IAES International Journal of Robotics and Automation (IJRA)
Vol. 15, No. 2, June 2026, pp. 427~444
ISSN: 2722-2586, DOI: 10.11591/ijra.v15i2.pp427-444

Robust efficient ego-vehicle path prediction based on Bezier curves for autonomous driving

Hanan H. Hussein[1], Ahmed Atef[2], Mohamed Hanafy Radwan[2]
[1]Computers and Systems Department, Electronics Research Institute, Cairo, Egypt
[2]Valeo Innovations and Technology hub in Egypt (VITE), Smart Village, Cairo, Egypt

Article Info

Article history:

Received Feb 20, 2026
Revised Apr 29, 2026
Accepted May 17, 2026

Keywords:

Automatic emergency braking
Autonomous driving
Bezier curves
Collision avoidance
Path prediction
Third-order polynomial

ABSTRACT

Accurate ego-vehicle path prediction is essential for safety-critical functions in advanced driver assistance systems (ADAS), such as automatic emergency braking (AEB) and collision avoidance. Existing models based on Clothoid curves are typically not sufficient in expressing complex maneuvers and are not highly adaptive to various vehicle dynamics. In addition, these models struggle with accuracy in circular maneuvers and fail to use in complex paths (*e.g.*, S-shapes). This paper proposes a novel representation of the ego-vehicle path prediction using Bezier curves. The proposed Bezier curves are composed of two Cartesian third-order polynomial functions. They are formulated efficiently to model both circular and S-shaped trajectories with high accuracy and low computational cost. Our method significantly reduces prediction error, achieving over 95% improvement in average Euclidean distance error compared to Clothoidal models along about 50 m paths in controlled circular scenarios. The proposed algorithm, designed with O(n) complexity, is suitable for real-time applications on low-power automotive hardware. Its effectiveness is demonstrated through simulation using CarMaker, and a collision estimation module for AEB is developed based on the predicted paths.

Corresponding Author:

Hanan H. Hussein
Computers and Systems Department, Electronics and Research Institute (ERI)
12611 El-Nozha, Cairo, Egypt
Email: hananhussein@eri.sci.eg

1. INTRODUCTION

Road safety and driving efficiency have emerged as major concerns over the last decade, and advanced driver assistance systems (ADAS) has come forward [1]. ADAS comprises a set of technologies meant to enhance the safety of the vehicle and the comfort of driving by making drivers better at taking informed decisions in avoiding danger. These technologies combine sensors, cameras, and radar with AI to track everything in the surroundings and provide real-time feedback or act automatically. The importance of ADAS is not only safety, but also even the road to the future of autonomous driving lies through the path of ADAS. Since vehicle automation technology advances, it will go on to fully autonomous systems. This is a foundational step in progressively adding automation to driving tasks with driver involvement.

Many active safety features of ADAS are based on the prediction of the ego-vehicle path of motion from the current ego-vehicle state of motion [2]. The ego-vehicle path of motion is the planned route or trajectory that an automated vehicle is expected to follow to reach its destination safely and efficiently. It considers the vehicle's current position, speed, surroundings, and any obstacles along the way. For example, algorithms responsible for end-user functionality of automatic emergency braking (AEB) use the ego-vehicle predicted path and paths estimated for surrounding detected objects to compute the expected positions of

collision [3]. According to the expected positions of collision with surrounding objects, the AEB algorithms are capable of considering the suitable braking actions to avoid collisions, especially on highway roads. As a result, the accuracy of the predicted ego-vehicle path and its robustness with the changing ego-motion state play an important role in the functionality of AEB.

The predicted ego-vehicle path could be mainly dependent on the ego-motion instantaneous radius of rotation, which is dependent on the instantaneous input steering made by the driver. As a result, the predicted path could be based on a circle with its radius varying instantaneously to match the current instantaneous radius of rotation. Such a circle becomes with very high radius (almost a straight line) for very small steering action. The circular presentation of the predicted ego-vehicle path is simple and efficient to cover most of the basic driving situations where no more information regarding the road markers (lanes) or other specific driver input actions (like lane changes) are provided [4].

The main limitation in the circular presentation of the ego-vehicle path is that it assumes that the vehicle will continue moving with the same curvature (same radius of rotation), as it considers only the ego-motion instantaneous calculations [5]. On the other hand, other important factors could greatly affect the predicted path and worth to be considered as well. The vehicle moving on a certain lane should probably be expected to continue moving in the same lane in a normal driving situation unless the driver sets any of the blinkers to indicate moving right or left. As a result, the lanes detected by long-range sensors (like RADAR or LiDAR) should be considered in the estimation of the predicted path. Moreover, the driver input actions to change the current lane should be considered as well in the predicted path estimation. It is clear that the circular presentation of the predicted path is not suitable for non-circular rounding lanes, as in snake-like lanes, or for lane change scenarios where vehicle moves in S-shape path. Thus, there is a need for a more generic and robust approach to present the predicted ego-vehicle path in both basic driving situations and complex ones rather than the circular approach.

Bezier curves can be a proper generic solution to support basic and complex driving maneuvers in presenting predicted ego-vehicle path [6]. Bezier polynomials, according to their order, can generate paths with varying steering either not counter steering or counter-steering, starting from straight paths then circular ones and ending with S-shapes or double S-shape (like take-over maneuvers) [7]. In order not to increase its complexity much, S-shape paths can be achieved by third-order polynomial Bezier equations. Bezier curves based on a combination of two third-order polynomials can provide circular predicted paths for basic maneuvers plus S-shape paths for lane change or snake-shaped lanes. Moreover, it is capable of presenting the Clothoidal cubic polynomial curve model, which could be efficient in presenting not counter steering paths as in [8], [9]. This means that the Clothoidal path prediction on both highways and urban roads is a subset of the proposed Bezier curves path prediction, where the proposed approach is capable of covering circular maneuvers that are not fully covered by the regular Clothoids. Thus, third-order polynomial Bezier equations can be an efficient, robust, and cheap solution to present predicted ego-vehicle paths for both not counter steering paths (circular paths and Clothoids) and counter steering paths (S-shapes) [10]. It can be the proper solution for path prediction and planning used in high-speed ADAS features like AEB [11], lane keep assist (LKA) [12], and lane change assist (LCA) [13], and also in low-speed maneuvers like parking [14].

Bezier curves has proven high efficiency in path planning either at the level of long-range route planning [15] or short-range maneuvers planning like at intersections in roundabouts [16]. It can connect efficiently and smoothly between intersection points while guaranteeing continuous curvature and steering. While Bezier curves are commonly used for smooth trajectory generation, alternatives such as B-splines and NURBS also offer strong continuity and flexibility in trajectory design, as detailed in the seminal work by Piegl and Tiller [17]. The order of the Bezier curve and its complexity shall increase with the increase in the connected intersection points that shall be provided based on given map nodes. Based on its efficiency in path planning, the Bezier curves can be used efficiently as well in path prediction. The number of connected points or frames in path prediction shall be very limited compared to path planning, where a frame is defined by a location point and orientation angle. This is because the path prediction covers a limited space, as it is mainly based on the instantaneous ego-vehicle motion state and perception information of the surrounding environment. Consequently, there is a strong demand for enhancing path prediction techniques to be robust, efficient, and deployable on low-power automotive-grade hardware. Such a solution shall be based on Bezier curves and capable of running perfectly on different automotive controllers including low computing power ones with other complex applications and basic components on the automotive software stack running in parallel.

While Bezier curves are widely used for trajectory generation, their direct application to ego-vehicle path prediction faces challenges: i) control points are often selected heuristically, leading to curvature discontinuities during dynamic maneuvers; ii) high-order Bezier curves improve accuracy but increase computational cost, making them unsuitable for real-time ADAS [18]; and iii) most implementations focus on global path planning rather than kinematic-aware local prediction [19]. Thus, the problem tackled in this

paper is how to use the Bezier curves efficiently in ego-vehicle path prediction. The efficiency is intended based on the different aspects of accuracy, robustness, and cost.

Because of their smoothness characteristics and computational efficiency, Bezier curves have become a basic tool in autonomous vehicle trajectory generation. Third-order Bezier curves have lately been investigated for several uses in path planning and control. Using N-order polynomial search with boundary conditions, Vinayak *et al.* [16] presented a new Bezier curve control point search algorithm for autonomous navigation. Although their approach shows good path generation, it is less appropriate for real-time ego-vehicle path prediction since it concentrates mostly on global route planning and requires predefined geometric constraints. Arslan and Tiemessen [18] developed an adaptive Bezier degree reduction and splitting method for computationally effective motion planning in the framework of vehicle motion planning. Their work maintains a focus on offline path smoothing rather than dynamic prediction based on vehicle kinematics, even while it greatly advances the field by optimizing curve complexity. Similarly, Ding *et al.* [14] used Bezier curves mimicking Clothoids for perpendicular parking moves to produce smooth low-speed paths. But their method ignores the difficulties of high-speed path prediction and only considers organized parking situations.

Integration of Bezier curves with learning-based methods has been tried by several researchers. For risk assessment in vehicle motion planning, Wang *et al.* [3] coupled long short-term memory (LSTM) networks with Bezier parameter prediction. Although creative, such hybrid techniques bring computational complexity and lowered interpretability that might not satisfy the strict criteria of safety-critical ADAS functions. Moreover, such learning-based methods depend on heuristic control point selection that may result in curvature discontinuities during dynamic maneuvers. Existing approaches for ego-vehicle path prediction in ADAS primarily rely on geometric models such as Clothoid curves, polynomial fitting, or data-driven techniques in [19]. They present learning-based methods that depends on large datasets to predict trajectories. Unfortunately, it requires high computational resources and may not predict well to unseen scenarios.

