Trajectory Planning By Applying Optimal Velocity Profile Algorithm on Bernstein-Bézier Motion Primitives

Martina Loknar, Gregor Klančar, Sašo Blažič

Faculty of Electrical Engineering, Tržaška cesta 25, SI-1000 Ljubljana, Slovenia
E-mail: martina.loknar@fe.uni-lj.si

Abstract

This paper deals with minimal-time smooth trajectory planning for autonomous mobile systems. The resulting path is defined piecewise by multiple Bernstein-Bézier motion primitives that enable continuous velocity and curvature at the junctions. The solution is found by calculating travel time along Bernstein-Bézier segments with some free parameters by applying the optimal velocity profile algorithm that considers velocity, acceleration and jerk constraints. The proposed optimization approach was validated in a simulation environment.

1 Introduction

Path planning algorithms generate a geometric path from an initial to a final point through pre-defined via-points, either in the joint space or in the operating space of the robot. On the other hand, trajectory planning algorithms assign a time law to the geometric path. There is an increasing demand for robots and automatic machines to operate while some function is optimized and quite often the research is focused in time optimization. Path planning and trajectory planning algorithms are therefore attracting considerable interest and remain the crucial issues in the field of automation and robotics [1].

Defining the time of via-points progression influences the kinematic and the dynamic properties of the motion: the inertial forces and torques depend on the accelerations along the trajectory, while the vibrations of its mechanical structure are mostly determined by the values of the jerk. In order to satisfy kinematic feasibility of the vehicle the resulting path should be smooth; it should be realizable at high speed and at the same time be harmless for the robot in terms of avoiding excessive accelerations of the actuators and vibrations of its mechanical system [1]. Path smoothing is often not integrated in path planning but is usually done after the optimal path is found, which requires additional collision checks and can influence the path optimality. One promising approach would therefore be finding a smooth path to combine motion primitives in a path planning phase [2]. The smoothing techniques, not only limited to sample based planners, rely on using curves to interpolate or fit the given waypoints.

The first studies to obtain the shortest curvature constrained smooth paths consisting of straight lines and circular arcs were performed by [3], but the resulting tracks had a discontinuous curvature. This is why the authors in [4] used Dubins paths and smoothed them with clothoid arcs. The main advantage of clothoids is the linear change of their curvature. Unfortunately clothoids are defined in terms of Fresnel integrals; transcendental functions that cannot be solved analytically. This fact makes clothoids difficult to use in real-time applications so that for real-time motion planning many authors resort to curves with nonlinear curvature that are easier to compute, for example Bézier curves [2]. Other smoothing methods also include application of cubic, quintic or higher order polynomials [5] and B-splines [6].

This paper presents the results of a study where the optimal velocity algorithm was validated on fifth-order Bernstein-Bézier curve segments with continuous velocity and curvature transitions. The optimal velocity algorithm considers restrictions of speed, radial and tangential acceleration and radial and tangential jerk. The purpose of the analysis was to determine the travel time along Bernstein-Bézier motion primitives in order to find a time optimal corridor-restricted path with the minimum travel time and to provide a better insight into possible trajectory planning strategies.

2 Problem statement

Consider the problem of a ground vehicle with a mission defined by corridor and dynamical constraints in a two-dimensional free space. Our goal is to develop and implement an algorithm for generating a trajectory that satisfies these restrictions.

Let the motion of a particle along a three times continuously differentiable plane curve $\mathcal{C}$ be described as a function of time $t \in [0, t_f]$ by the position vector $\mathbf{r}(t)$ measured from a given fixed origin. Velocity $\mathbf{v}(t)$, acceleration $\mathbf{a}(t)$ and jerk $\mathbf{j}(t)$ vectors can be expressed in the tangential-normal form as:

$$\mathbf{v}(t) = v(t) \cdot \hat{\mathbf{T}} \tag{1a}$$

$$\mathbf{a}(t) = a_{\mathrm{T}}(t) \cdot \hat{\mathbf{T}} + a_{\mathrm{R}}(t) \cdot \hat{\mathbf{N}} \tag{1b}$$

$$\mathbf{j}(t) = j_{\mathrm{T}}(t) \cdot \hat{\mathbf{T}} + j_{\mathrm{R}}(t) \cdot \hat{\mathbf{N}}, \tag{1c}$$

where $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ are the unit tangent vector and the unit normal vector, respectively:

$$\hat{\mathbf{T}}(t) = \frac{\mathbf{v}(t)}{\|\mathbf{v}(t)\|}, \quad \hat{\mathbf{N}}(t) = \frac{\dot{\hat{\mathbf{T}}}(t)}{\|\dot{\hat{\mathbf{T}}}(t)\|}. \tag{2}$$

