Proceedings of the 43rd Chinese Control Conference
July 28-31, 2024, Kunming, China

Smooth and Time-optimal Improved-Bezier Trajectory Planning Based on Dynamics for Industrial Robots

Wei Cai, Yahui Gan, Fang Fang, Bo Zhou, Xianzhong Dai
1.School of Automation, Southeast University, Nanjing Jiangsu 210018
E-mail: 220221915@seu.edu.cn
2. Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing 210096, China
E-mail: ganyahui@yeah.net

Abstract: This paper presents a smooth and time-optimal improved-Bezier trajectory planning algorithm that addresses the avoidance of solving switching points and solves the problem of discontinuous torque on the basis of the time-optimal path parameterization (TOPP) algorithm. The proposed approach employs the TOPP-RA algorithm with a friction model, to construct the optimal velocity trajectory under the constraints of the geometric paths and joint torques. Moreover, the proposed algorithm utilizes an enhanced Bezier curve instead of the jerk constraint in the trajectory optimization model to achieve a smooth trajectory. The modified Bezier curve optimizes the peaks of the pseudo-velocity curve in the phase plane to change the discontinuity of joint torque. The proposed algorithm ensures absolute time optimality and achieve a smooth trajectory. The results of the MATLAB simulation experiment on a two-link motion system indicate the validity of the proposed algorithm.

Key Words: time-optimal , smooth trajectory planning, improved-Bezier trajectory

1 Introduction

Trajectory planning is a crucial focal point in the motion control of industrial robots. Efficiency is an important consideration for industrial manufacturers, and in robotics research, "time-optimal" is a focus of trajectory planning because it is directly related to the motion efficiency of robots. Time-Optimal Trajectory Planning (TOTP) stands out as a significant avenue for enhancing the productivity of robotic systems. TOTP aims to find the minimum-time to finish a predefined geometric path within the workspace of a robot system while complying with physical constraints.

Reference 1 proposes the TOTP algorithm aimed at seeking the shortest time to accomplish trajectories. The algorithm parametrizes the path to reduce optimization variables and consequently decrease the complexity of the optimization problem. It represents the robot's dynamical model as a function of trajectory parameters and transfoms the constraints of joint torques into constraints on these trajectory parameters. The "path-velocity" decomposition framework, first proposed in Reference 3, was initially applied to trajectory planning problems. Most current methods build upon and expand this framework. Over recent decades, many researchers have contributed to this issue, broadly falling into three categories: (1) dynamic programming methods [3-4], (2) convex optimization methods [5-7], and (3) numerical integration methods [8-12].

The dynamic programming approach seeks the optimal trajectory based on the Bellman principle, enabling the simultaneous consideration of multiple constraint conditions and objective functions. The convex optimization method transforms the time-optimal problem into a convex optimization problem, followed by the utilization of convex optimization software packages to improve the efficiency of the solution process. Anchored in Pontryagin's Maximum Principle, numerical integration method addresses problems structured in a Bang-Bang format under torque input constraints. Reference 1 proposes a path parameterization approach, transforming the resolution of the three dimensional spatial problem into a one-dimensional space . Within the constraints of joint torque, it pursues the maximum attainable velocity, identifying pseudo-acceleration switching points along the velocity boundary curve within the feasible force domain. By integrating along the predefined geometric path with either maximum or minimum pseudo-acceleration, it demonstrates that at every moment along the optimal speed curve, at least one joint torque reaches its maximum value. However, the NI method for trajectory optimization suffers from certain limitations: (1) challenges in resolving switching points, entailing computational time in numerical searches, and (2) abrupt pseudo-acceleration fluctuations resulting in discontinuities in joint torque, thereby hindering the execution of practical commands.

Reference 8 introduces the Admissible Velocity Propagation algorithm (AVP): within the velocity interval at the initial position and grounded in the constraints of motion dynamics, it computes the admissible range for end-effector path velocities. Reference 9 presents the TOPP-RA algorithm, utilizing the AVP algorithm to determine the controllable velocity set and employing a forward-solving approach to ascertain the maximum achievable velocity set. Theoretically, it adheres to the optimal velocity curve. The TOPP algorithm reduces the complexity associated with solving switching points, reducing computational overhead. Nevertheless, the issue of discontinuity in joint torque persists.

