R. Lattarulo, L. González and J. Perez, "Real-Time Trajectory Planning Method Based On N-Order Curve Optimization," *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*, 2020, pp. 751-756, doi: 10.1109/ICSTCC50638.2020.9259787.

https://doi.org/10.1109/icstcc50638.2020.9259787

Real-Time Trajectory Planning Method Based On N-Order Curve Optimization

1st Ray Lattarulo
Tecnalia, Basque Research and
Technology Alliance (BRTA)
Biscay, Spain
rayalejandro.lattarulo@tecnalia.com

2nd Leonardo González
Tecnalia, Basque Research and
Technology Alliance (BRTA)
Biscay, Spain
leonardo.gonzalez@tecnalia.com

3rd Joshue Perez
Tecnalia, Basque Research and
Technology Alliance (BRTA)
Biscay, Spain
joshue.perez@tecnalia.com

***Abstract*—In recent years, many functionalities were developed for Automated Vehicles (AVs) and some of them with close-to-market prototypes. A required topic is the generation of continuous trajectories that reduces the amount of discrete and pre-coded instructions while leading the vehicle safely. Consequently, this work presents a novel real-time trajectory planning approach based on numerical optimization of n-order Bézier curves and lane-based information. The generation of a feasible trajectory considers the vehicle dimension while driving into a lane-corridor. The nonlinear optimization problem was solved with the Bound Optimization BY Quadratic Approximation method (BOBYQA), and it uses the passengers' comfort, safety, and vehicle dynamics as constraints of the problem. The solution is validated in a simulation environment using a bus with a length of 12 meters. Moreover, the validation considered the roundabouts due to its complexity, nevertheless, the solution is scalable to other scenarios.**

I. INTRODUCTION

AVs have shown great potential for improving safety, the passengers' comfort, and the efficiency of the driving tasks. Nevertheless, optimal solutions for vehicle decisions, specifically for trajectory generation, are demanded in compliance with the geometry of the maps, vehicle dimensions, and dynamics [1].

Typically, the AVs' decision layer has a hierarchical structure, such as: i) routing based on passengers' requirements, traffic conditions, and road network; ii) reasoning process through a sequence of driving behaviors, e.g. turning, lane-changing, etc; and iii) the generation of feasible trajectories that lead the vehicle safely [2].

Concerning the last topic, the literature presents some issues in the trajectory generation field, mainly produced due to the complexity of the problem. A potential solution is to separate the trajectories in lateral and longitudinal motions. The longitudinal domain has been studied for a longer time, and this has produced robust and reliable speed profiles approaches. In 2015, Ford prepared a patent that considers the optimization of energy consumption and information from the cloud, e.g. the road grade, to maximize the speed efficiency [3]. Recently, Berkeley has proposed a real-time speed profile which is efficiently computed considering quadratic approximation of a non-convex optimization problem which includes vehicle dynamics, passengers comfort, and interaction with other road participants [4].

On the other hand, the lateral domain still lacks robustness. Nevertheless, these previous attempts could be divided into two different groups; sampling-based [5] and numerical optimization approaches [6].

Sampling-based methods demand a strong computational effort to generate a discrete set of pre-computed motion primitives considering road geometry, vehicle dynamics, or a combination of both. An offline trajectory generation approach was proposed considering the vehicle's reachable set of solutions and a lower layer that evaluates potential fields while avoiding collisions with moving obstacles [7]. Another method, inspired in the RRT technique, generates short trajectories based on a stochastic search and a steer function (inclusion of the vehicle motion) [8].

On the other hand, optimization-based methods generate trajectories considering the complete solution-space, with a small compromise in time to obtain the results [9]. A good example was the combination of a smooth cubic function and a Model Predictive Control method for the inclusion of vehicle dynamics that has permitted a comfortable and feasible driving experience [10]. Nevertheless, splitting the task of generating smooth trajectories and vehicle dynamics can provoke conditions where both cannot be satisfied at the same time.

