Obstacle Avoidance, Path Planning and Control for Autonomous Vehicles

Conference Paper · June 2019

DOI: 10.1109/IVS.2019.8814173

7 authors, including:

Hind Laghmara
Université de Haute-Alsace
8 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Thomas Laurain
Université de Haute-Alsace
22 PUBLICATIONS **71** CITATIONS
SEE PROFILE

Rodolfo Orjuela
Université de Haute-Alsace
61 PUBLICATIONS **516** CITATIONS
SEE PROFILE

Jean-Philippe Lauffenburger
Université de Haute-Alsace
75 PUBLICATIONS **298** CITATIONS
SEE PROFILE

Obstacle Avoidance, Path Planning and Control for Autonomous Vehicles

Hind Laghmara, Mohamed-Taha Boudali, Thomas Laurain, Jonathan Ledy, Rodolfo Orjuela,
Jean-Philippe Lauffenburger and Michel Basset.

***Abstract*— Obstacle avoidance requires three main levels in autonomous vehicles, namely, perception, path planning and guidance control. In this paper, a global architecture is proposed by taking into account the link between the three levels. On the environment perception level, an evidential occupancy-grid-based approach is used for dynamic obstacle detection. The poses of objects are therefore considered for trajectory generation. The latter is based on a smooth trajectory sigmoid function. Finally, the control guidance employs this obstacle avoidance trajectory to generate the appropriate steering angle. The whole strategy is validated on our experimental test car. The experimental results show the effectiveness of the proposed approach.**

I. INTRODUCTION

Research on Intelligent Transportation Systems raises a particular interest as it addresses challenging issues of autonomy and safety in complex environments. The main and mandatory steps for the conception of an autonomous vehicle are *Perception*, *Planning* and *Control*.

In fact, perception is composed of *Environment modeling* and *Localization*. They respectively rely on exteroceptive and proprioceptive sensors. Next, planning aims to generate an optimal trajectory based on the information delivered by the perception results in order to reach a given destination. Finally, the control module is dedicated to following the generated trajectory by commanding the vehicle's actuators.

Each module of the process will be covered in this paper for the specific case of obstacle avoidance. The integration of these tasks in a global architecture is the main contribution of this paper. The perception module insures a description of the environment according to an accurate grid representation. The use of Occupancy Grid Maps (OGM) is particularly convenient for obstacle avoidance because it allows to identify the navigable space and to locate static and dynamic objects in the scene. The poses of the objects to be avoided are then used at the path planning level which generates a trajectory and a speed profile according to a parameterized sigmoid function and a rolling horizon presented in [1]. The obtained curvature profile is considered as a reference path for the guidance control module. This level provides the appropriate steering angle to the vehicle according to a lateral guidance controller using the Center of percussion (CoP) instead of the classic center of gravity. The proposed controller is based on a feed-forward and a robust state-feedback actions to respectively reduce the impact of the disturbance on the lateral error and to guarantee lateral stability [2].

The paper is organized as follows: Section II presents the global approach containing the different modules to be implemented for the obstacle avoidance. Section III presents the evidential grid-based approach for dynamic object detection. Section IV illustrates the obstacle avoidance algorithm which is based on the combination of a parametrized sigmoid function and a rolling horizon. The design of the controller based on a feed-forward coupled to a robust state-feedback is detailed in Section V. The experimental platform and results of the global approach are illustrated in Section VI. Finally, Section VII concludes the paper.

II. OBSTACLE AVOIDANCE STRATEGY

This section presents the headlines for the global obstacle avoidance strategy based on three modules as shown in Figure 1. Each level is shortly described in this section.