Gao *et al.* [20] introduced a self-supervised deep learning framework that optimizes depth and ego-motion estimation. Their novelty lies in the introduction of a feature quadtree loss, which replaces traditional photometric loss to better capture details. However, compared to our proposed Bezier-based path prediction, their method presents significant disadvantages in terms of computational overhead and architectural complexity. Furthermore, they focus on visual perception, our approach provides a continuous, kinematically-feasible path representation that is specifically optimized for active safety functions like AEB, particularly in complex S-shaped and circular maneuvers where visual-only pose estimation may encounter drift. Peng *et al.* [21] proposed a fixed dual third-order Bézier curve architecture that directly maps instantaneous ego-vehicle motion states to a continuous path representation. Unlike conventional methods that depend on iterative fitting, variable model structures, or data-driven training processes, the proposed formulation provides a deterministic and closed-form solution with low computational overhead. This enables consistent modelling of both circular and S-shaped trajectories within a unified framework, while maintaining real-time performance suitable for embedded automotive systems. However, Moreover, their system is built based on calculation executed on social pooling network infrastructure as they need to collect data from the vehicle itself and the surrounding environment.

The existing research suffers from three main constraints: First, few techniques solve the computational constraints of automotive-grade hardware. Second, most Bezier implementations concentrate on path planning rather than real-time prediction. Third, control points are usually determined geometrically rather than from vehicle kinematics. By means of a kinematic-aware Bezier formulation especially intended for ego-vehicle path prediction in ADAS applications, the kinematic constraints of the vehicle motion, like maximum lateral acceleration and maximum steering, can be appropriately considered.

Unlike these previous efforts, our work suggests a new approach that combines two third-order Bezier curves for real-time ego-vehicle path prediction. We derive the control points analytically based on the instantaneous vehicle state (yaw rate, speed, and orientation), which allows for high-accuracy path approximation with minimal computational overhead, in contrast to existing methods that either assume high-order curve fits or rely on numerical optimization. Integration with a collision estimation module for AEB validates our approach, which is especially well suited to safety-critical ADAS applications. To the best of our knowledge, this is the first work that uses a fixed dual third-order Bezier curve architecture for embedded predictive safety functions to predict paths from instantaneous motion states. It shall be noted that the proposed solution is validated with test cases of recorded vehicle test traces, including real signals of vehicle speed and yaw rate. However, the preprocessing of such signals before being fed to our solution is beyond the scope of the paper, which focuses on proving the algorithmic concept. Modelling uncertainties in the input signals and propagating them in the path prediction to have the generated path with accompanied uncertainty could be more efficient in the collision estimation for AEB with accompanied uncertainty as well. However, this principle shall be extended to our future work.

The main contribution in this paper, which is the subject of the granted patent [11], lies in solving the tackled problem through the formulation of a new representation of the predicted ego-vehicle path. The proposed new representation based on third-order polynomial Bezier equations has the following characteristics that are highly important for proper and efficient end-user functions of ADAS.

- Accuracy: The proposed solution achieves higher accuracy with less error (more than 90% in some scenarios) than the regular Clothoids when tested on circular maneuvers as justified in simulation results.
- Robustness: The proposed solution is capable of covering the different basic maneuvers starting from simple straight and circular, and ending with sinusoidal or S-shape maneuvers. Moreover, it is based on the input instantaneous speed and yaw rate, which can be provided for any vehicle type (car, bus, truck, van, and motorcycle).
- Cheapness: The solution as presented in the formulated algorithm (Algorithm 1 in section 3) has low complexity (O(n), where n presents the number of waypoints forming the generated path) that makes it capable of running on cheap targets.

In the next sections of the paper, we will show how the Bezier curves can be easily and efficiently used to present the predicted ego-vehicle path. Comparison plots of scenarios are presented for the generated Bezier path versus the one obtained from Clothoids against the ground truth reference road on CarMaker for circular paths. Section 2 clarifies the path prediction framework. It introduces the mathematical background for the third order polynomial Bezier equations, the Bezier curves obtained from the combination of two Bezier equations, and how to generate a Bezier path, which is the closest to a circular arc with a specific given radius of rotation. In section 3, we show the proposed approach for ego-vehicle predicted path generation. Simulation results are presented in section 4. Comparison plots of path prediction with Bezier curve versus Clothoidal polynomial against CarMaker ground truth and calculation of Bezier path collision points for AEB are presented. Finally, the paper is concluded in section 5 supported with future work.

2. PATH PREDICTION FRAMEWORK

Our proposed solution considers only the input motion state of the ego-vehicle, specifically its speed and yaw rate (which represent the instantaneous radius of rotation or curvature). As a result, the predicted path shall be a circular maneuver based on the instantaneous input curvature. As mentioned before, the proposed third-order polynomial Bezier equations are capable of covering more complex maneuvers like S-shapes. Moreover, prediction of S-shape or similar maneuvers require more input information of full perception of the lanes, status of the blinkers, and predetermination of the route and updated map of the road as in [15]. However, this could not be enough to predict snake maneuvers as for any reason the driver could change his decision by not following the r oute, or not changing the lane after toggling the blinker, or change more than one lane instead of one. There shall be solid criteria and clear requirements defined to cover the system behavior for such several use cases. The end-user function under study in our scenario is AEB, which is a highly safety critical feature according to the new car assessment program (NCAP) standards and test protocols [22]. Thus, the ego-vehicle path prediction proposed in this paper serving AEB functionality is based only on the input instantaneous current vehicle state, and it provides only predicted circular and straight maneuvers. More complex maneuvers shall be considered as future work for path planning rather than path prediction, where the generated path shall be followed by the automatic control system to serve other functionalities like LCA with clear requirements and standards defined to cover its different use cases [23].

This section clarifies the information related to the trajectory planning methodology utilizing real-time parametric Bezier curves. These curves obtained from two third order polynomial Bezier equations. The approach will concentrate on employing these curves for path prediction in highway and urban environments, including intersections, roundabouts, and lane changes, as well as for speed planning to ensure comfort and safety. The explanation of the planning framework will be separated into three sub-sections: i) the generic third order Bezier equation, i) how the Bezier curve can be obtained from combining two third order polynomial equations. iii) Bezier curve presentation for circular arc.

2.1. Third order polynomial Bezier equation

The cubic (third order) Bezier equation r(t) can be written in its generic form as follows:

$$r(t) = (1-t)^3 R_0 + 3(1-t)^2\, t\, R_1 + 3(1-t)\, t^2\, R_2 + t^3\, R_3 \quad (1)$$

The independent parameter t is in the range [0,1]. R_0 and R_3 are the values of $r(t=0)$ and $r(t=1)$, respectively. R_1 and R_2 are the control points with the values of $\left[\frac{\dot{r}(t=0)}{3} + r(t=0)\right]$ and $\left[r(t=\right.$

1) $-\frac{\dot{r}(t=1)}{3}\Big]$, respectively. As shown in Figure 1, the blue points (R_0 and R_3) are the initial and final values of the red Bezier curve. The green points (R_1 and R_2) are the control points at $t = 1/3$ and $t = 2/3$, respectively. The green control points obtained from the derivatives of the Bezier curve at its boundaries ($t = 0$ and $t = 1$) controls the shape of the Bezier curve $r(t)$. Table A.1 lists the mathematical symbols used in the Bezier polynomial equations, its definitions, and its evaluation in Appendix A.

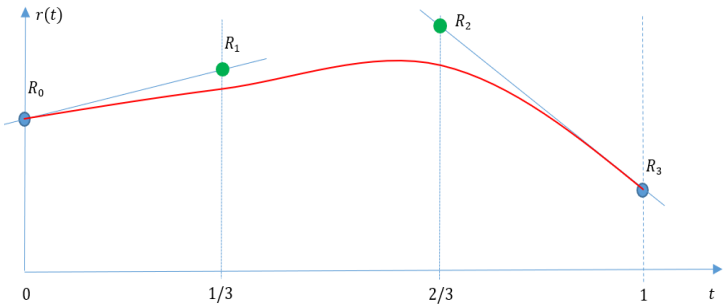

Figure 1. Plot of third order Bezier equation

2.2. Bezier curve obtained from the combination of two third order polynomial Bezier equations

According to (1), the third order polynomial Bezier equations for the two coordinate variables $x_p(t)$ and $y_p(t)$ can be as follows.

$$x_p(t) = (1-t)^3 X_0 + 3(1-t)^2\, t\, X_1 + 3(1-t)\, t^2\, X_2 + t^3\, X_3 \quad (2)$$

$$y_p(t) = (1-t)^3 Y_0 + 3(1-t)^2\, t\, Y_1 + 3(1-t)\, t^2\, Y_2 + t^3\, Y_3 \quad (3)$$

The coordinates x_p , y_p set as the local coordinates for the Bezier curve generation are characterized by the following boundary conditions as shown in Figure 2.

$$x_p(t=0) = X_0 = 0 \qquad x_p(t=1) = X_3 = M$$
$$y_p(t=0) = Y_0 = 0 \qquad y_p(t=1) = Y_3 = 0$$

