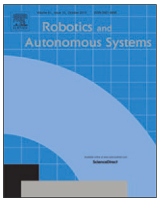
Robotics and Autonomous Systems 140 (2021) 103744

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



A new approach to time-optimal trajectory planning with torque and jerk limits for robot

Jian-wei Ma [*], Song Gao, Hui-teng Yan, Qi Lv, Guo-qing Hu

Key Laboratory for Precision and Non-traditional Machining Technology of the Ministry of Education, School of Mechanical Engineering, Dalian University of Technology, Dalian, China

ARTICLE INFO

Article history:
Received 4 December 2020
Received in revised form 29 January 2021
Accepted 10 February 2021
Available online 12 February 2021

Keywords:
Robotic system
Time-optimal trajectory planning
Convex optimization
Torque limits
Jerk limits

ABSTRACT

In this study, a new convex optimization (CO) approach to time-optimal trajectory planning (TOTP) is described, which considers both torque and jerk limits. The key insight of the approach is that the non-convex jerk limits are transformed to linear acceleration constraints and indirectly introduced into CO as the linear acceleration constraints. In this way, the convexity of CO will not be destroyed and the number of optimization variables will not increase, which give the approach a fast computation speed. The proposed approach is implemented on random geometric path of a 6-DOF manipulator. Compared with a similar method, the results show that the torque and jerk limits are addressed by a reasonable increase in the computation time. In addition, the maximum value of joint jerk reduces by over 80% and the joint torque curves are smoother in the comparison, which demonstrates that this approach has the ability to effectively restrain acceleration mutation.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Robot is nowadays widely used in industrial assembly lines and manufacturing systems, and the higher requirement for motion performance of the robotic system is also put forward. The time-optimal trajectory planning (TOTP) is a common requirement to improve the productivity of robotic system, which is to find the fastest way to traverse a predetermined path in the configuration space of a robotic system while respecting the physical constraints. The physical constraints of a robotic system usually include torque, jerk, acceleration and velocity limits, where the torque limits and the jerk limits are especially crucial to TOTP. Torque limits are constraints of joint torques which execute a certain motion of the robotic system. And jerk limits are of major importance to ensure that the trajectory is smooth enough to execute on the real robotic system. Thus, it is very important to correctly deal with TOTP with these physical constraints. However, many TOTP researches only consider torque limits without jerk limits, which will lead to acceleration mutation in trajectory and seriously affect motion stability.

In addition, TOTP problem has a wide range of applications in robotics, such as, cutting robot, welding robot and mobile robot, where the predetermined path needs to be executed at a high speed while satisfying the constraints. Therefore, it is particularly important to deal with TOTP properly for improving the motion performance of robotic system, which has gained much attention since its inception in the 1980s [1–4]. In past decades, a number of researchers have contributed to this problem, which can basically be divided into the following three groups.

Dynamic programming (DP): In [5], an DP algorithm to handle the optimal trajectory planning problem for long paths of manipulators is proposed. In [6], the torque constraints are introduced into the DP algorithm, but the number of optimization variables is also increased which results in an exponential increase in computation time. In a recent work [7], a new DP algorithm with torque rate and jerk limits is proposed, which uses a suitable interpolation in the phase plane to generate trajectories with continuous joint torque and acceleration constraints. However, this new DP algorithm can only obtain the suboptimal trajectories, and the computation time is high compared with other methods.

Numerical integration (NI): In [8], authors propose a method to generate the time-optimal trajectory for fixed path with bounded acceleration and velocity, but this method is complex to find the switch points, that is, the points where the change between the active actuator constraints can occur. In [9], the author proposes a time-optimal path parameterization algorithm to handle torque limits, which further improves the calculation method of finding the switch points. However, a major disadvantage of the above two methods is the neglect of jerk constraints, which results in the abrupt change of acceleration. In [10], a smooth TOTP algorithm is proposed, which is able to deal with torque and jerk limits. However, this method is not perfect for searching the switch points, which is not able to find out all switch points.

[*] Correspondence to: Key Laboratory for Precision and Non-traditional Machining Technology of the Ministry of Education, School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China.
E-mail address: mjw2011@dlut.edu.cn (J.-w. Ma).

https://doi.org/10.1016/j.robot.2021.103744
0921-8890/© 2021 Elsevier B.V. All rights reserved.

Convex optimization (CO): In [11,12], author utilizes cubic splines and particle swarm optimization to generate smooth trajectories of serial and parallel manipulators, which satisfies the constraints of velocity, acceleration and jerk. However, these algorithms do not take torque limits into account, which affect trajectory tracking accuracy and motion performance. On the basis of previous work, authors [13] propose a new method to TOTP based on CO, which can find out the switch points more quickly and easily. But this method does not consider the jerk limits. For instance, the authors [14] approximate jerk constraints by linear functions, so the smooth time-optimal trajectory is obtained. However, due to the increase in the number of optimization variables, the calculation time of this method increases greatly. In order to obtain time-jerk optimal trajectory, authors [15] use a non-dominated sorting genetic algorithm (NSGA-II) to optimize time and jerk at the same time, but they do not consider torque limits. In conclusion, CO is more difficult to deal with jerk limits, which will destroy the convexity of CO, increase the number of optimization variables and increase the computation time, if jerk limits are introduced into CO directly. However, compared with DP and NI, CO has a simple and stable way to find the switch points and has a fast computing speed with fewer optimization variables. As the results shown in [7], for the same TOTP problem, the DP method takes 8.5 times longer than the CO method. While in [13], for another TOTP problem, the NI method takes 15.3 times longer than the CO method. So far, there are very few studies based on CO to plan the time-optimal trajectory with torque and jerk limits.

Other methods: In addition to the three groups mentioned above, there are some other methods to the TOTP problem that are not part of the three groups. In [16], in order to generate smooth trajectory, authors utilize B-spline to interpolate trajectory of each joint and minimize the objective function containing the time and the integral of the squared jerk. Besides, the convex hull of B-spline is used to simplify the algorithm by making the control points satisfy the kinematic constraints. However, this algorithm does not consider torque limits. In [17], the properties of the roots multiplicity are utilized to build 9th order polynomial acceleration curves for manipulators, one limitation of this method is that only the acceleration limits can be predefined. In order to obtain a smooth and time-optimal S-curve trajectory, authors [18] utilize a piecewise sigmoid function to establish a jerk profile under the given constraints on velocity, acceleration and jerk. In [19], in order to minimize the traveling time, the energy and the distance of the end-effector, a constrained multi-objective particle swarm optimization algorithm is applied to plan the joint trajectory. Although higher order polynomials have better continuity, their oscillation characteristics tend to increase and their coefficients are difficult to calculated with a large number of high-order terms, which will affect the accuracy of the motion and the computational load of the system [20]. In addition, these methods do not take into account dynamic constraints, and the performance of the trajectories planned by these methods remain to be verified.

This study presents a new approach to TOTP with torque and jerk limits based on CO. The basic idea of this approach is to choose the acceleration to make the velocity as large as possible at every path point [1] (analysis and proof of the idea can be found in [1]). However, the velocity does not increase or decrease all the time, but changes frequently between acceleration and deceleration. In order to calculate the maximum velocity, it is crucial to locate the switch points respect to the constraints, at which the acceleration switches between maximum acceleration and maximum deceleration. Thus, it is the first step of this approach to locate the switch points. In this study, the method in [13] is utilized to search the switch points, in which forward maximum velocity curve, backward maximum velocity curve and feasible velocity profile are calculated respectively (analysis and proof of the method can be found in [13]). Since the formulation of the jerk limits is non-convex (see [21]), an algorithm is proposed to transform the jerk limits into acceleration constraints that are linear constraints in optimization problems. Based on the linear acceleration constraints and the torque limits, the convex optimization problems are formulated at each path point in 2D phase plane. By solving these convex optimization problems per path point in forward and backward processes, the time-optimal trajectory with torque and jerk limits is obtained. Compared to a similar TOTP method, the proposed approach is able to restrain the acceleration mutation effectively by constraining the maximum value of jerk. In addition, since the jerk limits are indirectly introduced into CO, the number of optimization variables do not increase, which bounds the cost of the proposed approach at a low computation. Thus, the results of this study provide a new method for the scientific community to deal with non-convex jerk constraints, when one calculates smooth optimal trajectory of robots based on convex optimization.