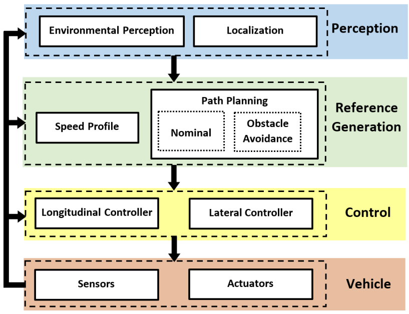

Fig. 1. Obstacle avoidance strategy

A. Perception Module

Perceiving the environment correctly and efficiently is mandatory for an autonomous vehicle. This study mainly focuses on the environmental perception to extract locations of static/dynamic objects as well as the drivable paths based on extereoceptive sensors. The localization part is not treated as the position of the vehicle is considered to be known and reliable. One of the most used approaches for both extracting an information of the road as well as surrounding objects is the Occupancy Grid (OG). It can be used for several applications as collision avoidance, sensor fusion, object tracking and simultaneous localization and mapping (SLAM) [3]. The basic idea of the OG is to represent a map of the environment as an evenly spaced field of binary random variables each representing the presence of an obstacle at that location in the environment [4]. It can be generated according to many formalisms to handle the noisy and uncertain sensor

IRIMAS EA7499 Université de Haute-Alsace, Mulhouse, France
firstname.lastname@uha.fr

measurement data with the assumption that the vehicle pose is known. In this paper, the OG is defined using Belief Theory introduced by Dempster and Shafer [5], [6] as it models uncertainty, imprecision and ignorance. It also allows a management of conflict in data fusion. More details are given in Section III.

B. Reference Generation Module

This module is dedicated to the definition of the trajectory and the corresponding speed profile to be followed by the vehicle. The planner receives the drivable areas and the obstacle positions from the perception module. From these information, a geometric trajectory as well as the speed profile can be generated. This paper focuses on the path planning strategy. This part aims to provide a nominal trajectory from the starting point to the final point based on the perceived drivable zones.

When detecting an obstacle, a second trajectory (obstacle avoidance trajectory) is calculated to ensure the safety and the comfort of the autonomous vehicle passengers, and to join the nominal trajectory after the avoidance. This avoidance trajectory can be obtained through local planification since it only concerns a small portion of the nominal trajectory. In order to reduce the computational cost of the trajectory generation algorithm, the rolling horizon method is applied, as in [1] whose work is extended in the present paper, see Section IV. Then, these trajectories (nominal and obstacle avoidance) can be considered as references for the control module, mainly the lateral controller.

C. Control Module

The control module is composed of two main parts: longitudinal and lateral controllers, ensuring the automated driving guidance. The main focus is given here on the lateral controller to deal with the obstacle avoidance. In fact, an appropriate steering angle δ_f is provided by the lateral controller in order to follow the desired path given by the reference generation module. The tracking of the desired path can be achieved by reducing two tracking errors, namely, the lateral error and the orientation error. Among the geometric and dynamic lateral guidance strategies existing in the literature [7], the dynamic approach based on the Center of Percussion (CoP) is employed here [8]. This choice relies on the performance of this control method. The CoP is a geometrical point located in front of the Center of Gravity (CoG) of the vehicle allowing to anticipate the lateral position error. A better trajectory tracking can then be expected. On the other hand, since the motion of the CoP is decoupled from the rear tire lateral forces [9], the lateral dynamic equations become less complex as will be shown in Section V.

III. GRID-BASED DYNAMIC OBSTACLE DETECTION

An OG is a representation which employs a multidimensional tesselation of space into cells where each one stores a knowledge of its state of occupancy [4]. Today, there is a large use of OGs due to the availability of much more powerful resources to handle their computational complexity and greediness. The construction of a grid has been applied in multiple dimensions (2D, 2.5D and 3D) [10] where each cell-state is described according to a chosen formalism. The most common one is the Bayesian framework which was adopted first by Elfes [4] followed by many extensions as the well-known Bayesian Occupancy Filter (BOF) [11]. Other works suggested a formalism based on Dempster-Shafer Theory which is also known as Evidential Theory and described thereafter.