Given a feasible segment of the path, which in our case is a Bernstein-Bézier curve, the optimization problem is to find the velocity profile $v(t)$ that reaches the end of the curve in minimum time in a way that none of the velocity, acceleration or jerk constraints from Eqs. $(3a, 3b, 3c)$ are violated:

$$0 \leq \|\mathbf{v}(t)\| \leq v_{\text{MAX}}; \quad \forall t \in [0, t_f], \tag{3a}$$
$$\frac{a_T^2(t)}{a_{T_{\text{MAX}}}^2} + \frac{a_R^2(t)}{a_{R_{\text{MAX}}}^2} \leq 1; \quad \forall t \in [0, t_f], \tag{3b}$$
$$\frac{j_T^2(t)}{j_{T_{\text{MAX}}}^2} + \frac{j_R^2(t)}{j_{R_{\text{MAX}}}^2} \leq 1; \quad \forall t \in [0, t_f]. \tag{3c}$$

The acceleration and jerk constraints are defined in a similar way as in [7].

3 Bernstein-Bézier motion primitives

An N-dimensional, n-th order Bernstein polynomial $\mathbf{r}_n(\lambda) : [0,1] \to \mathbb{R}^N$ can be defined as:

$$\mathbf{r}_n(\lambda) = \sum_{i=0}^{n} \mathbf{P}_{i,n} B_{i,n}(\lambda), \qquad \lambda \in [0,1], \tag{4}$$

where λ is normalized time ($0 \leq \lambda \leq 1$), $\mathbf{P}_{i,n} \in \mathbb{R}^N$ is the i-th control point and $B_{i,n}(\lambda)$ is the Bernstein polynomial basis defined as:

$$B_{i,n}(\lambda) = \binom{n}{i} \lambda^i (1-\lambda)^{n-i}, \tag{5}$$

for all $i \in \{0, \ldots, n\}$. Binomial coefficient is defined as:

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}. \tag{6}$$

Letting $\mathbf{P}_n = [\mathbf{P}_{0,n}, \ldots, \mathbf{P}_{n,n}] \in \mathbb{R}^{N \times (n+1)}$ be the vector of control points of $\mathbf{r}_n(\lambda)$, the Bernstein polynomial in Eq. (4) can be rewritten in matrix form as:

$$\mathbf{r}_n(\lambda) = \mathbf{P}_n \begin{bmatrix} B_{0,n}(\lambda) \\ B_{1,n}(\lambda) \\ \cdots \\ B_{n,n}(\lambda) \end{bmatrix}. \tag{7}$$

In the case when two or three-dimensional Bernstein polynomials are used to describe planar and spatial curves, Bernstein polynomials are often referred to as Bézier curves. These curves have useful path planning properties. The first and the last points of the Bernstein polynomial introduced in Eq. (4) are its endpoints:

$$\mathbf{r}_n(0) = \mathbf{P}_{0,n} \quad \text{and} \quad \mathbf{r}_n(1) = \mathbf{P}_{n,n}. \tag{8}$$

The N-dimensional, n-th order Bernstein polynomial also lies within the convex hull defined by its control points. Furthermore, the start and the end of the curve is tangent to the first and the last section of the convex polygon, respectively (Fig. 1).

$$\left.\frac{\mathrm{d}\mathbf{r}_n}{\mathrm{d}\lambda}\right|_{\lambda=0} = n(\mathbf{P}_{1,n} - \mathbf{P}_{0,n}), \tag{9}$$
$$\left.\frac{\mathrm{d}\mathbf{r}_n}{\mathrm{d}\lambda}\right|_{\lambda=1} = n(\mathbf{P}_{n,n} - \mathbf{P}_{n-1,n}). \tag{10}$$

Other properties of Bernstein polynomials (derivatives, calculating definite integrals, the de Casteljau's algorithm, degree elevation etc.) do not fall within the scope of this article; more details on this topic can be found in [8].

