



**HAL**  
open science

# Trajectory generator for autonomous vehicles in urban environments

Joshué Pérez Rastelli, Jorge Godoy, Jorge Villagra, Enrique Onieva

► **To cite this version:**

Joshué Pérez Rastelli, Jorge Godoy, Jorge Villagra, Enrique Onieva. Trajectory generator for autonomous vehicles in urban environments. IEEE ICRA - IEEE International Conference on Robotics and Automation, May 2013, Karlsruhe, European Union. hal-00789760

**HAL Id: hal-00789760**

**<https://hal.inria.fr/hal-00789760>**

Submitted on 18 Feb 2013

**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.

# Trajectory generator for autonomous vehicles in urban environments

Joshué Pérez<sup>1</sup>, Jorge Godoy<sup>2</sup>, Jorge Villagr a<sup>2</sup>, Enrique Onieva<sup>3</sup>

**Abstract**—Nowadays, some developments in the vehicle industry permit a safe and comfortable driving. However, several manufactures and research groups are still working in the improvement of the control strategies and path smoothing algorithms. In this paper, a new trajectory generation approach for autonomous vehicles in urban scenarios, considering parametric equations, is proposed. An algorithm that considers Bezier curves and circumference parametric equations for a real vehicle, specifically in roundabout and urban intersections is presented. This approach is generated in real time and can be adapted to dynamic changes in the route. A smooth trajectory generator computationally efficient and easily implementable is proposed. Moreover, this new trajectory generator reduces the control actions, generated with to a fuzzy controller. Some trials have been performed in an urban circuit with promising performance.

## I. INTRODUCTION

In the last decade, several improvements have been implemented in commercial vehicles. Moreover several research partnership have been working on different ADAS in real vehicles [1]. These implementations are focused in specific scenarios, monitoring and some partial control systems, such as: Assisted parking, Adaptive Cruise Control (ACC), Pre-crash systems, among others. So far, only partial executions on driverless vehicles are possible. In urban scenarios, a whole automation of a fleet of vehicles is a circumstance that vehicle manufacturers and governments recently defined as unaffordable in a near future[2].

In the Intelligent Transportation Systems (ITS) field, trajectory planning is one of the most studied topics. The Laboratory on Interactions between Vehicles, Infrastructure and Drivers (LIVIC) in France, have presented many advances in algorithms for adapting traffic on an urban scenarios such as highways [3]. A first stage defines the feasible maneuvers considering environment conditions, and a second stage evaluate the possible trajectories.

Recently, path-smoothing algorithms have been studied in the literature [4], [5], [6]. In Labakhua et al. [4] the authors consider smooth trajectories with low associated accelerations, introducing a velocity planning stage into the

interpolating curve planners. They studied three methods: cubic spline interpolation, trigonometric spline interpolation, and a combination of clothoids, circles and straight lines. All these solutions have good performance in simulation with a kinematics model of a vehicle [7].

Recent studies review the characterization of complex roads considering the GPS information [8]. The development of much more accurate maps is one of the new challenges of the ADAS regarding to road safety applications. Consider, for instance, the concept of enhanced maps (Emap) [8], where digital maps exploit the definition of clothoids to follow the road shape. However, so far, roundabouts were unconsidered because, as the authors highlight, these concrete scenarios deserve special attention.

Different authors consider two categories of curves in the generation of continuous-curvature path: parameterized by the curve's arc length (i.e. clothoids) and closed-form expression of position (i.e. parametric curves) [9], [5]. The clothoids have been used many times in trajectory generation, due to smoothing and the curvature grows with the distance from the origin. It allows a generation of real scenarios, in particular, in railways and highways [10]. Clothoids are also used to know a priori which is the shortest path, guarantying lateral accelerations and steering wheel speeds below given pre-set thresholds [6]. Regarding the use of path tracking for autonomous vehicles, the main advantage of clothoids is that they guarantee the comfort of passengers, since it guarantees a continuous curvature profile, which can be set by different algorithms (e.g. [9]).

