ISA Transactions 131 (2022) 639–649

Contents lists available at ScienceDirect

ISA Transactions

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

Practice article

Real-time time-optimal continuous multi-axis trajectory planning using the trajectory index coordination method

Suqin He [a,b], Chuxiong Hu [a,b,*], Shize Lin [a,b], Yu Zhu [a,b], Masayoshi Tomizuka [c]

[a] State Key Laboratory of Tribology, Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China
[b] Beijing Key Laboratory of Precision/Ultra-Precision Manufacture Equipments and Control, Tsinghua University, Beijing 100084, China
[c] Department of Mechanical Engineering, University of California, Berkeley, CA 94720-1740, USA

ARTICLE INFO

Article history:
Received 9 November 2021
Received in revised form 16 May 2022
Accepted 16 May 2022
Available online 27 May 2022

Keywords:
Trajectory planning
Time-optimal control
Real-time motion planning
State feedback

ABSTRACT

Real-time time-optimal trajectory planning exists in a wide range of applications such as computer numerical control (CNC) manufacturing, robotics and autonomous vehicles. Generally, the methods to generate time-optimal trajectory can be categorized as non-real-time methods and real-time methods. Non-real-time methods such as direct optimization method tend to generate time-optimal trajectory through nonlinear or linear programming while it is computationally prohibitive for high frequency real-time applications. Current real-time methods are computationally efficient but either deal with the sparse waypoint trajectories or sacrifice the time optimality a lot. This paper innovatively proposed a time-optimal switching trajectory index coordination (TOS-TIC) framework to solve the real-time time-optimal planning problem for continuous multi-axis trajectories. The proposed method is able to generate time-optimal trajectory for continuous geometric paths while considering the axial velocity and acceleration constraints. The time-optimality of the trajectory planned by TOS-TIC is nearly the same as the offline planned optimal results. Meanwhile, the proposed method is computationally efficient for even 5 kHz real-time applications. The main idea of TOS-TIC is coordinating several one-axis time-optimal switching controls to generate a modified control that decreases the state deviation from the desired trajectory. Several comparative experiments are carried out on an industrial biaxial linear motor stage. And the experimental results consistently verify that the proposed TOS-TIC real-time planner generates faster trajectory compared with the real-time lookahead method. In addition, the trajectory running time and final tracking error of the proposed method are nearly the same as the offline direct optimization method.

© 2022 ISA. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Real-time smooth trajectory planning is with wide application in computer numerical control (CNC) manufacturing systems [1–8], robotics [9–12], autonomous vehicles [13–15] and industrial process [16,17]. In most applications, the trajectory velocity, acceleration or jerk are required to be bounded due to drive capability and smoothness requirement. Meanwhile, short trajectory running time is preferable for high responsiveness and productivity. But the constraints of velocity/acceleration/jerk limit the decrease of trajectory running time. Therefore, the time-optimal trajectory planning with velocity/acceleration/jerk constraints is synthesized to find the balance between the minimum time objective and state constraints. Generally, the research methods solving this time-optimal trajectory planning problem can be classified as real-time methods and non-real-time methods.

The non-real-time methods plan the whole trajectory offline before sending the trajectory to the motion system for tracking. Early researches on robotic systems use phase plane method [18] to iteratively find the switching point of acceleration/deceleration on the trajectory. Thereafter, bidirectional searching method for acceleration constrained planning [19], searching methods in phase space for jerk constrained [20] or torque change rate constrained [21] planning, and greedy methods for 5-axis CNC machine tools [22] are proposed. Recently, E. Barnett et al. improved the bidirectional switching point searching method and introduced a C++ library named BA-TOTP [23], which is able to generate time-optimal feedrate profile for serial and parallel manipulators with both kinematic and dynamics constraints. However, these iteratively searching methods tend to be computationally inefficient for long trajectories. The direct optimization method [24,25] discretizes the trajectory and formulates a large optimization problem which can be solved using well established

* Corresponding author.
E-mail addresses: hesq16@mails.tsinghua.edu.cn (S. He), cxhu@tsinghua.edu.cn (C. Hu), linsz20@mails.tsinghua.edu.cn (S. Lin), zhuyu@tsinghua.edu.cn (Y. Zhu), tomizuka@me.berkeley.edu (M. Tomizuka).

https://doi.org/10.1016/j.isatra.2022.05.016
0019-0578/© 2022 ISA. Published by Elsevier Ltd. All rights reserved.

Nomenclature

Abbreviations

BA-TOTP	Bisection algorithm for time-optimal trajectory planning
CNC	Computer numerical control
LP	Linear programming
MPC	Model predictive control
NURBS	Non-uniform rational B-spline
QP	Quadratic programming
RTIPC	Real-time interpolator for parametric curves
SSA	Safe set algorithm
TIC	Trajectory index coordination
TOCP	Time-optimal control problem
TOS	Time-optimal switching

Latin Letters

$\boldsymbol{A}$	State matrix of the system
$\boldsymbol{B}$	Input matrix of the system
$\boldsymbol{u}$	Control input
$\boldsymbol{x}$	System state
$\mathcal{F}$	Feasible set
$\mathcal{U}$	Control admissible set
$\mathcal{X}$	State related sets
a	Acceleration
e	Tracking error
f, g, h	Path constraints
f_s	Control frequency
H	Hamiltonian
k	Control gains
P	System position
t	Time
t_f	Terminal time
T_s	Control cycle time
U	Control constraints
V	Velocity constraints
v	Velocity
X	State constraints

Greek Letters

α	Coefficient in trajectory index
$\boldsymbol{\eta}$	Multipliers of Hamiltonian
$\boldsymbol{\lambda}$	Costates of Hamiltonian
δ	State margins
γ	Reaching curves in phase space
Ω	Regions in phase space
Φ	Trajectory index
Ψ	Augmented trajectory index

Subscripts/Superscripts

$(\cdot)^{ref}$, $(\cdot)^{actual}$	Reference value and actual value
$(\cdot)^{rms}$, $(\cdot)^{max}$	Root-mean-square value and maximum absolute value
$(\cdot)_S$, $(\cdot)_V$	Variables of position and velocity
$(\cdot)_x$, $(\cdot)_y$	Variables of X-axis and Y-axis
$(\cdot)_\Phi$, $(\cdot)_\Psi$	Variables of trajectory index and augmented trajectory index
$(\cdot)_c$	Current state
$(\cdot)_m$, $(\cdot)_{1m}$, $(\cdot)_{2m}$	Upper bound of constraints
$(\cdot)_{u0}$, $(\cdot)_{u+}$, $(\cdot)_{u-}$	Regions or curves for 0, maximum and minimum control value

solvers. Due to the convenience of direct optimization methods in problem formulation and solution, it becomes the most popular method for offline time-optimal trajectory planning recently. Further researches also transform the nonlinear programming problem in [24] into the linear programming problem [26] or quadratic programming problem [10] which can be solved efficiently. Nevertheless, these planning methods discretize the trajectory into thousands of points or more, which causes too much computation burden to apply in high frequency real-time applications. Thus, one must firstly plan the whole trajectory offline before sending the trajectory to the motion system for tracking.

The real-time methods plan the whole trajectory or a part of trajectory online while the motion system is still running. Time synchronization methods are used in 5-axis CNC systems [6] and robotic systems [9,10] those require real-time responsiveness. L. Berscheid et al. analysed the time synchronization method for arbitrary target state and introduced a real-time jerk limited online trajectory generation library Ruckig [27] for multi-axis robots and machines. However, these methods can either deal with point-to-point trajectories or sparse waypoint trajectories, which limits their application for high precision continuous trajectories. Many researchers use real-time look-ahead methods [3] to find the key points based on the continuous trajectory curvature. Then these key points are connected by S-shape acceleration/deceleration profiles. The real-time look-ahead methods are also applied in commercial CNC systems such as Siemens 840D and FANUC 31i-series [4]. However, the resulting trajectories sacrifice the time optimality a lot compared with offline direct optimization methods [4]. Besides, only tangential velocity/acceleration/jerk of the trajectory can be constrained rather than the axial velocity/acceleration/jerk. The tangential constraint sacrifices the time optimality due to the insufficient usage of axis driving abilities. Zhong et al. considered the axial constraints and developed a real-time interpolator for parametric curves (RTIPC) based on feedrate look-ahead and acceleration look-ahead techniques [28]. Axial constraints precisely depict the drive ability thus improve the task efficiency significantly. Some researchers try the time-optimal control method which reformulate the planning problem into a time-optimal control problem of double integrator system [29,13] or triple integrator system [5]. The benefit of this method is that it can obtain real-time state-feedback form time-optimal control law for the trajectory planning. The research in [30] followed this idea and proposed the analytical state-feedback time-optimal switching (TOS) control law for triple integrator system with full state constraints and input saturation. But this method is currently restricted to one-axis trajectory planning.

In summary, the non-real-time methods generate time-optimal results but tend to be computationally inefficient, while real-time methods are fast but either deal with sparse waypoint trajectories or sacrifice the time optimality a lot. This paper intends to develop a novel time-optimal planning method for continuous multi-axis trajectories with axial kinematic constraints. Firstly, the multi-axis system is decoupled into independent one-axis systems. Then the time-optimal switching (TOS) control law is synthesized for decoupled one-axis systems. Next, the trajectory index coordination (TIC) method is proposed which

coordinates the decoupled one-axis TOS output and generates a modified multi-axis output. The goal of TIC is to decrease the system state deviation from the desired trajectory while minimizing the difference between the TOS output. By systematically unifying TOS and TIC, a TOS-TIC real-time planning framework is proposed. The TOS-TIC planner can work in open-loop form or closed-loop form depending on whether the real state feedback is used or not. The key contributions of this paper are summarized below:

- A novel real-time time-optimal trajectory planning method is proposed for continuous multi-axis trajectories. Compared with existing methods, the proposed TOS-TIC method is (1) computationally efficient for real-time applications, (2) near-time-optimal as the direct optimization results.
- A unified framework is proposed such that the trajectory planner and servo controller are coordinated for the same goal of decreasing the contour error.
- The proposed method extends the one-axis time-optimal switching research into multi-axis coordination strategy, and provides a real-time state-feedback multi-axis planning/control law.

