IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE



# An Efficient Linear Programming-Based Time-Optimal Feedrate Planning Considering Kinematic and Dynamics Constraints of Robots

Guanghui Liu, Qiang Li, Bohan Yang, Hualiang Zhang, and Lijin Fang

***Abstract*—This letter investigates the time-optimal trajectory generation for a six-degrees-of-freedom articulated robot moving along a given parametric path. In the generation procedure, besides the velocity, acceleration, and joint torque, the jerk is also constrained to enhance the smoothness of the robot's motion. Meanwhile, the trajectory generation is formulated as a convex optimization problem with a nonlinear objective function and constraints. Then, the problem is solved with a typical linear programming (LP) approach by discretizing the continuous path into many sampling points. Specifically, the time-optimal problem is formulated as maximizing the sum of the velocities at all discrete points instead of minimizing time. Moreover, the time-optimal trajectory generation with nonlinear jerk constraints is decoupled into two sub-LP problems, and the solution of the first sub-LP is employed to scale the nonlinear constraints. Finally, the proposed method is verified through robotic experiments. The results indicate that the smoothness of the generated trajectory improves significantly. Also, the trajectory planning accuracy and computational efficiency are increased by 36% and 62%, respectively.**

***Index Terms*—Jerk constraint, linear programming, time-optimal trajectory, velocity planning.**

## I. Introduction

TIME-OPTIMAL motion planning is a research hotspot in the robotic manipulation domain. Its goal is to find a smooth trajectory in minimized time. Such planning needs to consider the constraints of the kinematics/dynamics of the robot and the obstacles in the environment. One solution for motion planning is to divide motion planning into two sequential stages [1]: the path planning stage and the trajectory generation stage. In the trajectory stage, the joint-level velocity profile is generated given the path computed in the first stage and the constraints of robotic physical properties. At present, there are three main research lines for trajectory generation: dynamic programming, numerical integration, and convex optimization [2]. In dynamic programming, the state space is represented as a discrete grid [3], but it is time-consuming due to the curse of dimensionality. The numerical integration is computationally efficient, but determining the switching point is a great challenge [4] when the acceleration sign is changed.

Convex optimization is a widely used method for time-optimal trajectory generation, and its properties guarantee that the optimization result is the global optimum. It approximates infinite-dimensional optimization with a finite dimension through variable transformation and spatial discretization. Earlier studies discretized the planned path and formulated the trajectory generation as a second-order conic programming (SOCP) problem [5]. Many optimization solvers are available for SOCP, which simply formalize the task into a standard form [6]. However, the drawback of this method is that the computational cost is high due to the nonlinearity of the objective function and constraints. Linear programming (LP) provides an effective way to improve the computational efficiency of solving the convex optimization problem. Hauser et al. [7] developed a sequential linear programming (SLP) approach. Similarly, Nagy et al. [8] employed a special linear programming structure to solve the convex optimization problem. Consolini et al. [9] presented a forward-backward approach to the robotic arm and reduced the computational time when handling acceleration constraints. All the above methods linearize the objective function or nonlinear acceleration constraints, but none constrain the jerk. Using these algorithms without considering the jerk constraint will cause wear or damage to the actuators.

For smoother trajectories, researchers have begun constraining the jerk. For instance, Liu et al. [10] took the maximum allowable velocity as an initial solution into the nonlinear constraint items of acceleration and jerk to linearize the nonlinear constraints. This method enlarges the constraint's limits but eventually leads to reduced optimization efficiency and violations of acceleration or jerk. Alessandro et al. established a nonlinear and non-convex time-optimal pathing tracking formulation including jerk constraints, and they used convex relaxation

Manuscript received 5 September 2023; accepted 18 January 2024. Date of publication 29 January 2024; date of current version 9 February 2024. This letter was recommended for publication by Associate Editor Enrico Mingo Hoffman and Editor Lucia Pallottino upon evaluation of the reviewers' comments. This work was supported by the China Scholarship Council under Grant 202106080049. *(Corresponding authors: Guanghui Liu; Qiang Li.)*

Guanghui Liu is with the School of Artificial Intelligence, Shenyang University of Technology, Shenyang 110870, China (e-mail: guanghuiliu@sut.edu.cn).

Qiang Li is with the College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518122, China (e-mail: liqiang1@sztu.edu.cn).

Bohan Yang and Hualiang Zhang are with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110169, China (e-mail: yangbohan1997@gmail.com; zhanghualiang@sia.cn).

Lijin Fang is with the Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: ljfang@mail.neu.edu.cn).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3359547, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3359547

2377-3766 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

**Copyright ©2024 IEEE**

Authorized licensed use limited to: Northeastern University. Downloaded on April 17,2024 at 01:08:08 UTC from IEEE Xplore. Restrictions apply.

and McCormick envelopes to ensure the solving efficiency and reliability [11]. This method is also successfully used in flexible joint robots [12]. Dominik et al. developed a new dynamic programming approach to obtain the optimal path, which relies on interpolation in the phase plane so that the resulting joint accelerations and joint torques are continuous [13]. Wang proposed an approximate time-optimal feedrate planning algorithm that considers third-order constraints [2]. To address the complexity of robot feedrate scheduling and enhance the optimality of machining time, Xiao et al. developed a jerk-limited heuristic feedrate scheduling method with near-optimal time [14].

Aiming at the linearization of the convex optimization problem and improving the smoothness of the trajectory, the main contributions of this paper are as follows:

- A time-optimal robot trajectory planning model is proposed, which includes constraints on joint velocities, accelerations, jerks, and joint torques.
- The discretized form of the trajectory generation with nonlinear constraints is formulated, and it is decoupled into two solvable sub-LP problems.
- The influence of jerk constraints on the trajectory accuracy, smoothness, and computational cost of trajectory generation is evaluated quantitatively.

The rest of this paper is organized as follows. The trajectory generation is formulated as an optimization problem in Section II. In Section III, LP-based time-optimal trajectory generation without jerk constraints is introduced, and our proposed method for linearizing the nonlinear cost function is explained. In Section IV, the solution of trajectory generation considering jerk constraints is described. In Section V, experiments are carried out to (1) compare time-optimal trajectory planning with and without jerk constraints, (2) compare the computational efficiency of the proposed method with two other approaches, which consider the same constraints as the proposed method, and (3) investigate the application of robot position and posture synchronous interpolation. Finally, the conclusion is drawn in Section VI.

