



**HAL**  
open science

# Evidential-Based Approach for Trajectory Planning With Tentacles, for Autonomous Vehicles

Hafida Mouhagir, Reine Talj, Véronique Cherfaoui, François Aioun, Franck  
Guillemard

► **To cite this version:**

Hafida Mouhagir, Reine Talj, Véronique Cherfaoui, François Aioun, Franck Guillemard. Evidential-Based Approach for Trajectory Planning With Tentacles, for Autonomous Vehicles. IEEE Transactions on Intelligent Transportation Systems, 2019, 21 (8), pp.3485-3496. 10.1109/TITS.2019.2930035 . hal-02367449

**HAL Id: hal-02367449**

**<https://hal.science/hal-02367449v1>**

Submitted on 21 Dec 2020

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

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

# Evidential based approach for trajectory planning with tentacles, for autonomous vehicles

Hafida Mouhagir<sup>1,2</sup>, Reine Talj<sup>1</sup>, Véronique Cherfaoui<sup>1</sup>, François Aioun<sup>2</sup>, Franck Guillemard<sup>2</sup>

**Abstract**—Driving is a combination of continuous mental risk assessment, sensory awareness, and judgment, all adapting to extremely variable surrounding conditions. Several research projects are working on autonomous vehicle to robotize this complex task. The work presented in this paper focuses on reactive local trajectory planning in uncertain environment.

The environment uncertainty is one of the challenges that we face in trajectory planning. For autonomous vehicle, to be efficient, they need to be able to deal with this kind of uncertainty. In this work, we show that the theory of belief functions with its ability to distinguish between different types of uncertainty is able to provide significant advantages in the context of trajectory planning.

Using the belief functions, we build evidential grid that represents the surrounding environment. To plan a local trajectory, we generate a set of clothoid tentacles in the egocentred reference frame related to the ego-vehicle. Those tentacles represent possible trajectories that consider the current dynamical state of the vehicle and make a smooth variation in the vehicle dynamic variables. Once the representation of the environment and the possible trajectories are generated, an evaluation of each trajectory is carried out according to several criteria and the choice of the trajectory is made using the decision formalism of Markov Decision Process.

To demonstrate the effectiveness of our evidential approach, we apply it to scenarios where ego-vehicle has to make decision in uncertain dynamical environments, using a driving simulator (SCANeR<sup>TM</sup>Studio).

## I. INTRODUCTION

Autonomous vehicles (AVs) development is currently at the heart of academic and industrial research [1], [2] because of many potential benefits: reducing traffic accidents, congestion, parking problems and pollution emissions. However, autonomous vehicle technology also presents new challenges. One important problem is how to plan a safe trajectory for a vehicle, given uncertain knowledge about the surrounding environment.

The main purpose of local trajectory planning is to provide the vehicle with a safe and collision-free trajectory, while taking into account vehicle dynamics, the presence of static and dynamic obstacles, and traffic rules.

Planning approaches for autonomous on-road driving have been inventoried and summarized in the survey [3]. There are two main categories of trajectory planning: incremental approaches and local approaches.

The authors are with <sup>1</sup>Sorbonne universités, Université de Technologie de Compiègne (UTC), CNRS Heudiasyc UMR 7253, <sup>2</sup>PSA Groupe, Direction scientifique, Centre technique de Vélizy, France. E-mail: {hafida.mouhagir, reine.talj, veronique.cherfaoui}@hds.utc.fr, {françois.aioun, franck.guillemard}@mpsa.com

Incremental approaches aim to find a complete path from the initial position to the final position based on the most possibly complete representation of the environment. Two techniques are primarily used: Rapidly-exploring Random Trees (RRTs) [4], [5] and Lattice Planners (LP) [6]. These approaches provide trajectories that match the dynamic and kinematic constraints of the vehicle. But these trajectories are discontinuous and not smooth, with a relatively long planning horizon, so a significant calculation time.

On the other hand, local approaches search for a local trajectory to follow for a computation step. This feature has a great advantage because it allows the vehicle to navigate in a dynamic and uncertain environment. Local approaches are also divided into two categories: Geometric Curve Optimization and Model Predictive Control (MPC).

The first approach is based on a geometric curve optimization, in which a single trajectory optimization is performed [7] or several lateral shifts of a curve (e.g. splines, Bezier curves, clothoids, polynomials...) are evaluated [1].

The Model Predictive Control approach combines aspects of control engineering within the planning module [8] [9]. Within MPC, a dynamic model for the vehicle is used and with the controller inputs the optimization problem of finding the best trajectory is solved. With this technique, the more variables used to model the vehicle the more harder it gets to find a real-time trajectory.

In our work, we used the clothoid tentacle approach [1], [10]. The main advantage of this method is combining the search space and taking into consideration the vehicle dynamics limits while being a fast reactive method. A set of clothoid tentacles is generated in the egocentered reference frame related to the vehicle. Generated tentacles in a egocentered grid represent smooth and feasible trajectories by the vehicle. Choosing the best tentacle to execute is based on a Markov Decision Process evaluating several criteria. One of the criteria that interests us in this paper is the occupation criterion evaluated through evidential grids (representation of the vehicle surrounding environment).

Autonomous vehicles interact with their environment through sensors and actuators. The goal is to choose the "right" action according to their internal state and the perception of the surrounding environment. In general, the physical limitations and computing resources make the environment only partially known by the ego-vehicle. As a result, a vehicle operating in a real environment faces many sources of uncertainty: the time course of driving situations can not be predicted without uncertainty because other road users behave in an unpredictable way and their goals and plans can

not be measured. More importantly, the ego-vehicle can only perceive part of the current situation with its sensors, because the measurements are noisy and part of the environment is occluded.

Partially Observable Markov Decision Process (POMDP) [11] is a framework used to solve the environment unpredictability as well as the uncertainty of the perception and other road users behavior. In [12], the authors use continuous POMDP in decision making to address both problems of noisy sensor measurements and the environment occlusion in intersection scenarios. This method is promising but remains very expensive in computing time.

In our study, we take into account the uncertainties of the environment through the use of the evidential occupancy grids.

The occupancy grids are used to model the environment. The ego-vehicle's surrounding environment is discretized into grid cells where each cell has a probability of being occupied. There are a variety of successful probabilistic algorithms based on occupancy grid maps for path planning [13], [14], and exploration [15].

One of the limitations of probabilistic occupation grids is the impossibility of distinguishing the various types of uncertainty [16]. The belief functions [17], [18] provide a powerful alternative to classical probability theory because the mass of belief can not only be assigned to an occupied cell  $O$  or free  $F$ , but also to the subset of the two states  $\{O, F\}$  as well as to the empty set  $\emptyset$ , thus providing the robot (ego-vehicle) with additional information on the environment. The mass assigned to  $\{O, F\}$  explicitly expresses ignorance (the robot has not observed the cell) whereas the mass attributed to  $\emptyset$  represents contradictory evidence, indicating contradictory measures were obtained.

Using a probabilistic grid, the uniform distribution of probabilities  $P(O) = P(F) = 0.5$  could correspond to ignorance or conflict, even if the reaction of the robot could be different. For example, in the case of high ignorance, the robot must move to the corresponding zone during the exploration in order to obtain measurements of the cell. On the other hand, a high degree of conflict indicates that the area is potentially dangerous because different states of occupation have been observed, and simply obtaining additional measures may not resolve the situation (for example, there may be unexpected dynamics in the environment as pedestrians moving around the robot).

The additional information provided by evidential grid maps have been used to solve problems in a number of works. Evidential occupancy grid maps were first proposed in [19] for a robot using sonar sensors, but other sensors like radar [20] and laser scanners [21] have also been used. Different combination rules have been investigated [22], [23]. A typical application of evidential grid maps, for example, is assessing the quality of maps [24] and detecting moving objects [25], [26].

In the context of trajectory planning under perception uncertainties, evidential grids were used [27], [28], where the approach was designed and validated for robots with

a very simple action configuration (moving from one cell to another). Our goal in this paper is to use the evidential grids for autonomous vehicles where the action space is more complicated (steering angle and/or deceleration or acceleration). The planning strategy used is the planning method with clothoid tentacles [29] that considers the current dynamical state of the vehicle and makes a smooth variation in the vehicle dynamic variables. To choose the best trajectory to execute, we use reward system of a Markov Decision Process-like model to evaluate generated tentacles regarding several criteria considering uncertainty represented by the evidential grid.

Based on our previous results [30][31], this paper presents a trajectory planning method combining the clothoids and MDP techniques with the use of evidential grids to integrate perception uncertainty. Then, it proves the performances of the approach by a global validation, under the simulator SCANer<sup>TM</sup>Studio. These simulation tools allow to integrate the dynamics of vehicle in closed-loop with different developed modules. Moreover, we have been able to test more complicated and realistic scenarios in the presence of several vehicles.

