



**HAL**  
open science

## Continuous curvature planning with obstacle avoidance capabilities in urban scenarios

David González Bautista, Joshue Pérez Rastelli, Ray Lattarulo, Vicente Milanés, Fawzi Nashashibi

► **To cite this version:**

David González Bautista, Joshue Pérez Rastelli, Ray Lattarulo, Vicente Milanés, Fawzi Nashashibi. Continuous curvature planning with obstacle avoidance capabilities in urban scenarios. 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), Oct 2014, Qingdao, China. hal-01086888

**HAL Id: hal-01086888**

**<https://inria.hal.science/hal-01086888>**

Submitted on 25 Nov 2014

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

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

# Continuous curvature planning with obstacle avoidance capabilities in urban scenarios

David González, Joshue Pérez, Ray Lattarulo, Vicente Milanés and Fawzi Nashashibi

**Abstract**—This paper presents a continuous curvature planning algorithm with obstacle avoidance capabilities. The automated system generates a collision free path that considers vehicle's constraints, the road and different obstacles inside the horizon of view. The developed planning module was integrated in the RITS (former IMARA) autonomous vehicle architecture. The goal of this module is to obtain an accurate, continuous and safe path generation, by implementing parametric curves. To this end, a continuous curvature profile when calculating vehicle trajectory is introduced. It also permits to generate different speed profiles, improving the comfort by reducing lateral accelerations in the driving process. These algorithms have been implemented in simulated -ProSiVIC- and real platforms -Cybercars- showing good results in both cases. This approach is currently being implemented in the framework of the EU CityMobil2 project.

## I. INTRODUCTION

The development of autonomous driving systems is strongly related to technological advances to get a safety, efficient and comfortable road transportation. Recent studies show that autonomous vehicles would have a higher level of acceptability in our society [1]. When developing driverless cars, one can breakdown the algorithms in different ones according to the driving area (highway, urban environment, mixed traffic corridor) [2]. The complexity of the control system in each scenario is highly related to the traffic, infrastructure and safety characteristics demanded by the users. Due to this, traffic in congested urban areas is one of the most difficult tasks to manage.

Some Advanced Driver Assistance Systems (ADAS) show improvements for road monitoring and partial control of transportation systems, such as: emergency braking [3], keeping inter-vehicle safe distance, automated parking [4], obstacle avoidance and so on. Moreover, many platforms have been tested in real urban scenarios and highways (v.g.: Google driverless cars, DAIMLER's automated vehicle, PROUD platform from Parma University [5], [6] etc), showing significant progresses in the field. However, some unsolved challenges remains on dynamic trajectory generation, collision avoidance, comfort and safe driving.

The trajectory planning and control of autonomous vehicles has to mainly consider driving laws and road layout (number of lanes, pedestrians, cyclists and so on)[7], [8]. Among the different dynamic trajectory generation algorithms, Bézier curves are a suitable candidate in terms

of computation efficiency for real-time applications. Some authors have previously implemented them in path planners with anti-collision behavior [9]. They estimated trajectories based on laser data. The simulation showed that Bézier curves are a good solution for the curvature continuity in urban scenarios. However, passengers comfort with different speed profiles was not considered.

Digital navigation maps can contain errors or unexpected situations, like accidents or blocked streets. In Zinoune et al.[10], a detection of map errors for autonomous vehicles, based on embedded sensors, was presented. They use a trajectory generation that makes a statistical comparison with geometric data in the maps. This approach allows the improvements of the trajectories based on local information. Recently, other authors have proposed online maneuvers generation for path planning and obstacle avoidance [11]. Other path planning methods as state lattice planner, predictive constraint-based planning and spline-based have been proposed [11]. This paper presents a trajectory planning for an accurate, continuous and safe path generation, by considering vehicle and infrastructure limits and obstacles inside the horizon of view. This approach is based on parametric curve generation (Bézier polynomials) for a comfortable driving, accounting for lateral acceleration. It permits to determine a dynamic longitudinal control according to the type of the route.

The paper is organized as follows: A curvature profile optimization for different intersections is presented in Section II. Section III presents the obstacle avoidance module, which is capable to plan an obstacle free path. This module also considers the curvature to set the speed profile in the path, explained in Section IV. The validation of the algorithms is carried out in Section V, using simulated environments and real platforms.

