2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)
September 12-14, 2018, Madrid, Spain

Smooth Obstacle Avoidance Path Planning for Autonomous Vehicles

Wael Ben-Messaoud
IRIMAS EA7499
Université de Haute-Alsace
Mulhouse, France
wael.ben-messaoud@uha.fr

Michel Basset
IRIMAS EA7499
Université de Haute-Alsace
Mulhouse, France
michel.basset@uha.fr

Jean-Philippe Lauffenburger
IRIMAS EA7499
Université de Haute-Alsace
Mulhouse, France
jean-philippe.lauffenburger@uha.fr

Rodolfo Orjuela
IRIMAS EA7499
Université de Haute-Alsace
Mulhouse, France
rodolfo.orjuela@uha.fr

***Abstract*—Obstacle avoidance and overtaking are important manoeuvres for autonomous driving. The collision avoidance with a vehicle, a pedestrian or any obstacle and the generation of a feasible continuous curvature trajectory represent the major problems faced by researchers to provide a safe path planning solution. This paper presents an algorithm able to avoid vehicles or obstacles by proposing a smooth local modified trajectory of a global path. The proposed method is based on the combination of a parameterized sigmoid function and a rolling horizon. The major advantage of this method is the reactivity to the obstacle motion and the generation of a smooth trajectory in a low execution time. The lateral and the longitudinal safety distances are easily parameterized when generating the avoidance trajectory. Simulation results show that the algorithm performs collision avoidance manoeuvres for static and dynamic obstacles in an effective way. The method is validated by applying the avoidance approach on real measured trajectory.**

***Index Terms*—autonomous vehicle, path planning, obstacle avoidance, sigmoid function.**

I. INTRODUCTION

Autonomous driving is currently an active research subject especially for perception and collision avoidance. An autonomously driven vehicle must, first of all, be able to determine its path with respect to the perceived static and dynamic environment. After that, it plans the movement to be carried out to achieve its objective by respecting a certain number of constraints related to the environment (presence of obstacles) and vehicle dynamics (planning stage). Thus, the complete process that allows a vehicle to navigate on its environment can be summed up in highly interdependent modules: perception (including localization), planning and control [1]. Trajectory planning for autonomous vehicles has been studied in different fields like robotics, autonomous cars, unmanned vehicles, underwater vehicles and drones [2]–[6]. The main challenge when generating a trajectory is ensuring its feasibility by taking into consideration several constraints related to vehicle dynamics, the road profile, etc. The need of a low execution time is also an essential factor in order to be implemented in real time. We can distinguish two categories of path planning approach for obstacle avoidance: the global planning and the local planning. The principle of global approaches is to determine a complete movement between an initial position and a final position from a priori known environment information (e.g. provided by a digital map and a localization system). The most common used methods for global path planning are Dijkstra [7], A* and Rapidly-exploring Random Tree RRT [8]. These methods are time-consuming as the calculation of the whole trajectory in real time is slow especially when the map is wide and thus, it can hardly be applied in real time especially for autonomous cars and when obstacles are dynamic.

Local approaches restrict the trajectory calculation only on a limited time/distance window by taking the perceived environment into account. The local path is limited to the perception range of the sensors or camera. These methods are used as the whole path is unknown in the majority of cases. The reference trajectory is calculated simultaneously when the displacement is started. Using these methods, the navigation can be ensured in a partially known environment and also with dynamic obstacles.

Several local methods based on interpolating the trajectory by mathematical expressions are proposed in the literature. Clothoid curves [9], [10], polynomial curves [11], Bézier curves and spline curves [12], [13] are some interesting examples of these methods. Clothoid curves are used for instance in tentacle methods [10]. The latter is based on the selection of one trajectory from a set of candidate trajectories using an optimization criterion. Yet, its major issue is the long computation time because a great number of paths is generated at each step before the selection of the best candidate through optimization. Polynomial, Bézier and Spline curves are able to generate a smooth trajectory by linking different way-points representing the free-space navigation for the vehicle. A better compromise must be reached between the order of the mathematical function and the computation time. Sigmoid-based curve is a local planning method based on an easily configurable path [14]. The parameters of the developed sigmoid function are able to adjust the smoothness of the obstacle avoidance manoeuvre and the safety distance between the ego vehicle and the leading vehicle to be avoided. In local planning, planning area or horizon computation is performed such as in the Horizon Planning Approach which consists in planning the trajectory by dividing the space into regions [15], [16]. This allows quickly computing trajectories by dividing the problem into sub-problems. The horizon can be expressed in time [17] or in distance [18], [19].