The velocity trajectory resulting from the optimal time trajectory planning based on joint torque constraints shows peaks, meaning it is discontinuous in terms of acceleration and torque at certain points. Achieving continuity and smoothness in either acceleration or torque has been a central focus. References 7 and 10 conder the realm of pseudo-Jerk constraints on the trajectory. Ma, J-w et al. transform these pseudo-Jerk constraints into a pre-limiting envelope characterized by linear pseudo-acceleration constraints. Concurrently, they enforce the constraint of zero acceleration while strategically designating the extremal points of maximum feasible velocity as pivotal transition

*This work was supported by the National Natural Science Foundation of China under Grant No. 61873308, 61503076. Natural Science Foundation of Jiangsu Province under Grant No. BK20150624.

points. Solving the position of the new conversion point is then initiated, constrained by both pseudo-Jerk constraints and the pre-envelope of pseudo acceleration, the positions of transition points between adjacent intervals are systematically re-evaluated. This approach effectively suppresses pseudo-acceleration discontinuities. However, it is pertinent to note that this method involves an intricate sequence of preprocessing steps and lacks a comprehensive theoretical underpinning.

Reference 13 tackles the consideration of Jerk constraints within the domain of joint angular momentum. Hung Pham et al. consider the Time-Optimal Path Parameterization (TOPP) method to solve the third-order inequality constrained problem of joint jerk limitations. Specifically, their contributions to resolving the connection dilemma between optimal trajectories involve the utilization of Newton's iteration method. This method identifies the minimum Jerk curve that seamlessly bridges the gap between the maximum Jerk curves, a construct aptly termed as the "bridge." A judiciously applied "extend" method facilitates the determination of optimal Jerk values at these singular points. And it is pointed out that the optimal trajectory structure at this time can alternate into the maximum Jerk curve and the minimum Jerk curve, and ultimately concluding with the maximum Jerk.

In this paper, the combination of the position–velocity phase plane $(s-\dot{s})$ and smooth Improved-Bezier curve makes the proposed algorithm obtain a smooth and time-optimal trajectory. This article employs the numerical integration method and utilizes the TOPP-RA algorithm to solve for the pseudo-velocity trajectory. Finally, an improved Bezier curve is employed to smooth the pseudo-velocity trajectory curve.

The rest of this paper is organized as follows. The equation addressing the problem of time-optimal trajectory of the robotic manipulators is shown in section 2. Section 3 presents the solution for the problem of trajectory planning. Simulations were used to validate the effectiveness of the proposed method in section 4. Finally, Section 5 offers some concluding remarks.

2 Problems Formulation

In order to improve the motion speed of the robot, trajectory planning must take into account the dynamic characteristics of the robot. Given that the dynamic attributes of the robot vary with changes in position, orientation, and payload, it becomes imperative to establish precise dynamic models for the robot.

2.1 Torque Constraint

First, the formulation of the TOTP problem is established. Contemplating a robotic system with n degrees of freedom, accounting for parameters such as link mass, center of mass position, inertia tensors, and frictional forces, the governing equations of motion encapsulating the behavior of the robotic system can be formalized as follows:

$$M(q)\ddot{q}+\dot{q}^{T}C(q)\dot{q}+F(q,\dot{q})+G(q)=\tau \tag{1}$$

The variables $q,\dot{q},\ddot{q}$ respectively denote the joint angles, joint velocities, and joint accelerations. The symbol $M(q)$ represents the inertia matrix, $C(q)$ represents the Coriolis matrix, $F(q,\dot{q})$ represents the frictional force model encompassing Coulomb and viscous friction, $G(q)$ signifies the gravity vector, and τ represents the joint torque vector.

Reference 1 and 2 demonstrate that when the end-effector of a robot moves along the predetermined geometric path, joint angles can be expressed as functions of the path parameter s. This reduction in the number of optimization variables transforms the optimization of joint space problems into a one-dimensional optimization problem in parameter space. Joint velocities and joint accelerations can be obtained by taking the first and second derivatives of joint angles with respect to the parameter s:

$$\dot{q}(s)=q'(s)\dot{s} \tag{2}$$

$$\ddot{q}(s)=q'(s)\ddot{s}+q''(s)\dot{s}^{2} \tag{3}$$

Where $\dot{s}$ is pseudo-velocity, $\ddot{s}$ is pseudo-acceleration, and the rest items can be gained as:

$$\begin{cases} \dot{s}=ds/dt & \ddot{s}=d^{2}s/dt^{2} \\ q'(s)=\partial q/\partial s & q''(s)=\partial^{2}q/\partial s^{2} \end{cases} \tag{4}$$

The torque limit for the motor can be expressed as:

$$\tau_{min}\le\tau\le\tau_{max} \tag{5}$$

Where τ_{min},τ_{max} denote the lower and the upper limit of torque, respectively.

