Chapter 7

Path Planning Based on Parametric Curves

Lucía Hilario Pérez, Marta Covadonga Mora Aguilar,
Nicolás Montés Sánchez and
Antonio Falcó Montesinos

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.72574

Abstract

Parametric curves are extensively used in engineering. The most commonly used parametric curves are, Bézier, B-splines, (NURBSs), and rational Bézier. Each and every one of them has special features, being the main difference between them the complexity of their mathematical definition. While Bézier curves are the simplest ones, B-splines or NURBSs are more complex. In mobile robotics, two main problems have been addressed with parametric curves. The first one is the definition of an initial trajectory for a mobile robot from a start location to a goal. The path has to be a continuous curve, smooth and easy to manipulate, and the properties of the parametric curves meet these requirements. The second one is the modification of the initial trajectory in real time attending to the dynamic properties of the environment. Parametric curves are capable of enhancing the trajectories produced by path planning algorithms adapting them to the kinematic properties of the robot. In order to avoid obstacles, the shape modification of parametric curves is required. In this chapter, an algorithm is proposed for computing an initial Bézier trajectory of a mobile robot and subsequently modifies it in real time in order to avoid obstacles in a dynamic environment.

Keywords: path planning, mobile robots, parametric curves, Bézier curves

1. Introduction

In the last years, intelligent vehicles have increased their capacity up to the point of being able to navigate autonomously in structured environments. Implementations, such as Google [1] (with more than 700,000 hours of autonomous navigation in different scenarios), are an example of the effort made in this area. However, there is still a long way to go until we found real

IntechOpen

© 2018 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

autonomous cars on the roads, as there are both technical and legal problems involved [2, 3]. The intelligent system is composed of three different groups and subgroups: acquisition and perception, decision and actuation-control.

Although the vast majority of the literature often depicted the problems by focusing mainly on these groups or subgroups of processes, functionality in intelligent vehicles, or in mobile robots in general, cannot be conceived as composed of separate blocks, and therefore, a sufficiently efficient system can only be achieved if all the systems work in unison.

This chapter is devoted to the use of parametric curves in the field of robotics. Parametric curves are mainly used in the decision block when the path is defined. However, they are also employed in other blocks, and some of their properties are beneficial for other processes.

Focusing on the decision-making block, the "path planning" or the design of the path to follow has been the subject of study in the last decades, where many authors divide the problem into global and local path planning. On the one hand, the global path planning generates an overall path composed of a set of points to be followed, covering large distances and considering static obstacles in the environment. On the other hand, the local path planning constructs a short path with much more precision, even in continuous form, taking into account unexpected obstacles that may appear.

In general, path planning techniques can be grouped into four large groups: graph search, sampling, interpolating and numerical optimization, see [3]:

- *Graph search-based planners* search a grid for the optimal way to go from a start point to a goal point. Algorithms, such as Dijkstra, A-Start (A *) and its variants Dynamic A* (D*), field D*, Theta*, etc., have been extensively studied in the literature.

- *Sampling-based planners* try to solve the search problem restricting the computational time. The idea is to randomly explore/sample the configuration space, looking for connections between source and destination. The main problem is that the solution is suboptimal. The most common techniques are the probabilistic roadmap method (PRM) and the rapidly exploring random tree (RRT).

- *Interpolating curve planners* try to insert a new group of data within the previously defined data group. In other words, both graph search and sampling-based planners are global planners that provide a rough approximation of the solution. In this case, it is a matter of interpolating this group of points. At this stage, the design of the trajectory is when the properties of continuity, smoothness and geometrical restrictions of the vehicle, among others, intervene. Computer-aided geometric design (CAGD) techniques are generally used to smooth the gross path provided by the global planner. The use of lines and circles is usually employed as a first solution, with Dubin's curves defined when the vehicle moves forward and Reed and Sheep's curves when the vehicle moves backward. The clothoid appears as a solution to the discontinuity in curvature between the line and the circle since, by definition, it has a constant relationship between the length of the arc and its curvature. The polynomial curves are another alternative to the previous ones. The modification of its coefficients allows taking into account, among others, the adjustment of positions, curvature restrictions, etc.

- *Numerical optimization* is generally used to minimize or maximize a numerical function that depends on different variables such as smoothness, continuity, velocity, acceleration, jerk, curvature, etc.

In [3], the use of parametric curves is included in the category of *interpolating curve planners.* The most commonly used parametric curves in robotics are Béziers, B-splines, rational Bézier curves (RBCs), and non-uniform rational B-splines (NURBSs). A summary of their properties can be low computational cost, intrinsic softness, easy malleability through control points, and universal approximation. For these reasons, parametric curves are not only relevant as interpolators, but also recently they are being used in combination with many other algorithms that have effects on all the other blocks of an intelligent system [3].

The chapter is organized as follows. Section 2 provides a mathematical definition of the most used parametric curves as well as a description of their properties (Bézier, B-spline, RBC, and NURBS). Section 3 offers a state of the art of the use of parametric curves in robotics and an overview of current trends. Along the lines of the new trends in the use of these curves, Section 5 proposes a method of deformation of parametric curves aimed at modifying the trajectory in real time in order to avoid collisions. Section 6 presents the reader the conclusions.

2. Definitions: parametric curves

Curves in both space and plane are a part of the geometry necessary to represent certain shapes in different areas. Curves arise in many applications, such as art, industrial design, mathematics, architecture, engineering, etc.

2.1. Different ways of defining a curve. Advantages and disadvantages

There are different ways of defining a curve: implicit, explicit, and parametric.

