2015 IEEE Intelligent Vehicles Symposium (IV)
June 28 - July 1, 2015. COEX, Seoul, Korea

Local Trajectory Planning and Tracking of Autonomous Vehicles, Using Clothoid Tentacles Method

Chebly Alia, Tagne Gilles, Talj Reine and Charara Ali

***Abstract*— In general, autonomous navigation requires three key steps, the perception of the environment surrounding the vehicle, the trajectory planning and the actuators control. Numerous works on the localization, perception, generation of occupancy grids and control of vehicles were developed within the ASER team at Heudiasyc laboratory. The work presented in this paper covers, essentially, trajectory planning and is based on the results of these works. The challenge is to avoid static and dynamic obstacles at high speed, using real time algorithms. The planning method developed in this work uses an empirical approach for local path planning. This approach consists on drawing clothoid tentacles in the ego-centered reference frame related to the vehicle. An occupancy grid represents the environment surrounding the vehicle and is considered to be ego-centered around it. Using the information of the occupancy grid, each tentacle is classified as navigable or not navigable. Among the navigable tentacles, only one tentacle is chosen as the vehicle reference trajectory using several criteria. The chosen tentacle is then applied to the vehicle using a lateral controller based on Immersion and Invariance principle (*I*&*I*).**

I. INTRODUCTION

Autonomous driving technology is a field of research that aims to achieve the optimum in automotive safety and comfort. This field has received considerable interest in the academic, industrial and military domains. In fact, the development of autonomous vehicles requires the use of perception, path planning and control technologies. The work presented in this paper focuses on trajectory planning for autonomous vehicles navigation. The purpose of planning is to compute a local trajectory based on several criteria in order to avoid obstacles while following a global trajectory defined by a GPS or a global map. To achieve this goal, we first collect the information acquired from the perception of the vehicle environment, then, by applying our algorithm, we generate a path that seems to be the best path to follow and finally the vehicle is controlled in order to follow the chosen path. The controller used in this work is based on the notions of system immersion and manifold invariance (*I*&*I*) and is presented later in the paper. Most approaches caring about obstacle avoidance are local, they do not seek to model the environment as a whole but rather they use sensor measurements to deduce secure orders [7]. We present below some local planning methods developed in the literature.

The Wall Following Algorithm presented in [16], aims to move a robot in a desired direction throughout a wall or parallel to it. This method requires only a single distance sensor and is often used in indoor applications.

Another algorithm, presented in [8], is the Potential Fields Method. This method defines a function that affects large potential to obstacles and low potential to objectives points of the trajectory. Thus, the problem is reduced to an optimization problem that aims to find the commands causing the robot to a global minimum of the function. There are many extensions to this method such as the virtual force fields [3], the field vector histograms [4], VFH + extensions [13] and VFH [14]. Unfortunately, the implementation of these methods requires a lot of calculation which may be constraining in real-time navigation applications. In addition, the function may converge to a local minimum instead of the global minimum.

The technique of SLAM (Simultaneous Localization And Mapping) considers an autonomous vehicle, placed at an unknown location in an unknown environment [11] and asks if it is possible to simultaneously build an incrementally map of environment while locating the vehicle in a fixed cartesian coordinate system related to the earth. Theoretically, SLAM can now be considered a solved problem. However, important issues remain realizing more general SLAM solutions [6].

The Tentacles method imitates the behavior of an insect that uses its antennae to detect and avoid obstacles. Indeed, the basic approach is to use a set of virtual antennas that we call tentacles and an egocentric occupancy grid around the vehicle [15]. The occupancy grid expresses the state of the environment surrounding the vehicle and determines obstacles positions, in case of their existence. The tentacle is a geometric shape that models the likely trajectory of the vehicle and we can find in the literature multiple shapes of tentacles.