As a result, (2) and (3) can be simplified to be as follows.

$$x_p(t) = 3\, X_1\, (1-t)^2\, t + 3\, X_2\, (1-t)\, t^2 + M\, t^3 \quad (4)$$

$$y_p(t) = 3\, Y_1\, (1-t)^2\, t + 3\, Y_2\, (1-t)\, t^2 \quad (5)$$

$\{X_1 , X_2\}$ and $\{Y_1 , Y_2\}$ are the control points of $x_p(t)$ and $y_p(t)$, respectively. Thus, they are characterized by the following equations.

$$X_1 = x_p(t=0) + \frac{\dot{x}_p(t=0)}{3} = \frac{1}{3}\,\dot{x}_p(t=0) \qquad Y_1 = y_p(t=0) + \frac{\dot{y}_p(t=0)}{3} = \frac{1}{3}\,\dot{y}_p(t=0)$$
$$X_2 = x_p(t=1) - \frac{\dot{x}_p(t=1)}{3} = M - \frac{1}{3}\,\dot{x}_p(t=1) \qquad Y_2 = y_p(t=1) - \frac{\dot{y}_p(t=1)}{3} = -\frac{1}{3}\,\dot{y}_p(t=1)$$

Thus, the derivatives at the boundaries can be deduced as follows.

$$\dot{x}_p(t=0) = 3\, X_1 \quad (6)$$

$$\dot{y}_p(t=0) = 3\, Y_1 \quad (7)$$

$$\dot{x}_p(t=1) = 3\,(M - X_2) \quad (8)$$

$$\dot{y}_p(t=1) = -3\, Y_2 \quad (9)$$

The orientation of the vehicle in the defined local Bezier Coordinates (x_p, y_p) is presented with θ_r. The tangent of the orientation of the vehicle can be defined as follows.

$$tan(\theta_r) = \frac{\Delta y_p}{\Delta x_p} = \frac{\dot{y}_p}{\dot{x}_p} \tag{10}$$

Based on (6) – (10), we can deduce the relation between the control points of the Bezier equations in the two defined coordinates.

$$\tan(\theta_{r_0}) = t_0 = \frac{\dot{y}_p(t=0)}{\dot{x}_p(t=0)} = \frac{Y_1}{X_1} \qquad Y_1 = t_0 \; X_1 \tag{11}$$

$$\tan\left(\theta_{r_f}\right) = t_f = \frac{\dot{y}_p(t=1)}{\dot{x}_p(t=1)} = \frac{-Y_2}{M - X_2} \qquad Y_2 = -t_f \; (M - X_2) \tag{12}$$

By substituting (11) and (12) in (5), the Bezier equation in the local coordinates (x_p, y_p) becomes as:

$$y_p(t) = 3\, t_0 \; X_1 \; (1-t)^2 \, t - 3\, t_f \; (M - X_2) \; (1-t) \, t^2 \tag{13}$$

where $x_p(t)$ and $y_p(t)$ are as presented in (4) and (13), respectively. They show that the Bezier curve to be generated between two input frames with defined t_0 and t_f is mainly a function in the places of the control points X_1 and X_2. In order to simplify the values of the two control points by combining them in one variable with their functionality almost preserved, they can be assumed to be equally placed from the boundaries with a distance b. Then, the values of X_1 and X_2 can be assumed to be $(b\, M)$ and $(1-b)\, M$, respectively. The range of b is $[0, 0.5]$. As a result, the generated Bezier curve can be finally presented as a function depending on b as:

$$x_p(t) = (3bM)\, t + 3M(1-3b)\, t^2 - 2M(1-3b)\, t^3 \tag{14}$$

$$y_p(t) = (3bMt_0)t - 3bM\left(t_f + 2t_0\right) t^2 + 3bM\left(t_f + t_0\right) t^3 \tag{15}$$

Figure 2 shows a general plot of the Bezier curve generated from the combination of the two third order polynomial Bezier (14) and (15). The challenge afterwards is to look for the best value of b that could serve a specified functionality by the generated Bezier path. Roughly, the setting of b with $1/3$ could be mostly sufficient to have reasonable Bezier paths connecting a specific initial frame $(0, 0, \theta_{r_0})$ with a specific target one $(M, 0, \theta_{r_f})$ in the local Bezier coordinates (x_p, y_p). Setting (b) with $(1/3)$ shall be capable of presenting cubic clothoidal polynomial, where $x_p(t) = M\, t$ is linear, and $y_p(t) = (Mt_0)t - M\left(t_f + 2t_0\right) t^2 + M\left(t_f + t_0\right) t^3$ is a third-order polynomial. However, for specific desired paths, which we can enclose mainly in circular paths and S-shape ones, the value of b needs to be further investigated to have the best outcome from the generated Bezier path to fit with the desired functionality. The next section explains how the value of b can be computed to have the Bezier path as close as possible to a circle with specific given radius of rotation.

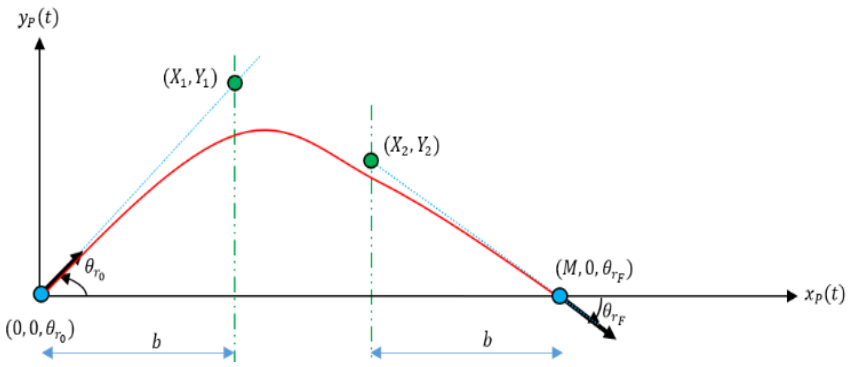

Figure 2. Plot of Bezier curve from combination of two third order Bezier equations

2.3. Bezier curve presentation for circular path

According to the parametric third order polynomial in (4) and (5), the control points are deduced from the geometry of Figure 3 similar to [24] as follows. The red points in the figure correspond to initial and goal frames (boundary points (X_0, Y_0) and (X_3, Y_3)), while the green points correspond to the control points (X_1, Y_1) and (X_2, Y_2). The blue star is the center of rotation of the circular arc.

$$\begin{matrix} X_0 = 0 \\ Y_0 = 0 \end{matrix} \qquad \begin{matrix} X_1 = \epsilon R\cos(\theta^*) \\ Y_1 = \epsilon R\sin(\theta^*) \end{matrix} \qquad \begin{matrix} X_2 = M - \epsilon R\cos(\theta^*) \\ Y_2 = \epsilon R\sin(\theta^*) \end{matrix} \qquad \begin{matrix} X_3 = M \\ Y_3 = 0 \end{matrix}$$

where (R) is the radius of the target circle that Bezier curve is intended to be as close as possible. The term (ϵ) presents the factor of the radius by which the tangential distance from the initial and goal positions are considered to calculate the control points. The angle (θ^*) is both the initial and goal orientation angles in the local coordinates according to the characteristics of a circular arc as shown $(\theta_{r_0} = -\theta_{r_f} = \theta^*)$. Thus, the target connection arc has net angle $(2\theta^*)$. By substitution in (4) and (5), with the corresponding values of control points, the following equations can be obtained.

$$x_p(t) = 3\epsilon R\cos(\theta^*)(1-t)^2 t + 3(M - \epsilon R\cos(\theta^*))(1-t)t^2 + Mt^3$$
$$y_p(t) = 3\epsilon R\sin(\theta^*)(1-t)^2 t + 3\epsilon R\sin(\theta^*)(1-t)t^2$$
$$\because M = 2R\sin(\theta^*)$$
$$\therefore x_p(t) = 3\epsilon R\cos(\theta^*)[t - 2t^2 + t^3] + 3(2R\sin(\theta^*) - \epsilon R\cos(\theta^*))[t^2 - t^3] + 2R\sin(\theta^*)t^3$$

$$x_p(t) = [3\epsilon R\,cos(\theta^*)]\,t + [6R\,sin(\theta^*) - 9\epsilon R\,cos(\theta^*)]\,t^2 + [6\epsilon R\,cos(\theta^*) - 4R\,sin(\theta^*)]\,t^3 \quad (16)$$

$$y_p(t) = 3\epsilon R\,sin(\theta^*)\,[t - t^2] \quad (17)$$

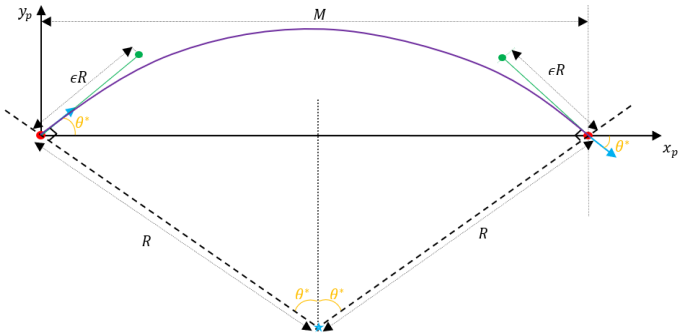

Figure 3. Circular Bezier derivation

In order to have the Bezier path as close as possible to the circular path, the middle point (at $t = 0.5$) is selected so that the Bezier values are on the circular path. Such a simplified approach is very close to the optimized approach derived in Appendix B. However, it is greatly more efficient for realization on embedded targets with restricted resources and real time constraints than the optimized approach which needs numerical integration computing as proved in the Appendix B to be performed every processing cycle.

