A PARABOLIC-TRANSITION-BASED TRAJECTORY OPTIMIZATION APPROACH FOR MULTI-JOINT COORDINATION OF A 4-DOF PICKING MANIPULATOR

Haixin ZOU[1], Xiangjun ZOU[2], Mingxiang GUAN[3]*

This paper introduces an original trajectory-optimization framework for a four-degree-of-freedom picking manipulator, featuring two key innovations. First, each joint trajectory employs parabolic blends with enhanced trapezoidal acceleration–*deceleration interpolation to achieve second-order continuity in position, velocity, and acceleration, thereby minimizing vibration and jerk. Second, multi-joint coordination leverages virtual via-points and unified time-base synchronization to ensure all joints reach critical positions simultaneously under velocity and acceleration limits, reducing end-effector irregularities and collision risk. MATLAB simulations demonstrate that, compared with the cubicpolytraj toolbox function, the proposed method suppresses acceleration spikes, maintains smooth kinematic profiles, and achieves real-time, high-precision trajectory planning.*

Keywords: trajectory planning; parabolic transition; trapezoidal interpolation; multi-joint coordination; picking manipulator.

1. Introduction

With advances in intelligent agriculture and automated manufacturing, manipulators play a key role in complex tasks like picking and sorting [1,2]. Trajectory planning, a core of motion control, directly impacts end-effector accuracy, vibration suppression, and system safety [3]. Traditional trapezoidal acceleration/deceleration interpolation—valued for its simplicity and efficiency—is widely used in industrial robots but features acceleration jumps at phase transitions, causing vibrations and control oscillations that undermine precision and flexibility [4,5].

In single-joint interpolation, the Linear Segment with Parabolic Blend (LSPB) removes velocity discontinuities by adding parabolic segments, but its "time symmetry" assumption causes waypoint deviations in multi-segment

[1] Lecturer, School of Information and Communication, Shenzhen University of Information Technology, China, e-mail: zouhaixin123@qq.com

[2] Prof., College of Engineering, South China Agricultural University, China, e-mail: xjzou1@163.com

[3]* Prof., School of Information and Communication, Shenzhen University of Information Technology, China, e-mail: guanmx@sziit.edu.cn (corresponding author)

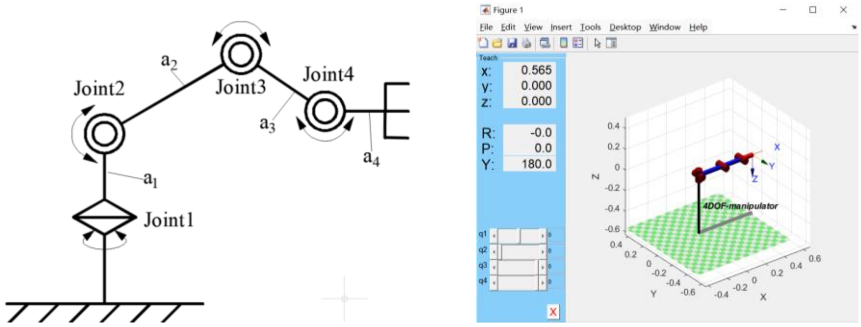
trajectories. To address this, Al-khayyt et al. [6] used Particle Swarm Optimization to globally tune LSPB timing coefficients for precise waypoint passage; Shrivastava et al. [7] fused linear polynomials with parabolic blends and applied weighted smoothing under redundancy optimization to improve transition continuity and tracking accuracy; Quiñonez et al. [8] developed a homotopic-iteration framework with via-point constraints to greatly reduce oscillations and ensure velocity and acceleration continuity; Raji et al. [9] inserted “virtual” intermediate nodes via a septic polynomial and rigorously solved boundary conditions to produce smooth, higher-order continuous trajectories that suppress vibrations. Building on these, Qian et al. [10] applied quintic B-splines for smooth dynamic planning of a 3-DOF parallel mechanism, suppressing vibration sources; Tao et al. [11] proposed an optimized B-spline path planner to further enhance control continuity and stability in complex robots; Song et al. [12] and Zheng et al [13] employed RBF neural networks to enhance tracking performance and suppress vibration for flexible manipulators, demonstrating the effectiveness of learning-based approaches for dynamic compensation; Chembuly and Voruganti [14] combined artificial potential fields with nonlinear optimization to generate smooth, obstacle-avoiding trajectories for redundant manipulators while minimizing acceleration and vibration. Meanwhile, although cubic and quintic polynomials ensure position and velocity continuity, acceleration and higher derivatives remain discontinuous. Senwal et al. [15] compared interpolation methods on a 7-DOF arm and found quintic polynomials smoother in acceleration than cubic, but with much higher computational and tuning demands; Chettibi et al. [16] reached similar conclusions.

Time normalization and virtual waypoints ensure all joints pass key points smoothly under a unified time base, reducing irregular end-effector motion and collision risk. Sidiropoulos et al. [17] introduced dynamic movement primitives with spatiotemporal normalization and waypoint insertion for smooth, coordinated joint motion. Su et al. [18] examined how coupled dynamics affect synchronization via trajectory coordination and simulation errors, highlighting the role of dynamic coupling. In dynamic and cluttered environments, simultaneous obstacle prediction and real-time adaptive trajectory replanning have also been investigated to ensure safe manipulator operation under environmental uncertainty [19]. Obe et al. [20] used genetic algorithms for multi-objective joint-trajectory optimization, improving task-space accuracy, shortening motion time, cutting energy use, and suppressing acceleration peaks.