However, the main drawback of clothoids is that they do not have an analytical solution; consequently the trajectory generation requires a non negligible computational cost, since they require Fresnel integrals to be computed. Moreover, in urban environment -with frequent turns 90 degrees bends- the trend is to use clothoid-circle-clothoid [8], [11] concatenations, which results in a more complex tracking of the generated trajectory. For this reason, others authors have used others metrics for real time implementations, such as splines [12], or Bezier Curves [5].

Among all the techniques used in path planning, Bezier curves have the advantage of simplicity, curvature continuity, and computational efficiency due to its closed form expressions [13]. Several works have been done in this aim, for instance, the Toyota Technological Institute have tested Bezier curve based path planner in a single-seater electronic vehicle with an anti-collision system [11]. Here, the authors generate an alternative path for avoiding the obstacle, however, the curvature, jerk or lateral acceleration have been despised in the path generation. Choi et al. explain an approach based

<sup>1</sup>Joshu e P erez is with IMARA team at INRIA research center, Paris - ROCQUENCOURT, France e-mail: joshue.perez\_rastelli@inria.fr

<sup>2</sup>The authors Jorge Godoy and Jorge Villagr a are with the Industrial Computer Science Department, Centre of Automation and Robotics (CAR), Universidad Polit cnica de Madrid - Consejo Superior de Investigaciones Cientificas, (UPM-CSIC), La Poveda- Arganda del Rey, 28500 Madrid, Spain e-mail: (jorge.godoy, jorge.villagra)@csic.es

<sup>3</sup>Enrique Onieva is with the Deusto Institute of Technology (DeustoTech), University of Deusto, Bilbao 48007, Spain email: enrique.onieva@deusto.es

on Bezier curves for the path planning algorithm as an alternative to cubic splines, using simulations of the dynamic in a differential wheeled robot [13]. Others authors, as Yang and Sukkarieh [5], have simulated continuous-curvature path for mobile robots. They have developed an algorithm based on parametric cubic Bezier curves, which satisfies both the curvature continuity and maximum curvature constraints. Only four points are required for a cubic Bezier curve generation, besides the algorithm guaranties the smoothness with junctions points among different curves and straight segments [5].

The comfort driving in autonomous vehicle mainly depends on a good reference trajectory generation and a fine controller tuning. This last stage has to track as close and comfortably as possible the path generated in the first stage. Considering these premises and the AUTOPA program experience in control of autonomous vehicles [14], [15], [16], [17], [6], a new guidance system based on straight stretches, Bezier curves, circumference parametric equations and fuzzy logic controllers is proposed in this work. The whole system permits a better tracking in urban scenarios, considering a comfortable driving and a continuous path smoothing between the different segments. In brief, the different phases of this work and its more important features are listed below:

- 1) A good trajectory generation considering Bezier curves fitting the problem waypoints smoothly guaranteeing the curvature continuity all along the turn and continuous concatenation with straight segments reference.
- 2) The tracking points are generated in real time considering continuous curvature.
- 3) The trajectory generator bounds the discontinuities between the lineal and curves segments, since it force the Bezier control point straight segment.
- 4) Moreover, geometric continuous equations are used for urban roundabouts: entry, inside and exit.
- 5) A fuzzy logic controller have been tuned, based on [17] for the new tracking point generation.

The guidance system described in this work is computational efficient and easy to implement in real vehicles (electric Citroen van). The path generated guarantees the vehicle within the road. Moreover, this system permits to reduce the longitudinal speed while the curvature generated affect the comfort of the driving.

The rest of the paper is organized as follows: section II explains the different level of the control architecture. The continuous path generation based on Bezier curves, and the fuzzy controller implemented to track it, are explained in section III and IV respectively. Finally, the trials and conclusions of this work are presented in sections V and VI, respectively.

## II. CONTROL ARCHITECTURE

In this section the control architecture proposed is explained (figure 1). The structure is based on the previous works developed for the AUTOPA program [16], [17]. Following the directives presented in [18], the modularity is a priority in the design of each module, permitting to exchange

blocks according to different conditions and situations. The control architecture is organized in four stages: acquisition, localization, decision and action.