Substituting (2) , (3)and (5) into (1) yields a reduced order system.

$$\tau_{\min}\le M_{s}(s)\ddot{s}+C_{s}(s)\dot{s}^{2}+F_{s}(s,\dot{s})+G_{s}(s)\le\tau_{\max} \tag{6}$$

$$\begin{cases} M_{s}(s)=M(q)\cdot q'(s), \\ C_{s}(s)=M(q)\cdot q''(s)+q'(s)^{T}C(q)\cdot q'(s), \\ F_{s}(s,\dot{s})=F(q,q'(s)\dot{s}),G_{s}(s,\dot{s})=G(q). \end{cases} \tag{7}$$

Note that $\dot{S}\ge0$ for $t\in[0,T]$. Now (6) describes the same dynamic system as (1) in terms of the scalar path coordinate s and its derivatives.

The joint torque constraints can be translated into constraints on pseudo-velocity $\dot{s}$ and pseudo-acceleration $\ddot{s}$, as shown in Figure.1. The feasible region Ω is formed by the torque constraints on the phase plane $\dot{s}-\ddot{s}$.

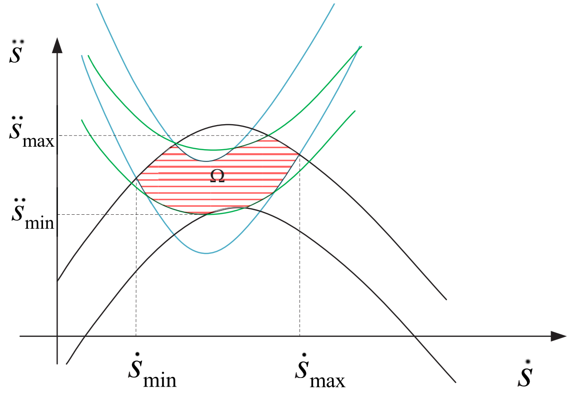

Figure. 1: Admissible region in the plane

2.2 Optimal Problem Formulation

The cost function for the time-optimal trajectory planning problem can be defined as:

$$T=\int_{0}^{T}1dt \tag{8}$$

By the transformation of the integration variable t to s , the cost function can be rewritten as:

$$T = \int_0^T 1dt = \int_0^{s(T)} \frac{1}{\dot{s}} ds \tag{9}$$

Therefore, (9) can be minimized by maximizing the pseudo-velocity under given constraints and the time optimal trajectory planning problem can be formulated as:

$$\min_{\dot{s},\ddot{s}\in\Omega} T = \int_0^{s(T)} \frac{1}{\dot{s}} ds, \tag{10}$$

s.t.

$$\begin{cases} \tau_{\min} \le M_s(s)\ddot{s} + C_s(s)\dot{s}^2 + F_s(s,\dot{s}) + G_s(s) \le \tau_{\max}, \\ s(0)=0, s(T)=L, \dot{s}(0)=\dot{s}_0, \dot{s}(T)=\dot{s}_T, \\ \dot{s}(t) \ge 0, t\in[0,T], \end{cases} \tag{11}$$

Where Ω is the admissible region. The dynamic and kinematic constraints must be observed during the whole trajectory planning process.

3 Time-optimal and Smooth Trajectory Algorithm

3.1 NI Algorithm

The second section provides the comprehensive exposition of the process involving the application of the NI method to address the optimization of trajectories. As shown in Figure.2, firstly, the maximum pseudo velocity at each discrete point is calculated, and the time optimal trajectory is solved under the constraint of the pseudo acceleration $\ddot{s}$ at each discrete point.

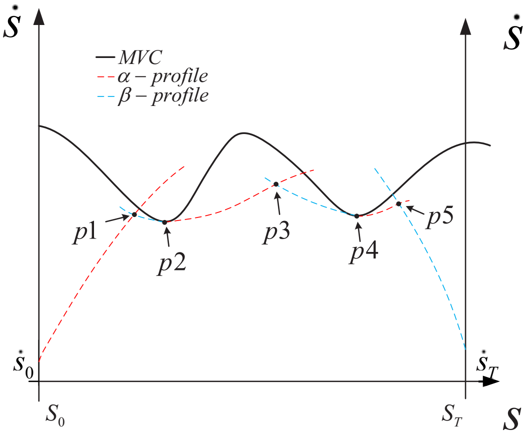

Figure. 2: NI method trajectory planning process

In order to achieve the optimal time of the entire planned trajectory, the robot is required to run alternately at the maximum acceleration and maximum deceleration during operation, so the optimal velocity profile curve is obtained by splicing $\beta-profile$ and $\alpha-profile$ alternately, as shown in Fig.2. However, the challenge lies in determining the intersection points of $\alpha-profile$, $\beta-profile$ and MVC, as exemplified by points $p1,p2,p3,p4,p5$ in Figure.2. the existence of these switching points, such as $p1,p3,p5$,leads to the discontinuity of the pseudo-acceleration $\ddot{s}$, which further leads to the discontinuity of the joint acceleration through (6), which makes this trajectory unfeasible in the actual process.