A. Using Belief Theory

Generalizing the probability theory, Belief Theory offers an adequate representation of the data and source imperfections and thus is appropriate for perception in ITS. It offers a wide range of fusion operators handling these properties according to the application. Some of the studies using the Belief frame for constructing OGs can be found in [12], [13]. This work derives from the study of [13] which suggests an approach based on the resulting conflict for mobile object detection and drivable space determination. For that, a frame of discernment is defined to include the states of a cell considering it to be Free (F) or Occupied (O). The discernment frame is then $\Omega = \{F, O\}$. The referential power set frame contains all possible combinations of the hypotheses defined as: $2^\Omega = \{\emptyset, F, O, \{F, O\}\}$. To express the belief in each state, a mass function $m(.)$ is defined to respectively express conflict $m(\emptyset)$, Free state $m(F)$, Occupied state $m(O)$ and the unknown state $m(\{F, O\})$.

B. Sensor model

Basically, a *sensor model* is how the *mass function* of a state according to a measure is calculated. In our application, the sensor to be used is a 3D multi-echo LIDAR (see Section VI). The input data will include ranges r_i and an angle θ_i according to a point p_i. According to this set of data, a Scan Grid (SG) in polar coordinates is constructed. Each row of SG corresponds to an angular sector $\Theta = [\theta^-, \theta^+]$ for which a cell is defined in $R \times \Theta$. The range of a cell is $R = [r^-, r^+]$ which means that each cell is defined by a pair on which a mass is attributed as $m\{\Theta, R\}$. The masses corresponding to each proposition $A \in \Omega$ are found hereby [13]:

$$m\{\Theta, R\}(\emptyset) = 0 \tag{1}$$

$$m\{\Theta, R\}(F) = \begin{cases} 1 - \mu_F & r_i \in R \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$m\{\Theta, R\}(F) = \begin{cases} 1 - \mu_O & \text{if } r^+ < \min(r_i) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$m\{\Theta, R\}(O) = \begin{cases} \mu_F & r_i \in R \\ \mu_O & \text{if } r^+ < \min(r_i) \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

where μ_F and μ_O respectively correspond to the probability of false alarm and missed-detection of the sensor. For simplicity reason, these mass functions will be noted $m(\emptyset)$, $m(O)$, $m(F)$ and $m(\Theta)$.

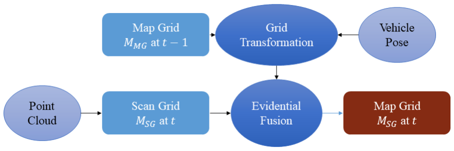

Fig. 2. Map Grid Construction

Figure 2 illustrates the process of building and updating the MG using the sensor point cloud provided at a time t. This update is done according to multi-grid evidential fusion. This is the most interesting part of the process as it allows the temporal update of the map grid and also the evaluation of the state of cells. Among the various operators in Belief Theory, the Dempster-Shafer rule of combination is used:

$$m_{MG,t} = m_{MG,t-1} \oplus m_{SG,t} \tag{5}$$

where $m_{MG,t}$ and $m_{SG,t}$ are resp. the mass function of the Map grid and Scan Grid at time t. The operator is defined as:

$$(m_1 \oplus m_2)(A) = K \sum_{\forall B,C \in 2^\Theta, B \cap C = A, A \neq \emptyset} m_1(B).m_2(C). \tag{6}$$

where

$$K^{-1} = 1 - \sum_{\forall B,C \in 2^\Theta, B \cap C = \emptyset} m_1(B).m_2(C) \tag{7}$$