In [15], the shape adopted for tentacles is circular. The weakness of this approach appears in considering all the tentacles generated for a certain speed as trajectory candidates even if their curvature is not well-suited to the current vehicle steering angle. Moreover, the width of each tentacle used to decide whether the tentacle is free of obstacles or not is much larger than necessary, almost twice the width of the vehicle [8], to take into account the transient phase needed to converge to the trajectory, given that the initial dynamical state of the vehicle is not considered in the tentacles shape.

reasoning why circular is bad

In [5], [7], the tentacles used allow the vehicle to go to the left or the right side with different offset shifts, while remaining substantially parallel to the base path. This form of tentacle is well-suited for high speed but its major drawback is its dependence on the base path.

Heudiasyc, CNRS, UMR 7253, Université de Technologie de Compiègne, 60200 Compiègne, France alia.chebly@hds.utc.fr gilles.tagne@hds.utc.fr reine.talj@hds.utc.fr ali.charara@hds.utc.fr

In [8], the form of clothoids was adopted for off-road autonomous navigation. This kind of tentacle considers the current steering angle of the vehicle so the tentacle width used to decide whether this tentacle is free or not could be slightly higher than the vehicle width.
As the tentacles method is assumed to be a fast reactive method and as the form of clothoid makes the approach more realistic, the tentacles method with clothoid forms was adopted in our work. The algorithms were done in MATLAB and then validated using data acquired from a vehicle navigation simulator (Scaner Studio). We present in Section II our navigation strategy. Section III reports some results, on both simulated and real data, while Section IV concludes the paper.

II. PRESENTATION OF THE NAVIGATION STRATEGY

Our path planner algorithm must generate a trajectory able to avoid obstacles and push the vehicle toward the reference trajectory. The first step is the tentacles generation. Then, to guarantee a secure navigation, we generate around each tentacle a classification zone defined later in the paper. In parallel, using sensors data, we construct an occupancy grid that clearly shows the environment state around the vehicle. The superposition of the classification area with the occupancy grid allows us to determine which tentacles can be classified as a secure path. These tentacles are called navigable tentacles. If we find several navigable tentacles, we have to choose the best tentacle using certain criteria defined afterwards. The best tentacle is then considered as a trajectory to execute. In order to execute the selected trajectory, a path tracking controller derives the steering angle commands from the geometric parameters of the selected tentacle. If we can't find any navigable tentacle, we search the tentacle having the greatest distance to the first obstacle and we proceed to brake the vehicle with a constant deceleration along this tentacle.

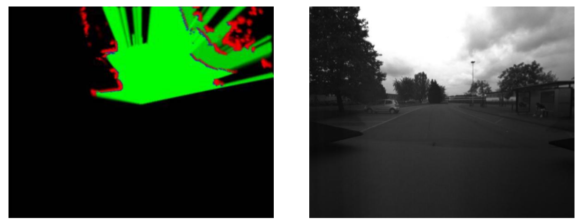
A. Occupancy Grid

The occupancy grid is a metric and discrete representation used in robotics. It represents the environment around the vehicle by a set of square cells where each cell could correspond to a free or occupied space. To construct it, the vehicle must be equipped with a set of sensors (camera, LIDAR, radar). In our work, we used a LIDAR installed in the front axis of the vehicle as shown in Fig. 1. The LIDAR uses laser beams to detect the presence of obstacles and determine their positions. The sensor provides a measurement every rotation period (100 ms in our work).

At every moment the sensor provides a new measure, the instant occupancy grid, considered to be ego-centered around the vehicle, is built. This 2D occupancy grid is constituted of $800*800$ cells for the simulation section and $400*400$ cells for the experimental part. The size of each cell is 25cm $*$ 25cm. We do not accumulate data as the frequency used for grid generation seems sufficient to ensure safe navigation. Fig. 2 shows an occupancy grid example where in the right

Fig. 1: The LIDAR mounted on the experimental vehicle.

panel we can see the real road image and in the left one the resulting grid. The green color in the occupancy grid shows a navigable space, the red shows an obstacle while the blue represents a noise or an uncertain cell. In the simulation, an occupied cell will have the value 1 and a free cell will have the value 0. Note that the blue cells are considered as occupied cells.