From the above research progress, it can be seen that although cubic polynomial interpolation (such as MATLAB's cubicpolytraj) performs well in continuity of position and velocity, discontinuities in acceleration and higher-order derivatives often lead to significant end-effector vibration. Higher-order polynomials and B-splines can further improve smoothness but entail greater computational complexity and parameter tuning difficulties, which are unfavorable for real-time online planning. To address this trade-off, this paper focuses on a 4-DOF pick-and-place manipulator and proposes an improved trapezoidal interpolation method with parabolic transition segments. By introducing parabolic transitions at the single-joint level, the method achieves smooth velocity and acceleration profiles, while virtual via-points and unified time-synchronization ensure coordinated, continuous multi-joint motion. Simulation results demonstrate that the approach significantly reduces trajectory vibrations, maintains low computational cost, and offers strong engineering applicability, providing a practical solution for real-time, high-precision trajectory planning in flexible robotic systems.

2. A 4-DOF Picking Manipulator and MATLAB Model

As shown in Fig.1, this study takes a 4-DOF serial manipulator for agricultural picking as the research object and establishes a joint-space trajectory planning model in MATLAB based on its D-H parameters and joint motion constraints as shown in table 1.



X: 0.565
Y: 0.000
Z: 0.000

R: -0.0
P: 0.0
Y: 180.0

(a). 4-DOF picking manipulator (b) 3D Model and parameters in MATLAB

Fig. 1. The 4-DOF serial picking manipulator

Table 1

D-H Parameters of the 4-DOF Manipulator

Joint (i)	θ_i (variable)	d_i (m)	a_i (m)	α_i (rad)	Joint Limit (°)
1	θ_1	0	0	$+\pi/2$	[-90, +90]
2	θ_2	0	0.25	0	[0, +135]
3	θ_3	0	0.2	0	[-135, 0]
4	θ_4	0	0.115	$+\pi/2$	[-135, +30]

Note:θ_i: Joint variable (angle for revolute joints)
d_i: Offset along previous z to the common normal (all 0 here)
a_i: Length of the common normal (link length)
α_i: Angle about the common normal, from the previous z-axis to the current z-axis
Joint Limit (°): Allowed range for each joint angle (in degrees)

3. Local path point trajectory planning method based on parabolic transition and improved trapezoidal interpolation

The trapezoidal interpolation algorithm is widely used in motion control because it is simple to implement and effective in generating smooth velocity profiles. Building on this algorithm, this paper develops local point-to-point (PTP) and continuous-path trajectory planning schemes for joint-space waypoints (also referred to as via-points) by combining trapezoidal acceleration − deceleration profiles with parabolic transition segments.

3.1 Trapezoidal interpolation algorithm

When a single joint of the manipulator performs a point-to-point (PTP) motion between two positions, its velocity follows a trapezoidal profile consisting of three phases: acceleration, constant velocity, and deceleration. As illustrated in Fig. 2, the motion can be divided, from the velocity perspective, into an acceleration phase from T0 to T5 (with duration T05), a constant-velocity phase from T5 to T6 (duration T56), and a deceleration phase from T6 to T11, separated by the time instants T5 and T6. From the acceleration perspective, the motion is further subdivided into three sub-phases: a jerk-increase sub-phase, a constant-acceleration sub-phase, and a jerk-decrease sub-phase, delimited by T2 and T3 for the acceleration part and symmetrically by T8 and T9 for the deceleration part.

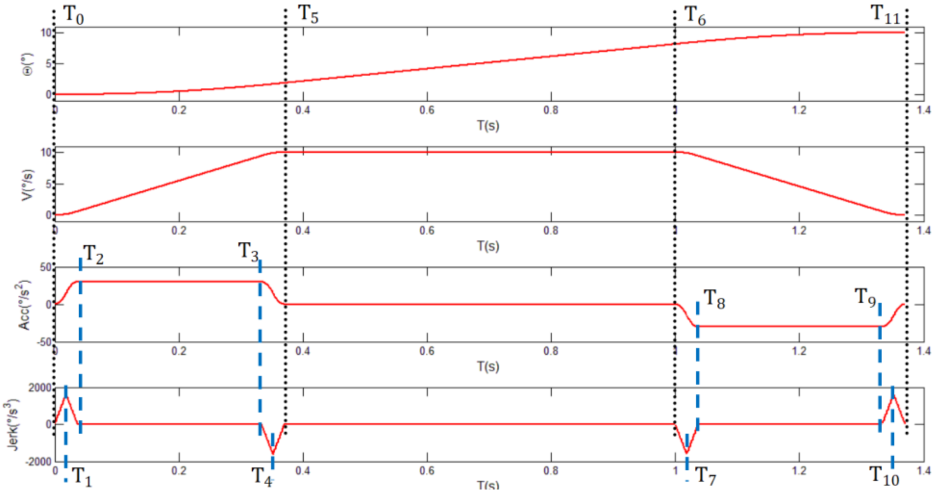


Fig. 2 PTP algorithm based on parabolic transition and trapezoidal acceleration-deceleration interpolation