The resulting masses $m_{MG,t}(A)$ define the state of each cell which depends on the previous state and the new measure. The resulting masses according to each state are found as follows [13]:

$$\begin{aligned} m_{MG,t}(O) =& m_{SG,t}(O).m_{MG,t-1}(O) + m_{SG,t}(\Omega). \\ & m_{MG,t-1}(O) + m_{SG,t}(O).m_{MG,t-1}(\Omega) \\ m_{MG,t}(F) =& m_{SG,t}(F).m_{MG,t-1}(F) + m_{SG,t}(\Omega). \\ & m_{MG,t-1}(F) + m_{SG,t}(\Omega).m_{MG,t-1}(F) \\ m_{MG,t}(\Omega) =& m_{SG,t}(\Omega).m_{MG,t-1}(\Omega) \\ m_{MG,t}(\emptyset) =& m_{SG,t}(O).m_{MG,t-1}(F) + m_{SG,t}(F). \\ & m_{MG,t-1}(O) \end{aligned} \tag{8}$$

with $m_{MG,t}(\emptyset)$ being the combined mass expressing conflict. Basically, this property shows the discordance between the knowledge expressed at $t-1$ and t. The reason for which a conflict appears is that when a cell changes its state from F to O or vice-versa. Therefore, the detection of this conflict can lead to the evaluation of the dynamic cells. The conflict allows to label the occupied cells which change their state according to two types of conflict:

$$\begin{aligned} C_1 =& m_{SG,t}(O).m_{MG,t-1}(F) \quad \text{from } F \text{ to } O \\ C_2 =& m_{SG,t}(F).m_{MG,t-1}(O) \quad \text{from } O \text{ to } F \end{aligned} \tag{9}$$

where $m_{MG,t}(\emptyset) = C_1 + C_2$.

The fusion process normalizes the state masses by the total conflict but this information is considered for labeling the mobile cells which define a dynamic object. The poses of each detection are then used as an input for trajectory generation in the next section.

IV. TRAJECTORY GENERATION

This section is dedicated to the path planning, i.e. creating a geometric trajectory (following coordinated points $A_i(x_i, y_i)$). Since this paper aims to validate the feasibility of the proposed avoidance architecture, the speed profile as well as the associated longitudinal control are not considered. As mentioned in Section II, the path planning module has two objectives: generating a global nominal trajectory according to the starting and arriving points and generating a local trajectory to avoid a detected obstacle. Here, the focus is given in the avoidance trajectory generation. This avoidance trajectory must respect safety criteria particularly regarding the longitudinal and lateral distances with the obstacle. These distances can be equal, creating a circular safety zone around the obstacle as recently proposed in [1]. This paper proposes a generalization of this method by considering the global case in which the lateral and longitudinal safety criteria are different.

In order to get the geometrical form of the trajectory, several mathematical approaches exist, based on functions such as clothoïd, Bézier or spline curves [14], [15]. An extensive review of all these geometrical methods is given in [16]. These methods have interesting characteristics (degree of smoothness, choice of the best trajectory among a collection of candidates...) but their computational cost can be high. Among them, sigmoïd functions represent a fair trade-off between smoothness and computational cost. The considered approach proposes to combine this mathematical method with a local Horizon planner in order to reduce the computational cost. The advantages of this planning method is largely discussed in [1]. This local planner considers information of the detected obstacle from the occupancy grids to define the appropriate smooth avoidance maneuver and returns to the nominal trajectory.

A. Geometrical Avoidance

Figure 3 presents the different trajectories: the nominal one, the safety zone and the final smooth avoidance trajectory. The safety zone definition R is the first step after the detection of an obstacle. This area is not navigable to avoid collision due to the proximity with the obstacle. L_x and L_y, respectively the semi-major and semi-minor axes of the ellipse, are the safety criteria to define the area. Once it is defined, the avoidance trajectory can be designed. In order to ensure the comfort of the passengers, a sigmoïd-based function is chosen. In Figure 3, A is referring to the starting point (i.e. center of gravity of the ego-vehicle), B is the flexion point of the sigmoïd and WP is the way-point to be reached. The degree of smoothness C (verifying $C \in [0;1]$) can be tuned such that the avoidance trajectory can be defined as:

$$Y_m(n) = \frac{L_y}{1 + e^{C(L_y - L_x)}} \tag{10}$$

In order to obtain an algorithm which is robust when the obstacle is moving, the whole procedure (determining the safety zone and calculating the way-point of the sigmoïd-based function) is repeated for every sample of the horizon vector.