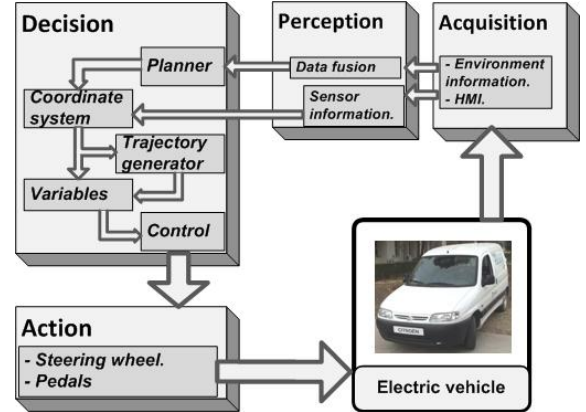


Fig. 1. Control architecture proposed

### A. Acquisition and localization

The acquisition stage receives the information from the environment. In straight segments the vehicle follows a reference line, but when it detects a curve or a roundabout, through the infrastructure or the previous information of the maps, the trajectory generator block – in the decision stage – is activated. The environment information and the Human Machine Interface module reads the maps to follow in the route. The maps are composed of the UTM coordinate of the middle curves, intersection and roundabouts. The route information, traffic signal, exits or blocked paths is previous known as in [19]. The HMI interface allows the user to change some parameters in the driving, as well as: lane-change, loading another route, aborting the autonomous driving, among others.

The localization stage receives the information from the acquisition stage. It uses the positioning system based on a Differential GPS (DGPS), with a maximum error of 50 cm, located over the rear axle of the vehicle, and a Inertial Measurement Unit (IMU), described in [20]. Moreover, a communication system with a central station and other vehicle is used [21]. Finally, a RFID sensor network for traffic signal detection and identification of emergency situations have been implemented [22].

### B. Decision

The decision stage uses the information in real time from the localization stage. The main contributions of this new approach are focused in trajectory generator and control modules, which are explained in the next two sections. The rest modules are described as follow:

- **Planner:** this module reads the route from a XML file in the environment information module. The first two points are used to calculate a start straight segment. The *Planner* sends this information to the *Coordinate system* module, which estimates dynamic maps for

specific scenarios. e.g.: a roundabout is considered as an intersection with constant radius. Finally, this module fits the reference speed of the vehicle, but external signal for change this speed in each stretch are also considered.

- **Coordinate system:** this module translate the UTM coordinate from the GPS to local position coordinates, where the beginning of each stretch is the new Cartesian zero, and the end is the size of the same stretch. This module permits to calculate the control variables.
- **Variables:** this module estimates the variables to the control according to the coordinate system. Control is based on the previous developed in our group [17], but adjusting the parameters for this new trajectory generation. The variables are: the angular error (in degrees) and the lateral error (meters) for the steering wheel position, and the Distance to the bend (meters) and the actual velocity (km/h) for the angular speed steering wheel.

### C. Action

The action stage concerns three peripherals connected to the steering bar, the throttle and brake systems respectively. The first one is controlled by a DC motor with a gear reduction fixed to the vehicle steering bar. The low level control uses an external PID, where the control loop is closed by an optical encoder in the motor axle [18].

The throttle pedal is connected through a digital-analogue card. Finally, a hydraulic braking system has been connected in parallel with the original brake system of the vehicle. The pressure is controlled through the same digital-analogue card [22]. Three switches are used to commute between the original mode (manual driving) and the autonomous driving mode.

## III. TRAJECTORY GENERATOR

The main aim of this work is to propose a better path generation for the lateral control in autonomous vehicles considering parametric equations coupled to straight segments in urban environments. Collision avoidance functions are not considered in this work.

This module is activated when the vehicle enters in a curve or a roundabout. For straight segments, two continues point are used to generate the reference path. Afterwards, these points are used to generate the trajectory. For curve segments, a Bezier curve generation has been used.