The rest of this paper is organized as follows. Section 2 formulates TOTP in a general setting. Section 3 presents the proposed approach to planning time-optimal trajectory with torque and jerk limits. Section 4 reports the extensive experimental results to demonstrate the gains in robustness and performance permitted by the new approach. Finally, Section 5 offers some concluding remarks and directions for future research.

2. Problem formulation

In this section, the formulation of TOTP problem is firstly defined here. Consider a n-DOF robotic system, and the joint coordinate vector is denoted by a n-dimensional vector $\mathbf{q}$. The equations of motion that describe the behavior of the robotic system can be formulated as

$$\mathbf{M}\left(\mathbf{q}\right)\ddot{\mathbf{q}}+\dot{\mathbf{q}}^{T}\mathbf{C}\left(\mathbf{q}\right)\dot{\mathbf{q}}+\mathbf{g}\left(\mathbf{q}\right)=\boldsymbol{\tau}. \tag{1}$$

In which, $\mathbf{M}\left(\mathbf{q}\right)$ denotes the inertia matrix, $\mathbf{C}\left(\mathbf{q}\right)$ denotes the Coriolis and centrifugal terms, and $\mathbf{g}\left(\mathbf{q}\right)$ stands for the gravitational terms and Coulomb friction terms [22–24]. $\ddot{\mathbf{q}}$ and $\dot{\mathbf{q}}$ are second order and first order derivatives of $\mathbf{q}$ with respect to time, which denote joint acceleration and joint velocity, and $\boldsymbol{\tau}$ denotes the joint torque.

The geometric parameters of the robot system used in this study are shown in Table 1. It is important for robotic system, like the one shown in Fig. 1 and Table 1, to plan velocity and acceleration of the predetermined motion path before executing. However, for a predetermined path, the velocity of the path is not yet clear, and the traversing time is also not clear. Thus, the derivatives of $\mathbf{q}$ with respect to time cannot be calculated. In order to solve this problem, the vector $\mathbf{q}$ can be considered as a function $\mathbf{q}\left(s\right)$ with the path parameter s. Then the derivatives of $\mathbf{q}$ with respect to time can be instead of the derivatives with respect to the path parameter s, as

$$\begin{cases}\dot{\mathbf{q}}=\mathbf{q}'\dot{s}\\ \ddot{\mathbf{q}}=\mathbf{q}''\dot{s}^{2}+\mathbf{q}'\ddot{s}\end{cases} \tag{2}$$

where $s,\dot{s},\ddot{s}$ denote the position, velocity, and acceleration of the path, respectively. In the rest of this study, $()'$ denotes the derivative with respect to the path parameter s, while $\dot{()}$ denotes the derivative with respect to time.

Substituting Eq. (2) into Eq. (1), the motion equations for the robotic system with respect to $\mathbf{q}$ can be transformed as that with respect to s, just as

$$\mathbf{a}\left(s\right)\ddot{s}+\mathbf{b}\left(s\right)\dot{s}^{2}+\mathbf{c}\left(s\right)=\boldsymbol{\tau}\left(s\right) \tag{3}$$

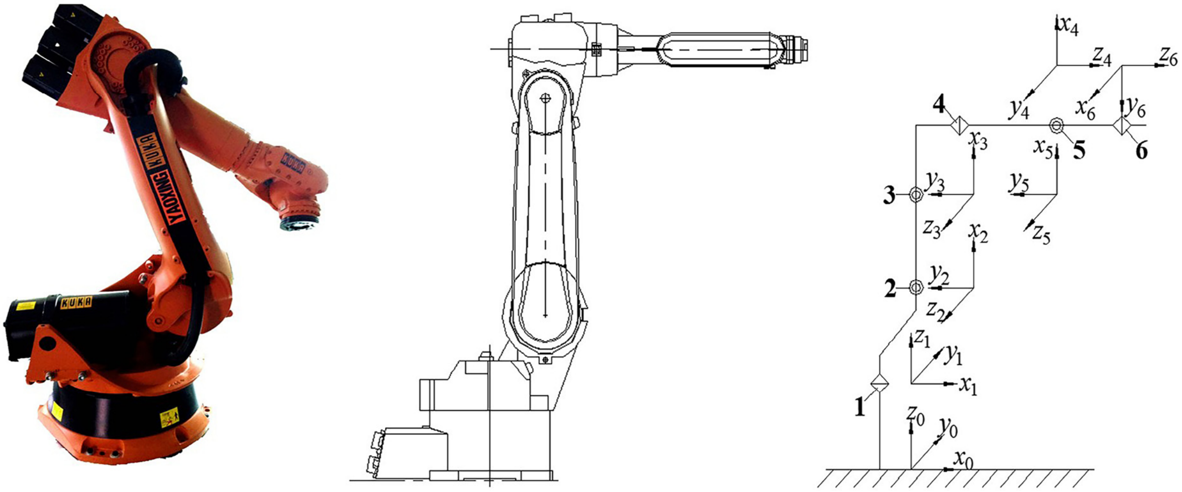


Fig. 1. A typical kind of robotic system with 6-DOF, which is usually called industrial manipulator. The motion equations of industrial manipulator can be described as Eq. (1). The labels represent the joints and the local coordinates of industrial manipulator.

Table 1
MDH parameters of the 6-DOF manipulator used in this study.

	Joint angle θ (rad)	Link offset d (mm)	Link length a (mm)	Twist angle α (rad)
Joint1	θ_1	0	0	0
Joint2	$\theta_2(\pi/2)$	0	160	$\pi/2$
Joint3	θ_3	0	790	0
Joint4	θ_4	795	155	$\pi/2$
Joint5	θ_5	0	0	$-\pi/2$
Joint6	$\theta_6(\pi/2)$	145	0	$\pi/2$

where $\mathbf{a}(s) = \mathbf{M}(\mathbf{q}(s))\mathbf{q}'(s)$, $\mathbf{b}(s) = \mathbf{M}(\mathbf{q}(s))\mathbf{q}''(s) + \mathbf{q}'(s)^T \mathbf{C}(\mathbf{q}(s))\mathbf{q}'(s)$, $\mathbf{c}(s) = \mathbf{g}(s)$, $\boldsymbol{\tau}(s) = \boldsymbol{\tau}(\mathbf{q}(s))$. In order to easily design the convex optimization problems, Eq. (3) is further simplified as

$$\mathbf{a}(s)\,u + \mathbf{b}(s)\,x + \mathbf{c}(s) = \boldsymbol{\tau}(s) \tag{4}$$

where $u = \ddot{s}$ denotes the acceleration of the path parameter s, $x = \dot{s}^2$ denotes the squared velocity of the path parameter s. In the rest of this paper, the velocity mentioned means squared velocity, except joint velocity.

The proposed approach discretizes the predetermined path into N path points and assumes that the motion between adjacent path points is uniformly accelerated (or decelerated). Based on this hypothesis, it should be noted that

$$\frac{d\dot{s}^2}{ds} = 2\dot{s}\frac{d\dot{s}}{dt}\frac{dt}{ds} = 2\ddot{s} = 2u \tag{5}$$

Then, the relationship of velocity between adjacent path points can be got as

$$x_{i+1} = x_i + 2\Delta_i u_i, \quad i = 1, \ldots, N-1 \tag{6}$$

where $\Delta_i = s_{i+1} - s_i$ is the interval between adjacent path points. According to Eq. (6), if the acceleration can be determined at all path points, the velocity curve can be recursively calculated. Similarly, in order to obtain the relationship of acceleration between adjacent path points, it should be noted that

$$\frac{d\ddot{s}}{ds} = \frac{d\ddot{s}}{dt}\frac{dt}{ds} = \frac{\dddot{s}}{\dot{s}} = \frac{\dot{u}}{\sqrt{x}} \tag{7}$$

In which, $\dot{u} = \dddot{s}$ denotes the jerk of the path parameter s. Then, the acceleration curve can also be recursively calculated as

$$u_{i+1} = u_i + \frac{2\Delta_i \dot{u}_i}{\sqrt{x_i + 2\Delta_i u_i} + \sqrt{x_i}}, \quad i = 1, \ldots, N-1 \tag{8}$$