$$x_p(t = 0.5) = \left[\frac{3}{2} - \frac{9}{4} + \frac{6}{8}\right]\epsilon\, R\cos(\theta^*) + \left[\frac{6}{4} - \frac{4}{8}\right]R\sin(\theta^*) = R\sin(\theta^*) = \frac{M}{2}$$
$$y_p(t = 0.5) = \frac{3}{4}\epsilon R\sin(\theta^*) = R - R\cos(\theta^*)$$
$$\therefore \epsilon = \frac{4}{3}\left[\frac{1-\cos(\theta^*)}{\sin(\theta^*)}\right]$$

As mentioned in the previous section, the control points (X_1) and (X_2) are assumed at (bM) and $((1-b)M)$, respectively. The term b, which is necessary to compute the location of the control points, can be computed by substitution in (14) and (15) with $(t_f = -t_0)$ and $(t_0 = \tan(\theta_{r_0}) = \tan(\theta^*))$, then comparing against (16) and (17), respectively. Same conclusion can be simply obtained by equating the two values of control

point ($X_1 = b\,M = \epsilon\,R\cos(\theta^*)$).

$$\therefore b = \frac{\epsilon\,R\cos(\theta^*)}{M} = \frac{\epsilon}{2\,\tan(\theta^*)} = \frac{2}{3}\left[\frac{\cos(\theta^*) - \cos^2(\theta^*)}{\sin^2(\theta^*)}\right]$$

Since the angle (θ^*) is half the difference between the initial frame orientation angle (θ_0) and goal frame orientation angle (θ_f) in the global coordinates (ego-vehicle coordinates to be introduced in the next section), then calculation of (b) for circular Bezier can be simplified in the following two equations.

$$\Delta\theta = \left|\theta_f - \theta_0\right| \tag{18}$$

$$b = \frac{2}{3}\left[\frac{cos(\frac{\Delta\theta}{2}) - cos^2(\frac{\Delta\theta}{2})}{sin^2(\frac{\Delta\theta}{2})}\right] \tag{19}$$

3. THE PROPOSED APPROACH FOR EGO-VEHICLE PREDICTED PATH GENERATION

This section details the generation of the circular predicted ego-vehicle path with Bezier curve based on the derived in (18) and (19) to compute the control points. The predicted path is generated in the ego-vehicle (global) coordinates (x, y) as shown in Figure 4. The coordinate (x) presents the longitudinal axis of the ego-vehicle, while the coordinate (y) presents the lateral axis. Given a specific instantaneous radius of rotation of the vehicle (R), a specific circular path can be defined with its center of rotation as the point of intersection of the perpendicular lines to the four wheels directions (blue star in Figure 4).

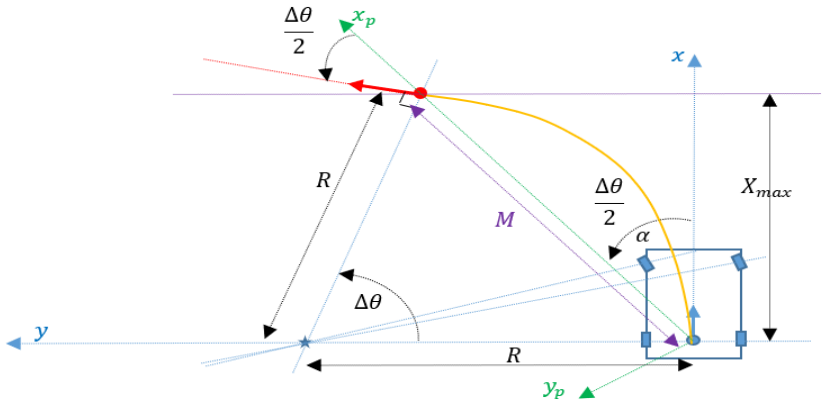

Figure 4. Circular predicted ego-vehicle path presented with Bezier curve

A goal frame position can be defined by the intersection of the circular path with horizontal line parallel to the ego-vehicle lateral axis and located at a defined maximum distance from the ego-vehicle center of the rear axle. Such maximum distance is defined as the range limit of ego-vehicle path prediction (X_{max}). The initial ego-vehicle frame is in blue in Figure 4, while the target circular Bezier path is in orange, and the goal frame obtained from the intersection is in red. Also, in Figure 4, the local Bezier coordinates (x_p, y_p) are in green, while the global ego-vehicle coordinates (x, y) are in blue. The angle (α) is the rotation angle between the two coordinates. The absolute orientation difference between the initial and goal frames is ($\Delta\theta$) as in [24]. According to the geometry of circular arc in Figure 4, it is clear that $\left(\alpha = \frac{\Delta\theta}{2}\right)$.

It is worth mentioning that the instantaneous radius of rotation (R) provided by the ego-motion calculations can be computed as the ratio between the ego-vehicle's instantaneous speed (v) and yaw rate $(\dot{\theta})$. Such calculation could be more accurate at significant speed and yaw rate values (especially at high speeds). At lower speeds where the yaw rate could be a bit noisy, it could be better to calculate the ego-vehicle radius of rotation as the ratio between the wheelbase and the tangent of the front wheel angle assuming the rear wheels are not steerable as revealed in the kinematics model in [25]. This calculation is based on accurate estimation of the front wheel angle given the measured steering wheel angle.

Based on the geometry in Figure 4 and the reached formulas in previous sections, the Bezier path generation for circular prediction of ego-vehicle path in global coordinates (x, y) can be detailed as follows given (X_{max}) and $\left(R = v/\dot{\theta}\right)$.

$$\Delta\theta = \sin^{-1}\left(\frac{X_{max}}{R}\right) \qquad M = 2R\sin\left(\frac{\Delta\theta}{2}\right)$$

$$b = \frac{2}{3}\left[\frac{\cos\left(\frac{\Delta\theta}{2}\right) - \cos^2\left(\frac{\Delta\theta}{2}\right)}{\sin^2\left(\frac{\Delta\theta}{2}\right)}\right] \qquad t_0 = -\tan\left(\frac{\Delta\theta}{2}\right)$$

$$x_p(t) = (3bM)\,t + 3M(1-3b)\,t^2 - 2M(1-3b)\,t^3 \qquad y_p(t) = (3bMt_0)\,t(1-t)$$

$$x(t) = x_p(t)\,\cos\left(\frac{\Delta\theta}{2}\right) - y_p(t)\,\sin\left(\frac{\Delta\theta}{2}\right) \qquad y(t) = x_p(t)\,\sin\left(\frac{\Delta\theta}{2}\right) + y_p(t)\,\cos\left(\frac{\Delta\theta}{2}\right)$$

Regardless of circular paths, the formula of third order parametric polynomial equations of the Bezier curve in (14) and (15) is sufficient to generate reasonable paths connecting specific initial and goal frames with $(b = 1/3)$. This shall cover the basic third-order Clothoidal polynomials revealed in [8] and [9]. However, for S-shape paths with high steering at low-speed maneuvers (like in parking), it could be better to set $(b = 0.4)$ based on experimental trials, but this is not highly relevant to path prediction, which is needed mostly at high speeds. It shall be noted that increasing the value of (b) (moving the control points to the middle) leads to having the steering actions more at the middle of the path, while decreasing the value of (b) (moving the control points to the boundaries) leads to have the steering more at the start and end of the path.

As the path prediction is based on the current instantaneous ego-vehicle motion state of speed and yaw rate covering only circular maneuvers, then the proposed solution can be formulated in the following Algorithm 1. The algorithm expects as input the instantaneous longitudinal speed and yaw rate of the ego-vehicle, and provides as an output the generated predicted path presented with (x, y) points defined in the local Cartesian coordinates of the ego-vehicle shown in Figure 4. The generated points presenting the predicted path $\left(x(t), y(t)\right)$ are function on the independent parameter (t), which varies from 0 to 1 with step (Δt) defined according to the distance resolution for path generation (Δs). Such distance resolution and range limit (X_{max}) shall be considered as input to the algorithm as well. A new parameter $\left(A_{y_{max}}\right)$ is added presenting the maximum limit of lateral acceleration for a normal driving scenario with no significant lateral slipping to limit the path prediction error that assumes pure circular motion. Finally, a curvature threshold (C_{th}) shall be defined to differentiate between the straight and circular path generations, which could be an input to the algorithm. If the current curvature is greater than the defined threshold then the maneuver is circular, otherwise it is straight. It is clear from the algorithm formulated as follows that its complexity is $O(n)$, where (n) presents the number of waypoints forming the generated path $\left(n = \frac{1}{\Delta t} + 1\right)$. Roughly, (C_{th}) can be assumed as $(1/[4 * X_{max}])$ so that the generated path is almost straight at such curvature threshold and below.