978-1-5386-3543-8/18/$31.00 ©2018 IEEE

In this paper, we will present a new algorithm based on the sigmoid function and the rolling horizon method. The proposed approach is able to locally plan trajectories for autonomous vehicles, especially for collision avoidance and overtaking manoeuvers. Based on a global reference path, the algorithm generates a local trajectory in a very small execution time (less that a few tens of μs as illustrated in the simulation section) to avoid the static and dynamic obstacles. The smooth path in the rolling horizon is optimized in terms of the execution time when calculating the safe obstacle avoidance trajectory. The paper is decomposed into three sections: in the second section, the problem statement of the obstacle avoidance methods is defined. In section III, the new proposed method is explained and the steps of the algorithm are presented. Finally, the validation of the method is done using simulation results for different situations and for real measured trajectory.

II. PROBLEM STATEMENT

The main issue when avoiding detected obstacles or overtaking vehicles is to propose an easily configurable online method to modify an initial trajectory. The challenge is to find a compromise between the expected configurable trajectory and its reactivity to the variation of the obstacles position. The easy configuration of the path is necessary to correlate it to the vehicle speed. The low execution time of the generated trajectory represents an important criterion also as the trajectory must be implemented in real time.

This paper deals with a local path modification of a global reference trajectory. Its focus is to propose a safe trajectory by avoiding static and dynamic obstacles thanks to a lane change manoeuvre. The initial trajectory can be provided from a global path planner using for instance a digital map database or a recorded map. The architecture to control the autonomous vehicle is generally composed of different steps (see Fig. 1) which are perception (including localization), trajectory generation and control such as the one developed in our laboratory [1].

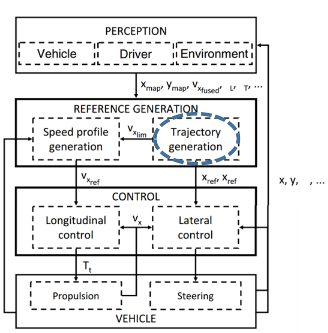

Fig. 1. Architecture of the control of the vehicle proposed by [1].

In this study, we focus on the trajectory generation module by supposing that the ego vehicle, i.e. the car which has to perform the lane change, is accurately localized and that the perception module provides the distance between the ego vehicle and the obstacles. Long range sensors like Light Detection and Ranging (LIDAR) sensors and camera can for instance be used to detect and localize obstacles [20], [21]. The dimensions of the avoided obstacle or vehicle are supposed to be known. In practice, the dimensions of the avoided obstacle can be measured in real time using LIDAR. We also assume that the moving obstacle velocities are known and that the avoidance task is performed with constant longitudinal speed for the ego-vehicle.

Our developed work associates two approaches: [14] where the authors propose the use of a sigmoid function to avoid the obstacle and the use of a rolling horizon to plan for a decomposed area of the path [15]. We propose to combine these ideas to provide an easily configurable continuous curvature path, updated at every step and to guarantee a lateral and a longitudinal safety distance to avoid obstacles. The rolling horizon approach will allow to decompose the global trajectory to local parts and calculate at each horizon step the next predicted trajectory.

