Received: 24 July 2025 | Revised: 30 September 2025 | Accepted: 10 April 2026 | Published online: 30 April 2026

RESEARCH ARTICLE

Journal of Climbing and Walking Robots
2026, Vol. 00(00) 1–10
DOI: 10.47852/bonviewJCWR62026914

BON VIEW PUBLISHING

New Trajectory Planning Based on Bezier Curve Using Particle Swarm Optimization

Chol Jun Han[1,*], Kwang Rim Song[2] and Sung-Gyu Pak[3]

[1] *Robotics Institute, Kim Chaek University of Technology, Democratic People's Republic of Korea*

[2] *Faculty of Automation Engineering, Kim Chaek University of Technology, Democratic People's Republic of Korea*

[3] *Department of Physics, University of Science, Democratic People's Republic of Korea*

Abstract: In this paper, a new trajectory planning based on Bezier curve is proposed to generate a smooth and time-optimal trajectory for point-to-point motions. The 10th-order Bezier curve is used to generate the path, and the trajectory with time is generated by time re-parameterization based on the fourth-order Bezier curve. The trajectory is determined by only two parameters. Then, using the accelerated particle swarm optimization method, the minimum execution time and two parameters under kinematic constraints (velocity, acceleration, and jerk) are determined to obtain the time-optimal trajectory. The obtained trajectory ensures that the velocity, acceleration, and jerk values at the beginning and end points are all zero and also guarantees the flexibility of the motion. The synchronization of all joints is also given for the effective actuator operation. The simulation of a 6-degree-of-freedom (DOF) robot manipulator shows the effectiveness of the proposed method to satisfy the optimal execution time. Also, the analysis using a mass-spring-damper system with a single-DOF shows well the characteristics for reducing residual vibration.

Keywords: trajectory planning, Bezier curve, particle swarm optimization, robotic manipulator, point-to-point motion

1. Introduction

Nowadays, robots are widely used in all fields, including the economy and military, and their importance and significance are highlighted. In particular, robots have become more and more important devices due to their superior advantages in automation and unmanned manufacturing in the industrial field [1–5]. The robot should not only move quickly but also move smoothly to ensure high productivity and quality. To this end, a number of works have been carried out by many researchers, where point-to-point trajectory planning of robots is an important research issue [6–9].

Polynomials, B-splines, and Bezier curves have been widely used to generate motion trajectories [10–13]. The trajectory generation is carried out using third-, fourth-, and fifth-order polynomials, but the acceleration values at the beginning and end of the motion are nonzero [14]. Boryga and Graboś [15] presented acceleration profiles with fifth-, seventh-, and ninth-order polynomials. The fifth-order B-spline function was used to generate trajectories satisfying jerk's continuity under kinematic constraints [16]. Liu et al. [17] used cubic splines in the workspace and septuple B-splines in the joint space to generate time-optimal and jerk-continuous trajectories. Wang et al. [18] planned trajectories with a fourth-order polynomial determined by one coefficient. Gasparetto and Zanotto [19, 20] generated trajectories by cubic B-splines and quintic splines. Trajectory planning of industrial parallel mechanisms has been carried out by means of a composite polynomial of Bezier curves and cubic polynomials [21]. Liu et al. [22] proposed a flexible trajectory planning method for parallel manipulators with jerk constraints, and the trajectory is planned with a fifth-order polynomial spline. With a fifth-order B-spline or a fifth-order polynomial spline, displacement, velocity, acceleration, and jerk were all continuous. Alfatih et al. [23] performed velocity control of a mobile robot with trajectory planning based on Bezier curve. Zhao et al. [24] used a fifth-order polynomial for point-to-point trajectory planning of an under-constrained cable-suspended parallel robot. Boryga [25] performed trajectory planning by expressing the acceleration of the end-effector of the robot by a seventh-order polynomial. Liu et al. [22] generated a flexible trajectory for a fast pick-and-place parallel robot using a fifth-order B-spline. In general, the displacement, velocity, acceleration, and jerk of the trajectory with higher-order polynomial or higher-order spline functions are all continuous and flexible, but many coefficients should be determined to generate the trajectory [26, 27]. Optimization algorithms are widely used as a method to determine the trajectories and coefficients corresponding to different objectives (time optimization, jerk optimization, energy optimization, etc.) [28, 29].

Kucuk [30] generated a trajectory that achieves minimum motion by combining the seventh-order polynomial and cubic spline using the particle swarm optimization (PSO) algorithm for

***Corresponding author:** Chol Jun Han, Robotics Institute, Kim Chaek University of Technology, Democratic People's Republic of Korea. Email: hcj901128@star-co.net.kp

© The Author(s) 2026. Published by BON VIEW PUBLISHING. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

serial and parallel manipulators. Machmudah et al. [31] used a genetic algorithm (GA) and PSO to find a feasible sixth-order polynomial joint trajectory in complex geometries. Xin et al. [32] carried out the trajectory planning using the PSO algorithm to minimize the residual vibration of the spatial robot. Song et al. [33] used an improved PSO algorithm to schedule the continuous trajectory of a mobile robot using higher-order Bezier curves. Chen et al. [34] proposed an improved fifth-order B-spline interpolation model, using a quantum-behaved PSO algorithm to determine the optimal time parameters. A robotic trajectory planning particle swarm optimization algorithm was introduced to optimize the joint angles or paths of mechanical arm movements [35]. For the time-optimal trajectory planning problem during robot arm motion, the segmented polynomial interpolation function-based method with a locally chaotic PSO algorithm was used [36]. To solve the robot trajectory planning problem with short running time, a time-optimal trajectory planning algorithm was proposed based on the improved simplified PSO [37]. A trajectory competitive multi-objective PSO algorithm was proposed to search for a set of Pareto optimal solutions of the time-energy-jerk optimal trajectory of the robot [38]. The trajectory planning was developed using the PSO algorithm to determine the position of the robot at each point during the motion [39].

In this paper, we propose a time-optimal trajectory planning method based on higher-order Bezier curves. The 10th-order Bezier curve and the fourth-order Bezier curve are determined by two parameters, based on which the trajectory satisfying the kinematic constraints is given. Then, the accelerated PSO algorithm is adopted to determine the minimum execution time of the trajectory.

The rest of this paper is organized as follows. Section 2 presents the trajectory planning method based on Bezier curve. Section 3 presents the algorithms to determine the minimum execution time and two parameters using the accelerated PSO method and presents the time-based synchronization strategy. Section 4 presents the trajectory analysis for the 6-degree-of-freedom (DOF) robot manipulator and the residual vibration estimation for the tracking efficiency of the trajectory. The conclusion is drawn in Section 5.