3.2 TOPP-RA Algorithm

Reference 9 indicated that the TOPP-RA algorithm can effectively bypass the issue of solving switching points. It takes the initial and final velocity profiles of the path as input, utilizes the AVP to compute the controllable set. Specifically, it determines the velocity set R_i at the current time from the velocity set of the path position, solves for the velocity set R_{i-1} at the previous time, and forward solves the maximum reachable velocity set. It is noteworthy that the RA method differs in the approach to solving pseudo-velocity trajectories but still requires solving the MVC curve.

Both the RA algorithm and the NI algorithm require the premise of dynamic model constraints. Combining equations (6) and (7), the equation can be rewritten as (12).

$$\begin{cases} M_s(s)\ddot{s} + C_s(s)\dot{s}^2 + F_s(s,\dot{s}) + G_s(s) - \tau_{\max} \le 0 \\ -M_s(s)\ddot{s} - C_s(s)\dot{s}^2 - F_s(s,\dot{s}) - G_s(s) + \tau_{\min} \le 0 \end{cases} \tag{12}$$

Where $F_s(s,\dot{s}) = K_r \cdot q'(s)\dot{s} + f_k$, K_r represents the viscous friction coefficient, and f_k denotes Coulomb friction force.

For the convenience of calculation, the symbolic matrix is set up as follows:

$$\begin{cases} A(s) = \begin{bmatrix} M_s(s) \\ -M_s(s) \end{bmatrix} \\ B(s) = \begin{bmatrix} C_s(s) \\ -C_s(s) \end{bmatrix} \end{cases} \begin{cases} D(s) = \begin{bmatrix} K_r \cdot q'(s) \\ -K_r \cdot q'(s) \end{bmatrix} \\ E(s) = \begin{bmatrix} f_k + G_s(s) - \tau_{\max} \\ -f_k - G_s(s) + \tau_{\min} \end{bmatrix} \end{cases} \tag{13}$$

After rearranging (12) and (13), the following formula can be obtained:

$$A(s)\ddot{s} + B(s)\dot{s}^2 + D(s)\dot{s} + E(s) \le 0 \tag{14}$$

As velocity and position can both be obtained by integrating acceleration, acceleration is used as the control input for the system, with the square of the path velocity serving as the state variable, which can be expressed as follows:

$$\begin{cases} x_i = \dot{s}_i^{\,2} \\ u_i = \ddot{s}_i \end{cases} \tag{15}$$

The general second-order constraints at the current position s_i are illustrated in Figure.3, where Ω represents the set of states(u_i, x_i) satisfying the dynamic constraints. χ_i denotes the maximum set of state quantities achievable by the constraint plane Ω at the current position, and U_i represents the maximum range of control inputs under the current state.

$$\Omega_i = \left\{(u,x) \mid A_i u + B_i x + D_i\sqrt{x} + E_i \in \xi\right\} \tag{16}$$

$$\begin{cases} \chi_i = \{x \mid \exists u : (u,x) \in \Omega_i\} \\ U_i = \{u \mid (u,x) \in \Omega_i\} \end{cases} \tag{17}$$

One can see Ω_i as the projection of ξ on the plane $\dot{s}^2 - \ddot{s}$. Since ξ is a polytope, Ω_i is a polygon.

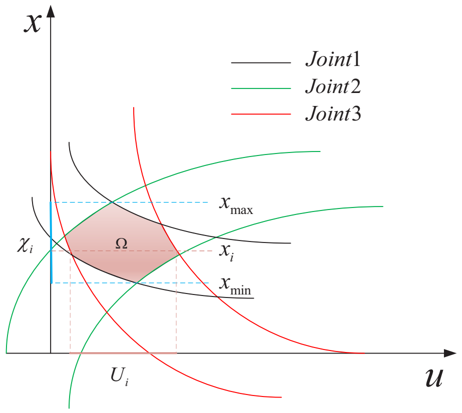

Figure. 3: The general second-order constraints

a) Solution for the Reachable Sets.