The remainder of this paper is structured as follows. In Section II, we present the trajectory planning algorithm based on clothoid tentacles as well as the construction of the evidential grids for trajectory planning. In Section III, we explain how we do use MDP reward system with evidential grids. First results using real data and SCANer<sup>TM</sup>Studio simulator are discussed in Section IV followed by a conclusion of the paper including an outlook in Section V.

## II. TRAJECTORY PLANNING STRATEGY

At a local on-road level, the trajectory planning goal is the computation of an obstacle free route while following a desired global reference trajectory defined on a global map. The generated trajectory must satisfy the vehicle's kinematic limits based on vehicle dynamics and constrained by the navigation comfort, respect lane boundaries and traffic rules, while avoiding, at the same time, static and dynamic obstacles.

Our trajectory planning strategy can be divided into three main steps (Fig. 1):

- Generating tentacles which will represent dynamically feasible trajectories.
- Creating and updating occupancy grid with data coming from exteroceptive sensors.
- Choosing of the best tentacle that the vehicle will execute.

### A. Clothoid tentacles for trajectory planning

In this work, we use the clothoid tentacles method for local on-road trajectory planning. This method is based on generating a set of clothoids tentacles as possible trajectories on a egocentric occupancy grid around the vehicle [29].

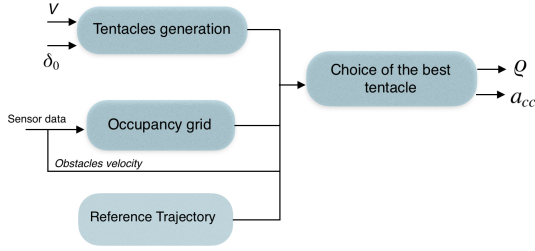


Figure 1: Trajectory planning strategy ( $V$ : the vehicle velocity,  $\delta_0$ : the current steering angle,  $\rho$ : the curvature setpoint and  $a_{cc}$  acceleration setpoint).

Clothoid is a curve whose curvature varies linearly with curvilinear abscissa, also known as an Euler spiral, Cornu spiral or linarc. Its expression is presented by (1):

$$\rho = \frac{2}{k^2} s \quad (1)$$

where  $\rho$  is the clothoid curvature,  $s$  is the curvilinear abscissa and  $k$  is a constant, representing the clothoid parameter.

The initial curvature  $\rho_0$  of the tentacles is calculated from the current vehicle steering angle  $\delta_0$

$$\rho_0 = \frac{\tan \delta_0}{L}$$

where  $L$  is the vehicle's wheelbase. For a fixed velocity, all the tentacles begin at the center of gravity of the vehicle and take the shape of clothoids (Fig. 2).

The last step for generating a set of tentacles is to specify the maximal curvature  $\rho_{max}$  that the vehicle can execute at its current speed, without losing its stability:

$$\rho_{max} = \frac{a_{max}}{V_x^2} \quad (2)$$

where  $a_{max}$  is the maximum lateral acceleration of the vehicle and  $V_x$  is the current speed.

Thereafter, we generate a set of clothoid tentacles with a curvature that varies from  $-\rho_{max}$  to  $\rho_{max}$  within a given distance from the origin of the tentacles. The length of tentacles increases with the increase of the velocity. We assume that all tentacles generated for a given velocity have the same length.

After tentacles generation, we classify them as navigable or not navigable using the information of the occupancy grid. Each tentacle is overlaid on the occupancy grid and a support zone is defined around it. The support zone's width corresponds to Ego-vehicle width. Then, the occupancy of the cells of the support zone is evaluated. If a tentacle passes by an obstacle on a radius of  $L_s = 1s * V_x$  (with  $V_x$ : the ego-vehicle velocity) in front of the Ego-vehicle, this tentacle will be classified as not navigable, otherwise it's navigable (Fig. 2).

In case no tentacle is navigable, we choose the clearest tentacle (with the largest distance from the obstacle which is the curvilinear distance along the tentacle from the clothoid root to the intersection with the obstacle) to brake, and we re-evaluate again the tentacles in the next iteration. This enables a safe navigation while decelerating.

The main advantage of the clothoid approach is taking into

consideration the current dynamical state of the vehicle and making a smooth variation in the vehicle dynamic variables.

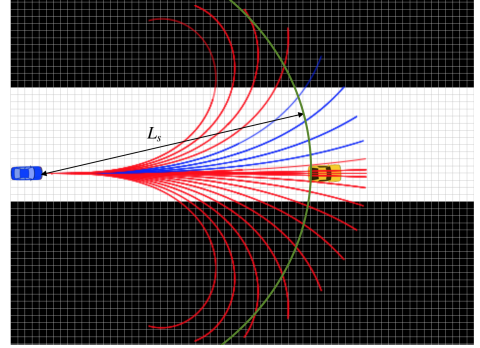


Figure 2: Clothoid tentacles generated on an egocentric occupancy grid (black space represents occupied space, and white space represents free space). Red tentacles are not navigable and the blue ones are navigable. Road edges are considered obstacles.

The selection of the best navigable tentacle to execute is based on several criteria:

- The occupation criterion to evaluate if the tentacle is free from obstacles using occupancy grid.
- The trajectory criterion to evaluate the distance between each tentacle and the reference trajectory.
- Traffic rules criterion: we take into consideration the respect of safety distance while overtaking.

In the following, we will explain how we build an occupancy grid used to evaluate the occupancy of the environment surrounding the vehicle.

### B. Evidential occupancy grid

The occupancy grids are used as environment model. If the grid cells are filled with obstacle information in the form of evidence (mass or belief values for instance II-B), we call this kind of grids the "Evidential occupancy grids".

#### Evidential framework

The theory of belief functions, also known as Dempster-Shafer theory (DST), was proposed by Dempster [32], and developed, among others, by Shafer [17] and Smets [33]. This formalism gained its popularity thanks to various interesting properties. DST not only generalizes the probability theory, but the possibility theory as well.

Let  $w$  be an unknown quantity with possible values in a finite domain  $\Omega$ , called the frame of discernment. A piece of evidence about  $w$  may be represented by a mass function  $m$  on  $\Omega$ , defined as a function  $2^\Omega \rightarrow [0, 1]$  such that  $m(\emptyset) = 0$  and  $\sum_{A \subseteq \Omega} m(A) = 1$ .

In the theory of Dempster-Shafer, a frame of discernment  $\Omega$  is defined to model a specific problem. In the occupancy grid framework, the frame of discernment is defined as:  $\Omega = \{O, F\}$ , referred as the states (occupied or free) of each cell. The power set is defined as  $2^{|\Omega|} = \{\emptyset, F, O, \Omega\}$ , with  $|\Omega|$  is the cardinality of the set.

For quantitatively supporting the cell states, a mass function (also referred as Basic Belief Assignment *BBA*) is calculated and provides four beliefs  $[m(F) m(O) m(\Omega) m(\emptyset)]$ ,

where  $m(A)$  represents respectively the quantity of evidence that the space is *Free*, *Occupied*, *Unknown* or *Conflict*.

### Evidential occupancy grids generation

In this section, we present the perception grids used in our approach. The construction of these grids is based on three kind of input data: 1) Data coming from a range sensor in order to be able to have information about the occupancy of the cells. 2) Road limits or lane information in order to distinguish between free navigable or not navigable space. 3) Absolute velocity of dynamic obstacles.

The first step consists on building the *PerceptionGrid* from different data sensors. For every sensor measurement, a *Grid* is built with sensor model that translates the sensor information into an ego-centered grid. The value of masses depends of the resolution of the grids and sensor performances. The different *Grids* are combined in an unique resulting *PerceptionGrid* (Fig. 5.a).

After *PerceptionGrid* processing, each cell has a mass function with four beliefs on the state of the cell  $[m(F) m(O) m(\Omega) m(\emptyset)]$ . Let consider a concrete case to illustrate these concepts,  $[m(F) m(O) m(\Omega) m(\emptyset)] = [0.7 0.3 0]$  indicates an *Occupied* cell with 0.7 as a belief, the rest of the mass is in *Unknown*.  $[m(F) m(O) m(\Omega) m(\emptyset)] = [0.6 0 0.4]$  shows a belief of 0.6 in *Free* state, the rest of mass being on *Unknown*.

The *PerceptionGrid* gives us information about the occupation of the environment, this information helps us to plan a trajectory to avoid collisions. However, autonomous vehicles must be capable of avoiding collision, lane keeping and carrying out an overtaking maneuver while keeping safety distances. Therefore, we build a *PlanGrid* (planning grid) based on the *PerceptionGrid* by adding road limits and by expanding obstacles on the road to include safety distances.

The second step as mentioned above is using map information to add a simple mask to the *PerceptionGrid* in order to integrate the edges of the road by changing the BBA of their cells (Fig. 5.b).

The next section explains how we include safety distances.

### C. Respect of safety distance

In Europe road regulation, road driving users should, under normal conditions, maintain a minimum distance from vehicles ahead, while following and after an overtaking manoeuvre. This distance corresponds to the distance traveled during a time interval of two seconds (Fig. 3).