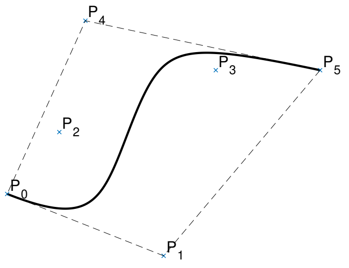

Figure 1: Fifth order Bernstein-Bézier curve with its convex hull (dotted lines). The curve is tangent to the sides of the convex hull, line segments $\overline{P_0P_1}$ and $\overline{P_4P_5}$.

3.1 Merging of motion primitives

For the sake of generality, we consider a merging of two Bézier curves of different orders, $\mathbf{r}_{n_j}^{j}(\lambda)$ of order n_j and $\mathbf{r}_{n_{j+1}}^{j+1}(\lambda)$ of order n_{j+1}, into a spline. The three conditions for continuous first and second derivatives (C^2 continuity) in the junction are:

$$\lim_{\lambda\to1} \mathbf{r}_{n_j}^{j}(\lambda) = \lim_{\lambda\to0} \mathbf{r}_{n_{j+1}}^{j+1}(\lambda), \tag{11}$$
$$\lim_{\lambda\to1} \frac{d\mathbf{r}_{n_j}^{j}(\lambda)}{d\lambda} = \lim_{\lambda\to0} \frac{d\mathbf{r}_{n_{j+1}}^{j+1}(\lambda)}{d\lambda}, \tag{12}$$
$$\lim_{\lambda\to1} \frac{d^2\mathbf{r}_{n_j}^{j}(\lambda)}{d\lambda^2} = \lim_{\lambda\to0} \frac{d^2\mathbf{r}_{n_{j+1}}^{j+1}(\lambda)}{d\lambda^2}, \tag{13}$$

This yields the following relations between the control points of both Bézier curves:

$$\mathbf{P}_{0,n_{j+1}}^{j+1} = \mathbf{P}_{n_j,n_j}^{j}, \tag{14}$$
$$\mathbf{P}_{1,n_{j+1}}^{j+1} = (1 + \frac{n_j}{n_{j+1}})\mathbf{P}_{n_j,n_j}^{j} - \frac{n_j}{n_{j+1}}\mathbf{P}_{n_j-1,n_j}^{j}, \tag{15}$$
$$\begin{aligned}\mathbf{P}_{2,n_{j+1}}^{j+1} &= \left(1 + \frac{n_j}{n_{j+1}}\left(2 + \frac{1+n_j}{1+n_{j+1}}\right)\right)\mathbf{P}_{n_j,n_j}^{j} \\ &\quad - 2\left(1 + \frac{n_j(1+n_j)}{n_{j+1}(1+n_{j+1})}\right)\mathbf{P}_{n_j-1,n_j}^{j} \\ &\quad + \frac{n_j(1+n_j)}{n_{j+1}(1+n_{j+1})}\mathbf{P}_{n_j-2,n_j}^{j}.\end{aligned} \tag{16}$$

3.2 Path generation

Bézier curves constructed by large numbers of control points are numerically unstable. For this reason, in path planning, it is desirable to construct a smooth way by joining together low degree Bézier curves.

We employed the curves of the fifth order, as this is the least degree of Bézier curves that can satisfy the requirement for curvature (C^2) continuity. The fifth-order Bernstein -Bézier curve $\mathbf{r}_5(\lambda) = [x(\lambda), y(\lambda)]^{\mathrm{T}}$ is defined by six control points $\mathbf{P}_{i,5} = [x_i, y_i]$, $i \in \{0, 1, \ldots 5\}$.

$$\begin{aligned}\mathbf{r}_5(\lambda) &= (1-\lambda)^5\mathbf{P}_{0,5} + 5\lambda(1-\lambda)^4\mathbf{P}_{1,5} \\ &\quad + 10\lambda^2(1-\lambda)^3\mathbf{P}_{2,5} + 10\lambda^3(1-\lambda)^2\mathbf{P}_{3,5} \\ &\quad + 5\lambda^4(1-\lambda)\mathbf{P}_{4,5} + \lambda^5\mathbf{P}_{5,5}.\end{aligned} \tag{17}$$

