Towards Trajectory Planning of a Robot Manipulator with Computer Algebra using Bézier Curves for Obstacle Avoidance*

Ryo Hatakeyama[1], Akira Terui[2,*] and Masahiko Mikawa[3]

[1]Master's Program in Mathematics, Graduate School of Science and Technology, University of Tsukuba, Tsukuba 305-8571, Japan
[2]Institute of Pure and Applied Sciences, University of Tsukuba, Tsukuba 305-8571, Japan
[3]Institute of Library, Information and Media Science, University of Tsukuba, Tsukuba 305-8550, Japan

Abstract
This paper discusses the trajectory planning of a robot manipulator with computer algebra. In the operation of robot manipulators, it is important to make the trajectory of the end effector so that it does not collide with obstacles. For this purpose, we have proposed a method of generating a method of trajectory planning using cubic spline curves. However, the method has the disadvantage that the trajectory may not be included in the feasible region of the manipulator, thus an extra test for the inclusion of the curve is needed. In this paper, we propose a new method of generating a trajectory using Bézier curves, which is guaranteed to be included in the feasible region.

Keywords
Trajectory planning, Quantifier elimination, Inverse kinematics, Bézier curves

1. Introduction

In this paper, we discuss the trajectory planning of a robot manipulator with computer algebra. A manipulator is a robot with links, which correspond to human arms, and joints, which correspond to human joints, connected alternately. The end effector is the component located at the tip of the link that is farthest from the base of the manipulator. The inverse kinematics problem examines whether it is possible to place the end effector at a given point and orientation, and if it is possible, the problem is to find the joint configuration for that placement. The trajectory planning problem is to find a path for the end effector to move from the start point to the endpoint without colliding with obstacles.

There are numerous methods for solving the inverse kinematics problem of manipulators using computer algebra proposed to date [1, 2, 3, 4, 5, 6]. Among them, we have proposed one that enhances computation efficiency with the use of Comprehensive Gröbner Systems (CGS) [7], and certifies the existence of solutions to inverse kinematics problems using the Quantifier Elimination (QE) which is based on the CGS, so-called the CGS-QE [8] method [9]. Furthermore, we have extended the method to trajectory planning using a straight-line path [10].

By using the straight-line path, the trajectory may interfere with obstacles. Therefore, it is possible to design the trajectory to avoid obstacles using a polyline, but doing so may result in discontinuities in velocity and acceleration of the manipulator at the vertices of the polyline, which could destabilize the operation. To achieve smooth movement of the end effector, we have

SCSS 2024: 10th International Symposium on Symbolic Computation in Software Science, August 28–30, 2024, Tokyo, Japan
*This work was partially supported by JKA and its promotion funds from KEIRIN RACE.
*Corresponding author.
s2320143@u.tsukuba.ac.jp (R. Hatakeyama); terui@math.tsukuba.ac.jp (A. Terui); mikawa@slis.tsukuba.ac.jp (M. Mikawa)
https://researchmap.jp/aterui (A. Terui); https://mikawalab.org/ (M. Mikawa)
0000-0003-0846-3643 (A. Terui); 0000-0002-2193-3198 (M. Mikawa)

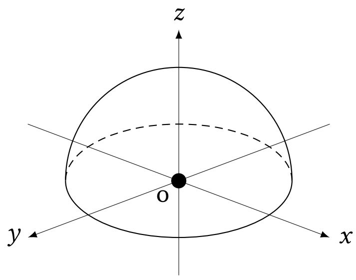

Figure 1: The feasible region of the manipulator.

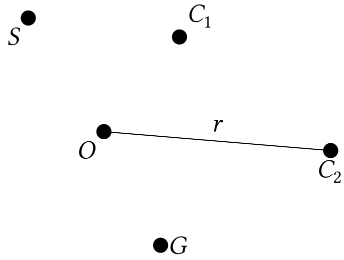

Figure 2: Defining the radius of the feasible region.

proposed trajectory planning using a spline curve [11]. However, the spline curve may not be included in the feasible region of the manipulator, thus an extra test for the inclusion of the curve is needed.

In this paper, we propose methods for trajectory planning using Bézier curves. The shape of a Bézier curve is determined by the placement of control points, and the curve has the property of always being contained within the convex hull formed by the control points as vertices. Utilizing this property, the goal is to generate trajectories that avoid obstacles without deviating from the feasible region. Although several proposals have been made for the use of Bézier curves in general trajectory planning [12, 13, 14], to the best of the authors' knowledge, the utilization of Bézier curves in trajectory planning using computer algebra is considered to be almost nonexistent.