Consequently, this work proposes a novel optimization-based method for a safer trajectory-planning of AVs. It has used n-order Bézier curves and their convex-hull property, among others, to generate optimal and safe trajectories contained into the path while considering the dimensions of the vehicle, its dynamics, and the comfort of the passengers as problem constraints.

The rest of this paper is divided as follows, section II reviewed the main benefits of the Bézier curves and the optimization method used. Next, section III presents the topology that was used for the decision processes of the vehicle. Section IV presents the trajectory planning approach considering the lateral domain. Section V has a description of the test case. Sections VI and VII present the results and discussions of the work, and the contribution finishes with the conclusions and future works on section VIII.

II. CONCEPTS AND BASIS

Our contribution to the optimal paths of AVs is highly dependent on geometry. Consequently, this section presents the principal components used for their generation. The first one is the Bézier curve and the second one is the optimization method *BOBYQA*.

A. BEZIER CURVES

Bézier curves are part of the spline family. An example of them is presented in Fig. 1. These curves are described by the equation:

$$\mathbf{B}(t|n, P_0, ..., P_n) = \sum_{i=0}^{n} b_i \mathbf{P}_i, \quad b_i = \binom{n}{i} t^i (1-t)^{n-i} \quad (1)$$

where $\{b_i \in \mathbb{R}\}$ is the Bernstein polynomial, $\{\mathbf{P}_i \in \mathbb{R}^2\}$ are the control points used to generate the curve, $\{n \in \mathbb{N}^+\}$ is the Bézier order and $\{t \in \mathbb{R}, t = [0, 1]\}$ is the parameter used to construct the curve. This type of curves has been widely used in Automated Vehicles with promising results in trajectory generation. Further information about them can be found in [11], [12].

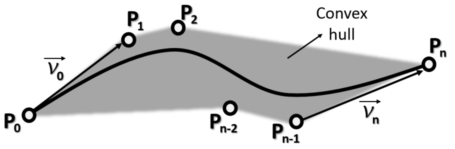

Fig. 1. Bézier and its properties

The Bézier curves have a set of characteristics that make them robust and efficient for AVs' path planning. Nevertheless, this work relies on two of them:

Property 1: *The starting point of the Bézier curve corresponds with the control point* $\mathbf{P}_0$*, and its ending point corresponds with* $\mathbf{P}_n$*.*

Property 2: *The curve lies into the convex hull formed by its control points.*

B. OPTIMIZATION METHOD

The trajectory generation problem of the AVs, considering the vehicle dimensions and lane boundaries, is a highly nonlinear problem with non-smooth, discontinuous, and non-convex objective functions. Derivative-based optimization methods are weak against this problem [13]. However, modern derivative-free methods can achieve good results.

The developments in derivative-free methods have been improved with the emergence of new solvers, such as: *NOMAD*, *OQNLP*, *NEWUOA*, *BOBYQA*, among others [14]. Some of these methods have the following disadvantages: (i) the *NOMAD* method generates a global optimal solution that demands a long computation time [1]. (ii) *OQNLP* demands smooth constraints to find a global optimal [2]. (iii) *NEWUOA* solves unconstrained optimization problems; this is time-consuming and inefficient for a difficult problem which has well-defined constraints [15].

In these terms, the *BOBYQA* method has been used to solve the online optimization problem. It was originally a Fortran package in charge of finding the minimum value of a function $\{F(\mathbf{x}),\ \mathbf{x} \in \mathbb{R}^n\}$ subject to the bound constraints $\{a_i \leq x_i \leq b_i : i = 1, 2, \ldots, n\}$, where $\mathbf{x}$ is the vector to be optimized [16]. The method does not require pre-computed derivatives and that is a major benefit in this type of highly nonlinear problems. The *BOBYQA* algorithm generates a quadratic approximation of the objective function $F(\mathbf{x})$ in the form of $\{Q(x_k) = F(x_k) : k = 1, 2, \ldots, m\}$, with m the number of discretization steps. This approximation, along with the use of the *truncated conjugate gradient method* [17], permits solve the objective function efficiently and robustly.