2. Trajectory Planning Based on Bezier Curve

2.1. Path generation by 10th-order Bezier curve

In general, the nth-order Bezier curve is defined as:

$$\theta(u) = \sum_{i=0}^{n} \binom{n}{i} (1-u)^{n-i} u^i P_i \tag{1}$$

Expanding Equation (1) yields Equation (2):

$$\begin{aligned} \theta(u) = {} & (1-u)^n P_0 + \binom{n}{1}(1-u)^{n-1} u P_1 \\ & + \cdots + \binom{n}{n-1}(1-u) u^{n-1} P_{n-1} \\ & + u^n P_n, u \in [0,1] \end{aligned} \tag{2}$$

Where:

P_0, P_n: the start point and end point of Bezier curve,

$P_1, ..., P_{n-1}$: control points of Bezier curve,

u: parameter influencing the distribution of interpolation points.

Rewriting Equation (1) yields Equation (3):

$$\theta(u) = \sum_{i=0}^{n} b_{i,n}(u) P_i, \quad u \in [0,1] \tag{3}$$

The polynomial shown in Equation (4) is called the Bernstein-based polynomials of degree n.

$$b_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i}, \quad i = 0, ..., n \tag{4}$$

Also,

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \tag{5}$$

We use the 10th-order Bezier curve to generate a flexible path. Based on the given control points $P_0, P_1, \cdots, P_9, P_{10}$, a 10th-order Bezier curve can be derived as follows:

$$\begin{aligned} \theta(u) &= \sum_{i=0}^{10} \frac{10!}{i!(10-i)!} u^i (1-u)^{10-i} P_i \\ &= (1-u)^{10} P_0 + \binom{10}{1}(1-u)^9 u P_1 \\ &\quad + \cdots + \binom{10}{10-1}(1-u) u^9 P_9 \\ &\quad + u^{10} P_{10}, u \in [0,1] \end{aligned} \tag{6}$$

Expanding the above expression in matrix form yields Equation (7):

$$\theta(u) = \mathbf{u^T M p} \quad 0 \le u \le 1 \tag{7}$$

Where:

$$\mathbf{M} = \begin{bmatrix} 1 & -10 & 45 & -120 & 210 & -252 & 210 & -120 & 45 & -10 & 1 \\ -10 & 90 & -360 & 840 & -1260 & 1260 & -840 & 360 & -90 & 10 & 0 \\ 45 & -360 & 1260 & -2520 & 3150 & -2520 & 1260 & -360 & 45 & 0 & 0 \\ -120 & 840 & -2520 & 4200 & -4200 & 2520 & -840 & 120 & 0 & 0 & 0 \\ 210 & -1260 & 3150 & -4200 & -3150 & -1260 & 210 & 0 & 0 & 0 & 0 \\ -252 & 1260 & -2520 & 2520 & -1260 & 252 & 0 & 0 & 0 & 0 & 0 \\ 210 & -840 & 1260 & -840 & 210 & 0 & 0 & 0 & 0 & 0 & 0 \\ -120 & 360 & -360 & 120 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 45 & -90 & 45 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -10 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{u} = \begin{bmatrix} u^{10} \\ u^9 \\ \vdots \\ u \\ 1 \end{bmatrix}, \mathbf{p} = \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_9 \\ P_{10} \end{bmatrix}$$

2.2. Trajectory planning with time re-parameterized Bezier curve

In trajectory planning, the motion planning well adapted to the trajectory path ensures good performance in programming the motion of robotic manipulators and automated machines. We associate the parameter with time to generate a trajectory with

time. The relationship between the parameter and time is set by a fourth-order Bezier curve as Equation (8):

$$\begin{cases} u(t_r) = \sum_{j=0}^{4} b_{j,4}(t_r) W_j \\ t_r = \dfrac{t}{t_{total}} \end{cases} \tag{8}$$

where $W_0, ..., W_4$ $(0 \le W_j \le 1, j = 0 \sim 4)$ are control points of the curve parameter u and t_{total} is the total motion time from the start point to the end point, $t \in [0, t_{total}]$. Then the control points of u are set as follows:

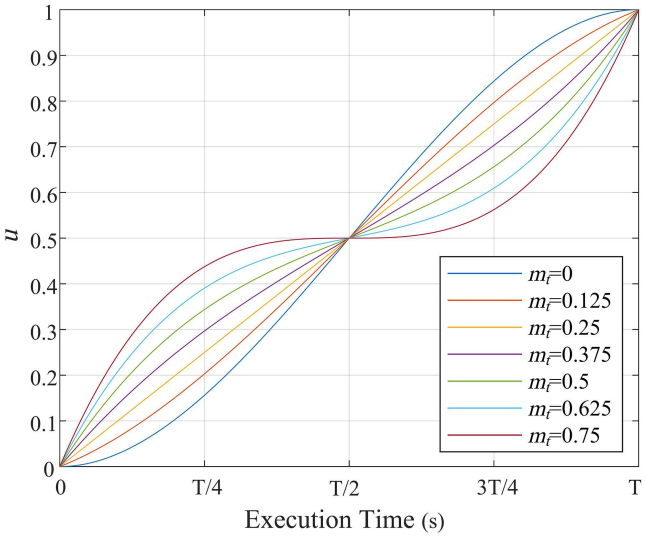
$$\mathbf{w} = \begin{bmatrix} W_0 \\ W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} = \begin{bmatrix} 0 \\ m_t \\ 0.5 \\ 1 - m_t \\ 1 \end{bmatrix} \tag{9}$$

Since the curve parameter u must have a value of [0, 1] when t varies between $[0, t_{total}]$, it must be $W_0 = 0$, $W_4 = 1$. Also, to satisfy the symmetry of the acceleration and deceleration stages of the trajectory, it is set as in Equation (9). Here, it should be noted that, by the property of the Bezier curve, if the control points are not properly selected, u does not monotonically increase. The relationship between u and t is shown in Figure 1.

