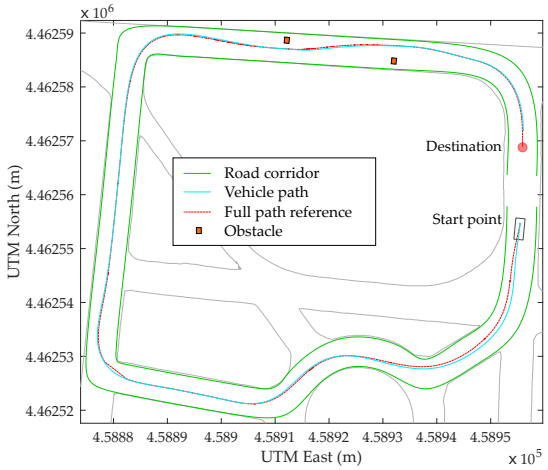
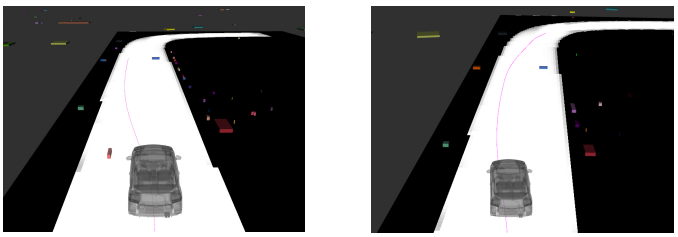


Fig. 15. Resulting paths in scenario 2

vehicle is following to reach the destination point. Thus, this scenario includes a greater complexity with respect to the previous one.

Fig. 15 shows the concatenation of final paths together with the real path of the vehicle. In this case, the trajectory is quickly corrected to avoid the obstacles satisfactorily even with noisy perception information. In addition, Fig. 16 shows two consecutive screenshots of the 3D visualization while the vehicle was avoiding the obstacles and Fig. 17 shows two front vehicle pictures at similar instants of the trial, where obstacles (two cardboard boxes) can been seen.

Regarding computing time, the average planning time for the whole trial was 193.18 ms with a maximum of 272.51 ms for the whole experiment. In comparison with the previous



(a) Avoiding first obstacle.

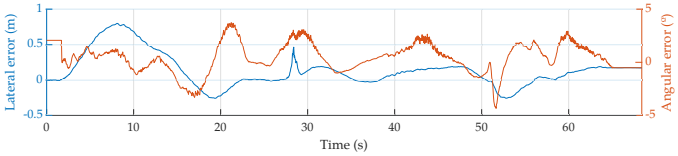
(b) Avoiding second obstacle.

Fig. 16. 3D visualization screenshots while avoiding obstacles

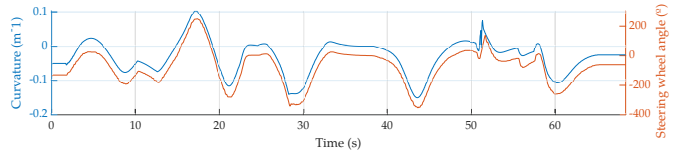
(a) Avoiding first obstacle.

(b) Avoiding second obstacle.

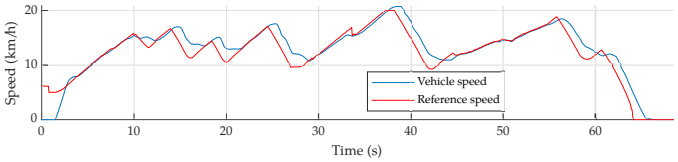
Fig. 17. Pictures of the frontal vehicle view while avoiding obstacles



(a) Lateral and angular control errors during the trial.



(b) Path curvature and steering wheel angle during the trial.



(c) Reference and vehicle speed during the trial.

Fig. 18. Trajectory tracking in scenario 2

case, the mean planning computation time has been increased due to the obstacles avoidance requests. Note that depending on the shape of the ahead section of the road corridor that fall within the grid, the amount of reference points used to generate path candidates vary along the trial. However, it is limited to the value of parameter n_{rp}. Taking into account that the occupancy grid has been parametrized to achieve a balance between computing time, ahead distance and planning space exploration, typically a lower number of reference points are used and consequently a lower amount of candidates are generated.