Our approach used the optimization module of the *DLib* toolkit to compute the *BOBYQA* method. It has C++ and python interfaces distributed under a boost open source license [18].

III. VEHICLE DECISION PROCESS

The literature has a great number of contributions to AVs software and hardware architectures. Also, some of those contributions have considered more detailed representations than other ones. This work has used the vehicle architecture reviewed in [19]. This set-up has a good level of detail in terms of the vehicle decision process. It is divided in three stages which are the *global*, *behavioral*, and *local planning*.

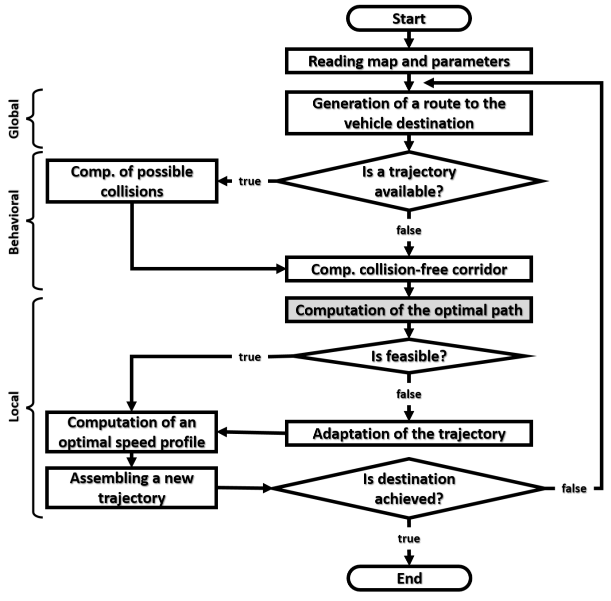
Start
Reading map and parameters
Global
Generation of a route to the vehicle destination
Behavioral
Comp. of possible collisions
true
Is a trajectory available?
false
Comp. collision-free corridor
Local
Computation of the optimal path
true
Is feasible?
false
Computation of an optimal speed profile
Adaptation of the trajectory
Assembling a new trajectory
Is destination achieved?
false
true
End

Fig. 2. Flowchart of the assumed vehicle decision process

Fig. 2 presents the information flow through the three previously-mentioned stages. They are presented as a sequence

[1] https://www.inverseproblem.co.nz/OPTI/index.php/Solvers/NOMAD

[2] https://tomopt.com/tomlab/products/oqnlp/

to simplify the explanation although, some of its tasks could be parallelized. The contribution of this work targets the *computation of an optimal path*, and the figure presents its location in gray color.

This work assumes the presence of a *global planning* which computes a route to lead the vehicle to a destination. The route contains road information, such as the geometry of the lanes, speed limits, traffic signals, etc.

After, the road information is given to the *behavioral planning*. It calculates a collision-free corridor that defines the driveable road's area where the vehicle can move safely. This part uses the information of the trajectory previously computed for the verification and mitigation of any possible collision that could take place in the future. As before, the existence of this part is assumed, for further information refers to [20].

The last part of the decision process is the *local planning* which generates the optimal path and speed profile. An optimal vehicle trajectory is obtained after combining both of them. The computation of the vehicle trajectory obeys the boundaries of the collision-free corridor and the vehicle dimensions. On these terms, the main contribution of the work is in terms of the optimal path, presented in section IV. The speed profile generation was explained in a previous contribution and refer to [21] for further information.

IV. PATH PLANNING APPROACH

Some authors have established that highly-precise prerecorded map information will be needed to execute AVs' tasks under real traffic circumstances [22]. Nevertheless, a recent demonstration has shown a camera approach that had a good performance under the difficulties of the roads of Jerusalem[3].