2.1.1. Implicit and explicit expression of a curve

The coordinates (x, y) of the points of an implicitly defined plane curve verify that:

$$F(x,y)=0 \tag{1}$$

for some function F. If the curve is in R^3, then the curve must satisfy these two conditions simultaneously:

$$F(x,y,z)=0 \text{ and } G(x,y,z)=0 \tag{2}$$

The explicit representation of a curve clears one of the variables as a function of the other. In the plane, the coordinates (x,y) of the points in the curve explicitly defined satisfy either.

$$y = f(x) \text{ or } x = g(y) \tag{3}$$

In the case of a curve in the space, its explicit form could be defined as:

$$x = f(z) \text{ and } y = g(z) \tag{4}$$

2.1.2. Parametric expression of a curve

In this case, the coordinates of a parametric curve are expressed as a function of a parameter, for example, u. The definition of a curve defined in R^n could be done as in (5), where functions α_i are the coordinate functions or component functions. The image of $\alpha(u)$ is called the trace of α and α(u) is the parametrization of α.

$$\alpha : [a,b] \to \Re^n \,/\, \alpha(u) = (\alpha_1(u), \ldots, \alpha_n(u)); u \in [a,b] \tag{5}$$

Parametric curves are the most used in computer graphics and geometric modeling because the curve points are calculated in a simple way. In contrast, the calculation of the points through the implicit expression is much more complex.

Within the parametric curves, it is possible to differentiate between polynomial curves and rational curves. Polynomial curves are those whose component functions are polynomials, and rational curves are those expressed as the quotient of polynomials. The representation in the form of parametric curves allows a great variety of curves, some known, some strange, some complex and others surprising for their symmetry and beauty.

The advantageous properties of the parametric curves that make them widely used are intuitivity, flexibility, affine-invariant, fast computation, and numerical stability.

In order to model complicated shapes or surfaces, it is necessary to introduce a way of representing curves based on a polygon. From this idea, the most used parametric curves arise in computer-aided geometric design (CAGD): Bézier, B-splines, RBC, and NURBS. **Figure 1** shows a schematic of the most important curves in CAGD. It can be seen how NURBS are the most general curves, and Bézier are the most particular ones. Among them, Bézier is the simplest, possessing properties that make them be the most extensively used.

2.2. Most common parametric curves: Bézier, B-spline, NURBS, and RBC

2.2.1. Bézier curves

Bézier curves arose as a result of the car modeling in both Renault and Citroën companies, by the engineers Pierre Bézier and Casteljau. The simplicity in the manipulation of these curves makes their use and application widespread.

The popularity of the Bézier curves is due to their numerous mathematical properties that facilitate their manipulation and analysis. Moreover, their use does not require great mathematical knowledge, which is very interesting for designers who shape objects.

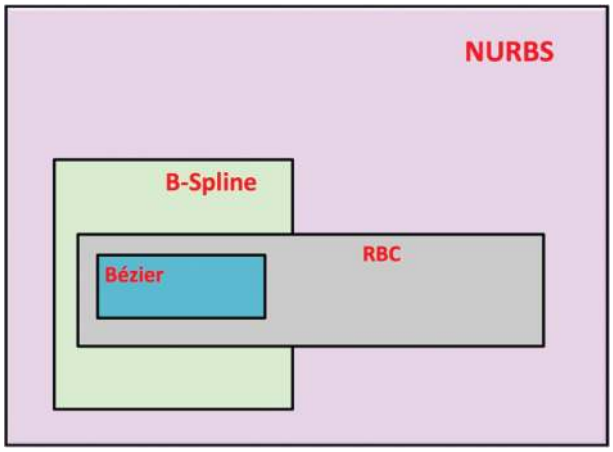

Figure 1. Classification of the most important curves in CAGD.

A Bézier curve of degree n is specified by a sequence of $(n + 1)$ control points, and its explicit expression is (6). The polygon that joins the control points is called the control polygon, and the functions or bases used are the Bernstein polynomials $B_{i,n}(u)$, defined in (7).

$$\alpha(u) = \sum_{i=0}^{n} \mathrm{P}_i B_{i,n}(u); u \in [0,1] \tag{6}$$

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i}; i = 0, \square\ , n \tag{7}$$

The dimension of the vector containing the control points is related to the dimension of the space where the curve is represented.

2.2.2. *Rational Bézier curves*

A conic is a curve obtained as the intersection of a plane with the surface of a double cone. There are three types of irreducible conics: hyperbola, parabola, and ellipse. Parabolas can be parameterized by polynomial functions, but hyperbolas and ellipses need rational functions such as RBC. The explicit definition of an RBC is (8), where P_i are the control points, $B_{i,n}(u)$ are the Bernstein bases, and ω_i are weights associated with each control point. These weights allow a new way of modifying the curve.

$$\alpha(u) = \frac{\sum_{i=0}^{n} \omega_i \mathrm{P}_i B_{i,n}(u)}{\sum_{i=0}^{n} \omega_i B_{i,n}(u)}; u \in [0,1] \tag{8}$$

2.2.3. B-spline curves

B-splines are polynomial curves defined in pieces, continuously differentiable up to a prescribed order. The name spline is a word that means "elastic slats". These slats were used by craftsmen to create curves describing the surfaces to be built, such as boat hulls and aircraft fuselages. Constrained by weights, these elastic slats or splines assume a shape that minimizes their elastic energy.

B-spline curves were developed to overcome the limitations of Bézier curves: the need for a local control of the curve, the difficulty in imposing C2 continuity and the fact that a number of control points of a Bézier curve imposes its degree.

Analogous to the definition of a Bézier curve, a B-spline curve of degree k (or $k + 1$ order) is expressed in (9) as an affine combination of certain control points P_i, where $N_{i,k}$ are polynomial functions by pieces with finite support of order k (degree k-1, meaning that they are zero out of a finite interval) that satisfy certain conditions of continuity. Each of these functions can be calculated using the Cox-de-Boor recursive formulas.

$$\alpha(u) = \sum_{i=0}^{n} \mathrm{P}_i N_{i,k}(u) \tag{9}$$

B-splines can be defined by a recurrence relationship; simplicity is considered a double infinite sequence of simple nodes such that for all i. B-splines are then defined through the following recurrence relationship.