(1)Jerk profile. Examining the jerk curve within the interval T_0 and T_5, acceleration interpolation based on parabolic transitions yields formula (1): where J is the jerk, k is a constant, t is time, and T_0...T_5 mark the phases of ramp-up, hold, and ramp-down sub-phases. This piecewise design smooths the changes in acceleration at the start and end of motion, reducing mechanical shock.

$$J = \begin{cases} kt & [T_0, T_1] \\ -kt & [T_1, T_2] \\ 0 & [T_2, T_3] \\ -kt & [T_3, T_4] \\ kt & [T_4, T_5] \end{cases} \tag{1}$$

(2)Time proportions. To trade off smoothness and speed, the accel/decel period uses a 1:1:16:1:1 ratio. Each “1” is a smoothing phase at the beginning or end, and “16” is full-acceleration time. For very short moves, fixed ratios may no longer be feasible, in which case the sub-phase durations must be adjusted or the profile degenerates into a triangular one without a constant-velocity segment. The calculation formulas for each time segment within the interval T0 to T5 are shown in the Table 2. Considering motion symmetry, the similar calculation formulas are used for each time segment within the interval T6 to T11.

This algorithm merges trapezoidal velocity profiles with parabolic transitions to produce smooth, continuous displacement, velocity, and acceleration curves. It stabilizes joint motion with minimal impact while fully leveraging the system’ s acceleration and velocity limits, enabling fast, efficient operation with low computational cost. For very small displacements or short durations, adjustments are needed—for instance, switching to a symmetric two-segment (triangular) acceleration－deceleration profile when no constant-velocity phase is possible.

Table 2

The calculation formulas for each time segment within the interval

	t_1=t; T_0<t<T_1	t_2=t-T_1; T_1<t<T_2	t_3=t-T_2; T_2<t<T_3	t_4=t-T_3; T_3<t<T_4	t_5=t-T_4; T_4<t<T_5
Initial	$\mathrm{J}_{01} = 0$ $a_{01} = 0$ $\mathrm{V}_{01} = 0$ $\mathrm{S}_{01} = 0$	$\mathrm{J}_{02} = \mathrm{J}_1$ $a_{02} = a_1$ $\mathrm{V}_{02} = V_1$ $\mathrm{S}_{02} = \mathrm{S}_1$	$\mathrm{J}_{03} = 0$ $a_{03} = a_2$ $\mathrm{V}_{03} = V_2$ $\mathrm{S}_{03} = \mathrm{S}_2$	$\mathrm{J}_{04} = \mathrm{J}_3 = 0$ $a_{04} = a_3$ $\mathrm{V}_{04} = V_3$ $\mathrm{S}_{04} = \mathrm{S}_3$	$\mathrm{J}_{05} = \mathrm{J}_4$ $a_{05} = a_4$ $\mathrm{V}_{05} = V_4$ $\mathrm{S}_{05} = \mathrm{S}_4$
Running time	$\mathrm{J}_1 = kt_1$ $a_{01} = \frac{1}{2}kt_1^2$ $\mathrm{V}_{01} = \frac{1}{6}kt_1^3$ $\mathrm{S}_{01} = \frac{1}{24}kt_1^4$	$\mathrm{J}_2 = -kt_2 + \mathrm{J}_{02}$ $a_2 = \frac{-1}{2}kt_2^2 + \mathrm{J}_{02}t_2 + a_{02}$ $\mathrm{V}_2 = \frac{-1}{6}kt_2^3 + \frac{1}{2}\mathrm{J}_{02}t_2^2 + \ldots$ $a_{02}t_2 + \mathrm{V}_{02}$ $\mathrm{S}_2 = \frac{-1}{24}kt_2^4 + \frac{1}{6}\mathrm{J}_{02}t_2^3 + \ldots$ $\frac{1}{2}a_{02}t_2^2 + \mathrm{V}_{02}t_2 + \mathrm{S}_{02}$	$\mathrm{J}_3 = 0$ $a_3 = a_{03}$ $\mathrm{V}_3 = a_3t_3 + \mathrm{V}_{03}$ $\mathrm{S}_3 = \frac{1}{2}a_{03}t_3^2 + \ldots$ $\mathrm{V}_{03}t_3 + \mathrm{S}_{03}$	$\mathrm{J}_4 = -kt_4$ $a_4 = \frac{-1}{2}kt_4^2 + a_{04}$ $\mathrm{V}_4 = \frac{-1}{6}kt_4^3 + a_{04}t_4 + \ldots$ V_{04} $\mathrm{S}_4 = \frac{-1}{24}kt_4^4 + \frac{1}{2}a_{04}t_4^2 + \ldots$ $\mathrm{V}_{04}t_4 + \mathrm{S}_{04}$	$\mathrm{J}_5 = -kt_5 + \mathrm{J}_{05}$ $a_5 = \frac{1}{2}kt_5^2 + \mathrm{J}_{05}t_5 + a_{05}$ $\mathrm{V}_5 = \frac{1}{6}kt_5^3 + \frac{1}{2}\mathrm{J}_{05}t_5^2 + \ldots$ $a_{05}t_5 + \mathrm{V}_{05}$ $\mathrm{S}_5 = \frac{1}{24}kt_5^4 + \frac{1}{6}\mathrm{J}_{05}t_5^3 + \ldots$ $\frac{1}{2}a_{05}t_5^2 + \mathrm{V}_{05}t_5 + \mathrm{S}_{05}$