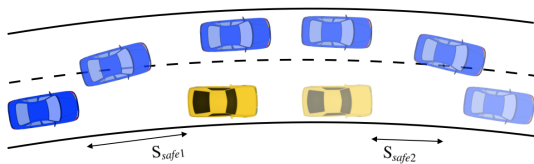


Figure 3: Longitudinal safety distance before ( $S_{safe1}$ ) and after ( $S_{safe2}$ ) overtaking.

To respect safety distances, we propose a longitudinal expansion of obstacles on the road in the occupancy grid.

In a first version of our algorithm [34], we expanded the obstacles by the distance travelled by it in two seconds taking into consideration its velocity. But this distance was too conservative therefore we adopted the formula of [35] that takes into consideration the relative velocity between the two vehicles and the difference between human and machine reaction time:

$$S_{safe}(v_p, v_f, a_{max}, \delta) = \frac{1}{2a_{max}}(v_f^2 - v_p^2) + v_f\tau \quad (3)$$

where  $v_f, v_p$  are respectively the velocity of the following and preceding vehicles,  $a_{max}$  is a potential acceleration and  $\tau$  is the reaction delay. The difference in reaction time between the human being ( $\tau_{human} = 2 s$  [36]) and the machine ( $\tau_{machine} = 0.3 s$  [37]) will make it possible for the ego-vehicle to harmonize with traffic flow while guaranteeing a safe distance.

The expansion is made by adding circles ( $\varphi$ ) with varying diameter which satisfies the following equation:

$$d_{i-front} = d_0 - i \frac{d_0 - 0.5}{S_{safe2}} \quad i = 1..S_{safe2} \quad (4)$$

with  $d_0 = 3m$  is the diameter of the circle representing the preceding vehicle ( $\varphi_1$ ).  $S_{safe1}$  (Fig. 4) represents the safety distance to keep before overtaking.  $S_{safe2}$  (Fig. 4) is the second safety distance to keep between the two vehicles in the end of the overtaking maneuver.

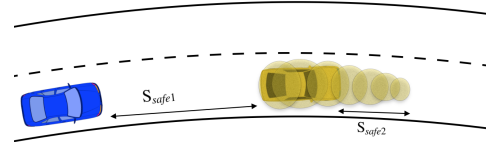


Figure 4: Example of an obstacle widening

This longitudinal extending of the obstacle is done by changing the masses attributed to the front cells of an obstacle within a safety distance (Fig. 5d). The spatial propagation modeling of an obstacle according to its velocity is inspired by the discounting operation method. Discounting in its basic form requires the decay factor  $\alpha$  to be specified, and it is defined as follows:

$$\alpha^k m_{C_{ij}}(O) = (1-\alpha_k) \cdot m_{C_{ij}}(O) + \alpha_k \text{ if } C_{ij} \in \varphi_k \quad (5)$$

$$\alpha^k m_{C_{ij}}(B) = (1-\alpha_k) \cdot m_{C_{ij}}(B) \quad \forall B \neq O$$

with  $m_{C_{ij}}(O)$  being the mass about the *Occupied* state of the cell with the coordinate  $(i, j)$ . The decay factor  $\alpha$  decreases as follows:

$$\alpha_k = \alpha - k \frac{\alpha - 0.02}{S_{safe2}} \quad k = 1..S_{safe2} \quad (6)$$

with  $\alpha = 0.8$ .

Figure 5 illustrates the different stages of the *PlanGrid* construction. The resulting *PlanGrid* provides a virtual occupation of the environment, for example the road edges are

considered occupied even if they are free of obstacles in order to respect traffic rules.

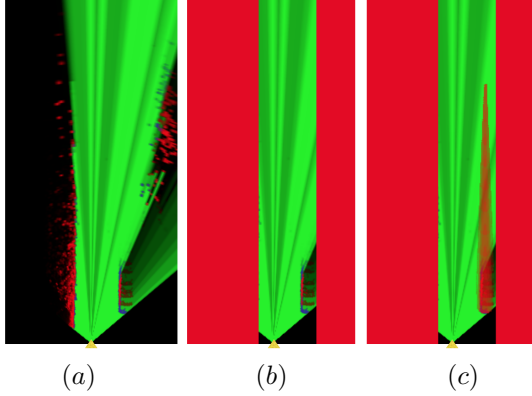


Figure 5: The yellow triangle represents the Lidar position in the front of the vehicle. The *PlanGrid*(b) represents *PerceptionGrid*(a) with the road's edges information. The *PlanGrid*(c) presents a longitudinal obstacle widening.

### III. MARKOV DECISION PROCESS BASED ON EVIDENTIAL GRIDS

#### A. Markov Decision Process

After classifying the tentacles, we evaluate the navigable tentacles using several criteria: the tentacle's occupation, its distance from the reference trajectory and the overtaking criterion. To model the problem of planning with all the criteria to be taken into consideration [30], we used a *Markov Decision Process* (MDP) like model.

A MDP is a discrete-time state-transition system. The agent (Ego-vehicle) observes the state (environment around each tentacle) and performs an action (tentacle execution) accordingly. The system then makes a transition to the next state and the agent receives some reward.

It can be described formally with 5 components  $(S, A, T, R, \gamma)$

- $S$  is the set of states, they are represented by circles around tentacles of diameter  $2m$  (Fig. 6), their diameter represents the width of the vehicle. Each tentacle is composed of  $n_s$  states, and we generate  $n_t$  tentacles
- $A(s) : S \rightarrow A$  is the set of actions, each tentacle represents an action, so we dispose of  $n_t$  actions
- $T : S \times S \times A \rightarrow [0, 1]$  defines the transition probabilities from one state to another, we assume that we do not have a possible transition from one tentacle to another
- $R : S \times A \rightarrow \mathbb{R}$  is the reward given to the agent depending on its current state and the action taken. We define a different reward for each state according to several criteria mentioned above
- $\gamma \in [0, 1]$  is the discount rate used to calculate the long-term attenuation

#### Reference trajectory criterion

The reference trajectory corresponds to the global path that the ego-vehicle must follow all the time. However, the Ego-vehicle can deviate from the reference trajectory to avoid

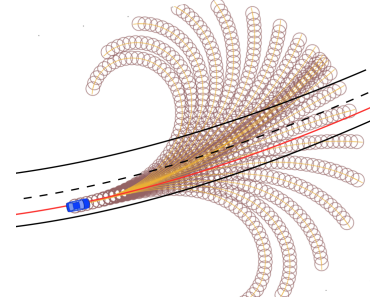


Figure 6: Modeling the problem of planning with tentacles using MDP. The red line represents the reference trajectory.

obstacles or to carry out an overtaking but must return to it. We use the lateral distance (Fig. 7) between each tentacle and the reference trajectory to evaluate the tentacles regarding this criterion. Details are presented in [30].

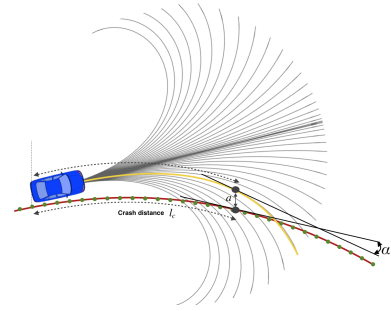


Figure 7: The distance between the tentacle and the reference trajectory.

The crash distance  $l_c$  is the distance required to stop a vehicle with a velocity  $V_x$ , with a maximum longitudinal deceleration  $a_m = 1.5 \text{ m/s}^2$  which maintains the comfort of the passengers. It is calculated by:

$$l_c = \frac{V_x^2}{2a_m} \quad (7)$$

For each tentacle, a set of measurements  $d_i$  is calculated by taking both the lateral distance  $a_i$  and relative tangent orientations  $\alpha_i$  between the tentacle and the reference trajectory. Each  $d_i$  is calculated at a curvilinear distance,  $\kappa_i l_c$  from the vehicle position, on the tentacle

$$d_i = a_i + c_\alpha \alpha_i \quad (8)$$

here,  $c_\alpha$  represents a scale between the linear distance and the tangent's angle.

We combine all the distances as follows:

$$d = \sum_{i=1}^n \lambda_i d_i \quad (9)$$

with  $\lambda_i$  is weighting constants.

All states of the same tentacle receive a reward  $R(s_{|trajectory})$  according to the distance  $d$ :

$$R(s_{|trajectory}) = R_{trajectory} - d \quad (10)$$

with  $R_{trajectory} = R_b$  a fixed constant reward corresponding to the maximum reward related to the proximity to the reference trajectory.

### Overtaking criterion

A small additional reward  $R(\text{overtaking}) = R_l$  is added to the tentacles of the left. This reward takes effect in case of total symmetry of tentacles ( $\phi_0 = 0$ ) when executing an overtaking manoeuvre, which allows an overtaking by the left.