The remainder of the paper is organized as follows. Section 2 formulates the optimal control problem. Section 3 introduces the TOS method. Section 4 proposes the TIC method. Section 5 proposes the TOS-TIC real-time planning framework. Section 6 introduces the experiment designs and results. Section 7 concludes the paper.

2. Problem formulation

As mentioned in the introduction, the real-time time-optimal trajectory planning problem for multi-axis continuous path with axial velocity and acceleration constraints is still not well solved yet. However, the trajectory planner solving this problem is important in many practical scenarios such as CNC systems [1–6] and robotics [9,10]. This problem is essentially a time-optimal control problem (TOCP) where the acceleration is the control input, and the velocity/displacement are system states. For each axis, the dynamic system is a double integrator system with state constraints and saturated control. In addition, all axes are coordinated to formulate a continuous geometric path, i.e., a path constraint between axes. Therefore, the whole TOCP can be synthesized as

$$\min_{\boldsymbol{u}\in\mathcal{U}} J = \int_0^{t_f} dt \tag{1a}$$

$$\text{s.t.}\quad \dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}\boldsymbol{u}(t) \tag{1b}$$

$$|\boldsymbol{x}(t)| \le \boldsymbol{X}_m \tag{1c}$$

$$|\boldsymbol{u}(t)| \le \boldsymbol{U}_m \tag{1d}$$

$$f(\boldsymbol{x}) = 0 \tag{1e}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0,\ \boldsymbol{x}(t_f) = \boldsymbol{x}_f \tag{1f}$$

where (1a) is the objective function minimizing the terminal time t_f, (1b) is the system dynamics constraint, (1c) is the state constraint, (1d) is the control constraint where the control *admissible set* $\mathcal{U} = \{\boldsymbol{u} : |\boldsymbol{u}(t)| \le \boldsymbol{U}_m\}$, (1e) is the path constraint and (1f) are initial condition and terminal condition constraints. And the operator $|\cdot|$ and $\le$ in (1c) and (1d) are performed in accordance with corresponding elements.

The TOCP (1) is the general structure of multi-axis planning problem. Specifically, for the two-axis velocity/acceleration constrained planning problem, one have $\boldsymbol{u} = [u_1, u_2]^\top \in \mathbb{R}^2$, where u_1 and u_2 are accelerations of axis 1 and axis 2, respectively. And $\boldsymbol{x} = [x_1, x_2, x_3, x_4]^\top \in \mathbb{R}^4$, where x_1, x_3 are displacements and velocities of axis 1, and x_2, x_4 are displacements and velocities of axis 2. Accordingly, one has

$$\boldsymbol{A} = \operatorname{diag}\left\{\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\right\},\quad \boldsymbol{b} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}^\top \tag{2}$$

The control constraint in (1d) is $\boldsymbol{U}_m = [U_{1m}, U_{2m}]^\top$, where $U_{1m}, U_{2m} \in \mathbb{R}^+$ are acceleration upper bounds. The state constraint in (1c) contains only velocity constraints, i.e., $|x_2| \le V_{1m}$, $|x_4| \le V_{2m}$. For the two-axis system, the path constraint in (1e) is actually a planar curve with respect to (x_1, x_3), which is usually represented by implicit equation $f(x_1, x_3) = 0$ or explicit equations $x_3 = g(x_1)$ and $x_1 = h(x_3)$, where $f : \mathbb{R}\times\mathbb{R} \mapsto \mathbb{R}$, $g : \mathbb{R} \mapsto \mathbb{R}$ and $h : \mathbb{R} \mapsto \mathbb{R}$. In this paper, it is assumed that the desired path is smooth enough, i.e., $f, g, h \in \mathbb{C}^3$.

The common method of solving TOCP (1) is through the Pontryagin minimum principle [31] in optimal control theory. In preparation for the further analyses in the next section, a series of optimality conditions are synthesized. First, the Hamiltonian is constructed as

$$\begin{aligned} H(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\eta}, t) = {} & 1 + \lambda_1 x_2 + \lambda_2 u_1 + \lambda_3 x_4 + \lambda_4 u_2 \\ & + \eta_1(|x_2| - V_{1m}) + \eta_2(|x_4| - V_{2m}) + \eta_3 \ddot{f}(\boldsymbol{x}, \boldsymbol{u}) \end{aligned} \tag{3}$$

where $\boldsymbol{\lambda}(t) = [\lambda_1, \lambda_2, \lambda_3, \lambda_4]^\top \in \mathbb{R}^4$ are costates, and $\boldsymbol{\eta}(t) = [\eta_1, \eta_2, \eta_3]^\top \in \mathbb{R}^3$ are scalar multipliers.

Remark 1. The path constraint in the Hamiltonian is represented by $\ddot{f}(\boldsymbol{x}, \boldsymbol{u}) = 0$ rather than $f(\boldsymbol{x}) = 0$ since the path constraint $f(\boldsymbol{x}) = 0$ is not explicitly containing control $\boldsymbol{u}$ [32]. Actually, for the constraint $f(x_1, x_3) = 0$, the first and second order derivative are

$$\begin{aligned} \dot{f} &= f_{x_1} x_2 + f_{x_3} x_4 \\ \ddot{f} &= f_{x_1x_1} x_2^2 + 2 f_{x_1x_3} x_2 x_4 + f_{x_3x_3} x_4^2 + f_{x_1} u_1 + f_{x_3} u_2 \end{aligned} \tag{4}$$

where $f_{x_i} = \frac{\partial f}{\partial x_i}$ and $f_{x_ix_j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$ for $i, j = 1, 2, 3, 4$. Thus the relative degree from $\ddot{f}$ to u_1 or u_2 is 1.

The costates $\boldsymbol{\lambda}(t)$ satisfy the following Euler–Lagrange equations

$$\begin{aligned} -\dot{\lambda}_1 &= \frac{\partial H}{\partial x_1} = \eta_3 \frac{\partial \ddot{f}}{\partial x_1}, \quad -\dot{\lambda}_2 = \frac{\partial H}{\partial x_2} = \lambda_1 \pm \eta_1 + \eta_3 \frac{\partial \ddot{f}}{\partial x_2} \\ -\dot{\lambda}_3 &= \frac{\partial H}{\partial x_3} = \eta_3 \frac{\partial \ddot{f}}{\partial x_3}, \quad -\dot{\lambda}_4 = \frac{\partial H}{\partial x_4} = \lambda_3 \pm \eta_2 + \eta_3 \frac{\partial \ddot{f}}{\partial x_4} \end{aligned} \tag{5}$$

The sign of $\pm$ depends on which boundary is reached. And the multipliers $\boldsymbol{\eta}(t)$ satisfy

$$\begin{aligned} &\eta_1(|x_2| - V_{1m}) = 0,\ \eta_2(|x_4| - V_{2m}) = 0 \\ &\eta_1(t) \ge 0,\ \eta_2(t) \ge 0,\ \eta_3(t) \ge 0 \end{aligned} \tag{6}$$

Suppose the optimal control law is $\boldsymbol{u}^*(t)$ and the resulting optimal state trajectory is $\boldsymbol{x}^*(t)$. According to the Pontryagin minimum principle [31], the optimal $\boldsymbol{u}^*(t)$ minimizes the Hamiltonian. Therefore, one has

$$H(\boldsymbol{u}^*, \boldsymbol{x}^*, \boldsymbol{\lambda}, \boldsymbol{\eta}, t) = \min_{|\boldsymbol{u}| \le \boldsymbol{U}_m} H(\boldsymbol{u}, \boldsymbol{x}^*, \boldsymbol{\lambda}, \boldsymbol{\eta}, t) \tag{7}$$

Substituting (4) into (3) one obtains

$$\begin{aligned} H(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\eta}, t) = {} & (\lambda_2 + \eta_3 f_{x_1}) u_1 + (\lambda_4 + \eta_3 f_{x_3}) u_2 \\ & + 1 + \lambda_1 x_2 + \lambda_3 x_4 \\ & + \eta_1(|x_2| - V_{1m}) + \eta_2(|x_4| - V_{2m}) \\ & + \eta_3(f_{x_1x_1} x_2^2 + 2 f_{x_1x_3} x_2 x_4 + f_{x_3x_3} x_4^2) \end{aligned} \tag{8}$$

Noticing (7), one can obtain the following control law to minimize the Hamiltonian

$$\boldsymbol{u}^*(t) = \underset{\substack{|\boldsymbol{u}|\le \boldsymbol{U}_m \\ \dot{f}(\boldsymbol{x},\boldsymbol{u})=0}}{\arg\min} \left[(\lambda_2 + \eta_3 f_{x_1})u_1 + (\lambda_4 + \eta_3 f_{x_3})u_2\right] \tag{9}$$

It is difficult to obtain the analytical solution of (9) even for simple trajectories. However, there are many characteristics can be found from (9). The most important characteristic to be discussed in this paper is the coupling nature of the system, i.e., the two-axis system can be treated as two independent time-optimal controlled double integrator subsystems while they are coupled by path constraint f. Actually, the following theorem is summarized:

Theorem 1. *Suppose there is no path constraint $f(\boldsymbol{x}) = 0$ in TOCP (1), then it can be decoupled into two independent TOCP for double integrator systems, and their optimal control $u_1^*(t)$, $u_2^*(t)$ are independent bang-singular-bang control laws.*

Proof. In fact, the proof of Theorem 1 is quite straightforward referring to (9). For the simplicity of this manuscript, the proof is omitted temporarily. □

Theorem 1 provides a different idea of solving multi-axis TOCP (1) that one can temporally neglect the path constraint $f(\boldsymbol{x}) = 0$ and treat each axis separately at first. For each axis, the optimal control law $u_i^*(t)$ can be generated. Then, a coordination mechanism is synthesized to modify these control laws in order to make the multi-axis motion satisfying the path constraint $f(\boldsymbol{x}) = 0$. In this way, the difficult problem can be divided into two simpler sub-problems, i.e., solving the one-axis TOCP and synthesizing a coordination mechanism. To solve the one-axis TOCP, one can leverage the research works in [30], where the time-optimal control of triple integrator system with input saturation and full state constraints was studied, and an analytical state-feedback time-optimal switching (TOS) method was proposed. The TOS method will be briefly introduced in the next section, and a robust version for real-time implementation will be proposed. Besides, in order to synthesize a coordination mechanism, this paper proposes a trajectory index coordination (TIC) method which will be further discussed in Section 4.