As a result, the trajectory function $\theta(t)$ with time is formed. Then the velocity, acceleration, and jerk of the trajectory are time derivatives of the re-parameterized trajectory with respect to time, respectively. Equation (10) is the velocity, acceleration, and jerk functions:

$$\begin{cases} v(t) = \dfrac{d\theta(t)}{dt} \\ a(t) = \dfrac{d\theta^2(t)}{dt^2} \\ j(t) = \dfrac{d\theta^3(t)}{dt^3} \end{cases} \tag{10}$$

Figure 1
The relationship curve between u and t with m_t



Calculating Equation (9) using Equation (8) yields Equation (11):

$$\begin{cases} v(t) = \dfrac{1}{t_{total}} \dfrac{d\theta(u)}{du} \dfrac{du(t_r)}{dt_r} \\ a(t) = \dfrac{1}{t_{total}^2} \left[\dfrac{d^2\theta(u)}{du^2} \left(\dfrac{du(t_r)}{dt_r} \right)^2 + \dfrac{d\theta(u)}{du} \dfrac{d^2u(t_r)}{dt_r^2} \right] \\ j(t) = \dfrac{1}{t_{total}^3} \left[\dfrac{d^3\theta(u)}{du^3} \left(\dfrac{du(t_r)}{dt_r} \right)^3 + 3 \dfrac{d^2\theta(u)}{du^2} \dfrac{du(t_r)}{dt_r} \dfrac{d^2u(t_r)}{dt_r^2} \right. \\ \qquad \left. + \dfrac{d\theta(u)}{du} \dfrac{d^3u(t_r)}{dt_r^3} \right] \end{cases} \tag{11}$$

The next important issue is to determine the control points so that the trajectory satisfies the kinematic constraints. That is, to satisfy the smooth start and end, the values of velocity, acceleration, and jerk at the beginning and end of the movement must be zero, and it is to determine the path control points and the control points of the curve parameter so that the maximum values of velocity, acceleration, and jerk during the movement do not exceed the limit value. The velocity, acceleration, and jerk are obtained as follows:

$$\begin{cases} v(t)|_{t=0} = \dfrac{1}{t_{total}} (-10P_0 + 10P_1)(-4W_0 + 4W_1) = 0 \\ a(t)|_{t=0} = \dfrac{1}{t_{total}^2} (90P_0 - 180P_1 + 90P_2) \\ \qquad\qquad (12W_0 - 24W_1 + 12W_2) = 0 \\ j(t)|_{t=0} = \dfrac{1}{t_{total}^3} (-720P_0 + 2160P_1 - 2160P_2 + 720P_3) \\ \qquad\qquad (-24W_0 + 72W_1 - 72W_2 + 24W_3) = 0 \end{cases} \tag{12}$$

It is clear from Equation (12) that $P_0 = P_1 = P_2 = P_3$. Similarly, for a value of zero velocity, acceleration, and acceleration at $t = t_{total}$, $P_7 = P_8 = P_9 = P_{10}$ must be required. To satisfy the symmetry of the acceleration and deceleration stages of the trajectory, P_4, P_5 and P_6 are set as follows in Equation (13):

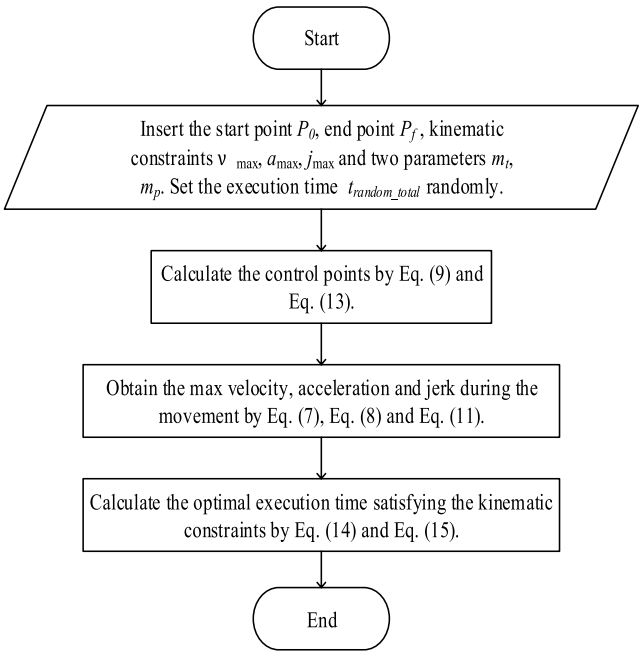
$$\begin{cases} P_4 = P_0 + m_p(P_{10} - P_0) \\ P_5 = (P_0 + P_{10})/2 \\ P_6 = P_0 + (1 - m_p)(P_{10} - P_0) \end{cases} \tag{13}$$

Here, in order for the displacement to increase monotonically, $m_p \in [0, 1.5]$ by the property of the Bezier curve. Consequently, the problem of planning a flexible and time-optimal trajectory is reduced to the problem of determining the parameters m_t and m_p satisfying kinematic constraints and minimum travelling time. As shown in Equation (11), when the control points are given, the velocity, acceleration, and jerk are proportional to the execution time. Therefore, when the maximum values of velocity, acceleration, and jerk for any execution time t_{random_total} are given by $v_{r\max}$, $a_{r\max}$, $j_{r\max}$, the actual optimal traveling time t_{actual_total} satisfying the kinematic constraint $v_{\max}$, $a_{\max}$, $j_{\max}$ is obtained as Equation (14):

$$\tau = \max \left\{ \frac{v_{r\max}}{v_{\max}}, \sqrt{\frac{a_{r\max}}{a_{\max}}}, \sqrt[3]{\frac{j_{r\max}}{j_{\max}}} \right\} \tag{14}$$

Given m_t and m_p, the $t_{actual\_total} = \tau \times t_{random\_total}$ algorithm to obtain the optimal execution time satisfying the kinematic constraints is shown in Figure 2.

Figure 2
Algorithm for time-optimal motion under kinematic constraints



Note: Algorithm for calculating optimal execution time satisfying kinematic constraints when m_t and m_p are given.

3. Time-Optimal Trajectory Using Particle Swarm Optimization Method

3.1. Determination of minimum execution time

As shown in Figure 3, considering the optimal execution time with a variation of m_t and m_p under a given kinematic constraint, it can be seen that the optimal value is easily obtained by the PSO algorithm.