### A. Bézier curves

The Bézier curve of degree n is represented as follow:

$$B(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i * P_j \quad (1)$$

where t is an parameter  $\in [0,1]$ ,  $B(t)$  are the control point generated and j the number of control point used (4). Figure 2 shows an example of Bezier curves generation in an intersection with 10 points. The control points of Bezier curves ( $P_0, P_1, P_2$  and  $P_3$ ) are calculated in the *Coordinate*

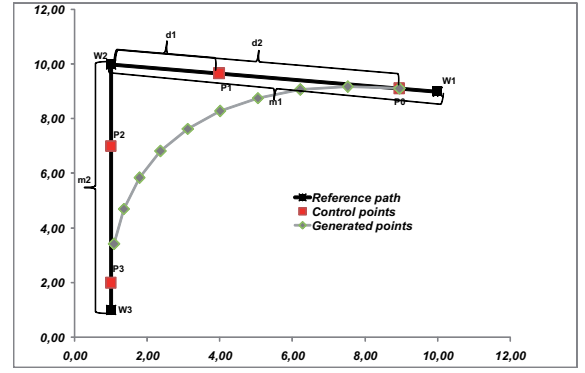


Fig. 2. Path generation with Bezier curves

*system* module, considering the reference path given in the *Planner* module ( $W_1, W_2$  and  $W_3$  in Fig. 2), as follow:

$$P_0 = W_1^j + (W_2^j - W_1^j) * (m_1 - d_1)/m_1 \quad (2)$$

$$P_1 = W_1^j + (W_2^j - W_1^j) * (m_1 - d_2)/m_1 \quad (3)$$

$$P_2 = W_2^j + (W_3^j - W_2^j) * d_2/m_2 \quad (4)$$

$$P_3 = W_2^j + (W_3^j - W_2^j) * d_1/m_2 \quad (5)$$

where  $d_1$  and  $d_2$  are fixed in 3 and 8 meters for urban intersections, and  $m_1$  and  $m_2$  are the distance between the points  $W_1 - W_2$  and  $W_2 - W_3$ , respectively. These values were estimated considering the dimension of our vehicle (1.75 meters of width) and the middle point ( $t=0.5$ ) of the Bezier curve generation. The subscript j indicates the coordinate calculated in the curve. Considering these four points, a continuous cubic Bézier curve path is obtained as follows:

$$B_t = P_0*(1-t)^3 + 3*P_1*t*(1-t)^2 + 3*P_2*t^2*(1-t) + P_3*t^3 \quad (6)$$

### B. Roundabout scenario

Once a roundabout is detected, the first step of the trajectory generator is to determine the adequate entering path for the vehicle. For this, a Bézier curve to couple the entering lane with the roundabout is generated; where the intersection between the roundabout (external lane) and the entering road define the control points. To simplify the process, it is assumed that the entering lane is straight. From this data, control points  $P_0$  and  $P_1$  are calculated using equations 2 and 3; where  $W_1$  is the starting point of the entering lane and  $W_2$  the intersection one. On the other hand,  $P_2$  and  $P_3$  are defined as the points located  $d_1$  and  $d_2$  meters away from the intersection point, over the roundabout trace.

After the entering path has been defined, the trajectory over the roundabout is calculated by the circumference's equations, increasing the angle in 0.1 radians per point (starting from  $P_3$ ). This procedure is repeated until the distance to the desired exit is less than  $d_2$ ; when this threshold is reached, the exit path is obtained in a mirrored way to the

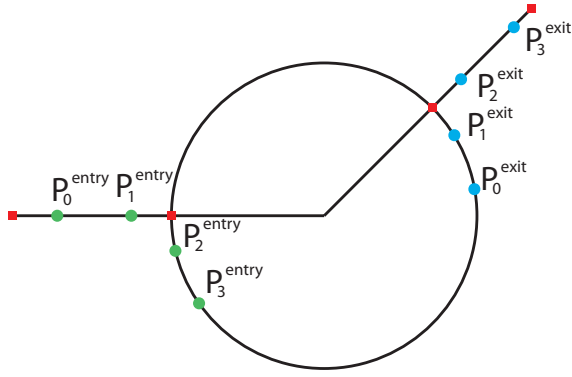


Fig. 3. Estimation of control points for entry /exit the roundabout.

entering one. Figure 3 shows the control points used for the estimation Bézier curves in the entrance and exit of the roundabout.