## II. MOTION PLANNING APPROACH

RITS previous works [12], [13] proposed a control architecture comprising six main stages: acquisition, perception, communications, decision, control and actuation. This paper is mainly focused on the decision stage where the path generation is located.

The decision stage hosts the motion planning, which coordinates two main tasks: the global and the local planner. The first one takes into account the location of the vehicle, its destination and a map from the database for planning the route. This raw route is modified in the local planner module, that is responsible for smoothing the raw path, turning it into

Authors are with the Robotics and Intelligent Transportation Systems (RITS) Team, Inria Paris-Rocquencourt, Domaine de Voluceau, 78153 Le Chesnay, France david.gonzalez-bautista, joshue.perez.rastelli, vicente.milanes, fawzi.nashashibi@inria.fr

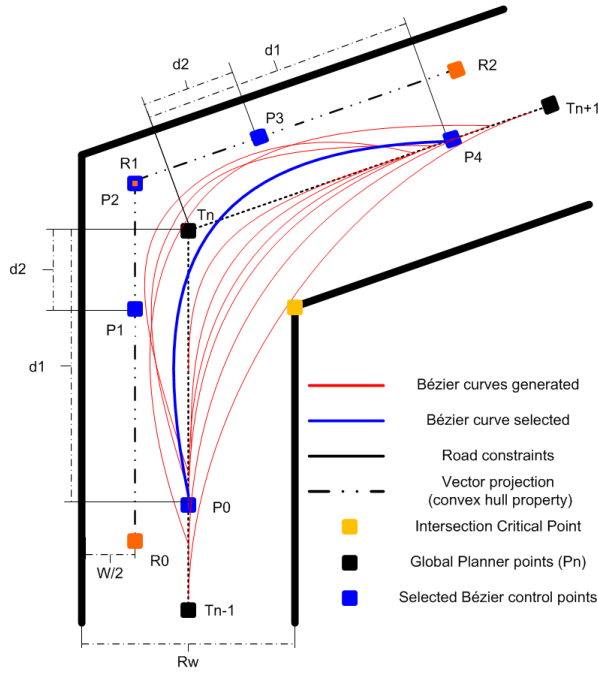


Fig. 1. Intelligent intersection curve generation

a drivable trajectory that considers the road constraints and the vehicle's limitations.

#### A. Intelligent intersection handling

The local planner manages turn stretches and intersections with Bézier curves, taking into account the route generated by the global planner ( $P_n$  points) [13]. It also includes constraints of the road (critical intersection points) and the vehicle limitations ( $W$  = vehicle width, maximum vehicle turning radius), as shown in Figure 1.

A geometrical approach considering the joining of straight and curved segments was carried out. The generation of the Bézier curve is as shown in Equation 1.

$$\mathbf{B}(t) = \sum_{i=0}^n P_i \binom{n}{i} (1-t)^{n-i} t^i \quad t \in [0, 1] \quad (1)$$

where  $n$  is the degree of the polynomial equation,  $P_i$  are the control points that define the curve.

To achieve a continuous curvature profile, a weighting algorithm has been implemented. Its computation gives the best parametric curve to be set in the intersection planning, achieving continuity between straight and curved segments.

The proposed approach generates several control points, evaluating the curvature profiles at the beginning and end of the curvature (giving each a weight in terms of the continuity), and comparing the maximum curvature with the maximum possible realizable by the vehicle (discarding the ones physically impossible to achieve). Moreover, the measure of the distance between the curve segments and the intersection critical point is also considered.

The curvature evaluation is performed as follows:

$$C(t) = \frac{\vec{Q}'(t) \times \vec{Q}''(t)}{\|\vec{Q}'(t)\|^3} \quad (2)$$

Equation 2 presents the curvature  $C(t)$  of a parametric curve. A 4th degree Bézier curve is implemented if the 3rd degree generated does not meet the requirements established (see Figure 1). The curve generation is defined as follows:

- The points  $R_0$ ,  $R_1$  and  $R_2$  describe the separation with respect to the external band of the road (with  $W/2$  of separation), where the internal control points will be set (i.e.  $P_1$ ,  $P_3$ , and  $P_2 = R_1$ ). This implementation is generated within its convex hull (see [13]), forcing the Bézier curve to fit the external road constraints.
- The maximum curvature in the Bézier curve must be realizable by the vehicle.
- The generated curves must fit the internal road constraints, i.e. all the curves have to be at least  $(W/2mt/s)$  away from the intersection critical point, inside the lane.