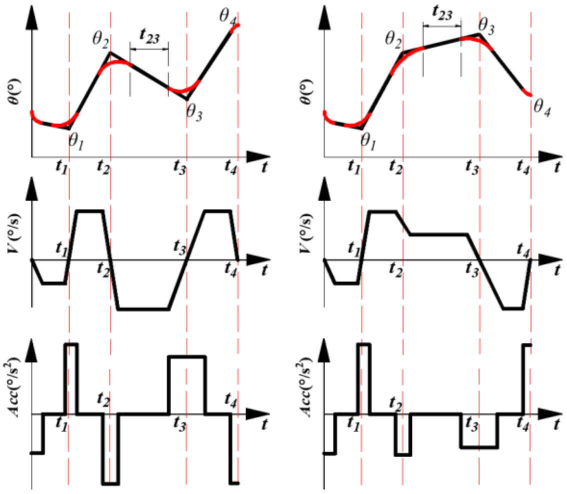
3.2 Via-point handling with trapezoidal interpolation

Consider a joint trajectory defined by 4 path points (θ_1, θ_2, θ_3, θ_4). Between each pair of consecutive waypoints, a trapezoidal velocity profile with parabolic blending is applied, so that the trajectory parameters can be determined from the waypoint positions, constant-velocity durations, and acceleration limits.

The sign of consecutive segments is critical: if two adjacent segments have opposite signs (indicating reversal of motion), the joint must decelerate to zero at the intermediate waypoint before changing direction. Conversely, if the signs remain the same, the joint can traverse the via-point with a non-zero velocity, as illustrated in Fig. 3.

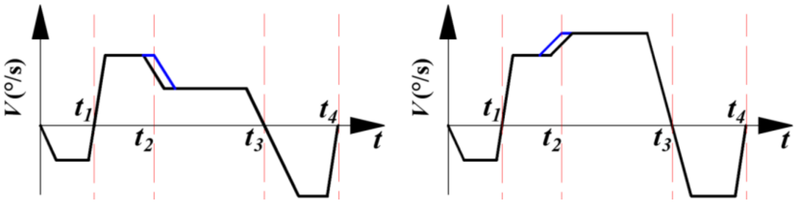
For example, when both segments θ_2 and θ_3 are the positive (see Fig. 3(b)), an improved rule is adopted as illustrated in Fig.4.: If the following segment has a lower target speed, the current velocity is maintained through θ_2 before decelerating. If the following segment has a higher target speed, the joint accelerates in advance, increasing the velocity before traversing θ_2.

Under identical motion constraints, this improved via-point handling increases the effective travel distance at higher speeds, thus shortening the overall motion time and improving operational efficiency.



(a). $(\theta_1, \theta_2, \theta_3, \theta_4)$ with (-,+,-,+) signs (b). $(\theta_1, \theta_2, \theta_3, \theta_4)$ with (-,+,+,-) signs

Fig. 3. Joint trajectories through four waypoints with different sign patterns



(a). keep the speed before slowing down (b). raise the speed beforehand

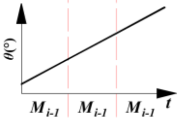
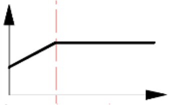
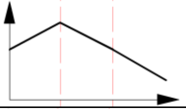
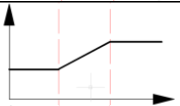
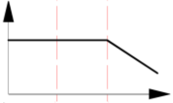
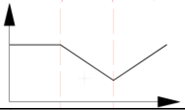
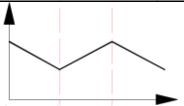
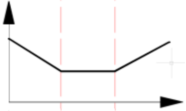
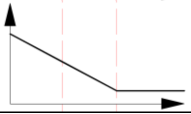
Fig. 4. Improved trapezoidal interpolation for via-point handling

3.3 Three-segment local planning using adjacent segments

This subsection proposes a three-segment local planning scheme that determines the trajectory of the current segment M_i based on its adjacent segments $M_{i–1}$ and M_{i+1}. For each non-virtual segment, the sign pattern of the three slopes (positive, zero, or negative) is mapped to one of the representative cases in Table 3, and the corresponding parabolic-transition trapezoidal profile is applied to compute the velocity and timing for that segment.