#### IV. FUZZY CONTROLLER

To perform the lateral control of the vehicle, two main controllers have been implemented. The first one was already described in [17] and it is used when the vehicle is over a straight segment or a single curve. The second one is an evolution of the first controller and it has been adjusted to control the vehicle trajectory inside the roundabouts. In this sense, a higher weight to those rules that keep the steering wheel turned to the left is necessary. This consideration is only valid in those environments where the traffic is right-handed. The input and output membership functions and were adjusted considering the less lateral error (of both variables), compared with human driver. Fig. 4 shows the membership functions defined for each input. Below, a detailed description of these functions is presented.

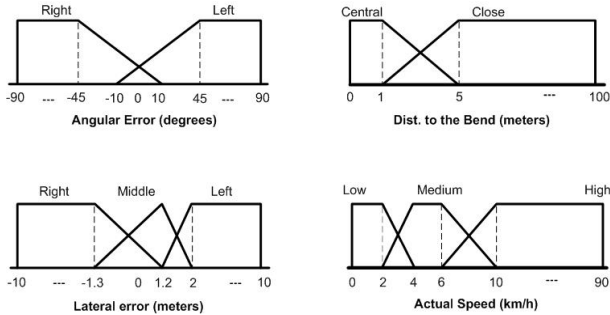


Fig. 4. Membership functions for the lateral control in roundabouts

- **Angular error:** Only two crossed membership functions are defined (upper-right image of Fig. 4). The central label has been removed since it contributed to make the system unstable when the angle to the reference line was close to zero.
- **Lateral error:** In this case, the three original membership functions are used. However, their ranges have been reduced and translate to the right – the middle label has been closed and set over 1.2 meters. This change would

help to keep the steering wheel turned to the left while the vehicle is inside the roundabout.

- **Distance to the bend:** Despite a roundabout could be considered as a constant bend in the road, and then a distance to it could not be formally defined, the name of this input variable has been kept beyond it is used for the distance to the next trajectory point. For this input, the range has been reduced and only two membership functions are defined. This input is very important for lane-change inside the roundabouts since the label *Close* would increase the steering wheel mobility.
- **Actual speed:** All membership functions defined in the first controller are kept. However, the range of this input has been reduced. The reason of this change is that the vehicles' speed inside a roundabout is lower than the normal speed in the rest of the urban environment.

Finally, the fuzzy rule set for the steering wheel position has been defined as:

**IF Angular error Left THEN Steering Pos Right**  
**IF Angular error Right THEN Steering Pos Left**  
**IF Lateral error Left THEN Steering Pos Right**  
**IF Lateral error Right THEN Steering Pos Left**  
**IF Lateral error Middle AND Angular error Left THEN Steering Pos HalfRight**  
**IF Lateral error Middle AND Angular error Right THEN Steering Pos HalfLeft**

Rules has been combined when the *Lateral error* is middle in order to reduce the control action according to the output values for this controller: Left (-1), HalfLeft(-0.5), HalfRight(+0.5) and Right(+1). For the second output – Angular Speed – the fuzzy rules set have been defined as:

**IF Dist. to bend Central AND Actual Speed Low THEN Angular speed Low**  
**IF Dist. to bend Central AND Actual Speed Medium THEN Angular speed Medium**  
**IF Dist. to bend Central AND Actual Speed High THEN Angular speed Med\_high**  
**IF Dist. to bend Close AND Actual Speed Low THEN Angular speed Medium**  
**IF Dist. to bend Close AND Actual Speed Medium THEN Angular speed Med\_high**  
**IF Dist. to bend Close AND Actual Speed High THEN Angular speed High**

where the output singletons are the same used in the first controller: Low (0.4), Medium (0.6), Med.high(0.8) and High(1).