## II. Problem Formulation

The robotic dynamics model can be represented as

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F_c(q)sgn(\dot{q}) \tag{1}$$

where $\tau$ denotes the joint torques; $q$, $\dot{q}$, and $\ddot{q}$ denote the joint's angle, velocity, and acceleration, respectively; $M(q) \in R^{k\times k}$ is a positive definite mass matrix, and $C(q,\dot{q}) \in R^{k\times k}$ is a matrix accounting for Coriolis and centrifugal effects, which is linear in the joint velocity; $F_c(q) \in R^{k\times k}$ is a matrix of Coulomb friction torques, which is dependent on the joint angle, while $G(q) \in R^k$ denotes the vector accounting for gravity. In this paper, viscous friction is ignored to reformulate the equations of motion as a linear set.

The trajectory of the robot in the configuration space can be obtained from the planned path in the operational space using inverse kinematics. In this paper, the joint trajectory is described in a parametric way $S(u)$, where $u$ denotes the parameter and $u \in [0,1]$. The parameter $u$ of the trajectory is a variable that evolves with time $t$. Without loss of generality, it is assumed that the trajectory starts at $t = 0$ and ends at $t = T$:

$$u(0) = 0 \le u(t) \le 1 = u(T) \tag{2}$$

According to the differential chain rule, the velocity, acceleration, and jerk of the robot can be expressed as the parameter and its derivatives

$$\begin{cases} \dot{q}(u) = S'(u)\dot{u} \\ \ddot{q}(u) = S'(u)\ddot{u} + S''(u)\dot{u}^2 \\ \dddot{q}(u) = S'(u)\dddot{u} + 3S''(u)\dot{u}\ddot{u} + S'''(u)\dot{u}^3 \end{cases} \tag{3}$$

where $\dot{u} = du/dt$, $S' = dS/du$, $S'' = d^2S/du^2$, and $S''' = d^3S/du^3$. Substituting $\dot{q}(u)$ and $\ddot{q}(u)$ into (3) yields the following expression for the equations of motion

$$\tau(u) = m(u)\ddot{u} + c(u)\dot{u}^2 + g(u) \tag{4}$$

where

$$\begin{cases} m(u) = M(q(u))S'(u) \\ c(u) = M(S(u))S''(u) + C(S(u),S'(u))S'(u) \\ g(u) = F_c(S(u))sgn(S'(u)) + G(S(u)) \end{cases} \tag{5}$$

and $sgn(\dot{q}(u))$ is replaced by $sgn(S'(u))$ based on (3) and the assumption that $\dot{u} > 0$ almost everywhere.

Similarly to [6], the time-optimal trajectory generation problem for the robot is subject to lower and upper bounds on the torque, velocity, acceleration, and jerk, which can be expressed as

$$T = \int_0^T 1dt$$
$$s.t. \begin{cases} \tau(u) = m(u)\ddot{u} + c(u)\dot{u}^2 + g(u) \\ \underline{\tau}(u) \le \tau(u) \le \overline{\tau}(u) \\ -\dot{q}_{max} \le S'(u)\dot{u} \le \dot{q}_{max} \\ -\ddot{q}_{max} \le S'(u)\ddot{u} + S''(u)\dot{u}^2 \le \ddot{q}_{max} \\ -\dddot{q}_{max} \le S'(u)\dddot{u} + 3S''(u)\dot{u}\ddot{u} + S'''(u)\dot{u}^3 \le \dddot{q}_{max} \\ \dot{q}(0) = \dot{q}_0, \dot{q}(T) = \dot{q}_T \end{cases} \tag{6}$$

where $\underline{\tau}(u)$ and $\overline{\tau}(u)$ are the lower bounds and upper bounds on the torque, respectively. $\dot{q}_{max}$, $\ddot{q}_{max}$, and $\dddot{q}_{max}$ represent the joint's maximum velocity, maximum acceleration, and maximum jerk, respectively. $\dot{q}_0$ and $\dot{q}_T$ denote the start velocity and end velocity of the path, and since the path has no boundary relations with other paths, they are both zero in this paper.

By substituting $\dot{u} = du/dt$ to the objective function of (6), the time-optimal problem can be transformed from the time domain $\{t\}$ to the parameter domain $\{u\}$

$$T = \int_0^T 1dt = \int_{u(0)}^{u(T)} \frac{1}{du/dt}du = \int_0^1 \frac{1}{\dot{u}}du \tag{7}$$

Defining the state variables $\alpha$ and $\beta$, (7) can be transformed into a convex problem [15]

$$\alpha(u) = \dot{u}^2, \quad \beta(u) = \ddot{u}. \tag{8}$$

Authorized licensed use limited to: Northeastern University. Downloaded on April 17,2024 at 01:08:08 UTC from IEEE Xplore. Restrictions apply.

$\alpha$, $\beta$ are related as follows

$$\alpha'(u) = \frac{d\dot{u}^2}{du} = 2\ddot{u} = 2\beta(u) \tag{9}$$

By substituting $\alpha(u)$ and $\beta(u)$ to (6), the time-optimal trajectory generation can be represented as

$$\min_{\alpha(u),\beta(u),\tau(u)} \int_0^1 \frac{1}{\sqrt{\alpha(u)}} du$$
$$s.t. \begin{cases} \tau(u) = m(u)\beta(u) + c(u)\alpha(u) + g(u) \\ \underline{\tau}(u) \le \tau(u) \le \overline{\tau}(u) \\ \left|\sqrt{\alpha(u)}S'(u)\right| \le \dot{q}_{max} \\ \left|\beta(u)S'(u) + \alpha(u)S''(u)\right| \le \ddot{q}_{max} \\ \left|(S'''(u)\alpha(u) + \frac{3S''(u)}{2}\alpha'(u) + \frac{S'(u)}{2}\alpha''(u))\sqrt{\alpha(u)}\right| \\ \le \dddot{q}_{max} \\ \alpha'(u) = 2\beta(u) \\ \alpha(0) = 0, \alpha(n) = 0 \end{cases} \tag{10}$$

The time-optimal trajectory generation is formulated as an optimization problem that contains a nonlinear objective function and multiple nonlinear constraints. To solve it efficiently, this study linearizes the function and constraints using an LP approach and considers constrained jerks to enhance the smoothness of velocity and acceleration. The linear optimization of (10) without jerk constraints is presented in Section III, and the linear optimization of the jerk constraints is discussed in Section IV.