Algorithm 1. Ego-vehicle predicted path generation

$[x(t), y(t)] = Generate\ Path\left(v, \dot{\theta}, \Delta s, X_{max}, A_{y_{max}}, C_{th}\right)$
  $A_y = v\,\dot{\theta}$
  $if\ \left(A_y < A_{y_{max}}\right)$
    $\Delta t = \Delta s / X_{max}$
    $C = \dot{\theta}/v$
    $if\ (|C| > C_{th})$
      $\Delta\theta = \sin^{-1}(X_{max} \times C)$
      $M = 2\sin\left(\frac{\Delta\theta}{2}\right) / C$
      $b = \frac{2}{3}\left[\frac{\cos\left(\frac{\Delta\theta}{2}\right) - \cos^2\left(\frac{\Delta\theta}{2}\right)}{\sin^2\left(\frac{\Delta\theta}{2}\right)}\right]$
      $t_0 = -\tan\left(\frac{\Delta\theta}{2}\right)$
      $for\ (t = 0 : \Delta t : 1)$
        $x_p(t) = (3bM)\,t + 3M(1-3b)\,t^2 - 2M(1-3b)\,t^3$
        $y_p(t) = (3bMt_0)\,t(1-t)$
        $x(t) = x_p(t)\,\cos\left(\frac{\Delta\theta}{2}\right) - y_p(t)\,\sin\left(\frac{\Delta\theta}{2}\right)$
        $y(t) = x_p(t)\,\sin\left(\frac{\Delta\theta}{2}\right) + y_p(t)\,\cos\left(\frac{\Delta\theta}{2}\right)$
      $end\ for$
    $else$

```
18:              for (t = 0 : Δt : 1)
19:                 x(t) = t × X_max
20:                 y(t) = 0
21:              end for
22:           end if
23:        end if
```

4. SIMULATION RESULTS

This section presents simulation results plots of path prediction with Bezier curves versus Clothoidal polynomial. It also presents the calculation of Bezier path collision points for AEB. Traces with different speeds and steering maneuvers are recorded from real vehicle test, including provided signals of vehicle speed and yaw rate. Such recorded traces of vehicle tests are mapped to test scenarios on CarMaker with the actual road maneuvering mapped to CarMaker Simulation. For each scenario, the ego vehicle driven path is extracted from CarMaker and plotted as a golden reference. The main idea from the CarMaker simulation is to build a reference path according to the vehicle road designed for the maneuver simulating the input recorded real vehicle signals. There shall be limitations in the mapping between the actual vehicle motion and the simulated one in CarMaker, especially that we are considering only the vehicle kinematics of speed and yaw rate, not the full dynamics. However, this limitation is acceptable for the normal driving testing scenarios under study with almost no side slipping or any excessive dynamical actions (for example, lateral acceleration doesn't exceed 0.5 m/s^2). As justified at the beginning of section 2, the simulation testing is dedicated for circular paths prediction according to the input instantaneous vehicle state of current maneuvering curvature.

4.1. Comparison plots of path prediction with Bezier curve versus Clothoidal polynomial against CarMaker ground truth

Both the predicted path based on Bezier third order parametric polynomial in (14) and (15) and Clothoidal third order polynomial equation ($y = a\,x^3 + b\,x^2 + c\,x + d$) (where the coefficients are computed based on third order curve fitting to the reference path) are compared against the golden reference of CarMaker driven road for certain basic circular and straight maneuvers test case scenarios. For every scenario, the path points are plotted in 2D Cartesian coordinates in meters, the error at a certain point on a given path is the Euclidean distance from it to its corresponding one in the reference golden path. Both the average accumulated error against the reference and the final position error are calculated. The average accumulated error is the average of errors at all the points along the path, while the final position error is only the error at the final point.

Figures 5 to 9 reveal such comparisons and the accompanied errors mentioned in the title of each figure, where the CarMaker golden reference path is in black, the Bezier path proposed is in red, and Clothoidal curve fitting polynomial is in blue labeled in the titles as 'Current Polynomial'. It is clear from the difference in both the average accumulated and final position errors the significant enhancement in the prediction accuracy, which can be expressed with the percentage decrease in the error. For test cases 1, 2, 3, and 4 for different relaxed and sharp circular maneuvers, the accuracy was enhanced about 95% or more. For test case 5 for a slightly straight maneuver, the accuracy enhanced about 50%.

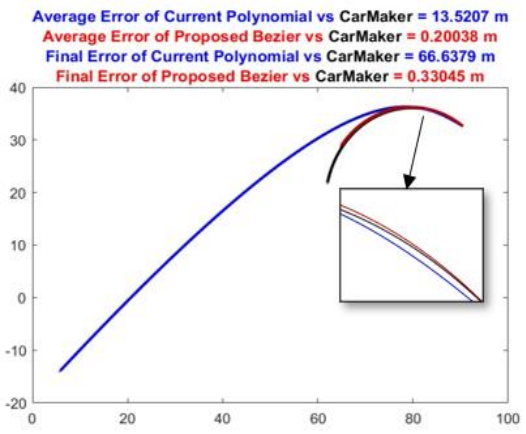

Figure 5. Comparison plot for test case 1

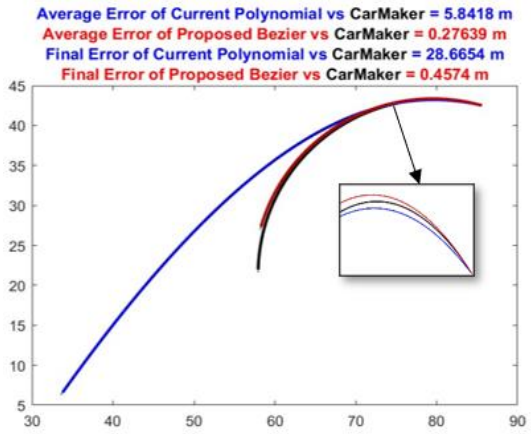

Figure 6. Comparison plot for test case 2

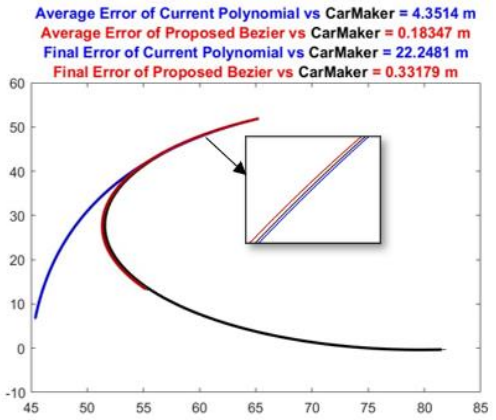

Figure 7. Comparison plot for test case 3

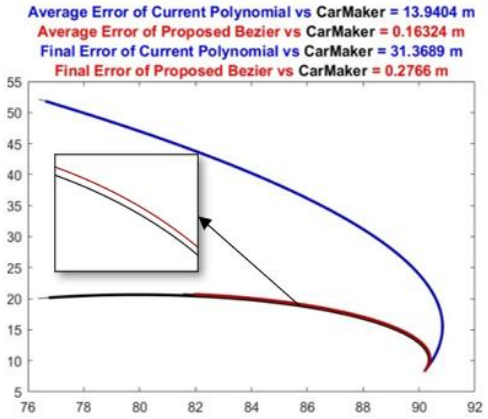

Figure 8. Comparison plot for test case 4

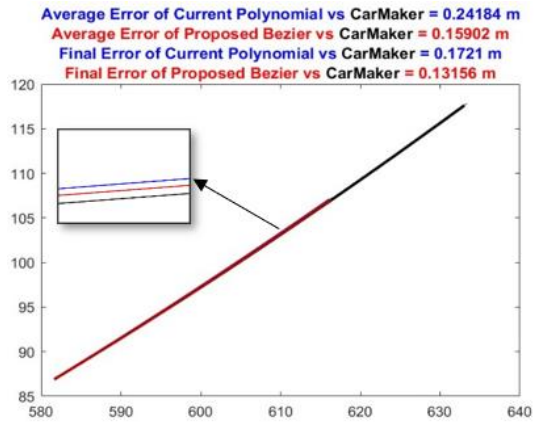

Figure 9. Comparison plot for test case 5

Table 1 lists the results for more testing scenarios of basic straight and circular maneuvers. Both the average accumulated error (AE) and final error (FE) in meters are declared for every scenario. The declared errors are for both the proposed Bezier and the Clotholidal curve fitting polynomial.

Table 1. Results for more testing scenarios

Test Case	Type	Bezier	Clothoidal
6	Circular	AE = 0.20829 FE=0.35567	AE = 10.646 FE=50.297
7	Circular	AE = 0.17798 FE=0.27376	AE = 6.682 FE=21.8966
8	Circular	AE = 0.18303 FE=0.31304	AE = 6.2132 FE=26.0942
9	Circular	AE = 0.23098 FE=0.27038	AE = 2.8793 FE=10.5197
10	Circular	AE = 0.20197 FE=0.26577	AE = 5.2157 FE=19.4045
11	Circular	AE = 0.18343 FE=0.28041	AE = 10.7426 FE=27.5829
12	Straight	AE = 0.08334 FE=0.12246	AE = 0.080107 FE=0.12218
13	Straight	AE = 0.097854 FE = 0.17988	AE = 0.099234 FE=0.17939

4.2. Calculation of Bezier path collision points for AEB

For AEB, it is not only required to generate the ego-vehicle predicted path but also to compute the position of collision with specific tracked object. Moreover, it could also be required to determine the exact

location of the collision at the front bumper of the ego-vehicle. In such a situation, the Bezier curve for path prediction can be generated at the center of the front bumper. For circular paths, the radius of rotation needed for Bezier calculation should be computed at the center of front bumper. It shall be based on the ego-vehicle's instantaneous speed (v) and yaw rate $(\dot{\theta})$, which are estimated and measured at the center of the front bumper. In addition, the orientation of the vehicle at initial and goal frames $(\theta_{r_0},\theta_{r_f})$ should be the direction of motion at the center of front bumper as well, which shall be the same as the ego-vehicle heading angle.