So far, the motion equations of robotic system are transformed to Eq. (4), where the torque limits can be introduced into CO as linear constraints. Therefore, the TOTP task in this study can be generally formulated as

Minimize:

$$T = \int_{t_{beg}}^{t_{end}} dt = \int_{s_{beg}}^{s_{end}} \frac{ds}{\sqrt{x}} = \sum_{i=1}^{N} \frac{ds}{\sqrt{x_i}} \tag{9}$$

Subject to:

$$\begin{cases} X_{\min} \le x_i \le X_{\max} \\ U_{\min} \le u_i \le U_{\max} \\ \boldsymbol{\tau}_{\min} \le \mathbf{a}_i u_i + \mathbf{b}_i x_i + \mathbf{c}_i \le \boldsymbol{\tau}_{\max} \\ X_{\min} \le x_{i+1} = x_i + 2\Delta_i u_i \le X_{\max} \\ u_{i-1} + \dfrac{2\Delta_{i-1}\dot{U}_{\min}}{\sqrt{x_{i-1} + 2\Delta_{i-1}u_{i-1}} + \sqrt{x_{i-1}}} \le u_i \\ \le u_{i-1} + \dfrac{2\Delta_{i-1}\dot{U}_{\max}}{\sqrt{x_{i-1} + 2\Delta_{i-1}u_{i-1}} + \sqrt{x_{i-1}}} \end{cases},$$
$$i = 2, \ldots, N-1 \tag{10}$$

where x and u are the optimization variables which is same with other optimization problem in this paper. T denotes the total traversing time of the predetermined path, $X_{\min}$ and $X_{\max}$ denote the bounds of velocity, $U_{\min}$ and $U_{\max}$ denote the bounds of acceleration, $\dot{U}_{\min}$ and $\dot{U}_{\max}$ denote the bounds of jerk, $\boldsymbol{\tau}_{\min}$ and $\boldsymbol{\tau}_{\max}$ denote the bounds of torque. In this approach, torque and jerk limits do not change along the path, while the bounds of acceleration constraints vary along the path which is different from the work in [13]. In addition, in order to maintain the convexity of CO and the number of optimization variables, the jerk limits are indirectly introduced into CO as the last constraint in Eq. (10). Therefore, there is no abrupt change of acceleration beyond jerk limits in the trajectory. Note that, the optimization problem in Eq. (9) is not solved as a whole problem, while it is divided into many specific optimization problems, where the maximum velocity is calculated at each path point. In the next section, how to construct optimization problems at each path point will be introduced.

3. TOTP with torque and jerk limits

In this section, the switch points are searched on the feasible velocity profile which is determined by the forward maximum velocity curve and the backward maximum velocity curve. For maintaining the convexity of CO and the number of optimization variables, the jerk limits are transformed into linear acceleration constraints and indirectly introduced into optimization problems as the linear acceleration constraints. At last, the time-optimal trajectory with torque and jerk limits is obtained based on the torque limits and the linear acceleration constraints. The flowchart of the whole proposed approach is illustrated in Fig. 2.

3.1. Switch points

In order to make starting velocity and ending velocity of the feasible velocity profile respectively equal to the predetermined starting velocity and ending velocity of the path, the maximum velocity curves are calculated twice in forward and backward processes. In the first process of searching switch points, the maximum velocity curve is calculated forward at each path point and named as the forward maximum velocity curve **F**. In the second process, another maximum velocity curve is calculated backward at each path point and named as the backward maximum velocity curve **B**. Then, the feasible velocity profile where switch points lie on is determined according to **F** and **B**.

Since the starting velocity of the path is predetermined, the forward maximum velocity curve **F** can be recursively calculated from the starting velocity as Eq. (6). For recursively calculating **F**, the velocity and the acceleration per path point respecting to velocity limits and torque limits need to be determined by optimization. Therefore, the forward maximum velocity curve **F** can be obtained by recursively forward solving formulas as follow

$$\begin{aligned} \max \quad & x_i + 2\Delta_i u_i \\ s.t. \quad & 0 \le x_i \le F_i \\ & \boldsymbol{\tau}_{\min} \le \mathbf{a}_i u_i + \mathbf{b}_i x_i + \mathbf{c}_i \le \boldsymbol{\tau}_{\max} \\ & X_{\min} \le x_i + 2\Delta_i u_i \le X_{\max} \end{aligned} \tag{11}$$

$$F_{i+1} = x_i + 2\Delta_i u_i, \quad i = 1, 2, \dots, N-1 \tag{12}$$

where F_{i+1} is the $i+1$th member of the forward maximum velocity curve **F**, which is calculated based on the optimization results at path point s_i. The objective function in Eq. (11) is to find x_i and u_i which can maximize the velocity x_{i+1} at path point s_{i+1}, which is set based on Eq. (6). Since the constraints in Eq. (11) are linear, the feasible region of constrains is a convex polytope on $\left\{\ddot{s}, \dot{s}^2\right\}$ ($\{u, x\}$) phase plane. By recursively solving optimization problems from path point s_1 to s_{N-1}, the results F_{i+1} form the forward maximum velocity curve $\mathbf{F} = \left\{x_{beg}, F_2, \dots, F_N\right\}$. Note that, the first member of **F**, which is predetermined, is the starting velocity of the path.

Similar with forward process, the backward maximum velocity curve **B** is also recursively calculated from the ending velocity of the path. The difference is that the formulas are slightly different from the ones of forward process because of the change of calculation order. Therefore, the backward maximum velocity curve **B** can be obtained by recursively backward solving formulas as follow

$$\begin{aligned} \max \quad & x_i \\ s.t. \quad & X_{\min} \le x_i \le X_{\max} \\ & \boldsymbol{\tau}_{\min} \le \mathbf{a}_i u_i + \mathbf{b}_i x_i + \mathbf{c}_i \le \boldsymbol{\tau}_{\max} \\ & 0 \le x_i + 2\Delta_i u_i \le B_{i+1} \end{aligned} \tag{13}$$

$$B_i = x_i, \quad i = N-1, \dots, 2, 1 \tag{14}$$

where B_i is the ith member of the backward maximum velocity curve **B**, which is equal to the optimization result x_i at path point s_i. The purpose of the optimization problem in Eq. (13) is to find maximum velocity x_i and corresponding acceleration u_i, which can make the velocity x_{i+1} satisfy the constraints at path point s_{i+1}. The feasible region of constrains in Eq. (13) is also a convex polytope on phase plane. By recursively solving optimization problems from path point s_{N-1} to s_1, the results B_i form the backward maximum velocity curve $\mathbf{B} = \{B_1, \dots, B_{N-1}, x_{end}\}$. Note that, the last member of **B**, which is predetermined, is the ending velocity of the path.

At each path point, the smaller value of the maximum velocity curves, **F** and **B**, is selected to determine the feasible velocity profile shown in Fig. 3, where the extreme points of the feasible velocity profile are taken as the switch points of the path. And the number of switch points is determined by the shape of the feasible velocity profile, which is related to the path executed by robot and the constraints of velocity, acceleration, jerk, torque.

3.2. Linear acceleration constraints

In [7] and [10], the jerk limits are directly introduced into algorithms by the third-order constraints equations with torque rate constraints. In this approach, if jerk limits are directly introduced into CO in the same way, the convexity of CO will be destroyed. What is worse, the optimization problems will transform from linear programming in 2D phase plane $\{u, x\}$ to nonlinear programming in 3D phase space $\{\dot{u}, u, x\}$, which will make the computation time increase greatly.

In order to avoid above disadvantages, the jerk limits are transformed into linear acceleration constraints which vary along the path and indirectly introduced into CO as the linear acceleration constraints. The linearity means that the acceleration constraints are linear in the optimization problems, while it does not mean the profile of the acceleration constraints is linear. For piecewise calculating the linear acceleration constraints, the feasible velocity profile is segmented by switch points in Fig. 3. The abscissa in Fig. 3 denotes the predetermined path, which is also segmented by switch points.