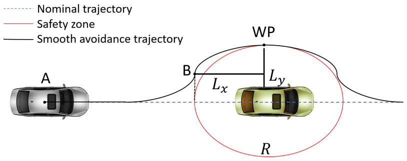

Fig. 3. Trajectory Planning

B. Horizon planning

In order to reduce the computational cost of the algorithm, a local horizon planner is used. Instead of following the whole avoidance trajectory, this one is divided into several segments. The local trajectory is computed at every sample of this discrete horizon, reducing the computational cost and making the algorithm robust to obstacle dynamics. Two parameters can be parametrized: the sample size and the horizon length. This last one depends on the equipped perception sensors (hardware constraints) and on the vehicle speed (rolling horizon). The sample step represents the subdivision of the trajectory in local segments. The whole principle is summarized in Figure 4.

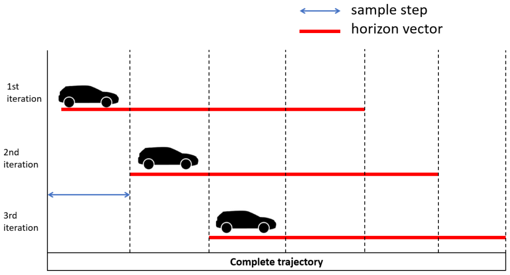

Fig. 4. Horizon Planning [1]

When the vehicle reaches the next sample step, the local horizon is computed again. As one can see, there is a common part between two iterations, allowing the algorithm to deal with dynamic obstacles. As in discrete-time domain, the choice of the sample size involves a trade-off between accuracy and computational cost. The algorithm iterates until the horizon vector reaches the end of the complete trajectory, i.e. when the perception sensors have covered all the subdivisions of the trajectory. This geometrical trajectory is the input for the guidance control level.

V. GUIDANCE CONTROL DESIGN

This section presents the lateral guidance controller design used in the control module illustrated in Figure 1. The lateral guidance aims to reduce both errors, the lateral error e_y between the CoG of the vehicle and the reference trajectory, and the orientation error e_ψ between the longitudinal axis of the vehicle and the reference trajectory as shown in Figure 5:

$$\dot{e}_y = v_y + v_x e_\psi \qquad e_\psi = \psi - \psi_{ref} \tag{11}$$

where v_x and v_y are the longitudinal and the lateral velocities at the CoG respectively, ψ the yaw angle and ψ_{ref} the desired yaw angle.

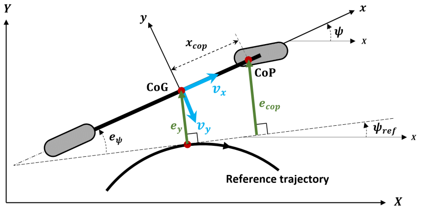

Fig. 5. Lateral and orientation errors using CoP

Here, instead of the classic CoG lateral error e_y, it is proposed to use the lateral error at the CoP defined as [9]:

$$e_{cop} = e_y + x_{cop} e_\psi \tag{12}$$

with x_{cop} the distance between the CoP and the CoG depending only on the vehicle configuration:

$$x_{cop} = I_z/(mL_f) \tag{13}$$

where m and I_z are the vehicle mass and the yaw inertia, and L_f the distance between the CoG and the front axle. It can be noticed from (12) that the CoP lateral error e_{cop} is higher than the lateral error e_y as shown in Figure 5. In this way, the lateral position error is anticipated and it can be expected a better trajectory tracking. Here the center of percussion (CoP) is employed as geometrical point on the vehicle in contrast to the classic employed controllers based on the CoG (center of gravity). The main advantage of the CoP is the complexity reduction of the lateral dynamics equations since the rear tire lateral force does not influence the motion of the CoP [9].