Table 3

Discussion Table of Local Planning for Three Path Segments

Slope of M_{i-1}	Slope of Current M_i	Slope of M_{i+1}	Graph (horizontal: time, vertical: angular displacement)
Positive	Positive	Positive	
Positive	Zero	Zero	
Positive	Negative	Negative	
Zero	Positive	Zero	
Zero	Zero	Negative	
Zero	Negative	Positive	
Negative	Positive	Negative	
Negative	Zero	Positive	
Negative	Negative	Zero	

The local planning procedure operates on a sequence of N waypoints by first adding two virtual points at the beginning and end, so that each trajectory segment can be analyzed together with its immediate neighbors. Each non-virtual segment M_i is considered within a local triple (M_{i-1}, M_i, M_{i+1}), and the sign pattern of the three segment slopes (positive, zero, or negative) is mapped to one of the representative cases summarized in Table 3. Once the corresponding case is identified, the improved trapezoidal interpolation with parabolic transitions from previous subsection is applied to determine the velocity profile and timing for that segment. This procedure is repeated sequentially over all segments, and other sign combinations can be handled analogously by referring to the same set of representative cases.

As a simple example, consider a joint moving from 0° to 5° and then to 10°. The maximum velocity is set to 10°/s for the first segment (0° to 5°) and 5°/s for the following segment (5° to 10°), with a maximum acceleration of 20°/s², and a ratio of 0.8 as defined in Section 3.1. By adding virtual waypoints at the beginning and end, the local planning scheme treats the two segments within the triplets (0,0,5,10) and (0,5,10,10), and applies the parabolic-transition trapezoidal interpolation to determine the corresponding velocity profiles. The resulting trajectory, illustrated in Fig. 5, maintains a velocity of 10°/s when passing the intermediate waypoint and then smoothly decelerates to 5°/s for the following segment, before finally coming to rest at the end point.

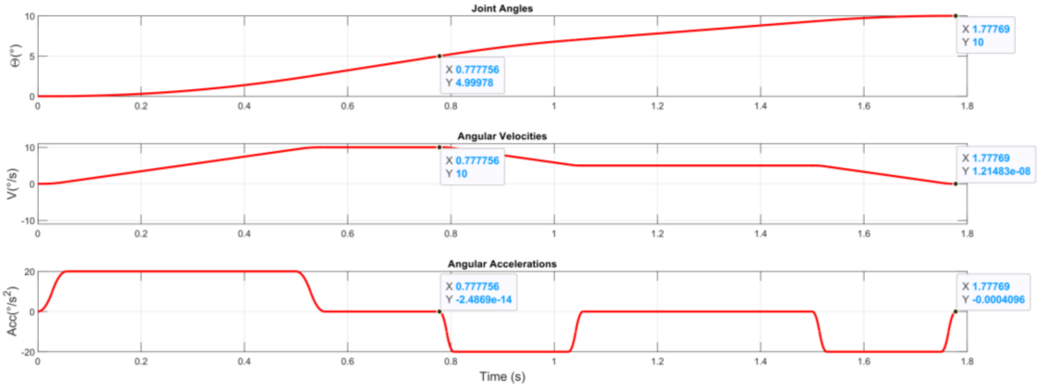


Fig. 5. The local planning for continuous three path segments (M0 to M2, and M1 to M3)

4. MATLAB-based experiments and result analysis

4.1 Path planning for the 4-DOF manipulator

For a given task, the path planner provides a discrete sequence of joint-space waypoints for each of the four joints. These waypoints are obtained in configuration space using a graph-search method such as A*, with a fixed angular resolution between adjacent points. For example, the joint path points for a given joint may be (0, 5, 10, 10, 5, 0, -5, -10, -15). The interval of 5° per path point indicates that the C-space was discretized in steps of 5° during modeling. The single-joint trajectory planning method described in the previous section is then applied segment by segment along each waypoint sequence.

To enable coordinated motion, multi-joint synchronization is performed on these preliminary trajectories. For each joint, the traversal time of every segment is first computed under the specified velocity and acceleration limits. For each waypoint index, the maximum segment time among all joints is selected as the reference duration, and the trajectories of the remaining joints are time-scaled accordingly so that all joints pass the corresponding waypoints simultaneously. Using the synchronized segment durations and the improved trapezoidal interpolation with parabolic transitions, the final joint trajectories are generated, and the resulting displacement, velocity, and acceleration profiles are obtained for subsequent analysis and control implementation.

4.2 Trajectory planning simulation

Trajectory planning simulations were conducted in MATLAB to compare the proposed method with the built-in cubic polytraj function, which represents a typical cubic polynomial trajectory planner. The same 4-DOF picking manipulator model described in Section 2 were used in all simulations.

Let the path points for the four joints be given as in formula (2). All joints are assumed to start from the initial position 0. Negative values indicate clockwise joint rotation, while positive values represent counterclockwise rotation; for example, -5° means the joint rotates 5 degrees clockwise.