Fig. 18 depicts the same information about the trajectory tracking that has been shown in the previous case. As can be observed, the vehicle is able to smoothly follow the trajectory during the whole trial, even when the obstacles are being avoided.

In order to analyse the comfort inside the vehicle during the test, Fig. 19 shows a density map to represent the measured longitudinal and lateral accelerations. It shows how most of the measured acceleration values fall within the limits (marked with a dashed white rectangle) established in the planning. However, some values are outside mainly due to the joint effect of vibrations induced by road imperfections and vehicle pitching and rolling in slopy and bank stretches.

Finally, Fig. 20a and 20b show the computation time and the number of grids of grids generated during this experiment,

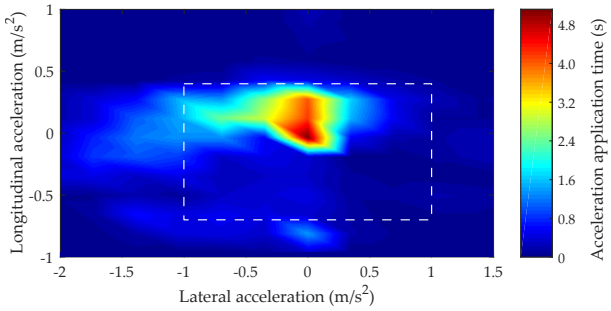
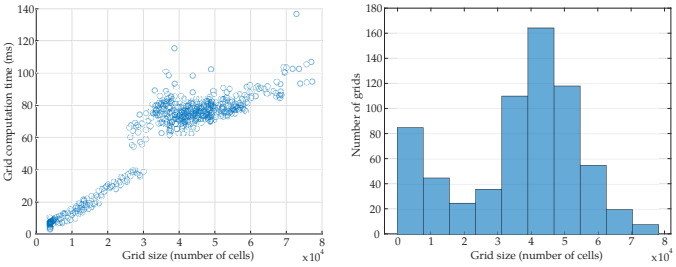


Fig. 19. Acceleration measured in the vehicle during the trial in scenario 2



(a) Computation time of occupancy grid regarding the grid size.

(b) Histogram with respect of grid size.

Fig. 20. Occupancy grid results in scenario 2

respectively. In this case, the computation times of the grids concentrates in around 75 ms for most of the grids computed during the test, whose sizes are mostly between 35000 and 55000 cells. Although this scenario includes obstacles, these results are similar to those obtained in the previous experiment. Note that the distribution of the grids according to their size is different from the previous case. This is caused by the different route followed in this scenario.

In conclusion, these experiments shows that the proposed motion planning algorithm using the occupancy grid as input is able to generate an important number of valid candidates and select the optimum one in a few milliseconds, even in sharp areas where consecutive curves must be overcome by the vehicle. Moreover, the behaviour of the proposed strategy is safer than the same path planning approach without considering localization uncertainty. Indeed, it can be observed how the obtained path tends to be more centred in the lane, as the probability of occupation is typically lower in this region than in the vicinity of the edges.

V. Conclusion

In this work, a probabilistic occupancy grid approach for motion planning that deals with localization uncertainty is proposed. This approach proposes a general way to fuse data coming from maps and perception systems on the occupancy grid. Moreover, a localization uncertainty propagation algorithm allows to compute the occupancy probability in each cell of the grid. Thus, the motion planning algorithms can be influenced by the current localization uncertainty of the vehicle.

Several experiments have been carried out in a real environment in order to test the performance of the proposed algorithms for localization uncertainty propagation and motion planning. The results show admissible computing times while safe and comfortable trajectories are generated.

Future work will focus on extending the capabilities of the approach by implementing the algorithms for parallelized computing architectures, as both localization uncertainty propagation and motion planning algorithm are highly parallelizable (for each grid cell and for each path candidate, respectively). In addition, machine learning techniques can be applied to enable a better use of the available computational resources when the computation time plays a critical role. On the other hand, the grid approach can be extended to consider the perception uncertainty as well as to investigate the effect of the ego-localization uncertainty in the perceived obstacles.