The complete path through the corridor consists of several Bernstein-Bézier curve sections. According to Eqs. (14-16) for $n_j = 5$ and $n_j = n_{j+1}$ the first three control points of the $(j+1)$-th Bernstein-Bézier curve $\mathbf{r}_5^{j+1}$ are:

$$\mathbf{P}_{0,5}^{j+1} = \mathbf{P}_{5,5}^{j}, \tag{18}$$

$$\mathbf{P}_{1,5}^{j+1} = 2\mathbf{P}_{5,5}^{j} - \mathbf{P}_{4,5}^{j}, \tag{19}$$

$$\mathbf{P}_{2,5}^{j+1} = 4\mathbf{P}_{5,5}^{j} - 4\mathbf{P}_{4,5}^{j} + \mathbf{P}_{3,5}^{j}. \tag{20}$$

where $P_{i,n}^{j}$ is the i-th control point of j-th curve $\mathbf{r}_5^j$. The last control point $\mathbf{P}_{5,5}^{j+1}$ is defined by the final position:

$$\mathbf{P}_{5,5}^{j+1} = \begin{bmatrix} x_5^{j+1} \\ y_5^{j+1} \end{bmatrix}. \tag{21}$$

The expressions for the control points $\mathbf{P}_{4,5}^{j+1}$ and $\mathbf{P}_{3,5}^{j+1}$ follow from evaluating the first and the second derivative of $\mathbf{r}_5^{j+1}$ in $\lambda = 1$ and setting its values to v_{spline} and a_{spline}, respectively:

$$\left.\frac{\mathrm{d}\mathbf{r}_5^{j+1}}{\mathrm{d}\lambda}\right|_{\lambda=1} = v_{\text{spline}}, \tag{22}$$

$$\left.\frac{\mathrm{d}^2\mathbf{r}_5^{j+1}}{\mathrm{d}\lambda^2}\right|_{\lambda=1} = a_{\text{spline}}. \tag{23}$$

$\mathbf{P}_{4,5}^{j+1}$ and $\mathbf{P}_{3,5}^{j+1}$ are defined by both the final position and the final orientation φ_5^{j+1}:

$$\mathbf{P}_{4,5}^{j+1} = \mathbf{P}_{5,5}^{j} - \tfrac{1}{5} v_{\text{spline}} \cdot \begin{bmatrix} \cos(\varphi_5^{j+1}) \\ \sin(\varphi_5^{j+1}) \end{bmatrix}, \tag{24}$$

$$\mathbf{P}_{3,5}^{j+1} = -\mathbf{P}_{5,5}^{j} + 2\mathbf{P}_{4,5}^{j} + \tfrac{1}{20} a_{\text{spline}} \cdot \begin{bmatrix} \cos(\varphi_5^{j+1}) \\ \sin(\varphi_5^{j+1}) \end{bmatrix}, \tag{25}$$

It should be pointed out that the values of the first and the second curve derivative in the $(j+1)$-th Bernstein-Bézier curve end point $\mathbf{P}_{5,5}^{j+1}$, v_{spline} and a_{spline}, are not true speed and tangential acceleration. The notation that we used simply reflects the analogy.

The flowchart in Fig.(2) describes the general principle of operation of the proposed trajectory planning algorithm. Corridor consists of m segments and j is the number of the current segment on which the calculation is performed: $j \in \{1, 2, \dots m\}$. The given coordinates and values of velocity, angle and tangential acceleration in $\mathbf{P}_{0,5}^{j}$ according to Eqs.(19-20) also determine the second and the third control point of the Bézier curve, $\mathbf{P}_{1,5}^{j}$ and $\mathbf{P}_{2,5}^{j}$. The problem is to calculate the three remaining control points of the Bézier curve along which the travel time is the shortest. To find the solution, MATLAB's integrated function `fmincon` was used. This is a solver-based nonlinear optimization approach that finds a minimum of a constrained nonlinear multivariable function. The starting point of the optimization is travel time along a Bézier curve, given by setting the coordinates, velocity, angle and tangential acceleration in the final control point $\mathbf{P}_{5,5}^{j}$. The control points $\mathbf{P}_{3,5}^{j}$ and $\mathbf{P}_{4,5}^{j}$ are determined according to the Eqs.(24-25). The travel time t along this Bézier curve is calculated by applying the algorithm that generates the optimal velocity profile. The function `fmincon` then iteratively modifies the free parameters ($(x, y), v, \varphi$ and a_T) in $\mathbf{P}_{5,5}^{j}$ in a way that the travel time along the curve decreases until the local minimum is found (t_{MIN} in Fig.(2)). This is a basic concept of the gradient method: from the starting point the optimization function at each iteration takes either a direct (Newton) step or a conjugate gradient step to reach the local minimum of the cost function. The calculation is repeated for each segment to finally determine the complete trajectory $\mathbf{G}_1$ along the corridor.