This paper is organized as follows. In Section 2, we introduce the feasible region of the manipulator and define the radius of the region. In Section 3, we introduce the Bézier curve and its properties. In Section 4, we propose two methods for trajectory planning using Bézier curves. In Section 5, we conclude and discuss future research direction.

2. Preliminaries

In this paper, the manipulator to be used is placed on the real space $\mathbb{R}^3$ with the global coordinate system, whose origin is located at the base of the manipulator. Assume that the end effector can be placed anywhere in the feasible region except the origin (see fig. 1).

The feasible region refers to the range that the end-effector of the manipulator can reach. Here, we assume that the feasible region of the manipulator is given as the surface and the interior of a hemisphere with the origin as the center and positive z coordinates.

As for the radius of the region, we define a value different from the radius that the actual manipulator can execute. In this case, we will define the radius by comparing the distances between the origin and the start and the endpoints of the end effector, and the farthest of the passing points, as described below.

An example is shown in fig. 2. Let us consider a trajectory with the start point S and the endpoint G that passes through two predetermined points C_1 and C_2. Assume that we have verified the end-effector can reach all the points except for the origin O by, for example, the method we have previously proposed [10]. S, C_1, C_2, and G are located as in fig. 2 with respect to the origin O. In this case, the point farthest from O among these is C_2. Therefore, the distance r between O and C_2 is defined as the r, radius of the feasible region. We see that, by the radius r in this way, the end-effector will of course be able to reach a point closer than the point where the end-effector is already determined to be placed.

3. The Bézier Curve

The Bézier curve [15] is a parametric curve on which the point is expressed as a polynomial function of the parameter t, defined as follows.

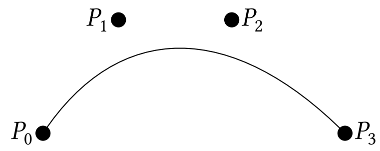

Figure 3: Construction of a 3-degree Bézier Curve.

Definition 1. Let $P_0, P_1, \ldots, P_n$ be different points. Then, the N-degree Bézier Curve $P(t)$ is defined as

$$P(t) = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} P_i, \quad 0 \le t \le 1,$$

where $P_0, P_1, \ldots, P_n$ are the control points. Note that the curve starts at P_0 and ends at P_n.

The method employs control points for producing the curve. Note that the curve does not pass through the control points except for the start and the endpoint but is attracted by them. It is as if the points exert a pull on the curve.

An n-degree Bézier Curve is constructed as follows (for an example of a 3-degree Bézier curve, see Figure 3). First, place $n+1$ control points $P_0, P_1, \ldots, P_n$, and take points $Q_0^{(1)}, Q_1^{(1)}, \ldots, Q_{n-1}^{(1)}$ that divide the line segment $P_0P_1, P_1P_2, \ldots, P_{n-1}P_n$ internally into $t : 1-t$, respectively. Next, take points $Q_0^{(2)}, Q_1^{(2)}, \ldots, Q_{n-2}^{(2)}$ that divide the line segments $Q_0^{(1)}Q_1^{(1)}, Q_1^{(1)}Q_2^{(1)}, \ldots, Q_{n-2}^{(1)}Q_{n-1}^{(1)}$, internally into $t : 1-t$, respectively. Repeat the same operation for n times to obtain $Q_0^{(n)}$, then the locus of is $Q_0^{(n)}$ for $0 \le t \le 1$ constructs the Bézier Curve.

One of the advantages of using Bézier Curves is the convex hull property of the curves [13]. For the convex hull $A = \{\sum_{i=0}^{n} c_i P_i \mid \sum_{i=0}^{n} c_i = 1, 0 \leqq c_i \leqq 1\}$ of the control points $P_0, P_1, \ldots, P_n$, we see that any point on the Bézier Curve $P(t)$ is included in A. In other words, if a polyhedron with each control point as a vertex is included in the feasible region, the Bézier Curve obtained from those control points is also included in the region. This idea will be used in our second method proposed below.