PSO is an intelligent optimization algorithm proposed to simulate the swarm behavior of animals, for example, fish and birds. Several studies show that PSO algorithms can outperform GAs and other traditional algorithms to solve many optimization problems. We use an accelerated PSO method to obtain the minimum execution time. In the accelerated PSO, the velocity vector is generated in Equation (15):

$$v_i^{k+1} = v_i^k + \alpha(r - 0.5) + \beta(x_{gb}^k - x_i^k) \tag{15}$$

where r is a random value between 0 and 1. The updating of the position is calculated as Equation (16):

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{16}$$

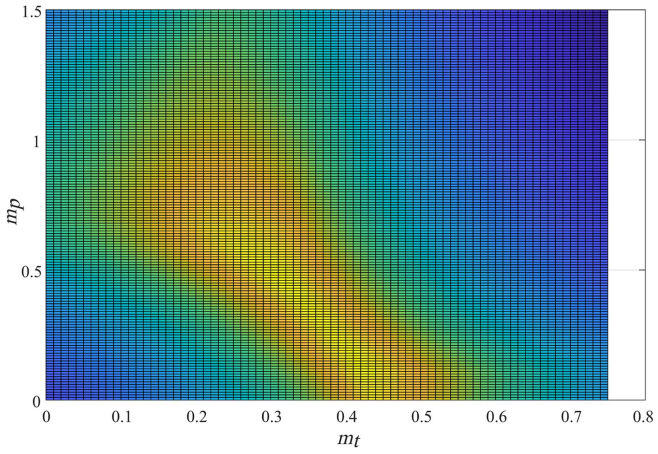
To further increase the convergence, we can further simplify:

$$x_i^{k+1} = (1 - \beta)x_i^k + \beta x_{gb}^k + \alpha(r - 0.5) \tag{17}$$

where α and β are the learning parameters and k is the iteration number. The paper uses Equation (17) and sets the objective function $f(x)$ for execution time. The algorithm flowchart for obtaining m_t and m_p, which allows to achieve the minimum execution time using the accelerated PSO method, is as follows.

With this algorithm, we can determine the minimum execution time and its corresponding m_t and m_p.

Figure 3
Optimal execution time with a variation of m_t and m_p



3.2. Synchronization of motion trajectories

Since kinematic constraints and target values are different for all joints of industrial robots, their execution times are also different. Therefore, in order to reduce the load of actuators, synchronization of motion trajectories needs to be performed. The paper applies time-based synchronization to ensure that all joints can simultaneously leave and stop at a specified position. The synchronization time is set as Equations (18):

$$T_{\min}^{sync} = \max\{T_k\}(k = 1, 2, \cdots, N) \tag{18}$$

Then, the synchronization coefficient is shown in Equation (19):

$$c_k = \frac{T_k}{T_{\min}^{sync}} \tag{19}$$

The velocity, acceleration, and jerk with time of all joints are as follows:

$$\begin{cases} v_k^{sync}(t) = c_k v_k(t) \\ a_k^{sync}(t) = c_k^2 a_k(t) \\ j_k^{sync}(t) = c_k^3 j_k(t) \end{cases} \tag{20}$$

Since $c_k \leq 1$, as shown in Equations (19) and (20), the velocity, acceleration, and jerk of joints after synchronization will decrease. In particular, the jerk becomes smaller. This can reduce the load on the actuator and increase the service life of the robot.

4. Results and Discussion

4.1. Trajectory planning of 6-DOF robotic manipulator

We use this method to perform trajectory planning of a 6-DOF robotic manipulator to show the advantages of the proposed method. The trajectory is generated in the joint space, and the simulated data are adopted from several literatures [6–9]. Table 1 shows the kinematic constraints and position data of the joints. And to apply the accelerated PSO algorithm, we set the number of particles $n = 50$ and the number of iterations *iterations_num* = 30, $\alpha = 0.7^k$, and $\beta = 0.5$.

Table 1
Position data and kinematic constraints of the robot joints

		Joint					
		1	2	3	4	5	6
Positions (rad)	Initial point (rad)	0	$-\pi/6$	0	$-\pi/3$	0	0
	Final point (rad)	$2\pi/3$	$\pi/6$	$\pi/4$	$\pi/3$	$-\pi/4$	$\pi/6$
Kinematic constraints	Velocity (rad/s)	8	10	10	5	5	5
	Acceleration (rad/s^2)	10	12	12	8	8	8
	Jerk (rad/s^3)	30	40	40	20	20	20