### Occupancy criterion

Each tentacle is evaluated in regard of its occupation. Grid information is used to assign appropriate rewards.

A tentacle discretization is made by using circles (their diameter represent the width of the vehicle with a margin of security). This discretization helps us to judge the tentacle's occupancy and serves to have a support zone around the tentacle which will allow the Ego-vehicle to circulate in a secure manner. Rewards are given to each circle based on its occupancy.

Using the binary grid, the state (circle) is considered as occupied if more than  $f_s$  (threshold) cells inside the state are occupied, otherwise it is considered as a free state. Then, an occupation reward is given

$$R(s_{|occupancy}) = \begin{cases} R_o < 0 & \text{if } s \text{ is occupied} \\ R_f > 0 & \text{if } s \text{ is free} \end{cases} \quad (11)$$

During overtaking scenarios while using binary grids, the ego-vehicle had a conservative behavior due to the widening of obstacle, because this widening is considered as a real obstacle. To rectify this, we used evidential grids instead. Explanations on the calculation of the occupancy reward using evidential grids are given in the next section.

Then the total reward of a tentacle is:

$$R(\text{tentacle}) = \sum_{k=0}^{n_s} \gamma_t^k R(s_k|trajectory) + \sum_{k=0}^{n_s} \gamma_o^k R(s_k|occupancy) + R(\text{overtaking}) \quad (12)$$

where  $\gamma_t$ , and  $\gamma_o$  (Equ. 12) are discount factors that can be used to change the behavior of our approach, and that represent distance attenuation of each kind of reward.  $n_s$  is the number of states per tentacle,  $s_k$  is the state number  $k$  in the tentacle.

The planning algorithm has been evaluated using binary grids [30], [34]. We noted that the selected trajectories are conservative. Therefore, we propose to use evidential grids which contain more information that we can use during the decision-making process (selecting a tentacle).

With the use of the evidential grids in place of the binary grids, the trajectory criteria and overtaking criteria remain unchanged. On the other hand, the occupancy criterion through the occupation reward changes because we have more information about the cell occupation that we can exploit with the evidential grids.

### B. Occupancy Reward definition based on evidential grid

We dispose of an evidential grid in which we draw states as circles around each tentacle. The overlaying of the states

on the grid gives matrix storing cells, with belief mass values (Fig. 8).

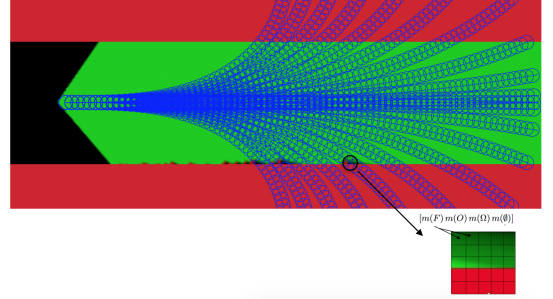


Figure 8: The discretization of each tentacle in circles (states), and the result of the overlaying with the cells of a *PlanGrid*

In order to define a reward regarding the occupancy of the state, we propose to process cells information using three different rules [38], [31]. We consider that each cell is a source of information about the occupancy of the state. All cells are defined in the same frame of discernment. For each rule, we attribute a different reward (Equations 11 to 13, where  $a_1, a_2, a_3, a_4$  are weighting parameters):

- **Conjunctive rule:** the first rule consists in combining all masses of the state matrix with conjunctive rule, the resulting mass function is  $m_{\cap}() = \cap_{i=1}^n m_{cell_i}()$ ,  $cell_i \in matrix$

$$Reward_{occupation-conj} = a_1 m_{\cap}(F) + a_2 m_{\cap}(O) + a_3 m_{\cap}(\Omega) + a_4 m_{\cap}(\emptyset) \quad (13)$$

with  $m_{1\cap 2}(A) = \sum_{B\cap C=A} m_1(B)m_2(C)$ ,  $\forall A \subseteq \Omega$  is the resulting mass using conjunctive rule proposed by Smets [18]. This rule is used to combine several BBAs provided by reliable and distinct information sources. By applying this rule, we obtain a consensus between all sources of information. The mass assigned to the empty set  $m_{1\cap 2}(\emptyset)$  quantifies the degree of disagreement between the two combined sources.

- **Dempster's rule:** the second rule consists in combining all masses of the state matrix with Dempster's rule [17], the resulting mass function is  $m_{\oplus}() = \oplus_{i=1}^n m_{cell_i}()$ ,  $cell_i \in matrix$

The occupation reward is defined:

$$Reward_{occupation-demp} = a_1 m_{\oplus}(F) + a_2 m_{\oplus}(O) + a_3 m_{\oplus}(\Omega) \quad (14)$$

with  $m_{1\oplus 2}(A) = K m_{1\cap 2}(A)$ ,  $\forall A \subseteq \Omega$  and  $m_1 \oplus m_2(\emptyset) = 0$  where  $K = (1 - m_{1\cap 2}(\emptyset))^{-1}$ . This rule is a normalized version of the conjunctive rule where the mass of the empty set must be reallocated over all focal elements when  $m_{1\cap 2}(\emptyset) \neq 0$  using a normalization factor  $K$ .

- **Cells number:** with this rule, we count the number of occupied, free and uncertain cells of the state matrix by

making a decision about their state. For that, we attribute the element  $A \in 2^\Omega$  if  $m(A) > 0.5$

$$\begin{aligned} \text{Reward}_{\text{occupation-cellNumb}} = & a_1 Nb(F) + a_2 Nb(O) \\ & + a_3 Nb(\Omega) \end{aligned} \quad (15)$$

In the next section, we present some results with a study of the different parameters  $a_1, a_2, a_3, a_4$

## IV. RESULTS

### A. System set-up and real example

There are three inputs in our perception system: vehicle pose, exteroceptive acquisition data and a map. First, a globally referenced pose is needed to localize the vehicle in the environment in terms of position and orientation compared with reference trajectory. The pose can typically be provided by a GPS system hybridized with an inertial measurement unit. Secondly, we use a *Lidar* as a perception sensor. This sensor can distinguish between free, occupied or uncertain space and to model it in  $2D$  ( $x, y$  coordinates) with respect to the vehicle body frame. We assume that we have the velocity of obstacles using Radar or car-to-car communication. In the validation tests, we used two vehicles: one with the *Lidar* sensor and the second one served as an obstacle to overtake with the velocity information. Finally, the map data provides information about the road surface.

The *Lidar* data was acquired at  $10 \text{ Hz}$  frequency. The ego-motion between two acquisitions is estimated using *CAN* data. For the purpose of demonstration, the scan grids of  $(20 * 40 \text{ m}^2)$  are built with uniform cells of size  $(0.1 * 0.1 \text{ m}^2)$ . We used the evidential grids generated by a C++ code [39] with data acquired on the experimental platform PACPUS [40].

The values of the other parameters of our algorithm are given in the following table:

$c_\alpha$	$\kappa_1$	$\kappa_2$	$\kappa_3$	$\lambda_1$	$\lambda_2$	$\lambda_3$	$R_t$	$R_l$
0.7	$\frac{1}{10}$	$\frac{1}{2}$	1	10	2	$\frac{1}{3}$	30	0.5

Table I: Values of the planning algorithm parameters.

### B. Choice of occupancy reward parameters and evaluation

One typical scene is chosen to compare the different combination approaches. The resulting *PlanGrid* is shown in Figure 9. The ego-vehicle velocity was set at  $20 \text{ m/s}$  and the preceding vehicle's velocity was  $14 \text{ m/s}$

The values of the parameters of the combination rules were determined empirically.

Rule	$a_1 : 1 \rightarrow 100$	$a_2 : -100 \rightarrow -1$	$a_3 : -20 \rightarrow 10$	$a_4$
Conj.	10	-10	-1	-10
Demp.	50	-20	-1	-
Cell-N.	20	-50	-2	-

Table II: Parameters of different combination rules.

During our tests, we collected perception data as evidential grids. These grids have been processed with *Matlab* as an

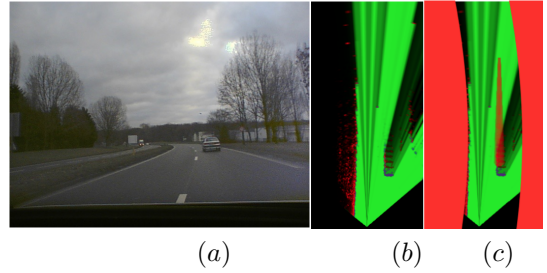


Figure 9: (a) represents a picture of the scene. (b) represents corresponding occupancy grid where Green represents Free space, Red represents Occupied space, black represents Unknown space and Blue represents Conflict cells. (c) represents the expansion of the mobile obstacle with road's edges adding.

input to our planning algorithm. We tested the different rules of combination in an overtaking situation using the *PlanGrid* of Figure 9. The criterion used to compare them is the safety distance at the end of the overtaking maneuver and the calculation time.