## III. LP-Based Time-Optimal Trajectory Generation Without Jerk Constraints

Time-optimal trajectory generation is an infinite-dimensional optimization problem, which can be transformed into a finite-dimensional convex optimization problem through variable transformation and spatial discretization. The parameter interval [0, 1] is sampled uniformly in a step of $h = \frac{1}{n}$ to obtain $\{u_0 = 0, u_1, \ldots, u_{n-1}, u_n = 1\}n + 1$ sampling points, and the objective function in (10) is discretized as

$$\min_{\alpha_i} 2h \sum_{i=0}^{n-1} \left(\frac{1}{\sqrt{\alpha_i} + \sqrt{\alpha_{i+1}}}\right) \tag{11}$$

Since the minimum motion time is equivalent to the maximum sum of velocities passing through all sampling points, (11) is reformulated in a linear form as

$$\max_{\alpha_i} \sum_{i=0}^{n} \alpha_i \tag{12}$$

After dividing the interval of the curve parameter $u \in [0, 1]$ into $n$ equal parts, $m(u)$, $c(u)$, and $g(u)$ in (10) at all sampling points can be calculated directly.

### A. Linearization of Velocity Constraints

Usually, the velocities over the sampling points in Cartesian space are constrained by chord error[1] and centripetal acceleration [17].

Constrained by the chord error

$$v_i \le \frac{2}{T_s}\sqrt{\frac{1}{\rho_i^2} - \left(\frac{1}{\rho} - \delta_{max}\right)^2} \tag{13}$$

where $v_i$ denotes the velocity of the $i$-th sampling point, $\delta_{max}$ represents the maximum chord error in Cartesian space, $T_s$ indicates the step length of interpolation, and $\rho_i$ is the curvature at the $i$-th point, which can be calculated as follows:

$$\rho_i = \frac{||S_i'||_2^3}{||S_i' \times S_i''||_2} \tag{14}$$

Constrained by the centripetal acceleration, the velocity should also satisfy:

$$v_i \le \sqrt{\frac{A_{max}^c}{\rho_i}} \tag{15}$$

where $A_{max}^c$ denotes the maximum centripetal acceleration in Cartesian space, so the restricted velocity $v_i^R$ can be represented as

$$v_i^R = \min\left\{\frac{2}{T_s}\sqrt{\frac{1}{\rho_i^2} - \left(\frac{1}{\rho} - \delta_{max}\right)^2}, \sqrt{\frac{A_{max}^c}{\rho_i}}, V_{max}\right\} \tag{16}$$

where $V_{max}$ denotes the maximum velocity of the robot in Cartesian space, and the constraints on the joint's velocity are as follows

$$\alpha_i \le \min\left\{\dot{q}_i^R, \dot{q}_{max}\right\} \tag{17}$$

where $i = 0, 1, \ldots, n$; $\dot{q}_i^R$ represents the maximum velocity of each sampling point in the joint space, and it can be transformed from $v_i^R$ through the inverse kinematics; $\dot{q}_{max}$ represents the joint maximum velocity.

### B. Linearization of the Acceleration Constraints

The first-order derivative of $\alpha$ can be approximated as

$$\alpha_i' = 2\beta_i \approx \frac{\alpha_{i+1} - \alpha_i}{h} \tag{18}$$

where $i = 0, 1, \ldots, n-1$.

The discrete forms of velocity and acceleration are shown in (19) and (20), respectively.

$$\dot{S}_i = S_i'\sqrt{a_i} \tag{19}$$

$$\ddot{S}_i = S_i'(u)\beta_i + S_i''(u)\overline{\alpha}_i \tag{20}$$

where

$$\overline{\alpha}_i = \eta_i\alpha_{i+1} + (1 - \eta_i)\alpha_i \tag{21}$$

[1]Chord error is defined as the Hausdorff distance between the desired tool path and the generated tool path. It is used to describe the accuracy of trajectory generation. Its definition can be found in Fig. 1(b) in [16].

Authorized licensed use limited to: Northeastern University. Downloaded on April 17,2024 at 01:08:08 UTC from IEEE Xplore. Restrictions apply.

$$\eta_i = \begin{cases} 1, & S_i' S_i'' \geq 0 \\ 0, & S_i' S_i'' < 0 \end{cases} \tag{22}$$

where $S_i'$=$S'(u_i)$, $S_i''$=$S''(u_i)$.

Therefore, the acceleration constraint is represented as

$$|(S_i' + 2hS_i''\eta_i)\alpha_{i+1} + (-S_i' + 2hS_i''(1-\eta_i))\alpha_i| \leq 2h\ddot{q}_{max} \tag{23}$$

where $\ddot{q}_{max}$ denotes the maximum acceleration.

From (22), we have $(S_i' + 2hS_i''\eta_i)(-S_i' + 2hS_i''(1-\eta_i)) < 0$. So, the constraint (23) can be split into the following two linear constraints

$$\begin{aligned} \alpha_i &\leq \frac{S_i' + 2hS_i''\eta_i}{S_i' - 2hS_i''(1-\eta_i)}\alpha_{i+1} + \left|\frac{2h\ddot{q}_{max}}{S_i' - 2hS_i''(1-\eta_i)}\right| \\ &= f^i(\alpha_{i+1}) \\ \alpha_{i+1} &\leq \frac{S_i' - 2hS_i''(1-\eta_i)}{S_i' + 2hS_i''\eta_i}\alpha_i + \left|\frac{2h\ddot{q}_{max}}{S_i' + 2hS_i''\eta_i}\right| \\ &= g^i(\alpha_i) \end{aligned} \tag{24}$$

Finally, the discretized form of the time-optimal trajectory planning without jerk constraints can be represented as

$$\max_{\alpha_i,\beta_i,\tau_i} \sum_{i=0}^{n} \alpha_i$$

$$s.t. \begin{cases} \tau_i = m(u)\beta_i + c(u)\alpha_i + g(u) \\ \underline{\tau}(u) \leq \tau_i \leq \overline{\tau}(u) \\ 0 \leq \alpha_i \leq \min\{\dot{q}_i^R, \dot{q}_{max}\}, i = 0, \ldots, n. \\ \alpha_i \leq f^i(\alpha_{i+1}), i = 0, \ldots, n-1. \\ \alpha_{i+1} \leq g^i(\alpha_i), i = 0, \ldots, n-1. \\ \beta_i = \dfrac{\alpha_{i+1} - \alpha_i}{2h}, i = 0, \ldots, n-1. \\ \alpha_0 = 0, \alpha_n = 0 \end{cases} \tag{25}$$

It indicates that the objective function and the constraints of (25) are all linear, which can be solved easily by LP solvers.