3. Time-optimal switching (TOS) method

Our previous research in [30] has studied the TOCP of triple integrator system with full state constraints and input saturation, where an analytical state-feedback optimal control law is obtained. In this paper, the decoupled one-axis system is actually a double integrator system with state constraints and input saturation. Thus, the results in [30] can be equivalently migrated to the analyses in this paper. Without loss of generality, the results of axis-1 is taken as an example in this section. The results for axis-2 is the same as axis-1.

The analytical bang-singular-bang control law in [30] is obtained with the help of dynamic programming principle. The general idea of the deduction can be separated into three steps:

1. Find the boundary control when $\lambda_2 = 0$. It is found out that the boundary control is $u_1^*(t) = 0$. Therefore, all possible control values of $u_1^*(t)$ are U_{1m}, $-U_{1m}$ or 0. Correspondingly, three sets of curves with different control values are generated.
2. Find the *final reaching curves* that pass through the goal $(x_{1g}, 0)$ in minimum time, i.e.,

$$\gamma_{u+}:\ u_1^* = U_{1m},\ x_1 = \frac{x_2^2}{2U_{1m}} + x_{1g},\ x_2 \le 0 \tag{10a}$$

$$\gamma_{u-}:\ u_1^* = -U_{1m},\ x_1 = -\frac{x_2^2}{2U_{1m}} + x_{1g},\ x_2 \ge 0 \tag{10b}$$

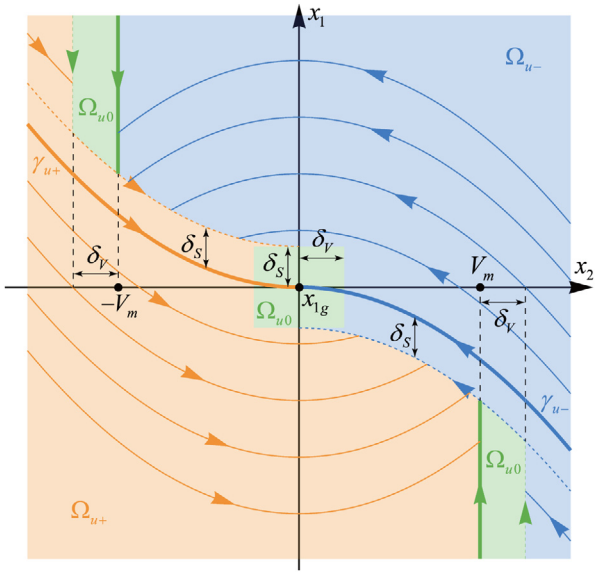

Fig. 1. Robust time-optimal switching (TOS) phase graph. Blue region: Ω_{u-} where $u_1^*(t) = -U_{1m}$. Orange region: Ω_{u+} where $u_1^*(t) = U_{1m}$. Green region: Ω_{u0} where $u_1^*(t) = 0$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3. Find the *switching curves* that reach γ_{u+} and γ_{u-} in minimum time. Then repeatedly find the switching curves that reach the switching curves at last step until the whole switching structure is formulated. An illustration is shown in Fig. 1.

As a result, the whole phase space is divided into three regions Ω_{u+}, Ω_{u-} and Ω_{u0} by these switching curves. The resulting TOS control law $u_1^*(t)$ in these regions are U_{1m}, $-U_{1m}$ and 0, respectively. Write the TOS control law in state-feedback form according to its location in phase graph as

$$u_1^*(x_1, x_2) = \begin{cases} U_{1m}, & (x_1, x_2) \in \Omega_{u+} \\ 0, & (x_1, x_2) \in \Omega_{u0} \\ -U_{1m}, & (x_1, x_2) \in \Omega_{u-} \end{cases} \tag{11}$$

The TOS control law (11) is analytical since all switching curves are analytical. In addition, the TOS control law is proved to be globally time-optimal.

However, the original results in [30] is not suitable for real world application since real world signals may contain noises. And the rigid boundary of switching curves may lead to chattering problem. Therefore, in this paper, a robust modification of the TOS in [30] is proposed. Certain state margins are given for x_1 and x_2, denoted as δ_S and δ_V where $\delta_S, \delta_V > 0$. Boundaries between different regions are relaxed considering the signal noises. The illustration of robust TOS is shown in Fig. 1. And the detailed expressions of Ω_{u+}, Ω_{u-} and Ω_{u0} are

$$\Omega_{u0}: \left\{ \begin{array}{l} |x_1| < \delta_S \text{ and } |x_2| < \delta_V, \\ V_m \le x_2 < V_m + \delta_V \text{ and } x_1 < \gamma_{u-} - \delta_S, \\ -V_m - \delta_V < x_2 \le -V_m \text{ and } x_1 > \gamma_{u+} + \delta_S \end{array} \right\} \tag{12a}$$

$$\Omega_{u-}: \left\{ \begin{array}{l} x_1 \ge \gamma_{u-} - \delta_S \text{ and } 0 \le x_2 < V_m + \delta_V, \\ x_2 \ge V_m + \delta_V, \\ x_1 > \gamma_{u+} + \delta_S \text{ and } -V_m < x_2 < 0 \end{array} \right\} \setminus \Omega_{u0} \tag{12b}$$

$$\Omega_{u+}: \left\{ \begin{array}{l} x_1 \le \gamma_{u+} + \delta_S \text{ and } -V_m - \delta_V < x_2 \le 0, \\ x_2 \le -V_m - \delta_V, \\ x_1 < \gamma_{u-} - \delta_S \text{ and } 0 < x_2 < V_m \end{array} \right\} \setminus \Omega_{u0} \tag{12c}$$

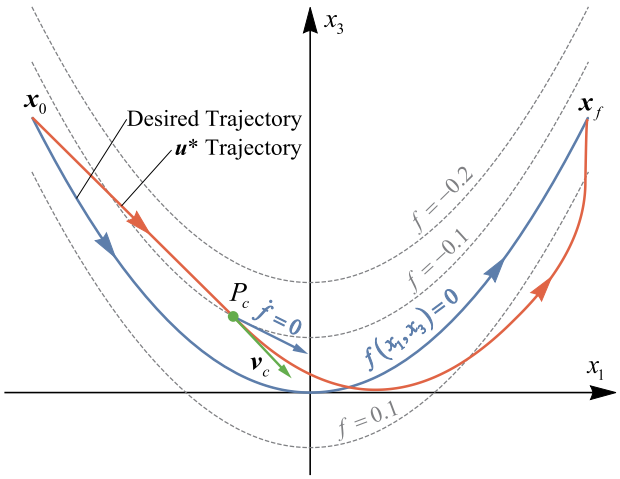

Fig. 2. Illustration of trajectory index. Blue curve: desired trajectory. Red curve: trajectory driven by $\boldsymbol{u}^*$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

It is noted that $\gamma_{u+} = x_2^2/2U_{1m} + x_{1g}$ and $\gamma_{u-} = -x_2^2/2U_{1m} + x_{1g}$ are used here for simplicity.

In real world applications, one can obtain smooth control output by choosing δ_S and δ_V according to their signals. Similarly, the $u_2^*(x_3, x_4)$ can be obtained in the same form as (11). The $\boldsymbol{u}^*(\boldsymbol{x}) = [u_1^*(x_1, x_2), u_2^*(x_3, x_4)]^\top$ will be used as an reference for further coordination in the next section.

4. Trajectory index coordination (TIC)

In Section 3, the time-optimal control law $\boldsymbol{u}^*(\boldsymbol{x}) = \left[u_1^*(x_1, x_2), u_2^*(x_3, x_4)\right]^\top$ for the decoupled axes is constructed. However, driving the multi-axis system with $\boldsymbol{u}^*(\boldsymbol{x})$ will deviate from the desired trajectory, as shown in Fig. 2. Thus, a multi-axis coordination mechanism is necessary to confine the multi-axis motion to the desired trajectory. To reach that goal, this paper proposes a trajectory index coordination (TIC) method which is inspired by the energy function based methods [33–36] in safety control research area, especially the safe set algorithm (SSA) proposed by C. Liu [36]. In these methods, a safety index function $\Phi(\boldsymbol{x})$ is used to evaluate the safety of an obstacle-interacting system. The system state within the safe set $\mathcal{X}_S = \{\boldsymbol{x} : \Phi(\boldsymbol{x}) \leq 0\}$ is regarded as safe. Then, certain safe control laws $\boldsymbol{u}_S(t)$ are synthesized to make $\dot{\Phi} < 0$ when $\Phi > 0$. In this way, any state outside the safe set $\mathcal{X}_S$ will be pulled back from the safe set, i.e., the system is always safe guaranteed. In this paper, the objective is to guarantee $f(\boldsymbol{x}) = 0$, which is similar to the objective of safety control $\Phi(\boldsymbol{x}) \leq 0$ except the equality constraint in this paper. Therefore, the analyse methods in safe control researches can help promote the solving of the problems in this paper.