For the sake of simplicity, a double infinite sequence of simple nodes a_i is considered such that $a_i < a_{i+1}$ for all i. Then, the B-splines $N_{i,k}$ are then defined through the recurrence relationships (10) and (11).

$$N_{i,0}(u) = \begin{cases} 1 \text{ if } u \in [a_i, a_{i+1}[\\ 0 \text{ otherwise} \end{cases} \tag{10}$$

$$N_{i,k}(u) = \frac{t - a_i}{a_{i+k-1} - a_i} N_{i,k-1}(u) + \frac{a_{i+k-t}}{a_{i+k} - a_{i+1}} N_{i+1,k-1}(u) \tag{11}$$

2.2.4. Non-uniform rational B-spline curves

Rational B-spline curves are obtained in a similar way as the RBCs from Bézier curves. The definition of a NURBS curve is:

$$\alpha(u) = \frac{\sum_{i=0}^{n} \mathrm{P}_i \omega_i N_{i,k}(u)}{\sum_{i=0}^{n} \omega_i N_{i,k}(u)} \tag{12}$$

PARAMETRIC CURVES IN CAGD	Polinomials curves (Non rationals)		Rationals curves (conics)	
	Bézier	B-Spline	Rational Bézier	NURBS
Convex Hull Property	Yes	Yes (only local)	Yes (if the weights are positive)	Yes (only local if the weights are positive)
Interpolation of first and last control point	Yes	No	Yes	No
Tangency to the control polygon first and last point	Yes	No	Yes	No
Invariant under affine transformations	Yes	Yes	Yes	Yes
Recurrent bases	No	Yes	No	Yes
Local control	No	Yes	No	Yes
Global control	Yes	No	Yes	No

Table 1. Comparison of the properties of parametric curves.

2.3. Comparison of properties

When dealing with curves, their representation is important, but their shape manipulation is a key factor in their usability. The object to be modeled will determine the type of parametric curve chosen, depending on the properties required. In **Table 1**, we can see a comparison of the properties of the parametric curves in CAGD. In the following section, some of the most relevant works in mobile robots using parametric curves are described.

3. Use of parametric curves in robotics: state of the art

3.1. Generation of trajectories of mobile robots through parametric curves

Predicting the movement of a robot is important as it implies the computation of a proper path that meets the kinematic and dynamic properties of the robot. Simply moving a mobile robot from an initial position (x_i, y_i, θ_i) to a final position (x_g, y_g, θ_g) safely implicates many research fields, which are involved in the generation of efficient path planning algorithms.

Many researchers consider parametric curves very useful in the construction of trajectories of wheeled robots, due to their advantageous properties able to improve trajectories produced by path planning techniques.

3.2. Trajectories of mobile robots defined by Bézier curves

The first relevant publications in robotics using Bézier curves are published in 1997 and 1998 [29, 30]. These works combine path planning and reactive control for a non-holonomic mobile robot introducing the concept of "Bubble Band" (bubble path). With an appropriate metric, bubbles are connected with Bézier curves, generating a path. These bubbles are the maximum free space reachable in any direction without risk of collision. This is due to the property of the convex hull and implies that if the control points are within the bubble, then the path approximation remains within the bubble. The planner, using a model of the environment, generates an initial path connecting the start and goal positions that may not be adequate. Next, the proposed algorithm generates a sequence of bubbles connecting both ends and replacing the original path, the *bubble band*. This band is exposed to the forces in the environment, and as a consequence, the band is modified.

In 2001, the concept of "bubble band" is used in [31]. In this case, dynamic obstacles are introduced in the environment. Simultaneously, in [32], also Bézier curves are used for local path planning. An initial path is computed using the generalized Voronoi graph (GVG) theory, which is mildly deformed maximizing the evaluation of a function. Candidates obtained as smooth paths are expressed with Bézier curves.

In 2003, a touchscreen was introduced in [33] to control a mobile robot, avoiding obstacles in real time. In this work, two algorithms are developed: the first one extracts a succession of important points, and the second one generates a path using cubic Bézier curves.

In 2007, the work in [34] introduces Bézier curves in cooperative collision avoidance for several mobile non-holonomic robots and is based on the previous contributions [35, 36]. Two tasks are developed: first, path planning based on Bézier curves for each individual robot in order to obtain its final position and, second, computation of an optimal path that minimizes a "penalty" function that accounts for the sum of the maximum times subject to the distances between the robots.

In 2008, [37] presents a preliminary framework that generates space trajectories for multiple unmanned aerial vehicles (UAVs) using 3D Bézier curves. The algorithm solves a constrained optimization problem in order to generate the trajectories. In this case, the optimization function penalizes an excessive length, as the shortest path is required, and the restrictions are the distances between the multiple UAVs. The system is non-linear, and numerical methods are applied to solve it.

It is worth mentioning the work of Choi et al. [38–46] related to the computation of trajectories of mobile robots designed from Bézier curves. In many of the publications, a constrained optimization problem is raised, where the function to be optimized is the curvature of a Bézier curve.

Finally, [47] presents a methodology based on the variation of the RRT that generates suitable trajectories for autonomous vehicles with holonomic constraints in environments with

obstacles. This algorithm is based on the use of seventh-order Bézier curves that connect the vertices of the tree. In this way, the generated paths meet the main kinematic constraint of the vehicle: the smoothness of the acceleration is guaranteed for the entire path by controlling the values of the curvature of the endpoints of each Bézier curve composing the tree. The proposed algorithm provides a rapid convergence to the final result. In addition, the number of vertices of the tree is reduced because the method allows the connections between the vertices of the tree with an unlimited range. The properties of seventh-order Bézier curves are also used to avoid static obstacles in the environment. This method was simulated with a small UAV. Since then, B-splines and Beziers curves have been used to generate search trees by a large number of researchers, see [7].

Recent efforts are being made to merge Bézier curves with numerical optimization, [4, 5]. In these works, a teleoperation is carried out where the operator indicates some points. The proposed algorithm calculates the path to continuity of curvature C1 and C2. In [6], something similar is proposed: nodes/points are initially generated between the start and the goal (collision-free) and then are joined by cubic Bézier curves with curvature constraints. Finally, cubic Bézier curves are used in [8] to solve the problem of roundabouts for automated vehicles: entry, departure, and crossing.

3.3. Trajectories of mobile robots defined by B-spline

In 1989, B-spline curves were incorporated in the design of robot trajectories. In [13], segments were added with the aim of generating the entire path near the desired one. This new trajectory did not go through the exact points. Later, in 1994, the work in [14] used B-splines for path planning but adding a temporal variable. In this case, the speed of the robot was controlled by the same B-spline. The same year, in [15], a fuzzy controller is designed to emulate spline curves for generating smooth motion trajectories. In 1999, the work [16] also used a B-spline curve to calculate the trajectory of a mobile robot by generating many points from a spline for the robot to follow them in the form of succession. Additionally, in [17], kinematic constraints were introduced in the path planning using B-spline curves to find the optimal temporal trajectory in a static environment.