Considering a set of starting states I_0. The i-th order reachable set $L_i(I_0)s$ is the set of state $x \in \chi_i$ such that there exists a state $x_0 \in I_0$ and a sequence of admissible controls u that steers the system from x_0 to x .The reaching state set for the i-th discrete point is defined as Z_i .It is achieved by applying a system input u to a specific state $\hat{x}$ from the initial state set at the previous time step, causing the system to transition from state $\hat{x}$ to state x . The state transition formula is determined based on $d\dot{s}^2 = 2\ddot{s}ds$.

$$x = \hat{x} + 2u\Delta s \tag{18}$$

Where $\Delta s = s_{i+1} - s_i$.

The i-th state set Z_i can only represent the collection of states from a certain state to the current state, and it does not guarantee that the current set satisfies the dynamic constraints. Additional constraints need to be added to the reaching set as follows:

$$I_i = Z_{i-1}(L_{i-1}(I_0)) \cap \chi_i \tag{19}$$

b) Solution for the Controllable Sets.

The controllable set solving is based on the terminal state, and the AVP algorithm is used to backward solve the state at the previous time step. Consider a set of desired ending states I_N . The i-stage controllable set $K_i(I_N)$ is the set of states $x \in \chi_i$ such that there exists a state $x_N \in I_N$ and a sequence of admissible controls u that steers the system from x to x_N .Consider a set of states I . The one-step set ψ_i is the set of states $x \in \chi_i$ such that there exists a state $\tilde{x} \in I$ and an admissible control u that steers the system from x to $\tilde{x}$.

$$\begin{gathered} \forall \hat{x} \in I_i,\ \exists x \in Q_i(I_i), \\ \forall u \in U_{i-1} = \{u \mid (u,x) \in \Omega_{i-1}\}, \\ s.t. \quad \hat{x} = x + 2u\Delta s \end{gathered} \tag{20}$$

c) Comprehensive Algorithmic Methodology

Considering the initial state information of path points and the terminal state information of end path points. In the first pass(backward), starting from the last grid point $(s_N, \dot{s}_N)$, the algorithm computes controllable sets (red intervals) recursively. In the second pass (forward), starting now from first grid point $(s_0, \dot{s}_0)$, the algorithm repeatedly selects the highest controls such that resulting velocities remain inside the respective controllable sets, as shown by the green dashed line segments in the Figure.4.

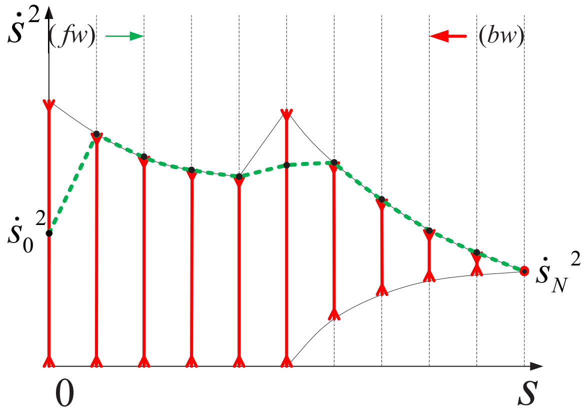

Figure. 4: Time-optimal path parameterization by TOPP-RA computes the optimal parameterization in two passes.

3.3 Improved-Bezier Smoothing Trajectory

the existence of these switching points, such as $p1, p3, p5$ in Figure.2, leads to the discontinuity of the pseudo-acceleration $\ddot{s}$, which further leads to the discontinuity of the joint acceleration through (6), which makes this trajectory unfeasible in the actual process. Diverging from other smoothing algorithms that directly optimize pseudo-acceleration curves or introduce higher-order constraints, this paper optimizes the discontinuity of pseudo-acceleration by addressing the trajectory at points of pseudo-velocity discontinuity in the phase plane $s - \dot{s}$ in order to obtain a smooth trajectory, This aims to mitigate the impact of discontinuous joint torques on the executed trajectory. The study employs an improved Bezier curve to optimize the smooth pseudo-velocity trajectory.

The general formula for a Bezier curve is as follows:

$$B(t) = \sum_{i=0}^{n} B_{n,i}(t) \cdot p_i = [p_0, p_1, p_2, \cdots, p_n] \begin{bmatrix} B_{n,0}(t) \\ B_{n,1}(t) \\ \cdots \\ B_{n,n}(t) \end{bmatrix} \tag{21}$$

Where $B_{n,i}(t) = C_n^i \cdot (1-t)^{n-i} \cdot t^i$ represents the basis function. Points p_i are the control point selected on the time-optimal curve to construct the Bezier curve.