(a) (b)

Fig. 2: Occupancy grid example: (a) Occupancy grid, (b) Real road image

B. Tentacles Generation

In our system, the vehicle speed covers a range from 0 to 15 m/s. A set of 41 tentacles is generated at each speed. All the tentacles are represented in the vehicle local coordinate system. They start from the vehicle center of gravity and take the form of clothoids. The clothoid has a linearly variable curvature with respect to the curvilinear abscissa [2]. Its expression is presented by

$$\rho = \frac{2}{k^2} s$$

where ρ is the clothoid curvature, s is the curvilinear abscissa and k is a constant. To draw a clothoid, we use a function, as in [2], that takes as input the clothoid curvature at initial point ρ_0, the curvature variation with respect to clothoid arclength $\frac{\Delta_\rho}{\Delta_l}$, the lenght of the clothoid curve from initial to final point $L_{tentacle}$, the number of points along the clothoid n, the orientation (yaw angle) of the vehicle at the initial point of the clothoid ϕ_0 and the coordinates of initial point $(x_0\ y_0)$. This function samples the clothoid to $\frac{L_{tentacle}}{n}$ points and for each point i the following integrals ((1) and (2)) are calculated as, [2]:

$$I1 = \int t \cos\left(\frac{\Delta_\rho}{\Delta_l} i^2 \frac{t^2}{2} + \rho_0 t i + \phi_0\right) dt \tag{1}$$

$$I2 = \int t \sin\left(\frac{\Delta_\rho}{\Delta_l} i^2 \frac{t^2}{2} + \rho_0 t i + \phi_0\right) dt \tag{2}$$

x_i and y_i calculated by (3) and (4) respectively, represent the coordinates of the point i of the clothoid.

$$x_i = x_0 + i * I1, \quad (3)$$

$$y_i = y_0 + i * I2, \quad (4)$$

Consequently, we can draw the clothoid specifying just the initial curvature ρ_0, the curvature variation with respect to the clothoid length $\frac{\Delta_\rho}{\Delta_l}$ and the clothoid length $L_{tentacle}$ (x_0, y_0 and ϕ_0 will be set to zero).
We assume that all the tentacles have the same length $L_{tentacle}$ at a given vehicle speed V_x. This length is calculated by

$$L_{tentacule}(m) = \begin{cases} t_0 \ V_x - L_0 & V_x > 1(m/s) \\ 2 \ (m) & V_x \leq 1(m/s) \end{cases} \quad (5)$$

where $t_0 = 7s$ and $L_0 = 5m$ are chosen so that the clothoid form corresponds to a feasible trajectory by the vehicle. The initial curvature of the tentacles is calculated from the current vehicle steering angle and depends on the vehicle speed [12], namely,

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

where δ_0 is the current vehicle steering angle and L is the vehicle length.
The last step for the generation of our tentacles is to specify $\frac{\Delta_\rho}{\Delta_l}$ for each tentacle. We define the maximum curvature ρ_{max} as the maximum trajectory curvature that a vehicle can execute at a given vehicle spped V_x [12], namely,

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

where a_{max} is the maximum lateral acceleration that guarantees stability of the vehicle.
We consider that the tentacles curvatures at a length L_c vary between $-\rho_{max}$ and $+\rho_{max}$ where L_c is the distance of the collision (defined later in the paper). The variation in curvature $\frac{\Delta_\rho}{\Delta_l}$ is calculated from the following formula [2],

$$\rho_0 + \frac{\Delta_\rho}{\Delta_l} L_c = \rho_{max}.$$

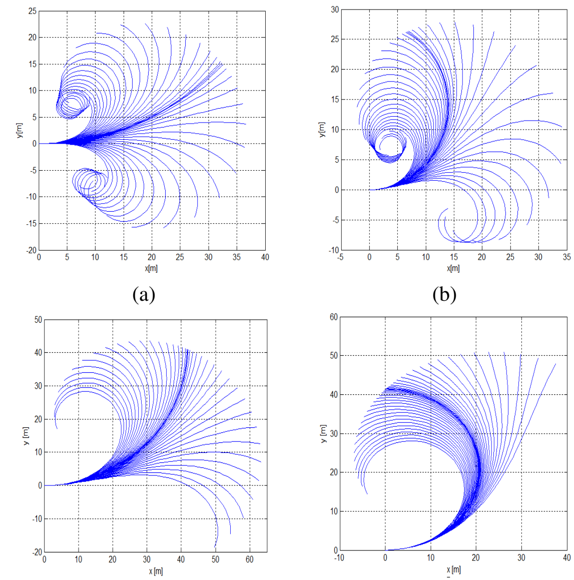
Consequently, each tentacle has a curvature varying from ρ_0 to ρ_{end}, where ρ_{end} is the final curvature on each tentacle and is calculated by