The location of all points in a 4th degree Bézier curve is described by the following equations:

$$P_0 = R_1 + d_1 \frac{T_{n-1} - T_n}{\|T_{n-1} - T_n\|} \quad (3)$$

$$P_1 = R_1 + \left( d_2 + \sqrt{\|T_n - R_1\|^2 + \frac{R_w - W}{2}} \right) \frac{R_0 - R_1}{\|R_0 - R_1\|} \quad (4)$$

$$P_3 = R_1 + \left( d_2 + \sqrt{\|T_n - R_1\|^2 + \frac{R_w - W}{2}} \right) \frac{R_2 - R_1}{\|R_2 - R_1\|} \quad (5)$$

$$P_4 = R_1 + d_1 \frac{T_{n+1} - T_n}{\|T_{n+1} - T_n\|} \quad (6)$$

where  $d_1$  and  $d_2$  are the separation distances between  $P_0$  and  $T_n$  and between  $P_4$  and  $T_n$  respectively (see Figure 1).  $R_w$  is the road width and  $W$  is the vehicle width. The point  $P_2$  will be fixed at the location of  $R_1$ .

3rd degree Bézier curve control points definition is done as follows: Equations 3 and 4 will remain the same. Equation 5 will calculate point  $P_2$  and Equation 6 will calculate point  $P_3$ . Then, the Bézier curves are evaluated, choosing the most suitable one (blue curve in Figure 1).

#### B. Adjacent consecutive intersections

The presented trajectory generation is able to plan turns and intersections. However, it was limited by the amount of space needed between two different consecutive curves to generate a continuous path. These close points (defined in the global planning) are a common situation in urban scenarios (especially at downtown), where the roads are not homogenous. A good approach of adjacent consecutive algorithm for this kind of situations is proposed in this section.

The limiting problem is due to the consecutive joint points of two (or more) curves, since both have a curvature profile different than zero. In previous works [14], [15], a study on the curvature continuance for Bézier and parametric curves

has been applied by considering the derivative of the curve at the junction point. The study of the Bézier curvature permits to find the best place to set the control points in order to have  $C^1$  continuity. To achieve this, it suffices to ensure that the ending tangent vector of the first curve is identical to the first tangent vector of the second curve, as in Equation 7.

$$\vec{Q}'_{(1)} = m(C_m - C_{(m-1)}) = \vec{R}'_{(0)} = n(D_1 - D_0) \quad (7)$$

where  $m$  and  $n$  are the order of each curve,  $C_0, C_1, \dots, C_m$  are the control points of  $Q_{(t)}$  and  $D_0, D_1, \dots, D_n$  are the control points of  $R_{(t)}$ . Since  $m$  and  $n$  are fixed, the relation depends on the distance between the mentioned control points.

In order to handle different adjacent intersection points (closer than the previous definition of fixed positions at  $20mts$ , as in [12]), the planner considers the previous  $Pg_{(n-1)}$  and the next  $Pg_{(n+1)}$  intersection points, evaluating the need for a joint between two curves, and if so, applying the previous mathematical procedure.

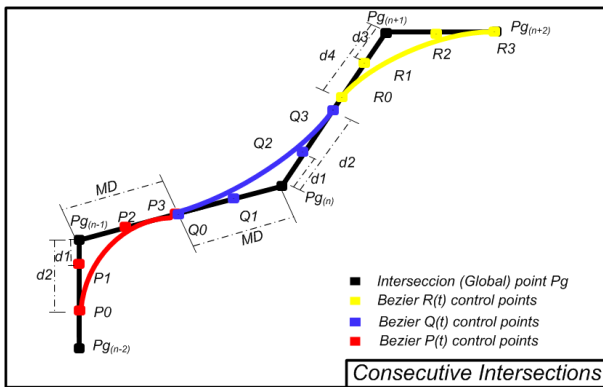


Fig. 2. Solution for consecutive intersections

Figure 2 shows three adjacent Bézier curves. In order to achieve  $C^1$  continuity in the path, the local planner takes into account the space between each intersection point ( $Pg_n$ ).