Figure 4
Motion profiles generated by time-based synchronization

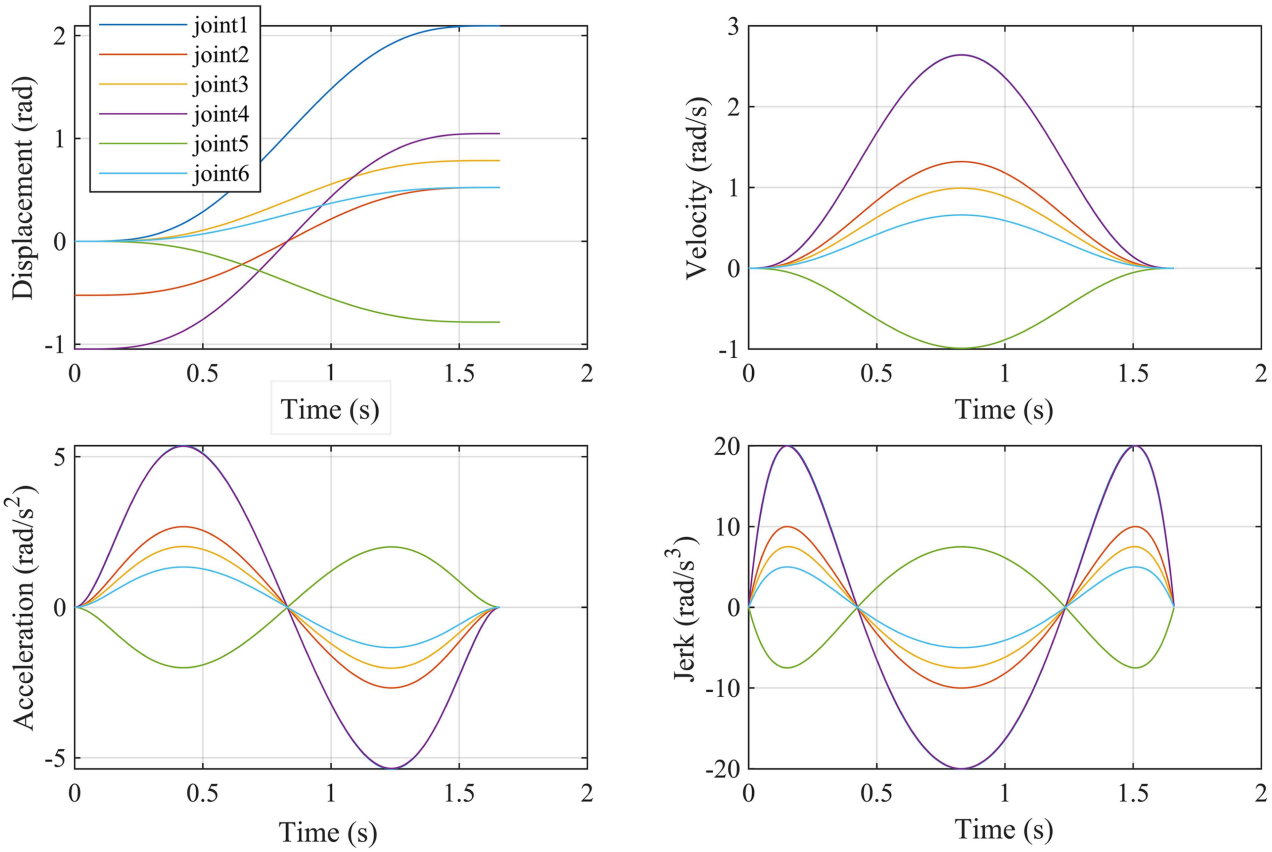


Figure 4 shows the results calculated by the method proposed in this paper. The position, velocity, acceleration, and jerk of all joints satisfy the kinematic constraints shown in Table 1. At the beginning and end of the movement, the values of velocity, acceleration, and jerk become zero, and a flexible trajectory is drawn. Table 2 shows the actual maximum velocity, acceleration, and acceleration before and after applying time-based synchronization. It can be seen from Table 2 that synchronization can significantly reduce the load on the actuators during movement.

Then we performed the calculations using the kinematic constraints shown in Table 1 in order to make comparisons with the methods in the references [6–9, 18]. Table 3 presents the execution time and maximum jerk values calculated by different methods, and Figure 5 shows the displacement profiles for joint 4, which are generated by the benchmark methods and the proposed method.

The results show that the proposed method generates less execution time than the benchmark methods [6, 7, 9, 18]. However, it is larger than the execution time obtained by the work of Wu et al. [8]. However, without parameters such as α and β introduced in refenrence by Wu et al. [8], it is possible to produce unique minimum execution time and time-optimal trajectories under kinematic constraints. And comparing with the actual maximum jerk values, the actual maximum jerk values obtained by the proposed method are larger than those obtained by the works of Perumaal and Jawahar [6] and Wang et al. [18]. However, the execution time is much faster. This increases reliability and availability.

4.2. Estimation of residual vibration

To evaluate the efficiency of the proposed method, we perform residual vibration analysis.

The dynamic model of the simplified uniaxial motion stage is shown in Figure 6. It includes a flexible base structure with a weight of m_1, which includes a motor for moving mass with equal weight of m_2. The frictional force between the moving mass and the machine base is neglected. The dynamics of the flexible base is assumed to be a single-DOF mass-spring-damper system with spring stiffness k and damping coefficient c.

Table 2
The actual max velocity, acceleration, and jerk values before and after synchronization

Velocity		Acceleration		Jerk	
Before synchronization	After synchronization	Before	After	Before	After
3.02(100%)	2.64(87.3%)	7.02(100%)	5.36(76.3%)	29.99(100%)	20(66.7%)
2.1(100%)	1.32(63.0%)	6.74(100%)	2.67(39.7%)	40(100%)	10(25.0%)
1.73(100%)	0.99(57.2%)	6.15(100%)	2.02(32.8%)	40(100%)	7.52(18.5%)
2.64(100%)	2.64(100%)	5.34(100%)	5.34(100%)	20(100%)	20(100%)
1.37(100%)	0.99(72.1%)	3.85(100%)	2.00(52.0%)	20(100%)	7.5(37.5%)
1.04(100%)	0.66(63.0%)	3.37(100%)	1.33(39.7%)	20(100%)	5(25.0%)

Table 3
Comparative results of several planning methods

		Performance measure	
Work	Trajectory model	Execution time (s)	Maximum Jerk (rad/s^3)
Perumaal and Jawahar [6]	3-segment sine jerk (sync acc-dec)	2.513	6.63 (joint 1, 4)
Fang et al. [7]	15-segment sigmoid jerk	1.876	20.30 (joint 1) (Smax=150 rad/s^4)
Wu et al. [8]	Locally asymmetrical 15-segment sine jerk	1.6286 ($\alpha = 1, \beta = 0.1$)	20 (joint 1, 4)
Han et al. [9]	Asymmetric 15-segment sigmoid jerk	1.816	24.90 (joint 1) (Smax = 273.05 rad/s^4)
Wang et al. [18]	High-order polynomial jerk	4.609	1.526 (joint 1)
Present	Higher-order Bezier curve	1.6594	20 (joint 1, 4)

Figure 5
Motion profiles of joint 4 for different trajectory planning method

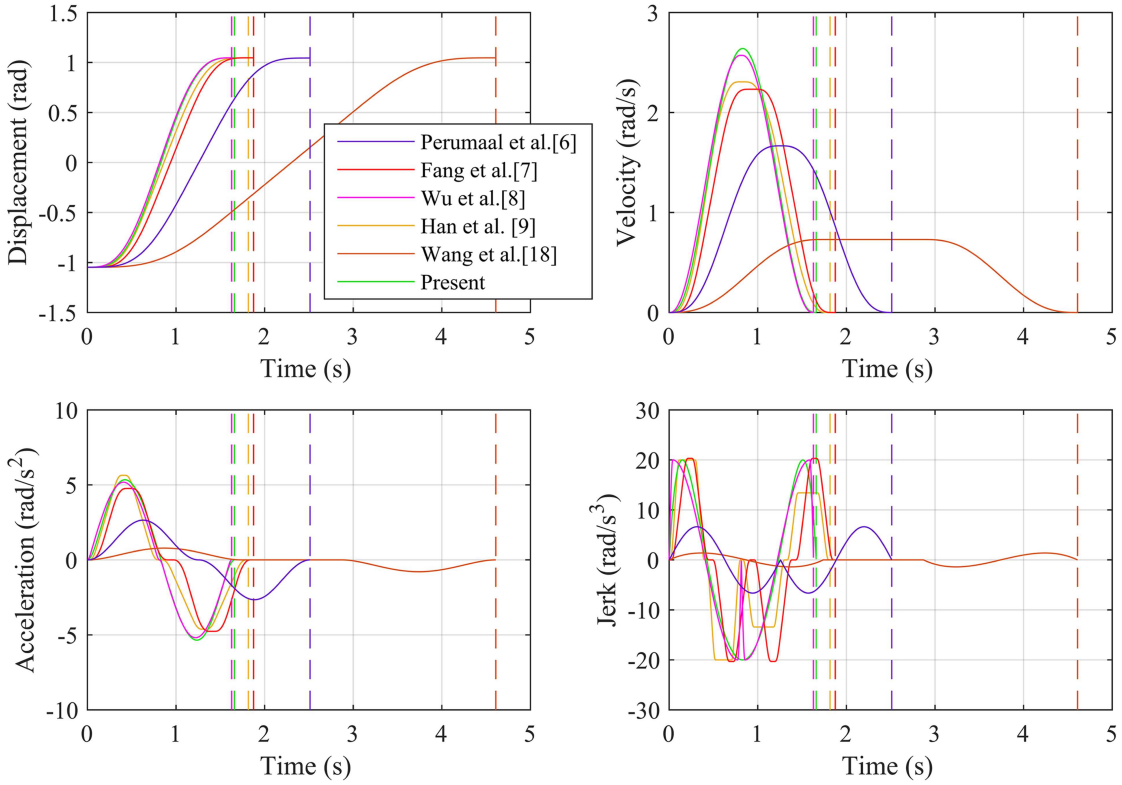
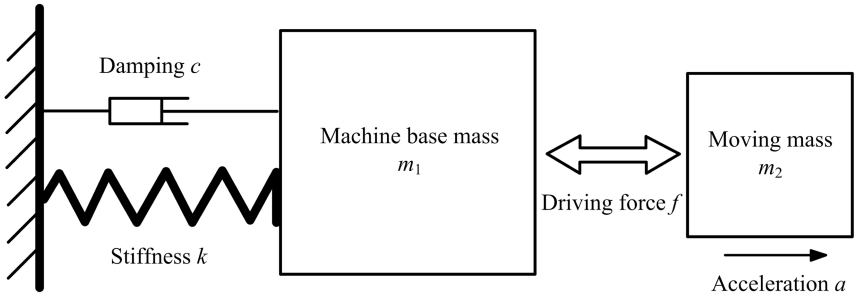