A. *Sigmoid-based function*

The sigmoid function (1) proposed in [14] is used to add a lateral offset from a straight line.

$$y(x) = \frac{B}{1 + e^{(-a(x-c))}} \tag{1}$$

where $y(x)$ is the lateral offset of the vehicle and x is the position in longitudinal direction; B represents the way position P3 to reach to generate the obstacle avoidance manoeuvre; c modifies the shape of the function and a is the slope of the sigmoid. Fig. 2 illustrates the shape of the function and its curvature when considering three values of the parameter a. Noticing that the point P2 is independent to a and its position can only be imposed by c.

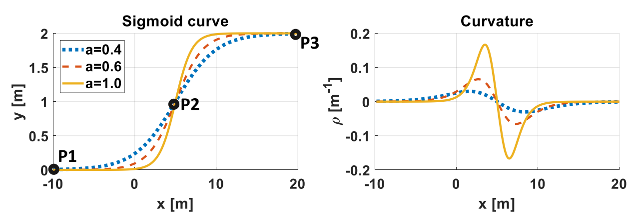

Fig. 2. Illustration of the sigmoid curve proposed in [22].

The inputs of the function are the obstacle position and the position of the vehicle. Then, the lateral offset is calculated depending on these parameters. In [22], the method proposed by the authors is limited to static obstacle avoidance. The execution time which is an important criterion is also not evaluated. This method can be improved by adapting it in real time and relating the planned trajectory distance to the sensors detection capabilities.

B. *Horizon planning approach*

The Horizon Planning approach (HP) consists in computing the path by dividing the drivable space into convex regions

[15]. The trajectory of each region is computed as the vehicle moves forward. The main advantage is the low execution time and the reactivity to the obstacle movement. This allows quickly computing trajectories and divides the problem into sub-problems but requires treating the continuity problem between each planned path. HP approach is widely used in the case of autonomous unmanned vehicles [22]. This method is useful when the environment is partially unknown. The dimension of each horizon must be correlated to the sensors perception range in order to be reactive to the sensed obstacles or the leading vehicle to be avoided.

III. Proposed Approach

A. Principle of the approach

The approach developed in this paper extends the use of the sigmoid-based path planning either to static or dynamic obstacles perceived in the driving scene. Our proposed algorithm is based on the two previously mentioned principles by using a configurable sigmoid function to provide an efficient and feasible path to the vehicle in a given horizon. The path takes into consideration the position and the motion of the perceived obstacles or the leading vehicle.

The algorithm is constituted of different parts able to react to any detected obstacle on the road. The initial reference path of the vehicle can be provided from a digital map database or a recorded map. The inputs of the algorithm are the vehicle position and the obstacle positions which represent the perception and localization parts as illustrated in Fig. 3. The local path is then calculated depending on the desired parameter of smoothness and the two parameters of safety distance to be respected during the avoidance manoeuvre. This parametric smooth variation is necessary to maintain a level of safety and comfort to the passenger of the vehicle and to respect the vehicle dynamics when doing a manoeuvre. The safety distance when avoiding obstacle can be configurable by taking into consideration the vehicle velocity. The way-point calculated represents the position to reach when avoiding the obstacle/vehicle based on the generated avoidance path.

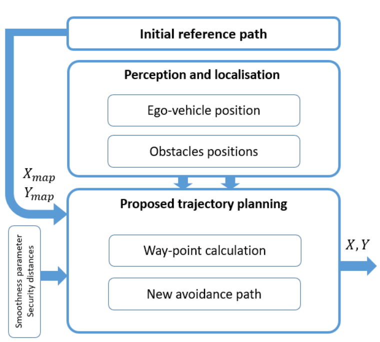

Fig. 3. Path-planning strategy.

The main objective is to calculate the next local path via the Rolling Horizon Planning Approach depending on the perception area of the vehicle sensors. The calculation of the next navigation area is automatically performed as the vehicle is moving forward. Thanks to the perception layer, the obstacle position is defined at every step. When considering moving objects, the next sampling trajectory is recalculated to be adapted to the new obstacle position.