4.1. Trajectory index

It is noticed that the value of $f(\boldsymbol{x})$ indicates how much the current state is deviated from the desired trajectory. Take a 2-dimensional case for example as shown in Fig. 2, different values of f represent different trajectories, and they seem to be parallel with each other. Actually, the value $\epsilon = f(x_1, x_3)$ is related to the *contour error* of planar curves [37,38], which denotes the shortest distance from current position to the desired trajectory. Therefore, designing a coordination mechanism that enforces $f(\boldsymbol{x}) = 0$ has a physical meaning in geometry.

However, since $\dot{f}$ is not explicitly containing control $\boldsymbol{u}$ as mentioned in Remark 1, it is impossible to design a coordination control law $\boldsymbol{u}$ to control $\dot{f}$ explicitly. Thus the second order derivative must be taken into account because $\ddot{f}$ is explicitly containing control $\boldsymbol{u}$. In fact, by controlling $\ddot{f}$ with certain control law $\boldsymbol{u}$, a new constraint $\dot{f}(\boldsymbol{x}) = 0$ is also enforced in the whole process. The constraint $\dot{f}(\boldsymbol{x}) = 0$ also has a physical meaning in geometry. As illustrated in Fig. 2, suppose the current position is P_c and the current velocity vector is $\boldsymbol{v}_c$ marked as the green arrow. At P_c, the equality $\dot{f} = 0$ depicts a vector marked as the blue arrow in Fig. 2, which is tangent with the curve $f =$ constant. The current velocity vector $\boldsymbol{v}_c$ is deviated from $\dot{f} = 0$. By enforcing $\dot{f} \to 0$, the velocity vector $\boldsymbol{v}_c$ will converge to the tangential direction of the desired trajectory, which prevents the potential off-course from the desired trajectory.

Based on the analysis above, the *trajectory index* is designed as $\Phi(\boldsymbol{x}) = \dot{f} + \alpha f$, where $\alpha \geq 0$ is a weighting factor. The trajectory index $\Phi : \mathbb{R}^4 \mapsto \mathbb{R}$ has the following properties

(P1) Φ is differentiable with respect to t, i.e., $\dot{\Phi} = \frac{\partial \Phi}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}$ exists everywhere.
(P2) $\frac{\partial \dot{\Phi}}{\partial \boldsymbol{u}} \neq 0$, so that the control can always affect $\dot{\Phi}$.

Obviously, any system state on the desired trajectory is within the trajectory set $\mathcal{X}_\Phi = \{\boldsymbol{x} : \Phi(\boldsymbol{x}) = 0\}$. But in the real motion control application, the actual system state may deviate from $\mathcal{X}_\Phi$ due to tracking error and actuation noise. At this time, certain control law is needed to pull back the system state from $\mathcal{X}_\Phi$. Considering the time derivative of $\Phi(\boldsymbol{x})$

$$\begin{aligned}\dot{\Phi} &= \nabla\Phi^\top \boldsymbol{A}\boldsymbol{x} + \nabla\Phi^\top \boldsymbol{B}\boldsymbol{u} \\ &= \frac{\partial \Phi}{\partial x_1}x_2 + \frac{\partial \Phi}{\partial x_3}x_4 + \frac{\partial \Phi}{\partial x_2}u_1 + \frac{\partial \Phi}{\partial x_4}u_2\end{aligned} \tag{13}$$

To ensure the attraction from $\mathcal{X}_\Phi$, the control $\boldsymbol{u}$ must be chosen from the *trajectory index set* $\mathcal{U}_\Phi = \{\boldsymbol{u} : \dot{\Phi} < 0 \text{ if } \Phi > 0, \dot{\Phi} > 0 \text{ if } \Phi < 0\}$. In practice, setting a $\dot{\Phi}$ margin related to the value of Φ will generate smoother motion when a control law $\boldsymbol{u} \in \mathcal{U}_\Phi$ is applied [39]. Therefore, in the rest of the paper, the trajectory index set $\mathcal{U}_\Phi$ is modified as $\mathcal{U}_\Phi = \{\boldsymbol{u} : \dot{\Phi} \leq -k_1\Phi \text{ if } \Phi > 0, \dot{\Phi} \geq -k_1\Phi \text{ if } \Phi < 0\}$ where $k_1 > 0$ is a control gain. From (13), one further has

$$\begin{aligned}\mathcal{U}_\Phi = \{\boldsymbol{u} :& \nabla\Phi^\top \boldsymbol{A}\boldsymbol{x} + \nabla\Phi^\top \boldsymbol{B}\boldsymbol{u} + k_1\Phi \leq 0 \text{ if } \Phi > 0, \\ & \nabla\Phi^\top \boldsymbol{A}\boldsymbol{x} + \nabla\Phi^\top \boldsymbol{B}\boldsymbol{u} + k_1\Phi \geq 0 \text{ if } \Phi < 0\}\end{aligned} \tag{14}$$

In summary, any control law $\boldsymbol{u} \in \mathcal{U}_\Phi$ guarantees the convergence to $\mathcal{X}_\Phi$. The next step is to find an optimal one. Leveraging the time-optimal control $\boldsymbol{u}^*$ synthesized in Section 3, the TIC optimization problem can be formulated as

$$\min_{\boldsymbol{u}} \|\boldsymbol{u} - \boldsymbol{u}^*\|_2 \tag{15a}$$

$$\text{s.t.} \quad \boldsymbol{u} \in \mathcal{U}_\Phi, \quad \boldsymbol{u} \in \mathcal{U} \tag{15b}$$

where $\mathcal{U} = \{\boldsymbol{u} : |\boldsymbol{u}(t)| \leq \boldsymbol{U}_m\}$ is the control admissible set, and $\boldsymbol{u} \in \mathcal{U}_\Phi$ is a linear constraint, thus problem (15) is a quadratic programming (QP) problem with only two decision variables $\boldsymbol{u} = [u_1, u_2]^\top$. The benefit of the QP is that the computation of solving (15) can be extremely efficient, which makes it applicable in real-time scenarios. The solution of (15) is named the trajectory index coordinated control output $\boldsymbol{u}_{TIC}$. Since $\boldsymbol{u}_{TIC}$ minimizes the difference between $\boldsymbol{u}^*$, it is conjectured that the trajectory running time is near-time-optimal by applying $\boldsymbol{u}_{TIC}$ control law, which will be further verified in experiments. In conclusion, the proposed TIC optimization has two merits: (1) the TIC control law drives any deviated system state back to the desired trajectory; (2) the overall trajectory running time is near-time-optimal.

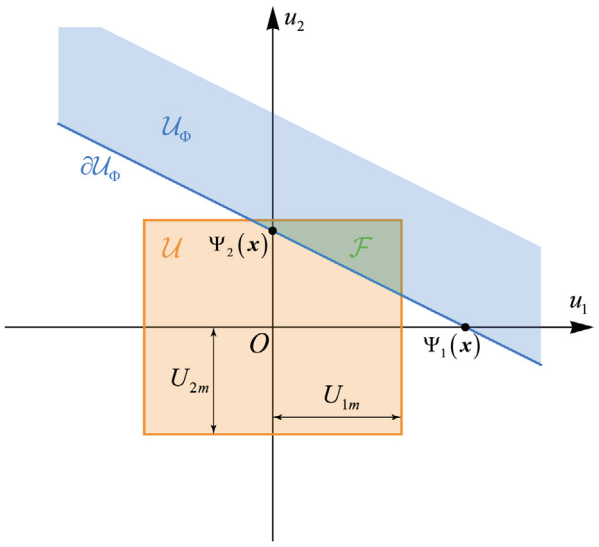

Fig. 3. Illustration of feasible set $\mathcal{F}$, admissible set $\mathcal{U}$ and trajectory index set $\mathcal{U}_\Phi$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.2. Augmented trajectory index

Sometimes, the original TIC optimization problem (15) may be infeasible, i.e., the feasible set $\mathcal{F} = \mathcal{U}_\Phi \cap \mathcal{U} = \emptyset$ in (15b). This condition can be visualized with an intuitive illustration of $\mathcal{F}$ as shown in Fig. 3. For the two-axis system, the admissible set $\mathcal{U}$ is a bounded box as the orange rectangle shown in Fig. 3. The trajectory index set $\mathcal{U}_\Phi$ is a half plane as the blue region shown in Fig. 3, where the boundary is represented by the line $\partial\mathcal{U}_\Phi = \{\boldsymbol{u} : \nabla\Phi^\top \boldsymbol{A}\boldsymbol{x} + \nabla\Phi^\top \boldsymbol{B}\boldsymbol{u} + k_1\Phi = 0\}$. If there is no intersection between $\partial\mathcal{U}_\Phi$ and $\mathcal{U}$, then the feasible set $\mathcal{F} = \emptyset$. Therefore, some kinds of augmented control law must be applied to guarantee the intersection between $\partial\mathcal{U}_\Phi$ and $\mathcal{U}$.