## IV. Planning With Jerk Constraints

Since (18) is an approximate estimation, the acceleration optimized by (25) will have many overruns over limits. Overruns will disappear only when there are infinitely many discretized sampling points, indicating that the optimization will become time-consuming. This section introduces the trajectory generation problem that includes jerk constraints. It can be used to constrain the jerk and realize the de-overrunning processing of the acceleration curve without increasing the sampling points.

Similar to (18), the second-order derivative of $\alpha$ can be approximated as

$$\alpha_i'' \approx \frac{\alpha_{i+1} + \alpha_{i-1} - 2\alpha_i}{h^2} \tag{26}$$

where $i = 1, 2, \ldots, n-1$.

Substituting (26) and (18) into the jerk constraint inequality of (10), let $\cdot(i)$ represent $\cdot(u)$ at the $i$-th sample point, the jerk constraint inequality can be discretized as

$$|(\chi_i\alpha_{i-1} + \gamma_i\alpha_i + \lambda_i\alpha_{i+1})|\sqrt{\alpha_i} \leq \dddot{q}_{max} \tag{27}$$

where $i = 1, 2, \ldots, n-1$, and

$$\begin{cases} \chi_i = \dfrac{S_i'}{2h^2} - \dfrac{3S_i''}{4h} \\ \gamma_i = S_i''' - \dfrac{S_i'}{h^2} \\ \lambda_i = \dfrac{S_i'}{2h^2} + \dfrac{3S_i''}{4h} \end{cases} \tag{28}$$

Specifically, the jerk at the start point and the end point can be estimated as

$$\dddot{q} = \frac{d\ddot{q}}{dt} = \ddot{q}\frac{d\ddot{q}}{d\dot{q}} = \frac{1}{2}\frac{d\ddot{q}^2}{d\dot{q}} \tag{29}$$

For the first interval in $[0, u_1]$,

$$\begin{aligned} |\dddot{q}_1| &= |\ddot{q}_1^2 - \ddot{q}_0^2|/(2\dot{q}_1 - 2\dot{q}_0) \\ &= |\ddot{q}_1^2/(2S_1'\sqrt{\alpha_1})| \\ &= |n^2 S_1'\sqrt{\alpha_1}/2| \end{aligned} \tag{30}$$

Similarly, for the last interval in $[u_{n-1}, 1]$,

$$|\dddot{q}_{n-1}| = |n^2 S_{n-1}'\sqrt{\alpha_{n-1}}/2| \tag{31}$$

The jerk constraints in the first and last intervals are represented as follows

$$\begin{cases} |n^2 S_1'\sqrt{\alpha_1}/2| \leq \dddot{q}_{max} \\ |n^2 S_{n-1}'\sqrt{\alpha_{n-1}}/2| \leq \dddot{q}_{max} \end{cases} \tag{32}$$

To enhance the optimization efficiency, an effective approach is to linearize the nonlinear constraints of (27) and (32). Let the solution to (25) be $a_i^*$. Then, any $\alpha_i$ satisfying the constraints of (10) will also satisfy (25), and we have $\alpha_i \leq \alpha_i^*$. $\sqrt{a_i^*/a_i}$ is multiplied on both sides of (27) and (32). Considering the inequality

$$1 \leq \frac{3}{2} - \frac{a_i}{2a_i^*} \leq \sqrt{\frac{a_i^*}{a_i}} \tag{33}$$

In practice, $\alpha_i$ is approximately equal to $\alpha_i^*$, i.e., (33) is an effective transformation.

The nonlinear terms in (27) and (32) are replaced by $\alpha_i^*$, thereby turning the jerk's nonlinear constraints into linear constraints. However, the final optimized acceleration often has jitters, which also results in un-smooth motion. An effective processing method is to sum the jerks and place them in the objective function with a certain weight

$$\max_{\alpha_i}\left(\sum_{i=0}^{n}\alpha_i - \mu\cdot\sum_{i=1}^{n-1}\left|\sqrt{\alpha_i^*}\chi_i\alpha_{i-1} + \sqrt{\alpha_i^*}\gamma_i\alpha_i + \sqrt{\alpha_i^*}\lambda_i\alpha_{i+1}\right|\right)$$

$$s.t. \begin{cases} 0 \leq \alpha_i \leq \alpha_i^* \\ \left|n^2 S_1'\sqrt{\alpha_1^*}/2\right| \leq J_{max}\left(\dfrac{3}{2} - \dfrac{\alpha_1}{2\alpha_1^*}\right) \\ \left|n^2 S_{n-1}'\sqrt{\alpha_{n-1}^*}/2\right| \leq J_{max}\left(\dfrac{3}{2} - \dfrac{\alpha_{n-1}}{2\alpha_{n-1}^*}\right) \\ \left|\sqrt{\alpha_i^*}\chi_i\alpha_{i-1} + \sqrt{\alpha_i^*}\gamma_i\alpha_i + \sqrt{\alpha_i^*}\lambda_i\alpha_{i+1}\right| \\ \leq J_{max}\left(\dfrac{3}{2} - \dfrac{\alpha_i}{2\alpha_i^*}\right) \end{cases} \tag{34}$$

Authorized licensed use limited to: Northeastern University. Downloaded on April 17,2024 at 01:08:08 UTC from IEEE Xplore. Restrictions apply.


Fig. 1. Writing task.