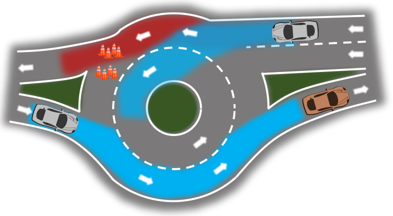

Fig. 3. Road and lane definition

One aspect in common between the approach of prerecorded maps and the one with cameras is the definition of the navigable space with precise lane boundaries. In these terms, the proposed path planning approach relies on the boundaries of a collision-free lane, like the ones depicted in blue in Fig. 3, to generate an optimal and safe path.

The collision-free corridor is defined by the left (B_L) and right (B_R) lane boundaries, as depicted in Fig. 4. The local planning will use these constraints to generate feasible, comfortable, and safe trajectories leading the vehicle controllers.

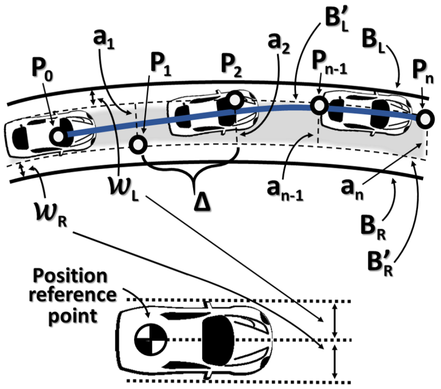

Fig. 4. Bézier control points positioning

The left and right boundaries are moved to the inner part of the collision-free corridor, considering the displacements W_L and W_R for the left and right sides. This action generates the new boundaries B'_L and B'_R depicted in Fig. 4 The values of W_L and W_R are the distance from the vehicle reference point to the outer left and right bound limits of the vehicle.

These bounds define an area that allows the generation of a *safe Bézier trajectory envelope* using the convex hull principle; this is the *property 2* explained in section II-A. The envelope ensures a trajectory that is contained into the lane limits whenever the Bézier's control points lay into the envelope. Fig. 4 presents this area in light gray color.

The initial control point $\mathbf{P}_0$ will have a position given by the following conditions:

1) the initial control point will be given by the current location of the vehicle if there is not a previous feasible-trajectory or the deviation between car and the trajectory is greater than a value Δ_{th}.
2) If the vehicle has a deviation lower than Δ_{th} respect to the previous feasible-trajectory, the first control point will be given by the projection of the vehicle position over the feasible trajectory. This condition ensures the continuity and smoothness of the path (*property 1* explained in section II-A).

This exception prevents a big deviation between the vehicle and the trajectory due to possible human intervention or abnormal operation of the controllers. For a small deviation, the point is projected over the previous feasible-trajectory to generate a small difference that will be corrected by the vehicle controllers, moving the vehicle in the desired direction.

After, a set of n-control points is created over the right bound. The separation between each other is equivalent to a distance of Δ (Fig. 4). For straight and soft bent segments, the high density of points is filtered out. They do not add additional information while increasing the non-linearity of the problem.

Finally, the optimization method will move the control points in the direction of the line segments $\{a_i : i = 1, 2, \ldots, n-1, n\}$ between the bound B'_L and B'_R of the lane, as Fig. 4 depicts; the line segments a_i are perpendicular to the bound B'_R and they end in B'_L. The distance over this axis is adjusted given the objective function:

[3] Web page: www.theverge.com/platform/amp/2020/1/7/21055450/mobileye-self-driving-car-watch-camera-only-intel-jerusalem

$$\min\left\{\sum_{j=1}^{m}\phi(\mathbf{s}_j,k_j),\ j=1,2,3,...,m\right\} \tag{2}$$

where $\mathbf{s}_j$ are the points used to reconstruct the path (interpolation points), k_j is the curvature associated to the point j.