Since the switch points locate between accelerating and decelerating segments, the accelerations at each switch point should be set as zero. In order to recursively calculate linear acceleration constraints, the maximum velocities at each switch point with zero acceleration need to be calculated as

$$\begin{aligned} \max \quad & x_i \\ s.t. \quad & 0 \le x_i \le F_i \\ & 0 \le x_i \le B_i \\ & 0 \le u_i \le 0 \\ & \boldsymbol{\tau}_{\min} \le \mathbf{a}_i u_i + \mathbf{b}_i x_i + \mathbf{c}_i \le \boldsymbol{\tau}_{\max} \end{aligned} \tag{15}$$

When recursively calculating linear acceleration constraints, the acceleration curve of smooth S-shaped velocity curve is desired, which can improve the performance on motion. For obtaining smooth S-shaped velocity curves in every segment, the linear

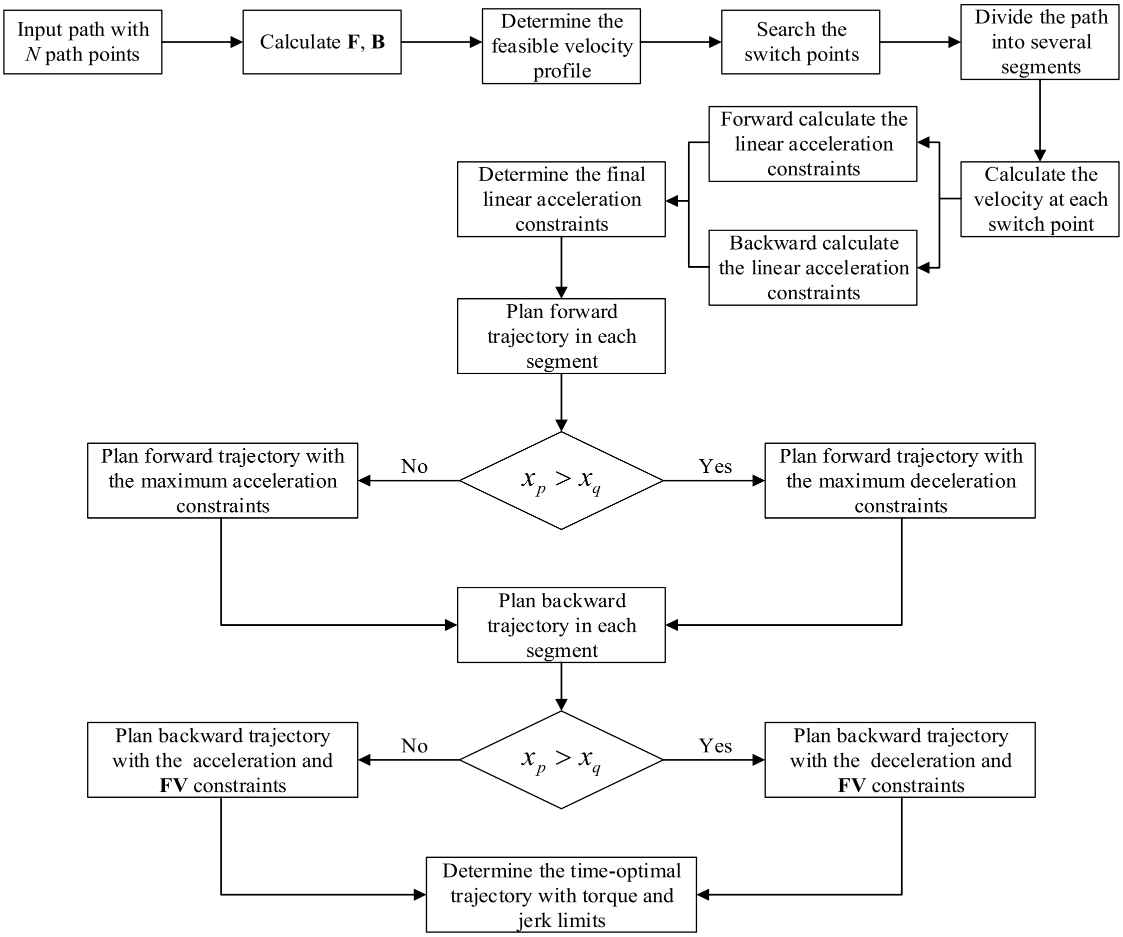

Input path with N path points
Calculate **F**, **B**
Determine the feasible velocity profile
Search the switch points
Divide the path into several segments
Calculate the velocity at each switch point
Forward calculate the linear acceleration constraints
Backward calculate the linear acceleration constraints
Determine the final linear acceleration constraints
Plan forward trajectory in each segment
$x_p > x_q$
No
Yes
Plan forward trajectory with the maximum acceleration constraints
Plan forward trajectory with the maximum deceleration constraints
Plan backward trajectory in each segment
$x_p > x_q$
No
Yes
Plan backward trajectory with the acceleration and **FV** constraints
Plan backward trajectory with the deceleration and **FV** constraints
Determine the time-optimal trajectory with torque and jerk limits

Fig. 2. Flowchart of planning time-optimal trajectory with torque and jerk limits.

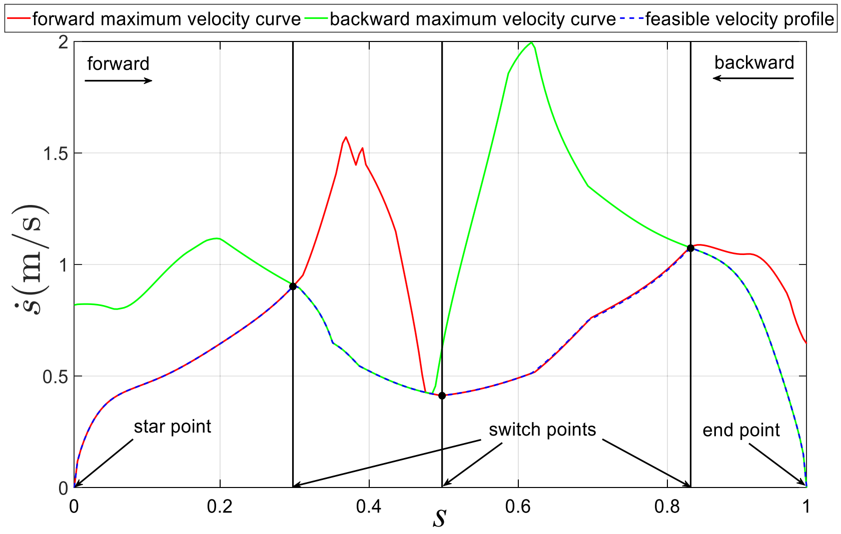


Fig. 3. Plot of the feasible velocity profile, the forward maximum velocity curve **F** and the backward maximum velocity curve **B**. The black points are the extreme points of the feasible velocity profile. The black lines located at the positions of switch points divide the profile and the path into several segments.

acceleration constraints are piecewise calculated in forward and backward processes. In the forward process, starting from the start point or switch points, according to Eq. (16), the piecewise acceleration constraints are calculated recursively, until reach the end point or switch points. The results are dashed lines shown in Fig. 4.