Figure 6
A single-degree-of-freedom mass-spring-damper system



Where the moving mass m_2 is driven with an acceleration a, the reaction force on the base structure m_1 will be $F = m_2 a$. The dynamics of the base structure can be expressed as

$$m_1\ddot{x} + c\dot{x} + kx = F = m_2 a \tag{21}$$

or

$$\ddot{x} + 2\xi\dot{x} + \omega_n^2 x = \frac{m_2}{m_1} a \tag{22}$$

where ω_n is the angular natural frequency of the base structure, ξ is the damping ratio, and x is the vibration response of the base structure.

To carry out the simulation study, it is assumed that the ratio of the moving mass over the machine base is 0.1, the undamped natural frequency of the base is 24 Hz, with a damping ratio of 3%, that is, $m_2/m_1 = 0.1$, $\omega_n = 150.8 rad/s$ (i.e., f_n = 24 Hz) and ξ = 3% [40]. For comparison, the profiles generated by the locally asymmetric trajectory planning method and the proposed method are evaluated. The residual vibration of the base frame is then determined by Equation (22). Kinematic constraints are shown in Table 4 [9, 40].

We set $\alpha = 1$, $\beta = 0.972$ for a locally asymmetric 15-segment sine jerk. The calculated motion profiles are shown in Figure 7, and the vibration of the base structure caused by stage motion using both profiles is shown in Figure 8. As shown in Figure 7, the total execution time for both profiles is 1 s. It can then be seen that the residual vibration caused by the locally asymmetric sinusoidal jerk profile is larger, with a peak-to-peak value of about 4.8 μm. The peak-to-peak value of the residual vibration caused by the proposed method is about 2.6 μm. The results show that the proposed method has good performance for residual vibration reduction, which indicates that the proposed method is suitable for high-speed point-to-point motion.

Table 4
Displacement and kinematic constraints

Displacement (m)		0.8
Kinematic constraints	Velocity (m/s)	1.24
	Acceleration (m/s^2)	6
	Jerk (m/s^3)	80

Figure 7
Motion profiles generated by both trajectory planning

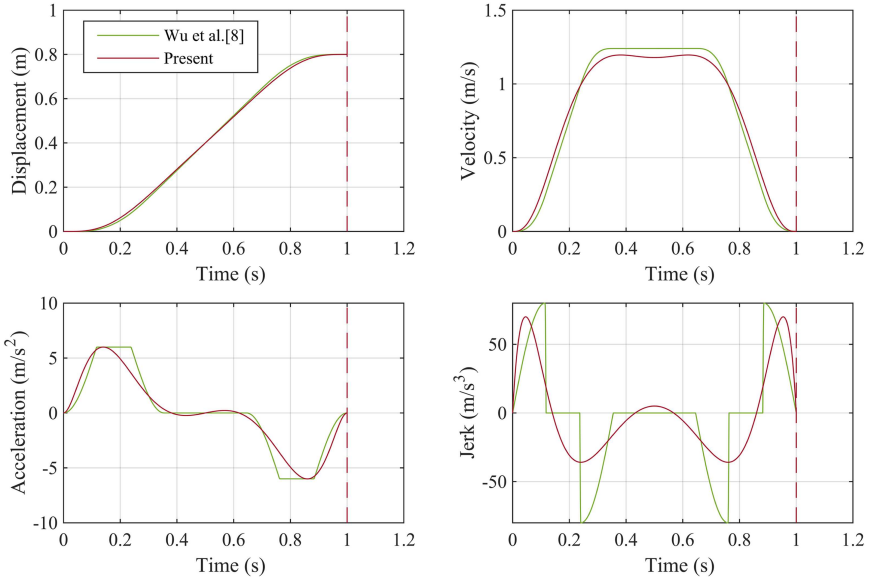
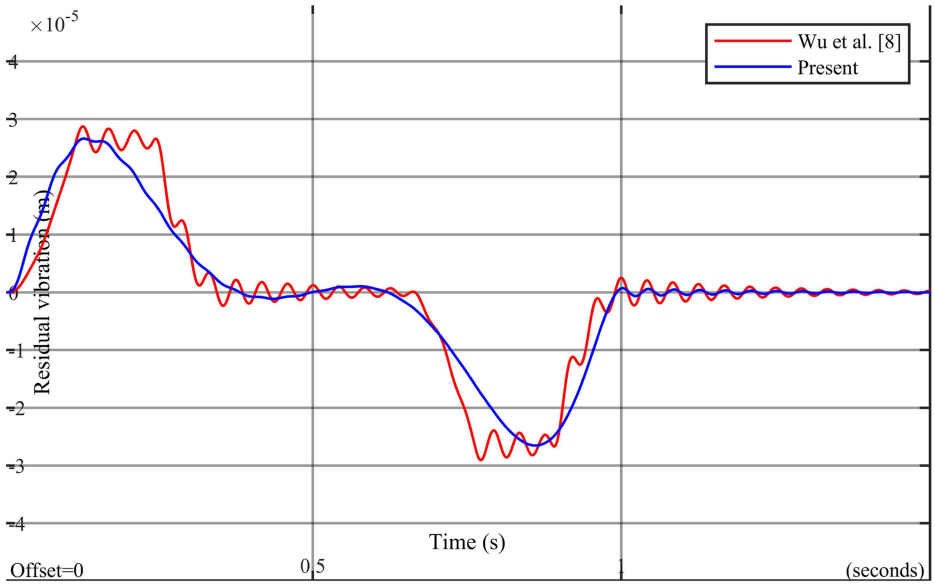