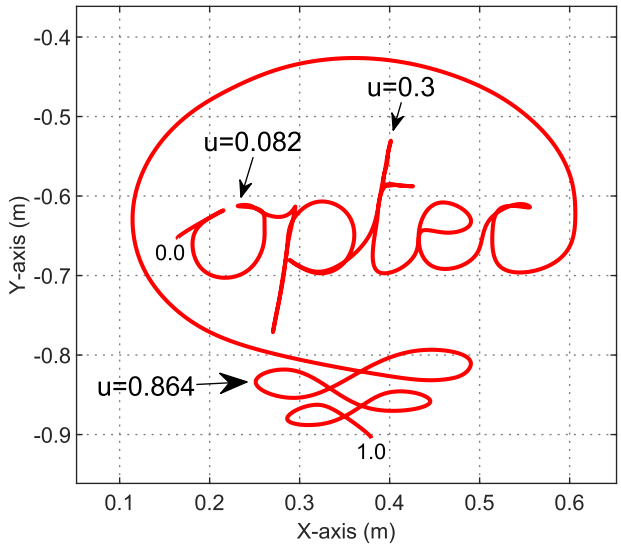


Fig. 2. Writing task path in the XY-plane.

where $\mu$ is the weight coefficient. Finally, to improve computational efficiency, the time-optimal trajectory planning problem including the jerk constraint is decomposed into two sub-LP problems. **(1)** Solve the solution $\alpha_i^*$ of (25) by LP solvers. **(2)** Substitute $\alpha_i^*$ into (34), and the final result is the optimal solution with jerk constraints.

Taylor's series is utilized to transform the curves from the parameter domain to the time domain, thus allowing the interpolation information to be sent at a certain time frequency.

$$u_{i+1} \approx u_i + T_s \dot{q}(u)\big|_{u=u_i} + \frac{T_s^2}{2}\frac{d\dot{q}(u)}{du}\dot{q}(u)\big|_{u=u_i} \tag{35}$$

where $T_s$ denotes the step length over the time domain.

## V. Experiments

Two complex paths are adopted to validate effectiveness of the proposed method. The actuator is a ROKAE-XB7 industrial robot, and its kinematics and dynamics information can be found in [18]. To avoid disturbance from a multi-dimensional path, the first task selected is a two-dimensional planar path. In this task, the accuracy and smoothness of generated trajectories with and without jerk constraints will be compared. Meanwhile, the computational efficiency is compared with that of SOCP in [5] and an LP in [10]. To verify the application of the proposed method in six-dimensional space, the second task is an impeller edge tracking experiment involving position-attitude synchronization. The two experiments are optimized by MATLAB R2022b installed on a laptop equipped with an Intel Core i5-4300 U CPU (2.49 GHz), and the optimization solver is the GUROBI of the YAMLIP toolbox [19].

### A. Experiment on a Planar NURBS Tool Path

As illustrated in Fig. 1, the writing task path in [5] is taken as an example, which is described by the NURBS parametric in this paper. Fig. 2 shows the NURBS curve in the XY-plane, and

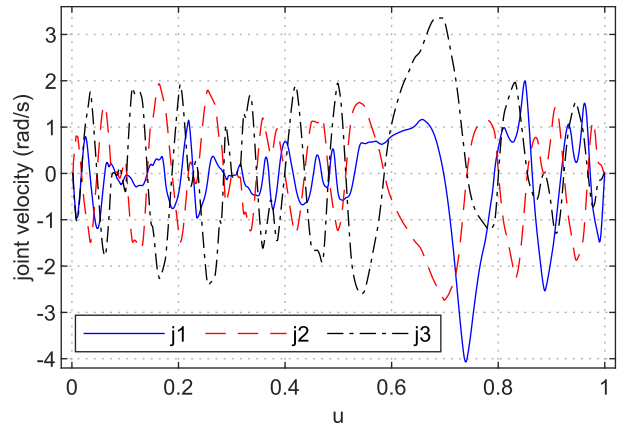


Fig. 3. Velocities of the first three joints with jerk constraints.

the detailed information regarding control points and weighted scales can be found in [5]. The maximum allowable chord error and the maximum centripetal acceleration are 0.0005 mm and 3000 mm/s$^2$, respectively. According to the performance limits of the robot, the maximum bounds of velocity, acceleration, and jerk in the joint space are 5 rad/s, 15 rad/s$^2$, and 500 rad/s$^3$, respectively. Besides, the number of sampling points is 2000.

To avoid reading difficulties due to excessive data, this paper only presents the velocity, acceleration, and jerk of the first three joints of the robot. Fig. 3 shows the joints' velocities optimized by (34) ($\mu = 1e-5$) with torque, velocity, acceleration, and jerk constraints. It can be seen that the joints' velocities are continuous and smooth, and the initial and final velocities are all 0, which meets the initial constraints.

Figs. 4 and 5 show the joint acceleration without and with jerk constraints, respectively. Without the jerk constraint, the acceleration of each axis exhibits slight overruns and a lot of jitters. To find the reason for exceeding the limit, three moments when the acceleration exceeded the limit are marked in

Authorized licensed use limited to: Northeastern University. Downloaded on April 17,2024 at 01:08:08 UTC from IEEE Xplore. Restrictions apply.

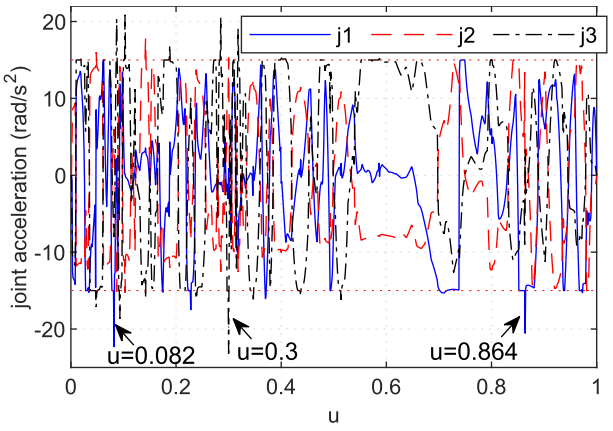


Fig. 4. Accelerations of the first three joints without jerk constraints.

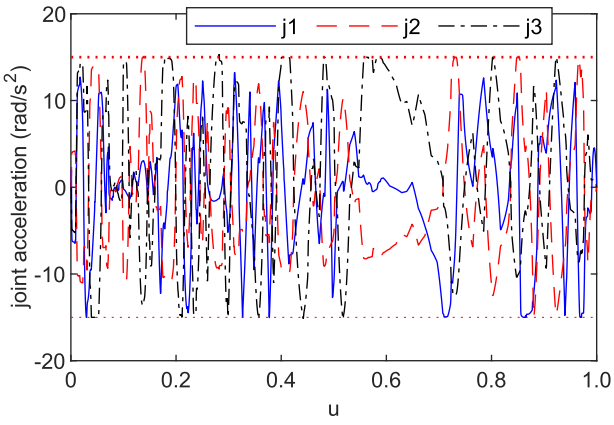


Fig. 5. Accelerations of the first three joints with jerk constraints.