The objective function $\phi(\mathbf{s}_j,k_j)$ is presented as a piecewise function which is switched by a feasibility criterion. The generation of the path is performed considering the vehicle width. Nevertheless, a small portion of the front or rear part of the vehicle could be guided out the corridor limits $[\mathbf{B}_L,\mathbf{B}_R]$ in the cases of long vehicles (e.g. bus or truck). Moreover, a kinematic bicycle model is used to associate the maximum turning radius with the maximum curvature that is feasible k_{max}; the path selection is unfeasible if its maximum curvature value is greater than k_{max}. The mathematical representation of the objective function is:

$$\begin{aligned}\phi(\mathbf{s}_j,k_j) &= \begin{cases}\phi^-(\mathbf{s}_j,k_j), & when\,feasible\\ \phi^+(\mathbf{s}_j,k_j), & when\,unfeasible\end{cases}\\ \phi^-(\mathbf{s}_j,k_j) &= -min\{d(\mathbf{s}_j,\mathbf{B}_R),d(\mathbf{s}_j,\mathbf{B}_L)\}-\frac{1}{\mid k_j\mid}\\ \phi^+(\mathbf{s}_j,k_j) &= max\{d(\mathbf{s}_j,\mathbf{B}_R),d(\mathbf{s}_j,\mathbf{B}_L)\}+max\{\mid k_j\mid\\ &\quad -k_{max},0\}\end{aligned} \tag{3}$$

$d(\mathbf{s}_j,\mathbf{B}_R)$ and $d(\mathbf{s}_j,\mathbf{B}_L)$ refer to the distance between the path point and the right or left bound respectively, and k_j is the curvature associated to the interpolated point $\mathbf{s}_j$.

The component $max\{d(\mathbf{s}_j,\mathbf{B}_R),d(\mathbf{s}_j,\mathbf{B}_L)\}$, in the case of an unfeasible weight $\phi^+(\mathbf{s}_j,k_j)$, is in charge of penalizing the objective due to the displacement of a part of the vehicle out of the corridor; and the component $max\{\mid k_j\mid -k_{max},0\}$ adds a contribution to a greater objective if the curvature limit is violated.

On the other hand, the component $-min\{d(\mathbf{s}_j,\mathbf{B}_R),d(\mathbf{s}_j,\mathbf{B}_L)\}$, of the feasible part of the weight function $\phi^-(\mathbf{s}_j,k_j)$, is in charge of centering the vehicle in the middle of the lane; and the component $-\frac{1}{|k_j|}$ decreases the contribution to the weight with a reduction of the curvature.

Previously, a speed profile solution was proposed using the maximum path distance as a problem constraint [20], [23]. In these terms, the path will have a maximum displacement distance related to the provided free-collision corridor, and this must be respected by the speed profile to avoid any unsafe condition out of its limits.

V. PROPOSED SCENARIO

The approach has been tested in simulation environments considering a vehicle's width of 2.5 meters and length of 12.0 meters. The authors tested the method on a segment of the Malaga's Port (Spain) that will be used as proving ground for AVs demonstrations in the up-coming months.

The upper part of Fig. 5 shows the complete path. The algorithm was validated in a portion that has tight and bend-segments. Moreover, the selected section is part of a roundabout that represents one of the most difficult scenarios for drivers and automated vehicles (lower part of Fig. 5).

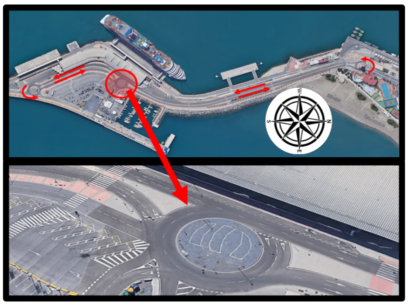

Fig. 5. The proposed scenario

This roundabout was split into three parts, considering the geometry of the road, for the analysis of the trajectory planning approach. The sections are the roundabout entry, driving in the roundabout, and the roundabout exit. The entry and exit have one part of the trajectory in a straight segment (curvature approximately 0.0), and its middle path has a curvature proportional to the roundabout radius. The central part of the roundabout must satisfy a narrow and continuous turning path.