Figure 8
Residual vibration by two methods



5. Conclusion

In this paper, a new trajectory planning method based on higher-order Bezier curves is proposed to generate flexible and time-optimal trajectories for point-to-point motions. The 10th-order Bezier curve used to generate the path and the fourth-order Bezier curve for the time re-parameterization were determined by one parameter, respectively. Then, using the accelerated PSO method, the minimum execution time and the corresponding two parameters under kinematic constraints (velocity, acceleration, and jerk) were determined to generate the time-optimal trajectory. By time-based synchronization, we generated synchronization trajectories of all joints that effectively reduce the load of actuators. The simulation of trajectory planning of a 6-DOF robot manipulator has shown the effectiveness and feasibility of the proposed method through comparison with other benchmark methods. In addition, the analysis using a mass-spring-damper system with a single-DOF showed well the characteristics of this method for reducing residual vibration. Future work will be devoted to apply the proposed method to precise work requiring complex spatial paths across multiple path points continuously.

Acknowledgment

This work was supported by Kim Chaek University of Technology, Democratic People's Republic of Korea. The supports are gratefully acknowledged.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data are available from the corresponding author upon reasonable request.

Author Contribution Statement

Chol Jun Han: Conceptualization, Methodology, Validation, Data curation, Writing – original draft, Supervision, Project administration. **Kwang Rim Song:** Software, Resources, Writing – review & editing. **Sung-Gyu Pak:** Formal analysis, Investigation, Visualization.

References

[1] Zhang, T., Zhang, M., & Zou, Y. (2021). Time-optimal and smooth trajectory planning for robot manipulators. *International Journal of Control, Automation and Systems*, *19*(1), 521–531. https://doi.org/10.1007/s12555-019-0703-3

[2] Alqudsi, Y. S., Kassem, A. H., & El-Bayoumi, G. (2023). A general real-time optimization framework for polynomial-based trajectory planning of autonomous flying robots. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *237*(1), 29–41. https://doi.org/10.1177/09544100221090690

[3] Hu, G., Ma, J., Wang, Y., Su, J., & Zhou, Z. (2023). Design and optimization of high flexible motion profile for high acceleration to reduce vibration. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *45*(12), 619. https://doi.org/10.1007/s40430-023-04520-3

[4] Carabin, G., & Scalera, L. (2020). On the trajectory planning for energy efficiency in industrial robotic systems. *Robotics*, *9*(4), 89. https://doi.org/10.3390/robotics9040089

[5] Liu, T., Cui, J., Li, Y., Gao, S., Zhu, M., & Chen, L. (2023). Time-optimal asymmetric s-curve trajectory planning of redundant manipulators under kinematic constraints. *Sensors*, *23*(6), 3074. https://doi.org/10.3390/s23063074

[6] Perumaal, S., & Jawahar, N. (2012). Synchronized trigonometric S-curve trajectory for jerk-bounded time-optimal pick and place operation. *International Journal of Robotics and Automation*, *27*(4), 385–395. https://doi.org/10.2316/Journal.206.2012.4.206-3780

[7] Fang, Y., Hu, J., Liu, W., Shao, Q., Qi, J., & Peng, Y. (2019). Smooth and time-optimal S-curve trajectory planning for automated robots and machines. *Mechanism and Machine Theory*, *137*, 127–153. https://doi.org/10.1016/j.mechmachtheory.2019.03.019

[8] Wu, Z., Chen, J., Bao, T., Wang, J., Zhang, L., & Xu, F. (2022). A novel point-to-point trajectory planning algorithm for industrial robots based on a locally asymmetrical jerk motion profile. *Processes*, *10*(4), 728. https://doi.org/10.3390/pr10040728

[9] Han, C. J., Song, K. R., & Rim, U.-R. (2024). An asymmetric S-curve trajectory planning based on an improved jerk profile. *Robotica*, *42*(7), 2184–2208. https://doi.org/10.1017/S0263574724000031

[10] Li, X., Lv, H., Zeng, D., & Zhang, Q. (2023). An improved multi-objective trajectory planning algorithm for kiwifruit harvesting manipulator. *IEEE Access*, *11*, 65689–65699. https://doi.org/10.1109/ACCESS.2023.3289207

[11] Wu, G., & Zhang, S. (2022). Real-time jerk-minimization trajectory planning of robotic arm based on polynomial curve optimization. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *236*(21), 10852–10864. https://doi.org/10.1177/09544062221106632

[12] Stretti, D., Fanghella, P., Berselli, G., & Bruzzone, L. (2023). Analytical expression of motion profiles with elliptic jerk. *Robotica*, *41*(7), 1976–1990. https://doi.org/10.1017/S0263574723000255

[13] Boryga, M. (2020). The use of higher-degree polynomials for trajectory planning with jerk, acceleration and velocity constraints. *International Journal of Computer Applications in Technology*, *63*(4), 337–347. https://doi.org/10.1504/IJCAT.2020.110414

[14] Jond, H. B., Nabiyev, V. V., & Benveniste, R. (2016). Trajectory planning using high order polynomials under acceleration constraint. *Journal of Optimization in Industrial Engineering*, *10*(21), 1–6. https://doi.org/10.22094/JOIE.2016.255

[15] Boryga, M., & Graboś, A. (2009). Planning of manipulator motion trajectory with higher-degree polynomials use. *Mechanism and Machine Theory*, *44*(7), 1400–1419. https://doi.org/10.1016/j.mechmachtheory.2008.11.003

[16] Junsen, H., Pengfei, H., Kaiyuan, W., & Min, Z. (2018). Optimal time-jerk trajectory planning for industrial robots. *Mechanism and Machine Theory*, *121*, 530–544. https://doi.org/10.1016/j.mechmachtheory.2017.11.006

[17] Liu, H., Lai, X., & Wu, W. (2013). Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. *Robotics and Computer-Integrated Manufacturing*, *29*(2), 309–317. https://doi.org/10.1016/j.rcim.2012.08.002