First, the intersection distances  $Pg_{n-1}Pg_n$  and  $Pg_nPg_{n+1}$  are calculated; if the control points maximum distance (MD) set in the planner is bigger than half the minor intersection distance (minor of  $(Pg_{n-1}Pg_n)/2$  and  $(Pg_nPg_{n+1})/2$ ) a junction is required.

Therefore, the minor of the calculated distances is MD. In Figure 2 is shown when  $Q_{(t)}$  (blue curve) needs to adjust to  $P_{(t)}$  (red curve) and  $R_{(t)}$  (yellow curve), being  $(Pg_{n-1}Pg_n)/2$  the constraint distance for the Bézier generation. Once the minor distance is defined, the segmentation process of the Bézier curve is done as in Section II.A.

### III. OBSTACLE AVOIDANCE MODULE

An obstacle avoidance module has been developed as part of the local planner (see Figure 3). This module is able to avoid static obstacles in the path by taking into account the trajectory already defined, the different obstacles from the perception stage, the road constraints and the vehicle limitations.

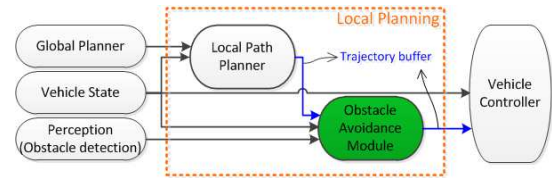


Fig. 3. Obstacle avoidance module added to the local planning

A buffer was set in the control architecture, as presented in [12]. It communicates the local planner with the obstacle avoidance module, and then the vehicle control as in Figure 3. This feature modifies the trajectory planned with a new one if needed by flushing the buffer.

To manage the obstacle avoidance, we take into account the current planned path, and if no obstacles are found, it bypasses the path information onto the controller. When obstacles are detected, an avoidance route is planned. The previous path is held until the maneuver is finished. This module (as well as the local planner), will plan a trajectory until the horizon of view of the sensors (set at  $30mts$  [13]).

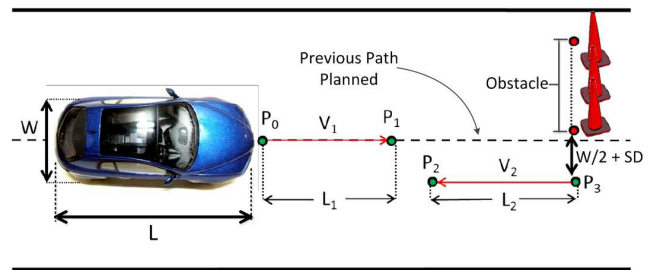


Fig. 4. Bézier control points generation in case of an obstacle avoidance maneuver (Beginning stage)

The obstacle avoidance maneuver is done in three main phases (beginning, middle, end), as in [16], which are described as follows:

- **Beginning:** First, the algorithm sets the control points of a new 3rd degree Bézier curve that redirects the vehicle into an obstacle free path. It triggers the beginning of the obstacle avoidance maneuver as in Figure 4. The starting point of the curve ( $P_0$ ) is in the front of the vehicle in the predefined path. Point  $P_3$  is placed beside the obstacle, with a distance of half the vehicle width plus a safe distance ( $W/2 + SD$ ). The algorithm chooses the position of the control points, taking into account the availability to put the end points of the curve in the same lane, either it is on the left or right side. Then, the minor curvature possible is determined by applying the previous segmentation described in Section II.A., to the inner control points of the curve (i.e.  $P_1$  and  $P_2$ ).
- **Middle:** Once the vehicle is in an obstacle free path, the middle phase ensures that the obstacle is overtaken. The length of the vehicle ( $L$ ) and also the length of the obstacle are taken into account to achieve this. In this

phase a straight segment is planned following the vector  $L_2$  from Figure 4. This consideration assures a trajectory continuity between the *Beginning* and the *Middle* phase of the maneuver.