4. Path planning using Bézier curves

We propose two methods for trajectory planning using Bézier curves. In the methods, we settle the following assumptions: In addition to the start and the endpoints, two points that the curve must pass through are given. Those points have the same y and z coordinates, respectively.

4.1. Method 1: path planning with single Bézier curve

In the first method, we construct a path with a single Bézier curve. We give the x coordinates of the control points equally distributed and select the control points using equality and inequality evaluation. Furthermore, we consider only the case where the curve is curved in the positive direction in both y and z coordinates.

If an n-degree Bézier Curve in $\mathbb{R}^3$ is represented as $P(t) = (P_x(t), P_y(t), P_z(t))$, each component is expressed as

$$P_x(t) = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} x_i, \quad P_y(t) = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} y_i, \quad P_z(t) = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} z_i, \tag{1}$$

using the control points $P_i(x_i, y_i, z_i)$ $(i = 0, 1, \ldots n)$. Here, we define the x coordinates of each control point satisfying that $P_x(t)$ is a linear polynomial in t. This approach has the advantage that it not only reduces the amount of calculation required to create the trajectory but also makes the operation of "finding the value of t from the x coordinate of a point on the curve" easier.

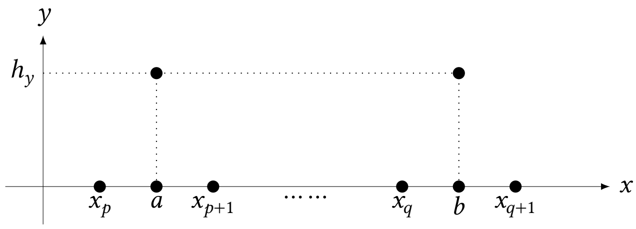

Figure 4: Position of (a, h_y) and (b, h_y).

To achieve this objective, simply place the x coordinates $x_0, x_1, \ldots, x_n$ of each control point in order at equal intervals, i.e., for $i = 0, 1, \ldots, n$, let

$$x_i = \frac{(n-i)x_0 + ix_n}{n}.$$

Substituting x_i into $P_x(t)$ in eq. (1) gives

$$P_x(t) = (1-t)x_0 + tx_n. \tag{2}$$

Next, we define the y and z coordinates of each control point. In eq. (2), denote the x coordinate of a point on the curve simply as $P_x(t) = x$, and solve eq. (2) for t, then we have

$$t = \frac{x - x_0}{x_n - x_0}. \tag{3}$$

Now, assume that there is an obstacle in the feasible region, and to avoid it, the y and z coordinates of the path must always exceed a certain value $h_y > 0$ and $h_z > 0$, respectively, in the interval $[a, b]$ in the x coordinate, where $x_p < a \le x_{p+1} \le x_q \le b < x_{q+1}$ for $1 \le p < q + 1 \le n$ (see Figure 4). Substituting $x = a$ and b for eq. (3) and denoting them as t_a and t_b, respectively, we obtain

$$t_a = \frac{a - x_0}{x_n - x_0}, \quad t_b = \frac{b - x_0}{x_n - x_0}.$$

For the y coordinate of the curve, among the tuples of $y_1, y_2, \ldots y_{n-1}$ that satisfy the following equalities and inequalities:

$$P_y(t_a) = P_y(t_b) = h_y, \quad y_0 \le y_1, \ldots, y_p, \quad h_y \le y_{p+1}, \ldots, y_q, \quad y_n \le y_{q+1}, \ldots, y_{n-1}, \tag{4}$$

select the one with the smallest sum $\sum_{k=p+1}^{q} y_k$. The reason for selecting the tuple with the smallest sum is to minimize the bulge in the positive direction of the y coordinate of the curve. Furthermore, to prevent each y_i value from becoming too small, a lower limit is set as constraints in eq. (4) so that it does not fall below the coordinate of the starting point before crossing the obstacle, and the coordinate of the endpoint after crossing the obstacle.

For the z coordinate of the curve, $z_1, z_2, \ldots, z_{n-1}$ can be calculated in the same way as $y_1, y_2, \ldots y_{n-1}$ are calculated in above, simply by replacing y_i with z_i. After obtaining $y_1, y_2, \ldots y_{n-1}$ and $z_1, z_2, \ldots, z_{n-1}$, put these values into $P_y(t)$ and $P_z(t)$ in eq. (1), respectively, which gives the coordinates of all control points, thus the curve becomes the trajectory of the end effector.