B. Rolling horizon planning approach

The horizon length depends on the detection range of the sensors and can be configured. It can be also correlated to the speed of the car and be adaptive at real time. All horizon vectors have the same length (Fig. 4); at the first iteration (iteration 1), the 1st horizon vector which represents the next local planned trajectory is calculated. When the vehicle position reaches the horizon step distance, the second horizon path is calculated for the next planned trajectory. Notice that a common part of each vector is recalculated at each iteration in order to update the trajectory for the obstacles position changes.

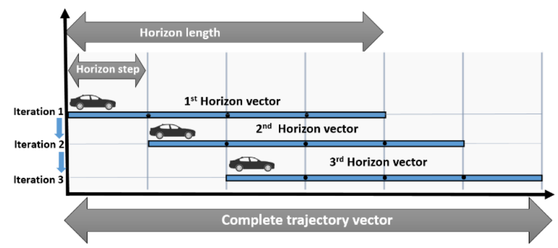

Fig. 4. The proposed rolling horizon planning approach.

The horizon step distance must be also well chosen as it concerns the distance from which the trajectory must be recalculated and updated. The ratio between the horizon length and the step length represents the number of points to be calculated in each horizon vector. Evidently, when the step length is small, the position vector (X, Y) at each horizon contains more points and will be more precise in the generation of the trajectory. The algorithm finishes when the last horizon vector reaches the limit of the complete trajectory vector. In the next sub-section, the steps of the algorithm are detailed.

C. Execution steps

The algorithm is decomposed into different steps as illustrated in Fig. 5.

The first step is to sense the existence of obstacles in the scene. If an obstacle is detected, its position $(X_0, Y_0)$is recorded. Then, the distance separating the initial local path from the first obstacle is calculated. And for each horizon, this vector of the Euclidean distances $d(N)$ between the vector of the initial trajectory $(X(N), Y(N))$ and the obstacle, is calculated (N represents the index of the elements of the horizon vector). The vector of the circular trajectory $Y_r(N)$ is then generated by applying the equality between $d(N)$ and the desired lateral safety distance S if $d(N) \leq S$ (Fig. 6).

The goal is to build a safety circular area around the obstacle following (2):

$$Y_r(N) = \sqrt{(S^2 - (X(N) - X_0)^2)} + Y_0. \qquad (2)$$

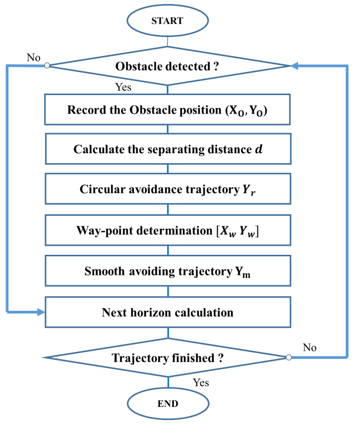

Fig. 5. Execution steps of the algorithm.

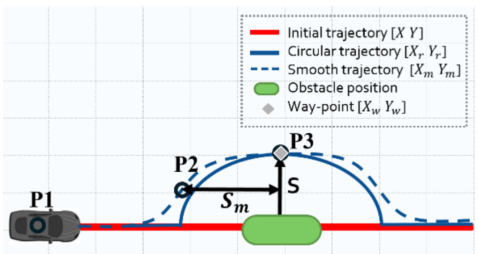

Fig. 6. Illustration of the initial and the avoiding trajectories in a local path and the way-point determination in presence of an obstacle.

The next step is the generation of a parameterized smooth trajectory linking the current vehicle position (P1) to the target way-point position (P3). This point guarantees an imposed safety distance with a smooth overtaking manoeuvre. The obstacle avoidance path $Y_m(N)$ (3) is calculated to reach P3:

$$Y_m(N) = \frac{Y_w}{(1 + e^{(C(d(N) - S_m)))}} \tag{3}$$

where C is a smoothness coefficient ($C \in [0\ 1]$) of the exponential function, S_m is the longitudinal safety distance determining the avoidance manoeuvre distance and $d(N)$ is the distance between the initial trajectory and the obstacle.