- *End*: This phase is performed symmetrically as the *Beginning* phase. The control points are set considering that the vehicle is able to return to the previous path. This phase, sets the starting control point in the front part of the vehicle ( $P_0$ ), and the ending point ( $P_3$ ) at a distance  $D$  (defined in 16m/s). The segmentation process of the curvature is generated as in II.A.

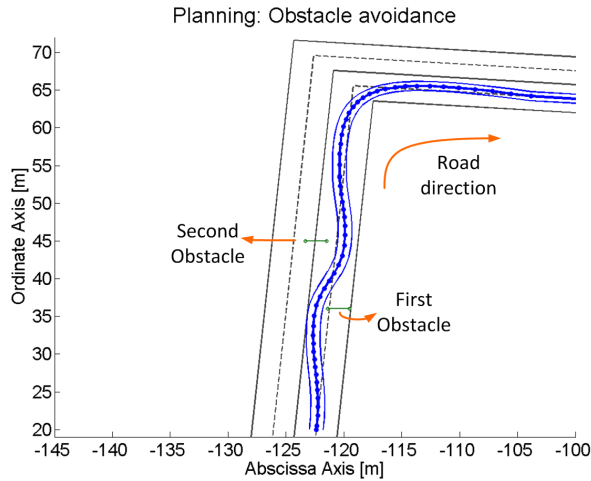


Fig. 5. Obstacle Avoidance with continuous curvature

Each phase can be recalculated at any point of the obstacle avoidance maneuver. If a new obstacle is found further ahead, the proposed method is able to redefine a new trajectory. Every time that an obstacle is found, the maneuver is restarted from the current location of the vehicle. Figure 5 presents the planning implemented when the vehicle follows the path, and obstacles are detected. The module considers this situation with the three phases previously stated, replacing the previous path and calculating a new one.

#### IV. LONGITUDINAL CONTROL IN CURVED SEGMENTS

Curvature continuity does not only affects the smoothness and safeness of the path, but also the comfort while driving through it. In order to set an adequate speed profile that ensures a comfortable travel for the passengers, our approach considers the curvature profile of the road.

Different levels of discomfort, associated to lateral and overall accelerations experienced by the passengers, have been set in the *ISO 2631-1 Standard* and implemented in previous works [17]. These levels are: *Not uncomfortable*, *A little uncomfortable*, *Fairly uncomfortable*, *Uncomfortable*, *Very uncomfortable*, *Extremely uncomfortable*. Different speed profiles can be generated taking into account these levels, and the lateral acceleration caused by the velocity in curved segments. The calculation of the overall r.m.s. acceleration acting on the seated human body is as presented by Equation 8:

$$a_w = \sqrt{k_x^2 a_{wx}^2 + k_y^2 a_{wy}^2 + k_z^2 a_{wz}^2} \quad (8)$$

where  $k_x = k_y = 1.4$ ,  $k_z = 1$ , are multiplying factors and  $a_{wx}, a_{wy}, a_{wz}$  are the r.m.s. accelerations in the different axis [17].

Since the vehicle operates in a two dimensional space, the  $a_{wz}$  component is zero; also, we can assume the longitudinal  $a_{wx}$  component as zero, because of the low speed profiles implemented in urban scenarios.

$$a_{wy} = V^2 C \quad (9)$$

Furthermore, the lateral acceleration component  $a_{wy}$  is as presented by Equation 9, where  $C$  is the curvature value of the evaluated segment and  $V$  is the velocity. Taking Equations 8, 9 and the previous considerations, we are left with Equation 10.

$$V_{max} = \sqrt{\frac{a_w}{k_y C}} \quad (10)$$

This last equation, considered in the *obstacle avoidance module*, sets a speed profile in the trajectory according to the curvature of each segment, and the level of comfort set by the user. Once the trajectory is planned in the *obstacle avoidance module*, it is sent onto the *vehicle controller module*.

#### V. VALIDATION TESTS

Different tests were carried out in simulated and real environments for the trajectory generation algorithm.

##### A. Simulated environments

The simulated experiments are developed in ProSivic<sup>1</sup>, a multi-sensory environment, and takes into account detailed models of real vehicles.