Acknowledgment

This work has been partially funded by the Spanish Ministry of Science, Innovation and Universities with National Project COGDRIVE (DPI2017-86915-C3-1-R), the Community of Madrid through SEGVAUTO 4.0-CM (S2018-EMT-4362) Programme, and by the European Commission through the Projects PRYSTINE (ECSEL-783190-2) and SECREDAS (ECSEL-783119-2).

References

[1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, pp. 33–55, March 2016.

[2] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, nov 2015.

[3] T. Gu, J. M. Dolan, and J.-W. Lee, "On-Road Trajectory Planning for General Autonomous Driving with Enhanced Tunability," in *Advances in Intelligent Systems and Computing*, pp. 247–261, 2016.

[4] T. Gu, J. Snider, J. M. Dolan, and J.-w. Lee, "Focused Trajectory Planning for autonomous on-road driving," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 547–552, IEEE, jun 2013.

[5] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 458–464, IEEE, jun 2014.

[6] J. P. Talamino and A. Sanfeliu, "Anticipatory kinodynamic motion planner for computing the best path and velocity trajectory in autonomous driving," *Robotics and Autonomous Systems*, vol. 114, pp. 93 – 105, 2019.

[7] J. Canny and J. Reif, "New lower bound techniques for robot motion planning problems," in *Foundations of Computer Science, 1987., 28th Annual Symposium on*, pp. 49–60, IEEE, 1987.

[8] S. Lazard, J. Reif, and H. Wang, "The complexity of the two dimensional curvatureconstrained shortest-path problem," in *Proceedings of the Third International Workshop on the Algorithmic Foundations of Robotics,(Houston, Texas, USA)*, pp. 49–57, Citeseer, 1998.

[9] Y. Wang, Z. Liu, Z. Zuo, Z. Li, L. Wang, and X. Luo, "Trajectory planning and safety assessment of autonomous vehicles based on motion prediction and model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 8546–8556, Sep. 2019.

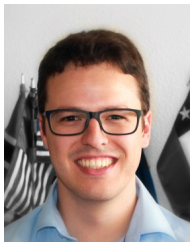
[10] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, *et al.*, "Making bertha drive—an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[11] C. L. Darby, W. W. Hager, and A. V. Rao, "An hp-adaptive pseudospectral method for solving optimal control problems," *Optimal Control Applications and Methods*, vol. 32, no. 4, pp. 476–502, 2011.

[12] L. S. Pontryagin, *Mathematical theory of optimal processes*. CRC Press, 1987.

[13] J.-C. Latombe, *Robot motion planning*, vol. 124. Springer Science & Business Media, 2012.