D. Path evaluation

The possibility to implement the approach in the vehicle depends mainly on the respect of the lateral vehicle dynamics limitation. One of the important parameters of the trajectory is the curvature ρ, it must be continuous and upper bounded. This parameter is expressed by the equation (4) which represents the ratio between the angle of the tangent to the path θ and the distance between two consecutive coordinates as follows:

$$\rho = \frac{\Delta\theta}{\sqrt{(\Delta X)^2 + (\Delta Y)^2}} \quad [m^{-1}]. \tag{4}$$

The angle of the tangent to the path is calculated using the following expression:

$$\theta = \tan^{-1}\left(\frac{\Delta Y}{\Delta X}\right). \tag{5}$$

IV. SIMULATION RESULTS

In this section, the simulation results of the obstacle avoidance method for left-hand overtaking direction will be presented in different scenarios. The right-hand overtaking direction is also possible by modifying the way-point of the curve to the right side of the obstacle. The execution time which represents a crucial parameter when implementing the approach in real time is evaluated. The simulations presented in this section are carried out using Matlab software. The computer used in simulation has a processor Intel® Core™ $i7-7820HQ$ CPU @ 2.90 GHz and 16 Go of RAM.

A. Influence of the smoothness parameter on the curvature

The parameter C in equation (3) has a direct influence on the smoothness of the generated path. This will provide an adaptive curvature of the path which can be correlated to the vehicle speed and dynamic behaviour to provide a feasible path for the vehicle. Simulation tests have been conducted using two values of parameter $C = 0.5$ and $C = 1$. The smaller value of C gives a smoother curve as shown in the Fig. 7. This simulation illustrates also the influence of this parameter on the curvature of the trajectory. The maximum curvature ρ is reduced from 0.2 for $C = 1$ to $\rho = 0.06$ for $C = 0.5$. As depicted in Fig. 7, the curve reaches the point P2 in S_m meters before the obstacle which is chosen at 10 meters for this example. The safety lateral deviation S is chosen to be 4 meters.

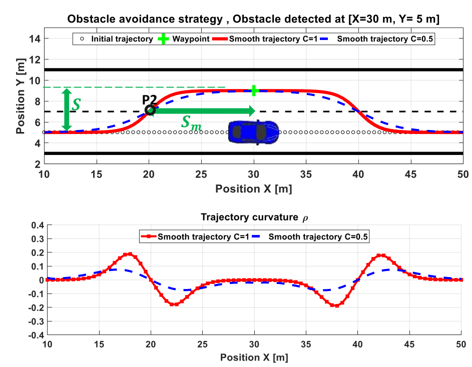

Fig. 7. Influence of the smoothness parameter C on the path avoidance trajectory and the curvature, two values of C are used for the simulation, $C = 1$ and $C = 0.5$.

B. Avoidance of a dynamic object (overtaking)

The simulation of the overtaking manoeuver is investigated in this sub-section, the ego vehicle is placed at the position $(X, Y) = (0, 5)$ meters at the beginning of the simulation whereas the leading vehicle to avoid is placed at $(X_0, Y_0) = (30, 5)$ meters of the ego vehicle. While the whole trajectory considered for simulation is 60 meters, the rolling horizon is set at 20 meters with the horizon step of 0.5 meter. So, every 0.5 meters, a new trajectory vector $Y_m(N)$ is generated. The calculation of the path takes into account several parameters: the obstacle coordinates, the smoothness parameters of the sigmoid function C, the lateral and the longitudinal safety distance. The initial trajectory is considered as a straight line. When executing the generation of the new path, the position of the dynamic obstacle is assigned at $(X_0, Y_0) = (30, 5)$ meters and moves to achieve $X = 37.5$ meters at the end of the simulation. The track of the leading vehicle is presented in grey color. The smoothness parameter C is imposed to 1 and the lateral safety distance S is equal to 4 meters and the longitudinal safety distance S_m is equal to 10 meters. The chosen rolling horizon is 20 meters but it can be easily adapted according to the sensor detection range.