Drawing inspiration from the rational quadratic Bezier curve, an optimization factor K is incorporated into the basis function associated with control point p_1 to determine the ultimate trajectory of the curve, with the weight coefficients for points p_0 and p_2 remaining constant.

$$K = \frac{\max\{|k1|, |k2|\}}{\sqrt{(k1)^2 + (k2)^2}} \tag{22}$$

$k1$, $k2$ represent the slope of the first two points and the last two points.

This article utilizes the least squares method to fit the discrete points on both sides of the intersections $p1, p3, p5$, determining the numerical values for $k1$ and $k2$ .As the acceleration of the discrete points must lie within a constrained plane, the fitted line is vertically shifted to its minimum point. The number of selected discrete points is related to the mean squared error (MSE) .

$$MSE = \frac{(\hat{y}(x_i) - y(x_i))^2}{n} \tag{23}$$

Where n is the number of selected discrete points.

As shown in Figure.5, the blue dashed line represents the Bezier curve, while the red line segment corresponds to the Bezier curve with the added optimization factor. It can be observed that the improved Bezier curve is closer to the original velocity trajectory.

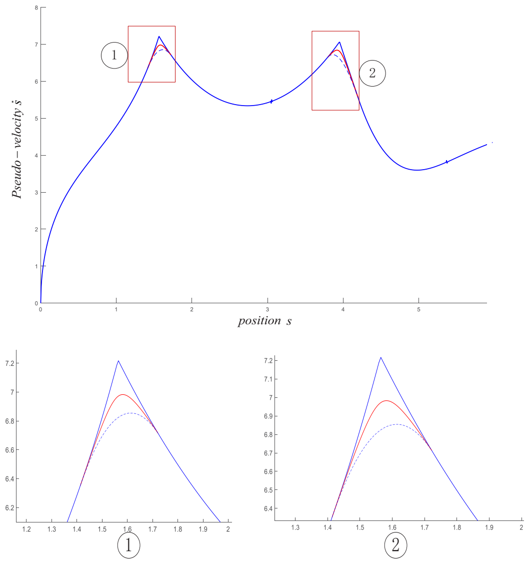

Figure. 5: Smoothing Trajectory

4 Simulation

This simulation was completed in MATLAB 2022a, mainly for the simulation of the two-link manipulator. The link model use the two_link and the position of the center of mass of the links are all at the center of link. The simulated trajectory is a circle with a center at (0, 1) and a radius of 0.5 meters. The torque constraints are as follows:

$$\begin{cases} \tau_{1\min} = -20N \cdot m, \tau_{1\max} = 20N \cdot m \\ \tau_{2\min} = -10N \cdot m, \tau_{2\max} = 10N \cdot m \end{cases} \tag{24}$$

We set $\dot{s}_0 = 0, \dot{s}_n = 0$ and the joint data obtained through simulation is shown in the following figures. The time optimal trajectory curve data obtained by RA algorithm is shown in Figure 6. The red curve in Figure.6 represents the controllable set data, while the blue curve represents the reachable set data. At the same time, we can get the acceleration of the curve as shown in Figure 7. It's noticeable that there are abrupt changes in acceleration occurring at points $Sp1, Sp2, Sp3$ in Figure.7.

The torque curve is shown in Figure.8. At the three points, $point1, point2, point3$,on the torque curve, there are abrupt changes in both joint 1 and joint 2 due to the discontinuity of the pseudo-acceleration in Fig.5. And the torque rate at the discontinuity point is ∞ in TOPP. This can lead to the trajectory being unable to be executed.