Two simulations are described. The first one is shown in Figure 6, which depicts the intelligent planning of an intersection. Different curvature profiles are calculated (blue lines), evaluated through a weighting algorithm as in Section II.A, and one is selected as the best suitable curve to apply to the current intersection (Red curve).

The second experiment is presented in Figure 7. It shows the performance of the vehicle following the given path (blue line), and implementing the obstacle avoidance module every time an obstacle is found (green lines), as described in Section III.

The versatility of the present approach allows to the planner to overtake every time an obstacle is found; regardless if the vehicle was already performing the maneuver (middle left of Figure 7) or passing through an intersection (upper left of Figure 7).

For the performance shown in Figure 7, different levels of comfort were taken into consideration to provide the speed suggestion presented in Figure 9: *Not uncomfortable*,  $a_{wy} = 0.315m/s^2$  (red line), *A little uncomfortable*,  $a_{wy} = 0.63m/s^2$  (gray line), *Fairly uncomfortable*,  $a_{wy} = 1m/s^2$

<sup>1</sup><http://www.civitec.com/>



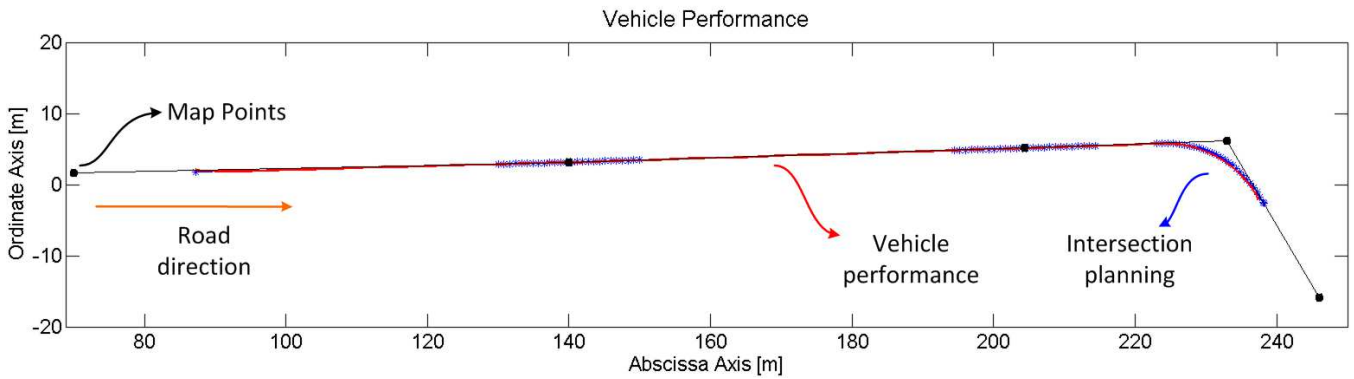


Fig. 8. Performance of our real platform -Cybercar- following the path planned

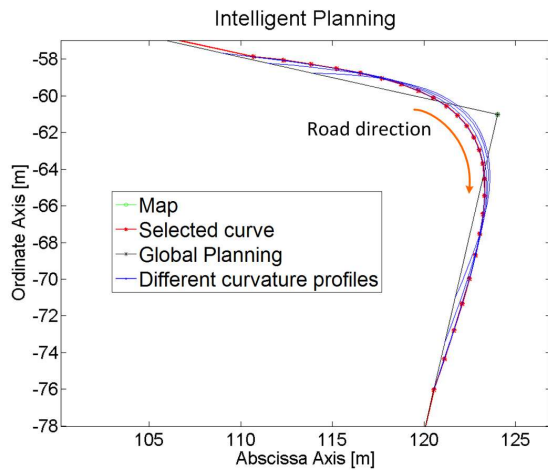


Fig. 6. Planning of different curvature profiles to find the best curve

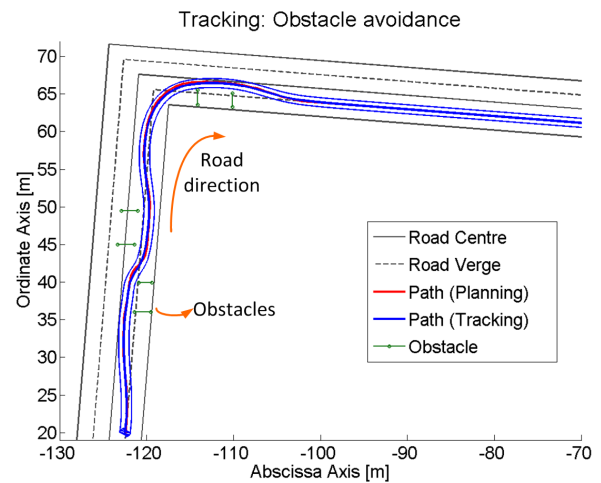


Fig. 7. Performance of the vehicle avoiding obstacles

(cyan line), *Uncomfortable*,  $a_{wy} = 1.6m/s^2$  (green line), *Very uncomfortable*,  $a_{wy} = 2.5m/s^2$  (blue line). This is done as in Section IV.

### B. Real Platforms

A prototype *Cybus* was used as validation platform. It is a drive-by-wire *Cybercar* vehicle with fully automation capabilities. This *Cybus* is equipped with four lasers, two in the front and two in the back (two near the bumper and two near the ceiling). The higher ones allow its localization by a SLAM method, while the lower ones permit the obstacle localization