These use cases have been considered only for a real-time trajectory generation based on free-collision lane information. In these terms, the approach is scalable to execute stop-and-go, lane change, overtaking, obstacle avoidance, and lane-keeping maneuvers based on camera or map information. The computation of the free-collision lane is a task of an upper layer of the vehicle decision module (behavioral planning) that future works will analyze.

VI. RESULTS

The analysis was split into three zones. All of them described the current value of the trajectory, in the sample time t_k, with a thick and blue line. The thin and light grey lines represented the previous values of the trajectory planned, in the sample time $\{t_i,\ i=k-1,k-2,k-3,...\}$. The thick black lines depict the boundaries of the path.

A. *Roundabout entrance*

The top of Fig. 6 shows the trajectory generation on the roundabout entry, starting in a path segment more or less straight. The curvature value of ≈ 0 verifies the statement aforementioned (bottom of Fig. 6).

A positive turning value is obtained with counterclockwise rotation and a clockwise rotation implies a negative value. In these terms, the trajectory has been adapted to the change of the turning sense in the entry. The curvature value was $-0.033[m^{-1}]$, equivalent to a turning radius of $\approx 30[m]$ (first trajectory samples), and the last samples had a value of $0.03[m^{-1}]$. This change of the curvature concavity permitted to fit the trajectory according to the shape of the entry.

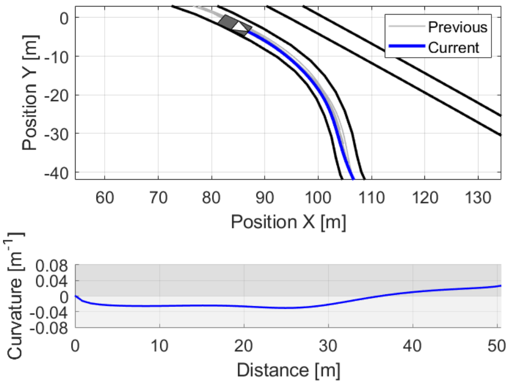

Fig. 6. Trajectory planning on the roundabout (enty)

B. Driving in the roundabout

Fig. 7 presents the results for the inner part of the roundabout. The trajectory described a smooth and monotonous counterclockwise turn in this section, finishing on the roundabout exit.

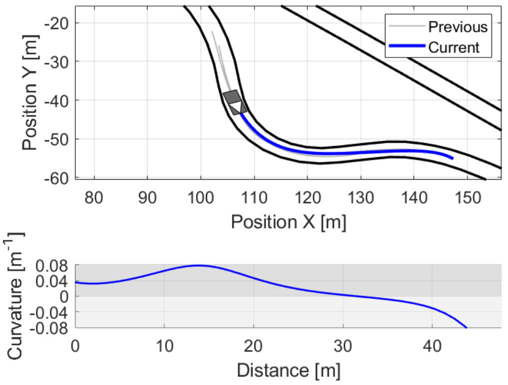

Fig. 7. Trajectory planning on the roundabout (central segment)

The first samples have a curvature related to the previous experiment (entrance) with a value of $0.04[m^{-1}]$ that is equivalent to a turning radius of $\approx 25[m]$. This curvature reached a maximum value of $0.08[m^{-1}]$ or a turning radius of $12.5[m]$, which is equal to the roundabout radius (circle fitting).

The last part of the trajectory has generated a change of sign (concavity) that permits to fit into the exit shape. This event provoked a minimum value of the curvature under $-0.08[m^{-1}]$, which could end up in a tight turn. These final values will be re-adapted in the following iterations of the real-time trajectory planner.

C. Roundabout exit

Fig. 8 depicts the roundabout exit. The curvature of the generated trajectory started with a magnitude proportional to the inverse of the roundabout radius. After, the curvature decreased to a minimum value of $-0.04[m^{-1}]$. This behavior is similar to the one observed in the entry due to the symmetry of the problem.