As shown in Figure 10, the corresponding paths at the right and left corners of the front bumper in such a case can be easily computed as parallel to the generated Bezier path at the center of front bumper in the local Bezier coordinates. The green and blue paths, which are parallel to the red, present the generated paths at the right and left corners of the front bumper, respectively. Assuming (w) is the width of the vehicle, the parallel paths can be computed as follows.

$$y_{p_R}(t) = y_p(t) - \frac{w}{2}\cos(\theta_{r_0}) \qquad x_{p_R}(t) = x_p(t) + \frac{w}{2}\sin(\theta_{r_0})$$
$$y_{p_L}(t) = y_p(t) + \frac{w}{2}\cos(\theta_{r_0}) \qquad x_{p_L}(t) = x_p(t) - \frac{w}{2}\sin(\theta_{r_0})$$

In order to compute the ego-vehicle location at collision and point of collision, the estimated path of the tracked object is transferred to the ego-vehicle local Bezier coordinates. For example, if the estimated path of an object is a straight line, then its transformation to the Bezier coordinates of the ego-vehicle is also a line. Similarly, if the tracking system can accurately estimate the target paths to be presented as third-order parametric polynomials (like the Bezier curve proposed to present the ego-vehicle path), then its transformation to the Bezier coordinates of ego-vehicle is also third-order parametric polynomials. Thus, the points of intersection of the transformed path with each of the three Bezier paths (the original path at the center and its parallel ones at right and left) are computed by solving third order polynomial equation as detailed in Appendix C. Figures 11-14 show samples of collision points calculation at the three Bezier curves (presenting the generated paths at center, right corner, and left corner of the front bumper) for different scenarios of tracked objects with straight predicted motion, with different angles of intersection, and at different distances. The X-Axis presents the Longitudinal Axis of the ego-vehicle, the Y-Axis presents the Lateral Axis of the ego-vehicle, and the origin point is the center of the ego-vehicle front bumper.

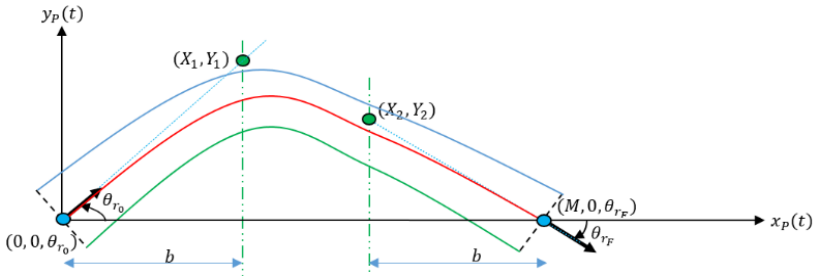

Figure 10. Parallel Bezier Paths for front bumper Boundaries to compute point of collision

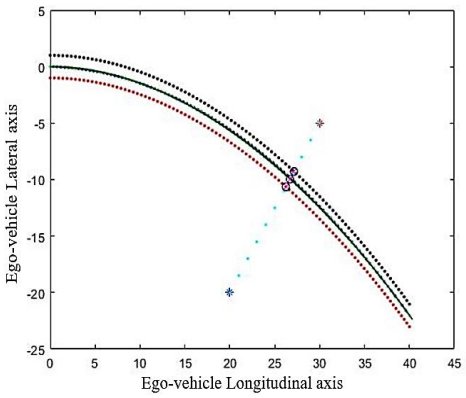

Figure 11. Collision points calculation for use case 1

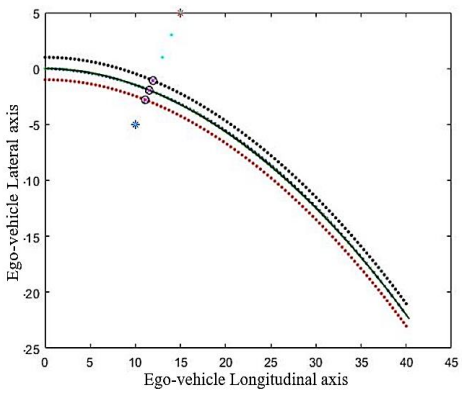

Figure 12. Collision points calculation for use case 2

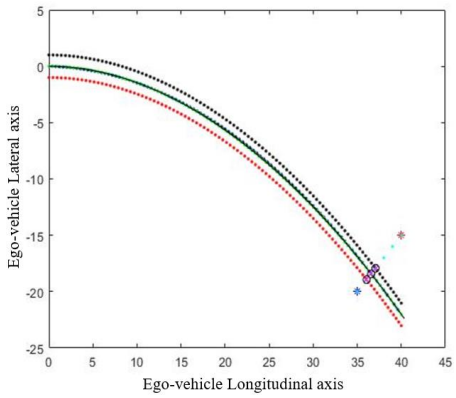

Figure 13. Collision points calculation for use case 3

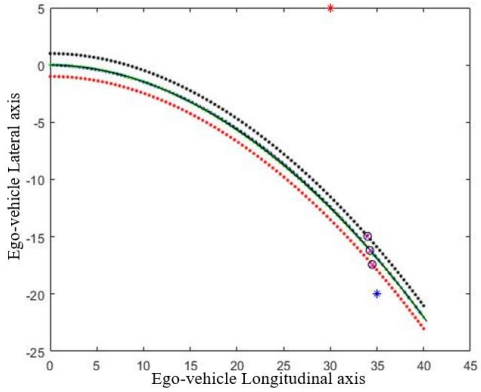

Figure 14. Collision points calculation for use case 4

5. CONCLUSION AND FUTURE WORK

This paper introduced a new approach for ego-vehicle path prediction framework based on two Cartesian third-order polynomial Bezier curves. Such approach is efficient, as it can accurately present the estimated ego-vehicle path based on its instantaneous kinematics with cheap calculations. Moreover, it is robust, as it is capable of covering the different driving maneuvers related to the different end-user functionalities. The performance of the proposed approach is tested on several scenarios in CarMaker against ground truth simulated roads to reflect its accuracy. It is compared against the basic third-order Clothoid-polynomials used regularly in path prediction on highways showing great enhancement in path accuracy that reaches more than 95% in some scenarios. The complete solution for collision estimation serving AEB is highlighted based on the proposed ego-vehicle path prediction.

The proposed approach is realized for the collision estimation in AEB and can be extended for other functionalities like the target selection in ACC and lateral control in LKA. More complicated scenarios for collision estimation can be revealed in the future where several tracked objects can be assumed with more complicated predicted paths for them based on parametric third-order polynomials. However, it shall be solved using the same third order solver presented in Appendix C. Moreover, more complex maneuvers like S-shapes shall be tackled with the proposed third-order Bezier polynomials for path planning rather than path prediction to include more functionalities like LCA. The proposed method demonstrates clear advantages in accuracy and computational efficiency when compared to traditional Clothoidal models. However, future work will focus on extending the evaluation to include additional benchmark methods such as B-spline interpolation, optimization-based trajectory prediction (*e.g.*, MPC), and learning-based approaches like LSTM or Transformer models. Such comparative studies will provide a more comprehensive understanding of the strengths and limitations of the dual Bezier formulation in a variety of realistic and high-dynamic driving scenarios, including real vehicle dynamics and uncertainties modelling.

FUNDING INFORMATION

Authors state there is no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Hanan H. Hussein					✓					✓	✓	✓		✓
Ahmed Atef		✓	✓				✓	✓	✓					
Mohamed Hanafy Radwan	✓			✓		✓			✓				✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors ensure that there is no potential conflict of interest possibly influencing the interpretation of data in the paper.

INFORMED CONSENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ETHICAL APPROVAL

All authors confirm that they have reviewed and approved the final manuscript. We declare that this work is original, has not been published previously, and is not under consideration elsewhere.

DATA AVAILABILITY

The authors ensure that data are available on demand.

REFERENCES

[1] K. D. P. Damsara and A. G. de Barros, "A systematic review on user acceptance of advanced driver assistance systems (ADAS)," *Transportation Research Procedia*, vol. 82, pp. 3472–3482, 2025, doi: 10.1016/j.trpro.2024.12.082.

[2] P. Lytrivis, G. Thomaidis, and A. Amditis, "Cooperative path prediction in vehicular environments," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, Oct. 2008, pp. 803–808. doi: 10.1109/ITSC.2008.4732629.

[3] H. Wang *et al.*, "Risk assessment and mitigation in local path planning for autonomous vehicles with LSTM-based predictive model," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 2738–2749, 2022, doi: 10.1109/TASE.2021.3075773.

[4] H. Guo, Q. Meng, D. Cao, H. Chen, J. Liu, and B. Shang, "Vehicle trajectory prediction method coupled with ego vehicle motion trend under dual attention mechanism," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–16, 2022, doi: 10.1109/TIM.2022.3163136.

[5] J. Zhu *et al.*, "Other vehicle trajectories are also needed: A driving world model unifies ego-other vehicle trajectories in video latent space," in *Proceedings AAAI Conference on Artificial Intelligence*, 2025, vol. 40, no. 16. doi: 10.1609/aaai.v40i16.38403.