The maximum joint velocity and acceleration were limited to ±10°/s and ±20°/s, respectively, and the ratio for the constant-acceleration phase was set to 0.8.

$$\begin{bmatrix} 0 & 0 & -5 & 0 & -5 & 0 & 5 & 10 & 15 \\ 0 & 5 & 10 & 10 & 5 & 0 & -5 & -10 & -15 \\ 0 & -5 & -10 & -15 & -10 & -5 & 0 & -5 & 0 \\ 0 & -5 & -10 & -5 & -5 & 0 & 5 & 0 & 5 \end{bmatrix} \tag{2}$$

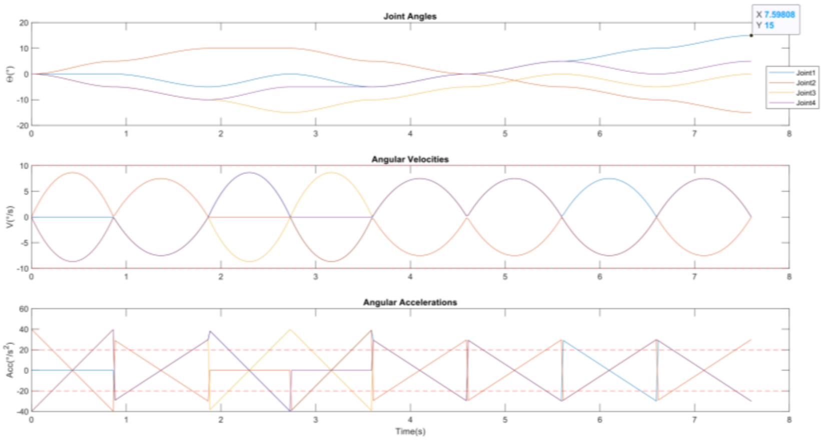


Fig. 6. Joint motion profiles of the MATLAB cubicpolytraj under ±10°/s limits

Based on these path points, the cubicpolytraj function is employed to generate and plot the displacement, velocity, and acceleration curves for each joint, as depicted in Fig. 6. When the maximum velocity constraint is limited to ±10°/s, the manipulator completes the motion in approximately 7.60s. However, the maximum acceleration reaches about ±40°/s², which significantly exceeds the desired acceleration constraint of ±20°/s². To evaluate the impact of acceleration limitation, Fig. 7 shows the results generated by the cubicpolytraj function when both velocity and acceleration are constrained to ±10°/s and ±20°/s², respectively. In this case, the velocity curves are notably constrained, resulting in a considerably longer motion duration of approximately 10.75s.

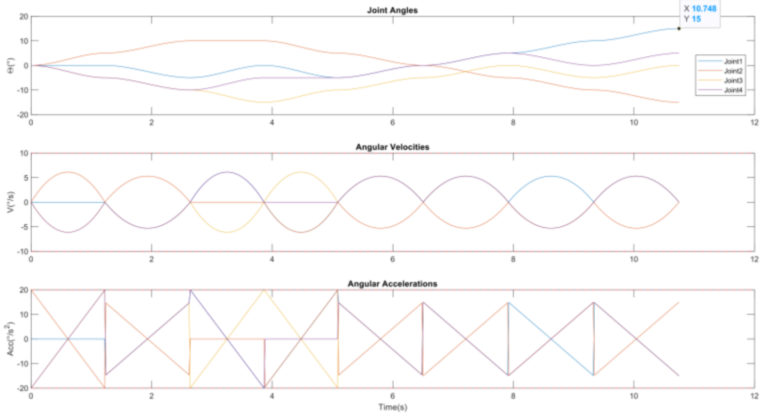


Fig. 7. Joint motion profiles of the MATLAB cubicpolytraj under ±10°/s and ±20°/s² limits

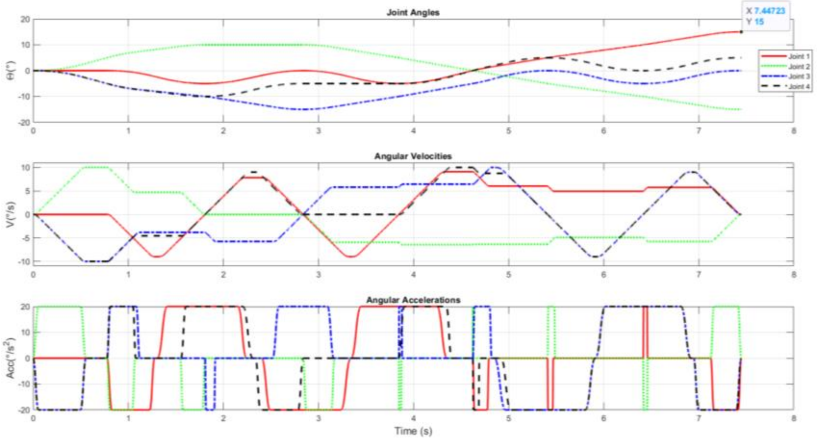


Fig. 8. Joint motion profiles of the proposed method

For comparison, the proposed algorithm in this paper was applied under the same motion constraints—maximum velocity limited to ±10°/s and maximum acceleration limited to ±20°/s²—to generate the corresponding joint trajectories, as shown in Fig. 8. It can be clearly observed that the proposed method fully exploits the allowable velocity and acceleration limits while maintaining a continuously differentiable acceleration profile. The manipulator completes the entire motion within approximately 7.45s, achieving a 0.15s (about 1.9%) time reduction relative to the cubicpolytraj function when only the velocity constraint is applied, and a 3.3s (about 30.7%) reduction compared with the case where both velocity and acceleration constraints are simultaneously imposed.