Figure 8 shows the performance on the *Cybus*. It depicts the map (black points), the different intersection and turns (blue points), and the good performance of the vehicle while following the path planned (red line).

The path following generated at the intersection is accurate, where no jumps or fast changes occur in the vehicle performance; showing a smooth and safe behavior. The curvature implemented by the controller shows a smooth behavior in Figure 10 (blue line), where the maximum curvature achievable by the *Cybus* is shown with a red line (4.4 1/m). Moreover, Figure 11 shows the suggested speed

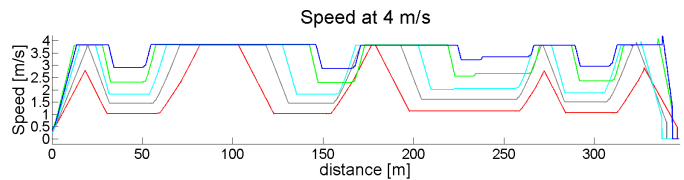


Fig. 9. Different speed profiles according to the levels implemented in [17]

profile in the intersection according to a *Not uncomfortable* level of comfort for the user. When the vehicle arrives to the curved segment, the suggested speed is lower in order to decrease the lateral accelerations and keep the *Not uncomfortable* level of comfort.

## VI. CONCLUSIONS AND FUTURE WORKS

Based on previous works, the development and implementation of different algorithms for a safe and comfortable path planning is presented. These are oriented to intersections and obstacles handling, where the main contribution to the RITS control architecture takes place in the local planning.

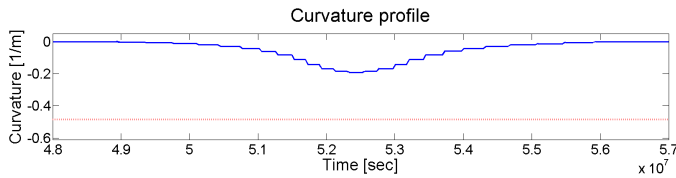


Fig. 10. Real curvature profile at the curved segment in Figure 8

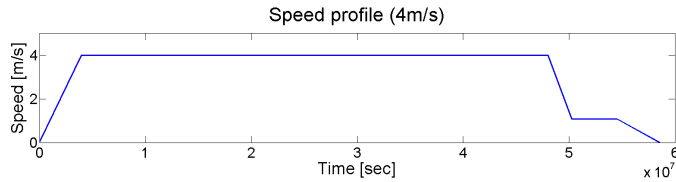


Fig. 11. Speed profile suggestion

Implementing parametric curves, different intersections are handled. By searching the position of the Bézier control points, a smooth path that fits the vehicle and road constraints can be planned (as explained in II.A).

The generation of these curves needed a space in between (20mts), in previous works. This is solved by taking into account the derivative of the parametric curve at its limits, where the position of the control points can be found in order to have a  $C^1$  continuity between adjacent intersections (as explained in II.B).

Furthermore, an obstacle avoidance module is included in the local planning stage of the architecture. This stage is capable to avoid static obstacles on the road, by defining an overtaking maneuver with three defined phases.