It is noticed that the line $\partial\mathcal{U}_\Phi$ intersects the u_1 axis and u_2 axis at

$$\begin{aligned}\Psi_1(\boldsymbol{x}) &= -\left(\frac{\partial\Phi}{\partial x_1}x_2 + \frac{\partial\Phi}{\partial x_3}x_4 + k_1\Phi\right)/\frac{\partial\Phi}{\partial x_2}\\ \Psi_2(\boldsymbol{x}) &= -\left(\frac{\partial\Phi}{\partial x_1}x_2 + \frac{\partial\Phi}{\partial x_3}x_4 + k_1\Phi\right)/\frac{\partial\Phi}{\partial x_4}\end{aligned} \tag{16}$$

Obviously, if $\Psi_1(\boldsymbol{x}) \in \mathcal{U}$ or $\Psi_2(\boldsymbol{x}) \in \mathcal{U}$, then the line $\partial\mathcal{U}_\Phi$ always has intersection with $\mathcal{U}$, i.e., the feasible set $\mathcal{F} \neq \emptyset$. Define the *augmented trajectory index* Ψ, where

$$\Psi(\boldsymbol{x}) = \begin{cases} \Psi_1(\boldsymbol{x})/U_{1m}, & \text{if } |\Psi_1(\boldsymbol{x})|/U_{1m} \le |\Psi_2(\boldsymbol{x})|/U_{2m} \\ \Psi_2(\boldsymbol{x})/U_{2m} & \text{if } |\Psi_1(\boldsymbol{x})|/U_{1m} > |\Psi_2(\boldsymbol{x})|/U_{2m} \end{cases} \tag{17}$$

Thus the feasibility condition $\mathcal{F} \neq \emptyset$ can be transcribed into $\boldsymbol{x} \in \mathcal{X}_\Psi = \{\boldsymbol{x} : |\Psi(\boldsymbol{x})| \le 1\}$. And certain augmented control law can be designed to guarantee $\boldsymbol{x} \in \mathcal{X}_\Psi$.

Considering the derivative of $\Psi(\boldsymbol{x})$

$$\begin{aligned}\dot{\Psi} &= \nabla\Psi^\top \boldsymbol{A}\boldsymbol{x} + \nabla\Psi^\top \boldsymbol{B}\boldsymbol{u}\\ &= \frac{\partial\Psi}{\partial x_1}x_2 + \frac{\partial\Psi}{\partial x_3}x_4 + \frac{\partial\Psi}{\partial x_2}u_1 + \frac{\partial\Psi}{\partial x_4}u_2\end{aligned} \tag{18}$$

Similar to the design of $\mathcal{U}_\Phi$ in (14), the *augmented trajectory index set* $\mathcal{U}_\Psi$ can be designed as follows in order to ensure the invariance of $\mathcal{X}_\Psi$.

$$\begin{aligned}\mathcal{U}_\Psi = \{\boldsymbol{u} :& \nabla\Psi^\top \boldsymbol{A}\boldsymbol{x} + \nabla\Psi^\top \boldsymbol{B}\boldsymbol{u} \le k_2(1-\Psi) \text{ if } \Psi \ge 0,\\ & \nabla\Psi^\top \boldsymbol{A}\boldsymbol{x} + \nabla\Psi^\top \boldsymbol{B}\boldsymbol{u} \ge -k_2(1+\Psi) \text{ if } \Psi < 0\}\end{aligned} \tag{19}$$

where $k_2 > 0$ is a control gain. Therefore, the feasibility condition can be guaranteed through the constraint $\boldsymbol{u} \in \mathcal{U}_\Psi$. Appending this constraint to the TIC optimization problem (15), the augmented TIC optimization problem is obtained

$$\min_{\boldsymbol{u}} \|\boldsymbol{u} - \boldsymbol{u}^*\|_2 \tag{20a}$$

$$\text{s.t.} \quad \boldsymbol{u} \in \mathcal{U}_\Phi, \quad \boldsymbol{u} \in \mathcal{U}_\Psi, \quad \boldsymbol{u} \in \mathcal{U} \tag{20b}$$

The constraint $\boldsymbol{u} \in \mathcal{U}_\Psi$ is linear, thus the problem (20) is still a QP problem, which preserving the efficient computation advantage in real-time applications.

Remark 2. If the path can be represented by explicit equation $x_3 = g(x_1)$, or transformed into explicit equation locally, one have $\frac{\partial\Phi}{\partial x_4} = 1$. The problem can be simplified if one only focus on the intersection of $\partial\mathcal{U}_\Phi$ on the u_2 axis, i.e., $\Psi(\boldsymbol{x}) = \Psi_2(\boldsymbol{x}) = -\left(\frac{\partial\Phi}{\partial x_1}x_2 + \frac{\partial\Phi}{\partial x_3}x_4 + k_1\Phi\right)$. And the feasibility condition is $\boldsymbol{x} \in \mathcal{X}_\Psi = \{\boldsymbol{x} : |\frac{\partial\Phi}{\partial x_1}x_2 + \frac{\partial\Phi}{\partial x_3}x_4 + k_1\Phi|/U_{2m} \le 1\}$. Accordingly, $\mathcal{U}_\Psi$ can be further simplified. $\nabla\Psi$ can be calculated as

$$\nabla\Psi = -\begin{bmatrix} \frac{\partial^2\Phi}{\partial x_1^2}x_2 + \frac{\partial^2\Phi}{\partial x_1\partial x_3}x_4 + k_1\frac{\partial\Phi}{\partial x_1} \\ \frac{\partial^2\Phi}{\partial x_2\partial x_1}x_2 + \frac{\partial\Phi}{\partial x_1} + \frac{\partial^2\Phi}{\partial x_2\partial x_3}x_4 + k_1\frac{\partial\Phi}{\partial x_2} \\ \frac{\partial^2\Phi}{\partial x_3\partial x_1}x_2 + \frac{\partial^2\Phi}{\partial x_3^2}x_4 + k_1\frac{\partial\Phi}{\partial x_3} \\ \frac{\partial^2\Phi}{\partial x_4\partial x_1}x_2 + \frac{\partial^2\Phi}{\partial x_4\partial x_3}x_4 + \frac{\partial\Phi}{\partial x_3} \end{bmatrix} \tag{21}$$

Remark 3. Sometimes, the constraint (20b) may be infeasible due to discrete calculation, i.e., $\mathcal{U}_\Phi \cap \mathcal{U}_\Psi \cap \mathcal{U} = \emptyset$. At that time, the $\mathcal{U}$ can be neglected for a moment. Then, one have $\mathcal{U}_\Phi \cap \mathcal{U}_\Psi \neq \emptyset$ since $\mathcal{U}_\Phi$ and $\mathcal{U}_\Psi$ are linear half plane regions. This modification guarantees the feasibility of (20). However, the resulting $\boldsymbol{u}_{TIC}$ may exceed the boundary for a moment. But in the experimental investigations, the excess can be tuned through k_2. No excess or acceptable excess performance can be tuned in an efficient way. The detailed theoretical analyses on the feasibility will be studied in further researches.

5. TOS-TIC real-time planning framework

Unifying the TOS time-optimal control $\boldsymbol{u}^*$ synthesized in Section 3 and the TIC coordination mechanism in Section 4, the TOS-TIC real-time planning framework is proposed as illustrated in Fig. 4. The servo system consists of two independent closed-loop position tracking system. The servo controllers take position errors e_1, e_2 as input and generate current control signals i_1, i_2 for the plant. Common feedback controllers such as PID controller can be applied here. The TOS-TIC real-time planner provides the reference position signal x_1^r, x_3^r for the servo system at every control cycle. Depending on whether the real state feedback is fed to the planner or not, the TOS-TIC real-time planner can be categorized into two types, the open-loop TOS-TIC and the closed-loop TOS-TIC. In Fig. 4, a switch symbol is used to illustrated the difference between open-loop TOS-TIC and closed-loop TOS-TIC.

5.1. Open-loop TOS-TIC real-time planner

For the open-loop TOS-TIC, the virtual state feedback from the virtual two-axis system output is fed to the planner, i.e., $\boldsymbol{x}_{FB} = \boldsymbol{x}^r = [x_1^r, x_2^r, x_3^r, x_4^r]^\top$, as the red thick line shown in Fig. 4. The virtual two-axis system is a double integrator system which takes the TIC control output u_1^{TIC} and u_2^{TIC} as inputs. The TIC control output actually represents the acceleration, thus the virtual system generates the ideally tracked velocity and position output. The open-loop TOS-TIC planner can work separately with real

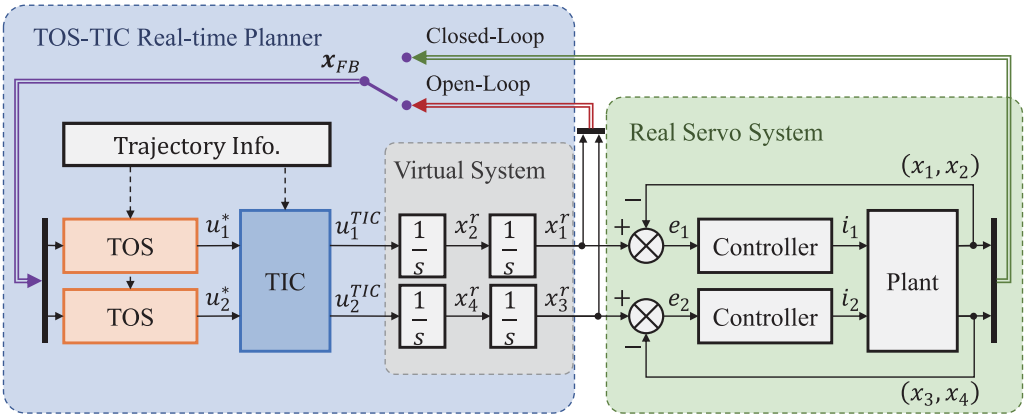

Fig. 4. TOS-TIC real-time planning system framework. Solid lines with arrow are scalar signals, thick hollow lines with arrow are vector signals. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

servo system. It can continuously generates the reference velocity or position for the real servo system. In this paper, the servo controller is working on the position loop, thus the reference position is generated from TOS-TIC planner. The benefit of open-loop TOS-TIC planner is that the design of planner and controller can be independent. For most industrial mechatronic systems, the servo controllers are supposed to be fixed with the hardware. The planner works individually to supply reference trajectories for the servo controller in real-time or non-real-time. Therefore, it is convenient to connect open-loop TOS-TIC planner with these servo systems, which leverages the time optimality and real-time ability of the TOS-TIC planner.