Based on the planar bicycle model [17], and using the tracking errors (11) and (12), the tracking error model used to design the CoP lateral guidance controller is:

$$\dot{\xi}(t) \;=\; A_c\xi(t) + B_c\delta_f(t) + D_c w_{ref}(t) \tag{14}$$

with $\xi = \left[\; e_{cop}, \dot{e}_{cop}, e_\psi, \dot{e}_\psi \;\right]^T$ the state vector error, δ_f the front steering angle and $w_{ref} = \left[\dot{\psi}_{ref}, \ddot{\psi}_{ref}\right]^T$ the vector containing the desired yaw rate and the desired yaw acceleration considered here as a disturbance. The matrices

A_c, B_c and D_c are such that:

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-2R_lC_f}{mv_x} & \frac{2R_lC_f}{m} & \frac{2R_lC_f(x_{cop}-L_f)}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-b}{I_zv_x} & \frac{b}{I_z} & \frac{bx_{cop}-c}{I_zv_x} \end{bmatrix},$$

$$B_c = \begin{bmatrix} 0 \\ \frac{2R_lC_f}{m} \\ 0 \\ \frac{2C_fL_f}{I_z} \end{bmatrix}, D_c = \begin{bmatrix} 0 & 0 \\ \frac{-2R_lC_fL_f}{mv_x} - v_x & -x_{cop} \\ 0 & 0 \\ \frac{-c}{Izv_x} & -1 \end{bmatrix}$$

where L_r is the distance between the CoG and the rear axle, $a = 2(C_f + C_r)$, $b = 2(C_fL_f - C_rL_r)$, $c = 2\left(C_fL_f^2 + C_rL_r^2\right)$, $R_l = (L_f + L_r)/L_r$. C_f and C_r are the front and the rear corning stiffnesses. Notice that the rear corning stiffness C_r is not employed in the second line of A_c and consequently the use of the CoP allows to reduce the number of uncertain parameters.

The lateral controller calculates an appropriate δ_f in order to ensure the convergence of the state vector error ξ towards zero. Furthermore, as the tracking error model dynamics is influenced by the vector w_{ref}, the controller must also ensure an attenuation level of its impact. To achieve these goals, a lateral controller consisting in a feed-forward coupled to a robust state feedback is proposed [2]:

$$\delta_f(t) = L_{FF}w_{ref}(t) - K_{FB}\xi(t) \qquad (16)$$

with L_{FF} and K_{FB} the feed-forward and the robust feedback gains respectively. *The feed-forward action* aims to partially eliminate the impact of the vector w_{ref} on the lateral error dynamics $\dot{e}_{cop}$. The benefits of CoP is that the obtained feed-forward does not require the knowledge of the rear corning stiffness C_r. *The state feedback action* guaranties the exponential convergence towards zero of the error vector ξ as well as attenuate the impact of the vector w_{ref} on the state variables. This robust control problem can be formulated using the linear matrix inequality (LMI) as shown in [2].

VI. EXPERIMENTS

A. Experimental Setup

The experimental platform ARTEMIPS is an autonomous test car equipped with several sensors : a high precision Oxford IMU (Inertial Measurement Unit) RT-3002 with DGPS technology, 2 IBEO LUX 2D 4 layers laser scanners, 2 VLP-16 Velodyne 3D laser scanners and a high range camera MANTA-G125 (cf. Figure 6). The RT-3002 is used as the reference sensor for position, velocity, acceleration and orientation measurements. LUX scanners are used to provide long range detection (in the form of 4 layers points clouds) in the front and in the rear of the car. The VLP-16 are used to complete the detection of the environment on both side of the car (they provide 16 layers point clouds with a 360^o surround view). ARTEMIPS is also equipped with 3 actuators with 2 integrated servo motors MAC-141 to control the steering wheel and the brake pedal, as well as a multifunctions NI-daq system to pilot the engine of the car. All sensors and actuators are linked together to an embedded computer that runs RTMaps software solution from Intempora. It is a platform dedicated to multi-sensors and multi-actuators systems. All developments in the car have been made with this software (from guidance to perception).