And in order to compare with the binary grids, we transform the *PerceptionGrid* (Fig. 9-c) into a binary grid using pignistic transformation. A cell is considered to be occupied if  $\text{bet}P(O) > \text{bet}P(F)$ , and free otherwise; with  $\text{bet}P(O) = m(O) + \frac{1}{2}m(\Omega)$  and  $\text{bet}P(F) = m(F) + \frac{1}{2}m(\Omega)$ .

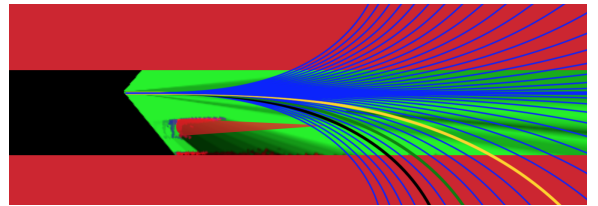


Figure 10: The chosen trajectory with three different combination rules. Cell-N rule: black, Conjunctive and Dempster rule: green. In yellow, the chosen trajectory with binary grid.

Rule	Conjunctive	Dempster	Cell-number	B.Grid
Time (s)	3, 3T	4, 7T	T	0.7T

Table III: Time computing for different combination rules and for the binary Grid. The  $T$  represents the duration of an iteration.

The result shows that the use of evidential grid enables us to process information about the unknown differently from the occupied space. In Figure 10, we observe that the binary grid choice is more conservative than the evidential grid ones. However, the used combination rules require a significant computation time (Table. III) comparing to the cell-number rule and the binary grid approach. More we need to consider uncertainties on perception, more the computation time is increased. Hence, the choice of the rule will be a compromise between reducing computation time and improving the planning behavior. Note that the computation time could be reduced by modifying the grid resolution.

### C. System set-up for simulation results

To test our trajectory planning algorithm, we performed co-simulations using *SCANeR*™ studio, a vehicle simulator, and *Matlab/Simulink*. The diagram in Fig. 11 summarizes the



overall architecture of co-simulation between SCANer™ studio and Matlab/Simulink. Using the SCANer™ studio sensors, the information necessary for the construction of the occupancy grid is collected as well as the data on the vehicle dynamics and the reference trajectory (center of the right lane). This information is then processed as input data of our planning algorithm, which sends after processing an acceleration or deceleration setpoint and a steering angle setpoint that the ego-vehicle must execute on the SCANer™ studio scenario.

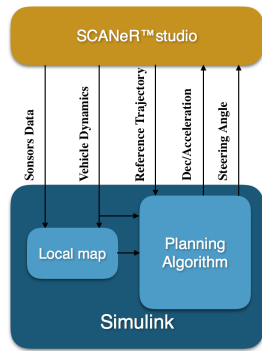


Figure 11: Algorithm validation scheme under Scanner Studio and Simulink.

### SCANer™ studio Sensors

In order to build a representation of the environment surrounding the ego-vehicle through occupancy grids, we could use the simulator ground truth but this solution is not generic with our simulator. So, we opted for the use of the information provided by different sensors available on SCANer™ studio. And for time computational reasons, simplified models of these sensors have been used to create simplified evidential grids.

Indeed, a reconstruction of the environment requires the detection of the obstacles present and the determination of the navigable space as well as the information on the road. Figure 12 shows the sensor configuration adopted on SCANer™ studio.

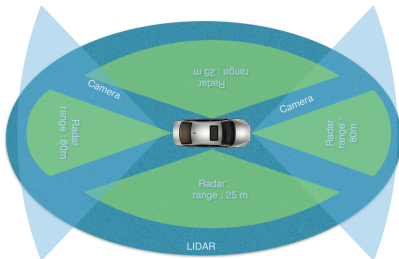


Figure 12: Diagram of sensor configurations adopted on SCANer™ studio.

This configuration consists of:

- 4 Radars: two front and rear radars with a range of 80m and horizontal field of view of 60° and vertical of 40°. Two right and left radars with a range of 25m and horizontal field of view of 120° and vertical of 40°.

- Camera with horizontal field of view of 120° and vertical of 60°.
- LIDAR with a horizontal field of view of 360° with a pitch of 3° and a range of 80m.

### Creating the binary grid

To create a binary occupancy grid, we use two types of sensors: the radar for the detection of obstacles and the camera for detecting the edges of the road in front of the ego-vehicle. All cells in the grid are initialized to zero (0) as a free state.

The radar provides the position of the obstacles as well as their type (car, pedestrian, truck) and velocity. Depending on the position and the category of the obstacle, we attribute value "1" to the occupied cells (for example, a car has a width 1.5m and a length of 4m). The rest of the cells are considered free (value "0"). The camera provides us with information about road edges in front of the ego-vehicle which we consider as an occupied space (cell filled by the value "1").

On a SCANer™ studio scenario and with a frequency of 10 Hz, our algorithm generates binary grids at each time step. Figure 13 shows an example of the resulting bit map and the corresponding scene.

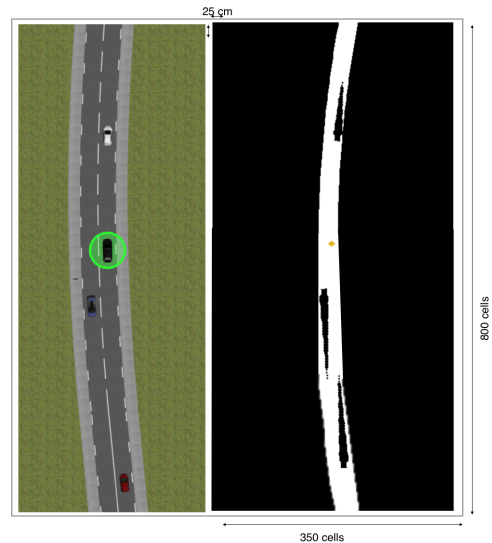


Figure 13: Example of a binary grid and the corresponding scene. Black represents occupied space and white represents free space. The black box in the middle shows the position of the ego-vehicle on the binary grid.

### Creating the evidential grid

The creation of an evidential occupancy grid [41] is based on the combination of three grids because each sensor provides information about a component of the environment. The camera provides information about the edges of the road while the radar provides on the position, type and velocity of obstacles and the Lidar provides information on free space.

The creation of an evidential occupancy grid steps are:

1. Creation of the road grid: from the road marking points coming from the camera sensor, we determine the borders

of the road. This detection is done only in front of the ego-vehicle. The cells  $C_{ij}$  of coordinates  $(i, j)$  belonging to the borders of the road have the following mass:  $m_1(C_{ij}) = [m_1(\emptyset) m_1(F) m_1(O) m_1(\Omega)] = [0 0 0.6 0.4]$

The rest of the cells have the following mass:  $m_1(C_{ij}) = [m_1(\emptyset) m_1(F) m_1(O) m_1(\Omega)] = [0 0 0 1]$

2. Creation of the Radar grid (obstacles): from the information on the obstacles detected by the Radar sensor, namely the classification, the position, the speed, the yaw angle, and the dimensions, the cells occupied are determined and a widening of obstacle is made. The occupied cells have the following mass:  $m_2(C_{ij}) = [m_2(\emptyset) m_2(F) m_2(O) m_2(\Omega)] = [0 0 0.8 0.2]$

The rest of the cells have the following mass:  $m_2(C_{ij}) = [m_2(\emptyset) m_2(F) m_2(O) m_2(\Omega)] = [0 0 0 1]$

3. Creation of the Lidar grid (free space): from the points of impact obtained with the Lidar sensor, we calculate the cells which constitute the free space around the ego-vehicle. The free cells have the following mass:  $m_3(C_{ij}) = [m_3(\emptyset) m_3(F) m_3(O) m_3(\Omega)] = [0 0.75 0 0.25]$

The rest of the cells have the following mass:  $m_3(C_{ij}) = [m_3(\emptyset) m_3(F) m_3(O) m_3(\Omega)] = [0 0 0 1]$

4. Combination of the grids: let  $C_{ij}$  be a coordinate cell  $(i, j)$  of the final grid, the allocation of the masses is carried out according to this equation:

$$m(C_{ij}) = \begin{cases} m_1(C_{ij}) & \text{if } m_1(O) > m_2(O) \\ m_2(C_{ij}) & \text{if } m_1(O) < m_2(O) \\ m_3(C_{ij}) & \text{if } m_1(O) = m_2(O) = 0 \end{cases} \quad (16)$$

The choice of the BBA values (initial masses) was made for the purpose of having  $m(O)$  of cells occupied by obstacles greater than  $m(O)$  of the cells occupied by the road edges. The cells corresponding to obstacles represent real occupied cells, however, the road borders don't represent a real occupation, but should be avoided to respect the road driving rules.  $m_2$  and  $m_3$  are generated from simulation of Radar and Lidar sensors, which give information about the occupation of the neighboring space [25].  $m_1$  are generated from the information of a map on the road borders [42], which are zones to be avoided, but can be tolerated if necessary for emergency situations.