5.2. Closed-loop TOS-TIC real-time planner

For the closed-loop TOS-TIC, the real state feedback from the real two-axis system output is fed to the planner, i.e., $\boldsymbol{x}_{FB} = \boldsymbol{x} = [x_1, x_2, x_3, x_4]^\top$, as the green thick line shown in Fig. 4. Since real state feedback is used, the TOS-TIC planner is actually a part of the closed-loop controller together with the servo controller. The trajectory index $\Phi(\boldsymbol{x}_{FB})$ indicates the actual state deviation from the desired trajectory. The goal of TIC is $\Phi(\boldsymbol{x}_{FB}) \to 0$. Thus the TIC helps minimizing the actual trajectory deviation, i.e., minimizing the contour error as mentioned in Section 4. Here, the TOS-TIC is more like a controller rather than a planner. As a matter of fact, this closed-loop framework unites the planner and controller to fulfil the same goal of minimizing contour error. This structure is similar to the idea of MPC motion planning [40]. However the overall stability analyses and detailed design instruction need to be studied in future works.

Remark 4. When using closed-loop TOS-TIC planner, the real state feedback signals may contain ripples due to inaccurate tracking and system dynamics. Applying the TIC control in (20) sometimes will lead to chattering problem. Therefore, a smooth modification of (20) as follows is more suitable for closed-loop TOS-TIC planner.

$$\min \|\boldsymbol{u} - \boldsymbol{u}^*\|_2 + \beta(\dot{\Psi} - \dot{\Psi}_{k-1})^2 \quad \text{s.t.} \tag{22a}$$

$$\boldsymbol{u} \in \mathcal{U}_\Phi, \quad \boldsymbol{u} \in \mathcal{U}_\Psi, \quad \boldsymbol{u} \in \mathcal{U} \tag{22b}$$

where $\dot{\Psi}_{k-1}$ is the derivative of Ψ in the last control cycle, and $\beta > 0$ is a weighting factor. The objective function (22a) penalizes the chattering of $\dot{\Psi}$, which will attenuate the chattering problem.

6. Experimental investigation

6.1. Experimental setup

To verify the effectiveness of the proposed TOS-TIC strategy, comparative experiments are implemented on an industrial X-Y iron-core linear-motor-driven stage as shown in Fig. 5(a). The X-axis is regarded as the axis-1 while the Y-axis is regarded as the axis-2. The proposed strategy and algorithms are implemented in Matlab/Simulink environment and real-time program is automatically operated in a dSPACE DS1202 control system. The position sensor is a linear encoder with resolution of 0.15625 μm after quadrature. And the controller board executes algorithms at a sampling frequency of $f_s = 5$ kHz, i.e., the control cycle time is $T_s = 200$ μs.

Different types of trajectories are utilized to test the trajectory adaptation ability of the proposed strategy. Without loss of generality, three kinds of trajectories are chosen, including sinusoidal curve, squircle curve and NURBS curve. It is noted that NURBS is the abbreviation of non-uniform rational B-spline, which is widely used in computer-aided design (CAD), manufacturing (CAM), and engineering (CAE). The equations of these reference trajectories are

1. **Sinusoidal Curve**:

$$x_3 - 0.05(1 - \cos 20\pi x_1) = 0, \ \ x_1 \in [-0.1, 0.1] \tag{23}$$

2. **Squircle Curve**:

$$(\frac{x_1}{0.1})^4 + (\frac{x_3}{0.08})^4 = 1 \tag{24}$$

3. **NURBS Curve**: 10 randomized control points are given and the interpolation is performed accordingly.

The image of these reference trajectories are shown in Fig. 5(b). For all trajectories, the axial velocity and acceleration constraints are $V_{1m} = V_{2m} = 0.4$ m/s, $U_{1m} = U_{2m} = 4$ m/s^2.

To evaluate the performance of the proposed TOS-TIC strategy, several well-established methods are also implemented to provide a comparison. Therefore, the following experiment cases are designed.

1. **Case 1**: Strict time-optimal planning using the direct optimization method in [26]. For each reference trajectory, discretize the trajectory with 40000 mesh points and formulate a sparse linear programming problem (LP). Solving this LP offline gives us a strict time-optimal reference for comparison.

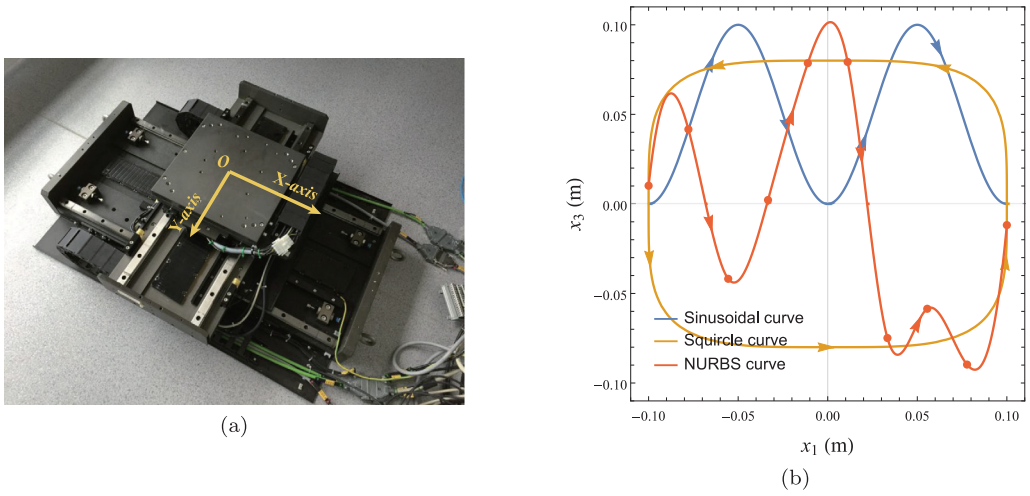

Fig. 5. (a) An industrial X-Y linear-motor-driven stage. (b) Reference trajectories used in experiments, and the point markers are NURBS control points.

2. **Case 2**: Real-time look ahead method in [3], which is widely used in feedrate planning for CNC. This method finds key points by looking ahead the reference trajectory, and connects them with S-shape acceleration/deceleration profiles. However, this method can only constrain the tangential velocity/acceleration of the trajectory instead of axial velocity/acceleration. Given $V_m = 0.4$ m/s, $U_m = 4$ m/s^2 for tangential constraint leads to the excess of axial velocity/acceleration. Thus for fair comparison, a lower tangential constraint value is given to obtain the same constrained axial velocity/acceleration.
3. **Case 3**: The proposed TOS-TIC strategy. Since the Case 1 and Case 2 are trajectory planners providing planned trajectories for the servo controllers, the open-loop structure is applied here as an online trajectory planner for comparison. The TIC in (20) is used and the parameters are $k_1 = 100, k_2 = 45, \alpha = 1, \delta_S = 10^{-8}, \delta_V = 10^{-4}$.

The servo controller for all cases are the same for fair comparison. And the following common performance indexes will be used to evaluate the quality of the planning and final tracking control.

- t_f, the final time, i.e., running time of the planned trajectory.
- $e^{rms} = \sqrt{\frac{1}{t_f}\int_0^{t_f} e(t)^2 dt}$, the root-mean-square value of the tracking error, where $e(t)$ is the tracking error.
- $e^{max} = \max_t |e(t)|$, the maximum absolute value of the tracking error over the concerned experimental segment.

6.2. Comparative experiments results

Firstly, the execution time of TOS-TIC planner during the experiment is tested and plotted as shown in Fig. 6. It is shown in the histogram that most execution time are between 4 μs and 6 μs, and the average execution time of TOS-TIC planner in every control cycle is 3.7818 μs, which only occupies 1.89% of the control cycle time T_s. This result verifies that the proposed TOS-TIC planner is computationally efficient in real-time applications.

Then the comparative experiments with different cases are carried out. The performance indexes are recorded in Table 1. Detailed experimental results are plotted in Fig. 7.

6.2.1. Sinusoidal curve results

From Table 1, it is obvious that the running time $t_f = 1.6042$ s in Case 2 is longer than that in Case 1 ($t_f = 1.4436$ s) and Case 3 ($t_f = 1.4452$ s). And the running time in Case 3 is nearly the same as that in Case 1. Compared with the strict time-optimal results in Case 1, the Case 3 (proposed TOS-TIC planner) increases the trajectory running time by 0.11%, while the Case 2 increases the trajectory running time by 11.12%. This near-time-optimality can also be evidenced with the v_x^{ref} and v_y^{ref} profile in Fig. 7(a). It is seen that the planned velocity profile in Case 3 is nearly the same as the optimal profiles in Case 1.

Another observation is that the planned velocity and acceleration for three cases are within their constraints except some minor excess. The excess of acceleration in Case 1 and Case 3 is due to the numerical calculation errors. There is also a chattering phenomenon on the acceleration profile in Case 3. However, it is neglectable since the acceleration output is fed to the virtual system with two integral step, where the resulting

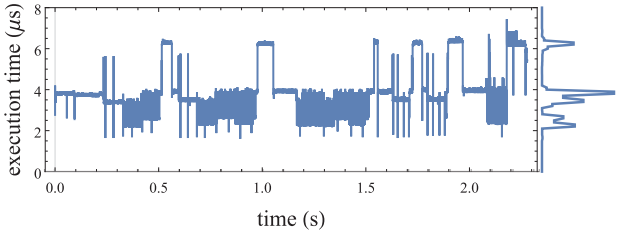

Fig. 6. Execution time of TOS-TIC planner during the experiment. The histogram distribution of the execution time is plotted in the marginal axis.