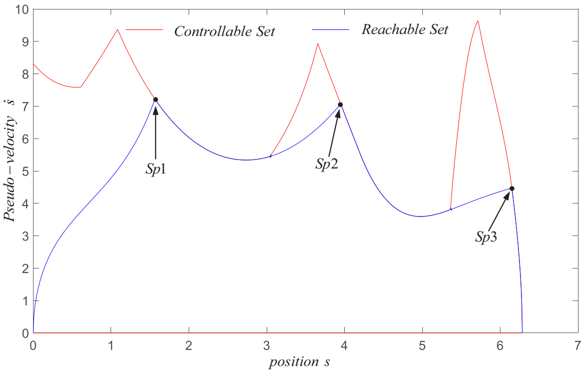

Figure. 6: TOPP-RA Trajectory

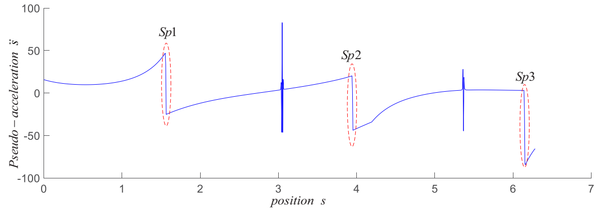

Figure. 7: Pseudo-Acceleration Curve

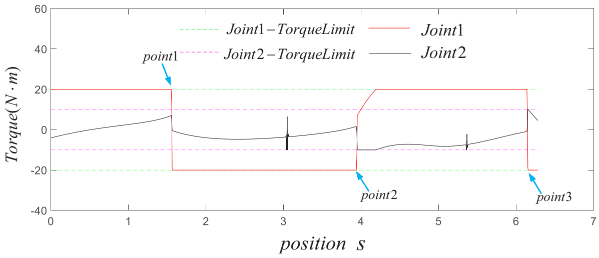

Figure. 8: Torque Curve

To get a continuous torque trajectory, this article utilizes an enhanced Bezier curve to optimize the smoothing of the pseudo-velocity trajectory curve. In this paper, the pseudo-acceleration discontinuity interval is first identified, and the improved Bezier curve is used to optimize the uneven trajectory on the discontinuity interval. The curve in Figure.9 is the obtained trajectory. The blue dashed line represents the Bezier trajectory, and the red curve represents the improved Bezier curve. It is evident that the improved Bezier curve closely approximates the actual curve. Meanwhile, this article will calculate the optimized acceleration and torque variation curves.

As shown in Figure.10, the acceleration curve optimized by the general Bezier curve still has small abrupt changes, and is discontinuous in the initial and end segments of trajectory optimization, while the improved Bezier curve has good continuity in the acceleration curve.