$$\begin{cases} u_{i+1,\max/\min} = u_i + \dfrac{2\Delta_i \dot{U}_{\max/\min}}{\sqrt{x_{i,\max/\min} + 2\Delta_i u_{i,\max/\min}} + \sqrt{x_{i,\max/\min}}}, \\ x_{i+1,\max/\min} = x_{i,\max/\min} + 2\Delta_i u_{i,\max/\min} \end{cases}$$

$$i = p, p+1, \ldots, q-1 \qquad (16)$$

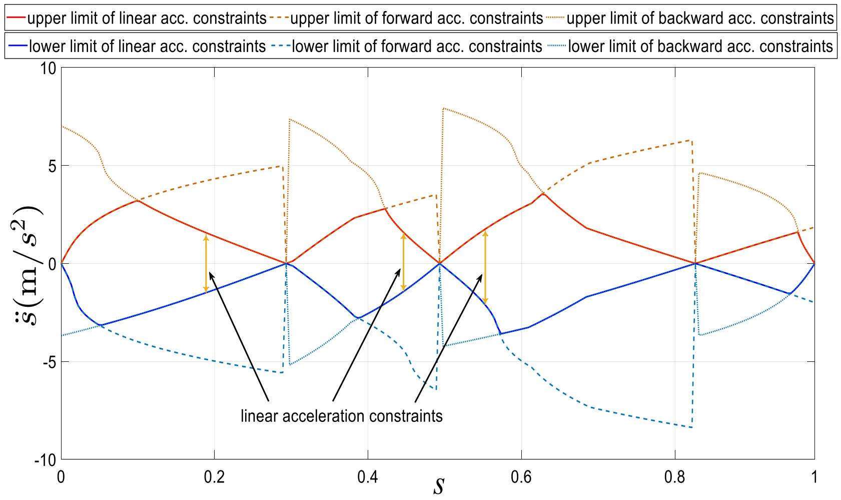


Fig. 4. Plot of linear acceleration constraints. The dashed lines indicate piecewise acceleration constraints calculated forward. The dotted lines indicate piecewise acceleration constraints calculated backward. The solid lines indicate the final linear acceleration constraints. The red lines are upper limit, while the blue lines are lower limit. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

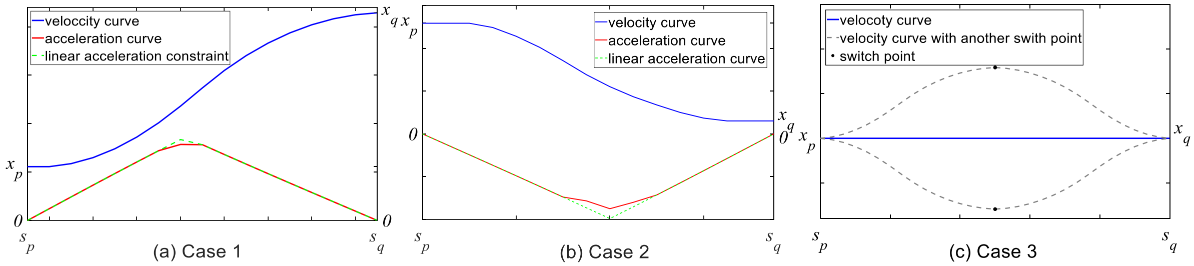


Fig. 5. Schematic diagrams of the above 3 cases. The blue lines denote the velocity curve **FV**. The red lines denote the acceleration curve, which consists of u_i per path point. The green lines denote the linear acceleration constraints, which is upper limit in (a) and is lower limit in (b). The gray lines denote velocity curve with another switch point. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where $u_{i,\max/\min}$ denotes the bounds of acceleration at ith path point s_i. $\dot{U}_{\max/\min}$ denotes the fixed bounds of jerk limits, which do not change along the path. p and q ($p < q$) denote the serial number of the start point and the end point of each segment.

In the backward process, starting from the end point or switch points, according to Eq. (17), the piecewise acceleration constraints are calculated recursively, until reach the start point or switch points. The results are dotted lines shown in Fig. 4.

$$\begin{cases} u_{i-1,\max/\min} = u_i + \dfrac{2\Delta_i \dot{U}_{\max/\min}}{\sqrt{x_{i,\max/\min} - 2\Delta_i u_{i,\max/\min}} + \sqrt{x_{i,\max/\min}}} \\ x_{i-1,\max/\min} = x_{i,\max/\min} - 2\Delta_i u_{i,\max/\min} \end{cases}, \quad i = q, q-1, \ldots, p+1 \tag{17}$$

In each segment, there are two sets of acceleration constraints which are calculated in the forward and backward processes respectively, and the intersection of two sets is taken as the final linear acceleration constraints as shown in Fig. 4.

3.3. TOTP with torque and linear acceleration limits

As the path is segmented in Section 3.2, the time-optimal trajectory with torque and jerk limits is planed piecewise in forward and backward processes. Then the time series can be calculated based on velocity curve of time-optimal trajectory. At last, 5th-order spline is used to construct the joint trajectories based on time series and joint angle series.

3.3.1. Forward trajectory

The trajectory in this section is determined by recursively calculating the maximum velocities at each path point respecting to velocity, acceleration, torque and jerk limits. In Section 3.2, the velocities at switch points are calculated by Eq. (15), and the starting velocity and the ending velocity of the path are predetermined. Thus, the starting velocity and the ending velocity of each segment are known. When piecewise planning the forward trajectory, the starting velocity and the ending velocity of the segment should be compared first. And there exist three cases as follow.

Case 1: If the starting velocity of the segment x_p is less than the ending velocity x_q, then the velocity in this segment is increased, like Fig. 5(a). In this segment, the velocity and acceleration of forward trajectory can be determined by recursively forward solving formulas as follow

$$\begin{aligned} \max \quad & x_i + 2\Delta_i u_i \\ s.t. \quad & FV_i \le x_i \le FV_i \\ & \max(u_{i,lb}, 0) \le u_i \le \min(u_{i,ub}, u_{i,\max}) \\ & \boldsymbol{\tau}_{\min} \le \mathbf{a}_i u_i + \mathbf{b}_i x_i + \mathbf{c}_i \le \boldsymbol{\tau}_{\max} \\ & 0 \le x_i + 2\Delta_i u_i \le B_{i+1} \end{aligned} \tag{18}$$

$$FV_{i+1} = x_i + 2\Delta_i u_i, \quad i = p, p+1, \ldots, q-1 \tag{19}$$

where $u_{i,\max}$ is the upper bound of the linear acceleration constraint at path point s_i which is determined in Section 3.2. $u_{i+1,lb} = u_i + \frac{2\Delta_i \dot{U}_{\min}}{\sqrt{x_i+2\Delta_i u_i}+\sqrt{x_i}}$ and $u_{i+1,ub} = u_i + \frac{2\Delta \dot{U}_{\max}}{\sqrt{x_i+2\Delta_i u_i}+\sqrt{x_i}}$

denote the bounds of acceleration at next path point s_{i+1}, which are the maximum change of acceleration between s_i and s_{i+1}. Obviously, $\max(u_{i,lb}, 0) \leq u_i$ ensures that the velocity in this segment must be increased, and $u_i \leq \min(u_{i,ub}, u_{i,\max})$ ensures that the change of acceleration between adjacent path points does not violate the jerk limits. Note that, in the optimization problems at each path point, the upper and lower bounds of the velocity constraints are equal ($FV_i \leq x_i \leq FV_i$). In another word, the velocity at next path point s_{i+1} is uniquely specified by Eq. (19) based on the optimization results at path point s_i. Therefore, these optimization problems avoid performing 2D projection, instead they will only require a few 1D projections per path point which can be performed extremely quickly. The velocity constraints at start point of the segment are predetermined (start point of path) or specified based on the optimization results in the previous segment (end point of previous segment). By solving Eqs. (18) and (19) at each path point recursively, the forward trajectory on the segment with increased velocity can be obtained.

Case 2: If the starting velocity of the segment x_p is greater than the ending velocity x_q, then the velocity in this segment is decreased, like Fig. 5(b). Similarly, the velocity and acceleration of forward trajectory in this segment can be determined by recursively forward solving formulas as follow

$$\begin{aligned} \max \quad & x_i + 2\Delta u_i \\ s.t. \quad & FV_i \leq x_i \leq FV_i \\ & \max(u_{i,lb}, u_{i,\min}) \leq u_i \leq \min(u_{i,ub}, 0) \\ & \boldsymbol{\tau}_{\min} \leq \mathbf{a}_i u_i + \mathbf{b}_i x_i + \mathbf{c}_i \leq \boldsymbol{\tau}_{\max} \\ & 0 \leq x_i + 2\Delta u_i \leq B_{i+1} \end{aligned} \tag{20}$$

$$FV_{i+1} = x_i + 2\Delta_i u_i, \quad i = p, p+1, \ldots, q-1 \tag{21}$$

where $u_{i,\min}$ is the lower bound of the linear acceleration constraint at path point s_i. In these optimization problems, the velocity at next path point s_{i+1} is also uniquely specified based on the optimization results at path point s_i. Similarly, $u_i \leq \min(u_{i,ub}, 0)$ ensures that the velocity in this segment must be decreased, and $\max(u_{i,lb}, u_{i,\min}) \leq u_i$ ensures that the change of acceleration between adjacent path points does not violate the jerk limits. By solving Eqs. (20) and (21) at each path point recursively, the forward trajectory on the segment with decreased velocity can be obtained.