$$\rho_{end} = \rho_0 + \frac{\Delta_\rho}{\Delta_l} L_{tentacle}.$$

Hence, for the first tentacle, $\frac{\Delta_\rho}{\Delta_l}$ is set to $\frac{-\rho_{max}-\rho_0}{L_c}$ and for the last tentacle $\frac{\Delta_\rho}{\Delta_l}$ is set to $\frac{\rho_{max}-\rho_0}{L_c}$. For other tentacles, the vector $\frac{\Delta_\rho}{\Delta_l}$ is sampled from $\frac{-\rho_{max}-\rho_0}{L_c}$ to $\frac{\rho_{max}-\rho_0}{L_c}$ taking a number of samples equal to the number of tentacles (here 41 samples). We can see in Fig. 3, the clothoids computed for different speeds and different initial steering angles.

C. Navigable Tentacles Selection

The output of our path planner is a tentacle considered as a trajectory to execute. However, as the vehicle can not precisely follow the tentacle shape, and to take into account the width of the vehicle, an area called classification zone is defined around the tentacle in order to guarantee a secure navigation. This area includes cells within a radius d_c to any point on the tentacle. Using the clothoids form, the distance d_c is taken slightly greater than the width of the vehicle, unlike the case of circular arcs where d_c is taken almost two times greater than the vehicle width. The geometrical description of this area is shown in Fig. 4.



(a) (b) (c) (d)

Fig. 3: Different clothoids at different speeds and different initial steering angles: (a) V_x=6m/s δ_0=0.1rad, (b) V_x=6m/s δ_0=0.3rad, (c) V_x=10m/s δ_0=0.1rad, (d) V_x=10m/s δ_0=0.3rad

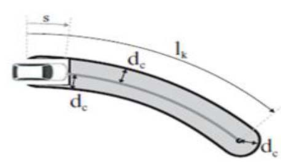


Fig. 4: Classifiction zone [15].

In our system, the d_c value is empirically specified (in m) for each speed by

$$d_c = \begin{cases} 1.4 + 0.2 \ \frac{V_x}{3} \ m & V_x < 3m/s \\ 1.6 + 0.6 \ \frac{V_x - 3}{15} \ m & 3m/s < V_x < 15m/s \end{cases}$$

where V_x defines the vehicle speed. The collision distance L_c, defined in (6), is the distance required to stop the vehicle traveling at a speed V_x with maximum longitudinal deceleration $a_{c_{max}}$ that maintains passenger comfort.

$$L_c = \frac{V_x^2}{a_{c_{max}}} \quad (6)$$

where $a_{c_{max}}$ is the maximum longitudinal deceleration of the vehicle, taken in our implementation as $a_{c_{max}} = 1.5 \ m/s^2$.

In our approach, the tentacle is classified as non-navigable if an obstacle is detected within a distance less than the collision distance L_c. If the obstacle is beyond L_c, the tentacle is classified as navigable.

D. Best Tentacle Choice

Only one tentacle among navigable tentacles is chosen as the best tentacle using three criteria $V_{clearance}$, $V_{curvature}$ and $V_{trajectory}$ defined below. This tentacle is then used as the vehicle's driving path. The three criteria should be calculated for each navigable tentacle. They are normalized to the interval [0, 1] and then linearly combined into a single function, namely,

$$V_{combined} = a_0 V_{clearance} + a_1 V_{curvature} + a_2 V_{trajectory}$$

where a_0, a_1 and a_2 are weighting parameters that can be used to change the behavior of our approach. This amounts to prefer a criteria more than another. Subsequently, each of these three criteria will be described apart.