The curvature profile is considered to reduce the lateral accelerations. The user is able to set a level of comfort in the navigation system, as this modules calculates the longitudinal speed profile to meet the requirements of the user.

These algorithms have been validated in simulations (with ProSivic) and in real platforms (Cybus), showing good results in both platforms and giving proof of its modularity and versatility.

In future works, improvements in the obstacle avoidance capability of the vehicle will be done. Dynamic obstacles and different vehicle communications (V2V and V2I) will be considered to achieve a safer and more comfortable navigation.

## ACKNOWLEDGMENTS

Authors wants to thank to the ARTEMIS project DE-SERVE and FP7 CityMobil2 for its support in the development of this work.

## REFERENCES

- [1] J. H. Frank M. F. Verberne and C. J. H. Midden, "Trust in smart systems sharing driving goals and giving information to increase trustworthiness and acceptability of smart systems in cars," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 2012.
- [2] M. Van Schijndel-de Nooij, B. Krosse, T. Van den Broek, S. Maas, E. Van Nunen, H. Zwijnenberg, A. Schieben, H. Mosebach, N. Ford, M. McDonald, and J. Sanchez, "Definition of necessary vehicle and infrastructure systems for automated driving," *SMART 2010/0064 European Commission*, 2011.
- [3] R. Zheng, K. Nakano, S. Yamabe, M. Aki, H. Nakamura, and Y. Suda, "Study on emergency-avoidance braking for the automatic platooning of trucks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–10, 2014.
- [4] F. Ghilardelli, G. Lini, and A. Piazzini, "Path generation using mbi4-splines for a truck and trailer vehicle," *Automation Science and Engineering, IEEE Transactions on*, vol. 11, no. 1, pp. 187–203, Jan 2014.
- [5] S. Ingolfo and V. Silva Souza, "Law and adaptivity in requirements engineering," in *Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2013 ICSE Workshop on*, May 2013, pp. 163–168.
- [6] A. Broggi, P. Cerri, S. Debattisti, M. Laghi, P. Medici, M. Panciroli, and A. Prioletti, "Proudpublic road urban driverless test: architecture and results," in *Intelligent Vehicles Symposium (IV), 2014 IEEE*, June 2014, pp. 931–937.
- [7] K. Jo and M. Sunwoo, "Generation of a precise roadway map for autonomous cars," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 3, pp. 925–937, June 2014.
- [8] J. Nilsson, M. Ali, P. Falcone, and J. Sjöberg, "Predictive manoeuvre generation for automated driving," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, Oct 2013, pp. 418–423.
- [9] L. Han, H. Yashiro, H. Nejad, Q. H. Do, and S. Mita, "Bzier curve based path planning for autonomous vehicle in urban environment," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 1036–1042.
- [10] C. Zinoune, P. Bonnifait, and J. Ibanez-Guzman, "A sequential test for autonomous localisation of map errors for driving assistance systems," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 1377–1382.
- [11] D. Madas, M. Nosratinia, M. Keshavarz, P. Sundstrom, R. Philippsen, A. Eidehall, and K.-M. Dahlen, "On path planning methods for automotive collision avoidance," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 931–937.
- [12] D. Gonzalez and J. Perez, "Control architecture for cybernetic transportation systems in urban environments," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 1119–1124.
- [13] J. Perez, R. Lattarulo, and F. Nashashibi, "Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles," in *Intelligent Vehicles Symposium (IV), 2014 IEEE*, 2014.
- [14] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *Robotics, IEEE Transactions on*, vol. 26, no. 3, pp. 561–568, 2010.
- [15] D. Walton, D. Meek, and J. Ali, "Planar  $G^2$  transition curves composed of cubic bézier spiral segments," *Journal of Computational and Applied Mathematics*, vol. 157, no. 2, pp. 453–476, 2003.
- [16] J. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro, "Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 438–450, Sept 2008.
- [17] L. Labakhua, U. Nunes, R. Rodrigues, and F. Leite, "Smooth trajectory planning for fully automated passengers vehicles: Spline and clothoid based methods and its simulation," in *Informatics in Control Automation and Robotics*. Springer Berlin Heidelberg, 2008, vol. 15, pp. 169–182.