Fig. 8 illustrates the simulation results of the previously mentioned situation. The three consecutive simulations show three situations: before, during and after the overtaking manoeuvre. At the beginning of the simulation (Fig. 8.a), the leading vehicle is not yet detected as the rolling horizon is 20 meters (green line). Subsequently, when the leading vehicle is detected, the new trajectory is generated and adapted in real time. Even if the obstacle is moving, its position is sensed and the way-point is updated by tracking it. The generated path is evaluated by calculating the curvature which is smooth and does not reach 0.2 m^{-1}.

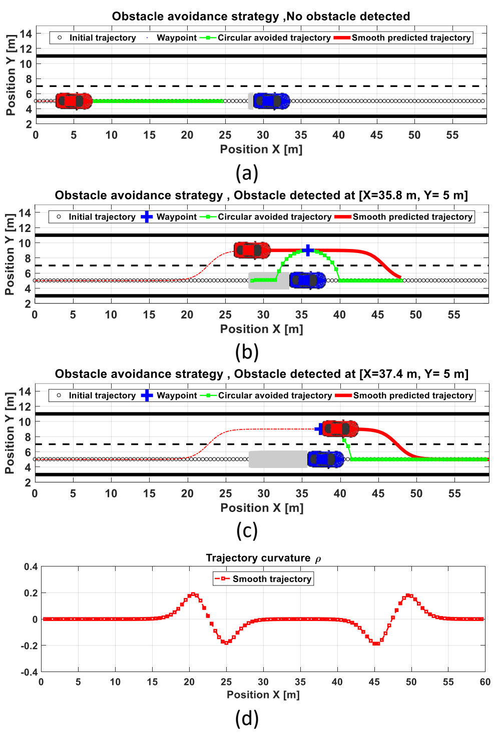

Fig. 8. Simulation results of the overtaking manoeuvre, the position of the leading vehicle is varying. During the overtaking manoeuver, three simulations are consequently showed before (a), during (b) and after (c) the manoeuvre. The curvature of the generated path is plotted on (d).

The execution time of each path vector (Fig. 9) is computed at each new step, the number of calculated horizons is $(60 - 20)/0.5 = 80$ calculated values for 20 meters of horizon and 0.5 m of step length. After five simulations, the average of the execution time of each horizon is very low and equal to 43 μs with 2 μs of standard deviation. This time represents the calculation of each calculated horizon containing $(20/0.5) = 40$ points corresponding to 20 meters of the planned trajectory.

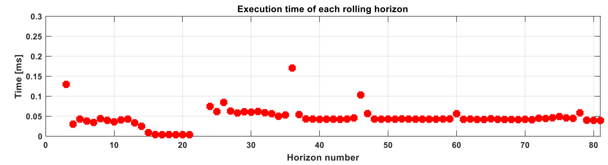

Fig. 9. Execution time of each horizon step to calculate the trajectory presented in Fig. 8. The average of the execution time is 43 μs to calculate 20 meters of trajectory.

C. Real map simulation

A real trajectory has been recorded at the Museum *Cité de l'Automobile* in Mulhouse to test the developed approach as illustrated in Fig. 10. An obstacle is placed on the map with given coordinates (X_0, Y_0). From the obstacle position and the reference trajectory, we applied our developed method which consists in avoiding the obstacle by guarantying imposed safety distances. The parameters of the sigmoid function used for the simulation test are: $C = 0.2$, lateral safety distance $S = 3\,m$ and the longitudinal safety $S_m = 15\,m$.

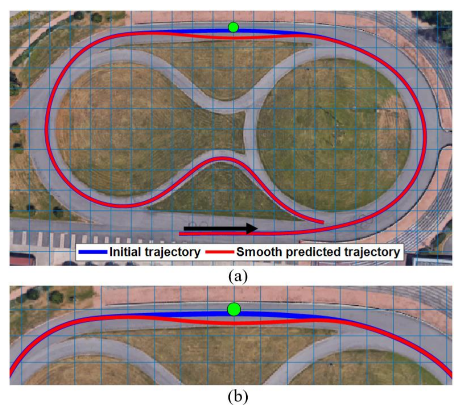

Fig. 10. Simulation results of a real path by placing a circular shaped obstacle on the reference trajectory plotted in blue (a). The new path after applying the avoiding method is in red. In (b) a zoom-in of the avoidance part of the path. A blue grid is plotted on the figure with a resolution of 5 meters.