The advantage of this method is that the x coordinate of a point on the curve is expressed as a linear polynomial t, thus the value of t can easily obtained from x as in eq. (3), and the curve can be flexibly designed according to the position and the size of the obstacles. However, at present, it is not guaranteed that the curve will be included in the feasible region, thus improvements of the method are required, such as not only suppressing the bulge of the curve but also adding new constraints so that the curve will be included in the feasible region.

4.2. Method 2: path planning with multiple Bézier curves

The second method is to construct a path connecting multiple 3-degree Bézier curves calculated under certain conditions. We use a Bézier curve of degree 3 because it is the curve of the smallest degree whose curvature can be controlled. This method makes effective use of the convex hull property mentioned above since it is guaranteed that the created curves do not go outside the feasible region.

We consider the connection of three cubic Bézier Curves $P(t), Q(t)$, and $R(t)$, where

$$P(t) = \sum_{i=0}^{3} \binom{3}{i} t^i (1-t)^{3-i} P_i, \quad Q(t) = \sum_{i=0}^{3} \binom{3}{i} t^i (1-t)^{3-i} Q_i, \quad R(t) = \sum_{i=0}^{3} \binom{3}{i} t^i (1-t)^{3-i} R_i. \tag{5}$$

Assume that, in eq. (5), $P_i, Q_i, R_i \in \mathbb{R}^3$ and we consider $P(t), Q(t), R(t)$ as vector-valued polynomials. As with the method in Section 4.1, we aim to draw a curve in which the y and the z coordinates exceed a certain value in a certain interval in the x coordinate, by drawing the first curve $P(t)$ from the starting point to that interval, the second curve $Q(t)$ avoiding the obstacles, and the third curve to the endpoint.

Since the endpoint of $P(t)$ and the start point of $Q(t)$, and the end point of $Q(t)$ and the start point of $R(t)$ are identical, respectively, we have $P(1) = Q(0)$ and $Q(1) = R(0)$, i.e., $P_3 = Q_0$ and $Q_3 = R_0$. In this path planning, we aim to express the other control points using the predefined points P_0, Q_0, R_0, R_3.

To make the connection smooth, we settle a constraint that the first and second derivatives at the connection points are equal for two adjacent curves, respectively. The necessary and sufficient conditions for $P'(1) = Q'(0)$ and $Q'(1) = R'(0)$ to hold are $2Q_0 = P_2 + Q_1$, $2R_0 = Q_2 + R_1$, respectively. Similarly, the necessary and sufficient conditions for $P''(1) = Q''(0)$ and $Q''(1) = R''(0)$ to hold are $2(Q_1 - P_2) = Q_2 - P_1$, $\quad 2(R_1 - Q_2) = R_2 - Q_1$, respectively [16]. So far, we see that the control points P_1, P_2, R_1, R_2 depend on Q_1 and Q_2 as

$$P_1 = 4Q_0 - 4Q_1 + Q_2, \quad P_2 = 2Q_0 - Q_1, \quad R_1 = 2R_0 - Q_2, \quad R_2 = Q_1 - 4Q_2 + 4R_0. \tag{6}$$

Now we determine Q_1 and Q_2. Let the feasible region be a hemisphere with a radius $r = \max\{\|P_0\|, \|Q_0\|, \|R_0\|, \|R_3\|\}$ centered at the origin and with $z > 0$. Here, $\|\cdot\|$ represents the norm of the vector. We require all remaining control points to be included in this hemisphere, i.e.,

$$\|P_i\| \le r, \quad \|Q_i\| \le r, \quad \|R_i\| \le r \quad (i = 1, 2),$$

which is reduced to solve the problem to find Q_1 and Q_2 that satisfy the above conditions. This problem is expressed as eliminating quantified variables in a quantified formula

$$\begin{aligned} \exists P_1 \exists P_2 \exists R_1 \exists R_2 ((P_1 = 4Q_0 - 4Q_1 + Q_2) \wedge (P_2 = 2Q_0 - Q_1) \wedge (R_1 = 2R_0 - Q_2) \wedge (R_2 = Q_1 - 4Q_2 + 4R_0) \\ \wedge (\|P_1\| \leqq r) \wedge (\|P_2\| \leqq r) \wedge (\|Q_1\| \leqq r) \wedge (\|Q_2\| \leqq r) \wedge (\|R_1\| \leqq r) \wedge (\|R_2\| \leqq r)). \end{aligned} \tag{7}$$

After solving eq. (7) and conditions on Q_1 and Q_2 are obtained, then choose Q_1 and Q_2 that makes $\|Q_1 - Q_0\|^2 + \|Q_2 - Q_1\|^2 + \|R_0 - Q_2\|^2$ as small as possible satisfying eq. (7), Then the rest of the control points $P_0, P_1, P_2, Q_0, R_0, R_1, R_2, R_3$ are determined, and the path of the end effector is obtained.