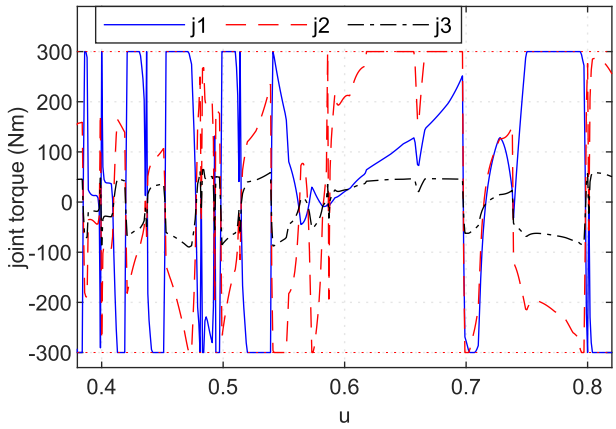


Fig. 6. Torques of the first three joints without jerk constraints.

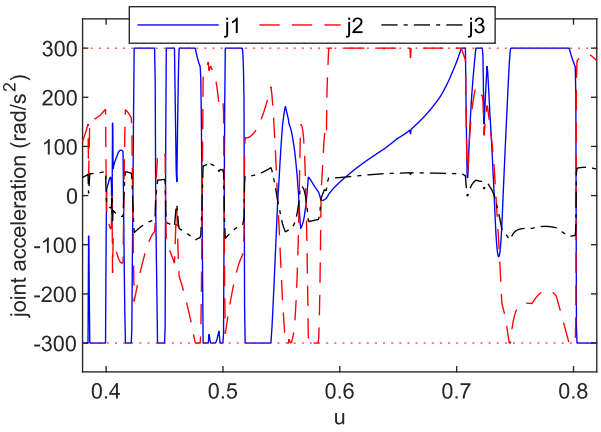


Fig. 7. Torques of the first three joints with jerk constraints.

Fig. 4, namely $u = 0.082$, 0.3, and 0.864, and they are also marked in Fig. 2 accordingly. It can be seen that the three accelerations exceeding the limit moments correspond to the three sharp corners of the curve. That is, the existence of sharp corners makes it difficult to effectively constrain acceleration. A potentially effective approach is to densely sample points at sharp corners, but this process is tedious, and the optimization is time-consuming. In contrast, with the jerk constraint proposed in this paper, the acceleration is smooth and no longer exceeds the limit. It is noteworthy that the value of $\mu$ in (34) will affect the optimization results. When the value of $\mu$ is too small, it will cause the acceleration in the optimization results to exceed the limit. For example, the sampling point of the writing task is 2000, and if the value $\mu$ is 0, acceleration jitters and over-limits can be significantly eliminated from the optimization results of (34), but there are still few that exceed the limit. If the value of $\mu$ is 0.5, acceleration will seldom exceed the limit.

To show the data more clearly, Figs. 6 and 7 illustrate the torque information of the first three joints between $u = 0.38$ and $u = 0.82$. It can be found that when $u \approx 0.4$ and $u \approx 0.58$, the torque of joint 2 in Fig. 6 fluctuates rapidly, while the torque in Fig. 7 is smooth. This shows that the curve planned by the proposed method is smoother and more feasible for moving.

TABLE I
PERFORMANCE COMPARISON OF NURBS PATHS UNDER DIFFERENT CONSTRAINTS

| Constraint | Time (s) | | Chord error (mm) | |
|---|---|---|---|---|
| | Planning | Machining | Mean | Maximum |
| Without jerk | 0.18 | 9.2227 | 0.00011 | 0.00089 |
| With jerk | 0.63 | 10.7486 | 0.00007 | 0.00041 |

Table I shows the planning time, machining time (the time for the robot to track along the writing task path), and chord errors of the results optimized by (25) and (34). With the jerk constraint, the planning time increases by 0.45 s, and the machining time also increases by 1.5259 s. Obviously, with jerk constraints, the optimization trajectory becomes smoother. Meanwhile, the average chord error decreases from 0.00011 mm to 0.00007 mm, and the maximum chord error also decreases from 0.00089 mm to 0.00041 mm. To quantify the improvement in the smoothness of the velocity profile that contains jerk constraints, the concept of feedrate fluctuations [20] is adopted. The average feedrate fluctuation without and with jerk constraints is 0.0788 and 0.0048, respectively. This indicates that the optimization velocity with jerk constraints is smoother. Additionally, the writing task was executed by a real industrial robot, as shown in

Authorized licensed use limited to: Northeastern University. Downloaded on April 17,2024 at 01:08:08 UTC from IEEE Xplore. Restrictions apply.

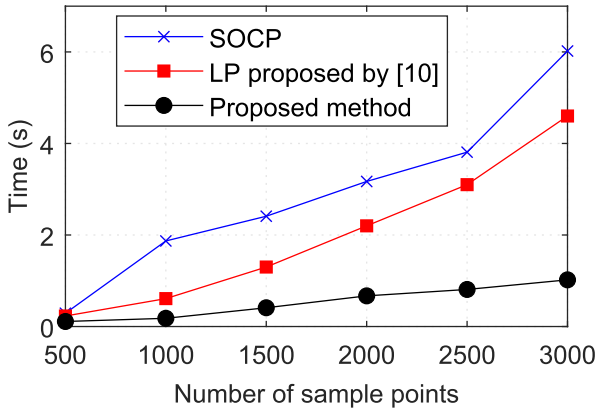


Fig. 8. Comparison of the time-consumption of three methods with different sampling points.