Case 3: If the starting velocity of the segment x_p is equal to the ending velocity x_q, then the velocity in this segment does not change, like Fig. 5(c). Otherwise, there must be another switch point in this segment.

At last, the optimization results in each segment form a complete forward trajectory, wherein the velocity curve is recorded as $\mathbf{FV} = \left\{FV_1 = x_{beg}, FV_2, \ldots, FV_N\right\}$.

3.3.2. Backward time-optimal trajectory

The backward time-optimal trajectory is planned in the same way presented in Section 3.3.1. The backward process can be regarded as another forward process where the calculation order is changed. In addition, B_{i+1}, the upper bound of velocity at the next path point, needs to be replaced by FV_{i+1} in Eqs. (18) and (20). At last, the optimization results in each segment form the complete time-optimal trajectory with torque and jerk limits.

Time series consists of the timestamps of all path points when the trajectory is executed to the corresponding one, in which the first element and the last element denote the start time and end time of the trajectory respectively. Since the trajectory is a function respect to time, the timestamp of each path point needs to be calculated for determining the trajectory. And the time series can be calculated according to Eq. (9) based on the velocity curve of time-optimal trajectory. In order to achieve the joint jerk continuity, 5th-order spline is used to construct the joint trajectories based on time series and joint angle series.

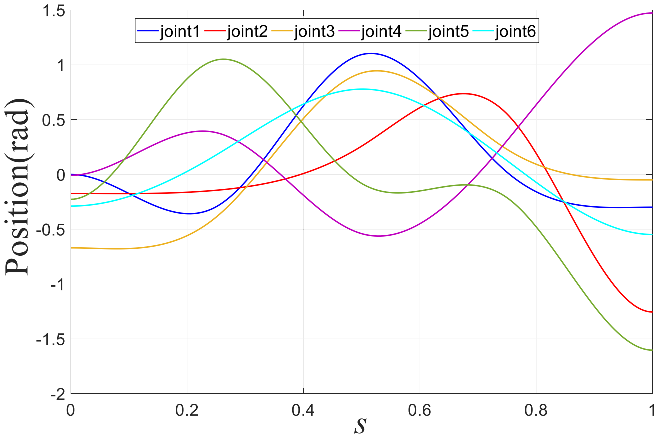


Fig. 6. The random geometric paths which this approach is implemented and test on.

4. Simulation results

In this section, the feasibility and the performance of the proposed approach will be demonstrated through several concrete and general examples. The proposed approach is performed in Python on a running Ubuntu with an Intel i5-7500 3.4-GHz CPU and 8-GB RAM. All optimization problems are solved with qpOASES [25]. This approach is implemented and test on random geometric path shown in Fig. 6 subjecting to constraints on velocity, acceleration, torque and jerk. Joint velocity, acceleration and torque limits are shown in Table 2. At the same time, the simulation results are compared with the results from the similar CO algorithm (TOPP-RA) proposed in [11] to verify the performance of this approach on restraining acceleration mutation which is caused by neglecting jerk limits.

In general, joint jerk constraints and path jerk constraints have the same effect on restraining acceleration mutation. In the approach proposed in this study, the jerk limits are set as path jerk constraints $\dddot{s}_{\max/\min} = \dot{U}_{\max/\min} = \pm 10$ m/s^3, which can effectively restrain acceleration mutation. However, there is a mapping relationship between the path jerk constraints and the joint jerk constraints. The exploration of this mapping relationship is left for future investigations.

Table 3 reports the different components of the computation time. Compared with TOPP-RA, the total computing time of the proposed approach is increased reasonably due to the addition of two new items. However, the jerk limits are also indirectly introduced into CO through these two items, which are able to make the performance of the approach better on restraining acceleration mutation. Even though the computation time of the proposed approach increase compared with TOPP-RA, the computation time still much less than the other methods based on NI and DP (see [13] and [7]). In addition, the computation time can be significantly reduced by using a better CPU (For more detail, see [13]).

Figs. 7 and 8 show the resulting trajectories from the two methods in the phase planes. Since there is no constraint on jerk in TOPP-RA, the acceleration between adjacent path points can change drastically and the slope of acceleration can be great, which is shown obviously in Fig. 8. In Fig. 7, the drastic change in acceleration is represented by the fact that there is no smooth

Table 2
Joint velocity, acceleration and torque limits for the 6-DOF manipulator.

Limits	Joint1	Joint2	Joint3	Joint4	Joint5	Joint6
Vel. (rad/ s)	3.92	2.61	2.85	3.92	3.02	6.58
Acc. (rad/s^2)	19.7	16.8	20.7	20.9	23.7	33.5
Tor. (N m)	309.2	642.4	382.4	117.6	46.6	10.8

Table 3
Breakdown of proposed method and TOPP-RA total computation time (ms) to plan a path discretized with 500 grid points.

	Calculate maximum velocity curves		Search switch points	Calculate linear acceleration constraints	Plan time-optimal trajectory	Total
	Forward pass	Backward pass				
Proposed algorithm	198.7	108.7	1.9	325.1	274.3	908.7
TOPP-RA	198.7	108.7	–	–	179.0	486.4

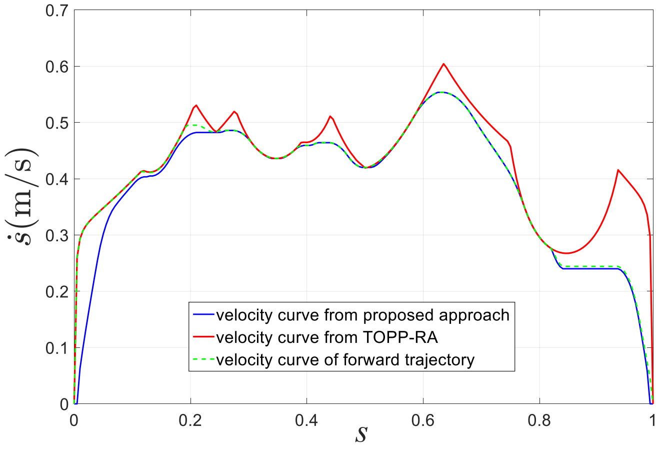


Fig. 7. Comparing the resulting velocity curve of trajectory from TOPP-RA without jerk limits (red solid line) and the one from proposed approach with torque and jerk limits (blue solid line). The green dashed line indicates the velocity curve (**FV**) of forward trajectory which is calculated in Section 3.3.1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

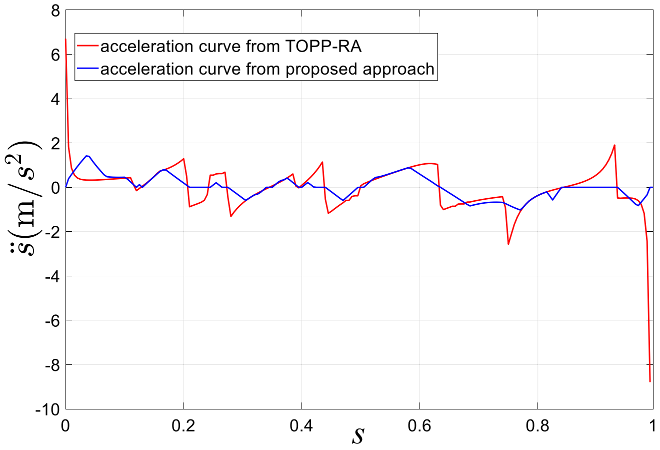


Fig. 8. Comparing the resulting acceleration curve of trajectory from TOPP-RA without jerk limits (red solid line) and the one from proposed approach with torque and jerk limits (blue solid line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