[6] G. E. Farin, *Curves and surfaces for CAGD: a practical guide*, 5th ed. San Francisco, CA, USA: Morgan Kaufmann, 2002.

[7] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *International Symposium on Combinatorial Search, SoCS 2008*, no. 1001.48105. pp. 18–80, 2008.

[8] C. F. Lin, A. G. Ulsoy, and D. J. LeBlanc, "Vehicle dynamics and external disturbance estimation for vehicle path prediction," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 3, pp. 508–518, May 2000, doi: 10.1109/87.845881.

[9] K. Ran, Y. Wang, C. Fang, Q. Chai, X. Dong, and G. Liu, "Improved RRT* path-planning algorithm based on the Clothoid curve for a mobile robot under kinematic constraints," *Sensors*, vol. 24, no. 23, p. 7812, 2024, doi: 10.3390/s24237812.

[10] M. Radwan, "Vehicle path planning based on Bezier curves (in Germany)," no. DE102021112920A1. May 2021. [Online]. Available: https://worldwide.espacenet.com/patent/search/family/083898672/publication/DE102021112920A1?q=DE102021112920A1

[11] M. Radwan, M. Mamdouh-Khairy, and A. Atef, "Path prediction for a vehicle (in Germany)," no. DE102021110099A1. Apr. 2021. [Online]. Available: https://worldwide.espacenet.com/patent/search/family/083508022/publication/DE102021110099A1?q=DE102021110099A1

[12] J. D. G. Montoya, E. L. S. Teixeira, A. Murilo, and R. R. Da Silva, "A comprehensive analysis of model predictive control for lane keeping assist system," *IEEE Access*, vol. 11, pp. 140216–140228, 2023, doi: 10.1109/ACCESS.2023.3342034.

[13] J. Yao, G. Chen, and Z. Gao, "Target vehicle selection algorithm for adaptive cruise control based on lane-changing intention of preceding vehicle," *Chinese Journal of Mechanical Engineering (English Edition)*, vol. 34, no. 1, p. 135, 2021, doi: 10.1186/s10033-021-00650-8.

[14] N. Ding, L. Cao, C. Duan, and J. Liao, "Geometric path plans for perpendicular parking based on Clothoid curve," in *Proceedings - 2023 2nd International Conference on Sensing, Measurement, Communication and Internet of Things Technologies, SMC-IoT 2023*, 2023, pp. 161–166. doi: 10.1109/SMC-IoT62253.2023.00036.

[15] Y. Zhao, L. Mo, and J. Liu, "Path planning based on traffic flow prediction for vehicle scheduling," in *2023 IEEE/CIC International Conference on Communications in China, ICCC 2023*, 2023, pp. 1–5. doi: 10.1109/ICCC57788.2023.10233352.

[16] A. Vinayak, M. A. Zakaria, K. Baarath, and A. P. P. A. Majeed, "A novel Bezier curve control point search algorithm for autonomous navigation using N-order polynomial search with boundary conditions," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2021, vol. 2021-September, pp. 3884–3889. doi: 10.1109/ITSC48978.2021.9564605.

[17] L. Piegl and W. Tiller, *The NURBS book*, 2nd ed., vol. 35, no. 02. Berlin, Germany: Springer, 1997. doi: 10.5860/choice.35-0952.

[18] O. Arslan and A. Tiemessen, "Adaptive Bézier degree reduction and splitting for computationally efficient motion planning," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3655–3674, 2022, doi: 10.1109/TRO.2022.3187296.

[19] A. Shaju, S. Southward, and M. Ahmadian, "Enhancing autonomous vehicle navigation with a clothoid-based lateral controller," *Applied Sciences (Switzerland)*, vol. 14, no. 5, p. 1817, 2024, doi: 10.3390/app14051817.

[20] Y. Gao *et al.*, "Joint optimization of depth and ego-motion for intelligent autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7390–7403, 2023, doi: 10.1109/TITS.2022.3159275.

[21] S. Peng, D. Chu, G. Li, L. Lu, and J. Wang, "EPN: An ego vehicle planning-informed network for target trajectory prediction," *arXiv preprint*, vol. arXiv:2412, 2025.

[22] Z. Zhiliang, L. Peng, X. Wang, Z. Zhibo, W. Tianhua, and C. Dandan, "Research on active safety testing based on China new car assessment program (C-NCAP)," in *Second International Conference on Physics, Photonics, and Optical Engineering (ICPPOE 2023)*, 2024, vol. 13075, p. 2. doi: 10.1117/12.3025949.

[23] L. Deng, W. Ni, T. Zhou, Y. Yu, and L. Zhai, "Analysis of vehicle assisted lane change system and autonomous lane change model," in *2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, Dec. 2022, pp. 1–6. doi: 10.1109/ICERECT56837.2022.10060490.
[24] J. R. Van Aken, "Drawing ellipses and elliptical arcs with piecewise cubic Bezier curve approximations," *arXiv preprint arXiv:2407.17675*, 2024.
[25] M. Hanafy, M. M. Gomaa, M. Taher, and A. M. Wahba, "Development of a technology for car's auto-parking using swarm search-based fuzzy control system," *International Journal of Modelling, Identification and Control*, vol. 17, no. 1, p. 85, 2012, doi: 10.1504/IJMIC.2012.048643.

APPENDIX A. LIST OF SYMBOLS

Table A.1. List of symbols

Parameter	Definition
R_0	Initial value of Bezier curve $r(t)$ @ $t = 0$
R_1	First control point of Bezier curve $r(t)$ @ $t = \frac{1}{3}$ $R_1 = r(t=0) + \frac{\dot{r}(t=0)}{3}$
R_2	Second control point of Bezier curve $r(t)$ @ $t = \frac{2}{3}$ $R_2 = r(t=1) - \frac{\dot{r}(t=1)}{3}$
R_3	Final value of Bezier curve $r(t)$ @ $t = 1$
$x_p(t)$	Third order polynomial Bezier curve function on t, presenting the value in x-axis of local Bezier coordinates
$y_p(t)$	Third order polynomial Bezier curve function on t, presenting the value in y-axis of local Bezier coordinates
$y_c(t)$	Circular arc function on t, presenting the value in y-axis of local Bezier coordinates
X_0	Initial value of Bezier curve $x_p(t)$ @ $t = 0$
X_3	Final value of Bezier curve $x_p(t)$ @ $t = 1$
Y_0	Initial value of Bezier curve $y_p(t)$ @ $t = 0$
Y_3	Final value of Bezier curve $y_p(t)$ @ $t = 1$
M	The final value of $x_p(t)$ in local Bezier coordinates; it is the Euclidean distance between the initial and goal frames
$\dot{x}_p(t)$	The first derivative of $x_p(t)$ with respect to t
$\dot{y}_p(t)$	The first derivative of $y_p(t)$ with respect to t
θ_r	The orientation of the vehicle in the local Bezier Coordinates (x_p, y_p)
θ_{r_0}	Initial value of θ_r at $t = 0$
θ_{r_f}	Final value of θ_r at $t = 1$
t_0	$\tan(\theta_{r_0})$
t_f	$\tan(\theta_{r_f})$
θ^*	Both the initial and goal orientation angles in the local coordinates for a circular arc ($\theta_{r_0} = -\theta_{r_f} = \theta^*$).
$\theta_c(t)$	Orientation angle of the tangent to the circular arc in the local Bezier coordinates as a function on t.
α	Rotation angle between the local Bezier coordinates (x_p, y_p) and the global ego-vehicle coordinates (x, y)
θ_0	Initial orientation of the ego-vehicle in the global ego-vehicle coordinates (x, y), $\theta_0 = \theta_{r_0} + \alpha$
θ_f	Final orientation of the ego-vehicle in the global ego-vehicle coordinates (x, y), $\theta_f = \theta_{r_f} + \alpha$
$\Delta\theta$	$\lvert\theta_f - \theta_0\rvert$
b	Location of the control points of $x_p(t)$ with respect to the boundaries, the values of X_1 and X_2 are $b\,M$ and $(1-b)\,M$, respectively
ϵ	The factor of the radius by which the tangential distance from the initial and goal positions are considered to calculate the control points for circular Bezier
R	Radius of Rotation of the intended Circular Path
X_{max}	Range limit of ego-vehicle path prediction in its longitudinal axis
$x(t)$	Generated Bezier path in the global coordinates, in the longitudinal axis of the ego-vehicle
$y(t)$	Generated Bezier path in the global coordinates, in the lateral axis of the ego-vehicle
$y = a\,x^3 + b\,x^2 + c\,x + d$	Third order polynomial path obtained from curve fitting with the golden reference path
w	Width of Ego-Vehicle
$x_{p_R}(t)$	Third order polynomial Bezier curve function with respect to t, presenting the value in x-axis of local Bezier coordinates for right corner of front bumper
$y_{p_R}(t)$	Third order polynomial Bezier curve function with respect to t, presenting the value in y-axis of local Bezier coordinates for right corner of front bumper
$x_{p_L}(t)$	Third order polynomial Bezier curve function with respect to t, presenting the value in x-axis of local Bezier coordinates for left corner of front bumper
$y_{p_L}(t)$	Third order polynomial Bezier curve function with respect to t, presenting the value in y-axis of local Bezier coordinates for left corner of front bumper
$A_{y_{max}}$	The maximum limit of lateral acceleration for a normal driving scenario