In order to validate the proposed avoidance path with respect to the vehicle dynamics, the trajectory presented in Fig. 10 is implemented on a virtual car in CarMaker software (IPG Automotive). The avoidance trajectory is tested with a guidance controller developed in the laboratory based on robust state feedback coupled with feed-forward action [23]. This test is performed at 40 km/h (the maximal speed guaranteeing the vehicle stability for this test road) and the results are plotted in Fig. 11. The figure time axis is limited to the zoomed trajectory in Fig. 10 in order to show vehicle dynamics around the obstacle avoidance. We notice that the curvature followed by the vehicle measured by the CarMaker follows the road trajectory. The lateral acceleration of the car changes slightly and is limited between 0 and -2 ms^{-2} . It is noted also that the variation of the yaw rate and the steering angle is low and the curves are continuous which enhance passenger comfort.

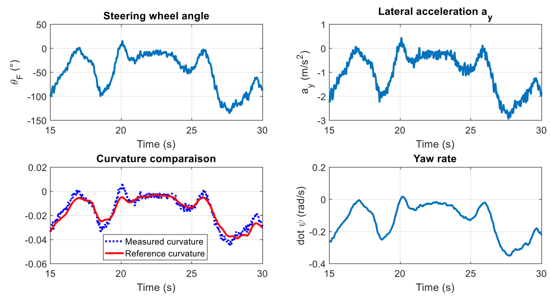

Fig. 11. Vehicle dynamics and curvatures plotted as a function of time and limited around the avoidance manoeuver

V. CONCLUSION AND FUTURE WORK

In this paper, the light was shed on path planning of autonomous vehicles. A new method of obstacle avoidance and overtaking has been presented. The idea consists in proposing a safe path for the vehicle instead of the initial trajectory. The method is based on the use of a parametric sigmoid function coupled with a rolling horizon approach to divide the trajectory in convex regions. Each horizon is calculated and updated at each step.This is then used to calculate online a portion of the path. The most interesting features of this method are the configurable smoothness of the trajectory and the very low execution time. The lateral and the longitudinal safety distances between the ego vehicle and the avoided obstacle can be easily parameterized depending on the speed of the obstacle. The developed approach has been validated by simulating different situations. The results of lane change when avoiding static or dynamic obstacles on a real recorded map prove the validity of our strategy. We plan to extend our algorithm for multi-obstacle avoidance as it is a common situation for overtaking manoeuvres. We are actually working on the correlation of the parameters of the sigmoid function to the vehicle speed and to the maximum allowed curvature and their real time adaptation. Further research will be focused on the integration of the path modification with the lateral control of the vehicle and their implementation on our laboratory autonomous car ARTEMIPS. Finally, considering this configurable low parameters method, a comparison with state-of-the-art local planning solutions will also be performed.

REFERENCES

[1] R. Attia, J. Daniel, J.-P. Lauffenburger, R. Orjuela, and M. Basset, "Reference generation and control strategy for automated vehicle guidance," in *the IEEE Intelligent Vehicles Symposium (IV)*, Madrid, Spain, 2012, pp. 389–394.

[2] O. Esrafilian and H. D. Taghirad, "Autonomous flight and obstacle avoidance of a quadrotor by monocular SLAM," in *the 4th International Conference on Robotics and Mechatronics (ICRoM)*, Tehran, Iran, 2016, pp. 240–245.

[3] M. Fraiwan, A. Alsaleem, H. Abandeh, and O. Aljarrah, "Obstacle avoidance and navigation in robotic systems: A land and aerial robots study," in *the 5th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, 2014, pp. 1–5.