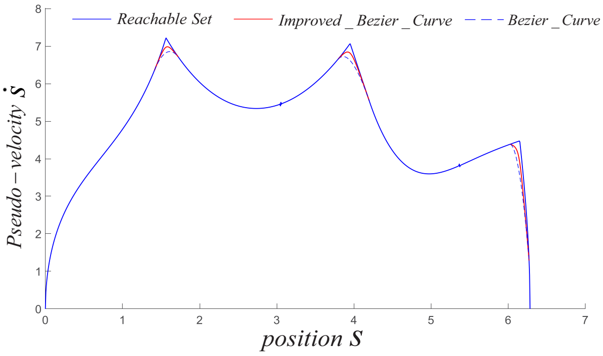

Figure. 9: Improved Bezier smoothing trajectory

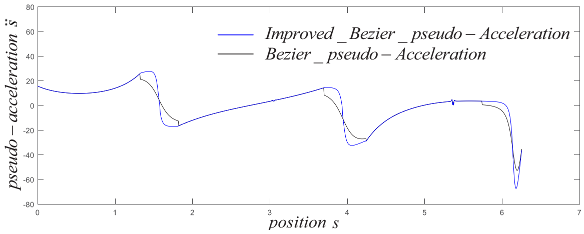

Figure. 10: Bezier and Improved_Bezier Pseudo-Acceleration Curve

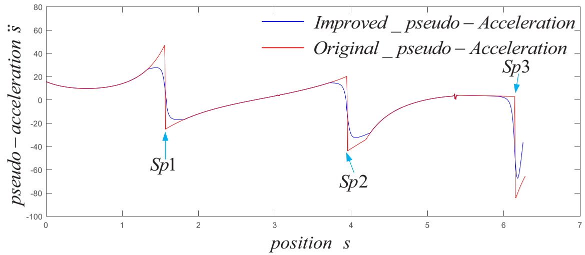

Figure. 11: Improved Pseudo-Acceleration Curve

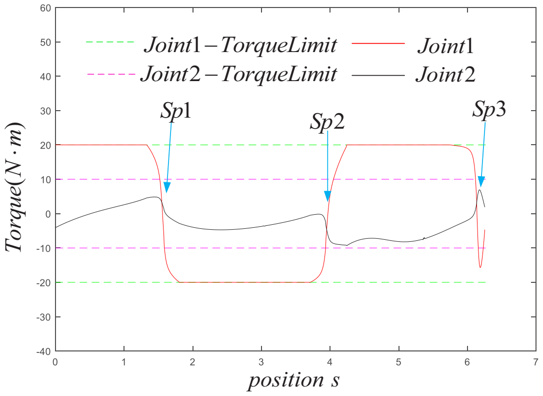

Figure. 12: Improved Torque Curve

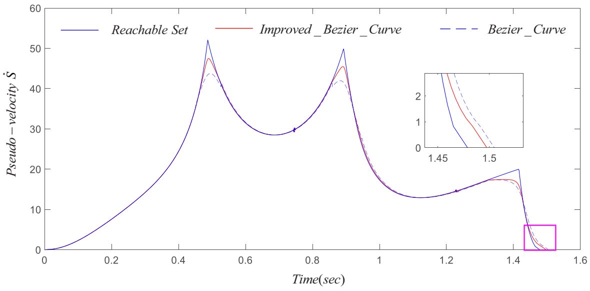

Figure. 13: The time of three curves in TOPP-RA algorithm

It displays a comparison between the improved acceleration and the original acceleration in Figure.11. It's noticeable that at the points $Sp1, Sp2, Sp3$ there are no longer a sudden spike, ensuring a smoother transition. Certainly, the abrupt change in the torque curve at the point shown in Figure.12 has become smooth as well due to the continuity of the pseudo-acceleration.

In Figure 13, the time required for the original trajectory is 1.4788 sec, while the time required for the trajectory optimized by the general Bezier curve is 1.5048 sec, while the time required for the trajectory optimized by the improved Bezier curve is 1.4975 sec. It can be seen that the improved Bezier curve increases by 0.0187 sec in time compared to the original trajectory, but reduces by 0.0073 sec compared to the general Bezier curve.

5 Conclusion

In this paper, the TOPP-RA algorithm is employed to derive the velocity-optimized trajectory. This algorithm can effectively avoid solving the optimal trajectory switching point, greatly reducing the complexity of the solution. In the trajectory planning of the two-link, there must be a joint at the maximum torque. Subsequently, the peaks in the velocity trajectory are smoothed using an improved Bezier curve. Experimental simulations validate that this approach efficiently enhances the trajectory's smoothness, consequently change the discontinuity in torque with only a small-time delay.

References

[1] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, “Time optimal control of robotic manipulators along specified paths,” The International Journal of Robotics Research, vol. 4, no. 3, pp. 3-17, 1985.

[2] Kant, K. and S. W. Zucker. "Toward efficient trajectory planning - the path-velocity decomposition." International Journal of Robotics Research, 1986, 5(3): 72-89.

[3] Singh S. K., Leu M. C. Manipulator Motion Planning In The Presence Of Obstacles And Dynamic Constraints [J]. International Journal of Robotics Research, 1991, 10(2): 171-87.

[4] Kaserer D., Gattringer H., Muller A. Nearly Optimal Path Following With Jerk and Torque Rate Limits Using Dynamic Programming [J]. IEEE Transactions on Robotics, 2019, 35(2): 521-8

[5] Zhang Q., Li S., Guo J.-X., et al. Time-optimal path tracking for robots under dynamics constraints based on convex optimization [J]. Robotica, 2016, 34(9): 2116-39.

[6] Consolini L., Locatelli M., Minari A., et al. Optimal Time-Complexity Speed Planning for Robot Manipulators [J]. IEEE Transactions on Robotics, 2019, 35(3): 790-7.

[7] Ma J.-W., Gao S., Yan H.-T., et al. A new approach to time-optimal trajectory planning with torque and jerk limits for robot [J]. Robotics and Autonomous Systems, 2021, 140.

[8] Quang-Cuong P., Caron S., Lertkultanon P., et al. Admissible velocity propagation: Beyond quasi-static path planning for high-dimensional robots [J]. International Journal of Robotics Research, 2017, 36(1): 44-67

[9] H. Pham, Q.C. Pham, A new approach to time-optimal path parameterization based on reachability analysis, IEEE Trans. Robot, 2018, 34 (3) 645–659.

[10] Zhang, T., Zhang, M. & Zou, Y. Time-optimal and Smooth Trajectory Planning for Robot Manipulators. Int. J. Control Autom. Syst ,2021, 19, 521–531.

[11] Ding Y.-D., Wang Y.-Y., et al. Smooth and proximate time-optimal trajectory planning of robotic manipulators. Transactions of the Canadian Society for Mechanical Engineering. 46(2): 466-476. 2022.

[12] Constantinescu D., Croft E. A. Smooth and time-optimal trajectory planning for industrial manipulators along specified path [J]. Journal of Robotic Systems, 2000, 17(5): 233-49.

[13] H. Pham and Q. -C. Pham, "On the structure of the time-optimal path parameterization problem with third-order constraints," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 679-68