Table 1
Comparative experiments results.

		Case 1	Case 2	Case 3
Online method		✗	✓	✓
Sinusoidal curve	t_f (s)	1.4436	1.6042	1.4452
	e_x^{rms} (μm)	13.98	12.08	14.17
	e_x^{max} (μm)	105.12	71.34	111.80
	e_y^{rms} (μm)	5.39	4.11	5.14
	e_y^{max} (μm)	23.80	21.40	28.31
Squircle curve	t_f(s)	1.7000	1.7830	1.7074
	e_x^{rms} (μm)	19.84	17.69	18.89
	e_x^{max} (μm)	57.12	68.11	68.99
	e_y^{rms} (μm)	4.23	4.06	4.58
	e_y^{max} (μm)	47.73	35.09	45.92
NURBS curve	t_f(s)	2.2706	2.5604	2.2796
	e_x^{rms} (μm)	6.88	9.57	6.47
	e_x^{max} (μm)	53.55	61.93	46.63
	e_y^{rms} (μm)	6.00	4.35	5.77
	e_y^{max} (μm)	47.63	35.68	43.44

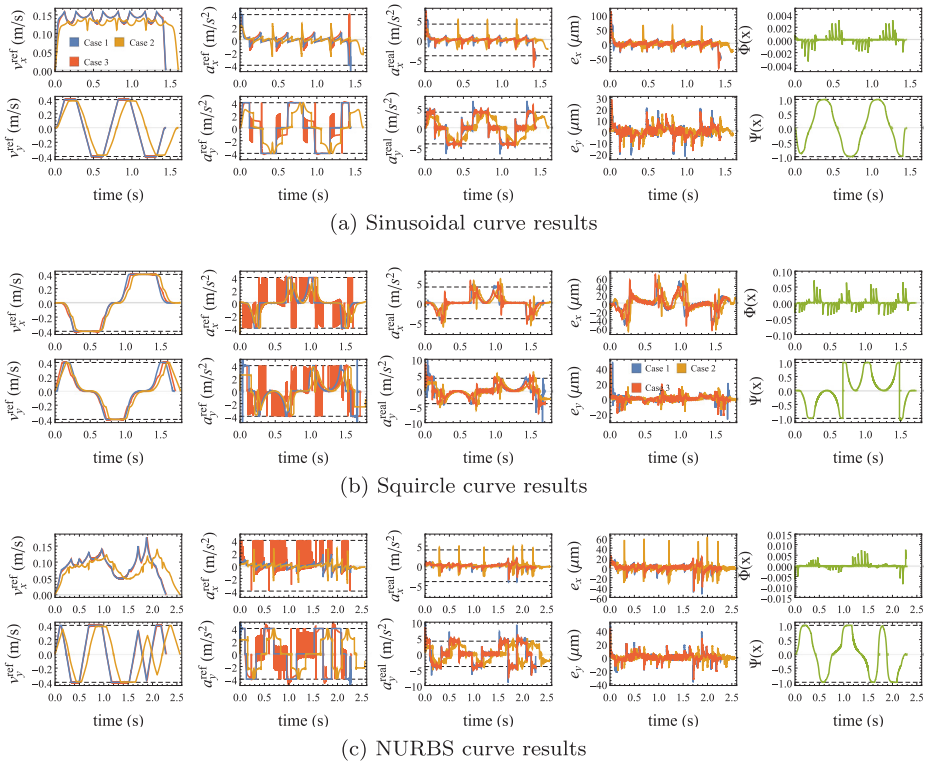

Fig. 7. Planned velocities v_x^{ref}, v_y^{ref}, planned accelerations a_x^{ref}, a_y^{ref}, actual accelerations a_x^{real}, a_y^{real}, tracking errors e_x, e_y and trajectory indexes Φ, Ψ for sinusoidal curve, squircle curve and NURBS curve.

velocity reference and position reference are smooth enough. This smoothed reference will not affect the final tracking error e_x, e_y in Fig. 7(a). It is seen that the final tracking errors in Case 3 ($e_x^{rms} = 14.17\ \mu m, e_y^{rms} = 5.14\ \mu m$) are basically the same as the tracking errors in Case 1 ($e_x^{rms} = 13.98\ \mu m, e_y^{rms} = 5.39\ \mu m$). The reason can be concluded from the actual acceleration profile $a_x^{actual}, a_y^{actual}$ in Fig. 7(a), where the actual acceleration for Case 1 and Case 3 are basically the same. For Case 2, the tracking errors $e_x^{rms} = 12.08\ \mu m, e_y^{rms} = 4.11\ \mu m$ are slightly smaller than Case 1 and Case 3 since the Case 2 uses a conservative tangential constraints to prevent the excess of axial constraints. While some acceleration a_x^{actual} peaks in Case 2 lead to some significant tracking error e_x peaks in Fig. 7(a). The trajectory index $\Phi(\boldsymbol{x})$ keeps converging to zero when it is deviated while $\Psi(\boldsymbol{x})$ is always within ± 1 as expected.

6.2.2. Squircle curve results

Similar to the results in sinusoidal curve, the running time $t_f = 1.7830$ s in Case 2 is longer than that in Case 1 ($t_f = 1.7000$ s) and Case 3 ($t_f = 1.7074$ s). And the running time in Case 3 is nearly the same as that in Case 1. Compared with the strict time-optimal results in Case 1, the Case 3 (proposed TOS-TIC planner) increases the trajectory running time by 0.44%, while the Case 2 increases the trajectory running time by 4.88%. This result also validates the near-time-optimality of the proposed TOS-TIC planner. It is also noticed that the planned velocities v_x^{ref}, v_y^{ref} of Case 1 and Case 3 in Fig. 7(b) have more obvious difference than the sinusoidal curve results. Sometimes the trajectory in Case 1 is faster than Case 3, sometimes it is slower. However this difference is counteracted and their final time are nearly the same.

For the tracking errors plotted in Fig. 7(b), the final tracking errors in Case 3 ($e_x^{rms} = 18.89\ \mu m, e_y^{rms} = 4.58\ \mu m$) are basically the same as the tracking errors in Case 1 ($e_x^{rms} = 19.84\ \mu m, e_y^{rms} = 4.23\ \mu m$), while slightly larger than the tracking errors in Case 2 ($e_x^{rms} = 17.69\ \mu m, e_y^{rms} = 4.06\ \mu m$). The tracking error difference is smaller than the results in sinusoidal curve, which indicates that three cases have similar tracking performances for the squircle curve trajectory. It is also noticed that the chattering of a_x^{ref} and a_y^{ref} in Case 3 will not affect the final tracking error. Similarly, the trajectory index $\Phi(\boldsymbol{x})$ keeps converging to zero when it is deviated while $\Psi(\boldsymbol{x})$ is always within ± 1 as expected. It is also noted that the discontinuity on $\Psi(\boldsymbol{x})$ is due to the switching between Ψ_1 and Ψ_2 in (17).

6.2.3. NURBS curve results

Similarly, the running time $t_f = 2.5604$ s in Case 2 is longer than that in Case 1 ($t_f = 2.2706$ s) and Case 3 ($t_f = 2.2796$ s). And the running time in Case 3 is nearly the same as that in Case 1. Compared with the strict time-optimal results in Case 1, the Case 3 (proposed TOS-TIC planner) increases the trajectory running time by 0.40%, while the Case 2 increases the trajectory running time by 12.76%. This result provides more evidence for the near-time-optimality of the proposed TOS-TIC planner.

The tracking errors, actual velocities and actual accelerations are plotted in Fig. 7(c). The final tracking errors in Case 3 ($e_x^{rms} = 6.47\ \mu m, e_y^{rms} = 5.77\ \mu m$) are basically the same as the tracking errors in Case 1 ($e_x^{rms} = 6.88\ \mu m, e_y^{rms} = 6.00\ \mu m$). However, different from the sinusoidal curve results and squircle curve results, the x-axis tracking error in Case 2 ($e_x^{rms} = 9.57\ \mu m$) is larger than Case 1 and Case 3, while the y-axis tracking error in Case 2 ($e_y^{rms} = 4.35\ \mu m$) is smaller than Case 1 and Case 3 as expected. This reason can be found from the planned acceleration a_x^{ref} and actual acceleration a_x^{actual} in Fig. 7(c), where they contain many acceleration peaks and the time of those peaks are exactly the same as the tracking error e_x peaks. It is also noticed that the chattering and minor excess of a_x^{ref} and a_y^{ref} in Case 3 will not affect the final tracking error. The trajectory index $\Phi(\boldsymbol{x})$ keeps converging to zero when it is deviated while $\Psi(\boldsymbol{x})$ is always within ± 1 as expected.

From the comparative results in Fig. 6, Table 1 and Fig. 7, it is concluded that:

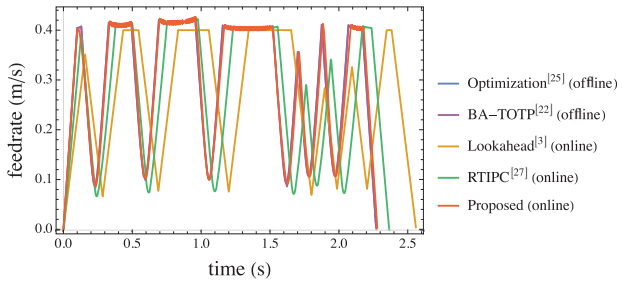

Fig. 8. Feedrate comparison between different methods on the NURBS curve.