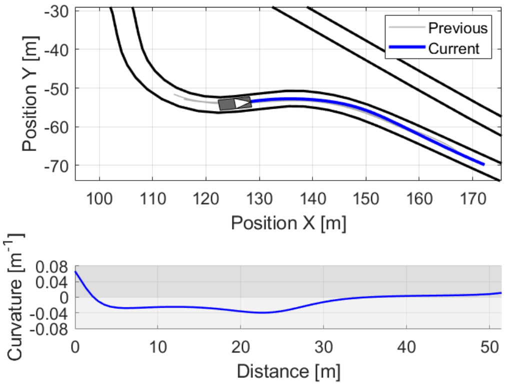

Fig. 8. Trajectory planning on the roundabout (exit)

The trajectory finishes with values of ≈ 0.0 in the straight segment. These values are interesting results because some tracking controllers demand continuous curvature trajectories [11]. These approaches have the precondition of producing a reliable description of the path curvature to generate the proper correction (controllers).

VII. DISCUSSION

The maximum lane distance was set to 50 meters (the top part of Fig. 9) to improve the computation. A vehicle at $50kph$ (max. speed in urban environments) can reduce its speed to $0kph$ in this distance, at a deceleration of $2m/s^2$. This maximum deceleration is under nominal operation, and it can be stronger in case of an emergency. The distance of the generated trajectory surpasses the $50m$ mark due to concatenating a previous feasible-trajectory with the current one. On the other hand, the length was lower than the maximum lane distance due to the reduction of the feasible trajectory distance before achieving a new feasible solution.

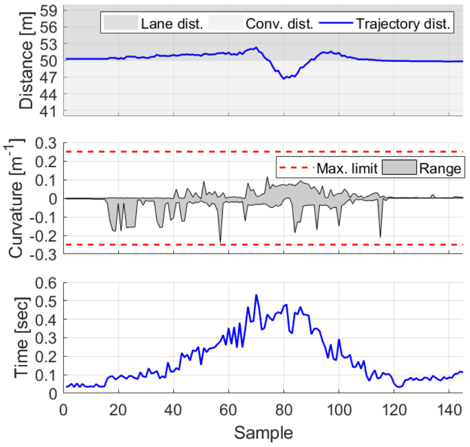

Fig. 9. Analysis of the trajectory distance, curvature and time.

The maximum value for the curvature was obtained with the approximation of the kinematic bicycle model:

$$k_{max} = \frac{tan(\alpha_{max})}{L}$$

α_{max} is the maximum value of the frontal wheel in the model, and L is the wheelbase [24]. In this test case, the values were $L = 3.5[m]$ and $\alpha_{max} = 40^{\circ}$ resulting in $k_{max} \approx 0.24[m^{-1}]$. In these terms, the maximum and minimum values of the generated trajectories have been within the bound limits (middle part of Fig. 9).

Lastly, the algorithm has presented a computation time lower than 0.53 seconds (bottom part of Fig. 9). This value can be analyzed as a maximum convergence distance of $17[m]$ for a vehicle at $120[kph]$ and $4[m]$ for a vehicle at $30[kph]$. The maximum convergence value has been obtained in the narrow bend segment where a lower speed is demanded. The trajectories have been generated at a sample rate of 0.05 seconds.

VIII. CONCLUSION

This work has presented a novel real-time trajectory planning method based on Bézier curves for the path generation. The optimal trajectory was found using a nonlinear local optimization method named *BOBYQA*. Moreover, it has used the convex hull principle of Bézier curves, produced from n-points, to generate a safe trajectory containing the vehicle into the lane given its dimension. A fast speed profile was considered, based on previous works.

The method has presented a total computation time in the range of 50 to 550 milliseconds on segments of 50 meters length. The algorithm has been tested in simulation environments and considering a real scenario context.

This paper has used roundabouts to validate the approach due to their complexity, nevertheless, other scenarios such as stop-and-go, lane change, overtaking, obstacle avoidance, stop maneuvers on a shoulder, lane returning, and merging can be addressed using this method via the definition of the collision-free corridors.