APPENDIX B. OPTIMIZED APPROACH TO PRESENT CIRCULAR PATH WITH BEZIER ONE

In order to evaluate how closely a Bezier curve can approximate a truly circular arc, we define both the Bezier path and the ideal circular path mathematically as follows:

$$y_p(t) = 3\epsilon R sin(\theta^*)[t - t^2] \quad \text{(B.1)}$$

$$y_c(t) = R[cos(\theta_c(t)) - cos(\theta^*)] \quad \text{(B.2)}$$

As shown in Figure B.1, (B.1) represents Bezier path while (B.2) represents Circular one.

$$sin(\theta_c(t)) = \frac{M}{R}(0.5 - t), cos(\theta_c(t)) = \frac{\sqrt{R^2 - M^2(0.5-t)^2}}{R} \quad \text{(B.3)}$$

$$sin(\theta^*) = \frac{M/2}{R}, cos(\theta^*) = \frac{\sqrt{R^2 - (\frac{M}{2})^2}}{R} \quad \text{(B.4)}$$

$$y_p(t) = \frac{3}{2}\epsilon M t(1-t) \quad \text{(B.5)}$$

So,

$$y_c(t) = \sqrt{A^2 + M^2 t(1-t)} - A \quad \text{(B.6)}$$

where $\sqrt{R^2 - \left(\frac{M}{2}\right)^2} = A$

By considering only half of the path from $t = 0$ to $t = 0.5$ due to symmetry,

$$Diff(\epsilon) = \int_0^{0.5}[y_c(t) - y_p(t)]^2 dt = \int_0^{0.5} {y_c}^2(t)dt + \int_0^{0.5} {y_p}^2(t)dt - 2\int_0^{0.5} y_c(t)y_p(t)dt \quad \text{(B.7)}$$

By substituting by (B.5) and (B.6) in (B.7),

$$\int_0^{0.5} {y_p}^2(t)dt = \frac{3}{80}M^2\epsilon^2 \quad \text{(B.8)}$$

$$2\int_0^{0.5} y_c(t)y_p(t)dt = 3M[B - \frac{A}{12}]\epsilon \quad \text{(B.9)}$$

where $\int_0^{0.5} t(1-t)\sqrt{A^2 + M^2 t(1-t)} = B$

$$\frac{dDiff(\epsilon)}{d\epsilon} = \frac{3}{40}M^2\epsilon - 3M\left[B - \frac{A}{12}\right] = 0 \quad \text{(B.10)}$$

$$\frac{d^2 Diff(\epsilon)}{d\epsilon^2} = \frac{3}{40}M^2 > 0 \rightarrow \text{Local Minimum} \quad \text{(B.11)}$$

$$\epsilon = \frac{40}{M}\left[B - \frac{A}{12}\right] \quad \text{(B.12)}$$

Approximate approach:

$$\epsilon_1 = \frac{4}{3}\left[\frac{1 - cos(\theta^*)}{sin(\theta^*)}\right] = \frac{4}{3}\left[\frac{R - \sqrt{R^2 - \left(\frac{M}{2}\right)^2}}{M/2}\right] \quad \text{(B.13)}$$

Optimized approach:

$$\epsilon_2 = \frac{40}{M}\left[\int_0^{0.5} t(1-t)\sqrt{R^2 - \left(\frac{M}{2}\right)^2 + M^2 t(1-t)}dt - \frac{\sqrt{R^2 - \left(\frac{M}{2}\right)^2}}{12}\right] \quad \text{(B.14)}$$

The form obtained in (B.13) used in (19) is an approximate presentation of (B.14). To illustrate the negligible difference between the Approximate and Optimized approaches, consider the following common numerical example:

For $M = 2m, R = 4m$; the resulting terms are $\epsilon_1 = 0.16935$, and $\epsilon_2 = 0.16974$. Hence, $\epsilon_1 \approx \epsilon_2$ with a difference of less than 0.0004.

This example demonstrates that the curvature error introduced by using the midpoint approximation is negligible. Therefore, solving the problem analytically with a circular path assumption (as discussed in

section 4.1) offers a sufficiently accurate and far simpler approach highly relevant for embedded targets constraints rather than the complex optimized complex one and its needed numerical integration every cycle.

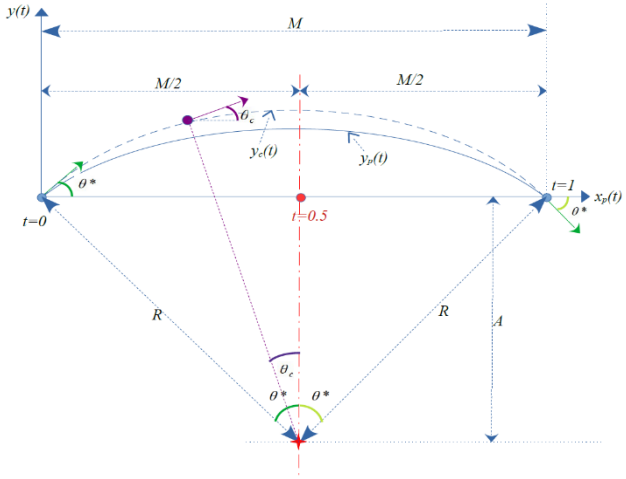

Figure B.1. Closest Bezier path to circular one

APPENDIX C. SOLVING THE THIRD-ORDER EQUATION

$$x^3 + a\,x^2 + b\,x + c = 0 \tag{C.1}$$

Step 1:

$$p = b - \frac{a^2}{3} \tag{C.2}$$

$$q = \frac{2a^3}{27} - \frac{ab}{3} + c \tag{C.3}$$

$$\Delta = \frac{q^2}{4} + \frac{p^3}{27} \tag{C.4}$$

Step 2:
Case I: $(\Delta > 0)$ only one real solution

$$x = \left(-\frac{q}{2} + \sqrt{\Delta}\right)^{\frac{1}{3}} + \left(-\frac{q}{2} - \sqrt{\Delta}\right)^{\frac{1}{3}} - \frac{a}{3} \tag{C.5}$$

Case II: $(\Delta = 0)$ repeated roots

$$x_1 = -2\left(\frac{q}{2}\right)^{\frac{1}{3}} - \frac{a}{3} \tag{C.6}$$

$$x_2 = x_3 = \left(\frac{q}{2}\right)^{\frac{1}{3}} - \frac{a}{3} \tag{C.7}$$

Case III: $(\Delta < 0)$ three real solutions

$$x_1 = \frac{2}{\sqrt{3}}\sqrt{-p}\,\sin\left(\frac{1}{3}\sin^{-1}\left(\frac{3\sqrt{3}q}{2(\sqrt{-p})^3}\right)\right) - \frac{a}{3} \tag{C.8}$$

$$x_2 = -\frac{2}{\sqrt{3}}\sqrt{-p}\,\sin\left(\frac{1}{3}\sin^{-1}\left(\frac{3\sqrt{3}q}{2(\sqrt{-p})^3}\right) + \frac{\pi}{3}\right) - \frac{a}{3} \tag{C.9}$$

$$x_3 = \frac{2}{\sqrt{3}}\sqrt{-p}\,\cos\left(\frac{1}{3}\sin^{-1}\left(\frac{3\sqrt{3}q}{2(\sqrt{-p})^3}\right) + \frac{\pi}{6}\right) - \frac{a}{3} \tag{C.10}$$

BIOGRAPHIES OF AUTHORS

Hanan H. Hussein received the BSc., MSc., and PhD in Electronics and Electrical Communications Engineering at Ain Shams University, Cairo, Egypt, in 2008, 2013 and 2020. Since 2014, she is researcher in Electronics Research Institute (ERI), Computers and Systems Department. She is the author of six conference papers, thirteen journal articles, and an editor of 5G book. She had the excellence reward from ERI in 2019, 2020. She won best Ph.D. prize. She is an HCIA-Datacom Huawei Instructor in ICT academy. Her research interests include wireless telecommunication, Wireless sensor networks, Ad hoc networks, MAC protocols, IoT, cellular networks (4G, 5G, and 6G), Vehicles communication, Device-to-Device, and ADAS systems. She can be contacted at email: hananhussein@eri.sci.eg.

Ahmed Atef Valeo Advanced Engineer. B.Sc. in Information Technology and Computer Science from Faculty of Computer and Information, Cairo University in 2016. Research track in Automotive industries dedicated to solutions for: Advanced Driving-Assistance Systems (ADAS), Constrained Motion Planning. Honored as Valeo Inventor with multiple contributions in both Platform and Customer specific developments including Product Adoptions. He can be contacted at email: ahmed.atef@valeo.com.

Mohamed Hanafy Radwan Valeo Senior Expert System Engineer in Advanced Driving-Assistance Systems - Senior Chief Software Engineer. PhD, MSc, and BSc in Computer and Systems Engineering from Faculty of Engineering, Ain-Shams University in 2015, 2012, and 2006, respectively. Research track in both Silicon and Automotive industries dedicated on intelligent solutions for: automated-digital-design-verification, advanced driving-assistance systems (ADAS), vehicle's communication systems (Digital Mobility). Honored as Valeo Top-Maker and Valeo Top-Inventor with several Proof-of-Concepts and P3/P2 Developments leading to several Product Adoptions, and 81 ADAS Inventions with 51 Filings leading to 35 Granted Patents in 2016-2025. This is besides several scientific prestigious Publications. He can be contacted at email: mohamed.radwan@valeo.com.