#### V. EXPERIMENTS

##### A. Test scenario

All the experiments were performed at the CAR's private driving circuit (figure V-A) where a roundabout has been recently builded. The roundabout radius is around 13 meters. It has 4 enters/exits and two lanes of 3 meters width each. Traffic is right-handed in all the circuit.



Fig. 5. Experimental test circuit.

### B. Urban trajectory with Bézier curves

The experiments described in this work have been implemented in an electric Citroen van. Two different experiments were carried out considering urban bends and roundabouts.

Figure 6 shows the behavior of the vehicle following the reference path in autonomous driving. The vehicle trajectory is represented by the gray line. Bézier points (black points) are generated when the vehicle is coming on the next curve. For straight segments, two points (the last and the first one of two consecutive Bézier curves) are used to generate the reference line. Figure 7 shows the lateral error throughout the experiment. This error is lower on straight segments. Although on curves it increases slightly, it is corrected when the vehicle turns, keeping it on the road. The higher error is reached when the vehicle past around the sharpest curve (bottom part of figure 6). This error is due to we are using the same controller presented in [17], with the control almost two meters in front of the vehicle. Finally, figure 8 shows the position of the steering wheel during the experiment.

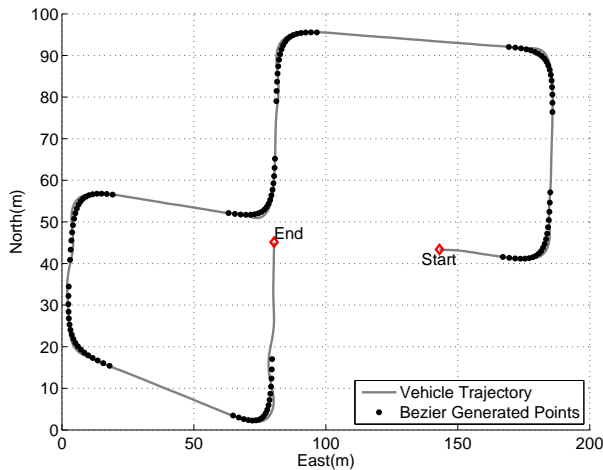


Fig. 6. Experiments using Bézier parametric points.

### C. Roundabouts

Figure 9 shows an experiment considering an entry, exit and lane changes over an urban roundabout. The parametric trajectory generated in the experiment is always followed for the control stage in both curves and inside the roundabout. In the first turn, the vehicle has the exit blocked, so it drives

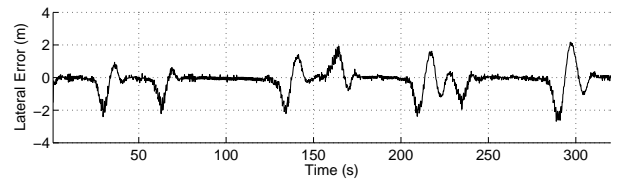


Fig. 7. Lateral error.

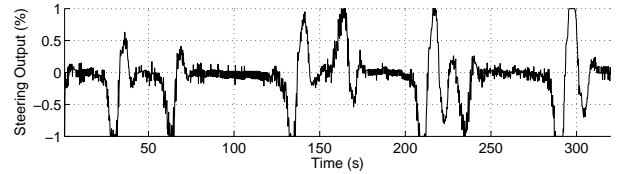


Fig. 8. Steering wheel position.

another turn. In this case, the vehicle goes to inside lane, and when the exit is free, it takes the outside lane and the output curve with Bézier generations.

Figure 10 and 11 shows the lateral and angular error inside the roundabout. Around the 65th second, the first lane change is generated (showed in the first leap of the figure). The vehicle returns to center of the new lane without overshoots. The control point was moved to the front wheel axle of the vehicle. Figure 12 shows the normalized output of the position controller, which is mainly -0.4 because the steering is always turned on the roundabout (anticlockwise), except when the lane changes are produced.