1) Clearance Criterion: This criterion expresses the distance that the vehicle can drive along a tentacle before hitting an obstacle. To calculate this value using (7), we directly calculate the distance between the vehicle and the first obstacle found on the tentacle L_0 using the occupancy grid and the classification zone of the tentacle.

$$V_{clearance}(L_0) = \begin{cases} 0 & \textit{free tentacle} \\ 2 - \frac{2}{1+e^{-c*L_0}} & \textit{otherwise} \end{cases} \tag{7}$$

where c is a constant calculated to obtain $V_{clearance}\,(L_{0.5}) = 0.5$ at a distance $L_{0.5} = 20m$ taken in our implementation as $c = \frac{\ln(1/3)}{L_{0.5}}$.

2) Change of Curvature Criterion: The value of this criterion indicates the change of curvature on the clothoid. It aims to provide a smooth path and prevent a wide -variation in steering. It is calculated by

$$V_{curvature} = \frac{\left|\frac{\Delta_\rho}{\Delta_l}\right|}{\frac{2\rho_{max}}{L_c}}$$

3) Trajectory Criterion: The value of $V_{trajectory}$ pushes the vehicle to follow a global reference trajectory, defined for example by GPS waypoints and a global map. However, the simplest method to estimate $V_{trajectory}$ is to consider a single point on the tentacle taken at the collision distance L_c and its corresponding point on the trajectory as shown in Fig. 5. For each tentacle, a measurement V_{dist} is calculated as shown in (8) by taking both the distance b between the point on the tentacle and its corresponding point on the trajectory as well as its relative tangent orientations α [15]. $V_{trajectory}$ is then the normalized value of V_{dist}. The combination of these two parameters (b and α) in the evaluation of the trajectory criterion leads us to choose the most oriented tentacle toward the reference trajectory even if the lateral error (represented by b) is not small.

$$V_{dist} = b + c_a \alpha \tag{8}$$

$$V_{trajectory} = \frac{V_{dist} - V_{min}}{V_{max} - V_{min}} \tag{9}$$

where c_a represents a scale between the linear distance and the tangent orientations ($c_a = 0.3\ m/rad$) ,V_{max} and V_{min} are

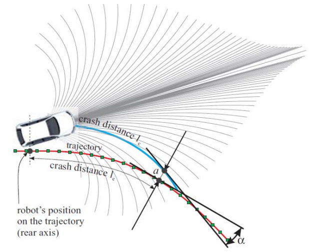


Fig. 5: $V_{trajectory}$ criterion calculation [15].

the maximum and minimum values of V_{dist} calculated for all the tentacles.

E. Trajectory Tracking

The best tentacle selected by the path planner is considered as a trajectory to execute. The vehicle is then controlled by a path controller that derives the steering commands from the selected tentacle parameters. The controller used in this work, is based on Immersion and Invariance principle ($I\&I$). The main idea of the $I\&I$ approach is to achieve the control objective by immersing the plant dynamics into a (possibly lower-order) target system that captures the desired behavior [1]. In fact, the $I\&I$ theory is to define a target dynamics and to design a control law that renders the manifold of the target dynamics attractive and invariant. In other words, we have to find a manifold in state-space that can be rendered invariant and attractive; with internal dynamics reflecting the desired closed-loop dynamics and to design a control law that brings the state of the system towards the manifold.

The main objective of the steering controller is to cancel the lateral error displacement with respect to a given trajectory, i.e., $e = \dot{e} = 0$ at the equilibrium. We define the off-the-manifold variable

$$z = \dot{e} + \lambda e, \quad \text{s.t.} \quad \lambda > 0 \tag{10}$$

The control input should be selected in such a way that the trajectories of the closed-loop system are bounded and $z = \dot{e} + \lambda e$ converges to zero. Notice that, when $z \to 0$, e converges exponentially to zero with the rate of convergence λ, yielding to the desired result. To this end, we choose

$$\dot{z} = -Kz, \quad \text{with} \quad K > 0. \tag{11}$$

where K represents the rate of exponential convergence of z to zero. After some calculations, one can find that the corresponding control input has the following expression:

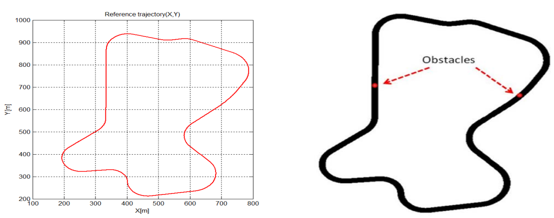
$$\begin{aligned} \delta_{I\&I} = &-\frac{m(K+\lambda)}{C_f}\dot{e} - \frac{mK\lambda}{C_f}e + \frac{C_f+C_r}{C_f}\beta \\ &+ \frac{L_f C_f - L_r C_r}{C_f V_x}\dot{\psi} + \frac{mV_x^2}{C_f}\rho \end{aligned} \tag{12}$$

where m , C_f , C_r, L_f, L_r are the vehicle parameters and correspond to the vehicle mass, the cornering stiffness of the front tire, the cornering stiffness of the rear tire, the distance between the front axle and the vehicle center of gravity and the distance between the rear axle and the vehicle center of gravity respectively; $\dot{\psi}$, β, e, $\dot{e}$ are the dynamic variables of the vehicle and define the yaw rate, the sideslip angle, the lateral error and its derivative respectively, while ρ corresponds to the desired curvature. The design of this controller guarantees robust stability and some a priori known performances (the response time of the controller is entirely determined by the parameters K and λ). In [10], the controller is validated according to several real driving scenarios. Simulations were performed on SCANeR Studio,a driving simulation engine, using experimental data acquired by the DYNA vehicle (a Peugeot 308) belonging to the Heudiasyc laboratory. The validation demonstrates the robustness and good performances of the proposed controller. For further information about the *I&I* controller, interested readers are invited to see [9].

III. RESULTS

A. Simulation Results

To validate the algorithm, it was applied to a scenario taken from the Scaner-Studio simulator. The data taken from Scaner-Studio was processed and simulated in Matlab. The vehicle speed is set to 6 m/s in the simulation. Using this data, we generate a global map showing the reference trajectory with its right and left borders. In this global map shown in Fig. 6, the navigable space of the road is illustrated by black cells (having the value 0) while the non-navigable space is illustrated by white cells (having the value 1).



(a) (b)

Fig. 6: Global map: (a) Reference trajectory, (b) global grid

After positioning the vehicle in the global map, the ego-centric local occupancy grid is generated. Then, we apply our algorithm to choose the best tentacle at each sampling step. The chosen tentacle is then applied to the vehicle using the lateral controller *I&I* presented above. The algorithm is validated upon the whole scenario shown in Fig. 6, yet we just show in Fig. 7 an obstacle avoidance scenario. The parameters a_0, a_1 and a_2 are set to 0.1, 0.2 and 0.5 respectively. The same scenario was tested using circular tentacles in order to compare the performance of our planner

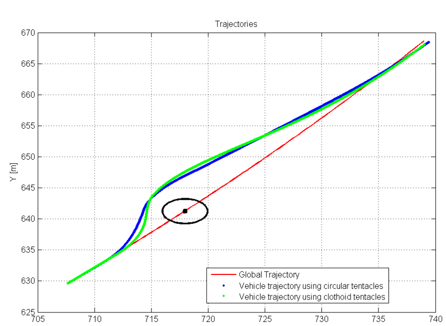


Fig. 7: Obstacle avoidance scenario tested with circular and clothoid tentacles.