As future works, this algorithm will be improved with the addition of the vehicle's length for locating the control points; this will ensure a path that maintains the vehicle area completely into the corridor. This method will be tested with long non-holonomic vehicles as an automated bus in real test cases.

ACKNOWLEDGMENT

This work was supported by the European Project SHOW from the Horizon 2020 program under the grant agreement No 875530.

REFERENCES

[1] J. Godoy, A. Artuñedo, and J. Villagra, "Self-generated osm-based driving corridors," *IEEE Access*, vol. 7, pp. 20 113–20 125, 2019.

[2] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu, and C. Hu, "A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1376–1386, 2019.

[3] D. P. Filev, J. O. Michelini, S. J. Szwabowski, P. R. McNeille, and S. Di Cairano, "Route navigation with optimal speed profile," Jul. 14 2015, uS Patent 9,081,651.

[4] C. Liu, W. Zhan, and M. Tomizuka, "Speed profile planning in dynamic environments via temporal optimization," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 154–159.

[5] Z. Kingston, M. Moll, and L. E. Kavraki, "Sampling-based methods for motion planning with constraints," *Annual Review of Control, Robotics, and Autonomous Systems*, 2018.

[6] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for bertha - a local, continuous method," *IEEE Intelligent Vehicles Symposium Proceedings*, 2014.

[7] H.-T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia, "Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2347–2354.

[8] L. Palmieri, S. Koenig, and K. O. Arras, "Rrt-based nonholonomic motion planning using any-angle path biasing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2775–2781.

[9] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method," *IEEE Transactions on Intelligent Transportation System*, 2018.

[10] H. Guo, C. Shen, H. Zhang, H. Chen, and R. Jia, "Simultaneous trajectory planning and tracking using an mpc method for cyber-physical systems: A case study of obstacle avoidance for an intelligent vehicle," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4273–4283, 2018.

[11] D. González, J. Pérez, R. Lattarulo, V. Milanés, and F. Nashashibi, "Continuous curvature planning with obstacle avoidance capabilities in urban scenarios," *IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[12] R. Lattarulo, L. González, E. Martí, J. Matute, M. Marcano, and J. Pérez, "Urban motion planning framework based on n-bézier curves considering comfort and safety," *Journal of Advanced Transportation*, 2018.

[13] K. R. Fowler, J. P. Reese, C. E. Kees, J. Dennis Jr, C. T. Kelley, C. T. Miller *et al.*, "Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems," *Advances in Water Resources*, vol. 31, no. 5, pp. 743–757, 2008.

[14] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293, 2013.

[15] M. J. Powell, "The newuoa software for unconstrained optimization without derivatives," in *Large-scale nonlinear optimization*. Springer, 2006, pp. 255–297.

[16] M. J. D. Powell, "The bobyqa algorithm for bound constrained optimization without derivatives," *Cambridge NA Report NA2009/06, University of Cambridge*, pp. 26 – 46, 2009.

[17] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," *School of Computer Science Carnegie Mellon University*, 1994.

[18] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[19] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2015.

[20] R. Lattarulo, D. He, and J. Perez, "A linear model predictive planning approach for overtaking manoeuvres under possible collision circumstances," *IEEE Intelligent Vehicles Symposium*, 2018.

[21] D. Heß, R. Lattarulo, J. Pérez, J. Schindler, T. Hesse, and F. Köster, "Fast maneuver planning for cooperative automated vehicles," *IEEE International Conference on Intelligent Transportation Systems*, pp. 1625–1632, 2018.

[22] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving," *International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[23] D. Heß, R. Lattarulo, J. Pérez, T. Hesse, and F. Köster, "Negotiation of cooperative maneuvers for automated vehicles: Experimental results," *IEEE Intelligent Transportation Systems Conference*, pp. 1545–1551, 2019.

[24] J. P. Rastelli, R. Lattarulo, and F. Nashashibi, "Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles," *IEEE Intelligent Vehicles Symposium*, pp. 510 – 515, 2014.