Figure 14 shows an example of the three grids (Road, Radar, Lidar) and the final occupancy grid.

#### D. Simulation results

The tests presented below were carried out in order to test our planning algorithm taking into account the dynamics of the vehicle on different driving scenarios. The occupation grids are generated at a frequency of 10 Hz. At each computation step, the algorithm generates 41 tentacles and evaluates each tentacle using the three decision criteria:

- The occupation criterion: this criterion allows to evaluate if the tentacle is free from obstacles
- The reference trajectory criterion: this criterion evaluates the proximity of each tentacle to the reference trajectory

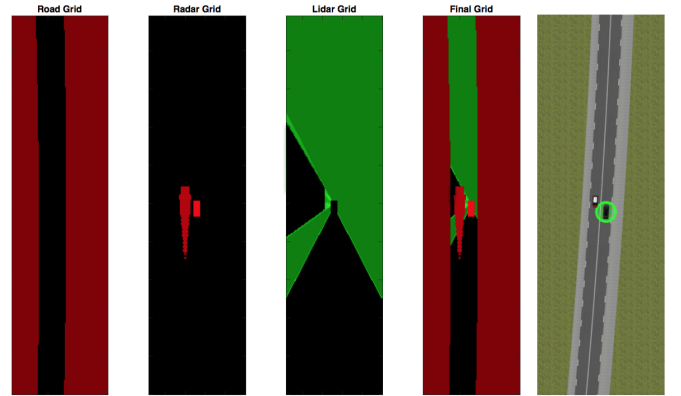


Figure 14: Example of the three occupancy grids from the different sensors as well as the final grid for a SCANer™ studio scene.

- The criterion of the traffic code: this criterion allows the overtaking by the left and the respect of the distance of security.

Regarding the control part of the steering angle and ego-vehicle speed, the planning algorithm sends the steering angle that corresponds to the chosen tentacle and an acceleration or deceleration setpoint that SCANer™ studio applies through a control block specific to the simulator.

*Test scenario* : In this scenario, we placed a static obstacle ahead the ego-vehicle (velocity of  $25 m/s$ ), and a dynamic obstacle with  $16.5 m/s$  velocity in the right lane (Fig. 15). The combination rule used in this scenario is the Cells number rule.



Figure 15: Scenario with a dynamic obstacle ( $16.5 m/s$ ) and a static obstacle. The ego-vehicle has an initial velocity of  $25 m/s$ . The red curve represents the reference trajectory.

The distance between the static and dynamic obstacles is chosen to include the safety distance with a safety margin that the ego-vehicle must respect during the folding maneuver. We consider that the reaction time of the red vehicle driver is  $\tau_{human} = 2 s$ , and  $a_{max} = 10 m/s^2$ . The safety distance to be respected is  $15.5m$ . The ego-vehicle is programmed to circulate at a speed of  $25m/s$ . In case no tentacle is navigable, the ego-vehicle will decelerate until it finds navigable tentacles and in this case it will accelerate until it regains its initial speed. Indeed, the proposed planning algorithm treats only the lateral trajectory, by considering the actual speed of the vehicle. The only modification of vehicle speed that the algorithm could propose, is to brake the vehicle when no navigable tentacle is detected. The speed is controlled in another module, to follow a given reference speed defined elsewhere. In our test scenarios, we implemented a basic speed controller and we defined a speed profile that respects stability conditions, to be able to test in closed-loop our planning algorithm.

Using binary grid in this scenario, the ego-vehicle decelerates and stops (Fig. 16). Due to the extending of dynamic

obstacle, the ego-vehicle don't find any navigable tentacle to change lane to the right.

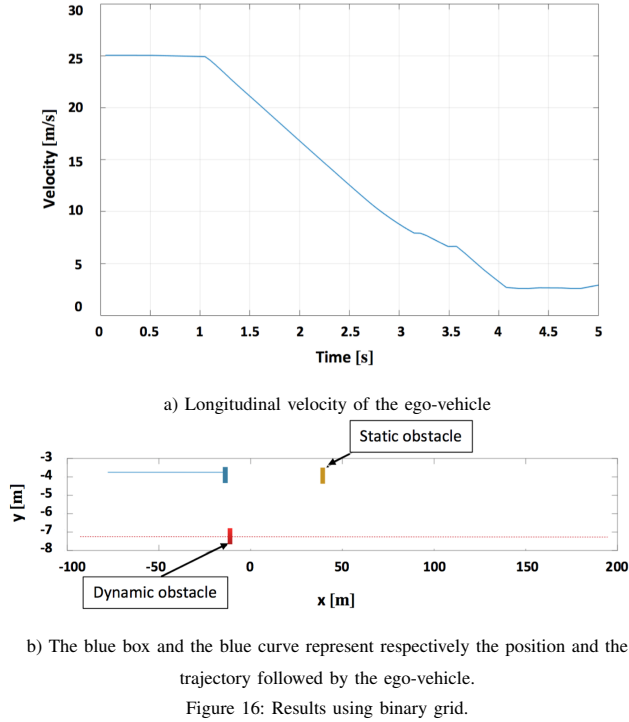


Figure 16: Results using binary grid.

Using evidential grids, the ego-vehicle manages to return to the reference trajectory but after a significant deceleration as shown in Figure 17. This deceleration is due to the mass values chosen for the road edges; we recall that with the rule of number of cells, a cell is considered occupied if  $m(O) > 0.5$ . So the tentacles that meet the edges of the road in a radius less than the safety distance are classified as not navigable.

We changed the mass on the road edges in the road grid:  $m_1(C_{ij}) = [m_1(\emptyset) m_1(F) m_1(O) m_1(\Omega)] = [0 \ 0 \ 0.5 \ 0.5]$  instead of  $m_1(C_{ij}) = [0 \ 0 \ 0.6 \ 0.4]$

Given that both cases give the same result with binary grid, the interest of this comparison is to study the influence of introduction of semantic rules in occupancy grid (consider the off-road space as occupied). By changing the masses of  $m_1$  from  $[0 \ 0 \ 0.6 \ 0.4]$  to  $[0 \ 0 \ 0.5 \ 0.5]$ , we reduce the occupation constraint of the road borders; instead of being occupied, it will be considered as uncertain space. The mass of  $[0 \ 0 \ 0.5 \ 0.5]$  allows to have more navigable tentacles, even those who attain the road borders. It should be noted that the choice of the best tentacle will take into account the mass on (uncertainty), to avoid the road borders if possible, and prioritize the lane.

In this case, the road edges are not considered occupied. Therefore, the planning algorithm did not classify the tentacles that reach the road edges within the security distance as not navigable tentacles, allowing the ego-vehicle to overtake the dynamic obstacle while avoiding the static obstacle.

Figure 18 shows the longitudinal speed and the lateral acceleration of the ego-vehicle.

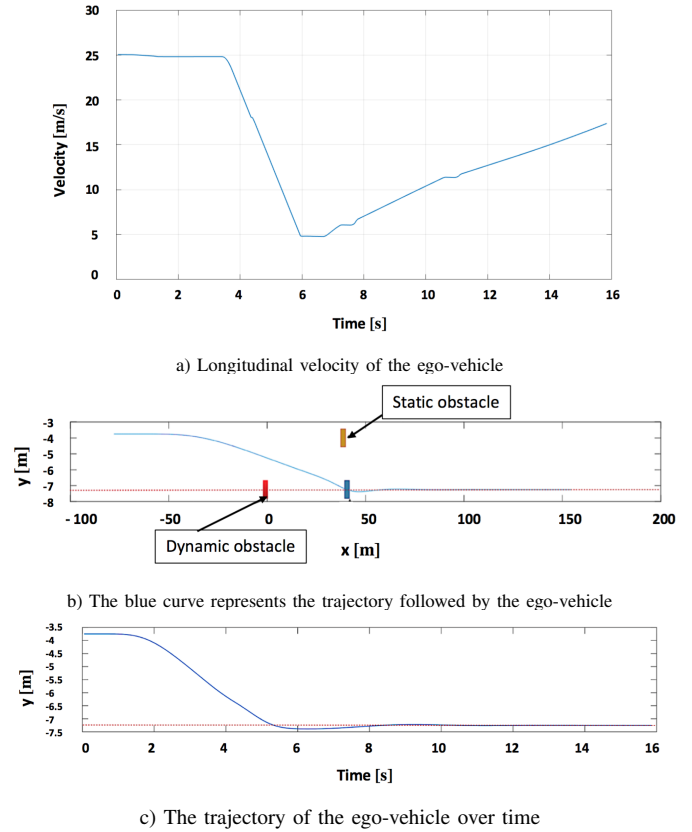


Figure 17: Results using evidential grid with cells number rule, and  $m_1(C_{ij}) = [0 \ 0 \ 0.6 \ 0.4]$

Figure 19.a represents the trajectory followed by the ego-vehicle as well as the position of the other vehicles at the end of overtaking and Figure 19.b represents the ego-vehicle's lateral position over time.