This method is promising since, if eq. (7) is solved, it gives a path that does not go beyond the feasible region and passes two points through for obstacle avoidance. Unfortunately, quantifier elimination for eq. (7) is computationally expensive and, at present, has not yet been successful. Therefore, by relaxing the original problem, we propose an alternative method to define a path in practical time.

4.2.1. Using 2-degree Bézier curves

In the alternative method, we create a path using three 2-degree Bèzier Curves

$$\hat{P}(t) = \sum_{i=0}^{2} \binom{2}{i} t^i (1-t)^{2-i} \hat{P}_i, \quad \hat{Q}(t) = \sum_{i=0}^{2} \binom{2}{i} t^i (1-t)^{2-i} \hat{Q}_i, \quad \hat{R}(t) = \sum_{i=0}^{2} \binom{2}{i} t^i (1-t)^{2-i} \hat{R}_i,$$

in place of $P(t), Q(t), R(t)$ in eq. (5), respectively, whose first-order differentials are identical at the connection points, together with $\hat{P}_2 = \hat{Q}_0, \hat{Q}_2 = \hat{R}_0$. Now we determine the other control points using only the predefined $\hat{P}_0 = P_0, \hat{Q}_0 = Q_0, \hat{R}_0 = R_0, \hat{R}_2 = R_3$. Since the first derivatives at the connection points are the same for two adjacent curves, we have $2\hat{Q}_0 = \hat{P}_1 + \hat{Q}_1, 2\hat{R}_0 = \hat{Q}_1 + \hat{R}_1$. For $r := \max\{\|\hat{P}_0\|, \|\hat{Q}_0\|, \|\hat{R}_0\|, \|\hat{R}_3\|\}$, we require all remaining control points to be included in the hemisphere centered at the origin with radius r, i.e., $\|\hat{P}_1\| \le r, \|\hat{Q}_1\| \le r, \|\hat{R}_1\| \le r$. Then, the problem is reduced to eliminate quantified variables in a quantified formula

$$\exists \hat{P}_1 \exists \hat{R}_1 ((\hat{P}_1 = 2\hat{Q}_0 - \hat{Q}_1) \wedge (\hat{R}_1 = 2\hat{R}_0 - \hat{Q}_1) \wedge (\|\hat{P}_1\| \le r) \wedge (\|\hat{Q}_1\| \le r) \wedge (\|\hat{R}_1\| \le r)). \tag{8}$$

Fortunately, by quantifier elimination, we have obtained the possible region of $\hat{Q}_1$ that satisfies eq. (8), then three Bézier Curves were obtained in the same way as described in the cubic Bézier Curves case. (The computation was done using the computer algebra system Wolfram Mathematica 13.3.1 in approximately 15.4 [s]. The computing environment is as follows: Intel Core i3-8130U 2.20GHz, 8GB RAM, Windows 11 Home.)

5. Concluding Remarks

We have proposed two methods for trajectory planning of manipulators using Bézier curves so that the generated curve does not go outside the feasible region. The first method can avoid obstacles with a minimum curve that matches the position and size of the obstacle, but it does not guarantee that any point on the curve is included in the region. The second method can guarantee that the curve does not go outside the region based on the constraint conditions, but its calculation cost is high and the position of the control point has not yet been obtained. To overcome this, we have proposed an alternative method using 2-degree Bézier curves, which can be calculated in a practical time.

Our future work includes the establishment of a better method for selecting the control points of a curve that overcomes current issues. We believe that method 2 is promising since the curve generated by the method is guaranteed to be included within the feasible region. The next issue to be solved is to improve computational efficiency by deriving more appropriate constraints, making the algorithm more efficient, and so on.

Once this method is established, we will move on to the stage of calculating the sequence of joint placements for the completed trajectory, leading to improved trajectory planning as proposed in our previous work [10].