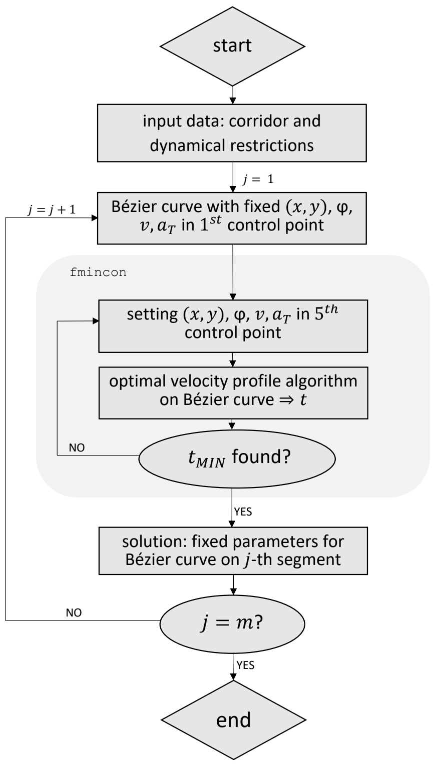
start
input data: corridor and dynamical restrictions
j = 1
j = j + 1
Bézier curve with fixed (x, y), φ, v, a_T in 1st control point
fmincon
setting (x, y), φ, v, a_T in 5th control point
optimal velocity profile algorithm on Bézier curve ⇒ t
NO
t_MIN found?
YES
solution: fixed parameters for Bézier curve on j-th segment
NO
j = m?
YES
end

Figure 2: The basic principle of operation of the proposed trajectory planning algorithm.

In order to study the effectiveness of the proposed method, the time optimal trajectory planning algorithm is executed in two other versions. In the second version, minimal travel time $t_{\text{MIN},2seg}$ is calculated along Bézier curves for two successive corridor segments (for j and $j+1$) and in the third version $t_{\text{MIN},3seg}$ it is calculated for three successive corridor segments (for j, $j+1$ and $j+2$). Then, only the control points $\mathbf{P}_{i,5}^{j}$ ($i \in \{0, \dots, n\}$) of the j-th segment are retained. This procedure is performed on the next segments until the the complete trajectory along the corridor is found: $\mathbf{G}_2$ in the second and $\mathbf{G}_3$ in the third version of the algorithm.

4 Simulation results

To demonstrate our trajectory planning method, we designed the problem as follows: compute the time optimal trajectory through the given 6-segment corridor with the following dynamical restrictions: $v_{\text{MAX}} = 1.5\,\text{m/s}$, $a_{\text{T}_{\text{MAX}}} = 2\,\text{m/s}^2$, $a_{\text{R}_{\text{MAX}}} = 4\,\text{m/s}^2$, $j_{\text{T}_{\text{MAX}}} = 6\,\text{m/s}^3$

and $j_{\mathrm{R_{MAX}}} = 8\,\mathrm{m/s^3}$. On the starting line, the values of velocity and tangential acceleration are $v = 1\,\mathrm{m/s}$ and $a_{\mathrm{T}} = 0.5\,\mathrm{m/s^2}$. The initial position is [0, 0] and the initial orientation is set parallel to the corridor center line.

Figure (3) shows the given six segment corridor and the optimal trajectories $\mathbf{G}_i$, depicted with a thick black line. The sets of discarded Bernstein-Bézier curves are depicted with thin grey lines. $\mathbf{G}_1$,$\mathbf{G}_2$, $\mathbf{G}_3$ are calculated based on knowledge of the shape of the current segment, two succesive segments and three successive segments, respectively.

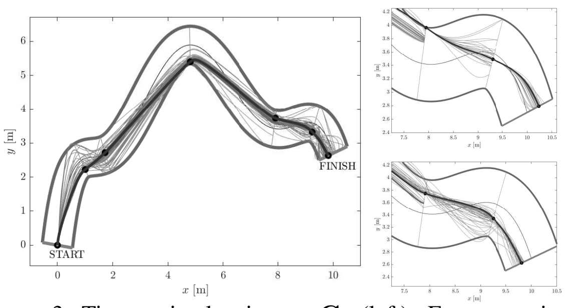

Figure 3: Time optimal trajectory $\mathbf{G}_3$ (left). For comparison: iteration steps in the last three segments for $\mathbf{G}_1$ (top right) and $\mathbf{G}_2$ (bottom right).