Note that the safety distance has been respected all along the overtaking maneuvers given that the safety distance is already integrated in the tentacle planning approach.

We can notice that the vehicle trajectory when  $m_1 = [0 \ 0 \ 0.6 \ 0.4]$  is closer to the boundary of the road compared to the case when  $m_1 = [0 \ 0 \ 0.5 \ 0.5]$ . What should be noticed in the comparison between both tests is the longitudinal speed. We see in Figure 17.a that the vehicle decelerates from 25 m/s to 5 m/s, when the mass  $m_1 = [0 \ 0 \ 0.6 \ 0.4]$ . That means that no navigable tentacle was found, that's why after overtaking, the vehicle overlaps laterally at a speed of 5 m/s. Despite the proximity to the border, the trajectory was safe because of the reduced speed. However, in the second case, with  $m_1 = [0 \ 0 \ 0.5 \ 0.5]$  the vehicle speed in Figure 18.a still approximately unchanged, what means that the planning algorithm arrives to identify navigable tentacle, almost on all the overtaking maneuver, due to the reduced constraint on the road border. Then, the vehicle still far from the border, due to its highest speed (25 m/s). Note that the planning algorithm calculates only feasible safe trajectories with respect to the vehicle speed, at any time.

The proposed planning method does not depend on the vehicle dynamics; it only considers some limit constraints with respect to these dynamic aspects. Hence, less amount

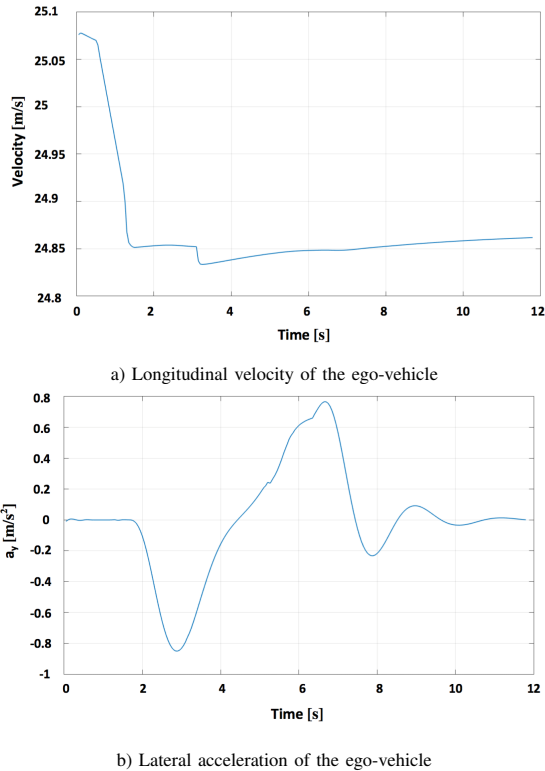


Figure 18: Results using evidential grid with different value mass for road edges, and  $m_1(C_{ij}) = [0 \ 0 \ 0.5 \ 0.5]$

of online computation is needed, compared to other model-based approaches like MPC. The proposed method is a reactive local trajectory planning method, suitable for the use of occupancy grid for environment representation. Note that the originality of the approach is to consider perception uncertainties through the use of evidential grids. The geometric form of the clothoids and the chosen MDP method used to select the best tentacle involve also relatively simple algorithms, suitable for real-time application.

However, some drawbacks could be mentioned, and will be addressed in future works. The Dempster and conjunctive rules, used to calculate the occupancy reward for the states using an evidential grid, need more computation time, compared to the cell-number rule and the binary grid approach. This issue will be checked in real-time tests. On the other hand, even that the form of clothoids is suitable for local maneuvers because it respects the dynamic constraints of the vehicle and the real form of roads on near horizon, however, the evaluation of the clothoid for middle and long distance is not so realistic. The vehicle should return parallel to the road direction for long horizons, instead of following the same chosen clothoid. This issue could be improved by coupling the clothoid method with a maneuver planner, an intermediate level between the local and the global planners.

## V. CONCLUSION AND PERSPECTIVES

In this work, the goal is to integrate uncertainty of the environment in the planning trajectory using evidential grid. Therefore, the potential use of the evidential grids was tested

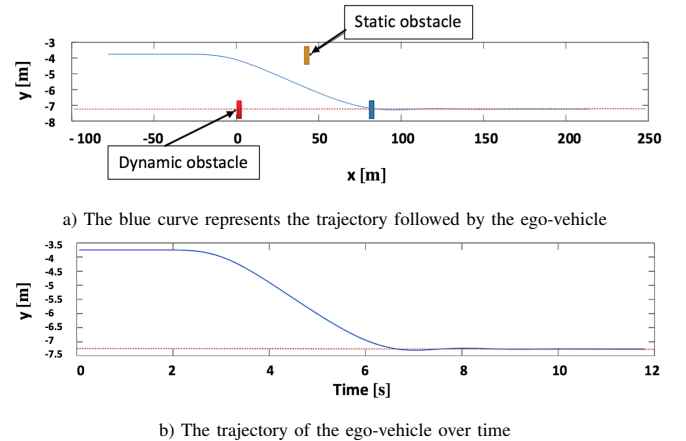


Figure 19: Results using evidential grid, and  $m_1(C_{ij}) = [0 \ 0 \ 0.5 \ 0.5]$

in the planning of trajectories with different combination rules. Real data was used and processed with *Matlab*. The simulation results show good performance of our algorithm in avoiding obstacles under uncertainty. However, the used combination rules require a significant computation time which makes its use in real time challenging. Otherwise, we worked under the assumption that each cell is a source of partial information of the overall state. In order to avoid working under this assumption, one of the perspectives is to compute masses on events as "more than  $n$  cells are occupied" where  $n$  is a threshold to be determined.

Another perspective would be to use evidential grids with semantic lane information [42]; Through the use of this kind of grid, we can have access to road edges and specially to traffic direction, which is important information that will allow the respect of traffic rules.

At this stage of our experiments we can already insist on the possibilities that the use of evidential grids allows for the planning of trajectories. We plan to continue the study with other scenarios and optimize the calculation time of the algorithm to test the other combination rules; and to test experimentally our approach on a robotized vehicle.

## ACKNOWLEDGE

This work was carried out within the framework of the Equipex ROBOTEX (Reference ANR-10-EQPX-44-01), which was funded by the French Government, through the program "Investments for the future" managed by the National Agency for Research.

## REFERENCES

- [1] M. Montemerlo, J. Becker, S. Bhat, D. Dolgov, H. Dahlkamp, S. Etinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, D. Langer, S. Klumpp, J. Levinson, A. Levandowski, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, and A. Petrovskaya. Junior : The Stanford Entry in the Urban Challenge. *Journal of Field Robotics*, 25(9), pp. 569-597, 2008.
- [2] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner. Three decades of driver assistance systems: Review and future perspectives. *IEEE Intelligent Transportation Systems Magazine*, 6(4), pp. 6-22, 2014.