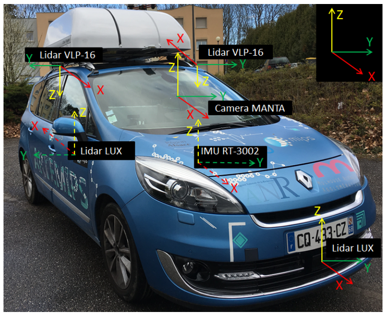

Fig. 6. The experimental platform ARTEMIPS and its reference frames.

B. Experimental Results

For readability purposes, the performance of the proposed architecture is evaluated through only one experimental scenario considering an obstacle avoidance situation. This test is performed at a constant speed $v_x = 10\,km/h$.

According to the recorded dataset from the four laser scanners, a point cloud is used for the construction of the OG according to the approach described in Section III. The surrounding scene and the object to be avoided are displayed in Figure 7. The temporal fusion of the OGs highlights the conflict which describes the dynamic cells. A Hierarchical Clustering algorithm is applied (From the Statistics and Machine Learning Toolbox in MATLAB) for the construction of the dynamic objects. They are represented in Figure 8 by the 3D bounding boxes. The displayed coordinates correspond to the pose of the vehicle based on the GPS data. The object to be avoided is the red one. It can be noticed that some false detections can be found and are due to the sensitivity of the approach to the positioning errors.

The experimental results are shown in Figure 9. It can be seen on the top left figure that the nominal trajectory intersects with the object's position whereas the generated path in red avoids the obstacle. It can also be observed that the lateral controller ensures a good trajectory tracking and allows to avoid the detected obstacle between $13\,s$ and $20\,s$. In this time interval, the controller generates a steering angle changing from positive values to negative values to avoid the obstacle and ensures small tracking errors ($|e_y| < 30\,cm$ and $|e_\psi| < 5\,\text{deg}$).

VII. CONCLUSION

This paper has presented a dynamic obstacle avoidance scheme based on three levels of perception, path planning and control guidance. The dynamic obstacle detection was done according to an evidential occupancy grid. The path planning generates a smooth trajectory based on a sigmoïd function in order to avoid the detected obstacle. Finally, the