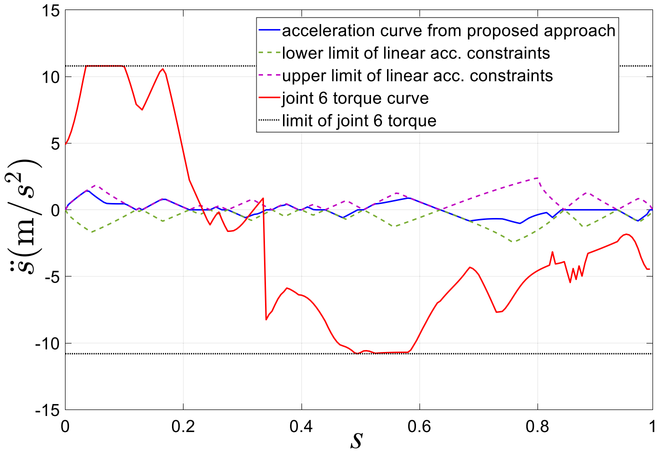


Fig. 9. The influence of constraints on acceleration curve of trajectory from proposed approach. The blue line indicates the acceleration curve of trajectory from proposed approach. The purple and green dashed lines indicate the upper limit of linear acceleration constraints curve and the lower limit of linear acceleration constraints curve respectively. The red line indicates joint6 torque curve and the black dotted line indicates the limit of joint 6 torque. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

transition between acceleration and deceleration segments on the velocity curve. By contrast, in Fig. 8, the proposed approach can effectively restrain acceleration mutation by indirectly introducing jerk limits into CO. At the same time, in Fig. 7, the transitions of velocity curve from proposed approach between acceleration and deceleration segments are smoother. Note that, the case 3 in Fig. 5 occurs when the velocity curve from proposed approach remains unchanged between $s = 0.85$ and $s = 0.95$ in Fig. 7. In simple terms, the velocity curve remains unchanged at this segment is to ensure that the constraints will not be violated on adjacent segments.

Fig. 9 shows the dominated constraints at each path point during the implementation of the proposed approach. The torque limit of joint 6 is the only one in effect among all joint torque limits. As can be summarized from Fig. 9 that the linear acceleration constraints are active near the switch points. At other path points, torque limits and velocity limit (FV_i) of forward trajectory which is the green dashed line in Fig. 7 calculated in Section 3.3.1 play a dominant role. Note that, velocity limit (FV_i) of forward trajectory is active between $s = 0.6$ and $s = 1$ which is shown obviously in Fig. 7.

Fig. 10 shows joint trajectories calculated from TOPP-RA and proposed approach. In particular, trajectories in A1-3 last 2.47 s, and trajectories in B1-3 last 3.16 s The maximum value of joint jerk curves from proposed approach reduces by averaged 81.88% compared with the ones from TOPP-RA, which is shown in Table 4. Thus, the joint acceleration curves transform more smoothly between the acceleration and deceleration segments at

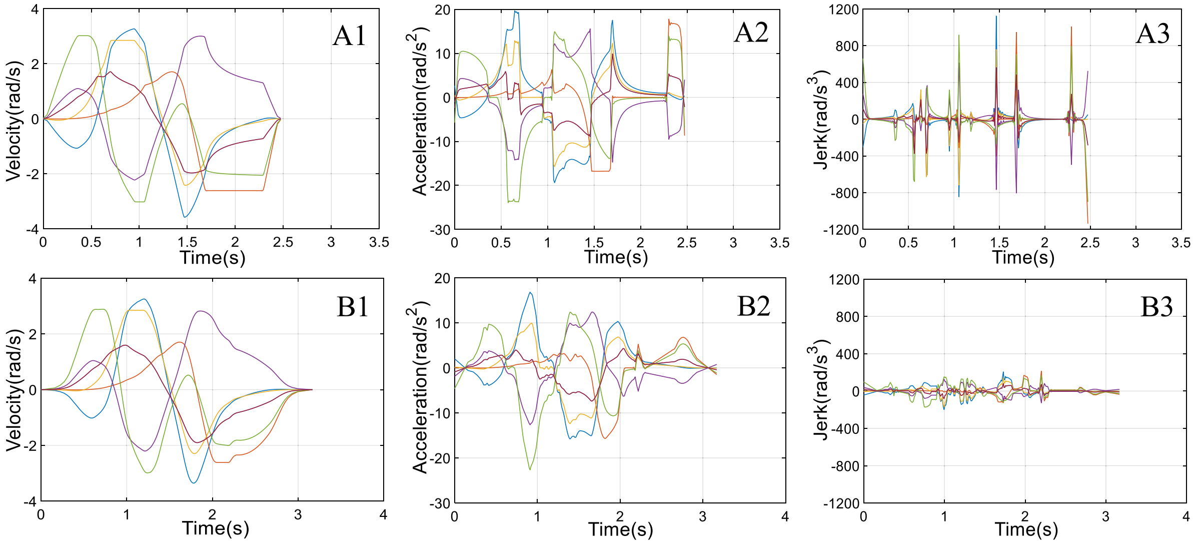
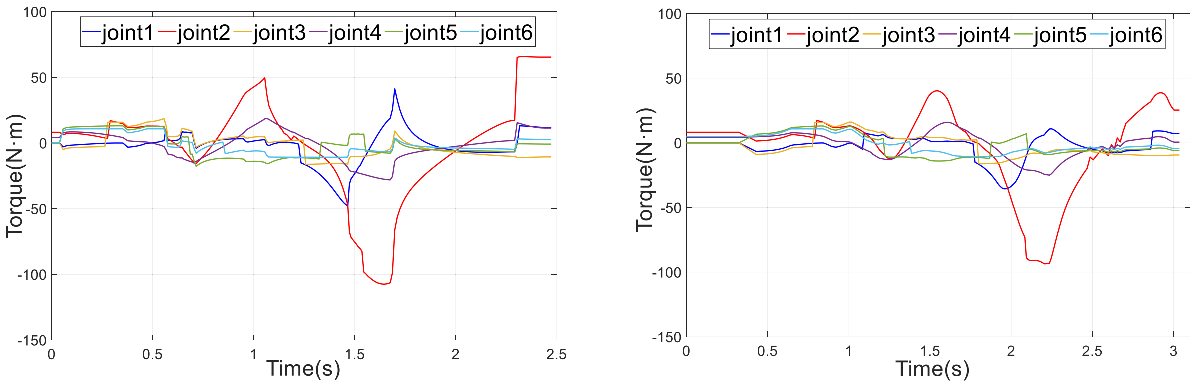


Fig. 10. Comparing the resulting trajectories from TOPP-RA without any constraint on jerk (A1-3) and the ones from proposed approach with $\dddot{s}_{\max/\min} = \dot{U}_{\max/\min} = \pm 10$ m/s^3 bound on jerk (B1-3).



(a) Joint torque curves from TOPP-RA

(b) Joint torque curves from proposed approach

Fig. 11. Comparing the resulting joint torque curves from (a) TOPP-RA without any constraint on jerk and (b) the ones from proposed approach with $\dddot{s}_{\max/\min} = \dot{U}_{\max/\min} = \pm 10$ m/s^3 bound on jerk.

Table 4
Comparing the maximum value of joint jerk curves from two methods.

	Joint1	Joint2	Joint3	Joint4	Joint5	Joint6
TOPP-RA (rad/s^3)	1123.3	1137.7	758.5	803.7	916.2	561.1
Proposed algorithm (rad/s^3)	205.8	213.6	170.7	126.7	172.6	81.3
Degree of decline (%)	81.68	81.23	77.50	84.24	81.16	85.51

the switch points, where the problem of acceleration mutation is effectively restrained. Furthermore, the joint velocity curves become smoother at the switch points by changing velocity ahead of time, which is more in line with the actual requirement of robotic system. And Fig. 11 shows joint torque curves from TOPP-RA and proposed approach. In comparison, the joint torque curves with jerk limits are smoother, which makes the motion of the robotic system smoother. In particular, the torque curve of joint 2 in Fig. 11(b) is smoother than the one in Fig. 11(a), where the torque rate decrease by 55.5%.