4.3 Result analysis

Fig. 9 presents a detailed comparison for Joint 1 between the proposed method and MATLAB’s cubicpolytraj under ±10°/s limits. The proposed scheme, which combines parabolic transitions with an improved trapezoidal acceleration – deceleration interpolation, achieves smoother single – joint motion, better multi – joint synchronization, and enhanced overall tracking performance. In particular, the velocity profile of the proposed method closely follows an ideal trapezoidal shape and fully utilizes the prescribed velocity and acceleration limits, whereas the cubicpolytraj profile exhibits oscillatory variations and larger acceleration spikes.

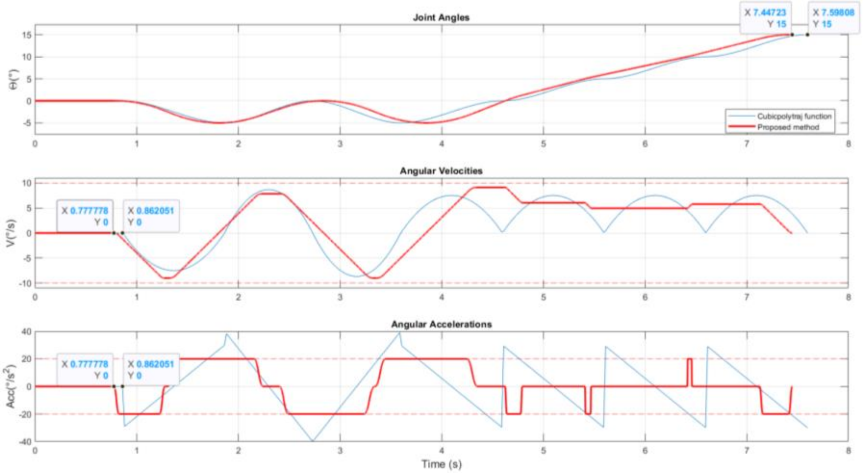


Fig. 9. Joint 1 Trajectory: cubicpolytraj under ±10°/s limits vs. Proposed Method

From the acceleration curves it can be observed that the trajectory generated by the proposed algorithm respects the ±20°/s² bounds almost everywhere and forms well–structured acceleration and deceleration phases, effectively avoiding the abrupt changes and discontinuities produced by cubicpolytraj. When a full trapezoidal acceleration–deceleration profile is feasible, around 80% of the motion time is spent at the maximum velocity of 10°/s, with the remaining 20% allocated to accelerating and decelerating. When the distance is too short to sustain a constant–velocity segment, approximately 80% of the motion occurs at the maximum acceleration of 20°/s², and only about 20% of the time is used for ramping the acceleration up or down. For very short segments, the algorithm automatically merges them with their neighbors so that acceleration continuity is preserved under the given constraints. Overall, the proposed planner shortens the total motion time by about 0.15s, corresponding to an improvement of roughly 1.9% compared with cubicpolytraj under ±10°/s limits. As the number of waypoints increases, the advantages of the proposed algorithm become even more pronounced.

At the algorithmic level, the parabolic transition guarantees second–order continuity of position, velocity, and acceleration while keeping the computational burden low. The local optimization strategy, which jointly adjusts each segment and its adjacent ones, yields smooth and adaptive velocity transitions at via points, and can handle both gentle and abrupt changes in speed within the allowed limits. Simulation results demonstrate that the proposed method adapts well to different motion demands, effectively suppresses vibration and control jitter, and thereby improves path continuity and manipulator stability relative to cubicpolytraj.

A key advantage of the proposed method is its local path-planning scheme, which evaluates each trajectory segment together with its two neighboring segments in a three-segment context. This design enables smooth and consistent velocity adjustments at via-points, effectively handling both gradual and abrupt speed changes between consecutive segments. In addition, the method adaptively optimizes the duration of acceleration and deceleration buffers while strictly enforcing the prescribed velocity and acceleration limits.

5. Conclusion

This paper presents an efficient trajectory optimization method for a 4-DOF picking manipulator that integrates parabolic transitions with an improved trapezoidal acceleration–deceleration interpolation scheme. Compared with the conventional cubic polynomial approach, the proposed method effectively eliminates acceleration discontinuities and reduces the total motion time by at least 1.9%. These improvements contribute to smoother dynamic responses, reduced mechanical stress, and increased operational stability, validating the method’s potential for high-precision robot motion control.

In addition to its computational efficiency, the proposed strategy demonstrates scalability for real-time implementation and strong compatibility with diverse motion constraints in industrial manipulators. Theoretical analysis confirms that integrating parabolic transitions into the trapezoidal interpolation framework ensures continuous second-order derivatives under both complete and truncated motion profiles, offering a generalized foundation for smooth trajectory generation under bounded kinematic conditions.

Future work will focus on several directions. First, the algorithm will be extended to 6-DOF manipulators to handle more complex kinematic couplings and spatial trajectories. Second, comprehensive robustness testing under varying payloads, speeds, and external disturbances will be conducted to characterize performance boundaries. Third, experimental validation on a physical robotic platform will be carried out to substantiate the numerical findings and evaluate practical feasibility. Finally, the integration of local and global optimization strategies will be explored to further enhance end-effector smoothness, energy efficiency, and motion naturalness in dynamically changing environments.