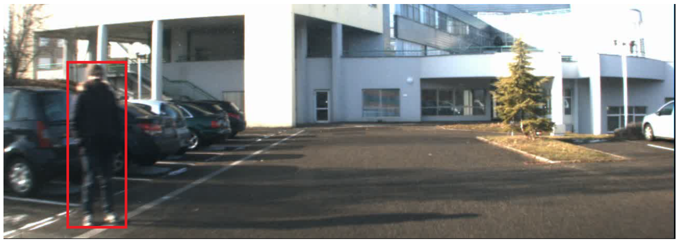

Fig. 7. Sequence used for the obstacle avoidance test

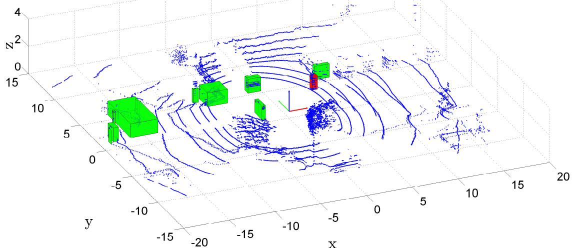

Fig. 8. Point cloud, origin coordinates and obstacle detection

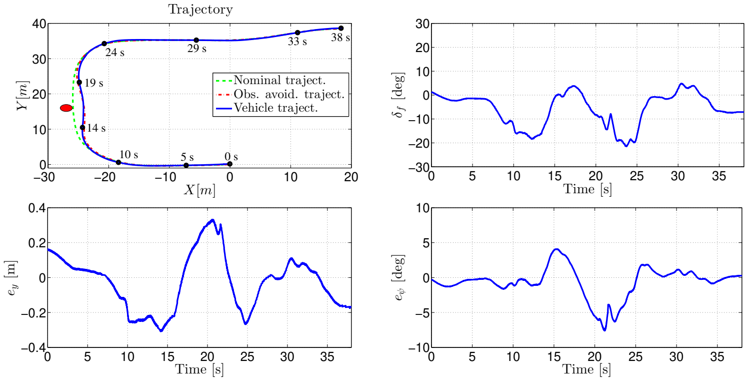

Fig. 9. Steering Controller Results

generated reference trajectory is followed by the vehicle through a lateral control based strategy at the Center of Percussion. Experimental results on our test vehicle show the adaptability of the proposed approach for an effective obstacle avoidance. Future work will include a consideration of a positioning strategy as well as an assessment of the approach in more complex scenarios.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support from Fondation Wallach (Mulhouse) in the context of the Project SIMPHA.

REFERENCES

[1] W. Ben Messaoud, M. Basset, J.-P. Lauffenburger, and R. Orjuela, "Smooth Obstacle Avoidance Path Planning for Autonomous Vehicles," in *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Madrid, Spain, 2018.

[2] M. Boudali, R. Orjuela, and M. Basset, "A comparison of two guidance strategies for autonomous vehicles," in *20th IFAC World Congress*, Toulouse, France, 2017.

[3] S. Wirges, C. Stiller, and F. Hartenbach, "Evidential occupancy grid map augmentation using deep learning," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

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

[5] A. P. Dempster, "A generalization of bayesian inference," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 30, no. 2, pp. 205–247, 1968.

[6] G. Shafer, *A mathematical theory of evidence*. Princeton, NJ, USA: Princeton University Press, 1976.

[7] D. Watzenig and M. Horn, *Automated Driving - Safer and More Efficient Future Driving*. Springer, 2017.

[8] S. C. Peters, E. Frazzoli, and K. Iagnemma, "Differential flatness of a front-steered vehicle with tire force control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, 2011.

[9] N. R. Kapania and J. C. Gerdes, "Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling," *Vehicle System Dynamics*, vol. 53, no. 12, pp. 1687–1704, 2015.

[10] A. Asvadi, P. Peixoto, and U. Nunes, "Detection and tracking of moving objects using 2.5d motion grids," in *18th International Conference on Intelligent Transportation Systems*, Las Palmas, Spain, 2015.

[11] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessiere, "Bayesian Occupancy Filtering for Multitarget Tracking: an Automotive Application," *International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, Jan. 2006.

[12] D. Pagac, E. M. Nebot, and H. F. Durrant-Whyte, "An evidential approach to map-building for autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 623–629, 1998.

[13] J. Moras, V. Berge-Cherfaoui, and P. Bonnifait, "Moving objects detection by conflict analysis in evidential grids," in *IEEE Intelligent Vehicles Symposium (IV)*, Baden-Daden, Germany, 2011, pp. 1122–1127.

[14] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *J. Field Robot.*, vol. 25, no. 9, pp. 640–673, Sep. 2008.

[15] J. Daniel, A. Birouche, J.-P. Lauffenburger, and M. Basset, "Navigation-based constrained trajectory generation for advanced driver assistance systems," *International Journal of Vehicle Autonomous Systems*, vol. 9, no. 3, 2011.

[16] D. Gonzalez, J. Perez, V. Milanes, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, April 2016.

[17] R. Rajamani, *Vehicle Dynamics and Control*, ser. Mechanical Engineering Series. Springer US, 2012.