Lately, in 2007, the works [18–20] developed a method to solve the path planning problem using cubic splines to avoid the obstacles. This method iteratively refined the path to be followed in order to obtain in real time a collision-free feasible path in unstructured environments. In [20], the path planning implementation based on B-spline is detailed. The use of splines allows to restrict the polynomials since the first derivative of $P_1,\ldots, P_{n\text{-}1}$ is continuous across the entire boundary. In addition, some constraints can be introduced on the first and last points to force a particular value of the derivative. These characteristics of the splines offer many advantageous properties to plan a suitable path. If a value of the derivative is imposed, a path can be generated starting from a specific position and having a direction imposed by the value of the derivative. Therefore, they can be generated and initialized from the current position and direction of the vehicle. The first derivative is proportional to the direction of the vehicle, then a non-continuous derivative could be obtained and, as a consequence, a non-feasible path for that type of vehicle. As the second derivative is proportional to the direction

of the angle, some discontinuities could force the vehicle to stop at each control point to adjust its direction.

B-splines curves allow an easy construction of smooth paths through control points. In order to avoid obstacles, control points are introduced near them, and methods are developed to move these control points away from the obstacles and move them to the free space.

Earlier methods also worked with splines to generate smooth paths also avoiding the surrounding obstacles [20, 21]. Nevertheless, these previous methods had a high computational cost when evaluating the overall path. In [18], the computational time and the viability of one of these algorithms are analyzed, since it is executed with an iterative method. Monte Carlo simulations indicate a high degree of success for complex environments. The running time is also measured and increases with the complexity of the environment. Finally, in [19], experimental results are provided. The main disadvantage of this algorithm is that the obstacle-free path is computed by means of an iterative method. Thus, the computational time will always increase with respect to other non-iterative methods.

A large number of researchers have also used parametric curves, and particularly B-splines and Béziers, to generate search trees as in [7].

3.4. Trajectories of mobile robots defined by NURBS

This type of parametric curve is used in the reconstruction of trajectories with the aim of generating smooth paths that approximate the real movement of the robot. In [22], the advantages and disadvantages of the NURBS curves are highlighted, providing a detailed study of their properties. In the field of robotics, the work [23] highlights advantageous properties of NURBS for path planning in both 2D and 3D.

In other works, such as [24–26], NURBS curves approximate or describe the path described by a robot arm PUMA 560. Programming by Demonstration is used to program the behavior of the robot, a good solution to automatically transfer the human knowledge to a robot. However, the NURBS trajectory does not guarantee the obstacle avoidance.

More recently, in [9–12] a predefined NURBS curve is used to improve its properties adjusting the weights.

3.5. Trajectories of mobile robots defined by RBC

In [27], an off-line methodology is presented to approximate a Clothoid (Fresnel integrals) to an RBC. Subsequently, [28] presents a method to obtain trajectories in real time with Clothoids. To do this, two steps are involved: the off-line definition of approximations of Clothoids with RBCs and the generation of online paths by scaling, rotating and moving the previous off-line curves. One of the advantages of this method is the off-line calculation since it considerably reduces the computational time. Throughout the process, the weight coefficients and control points remain invariant. In this work, it is guaranteed that an RBC has the same behavior as a Clothoid using a low order for the curve.

3.6. Current trends in the use of parametric curves in robotics

This comprehensive study of the use of the parametric curves evidences its importance in the design of trajectories of a mobile robot. They are not only used for interpolating points in the global map but also being integrated into global planners and in numerical optimizations. Although non-rational curves have a lower approximation capacity, researchers prefer them for their simplicity and easy manipulation. Among them, we must highlight the Bézier curves, which are the most used.

However, when the parametric curve is used as an approximation, the use of rational curves is significantly greater, as in the approximation to the clothoid and the circle. Recently, predefined rational curves are being used, where only the weights are modified. This can transform rational curves into manageable curves in comparison to non-rational curves.

Along the lines of merging the use of parametric curves with other types of algorithms in an intelligent navigation system, it is not only important to define the path of the robot, but also to avoid obstacles in the environment. Consequently, the initial trajectory must be modified in real time so that the mobile robot avoids the possible dynamic obstacles that may appear. In this sense, the Bézier trajectory deformation (BTD) algorithm, described in the next section, introduces the possibility of deforming a Bézier curve through a vector field, which can be used in mobile robotics. The temporal parameter is introduced in the Bézier curve to transform it into a path and a vector field is needed to modify the initial path.

4. Properties of parametric curves and its applications in robotics

In mobile robotics, two main needs have arisen when dealing with path planning of a mobile robot: definition of the initial path to follow and the possibility of modifying it in the presence of dynamic obstacles.

In the next paragraphs, the BTD algorithm is described [48, 49], which solves the abovementioned needs. It offers the possibility of defining the trajectory of a mobile robot through a Bézier curve and then modifies it by means of the repulsive forces derived from a predictive potential field (PF) method. Reactive methods or potential field methods generate obstacle-free paths for the robot. In these methods, the movement of the robot is determined by repulsive forces associated with obstacles and attractive forces associated to the goal position of the mobile robot. In this work, the potential field projection method (PFP) has been used [50, 51].

The set of discrete points provided by the posture prediction of the mobile robot is considered as *initial points* S_i of the original Bézier curve. These points belong to a reference path in the BTD algorithm. Subsequently, the set of repulsive forces obtained by the PFP is transformed into displacements by a dynamic particle model, which generates *endpoints* T_i that determines the modification of the original Bézier trajectory with the BTD. A modified Bezier trajectory free of obstacles is obtained that passes through the endpoints, as displayed in **Figure 2**.