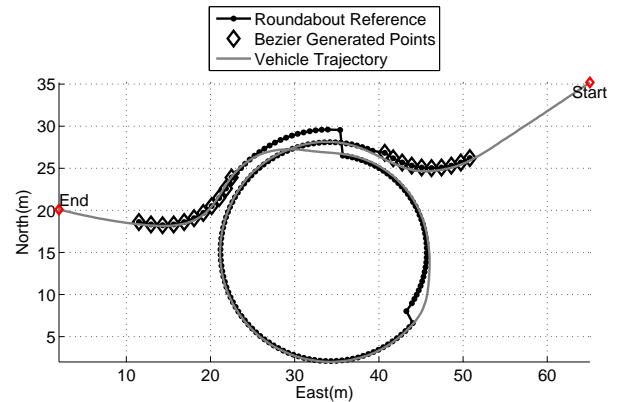


Fig. 9. Experiments in roundabout.

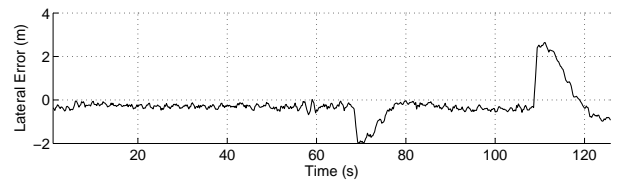


Fig. 10. Lateral error in roundabout.

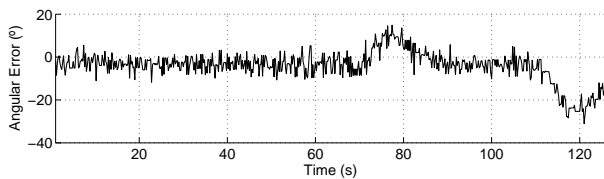


Fig. 11. Angular error in roundabout.

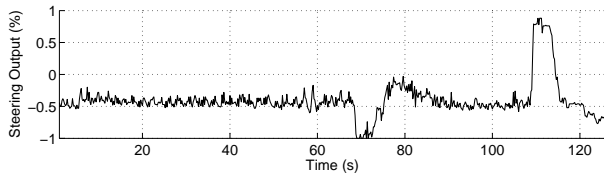


Fig. 12. Steering wheel position.

## VI. CONCLUSIONS AND FUTURE WORKS

In this paper, a trajectory generator and control scheme using parametric equations for autonomous vehicles was proposed. Different tests in urban scenarios demonstrate a good performance of the control architecture, based in previous developments. The path generated with Bézier curves and circle parametric equations requires less effort in the tracking process. For roundabout scenarios, only taking into account the radius, the UTM coordinates and the parametric equation of a circle, an automatic map generation has been achieved in a real vehicle and environment. The contributions on the planner, coordinated system and trajectory generator modules are significant, compared with previous works in the Autopia group, since a new dynamic trajectory generation with parametric equation has been achieved.

The parametric equations allow a computationally efficient and an easily implementation on our control algorithms. Next step in our research will consider lateral accelerations, curvature analysis, and control over the longitudinal actuator to improve the comfort on the vehicle. Moreover, other control variables, such as the curvature, should be added in the lateral control loop to improve the error presented in tight curves.

## ACKNOWLEDGMENT

J. Godoy wants to specially thank to the JAE program (Consejo Superior de Investigaciones Científicas) for its support in the development of this work.