5. Conclusion

A new TOTP approach based on CO is proposed, which is able to deal with the torque and jerk limits. The major feature of this approach is that the jerk limits are transformed into linear acceleration constraints and indirectly introduced into optimization problems as the linear acceleration constraints. In this way, the convexity of CO will not be destroyed and the number of optimization variables will not increase. Therefore, the convex optimization problems are maintained in the 2D phase plane, where some optimization problems are performed in 2D projections and others are performed in 1D projections. In addition, the proposed approach is implemented on random geometric paths compared with a similar method to TOTP and experimental results are performed. It is shown that the time-optimal trajectory planned by the proposed approach is smoother than the one planned by the similar method. Furthermore, it is also demonstrated that the new approach has a good performance on computation time and has the ability to restrain acceleration mutation.

Future work will be devoted to exploring the mapping relationship between the path jerk constraints and the joint jerk constraints. By replacing the path jerk constraints with the joint jerk constraints in this approach, the joint jerk can be controlled more accurately.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 51975098), Liaoning Revitalization Talents Program, China (No. XLYC1907006, XLYCYSZX1901, XLYC1801008), Science and Technology Innovation Fund of Dalian, China (No. 2018J12GX038 and 2019CT01), and the Fundamental Research Funds for the Central Universities, China.

References

[1] J.E. Bobrow, S. Dubowsky, J.S. Gibson, Time-optimal control of robotic manipulators along specified paths, Int. J. Robot. Res. 4 (3) (1985) 3–17, http://dx.doi.org/10.1177/027836498500400301.

[2] Z. Shiller, H.H. Lu, Robust computation of path constrained time optimal motions, in: IEEE International Conference on Robotics & Automation, 1990, pp. 144–149, http://dx.doi.org/10.1109/ROBOT.1990.125962.

[3] K. Shin, N. McKay, Minimum-time control of robotic manipulators with geometric path constraints, IEEE Trans. Automat. Control 30 (6) (1985) 531–541, http://dx.doi.org/10.1109/TAC.1985.1104009.

[4] F. Pfeiffer, R. Johanni, A concept for manipulator trajectory planning, IEEE J. Robot. Autom. 3 (2) (1987) 115–123, http://dx.doi.org/10.1109/robot.1986.1087500.

[5] M. Oberherber, H. Gattringer, A. Müller, Successive dynamic programming and subsequent spline optimization for smooth time optimal robot path tracking, Mech. Sci. 6 (2) (2015) 245–254, http://dx.doi.org/10.5194/ms-6-245-2015.

[6] D. Constantinescu, E.A. Croft, Smooth and time-optimal trajectory planning for industrial manipulators along specified paths, J. Robot. Syst. 17 (5) (2000) 233–249, http://dx.doi.org/10.1002/(SICI)1097-4563(200005)17:5<233::AID-ROB1>3.0.CO;2-Y.

[7] D. Kaserer, H. Gattringer, A. Muller, Nearly optimal path following with jerk and torque rate limits using dynamic programming, IEEE Trans. Robot. 35 (2) (2019) 521–528, http://dx.doi.org/10.1109/TRO.2018.2880120.

[8] T. Kunz, M. Stilman, Time-optimal trajectory generation for path following with bounded acceleration and velocity, Robot. Sci. Syst. (2013) 209–216, http://dx.doi.org/10.15607/rss.2012.viii.027.

[9] Q.C. Pham, A general, fast, and robust implementation of the time-optimal path parameterization algorithm, IEEE Trans. Robot. 30 (6) (2014) 1533–1540, http://dx.doi.org/10.1109/TRO.2014.2351113.

[10] H. Pham, Q.C. Pham, On the structure of the time-optimal path parameterization problem with third-order constraints, in: IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 679–686, http://dx.doi.org/10.1109/ICRA.2017.7989084.

[11] S. Kucuk, Optimal trajectory generation algorithm for serial and parallel manipulators, Robot. Comput.-Integr. Manuf. 48 (2017) 219–232, http://dx.doi.org/10.1016/j.rcim.2017.04.006.

[12] S. Kucuk, Maximal dexterous trajectory generation and cubic spline optimization for fully planar parallel manipulators, Comput. Electr. Eng. 56 (2016) 634–647, http://dx.doi.org/10.1016/j.compeleceng.2016.07.012.

[13] H. Pham, Q.C. Pham, A new approach to time-optimal path parameterization based on reachability analysis, IEEE Trans. Robot. 34 (3) (2018) 645–659, http://dx.doi.org/10.1109/TRO.2018.2819195.

[14] Q. Zhang, S.R. Li, X.S. Gao, Practical smooth minimum time trajectory planning for path following robotic manipulators, in: American Control Conference (ACC), 2013, pp. 2778–2783, http://dx.doi.org/10.1109/ACC.2013.6580255.

[15] J.S. Huang, P.F. Hu, K.Y. Wu, M. Zeng, Optimal time-jerk trajectory planning for industrial robots, Mech. Mach. Theory 121 (2018) 530–544, http://dx.doi.org/10.1016/j.mechmachtheory.2017.11.006.

[16] A. Gasparetto, V. Zanotto, A new method for smooth trajectory planning of robot manipulators, Mech. Mach. Theory 42 (4) (2007) 455–471, http://dx.doi.org/10.1016/j.mechmachtheory.2006.04.002.

[17] M. Boryga, A. Graboś, Planning of manipulator motion trajectory with higher-degree polynomials use, Mech. Mach. Theory 44 (2009) 1400–1419, http://dx.doi.org/10.1016/j.mechmachtheory.2008.11.003.

[18] Y. Fang, J. Hu, W. Liu, Q. Shao, J. Qi, Y. Peng, Smooth and time-optimal S-curve trajectory planning for automated robots and machines, Mech. Mach. Theory 137 (2019) 127–153, http://dx.doi.org/10.1016/j.mechmachtheory.2019.03.019.

[19] Z. Xu, S. Li, Q. Chen, B. Hou, MOPSO based multi-objective trajectory planning for robot manipulators, in: 2015 2nd International Conference on Information Science and Control Engineering (ICISCE), 2015, pp. 824–828, http://dx.doi.org/10.1109/ICISCE.2015.188.

[20] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics: Modelling, Planning and Control, Springer Science & Business Media, 2010, https://refhub.elsevier.com/S0094-114X(18)31789-0/sbref0021.

[21] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, M. Diehl, Time-optimal path tracking for robots: a convex optimization approach, IEEE Trans. Automat. Control 54 (10) (2009) 2318–2327, http://dx.doi.org/10.1109/TAC.2009.2028959.

[22] K. Hauser, Fast interpolation and time-optimization with contact, Int. J. Robot. Res. 33 (9) (2014) 1231–1250, http://dx.doi.org/10.1177/0278364914527855.

[23] S. Caron, Q.C. Pham, Y. Nakamura, Leveraging cone double description for multi-contact stability of humanoids with applications to statics and dynamics, Robot. Sci. Syst. (2015) http://dx.doi.org/10.15607/RSS.2015.XI.028.

[24] S. Caron, Q.C. Pham, Y. Nakamura, ZMP support areas for multicontact mobility under frictional constraints, IEEE Trans. Robot. 33 (1) (2017) 67–80, http://dx.doi.org/10.1109/TRO.2016.2623338.

[25] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, M. Diehl, QpOASES: A parametric active-set algorithm for quadratic programming, Math. Program. Comput. 6 (4) (2014) 327–363, http://dx.doi.org/10.1007/s12532-014-0071-1.

Jian-Wei Ma received his Ph.D. degree from School of Mechanical Engineering, Dalian University of Technology, China, in 2011. He is currently a Professor in School of Mechanical Engineering, Dalian University of Technology. His research interests include issues related to the numerical control technology, unconventional processing, robot-assisted machining.

Song Gao is a master candidate at the School of Mechanical Engineering, Dalian University of Technology, China. His research interests include issues related to time-optimal trajectory planning with kinematic and dynamic limits for robotic system.

Hui-teng Yan is a Ph.D. student at the School of Mechanical Engineering, Dalian University of Technology. His research interests include error identification and compensation for robotic system.

Qi Lv is a master candidate at the School of Mechanical Engineering, Dalian University of Technology, China. His research interests include issues related to dynamic parameter identification and trajectory tracking control for robotic system.

Guo-qing Hu is a Ph.D. student at the School of Mechanical Engineering, Dalian University of Technology. His research interests include five-axis CNC machining technology and motor motion control technology.