[18] Wang, H., Wang, H., Huang, J., Zhao, B., & Quan, L. (2019). Smooth point-to-point trajectory planning for industrial robots with kinematical constraints based on high-order polynomial curve. *Mechanism and Machine Theory*, *139*, 284–293. https://doi.org/10.1016/j.mechmachtheory.2019.05.002

[19] Gasparetto, A., & Zanotto, V. (2008). A technique for time-jerk optimal planning of robot trajectories. *Robotics and Computer-Integrated Manufacturing*, *24*(3), 415–426. https://doi.org/10.1016/j.rcim.2007.04.001

[20] Gasparetto, A., & Zanotto, V. (2007). A new method for smooth trajectory planning of robot manipulators. *Mechanism and Machine Theory*, *42*(4), 455–471. https://doi.org/10.1016/j.mechmachtheory.2006.04.002

[21] Dinçer, Ü., & Çevik, M. (2019). Improved trajectory planning of an industrial parallel mechanism by a composite polynomial consisting of Bézier curves and cubic polynomials. *Mechanism and Machine Theory*, *132*, 248–263. https://doi.org/10.1016/j.mechmachtheory.2018.11.009

[22] Liu, L., Chen, C., Zhao, X., & Li, Y. (2016). Smooth trajectory planning for a parallel manipulator with joint friction and jerk constraints. *International Journal of Control, Automation and Systems*, *14*(4), 1022–1036. https://doi.org/10.1007/s12555-014-0495-4

[23] Alfatih, M. F., Riyadi, M. A., & Setiawan, I. (2021). Speed control system in mobile robot based on Bezier curve trajectory. *IOP Conference Series: Materials Science and Engineering*, *1108*(1), 012015. https://doi.org/10.1088/1757-899X/1108/1/012015

[24] Zhao, T., Zi, B., Qian, S., & Zhao, J. (2020). Algebraic method-based point-to-point trajectory planning of an under-constrained cable-suspended parallel robot with variable angle and height cable mast. *Chinese Journal of Mechanical Engineering*, *33*(1), 54. https://doi.org/10.1186/s10033-020-00473-z

[25] Boryga, M. (2014). Trajectory planning of an end-effector for path with loop. *Strojniški vestnik – Journal of Mechanical Engineering*, *60*(12), 804–814. https://doi.org/10.5545/sv-jme.2014.1965

[26] Fang, Y., Hu, J., Qi, J., Liu, W., Wang, W., & Peng, Y. (2019). Planning trigonometric frequency central pattern generator trajectory for cyclic tasks of robot manipulators. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *233*(11), 4014–4031. https://doi.org/10.1177/0954406218806010

[27] Muller-Karger, C. M., Leonell Granados Mirena, A., & Scarpati Lopez, J. T. (2000). Hyperbolic trajectories for pick-and-place operations to elude obstacles. *IEEE Transactions on Robotics and Automation*, *16*(3), 294–300. https://doi.org/10.1109/70.850647

[28] Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: A systematic review. *Archives of Computational Methods in Engineering*, *29*(5), 2531–2561. https://doi.org/10.1007/s11831-021-09694-4

[29] Charilogis, V., Tsoulos, I. G., & Tzallas, A. (2023). An improved parallel particle swarm optimization. *SN Computer Science*, *4*(6), 766. https://doi.org/10.1007/s42979-023-02227-9

[30] Kucuk, S. (2017). Optimal trajectory generation algorithm for serial and parallel manipulators. *Robotics and Computer-Integrated Manufacturing*, *48*, 219–232. https://doi.org/10.1016/j.rcim.2017.04.006

[31] Machmudah, A., Parman, S., Zainuddin, A., & Chacko, S. (2013). Polynomial joint angle arm robot motion planning in complex geometrical obstacles. *Applied Soft Computing*, *13*(2), 1099–1109. https://doi.org/10.1016/j.asoc.2012.09.025

[32] Xin, P., Rong, J., Yang, Y., Xiang, D., & Xiang, Y. (2017). Trajectory planning with residual vibration suppression for space manipulator based on particle swarm optimization

algorithm. *Advances in Mechanical Engineering*, *9*(4), 1687814017692694. https://doi.org/10.1177/1687814017692694

[33] Song, B., Wang, Z., & Zou, L. (2021). An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. *Applied Soft Computing*, *100*, 106960. https://doi.org/10.1016/j.asoc.2020.106960

[34] Chen, W., Wang, H., Liu, Z., & Jiang, K. (2023). Time-energy-jerk optimal trajectory planning for high-speed parallel manipulator based on quantum-behaved particle swarm optimization algorithm and quintic B-spline. *Engineering Applications of Artificial Intelligence*, *126*, 107223. https://doi.org/10.1016/j.engappai.2023.107223

[35] Wu, N., Jia, D., Li, Z., & He, Z. (2024). Trajectory planning of robotic arm based on particle swarm optimization algorithm. *Applied Sciences*, *14*(18), 8234. https://doi.org/10.3390/app14188234

[36] Du, Y., & Chen, Y. (2022). Time optimal trajectory planning algorithm for robotic manipulator based on locally chaotic particle swarm optimization. *Chinese Journal of Electronics*, *31*(5), 906–914. https://doi.org/10.1049/cje.2021.00.373

[37] Hu, X., Wu, H., Sun, Q., & Liu, J. (2023). Robot time optimal trajectory planning based on improved simplified particle swarm optimization algorithm. *IEEE Access*, *11*, 44496–44508. https://doi.org/10.1109/ACCESS.2023.3272835

[38] Lan, J., Xie, Y., Liu, G., & Cao, M. (2020). A multi-objective trajectory planning method for collaborative robot. *Electronics*, *9*(5), 859. https://doi.org/10.3390/electronics9050859

[39] Ekrem, Ö., & Aksoy, B. (2023). Trajectory planning for a 6-axis robotic arm with particle swarm optimization algorithm. *Engineering Applications of Artificial Intelligence*, *122*, 106099. https://doi.org/10.1016/j.engappai.2023.106099

[40] Li, H. (2016). A jerk-constrained asymmetric motion profile for high-speed motion stages to reduce residual vibration. *International Journal of Computer Applications in Technology*, *53*(2), 149–156. https://doi.org/10.1504/IJCAT.2016.074453

How to Cite: Han, C. J., Song, K. R., & Pak, S. G. (2026). New Trajectory Planning Based on Bezier Curve Using Particle Swarm Optimization. *Journal of Climbing and Walking Robots*. https://doi.org/10.47852/bonviewJCWR62026914