[14] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
[15] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 883–921, 2015.
[16] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2210–2215, IEEE, 2005.
[17] J. Villagra, V. Milanés, J. Pérez, and J. Godoy, "Smooth path and speed planning for an automated public transport vehicle," *Robotics and Autonomous Systems*, vol. 60, no. 2, pp. 252–265, 2012.
[18] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 7681–7687, IEEE, 2010.
[19] J. Villagra and H. Mounier, "Obstacle-avoiding path planning for high velocity wheeled mobile robots," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 49–54, 2005.
[20] A. Schindler, "Vehicle self-localization with high-precision digital maps," in *2013 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pp. 134–139, IEEE, jun 2013.
[21] P. Themann, J. Kotte, D. Raudszus, and L. Eckstein, "Impact of positioning uncertainty of vulnerable road users on risk minimization in collision avoidance systems," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1201–1206, IEEE, 2015.
[22] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 849–882, 2015.
[23] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.
[24] A. Niehaus and R. F. Stengel, "Probability-based decision making for automated highway driving," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 626–634, 1994.
[25] M. Althoff, *Reachability analysis and its application to the safety assessment of autonomous cars*. PhD thesis, Technische Universität München, 2010.
[26] M. Schreier, V. Willert, and J. Adamy, "An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2751–2766, 2016.
[27] J. Hardy and M. Campbell, "Contingency planning over probabilistic obstacle predictions for autonomous road vehicles," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 913–929, 2013.
[28] W. Xu, J. Pan, J. Wei, and J. M. Dolan, "Motion planning under uncertainty for on-road autonomous driving," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2507–2512, May 2014.
[29] 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*, pp. 1875–1881, 2016.
[30] 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)*, pp. 392–399, IEEE, oct 2014.
[31] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated Driving in Uncertain Environments: Planning With Interaction and Uncertain Maneuver Prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, pp. 5–17, mar 2018.
[32] J. Godoy, A. Artuñedo, and J. Villagra, "Self-generated osm-based driving corridors," *IEEE Access*, vol. 7, pp. 20113–20125, 2019.
[33] A. Zingl, "A Rasterizing Algorithm for Drawing Curves," *Multimedia und Softwareentwicklung*, 2012.
[34] A. Artuñedo, J. Godoy, and J. Villagra, "A decision-making architecture for automated driving without detailed prior maps," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1645–1652, June 2019.
[35] A. Artuñedo, J. Villagra, and J. Godoy, "Real-time motion planning approach for automated driving in urban environments," *IEEE Access*, vol. 7, pp. 180039–180053, 2019.
[36] A. Artuñedo, J. Godoy, and J. Villagra, "A primitive comparison for traffic-free path planning," *IEEE Access*, vol. 6, pp. 28801–28817, 2018.
[37] P. S. Heckbert, "Graphics gems," ch. A Seed Fill Algorithm, pp. 275–277, San Diego, CA, USA: Academic Press Professional, Inc., 1990.
[38] R. Rajamani, *Vehicle Dynamics and Control*. Mechanical Engineering Series, Springer US, 2011.
[39] R. Rajamani, Han-Shue Tan, Boon Kait Law, and Wei-Bin Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE Transactions on Control Systems Technology*, vol. 8, pp. 695–708, July 2000.
[40] J. Godoy, J. Pérez, E. Onieva, J. Villagra, V. Milanés, and R. Haber, "A driverless vehicle demonstration on motorways and in urban environments," *Transport*, 2015.



Antonio Artuñedo received a B.Sc. in Electrical Engineering from the Universidad de Castilla – La Mancha, Spain in 2011 and a M.Sc. in Industrial Engineering from the Universidad Carlos III de Madrid in 2014. In 2019, he received his PhD in Automation and Robotics at the Technical University of Madrid (UPM), Spain in the AUTOPIA Program. His research interests include system modelling and simulation, intelligent control, motion planning and decision making in autonomous vehicles.

Jorge Villagra graduated in Industrial Engineering at the Universidad Politécnica of Madrid in 2002. He received his PhD in Real-Time Computer Science, Robotics and Automatic Control at the École des Mines de Paris (France) in 2006. He is leading AUTOPIA Program at CSIC since October 2016. He has developed his research activity in 6 different entities with a very intense activity in project setup and management, through over 35 international and national R&D projects, where he is or has been IP of 14 of these projects. He has published over 90 papers in international journals and conferences on autonomous driving, intelligent transportation systems, model-free control and new probabilistic approaches for autonomous vehicles.

Jorge Godoy received the electronic engineer degree from the Universidad Simón Bolívar in 2008. In 2011, he received his M.E. degree in Automation and Robotics from the Technical University of Madrid, finishing his PhD degree in the same area on 2013. From 2013 to 2017 he was the technical coordinator of the AUTOPIA Program. In November 2017 he was granted with a Juan de la Cierva fellowship for postdoctoral research at the Technical University of Madrid. His research interest includes intelligent transportation systems, autonomous driving and embedded AI-based control for autonomous vehicles.

Maria Dolores del Castillo received her PhD in Physics in 1990, started her research career in CSIC where she is a Tenured Scientist. Her research areas have evolved from automating industrial process and generating robots' behavior, passing by knowledge discovery and computational linguistics to more recently, focused on building psychological and biological plausible behavior models and technological platforms for human-machine interfacing. All this work has been oriented to understanding, assistance or rehabilitation of brain and human being by participating in multidisciplinary teams. She is co-author in more than one hundred publications and two patents.