[4] J. Larson, M. Bruch, R. Halterman, J. Rogers, and R. Webster, "Advances in autonomous obstacle avoidance for unmanned surface vehicles," Space and Naval Warfare Systems Center, San Diego, CA, Tech. Rep., 2007.

[5] J. Larson, M. Bruch, and J. Ebken, "Autonomous navigation and obstacle avoidance for unmanned surface vehicles," in *SPIE Unmanned Systems Technology VIII, Defense Security Symposium*, Orlando, FL, USA, 2006.

[6] H. Xu, L. Gao, J. Liu, Y. Wang, and H. Zhao, "Experiments with obstacle and terrain avoidance of autonomous underwater vehicle," in *the OCEANS 15 MTS/IEEE*, Washington, DC, USA, 2015, pp. 1–4.

[7] S. Skiena, "Dijkstras algorithm," *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley*, pp. 225–227, 1990.

[8] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011, pp. 723–730.

[9] F. Von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *Journal of Field Robotics*, vol. 25, no. 9, pp. 640–673, 2008.

[10] H. Mouhagir, V. Cherfaoui, R. Talj, F. Aioun, and F. Guillemard, "Using Evidential Occupancy Grid for Vehicle Trajectory Planning Under Uncertainty with Tentacles," in *the 20th IEEE International Conference on Intelligent Transportation*, Yokohama, Japan, 2017, pp. 1–7.

[11] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouveliere, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 589–606, 2010.

[12] D. Gonzalez, J. Perez, R. Lattarulo, V. Milanes, and F. Nashashibi, "Continuous curvature planning with obstacle avoidance capabilities in urban scenarios," in *The 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, China, 2014, pp. 1430–1435.

[13] 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. 3-4, pp. 269–296, 2011.

[14] M. Arbitmann, U. Sthlin, M. Schorn, and R. Isermann, "Method and device for performing a collision avoidance maneuver," US Patent US8 209 090B2, 2012.

[15] X. Wang and H. Kopfer, "Dynamic collaborative transportation planning: A rolling horizon planning approach," in *the 4th International Conference on Computational Logistics (ICCL'13)*. Copenhagen, Denmark: Springer, 2013, pp. 128–142.

[16] T. Schouwenaars, r. Fron, and J. How, "Safe receding horizon path planning for autonomous vehicles," in *the 40th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, 2002, pp. 295–304.

[17] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for BerthaA local, continuous method," in *the IEEE Intelligent Vehicles Symposium (IV)*, Michigan, USA, 2014, pp. 450–457.

[18] A. Broggi, P. Cerri, S. Debattisti, M. C. Laghi, P. Medici, M. Panciroli, and A. Prioletti, "Proud-public road urban driverless test: Architecture and results," in *the IEEE Intelligent Vehicles Symposium (IV)*, Michigan, USA, 2014, pp. 648–654.

[19] T. Gu, J. M. Dolan, and J.-W. Lee, "On-road trajectory planning for general autonomous driving with enhanced tunability," in *Intelligent Autonomous Systems*. Springer, 2016, vol. 13, pp. 247–261.

[20] A. Y. Hata, F. S. Osorio, and D. F. Wolf, "Robust curb detection and vehicle localization in urban environments," in *the IEEE Intelligent Vehicles Symposium (IV)*, Michigan, USA, 2014, pp. 1257–1262.

[21] A. Hata and D. Wolf, "Road marking detection using LIDAR reflective intensity data and its application to vehicle localization," in *the 17th International IEEE Conference on Intelligent Transportation Systems*, Qingdao, China, 2014, pp. 584–589.

[22] S. Vera, J. Cobano, G. Heredia, and A. Ollero, "Rolling-horizon trajectory planning for multiple UAVs based on pseudospectral collocation," in *the International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, USA, 2014, pp. 516–523.

[23] M. Boudali, R. Orjuela, and M. Basset, "A Comparison of Two Guidance Strategies for Autonomous Vehicles," in *the IFAC 2017 World Congress*, Toulouse, France, 2017, pp. 12 539–12 544.