## REFERENCES

- [1] M. Lu, K. Wevers, R. van der Heijden, and T. Heijer, "Adas applications for improving traffic safety," in *Proc. IEEE Int Systems, Man and Cybernetics Conf*, vol. 4, 2004, pp. 3995–4002.
- [2] A. Amditis, E. Bertolazzi, M. Bimpas, F. Biral, P. Bosetti, M. Da Lio, L. Danielsson, A. Gallione, H. Lind, A. Saroldi, and A. Sjogren, "A holistic approach to the integration of safety applications: The insafes subproject within the european framework programme 6 integrating project prevent," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 554–566, 2010.
- [3] S. Glaser, B. Vanholme, S. Mammari, 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.
- [4] L. Labakhua, U. Nunes, R. Rodrigues, and F. S. Leite, "Smooth trajectory planning for fully automated passengers vehicles: Spline and clothoid based methods and its simulation," in *Informatics in Control Automation and Robotics*, ser. Lecture Notes in Electrical Engineering, J. A. Cetto, J.-L. Ferrier, J. M. Costa dias Pereira, and J. Filipe, Eds. Springer Berlin Heidelberg, 2008, vol. 15, pp. 169–182.
- [5] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 561–568, 2010.
- [6] J. Villagra, V. Milanés, J. Perez, and J. Godoy, "Smooth path and speed planning for an automated public transport vehicle," *Robotics and Autonomous Systems*, vol. 60 (2), p. 252265, 2012.
- [7] M. Parent, "Advanced urban transport: Automation is on the way," *IEEE Intelligent Systems*, vol. 22, no. 2, pp. 9–11, 2007.
- [8] D. Betaille and R. Toledo-Moreo, "Creating enhanced maps for lane-level vehicle navigation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 786–798, 2010.
- [9] T. Fraichard and A. Scheuer, "From reeds and shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.
- [10] L. B. Cremean and R. M. Murray, "Model-based estimation of off-highway road geometry using single-axis lidar and inertial sensing," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2006*, 2006, pp. 1661–1666.
- [11] L. Han, H. Yashiro, H. T. N. Nejad, Q. H. Do, and S. Mita, "Bezier curve based path planning for autonomous vehicle in urban environment," in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, 2010, pp. 1036–1042.
- [12] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist, "Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 167–172, 2010.
- [13] J. wung Choi, R. Curry, and G. Elkaim, "Path planning based on bezier curve for autonomous ground vehicles," in *World Congress on Engineering and Computer Science 2008, WCECS '08. Advances in Electrical and Electronics Engineering - IAENG Special Edition of the oct. 2008*, pp. 158–166.
- [14] J. E. Naranjo, C. Gonzalez, R. Garcia, T. de Pedro, and R. E. Haber, "Power-steering control architecture for automatic driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 4, pp. 406–415, 2005.
- [15] J. I. Serrano, J. Alonso, M. D. del Castillo, and J. E. Naranjo, "Evolutionary optimization of autonomous vehicle tracks," in *Proc. IEEE Congress Evolutionary Computation*, vol. 2, 2005, pp. 1332–1339.
- [16] V. Milanés, J. Perez, E. Onieva, C. Gonzalez, and T. de Pedro, "Lateral power controller for unmanned vehicles," *Electrical Review*, vol. 86, no. 1, pp. 207–211, Jan. 2010.
- [17] J. Perez, V. Milanés, and E. Onieva, "Cascade architecture for lateral control in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 73–82, 2011.
- [18] J. Perez, C. Gonzalez, V. Milanés, E. Onieva, J. Godoy, and T. de Pedro, "Modularity, adaptability and evolution in the autopia architecture for control of autonomous vehicles," in *Proc. IEEE International Conference on Mechatronics ICM 2009*, 14–17 April 2009, pp. 1–5.
- [19] A. Furda and L. Vlacic, "Towards increased road safety: Real-time decision making for driverless city vehicles," in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics SMC 2009*, 2009, pp. 2421–2426.
- [20] V. Milanés, J. Naranjo, C. Gonzalez, J. Alonso, and T. de Pedro, "Autonomous vehicle based in cooperative gps and inertial systems," *Robotica*, vol. 26, no. 5, pp. 627–633, Sept. 2008.
- [21] J. Godoy, "Disco e implementacin de un sistema de gestin de trfico basado en vehculos automatizados y comunicaciones inalmblicas," Master's thesis, Universidad Politcnica de Madrid, 2011.
- [22] J. Perez, F. Seco, V. Milanés, A. Jimenez, J. Diaz, and T. de Pedro, "An rfid-based intelligent vehicle speed controller using active traffic signals," *Sensors*, vol. 10, no. 6, pp. 5872–5887, 2010.