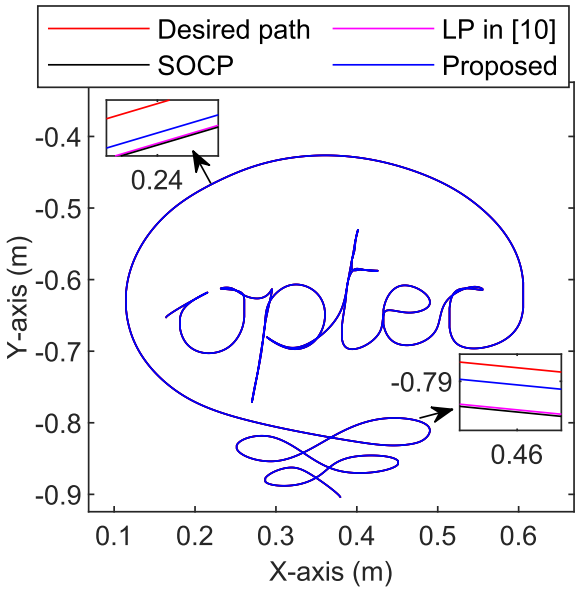


Fig. 9. Comparison between the desired path and real robot path.

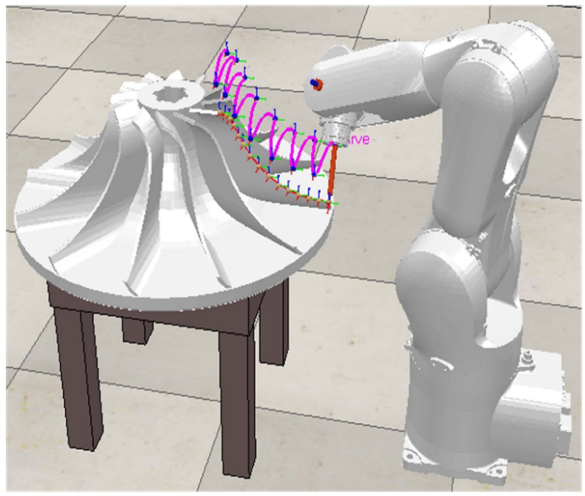

Fig. 10. Robot path planning for a complex curve.

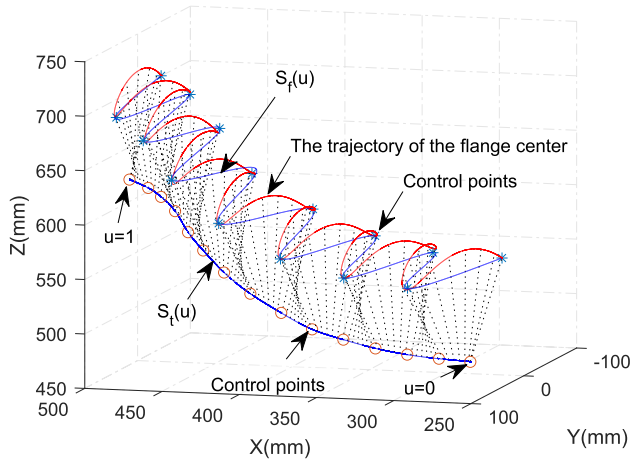


Fig. 11. Double NURBS paths for position and attitude synchronization.

the YouTube video https://youtu.be/McSoQKcG2FA. It can be observed that the robot's motion is smooth, which is consistent with the result shown in Fig. 3, i.e., all joints' velocities are smooth and continuous.

To verify the computational efficiency of the proposed method, the SOCP and the LP method in [10] were employed to solve the same problem. The computation time of the three methods at different sampling points is shown in Fig. 8, and it can be found that the proposed method is more efficient. Meanwhile, as the number of sampling points increases, the computational advantage of the proposed method becomes more prominent. In practice, users need to select more sampling points to ensure the accuracy of the curve. In this regard, the proposed method is more effective.

Finally, the real robot motion trajectories of the three methods are demonstrated in Fig. 9, where the red curve marks the desired path of the robot. It can be found that the robot motion trajectory of the proposed method is closer to the desired path than those of the other two methods. This indicates that the method proposed in this paper has a smaller contour error than the other two methods.

### B. *Experiment on a Six-Axis Robot Dual NURBS Paths*

As illustrated in Fig. 10 and the YouTube video https://youtube.com/shorts/VCq18w3FVsg, the edge of the impeller is the complex curve to be tracked. Not only is the robot tool tip required to track the impeller edge, but the center of the robot flange must also pass through the attitude control points in turn. In this case, two significantly different curves should be tracked by the tip of the robot tool and the center of the robot flange, respectively. Here, the same parameter vector is used to achieve synchronization of position and attitude. As demonstrated in Fig. 11, $S_t(u)$ describes the locus of the tooltip locations, and $S_f(u)$ describes the locus of the center of the robot flange. Both $S_t(u)$ and $S_f(u)$ have the same parameter setting to synchronize the two curves. The calculation of the unit vector $O(u)$ of tool

Authorized licensed use limited to: Northeastern University. Downloaded on April 17,2024 at 01:08:08 UTC from IEEE Xplore. Restrictions apply.

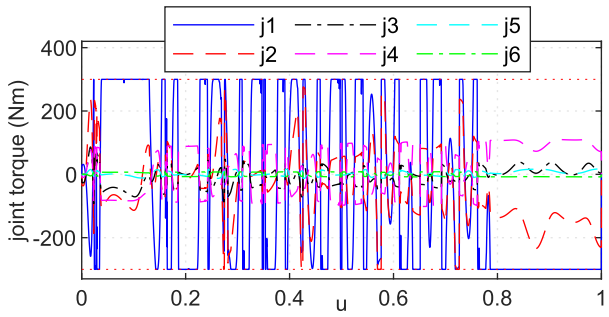


Fig. 12. The joints' torques for the robot motion in Fig. 10.