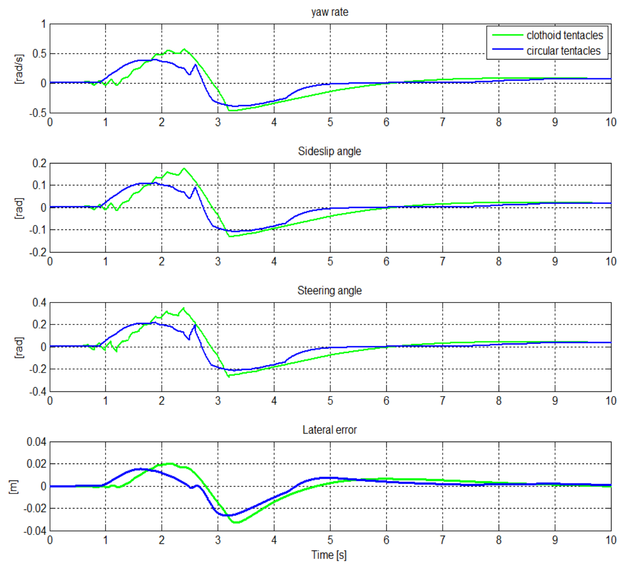


Fig. 8: Vehicle dynamic variables.

with other local trajectory planning algorithms. In Fig. 7, the red line corresponds to the reference trajectory, the blue and green lines correspond to the vehicle trajectory computed using circular tentacles and clothoid tentacles respectively and the black circle is assumed to be an obstacle on the road.

We observe that the vehicle succeeds in avoiding the obstacle and returning back to the reference trajectory when using clothoid tentacles as well as circular ones. However, as the initial dynamical state of the vehicle is not considered in the circular shape of the tentacles, the vehicle take a larger lateral security distance when driving around the obstacle, especially when trying to avoid the obstacle. Plus, using circular tentacles method, the vehicle would declare a narrow gate as impassable at high speeds while still being able to drive having other tentacles shapes. The clothoid tentacles method proposed in this work solves this ambiguity by considering the initial dynamical state of the vehicle and taking the classification zone slightly greater than the vehicle width. Thus, the vehicle take a smaller lateral security distance when driving around the obstacle or on a narrow gate. We also observe the variation of the vehicle dynamic variables such as the yaw rate, the sideslip angle, the steering angle and the lateral error in Fig. 8. The clothoid forms make the vehicle dynamic variables more smoother than the circular forms.

B. Preliminary Experimental Results

To evaluate the developed algorithm, we made an experimental test using a Citroën vehicle in the laboratory Heudiasyc. The vehicle is equipped with a LIDAR used to generate the occupancy grid and other sensors used for the acquisition of vehicle dynamics data such as its position, steering wheel angle, speed, acceleration; moreover a camera acquires the images of the road. As Fig. 9 shows, this test was done near the High School Charles de Gaulle in Compiègne.

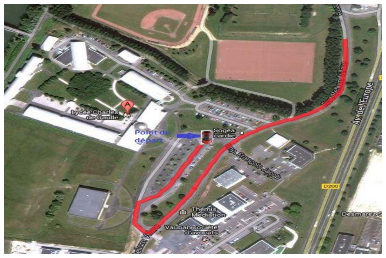
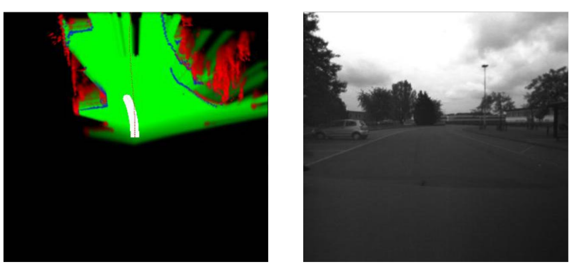


Fig. 9: The road where the test is performed.

As the vehicle is not robotized, it was driven by a driver with a mean speed of 22 km/h, so the data acquired, during this test, was used for the offline evaluation of our path planning algorithm. In Fig. 9, the starting point is indicated by the red car while the red path corresponding to the path followed by the vehicle during the experiment will be considered as the reference trajectory for our algorithm.