- [3] C. Katrakazas, M. Qaddus, 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*, 2015.
- [4] S.M. LaValle. Rapidly-exploring random trees: A new tool for path planning. *Computer Science Department, Iowa State University*, 1998.
- [5] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J.P. How. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5), pp. 1105-1118, 2009.
- [6] 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*, 12(4), pp.566-580, 1996.
- [7] V. Delsart, T. Fraichard, and L. Martinez. Real-time trajectory generation for car-like vehicles navigating dynamic environments. *IEEE International Conference on Robotics and Automation*, pp. 3401-3406, 2009.
- [8] D. Ferguson, T. M. Howard, and M. Likhachev. Motion planning in urban environments. *Journal of Field Robotics*, vol. 25 pp. 939-960, 2008.
- [9] D. Madas, M. Nosratinia, M. Keshavarz, P. Sundstrom, R. Philippsen, A. Eidehall, and K. M. Dahlen. On path planning methods for automotive collision avoidance. *IEEE In Intelligent Vehicles Symposium (IV)*, pp. 931-937, 2013.
- [10] A. Chebly, G. Tagne, R. Talj, and A. Charara. A local trajectory planning and tracking for autonomous vehicle navigation using clothoid tentacles method. *International IEEE Conference on Intelligent Vehicles Symposium (IV)*, pp. 764-769, 2015.
- [11] Cassandra A. R. & Kaelbling L. P. Littman, M.L. Learning policies for partially observable environments: Scaling up. *Proceedings of the Twelfth International Conference on Machine Learning Machine Learning Proceedings 1995*, p. 362, 1995.
- [12] S. Brechtel, T. Gindele, and R. Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs. *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, 28, 2014.
- [13] K. Sabe, M. Fukuchi, J. S. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara. Obstacle avoidance and path planning for humanoid robots using stereo vision. *IEEE International Conference on Robotics and Automation (ICRA 04)*, Vol. 1, pp. 592-597, 2004.
- [14] J.A. Meyer and D. Filliat. Map-based navigation in mobile robots: Ii: a review of map-learning and path-planning strategies. *Cognitive Systems Research*, 4(4), pp.283-317, 2003.
- [15] S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2), pp. 111-127, 2003.
- [16] T. Reineking and J. Clemens. Dimensions of uncertainty in evidential grid maps. *In International Conference on Spatial Cognition*, pp. 283-298. Springer International Publishing, 2014.
- [17] G. Shafer. A mathematical theory of evidence. *Princeton University Press*, 1976.
- [18] P. Smets and R. Kennes. The transferable belief model. *Artificial Intelligence*, Vol. 66, pp. 191-234, 1994.
- [19] D. Pagac, E. M. Nebot, and H. Durrant-Whyte. An evidential approach to map-building for autonomous vehicles. *IEEE Transactions on Robotics and Automation*, 14(4), pp.623-629, 1998.
- [20] J. Mullane, M. D. Adams, and W. S. Wijesoma. Evidential versus bayesian estimation for radar map building. *IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1-8, 2006.
- [21] T. Yang and V. Aitken. Evidential mapping for mobile robots with range sensors. *IEEE Transactions on instrumentation and measurement*, 55(4), pp.1422-1429, 2006.
- [22] J. Moras, J. Dezert, and B. Pannetier. Grid occupancy estimation for environment perception based on belief functions and pcr6. *In SPIE Defense+ Security*, pp. 94740P-94740P. International Society for Optics and Photonics, 2015.
- [23] X. Li, X. Huang, J. Dezert, L. Duan, and M. Wang. A successful application of dsmt in sonar grid map building and comparison with dst-based approach. *International Journal of Innovative Computing, Information and Control*, 3(3), pp.539-549, 2007.
- [24] J. Carlson, R. R. Murphy, S. Christopher, and J. Casper. Conflict metric as a measure of sensing quality. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2032-2039, 2005.
- [25] M. Kurdej, J. Moras, V. Cherfaoui, and P. Bonnifait. Controlling remanence in evidential grids using geodata for dynamic scene perception. *International Journal of Approximate Reasoning*, 55(1), pp.355-375, 2014.
- [26] T. Ike, B. Grabe, F. Knigge, and M. Hoetter. Evidence based analysis of internal conflicts from inverse sensor models. *IEEE In Intelligent Vehicles Symposium*, pp. 1236-1240, 2009.
- [27] J. Clemens, T. Reineking, and T. Kluth. An evidential approach to slam, path planning, and active exploration. *International Journal of Approximate Reasoning*, Vol. 73, pp.1-26, 2016.
- [28] P. Sermanet, R. Hadsell, M. Scoffier, U. Muller, and Y. LeCun. Mapping and planning under uncertainty in mobile robots with long-range perception. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2525-2530, 2008.
- [29] M. Himmelsbach, T. Luettel, F. Hecker, V. Hundelshausen, and H.-J. Wuensche. Autonomous off-road navigation for mucar-3, improving the tentacles approach: Integral structures for sensing and motion. *Kunstl Intell*, 2011.
- [30] H. Mouhagir, R. Talj, V. Cherfaoui, F. Aioun, and F. Guillemard. A markov decision process-based approach for trajectory planning with clothoid tentacles. *IEEE International Conference on Intelligent Vehicles Symposium (IV)*, pp. 1254-1259, 2016.
- [31] H. Mouhagir, V. Cherfaoui, R. Talj, F. Aioun, and F. Guillemard. Using evidential occupancy grid for vehicle trajectory planning under uncertainty with tentacles. *IEEE International Conference on Intelligent Transportation Systems Conference (ITSC)*, pp. 2225-2231, 2017.
- [32] A.P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38, 1976.
- [33] P. Smets. Decision making in the tbm: the necessity of the pignistic transformation. *Int.ernational Journal of Approximate Reasoning*, 38(2), pp.133-147, 2005.
- [34] H. Mouhagir, R. Talj, V. Cherfaoui, F. Aioun, and F. Guillemard. Integrating safety distances with trajectory planning by modifying the occupancy grid for autonomous vehicle navigation. *IEEE International Conference on Intelligent Transportation Systems Conference (ITSC)*, pp. 1114-1119, 2016.
- [35] M. Althoff and R. Losch. Can automated road vehicles harmonize with traffic flow while guaranteeing a safe distance? *IEEE International Conference on Intelligent Transportation Systems Conference (ITSC)*, pp. 485-491, 2016.
- [36] D.V. McGehee, E.N. Mazzae, and G.S. Baldwin. Driver reaction time in crash avoidance research: Validation of a driving simulator study on a test track. *In Proceedings of the human factors and ergonomics society annual meeting*, 44(20), pp.3-320, 2000.
- [37] M. Goebel, M. Althoff, G. Buss, M. v Farber, F. Hecker, B. HeiBing, S. Kraus, R. Nagel, F.P. Leon, F. Rattei, and M. Russ. Design and capabilities of the munich cognitive automobile. *In Intelligent Vehicles Symposium*, pp. 1101-1107, 2008.
- [38] H. Mouhagir, R. Talj, V. Cherfaoui, F. Aioun, and F. Guillemard. Trajectory planning for autonomous vehicle in uncertain environment using evidential grid. *The 20th World Congress of the International Federation of Automatic Control*, pp. 13056-13061, 2017.
- [39] J. Moras, V. Cherfaoui, and P. Bonnifait. Moving objects detection by conflict analysis in evidential grids. *IEEE Intelligent Vehicles Symposium*, vol(6), 2011.
- [40] Ph. Xu, G. Dherbomez, E. Hery, A. Abidli, and Ph. Bonnifait. System architecture of a driverless electric car in the grand cooperative driving challenge. *IEEE Intelligent Transportation Systems Magazine (ITS Mag.)*, Vol. 10, Issue 1, pp 47-59, 2018.
- [41] A-A. Hadid. Creation of evidential grid for autonomous vehicles. *Internship report at Laboratoire Heudiasyc*, 2017.
- [42] C. Yu, V. Cherfaoui, and P. Bonnifait. Semantic evidential lane grids with prior maps for autonomous navigation. *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1875-1881, 2016.

ABOUT THE AUTHORS



**Hafida Mouhagir** graduated as a system engineer and received her M.S degree in computer science from the Ecole Nationale Supérieure d'Ingénieurs de Sud d'Alsace (ENSISA), France in 2014. She received her Ph.D. degree in automatic and robotics from the Université de Technologie de Compiègne (UTC), France, in 2017. Her research interests, in Heudiasyc UMR 7253, cover trajectory planning, perception and decision applied to autonomous vehicles.



**Reine Talj** is a CNRS researcher, since 2010, affected to heudiasyc laboratory at Université de Technologie de Compiègne (UTC). She obtained a PhD thesis in Physics at Université Paris-Sud, on the modelling and control of fuel cells, in L2S and LGEP, in 2009; after a Master 2 in Automatics and Signal processing at Université Paris-Sud and Supelec, in 2006, and an engineering diploma in electrical and electronic engineering from Lebanese University-Faculty of engineering, Lebanon in 2005. In 2009/2010, she was a teacher-researcher

at Université Paris-Sud. Her actual research activities concern autonomous vehicles, more precisely control and trajectory planning, as well as chassis stabilization, and control of Systems of Systems in the context of the project Labex MS2T.



**Véronique Cherfaoui** is Professor at the Université de Technologie de Compiègne, Sorbonne Universités (UTC). She received her Ph.D. degree in robotics and control of systems from the UTC in 1992. Her research interests in the Heudiasyc-CNRS laboratory are multi-sensor data fusion, distributed data fusion, data association and real-time perception systems for intelligent vehicles. She is member of the laboratory of excellence on Systems-of-Systems (Labex MS2T) and member of the joined laboratory Renault-UTC-CNRS SI-

VALAB.



**François Aioun** received the Eng. degree in 1988 in computer science, electronics and automatic control, the A.E. degree in 1989, and the Ph.D. degree in 1993 in automatic control and signal processing from the University of South Paris, Orsay, France. His main research topics were vibration systems control and noise control. For two years, he conducted postdoctoral research on turbo generator voltage control of nuclear power plant, at the University Laboratory of Research in Automated Production, ENS-Cachan, France.

Then, he worked for two years on modeling of power plant steam generator and turbo generator power control at the Electricity of France Research Center, Chatou, France. Since 1997, he has been working in the automotive control area of Peugeot Citroen Automotive Company. His main subjects of interest are control methods for powertrain actuators, engines, and connected autonomous vehicles.



**Franck Guillemard** was born in France in 1968. He received the Ph.D. degree in control engineering from the University of Lille, Lille, France, in 1996. He is currently with the Scientific Department of PSA Peugeot Citroen, where he is in charge of advanced research concerning computing science, electronics, photonics, and control. He is also an expert in the field of modeling, design, and control of mechatronic systems for automotives.