Acknowledgments

This work was supported by the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2022B1515120002), the Project of Shenzhen Science and Technology Innovation Committee (Grant No. KJZD20240903103300002), the Project of Shenzhen Science and Technology Innovation Committee (Grant No. KCXFZ20240903094011015), and other Project (Grant No. PT2024E007; 2024HT009, KJ2025C005).

REFERENCES

[1]. *R. R. Shamshiri*, *C. Weltzien*, *I. A. Hameed, et al.* "Research and development in agricultural robotics: A perspective of digital farming", in International Journal of Agricultural and Biological Engineering, **vol. 11**, no. 4, 2018, pp. 1–14.

[2]. *T. Jin* and *X. Han*, "Robotic arms in precision agriculture: A comprehensive review of the technologies, applications, challenges, and future prospects", in Computers and Electronics in Agriculture, **vol. 221**, 2024.

[3]. *X. Li*, *H. Zhao*, *X. He*, *et al*. "A novel Cartesian trajectory planning method by using triple NURBS curves for industrial robots", in Robotics and Computer-Integrated Manufacturing, **vol. 83**, 2023.

[4]. *H. B. Lai*, *M. X. Wang*, *S. T. Liu*, *et al.* "Trajectory planning of 4-RR-(SS)2 parallel robot based on 345-corrected trapezoidal motion law", in Transactions of the Chinese Society for Agricultural Machinery, **vol. 55**, no. 4, 2024, pp. 411-420.

[5]. *C. Mirz*, *B. Corves*, *Y. Takeda*, *et al*. "Trajectory planning and active dynamic balancing for highly dynamic handling tasks, a comparative study", in Mechanism and Machine Theory, **vol. 214**, 2025.

[6]. *S. Z. S. Al-khayyt*, "Creating through points in linear function with parabolic blends path by optimization method", in Al-Khwarizmi Engineering Journal, **vol. 14**, no. 1, 2018, pp. 77-89.

[7]. *A. Shrivastava and V. K. Dalla*. "Multi-segment trajectory tracking of the redundant space robot for smooth motion planning based on interpolation of linear polynomials with parabolic blend", in Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, **vol. 236**, no. 16, 2022, pp. 9255–9269.

[8]. *Y. Quiñonez*, *O. Zatarain*, *C. Lizarraga*, *et al.* "Numerical method using homotopic iterative functions based on the via point for the joint-space trajectory generation", in Applied Sciences, **vol. 13**, no. 2, 2023.

[9]. *A. A. Raji, O. S. Asaolu and T. T. Akano*. "Joint space robot arm trajectory planning using septic function", in ABUAD Journal of Engineering Research and Development, **vol. 5**, 2022, pp. 110–123.

[10]. *S. Qian, K. Bao, B. Zi, et al*. "Dynamic trajectory planning for a three degrees-of-freedom cable-driven parallel robot using quintic B-splines", in Journal of Mechanical Design, **vol. 142**, no. 7: 073301, 2020.

[11]. *C. Tao, S. Cheng, F. Wang, et al.* "An optimization-based planner with B-spline parameterized input for continuous-time control of robots", in 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2024, pp. 3100-3107.

[12]. *Q. Song, S. Li, Q. Bai, et al*. "Trajectory planning of robot manipulator based on RBF neural network", in Entropy, **vol. 23**, no. 9, 2021.

[13]. Q. C. Zheng, Z. Y. Wei, P. H. Zhu, et al. "Adaptive vibration suppression of flexible manipulator based on improved RBF neural network", in U.P.B. Sci. Bull., Series D, **vol. 86**, no. 2, 2024, pp. 3-18.

[14]. *V. S. Chembuly and H. K. Voruganti*, "Trajectory planning of redundant manipulators moving along constrained path and avoiding obstacles". in Procedia computer science, **vol. 133**, 2018, pp. 627-634.

[15]. *V. B. Senwal, G. Ajith, S. Vardhineni, et al.* "Comparative analysis of interpolation methods for 7-DoF manipulator robot trajectory planning", in 2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), IEEE, **vol. 3**, 2025, pp. 1-7.

[16]. *T. Chettibi*, "Smooth point-to-point trajectory planning for robot manipulators by using radial basis functions", in Robotica, **vol. 37**, no. 3, 2019, pp. 539-559.

[17]. *A. Sidiropoulos* and *Z. Doulgeri*, "Dynamic via-points and improved spatial generalization for online trajectory planning with dynamic movement primitives", in Journal of Intelligent & Robotic Systems, **vol. 110**, no. 1, 2024.

[18]. *C. Su, S. Zhang, S. Lou, et al.* "Trajectory coordination for a cooperative multi-manipulator system and dynamic simulation error analysis", in Robotics and Autonomous Systems, **vol. 131**, 2020.

[19]. W. Liu, T. Zhou, C. Jing, et al. "Simultaneous obstacle prediction and real-time adaptive trajectory planning for robotic manipulators in dynamic environments", in U.P.B. Sci. Bull., Series D, **vol. 84**, no. 3, 2022, pp. 53–68.

[20]. *O. O. Obe, E. A. Ajayi and O. O. Odewale*, "Genetic algorithm based optimal trajectories planning for robot manipulators on assigned paths", in International Journal of Emerging Trends in Engineering Research, **vol. 8**, no. 8, 2020, pp. 4888–4893..