The definition of the BTD algorithm requires two steps:

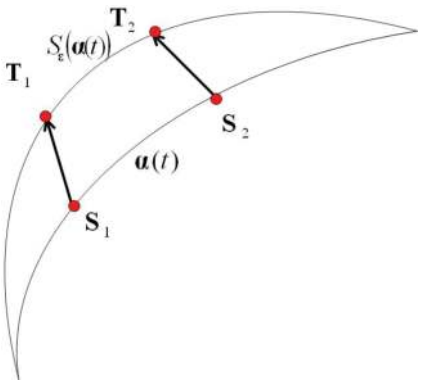

Figure 2. Deformation of a Bézier trajectory through a field of vectors.

a. *Definition of the trajectory using a Bézier curve*

A Bézier curve has a non-dimensional intrinsic parameter, u, as defined in (5). Since the Bézier curve represents the path of the robot, the intrinsic parameter must be defined as a temporal variable so that the position of each curve (robot position) is associated with an instant of time $t \in [t_0, t_f]$, where t_0 and t_f represent the initial and final times of the trajectory, respectively. The definition of the initial Bézier trajectory is (13), where n is the order, P_i are the control points and $B_{i,n}(t)$ are the Bernstein bases defined in (14). To avoid loops in the Bézier curve, second-order curves are used.

$$\alpha(t) = \sum_{i=0}^{n} \mathrm{P}_i \cdot B_{i,n}(t); t \in [t_0, t_f] \tag{13}$$

$$B_{i,n}(t) = \binom{n}{i} \left(\frac{t_f - t}{t_f - t_0}\right)^{n-i} \left(\frac{t - t_0}{t_f - t_0}\right)^{i}; i = 0, \cdots, n \tag{14}$$

The initial Bézier trajectory will be deformed in order to avoid the surrounding obstacles, by modifying the position of the control points from the initial position to the new one imposed by the PFP obstacle avoidance algorithm. The displacement of each control point $P_i$is denoted as ε_i, so that the vector $\varepsilon = [\varepsilon_0, \cdots, \varepsilon_n]$ is the displacement of all the control points defining the Bézier trajectory, also known as the perturbation vector of the deformed curve. The new modified Bézier trajectory $S_\varepsilon(\alpha(t))$ is defined in (15) and, consequently, the optimizing function used to solve the problem is defined as (16), where the vector ε is computed as in [49].

$$S_\varepsilon(\alpha(t)) = \sum_{i=0}^{n} (\mathrm{P}_i + \varepsilon_i) \cdot B_{i,n}(t); t \in [t_0, t_f] \tag{15}$$

$$\min_{\varepsilon} \int_{t_0}^{t_f} \| S_\varepsilon(\alpha(t)) - \alpha(t) \|_2^2 dt \tag{16}$$

This objective function minimizes changes in the shape of the initial Bézier trajectory as it minimizes the distance between the original Bézier trajectory and the modified one. This definition is suitable for holonomic mobile robots since the original path has been generated by a global path planner, and the original path is assumed to be already optimal.

b. *Number of Bézier curves*

High order Bézier curves are numerically unstable and, for that reason, in order to generate a complete Bézier trajectory the concatenation of k curves is required. Therefore, the optimization function (16) is replaced by (17), where $1 \le l \le k$, α_l is every Bézier trajectory (l), $S_\varepsilon(\alpha_l)$ is the modified Bézier trajectory, $\left[t_0^{(l)}, t_f^{(l)}\right]$ are the initial and end instants of the Bézier trajectory (l), and $\varepsilon^{(l)}$ is the perturbation vector of the modified curve (l).

$$f(\varepsilon) = \min_{\varepsilon^{(1)} \ldots \varepsilon^{(k)}} \sum_{l=1}^{k} \int_{t_0^{(l)}}^{t_f^{(l)}} \left\| S_\varepsilon\left(\alpha_l(t)\right) - \alpha_l(t) \right\|_2^2 dt \tag{17}$$

The number of repulsive forces depends on the order of the Bézier trajectory: $n_{(l)} - 1$.

c. The constraints of the optimization problem are:

i. *The mobile robot must follow a collision-free path:* The modified Bézier path must pass through the endpoints, so the robot does not collide with the obstacles the environment. The vectors joining the initial and end points are the repulsive forces obtained by the PFP method. The equation of this constraint is (18).

$$g_1 = \sum_{l=1}^{k} \sum_{j=1}^{\eta} \left\langle \lambda, T_j^{(l)} - S_\varepsilon\left(\alpha_l\left(t_j^{(l)}\right)\right) \right\rangle \tag{18}$$

ii. *The robot trajectory must be smooth:* this constraint implies imposing continuity and derivability in the joint points of two curves, expressed by Eq. (19).

$$\begin{aligned} g_2 &= \sum_{l=1}^{k-1} \left\langle \lambda, S_\varepsilon\left(\alpha_l\left(t_f^{(l)}\right)\right) - S_\varepsilon\left(\alpha_l\left(t_0^{(l+1)}\right)\right) \right\rangle \\ g_3 &= \sum_{l=1}^{k-1} \left\langle \lambda, S_\varepsilon\left(\alpha_l'\left(t_f^{(l)}\right)\right) - S_\varepsilon\left(\alpha_l'\left(t_0^{(l+1)}\right)\right) \right\rangle \end{aligned} \tag{19}$$

iii. Continuity between the present and future positions must be ensured: tangency must be maintained between the original Bézier trajectory and the deformed Bézier trajectory at the initial and end points of the trajectory. The equation is (20).

$$g_4 = \left\langle \lambda, \alpha_1'\left(t_0^{(1)}\right) - S_\varepsilon\left(\alpha_1'\left(t_0^{(1)}\right)\right) \right\rangle + \left\langle \lambda, \alpha_k'\left(t_f^{(k)}\right) - S_\varepsilon\left(\alpha_k'\left(t_f^{(k)}\right)\right) \right\rangle \tag{20}$$

With the objective function and the constraints, the Lagrangian function (21) is defined. In order to calculate the stationary points, the partial derivatives of the Lagrangian function are calculated and canceled, and a system of linear equations is obtained. The solution of this linear system is the perturbation vector of each control point in order to obtain the Bézier trajectory. In-depth information about the linear system obtained is described in [48].

$$L(\varepsilon,\lambda_i)=f(\varepsilon)-g_1-g_2-g_3-g_4 \tag{21}$$

4.1. Numerical simulation

In our numerical example, it is used Bézier trajectories of second order to avoid loops in the trajectory. For that reason, the number of Bézier curves will be equal to the number of repulsive forces generated with the selected predictive PF technique. One vector is placed per Bézier curve. To develop, the BTD algorithm is necessary to follow these two steps:

1. Calculation of the control points from the prediction horizon generated with the PFP

The control points are uniformly distributed throughout the prediction horizon generated by the PFP method. The model has been developed for holonomic robots, and therefore, the prediction of future positions provides a straight line. In this case, the control points calculated through the formulation are obtained in **Table 2**.