References

[1] J. Capco, M. S. E. Din, J. Schicho, Robots, computer algebra and eight connected components, in: Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation, ISSAC '20, ACM, 2020. doi:10.1145/3373207.3404048.

[2] J.-C. Faugère, J.-P. Merlet, F. Rouillier, On solving the direct kinematics problem for parallel robots, Research Report RR-5923, INRIA, 2006. URL: https://hal.inria.fr/inria-00072366.

[3] C. M. Kalker-Kalkman, An implementation of Buchbergers' algorithm with applications to robotics, Mech. Mach. Theory 28 (1993) 523–537. doi:10.1016/0094-114X(93)90033-R.

[4] S. Ricardo Xavier da Silva, L. Schnitman, V. Cesca Filho, A Solution of the Inverse Kinematics Problem for a 7-Degrees-of-Freedom Serial Redundant Manipulator Using Gröbner Bases Theory, Mathematical Problems in Engineering 2021 (2021) 6680687. doi:10.1155/2021/6680687.

[5] T. Uchida, J. McPhee, Triangularizing kinematic constraint equations using Gröbner bases for real-time dynamic simulation, Multibody System Dynamics 25 (2011) 335–356. doi:10.1007/s11044-010-9241-8.

[6] T. Uchida, J. McPhee, Using Gröbner bases to generate efficient kinematic solutions for the dynamic simulation of multi-loop mechanisms, Mech. Mach. Theory 52 (2012) 144–157. doi:10.1016/j.mechmachtheory.2012.01.015.

[7] V. Weispfenning, Comprehensive Gröbner Bases, J. Symbolic Comput. 14 (1992) 1–29. doi:10.1016/0747-7171(92)90023-W.

[8] R. Fukasaku, H. Iwane, Y. Sato, Real Quantifier Elimination by Computation of Comprehensive Gröbner Systems, in: Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 173–180. doi:10.1145/2755996.2756646.

[9] S. Otaki, A. Terui, M. Mikawa, A Design and an Implementation of an Inverse Kinematics Computation in Robotics Using Real Quantifier Elimination based on Comprehensive Gröbner Systems, Preprint, 2021. doi:10.48550/arXiv.2111.00384, arXiv:2111.00384.

[10] M. Yoshizawa, A. Terui, M. Mikawa, Inverse Kinematics and Path Planning of Manipulator Using Real Quantifier Elimination Based on Comprehensive Gröbner Systems, in: Computer Algebra in Scientific Computing: CASC 2023, volume 14139 of Lecture Notes in Computer Science, Springer, 2023, pp. 393–419. doi:10.1007/978-3-031-41724-5_21.

[11] Y. Shirato, N. Oka, A. Terui, M. Mikawa, An Optimized Path Planning of Manipulator with Spline Curves Using Real Quantifier Elimination Based on Comprehensive Gröbner Systems, in: SCSS 2024 Work-in-progress Proceedings, Open Publishing Association, 2024. To appear.

[12] U. Dincer, M. Cevik, Improved trajectory planning of an industrial parallel mechanism by a composite polynomial consisting of bëzier curves and cubic polynomials, Mechanism and Machine Theory 132 (2019) 248–263. doi:10.1016/j.mechmachtheory.2018.11.009.

[13] Y.-L. Kuo, C.-C. Lin, Z.-T. Lin, Dual-optimization trajectory planning based on parametric curves for a robot manipulator, International Journal of Advanced Robotic Systems 17 (2020) 1729881420920046. doi:10.1177/1729881420920046.

[14] Z. Xu, S. Wei, N. Wang, X. Zhang, Trajectory planning with bezier curve in cartesian space for industrial gluing robot, in: X. Zhang, H. Liu, Z. Chen, N. Wang (Eds.), Intelligent Robotics and Applications, Springer International Publishing, Cham, 2014, pp. 146–154. doi:10.1007/978-3-319-13963-0_15.

[15] P. Bézier, The Mathematical Basis of the UNIURF CAD System, Elsevier, 1986. doi:10.1016/C2013-0-01005-5.

[16] T. Tsuchihashi, Trajectory Generation and Control for Manipulator Using the Bezier Curve (in Japansese), Transactions of the Japan Society of Mechanical Engineers Series C 55 (1989) 124–128. doi:10.1299/kikaic.55.124.