In Fig. 10.b we see the real image of the road and in Fig. 10.a the occupancy grid corresponding to the real road image. In the occupancy grid, the green cells show the free space, the red cells show the cells occupied by an obstacle and the blue cells could correspond to noisy data. The white area included in the occupancy grid corresponds to the support area around the tentacle chosen by our algorithm while the dotted red curve represents the reference path traveled by the vehicle. The latter curve can present some errors because of the GPS uncertainty. Every 100ms, a new clothoid tentacle is calculated by our algorithm, in the local occupancy grid, based on the acquired real data. The proposed tentacle is compared to the real trajectory of the vehicle which represents the reference trajectory. The behavior of our algorithm seems coherent with respect to the desired objective. To proceed to more valid experimental validation, the algorithm will be implemented in the near future on a robotized vehicle in the laboratory Heudiasyc, in closed loop with our developed controller.



(a) (b)

Fig. 10: Occupancy grid with the support area of the chosen tentacle: (a) Occupancy grid, (b) Real road image

IV. CONCLUSION AND PERSPECTIVES

The simulation and experimental results show that our algorithm is well-suited against the expected objectives. In fact, the method is a fast reactive method since it does not perform data storage and the use of clothoids shape makes the method more realistic with respect to vehicle dynamics and real road structure. This method is very promising for vehicles autonomous navigation, it is suitable for real time application, moreover, the curvature criterion that we propose in this paper can be extended to choose some desired curvature at certain distance. Among the perspectives, we look to improve the algorithm by optimizing the number of calculated tentacles and the computing time, and by proposing other criteria. We also look to implement our algorithm in Scaner-Studio simulator and execute an online test using a robotized vehicle in order to highlight the validity of the developed trajectory planner. Further, an interesting extension of our method would be to consider moving obstacles such as moving cars or pedestrians. This improvement would be at the perception stage, by considering the obstacle's movement in the occupation grid. The method consists in projecting recognized obstacles and their expected movement in the occupancy grid. Then, each mobile object generates a set of obstacles representing the future movements of this obstacle. Obstacle movement parameters such as speed and direction have to be deduced from sensor data. Trajectory planning and tracking are then done relative to these previsions taking into account uncertainties for more security.

REFERENCES

[1] A. Astolfi, D. Karagiannis, and R. Ortega. *Nonlinear and Adaptive Control with Applications*. Springer, 2008.

[2] E. Bertolazzi and M. Frego. g^1 fitting with clothoids. *Matematical Methods in the Applied Sciences*, 2013.

[3] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1179–1187, 1989.

[4] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, 7(3):278–288, 1991.

[5] K. Chu, M. Lee, and M. Sunwoo. Local path planning for off-road autonomous,driving with avoidance of static obstacles. *IEEE Transactions On Intelligent Transportation Systems*, 13(4):1599 – 1616, December 2012.

[6] H. Durrant-Whyte and T. Bailey. *Simultaneous Localisation and Mapping (SLAM)*.

[7] Thrun et al. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 2006.

[8] M. Himmelsbach, T. Luettel, F. Hecker, F. von 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.

[9] G. Tagnes, R. Talj, and A. Charara. Immersion and invariance control for reference trajectory tracking of autonomous vehicles. In *the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 2322–2328, 2013.

[10] G. Tagnes, R. Talj, and A. Charara. Design and validation of a robust immersion and invariance controller for the lateral dynamics of intelligent vehicles. *ControlEngineeringPractice*, 40:81–92, 2015.

[11] R. Talj. *Véhicules Intelligents,Navigation autonome : Méthode des tentacules*, october 2011.

[12] Y. TANG. Trajectory planning for autonomous vehicles navigation. Master's thesis, University of Technologie of Compiégne, 2013.

[13] I. Ulrich and J. Borenstein. Vfh+: Reliable obstacle avoidance for fast mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 1572 – 1577, Leuven, Belgium, May 1998.

[14] I. Ulrich and J. Borenstein. Vfh*:local obstacle avoidance with look-ahead verification. In *IEEE International Conference on Robotics and Automation*, pages 2505–2511, San Francisco, April 2000.

[15] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche. Driving with tentacles: Integral structures for sensing and motion. *Journal of Field Robotics*, 25(9):640–673, 2008.

[16] X. Wang. *Wall Following Algorithm for a Mobile Robot Using Extended Kalman Filter*. PhD thesis, Auburn University, Auburn, Alabama, 2011.