2. Location of the repulsion forces on the Bézier curve

The control points of the Bézier curve are uniformly distributed, and the repulsion forces obtained with the PFP method are placed at the midpoint of each curve, except for the first and the last curves where they are placed at the first and last points, respectively.

Control Points of 1st curve	Control Points of 2nd curve	Control Points for the j-th curve
$P_0^{(1)}=\hat{x}(1)$	$P_0^{(2)}=P_2^{(1)}$	$3\le j\le k-2$
$P_1^{(1)}=\hat{x}(1)+\frac{1}{3}\cdot\left(\hat{x}(2)-\hat{x}(1)\right)$	$P_2^{(2)}=\hat{x}(2)+\frac{1}{2}\cdot\left(\hat{x}(3)-\hat{x}(2)\right)$	$P_0^{(j)}=\hat{x}(j-1)+\frac{1}{2}\cdot\left(\hat{x}(j)-\hat{x}(j-1)\right)$
$P_2^{(1)}=\hat{x}(1)+\frac{2}{3}\cdot\left(\hat{x}(2)-\hat{x}(1)\right)$	$P_1^{(2)}=P_0^{(2)}+\frac{1}{2}\cdot\left(P_2^{(2)}-P_0^{(2)}\right)$	$P_1^{(j)}=P_0^{(j)}$
		$P_2^{(1)}=\hat{x}(j)$

Control Points next to the last curve	Control Points of the last curve
$P_0^{(k-1)}=\hat{x}(k-2)+\frac{1}{2}\cdot\left(\hat{x}(k-1)-\hat{x}(k-2)\right)$	$P_0^{(k)}=P_2^{(k-1)}$
$P_2^{(k-1)}=\hat{x}(k-1)+\frac{1}{3}\cdot\left(\hat{x}(k)-\hat{x}(k-1)\right)$	$P_1^{(k)}=\hat{x}(k-1)+\frac{2}{3}\cdot\left(\hat{x}(k)-\hat{x}(k-1)\right)$
$P_1^{(k-1)}=P_0^{(k-1)}+\frac{1}{2}\cdot\left(P_2^{(k-1)}-P_0^{(k-1)}\right)$	$P_2^{(k)}=\hat{x}(k)$

Table 2. Calculation of control points from the prediction horizon: $\hat{x}$ is the vector containing the future trajectory and $P_i^{(j)}$ is the *i*-th control point of the j-th curve.

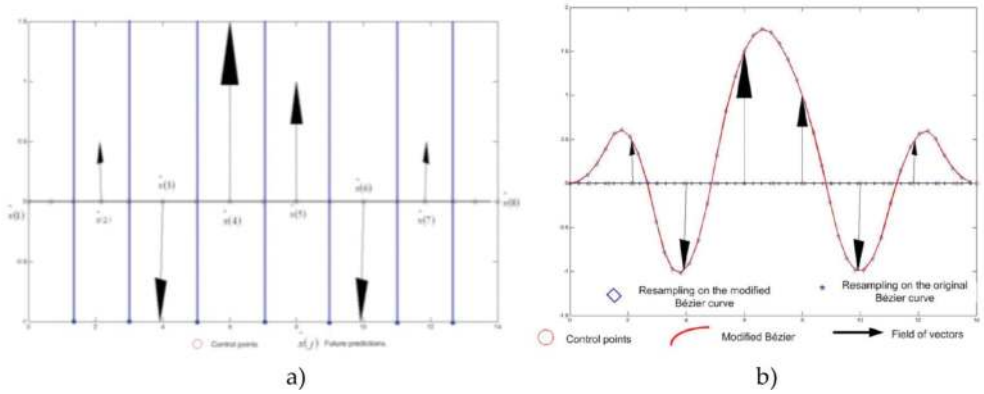

Figure 3. (a) Control points and future predictions of Bézier trajectory and (b) deformation of eight concatenated Bézier curves.

In **Figure 3(a)**, an example is shown, where a straight line represents the predicted optimal trajectory for a mobile robot obtained with the PFP algorithm. The control points needed to obtain the Bézier curves are displayed with red circles. The repulsive forces are placed in the proper positions of the predicted path. In this graphic example, there are eight points in the prediction horizon, and consequently, eight Bézier curves are concatenated in a straight line. The time devoted to perform trajectory is defined by the PFP prediction and has to be of 14 seconds. The time intervals corresponding to each curve, respectively, are [0,1.33], [1.33,3], [3,5], [5,7], [7,9], [9,11], [11,12.66], [12.66,14]. The representation of the resampling for the concatenation of eight Bézier curves is represented in **Figure 3(b)**.

5. Conclusion

This chapter details a comprehensive study of the use of parametric curves in the design of trajectories for holonomic and non-holonomic mobile robots. First, a brief introduction of the mathematical formulation and properties of the different curves is presented. Second, an exhaustive revision of literature regarding the use of parametric curves in path planning for mobile robots is developed. Third, a detailed description of the available techniques for path planning with parametric curves is presented, thoroughly describing the most important ones. Finally, an in-depth comparison is carried out between the different techniques of path deformation using Bézier curves, with their advantages and drawbacks. The Bézier curves are extensively used in these applications due to the simplicity of its definition and its easy handling and manipulation. The last section describes how to merge artificial potential field methods with Bézier curves as a solution for modifying a predefined trajectory in real time. Future works are related to the inclusion of other parametric curves, such as B-splines, RBC, and NURBS, in the proposed algorithm.