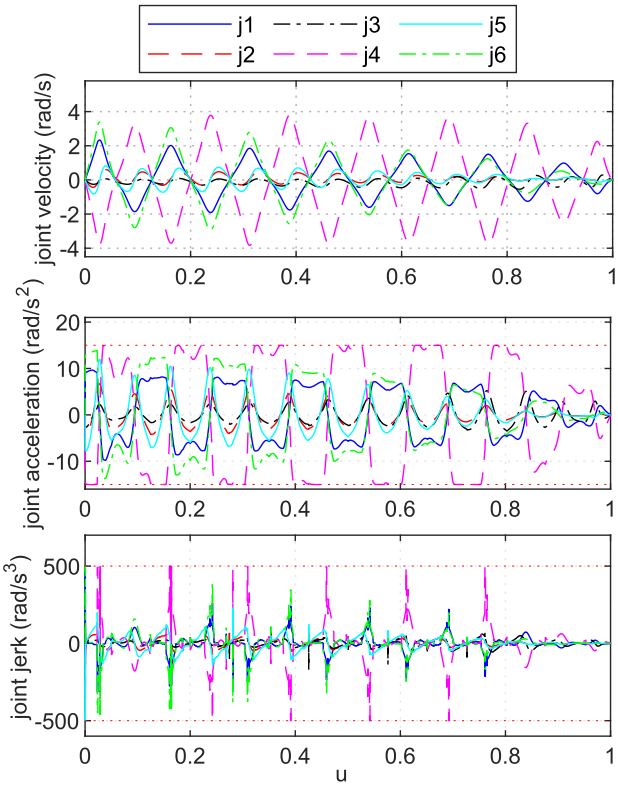


Fig. 13. Joints' velocities, accelerations, and jerks for the robot motion of Fig. 10.

orientation at path parameter $u$ is given below

$$O(u) = \frac{S_f(u) - S_t(u)}{||S_f(u) - S_t(u)||} \tag{36}$$

Finally, the sampling points can be converted from Cartesian space to joint space through inverse kinematics.

Fig. 12 presents the optimized torques of the joints, and Fig. 13 shows the optimized velocities, accelerations, and jerks, respectively. There are always variables running in a saturated state at any time, which meets the time-optimal requirements. Meanwhile, the joints' velocities and accelerations are smooth and continuous, satisfying the robot's motion requirements.

## VI. Conclusion

In this letter, the time-optimal trajectory generation is formulated as a convex optimization problem, and the objective function and constraints are linearized to improve computational efficiency. Considering the jerk constraints, this paper proposes transforming the complex nonlinear optimization problem into two sub-LP problems. In this approach, the computational efficiency is improved, and the requirement of a large number of sampling points is reduced. By taking a writing task profile and a complex edge of an impeller as examples, the planning time, machining time, and chord error between the trajectories without and with jerk constraints are compared. The results indicate that constraining the jerk enhances the smoothness of velocity and acceleration. Also, our method leads to higher computational efficiency than other methods.

## References

[1] Y. Chen, W. Dong, and Y. Ding, "An efficient method for collision-free and jerk-constrained trajectory generation with sparse desired way-points for a flying robot," *Sci. China Technological Sci.*, vol. 64, no. 8, pp. 1719–1731, 2021.

[2] R. Wang, Y. Xie, X. Chen, and Y. Li, "Path-constrained time-optimal motion planning for robot manipulators with third-order constraints," *IEEE/ASME Trans. Mechatron.*, vol. 28, no. 6, pp. 3005–3016, Dec. 2023.

[3] M. Oberherber, H. Gattringer, and A. Müller, "Successive dynamic programming and subsequent spline optimization for smooth time optimal robot path tracking," *Mech. Sci.*, vol. 6, no. 2, pp. 245–254, 2015.

[4] H. Pham and Q.-C. Pham, "A new approach to time-optimal path parameterization based on reachability analysis," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 645–659, Jun. 2018.

[5] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2318–2327, Oct. 2009.

[6] T. Lipp and S. Boyd, "Minimum-time speed optimisation over a fixed path," *Int. J. Control*, vol. 87, no. 6, pp. 1297–1311, 2014.

[7] K. Hauser, "Fast interpolation and time-optimization with contact," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1231–1250, 2014.

[8] Á. Nagy and I. Vajk, "LP-based velocity profile generation for robotic manipulators," *Int. J. Control*, vol. 91, no. 3, pp. 582–592, 2018.

[9] L. Consolini, M. Locatelli, A. Minari, A. Nagy, and I. Vajk, "Optimal time-complexity speed planning for robot manipulators," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 790–797, Jun. 2019.

[10] H. Liu, Q. Liu, P. SUn, Q. Liu, and S. Yuan, "The optimal feedrate planning on five-axis parametric tool path with geometric and kinematic constraints for CNC machine tools," *Int. J. Prod. Res.*, vol. 55, no. 13, pp. 3715–3731, 2017.

[11] A. Palleschi, M. Garabini, D. Caporale, and L. Pallottino, "Time-optimal path tracking for jerk controlled robots," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3932–3939, Oct. 2019.

[12] A. Palleschi et al., "Time-optimal trajectory planning for flexible joint robots," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 938–945, Apr. 2020.

[13] D. Kaserer, H. Gattringer, and A. Müller, "Nearly optimal path following with jerk and torque rate limits using dynamic programming," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 521–528, Apr. 2019.

[14] J. Xiao, S. Liu, H. Liu, M. Wang, G. Li, and Y. Wang, "A jerk-limited heuristic feedrate scheduling method based on particle swarm optimization for a 5-DOF hybrid robot," *Robot. Comput.- Integr. Manuf.*, vol. 78, 2022, Art. no. 102396.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[16] X. Du, J. Huang, L.-M. Zhu, and H. Ding, "Third-order chord error estimation for freeform contour in computer-aided manufacturing and computer numerical control systems," *Proc. Inst. Mech. Engineers Part B, J. Eng. Manufacture*, vol. 233, no. 3, pp. 863–874, 2019.

[17] L. Fang, G. Liu, Q. Li, and H. Zhang, "A high-precision non-uniform rational b-spline interpolator based on S-shaped feedrate scheduling," *Int. J. Adv. Manuf. Technol.*, vol. 121, no. 3/4, pp. 2585–2595, 2022.

[18] G. Liu, Q. Li, L. Fang, B. Han, and H. Zhang, "A new joint friction model for parameter identification and sensor-less hand guiding in industrial robots," *Ind. Robot*, vol. 47, no. 6, pp. 847–857, 2020.

[19] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2004, pp. 284–289.

[20] H. Zhao, L. Zhu, and H. Ding, "A parametric interpolator with minimal feed fluctuation for CNC machine tools using ARC-length compensation and feedback correction," *Int. J. Mach. Tools Manufacture*, vol. 75, pp. 1–8, 2013.

Authorized licensed use limited to: Northeastern University. Downloaded on April 17,2024 at 01:08:08 UTC from IEEE Xplore. Restrictions apply.