Table (1) lists the travel times t_i at the end of the separate corridor segments. The evidence from this study proves that the time optimal trajectory can be determined by our trajectory planning method. Furthermore, the more successive segments are taken into account when performing the calculation, the lower is the overall minimal travel time. Table (2) presents some computational properties of the proposed algorithm: number of iterations per segment $N_{\mathrm{it,seg}}$, number of points (per segment) `F-count` where the function evaluations took place, and the total computation time t_{COMP}. It should be pointed out that the problem has many constraints and the `F-count` can be significantly less than the total number of function evaluations.

$[s]$	t_1	t_2	t_3	t_4	t_5	t_6	$\sum_{i=1}^{6} t_i$
$\mathbf{G}_1$	1.79	0.80	2.93	2.48	0.96	0.85	9,81
$\mathbf{G}_2$	1,75	0,62	2,74	2,41	1,03	0,65	9,20
$\mathbf{G}_3$	1,73	0,62	2,75	2,43	1,01	0,64	9,18

Table 1: t_i and overall travel time $\sum_{i=1}^{6} t_i$ for $\mathbf{G}_i$.

	$N_{\mathrm{it,seg}}$	`F-count`	$t_{\mathrm{COMP}}[s]$
$\mathbf{G}_1$	25	250	246
$\mathbf{G}_2$	13	190	306
$\mathbf{G}_3$	19	305	738

Table 2: Computational properties of the proposed algorithm.

The current solutions to the time optimal trajectory generation problem under kinematic constraints (that include jerk limitations) exist mainly for robot manipulators, whereas the literature on the same topic in the case of autonomous vehicles is still scarce. A key problem with much of the literature is also that the approaches often do not support Cartesian constraints.

5 Conclusion

This paper presents a trajectory planning algorithm in a free space that is based on Bernstein-Bézier curves and considers corridor and dynamical (velocity, acceleration and jerk) constraints. Bézier curves provide an efficient way to generate the optimized trajectory and satisfy the constraints at the same time. The simulation results show that the minimal travel time through the corridor is the lowest in the case when three successive corridor segments are considered in the proposed method.

Despite some limitations, we believe that the results of this study could be the basis for our future research on similar receding horizon control methods to generate real-time optimal trajectories with dynamical restrictions. In the future, we will also consider a development of a more sophisticated way to determine whether the calculated path is within the corridor boundaries. Using higher order Bézier curves with more free parameters might also prove to be beneficial.

Acknowledgements

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0219).

References

[1] A. Gasparetta, P. Boscario, A. Lanzutti, and R. Vidoni, *Motion and Operation Planning of Robot Systems, Chapter 1 - Path Planning and Trajectory Planning Algorithms: A General Overview*, vol. 29. 2015.

[2] G. Klančar, M. Seder, S. Blažič, I. Škrjanc, and I. Petrović, “Drivable Path Planning Using Hybrid Search Algorithm Based on E* and Bernstein-Bézier Motion Primitives,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. 1957, pp. 1–15, 2019.

[3] L. E. Dubins, “On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents,” *American Journal of Mathematics*, vol. 79, no. 3, p. 497, 1957.

[4] M. Shanmugavel, A. Tsourdos, B. White, and R. Zbikowski, “Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs,” *Control Engineering Practice*, vol. 18, no. 9, pp. 1084–1092, 2010.

[5] F. Ghilardelli, G. Lini, and A. Piazzi, “Path Generation Using ηˆ4-Splines for a Truck and Trailer Vehicle,” *IEEE Transactions on Automation Science and Engineering*, vol. 11, pp. 187–203, jan 2014.

[6] T. Berglund, U. Erikson, H. Jonsson, K. Mrozek, and I. Söderkvist, “Automatic generation of smooth paths bounded by polygonal chains,” no. August 2014, 2001.

[7] M. Lepetič, G. Klančar, I. Škrjanc, D. Matko, and B. Potočnik, “Time optimal path planning considering acceleration limits,” *Robotics and Autonomous Systems*, vol. 45, no. 3-4, pp. 199–210, 2003.

[8] C. Kielas-Jensen and V. Cichella, “BeBOT: Bernstein Polynomial Toolkit for Trajectory Generation,” *IEEE International Conference on Intelligent Robots and Systems*, no. 3, pp. 3288–3293, 2019.