Author details

Lucía Hilario Pérez[1]*, Marta Covadonga Mora Aguilar[2], Nicolás Montés Sánchez[1] and Antonio Falcó Montesinos[1]

*Address all correspondence to: luciah@uchceu.es

1 Departamento Matemáticas, Físicas y Ciencias Tecnológicas, Universidad Cardenal Herrera-CEU, CEU Universities, Alfara del Patriarca, Spain

2 Department of Mechanical Engineering and Construction, Universitat Jaume I, Castellón, Spain

References

[1] https://www.google.com/selfdrivingcar/

[2] Van Schijndel-de Noóij M, et al. Definition of neccesary vehicle in infraestructure systems for automated driving. European Commission, Brusels, Belgium. SMART 2010/0064, 2011

[3] Gonzalez D, Perez J, Milanes V, Nashashibi F. A review of motion planning techniques for automated vehicles. IEEE Transactions on Intelligent Transportation Systems. 2016;**17**(4). DOI: 10.1109/TITS.2015.2498841

[4] Simba KR, Uchiyama N, Sano S. Real-time smooth trajectory generation for non-holonomic mobile robots using Bézier curves. Robotics and Computer-Integrated Manufacturing. 2016;**41**:31-42. doi.org/10.1016/j.rcim.2016.02.002

[5] Simba KR, Uchiyama N, Sano S. Real-time trajectory generation for mobile robots in a corridor-like space using Bézier curves, In: IEEE/SICE International Symposium on System Integration (SII). 2013. pp. 37-41. http://dx.doi.org/10.1109/SII.2013.6776639

[6] Cimurs R, Hwang J, II.H.Suh. Bézier curve-based smoothing for path planner with curvature constraints. IEEE International conference on Robotic computing. 2017. DOI: 10.1109/IRC.2017.13

[7] Elbanhawi M, Simic M, Jazar R. Randomized bidirectional B-spline parameterization motion planning. IEEE Transactions on Intelligent Transportation Systems. 2016;**17** (2):406-419. DOI: 10.1109/TITS.2015.2477355

[8] Gonzalez D, Perez J, Milanes V. Parametric-based path generation for automated vehicles at roundabouts. Expert Systems with Applications. 2017;**71**:332-341. DOI: 10.1016/j.eswa.2016.11.023

[9] Singh AK, Aggarwal A, Vashisht M, Siddavatam R. Robot motion planning in a dynamic environment using offset NURBS. IEEE ICIT. 2011:312-317. DOI: 10.1109/ICIT.2011.5754393

[10] Xidias EK, Aspragathos NA. Continuous curvature constrained shortest path for a car-like robot using S-Roadmaps. IEEE MED. 2013:13-18. DOI: 10.1109/MED.2013.6608692

[11] Jalel S, Marthon P, Hamouda A. NURBS based multi-objective path planning. In: Carrasco-Ochoa J, Martínez-Trinidad J, Sossa-Azuela J, Olvera López J, Famili F. eds. Pattern Recognition. MCPR 2015. Lecture notes in computer science, vol. 9116. Springer, Cham. 2015. doi.org/10.1007/978-3-319-19264-2_19

[12] Jalel S, Marthon P, Hamouda A. A new path generation based on accurate NURBS curves. International Journal of advanced Robotic systems. 2017;**13**(2), DOI:10.5772/63072

[13] Komoriya K, Tanie K. Trajectory design and control of a wheel-type mobile robot using B-spline curve. In Int. Workshop on Intelligence Robots&Systems. pp. 389-405. 1989. DOI: 10.1109/IROS.1989.637937

[14] Vázquez GB, Sossa AH, and Diaz de Leon S. Auto guided vehicle control using expanded time B-spline. In: IEEE International Conference on Systems, Man and Cybernetics. 1994;**3**:2786-2791. DOI: 10.1109/ICSMC.1994.400295

[15] Zhang J, Raczkowsky J, Herp A. Emulation of spline curve and its applications in robot motion control. In IEEE Int. Conference on Fuzzy Systems. 1994;**2**:831-836. DOI: 10.1109/FUZZY.1994.343843

[16] Eren H, Fung CC, Evans J. Implementation of the spline method for mobile robot path control. In IEEE Instrumentation and Measurement Technology Conference. 1999;**2**:739-744. DOI: 10.1109/IMTC.1999.776966

[17] Yamamoto M, Iwamura M, Mohri. Quasi time-optimal motion planning of mobile platforms in the presence of obstacles. In: International Conference on Robotics and Automation. ICRA. pp. 2958-2963. 1999. DOI: 10.1109/ROBOT.1999.774046

[18] Connors J, Elkaim G. Analysis of a Spline Based, Obstacle Avoding Path Planning Algorithm. IEEE Vehicle Technology Conference 2007. DOI: 10.1109/VETECS.2007.528

[19] Connors, Elkaim G. Experimental results of spline based obstacle avoidance of an off–road ground vehicle. ION Global Navigation Satellite Systems Conference. 2007. DOI: 10.1.1.154.2012

[20] Berglund T, Jonsson H, Sderkvist I. An obstacle-avoiding minimum variation B-spline problem. In: International Conference on Geometric Modeling and Graphics. 2003. DOI: 10.1109/GMAG.2003.1219681

[21] Shiller Z, Gwo YR. Dynamic motion planning of autonomous vehicles. IEEE Transactions on Robotics and Automation. 1991;**7**:241. DOI: 10.1109/70.75906

[22] Piegl L. On NURBS: A survey. In IEEE Computer Graphics and Applications. 1991;**11**(1): 55-71. DOI: 10.1109/38.67702

[23] Tatematsu N, Ohnishi K. Tracking motion of mobile robot for moving target using NURBS curve. In Int. Conference on Industrial Technology. 2003;**1**:245-249. DOI: 10.1109/ICIT.2003.1290283

[24] Aleotti J, Caselli S. Trajectory clustering and stochastic approximation for robot programming by demonstration. In: IEEE/RSJ Intelligent Robots and Systems. pp. 1029-1034. 2005. DOI: 10.1109/IROS.2005.1545365

[25] Aleotti J, Caselli S. Trajectory reconstruction with NURBS curves for robot programming by demonstration. In: IEEE International Symposium on Computational Intelligence in Robotics and Automation. pp 73-78. 2005. DOI: 10.1109/CIRA.2005.1554257

[26] Aleotti J, Caselli S. Grasp recognition in virtual reality for robot pregrasp planning by demonstration. In: IEEE International Conference on Robotics and Automation, ICRA. pp. 2801-2806. 2006. DOI: 10.1109/ROBOT.2006.1642125

[27] Montés N, Mora MC, Tornero J. Trajectory generation based on rational Bézier curves as clothoids. In: IEEE Intelligent Vehicles Symp., Istambul, Turkey. 2007;**505**:505-510. DOI: 10.1109/IVS.2007.4290165

[28] Montés N, Herraez A, Armesto L, Tornero J. Real-time clothoid approximation by rational bézier curves. In: International Conference on Robotics and Automation. Pasadena, CA, USA: ICRA. pp 2246-2251. 2008. DOI: 10.1109/ROBOT.2008.4543548

[29] Jaouni H, Khatib M, Laumond JP. Elastic bands for nonholonomic car-like robots: Algorithms and combinatorial issues. In: 3rd International Workshop on the Algorithmic Foundantions of Robotics (WAFR'98). 1998. ISBN:1-56881-081-4

[30] Khatib M, Jaouni H, Chatila R, Laumond JP. Dynamic path modification for car–like nonholonomic mobile robots. IEEE International Conference on Robotics and Automation, ICRA. 1997;**4**:2920-2925. DOI: 10.1109/ROBOT.1997.606730

[31] Graf B, Hostalet JM, Schaeffer C. Flexible path planning for nonholonomic mobile robots. In: 4th European workshop on advanced mobile robots (EUROBOT 01). pp. 199-206, Lund, Sweden, 2001. ISBN: 91-631-1464-X

[32] Nagatani K, Iwai Y, Tanaka Y. Sensor based navigation for car–like mobile robots using generalized voroni graph. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2. 2001, pp 1017-1022. DOI: 10.1109/IROS.2001.976302

[33] Hwang JH, Arkin RC, Know DS. Mobile robots atyour fingertrip: Bézier curve on–line trajectory generation for supervisory control. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, vol. 2003;**2**:1444-1449. DOI: 10.1109/IROS.2003.1248847

[34] Skrjanc I, Klancar G. Cooperative collision avoidance between multiple robots based on Bézier curves. In: Int. Conf. Information Technology Interfaces, pp. 451-455. 2007. DOI: 10.1109/ITI.2007.4283813

[35] Fujimori A, Teramoto M, Nikiforunk PN, Gupta MM. Cooperative collision avoidance between multiple robots. Journal of Robotic Systems. 2000;**17**(7):347-363. DOI: 10.1002/1097-4563(200007)17:7<347::AID-ROB1>3.0.CO;2-A

[36] Lamiraux F, Bonnafous D, Lefebvre O. Reactive path deformation for non- holonomic mobile robots. IEEE Transactions on Robotics and Automation. 2004;**20**:967-977. DOI: 10.1109/TRO.2004.829459

[37] Lizarraga M, Elklaim G. Spatially deconflicted path generation for multiple UAVs in a bounded airspace. In: IEEE/ION Position, Location & Navigation Symp. 2008. pp. 633-640. DOI: 10.1109/PLANS.2008.4570041

[38] Choi J. Real-time obstacle avoiding planning for autonomous ground vehicles. PhD thesis. California: University of California. December 2010. UMI Number:3442811

[39] Choi J, Curry R, Elkaim G. Path Planning Based on Bézier Curve for Autonomous Ground Vehicles. vol. 1. 158-166. IEEE Computer Society. 2008. DOI: 10.1109/WCECS.2008.27

[40] Choi J, Curry R, Elkaim G. Collision Free Real-Time Motion Planning for Omnidirectional Vehicles. In: European Control Conference. August 2009. ISBN: 978-3-9524173-9-3

[41] Choi J, Curry R, Elkaim G. Obstacle avoiding real-time trajectory generation and control of omnidirectional vehicles. In: American Control Conference (ACC). pp 5510-5515. St Louis, Missouri. 2009. DOI: 10.1109/ACC.2009.5160683

[42] Choi J, Curry R, Elkaim G. Smooth path generation based on Bézier curves for autonomous vehicles. In: World Congress on Engineering and Computer Sciences, WCECS. 2009. pp 668-673. DOI: 10.1.1.294.4485

[43] Choi J, Curry R, Elkaim G. Continuous curvature path generation based on Bézier curves for autonomous vehicles. International Journal of Applied Mathematics. 2010;**40**(2). DOI: 10.1.1.294.6438

[44] Choi J, Curry R, Elkaim G. Curvature-continuous trajectory generation with corridor constraint for autonomous ground vehicles. 49th IEEE Conference on Decision and Control, CDC. 2010. DOI: 10.1109/CDC.2010.5718154

[45] Choi J, Curry R, Elkaim G. Real-Time Obstacle-Avoiding Path Planning for Mobile Robots. In: AIAA Guidance, Navigation and Control, AIAA GNC, Toronto. 2010. DOI: 10.2514/6.2010-8411

[46] Choi J, Elkaim G. Bézier curves for trajectory guidance. In: World Congress on Engineering and Computer Sciences. pp. 625-630. San Francisco, CA, 2008. ISBN: 978-988-98671-0-2

[47] Neto A, Macharet DG, Campos MFM. Feasible RRT-based path planning using seventh order Bézier curves. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp 1445-1450. 2010. DOI: 10.1109/IROS.2010.5649145

[48] Hilario L, Montés N, Falcó A, Mora MC. Real-time trajectory deformation for potential fields planning methos. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS. pp 1567-1572. San Francisco, CA, 2011. DOI: 10.1109/IROS.2011.6094560

[49] Wu OB, Xia FH. Shape modification of Bézier curves by constrained optimization. Journal of Zhejiang University Science. pp. 124-127. 2005. ISSN: 1009-3095

[50] Mora MC, Tornero J. Path planning and trajectory generation using multi-rate predictive artificial potential fields. Proc. IEEE/RSJ IROS. 2008:2990-2995. DOI: 10.1109/IROS.2008.4651091

[51] Mora MC, Tornero J. Predictive and multirate sensor-based planning under uncertainty. IEEE Trans. on Intelligent Transportation Systems. 2015;**16**(3):1493-1504. DOI: 10.1109/TITS.2014.2366974