1. The proposed TOS-TIC real-time planner is computationally efficient in real-time applications.
2. The planned trajectory from the proposed TOS-TIC planner is near-time-optimal compared with the offline strict time-optimal results.
3. The planned trajectory from the proposed TOS-TIC planner is velocity-constrained and acceleration-constrained as expected.
4. Tracking the planned trajectory from the proposed TOS-TIC planner leads to the similar tracking performance as the offline optimal trajectory.
5. With the proposed TOS-TIC planning framework, the chattering and minor excess on axial acceleration will not affect the final tracking error.
6. With the proposed TOS-TIC planner, the convergence of trajectory index Φ is guaranteed when it is deviated, and the augmented trajectory index Ψ can be constrained.

6.2.4. Similar work comparison

The proposed TOS-TIC planner is further compared with more similar works to validate the effectiveness of the proposed method. The NURBS curve in Fig. 5(b) is used as reference trajectory, the velocity and acceleration constraints are the same as above-mentioned. The following classical and state-of-the-art methods are implemented for comparison.

- **Optimization (offline)** [26]: Strict time-optimal planning using the direct optimization method as in Case 1.
- **BA-TOTP (offline)** [23]: Bidirectional scanning and switching point searching method for time-optimal trajectory planning.
- **Lookahead (online)** [3]: Real-time look ahead method as in Case 2.
- **RTIPC (online)** [28]: Real-time interpolator for parametric curves using feedrate look-ahead and acceleration look-ahead methods.
- **Proposed (online)**: The proposed TOS-TIC strategy as in Case 3.

The tangential feedrate ($\sqrt{{v_x^{ref}}^2 + {v_y^{ref}}^2}$) profiles of the planning results are plotted in Fig. 8. The feedrate depicts the moving speed along the reference trajectory. It is obvious from Fig. 8 that the proposed TOS-TIC is almost the same as all offline optimal results (Optimization and BA-TOTP). Other online methods (Lookahead and RTIPC) are slower than offline methods and the proposed method. These results also verified that the proposed TOS-TIC method generates near-time-optimal results as offline strict-optimal results while being computationally efficient for real-time applications.

7. Conclusion

This paper has proposed a time-optimal switching trajectory index coordination (TOS-TIC) real-time planning framework for the time-optimal planning of continuous multi-axis trajectories with axial acceleration and velocity constraints. A series of trajectory planning experiments on sinusoidal, squircle and NURBS curves have been carried out on an industrial biaxial linear motor stage. The comparative results consistently verified that the proposed method is not only efficient for real-time applications but also near-time-optimal as offline strict-optimal results. Strictly speaking, there are still limitations of this research, such as the feasibility guarantee of the TIC optimization and the theoretical analyses on the closed-loop TOS-TIC real-time planner. Further researches will be focusing on the feasibility guarantee of the TIC optimization problem. Related methods such as Hamilton–Jacobi reachability analyses may be useful for that. Besides, further applications in practical systems such as CNC systems and intelligent robots/vehicles will be carried out, where the proposed TOS-TIC real-time planning strategy will have great potential of applications.

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 in part by the National Natural Science Foundation of China under Grant 51922059 and Grant 51775305, in part by Beijing Natural Science Foundation JQ19010.

References

[1] Haschke R, Weitnauer E, Ritter H. On-line planning of time-optimal, jerk-limited trajectories. In: 2008 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2008, p. 3248–53.
[2] Hu C, Ou T, Zhu Y, Zhu L. GRU-type LARC strategy for precision motion control with accurate tracking error prediction. IEEE Trans Ind Electron 2020;68(1):812–20.
[3] Zhao H, Zhu L, Ding H. A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of short line segments. Int J Mach Tools Manuf 2013;65:88–98.
[4] Zhang Y, Ye P, Zhao M, Zhang H. Dynamic feedrate optimization for parametric toolpath with data-based tracking error prediction. Mech Syst Signal Process 2019;120:221–33.
[5] Biagiotti L, Zanasi R. Time-optimal regulation of a chain of integrators with saturated input and internal variables: An application to trajectory planning. IFAC Proc Vol 2010;43(14):1278–83.
[6] Huang J, Lu Y, Zhu L-M. Real-time feedrate scheduling for five-axis machining by simultaneously planning linear and angular trajectories. Int J Mach Tools Manuf 2018;135:78–96.
[7] Wang Z, Zhou R, Hu C, Zhu Y. Online iterative learning compensation method based on model prediction for trajectory tracking control systems. IEEE Trans Indus Inform 2022;18(1):415–25. http://dx.doi.org/10.1109/TII.2021.3085845.
[8] Zhou R, Hu C, Wang Z, He S, Zhu Y. Nonlinearity compensation and high-frequency flexibility suppression based ric method for precision motion control systems. IEEE Trans Indus Inform 2022. 1-1, http://dx.doi.org/10.1109/TII.2022.3158944.
[9] Ezair B, Tassa T, Shiller Z. Planning high order trajectories with general initial and final conditions and asymmetric bounds. Int J Robot Res 2014;33(6):898–916.
[10] Zhao W-Y, He S, Wen C, Liu C. Contact-rich trajectory generation in confined environments using iterative convex optimization. In: Dynamic systems and control conference, Vol. 84287. American Society of Mechanical Engineers; 2020, p. V002T31A002.
[11] Moreno-Valenzuela J. Time-scaling of trajectories for point-to-point robotic tasks. ISA Trans 2006;45(3):407–18. http://dx.doi.org/10.1016/S0019-0578(07)60221-3.
[12] Yao P, Wei Y, Zhao Z. Null-space-based modulated reference trajectory generator for multi-robots formation in obstacle environment. ISA Trans 2021. http://dx.doi.org/10.1016/j.isatra.2021.05.033.

[13] Fehér M, Straka O, Šmídl V. Constrained time-optimal control of double-integrator system and its application in MPC. In: Journal of physics: conference series, Vol. 783. (1):IOP Publishing; 2017, 012024.
[14] Gu W, Cai S, Hu Y, Zhang H, Chen H. Trajectory planning and tracking control of a ground mobile robot:A reconstruction approach towards space vehicle. ISA Trans 2019;87:116–28. http://dx.doi.org/10.1016/j.isatra.2018.11.019.
[15] Li M, Guo C, Yu H, Yuan Y. Line-of-sight-based global finite-time stable path following control of unmanned surface vehicles with actuator saturation. ISA Trans 2021. http://dx.doi.org/10.1016/j.isatra.2021.07.009.
[16] Bisták P. Time sub-optimal control of triple integrator applied to real three-tank hydraulic system. In: International conference on computer aided systems theory. Springer; 2015, p. 25–32.
[17] Peng H, Zhao H, Wang X, Li Y. Robust motion trajectory optimization of overhead cranes based on polynomial chaos expansion. ISA Trans 2021;110:71–85. http://dx.doi.org/10.1016/j.isatra.2020.10.044.
[18] Bobrow JE, Dubowsky S, Gibson JS. Time-optimal control of robotic manipulators along specified paths. Int J Robot Res 1985;4(3):3–17.
[19] Dong J, Ferreira PM, Stori JA. Feed-rate optimization with jerk constraints for generating minimum-time trajectories. Int J Mach Tools Manuf 2007;47(12–13):1941–55.
[20] Zhang K, Yuan C-M, Gao X-S, Li H. A greedy algorithm for feedrate planning of CNC machines along curved tool paths with confined jerk. Robot Comput-Integr Manuf 2012;28(4):472–83.
[21] Constantinescu D, Croft EA. Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. J Robot Syst 2000;17(5):233–49.
[22] Beudaert X, Lavernhe S, Tournier C. Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path. Int J Mach Tools Manuf 2012;57:73–82.
[23] Barnett E, Gosselin C. A bisection algorithm for time-optimal trajectory planning along fully specified paths. IEEE Trans Robot 2021;37(1):131–45. http://dx.doi.org/10.1109/TRO.2020.3010632.
[24] Sencer B, Altintas Y, Croft E. Feed optimization for five-axis CNC machine tools with drive constraints. Int J Mach Tools Manuf 2008;48(7–8):733–45.
[25] Mora PR. On the time-optimal trajectory planning along predetermined geometric paths and optimal control synthesis for trajectory tracking of robot manipulators. University of California, Berkeley; 2013.
[26] Fan W, Gao X-S, Lee C-H, Zhang K, Zhang Q. Time-optimal interpolation for five-axis CNC machining along parametric tool path based on linear programming. Int J Adv Manuf Technol 2013;69(5):1373–88.
[27] Berscheid L, Kroeger T. Jerk-limited real-time trajectory generation with arbitrary target states. In: Proceedings of robotics: Science and systems. Virtual, 2021, http://dx.doi.org/10.15607/RSS.2021.XVII.015.
[28] Zhong W, Luo X, Chang W, Ding F, Cai Y. A real-time interpolator for parametric curves. Int J Mach Tools Manuf 2018;125:133–45. http://dx.doi.org/10.1016/j.ijmachtools.2017.11.010.
[29] Biagiotti L, Zanasi R. Online trajectory planner with constraints on velocity, acceleration and torque. In: 2010 IEEE international symposium on industrial electronics. IEEE; 2010, p. 274–9.
[30] He S, Hu C, Zhu Y, Tomizuka M. Time optimal control of triple integrator with input saturation and full state constraints. Automatica 2020;122:109240. http://dx.doi.org/10.1016/j.automatica.2020.109240.
[31] Hartl RF, Sethi SP, Vickson RG. A survey of the maximum principles for optimal control problems with state constraints. SIAM Rev 1995;37(2):181–218.
[32] Bonnans JF, Hermant A. Second-order analysis for optimal control problems with pure state constraints and mixed control-state constraints. Ann Inst H Poincaré C Anal Non Linéaire 2009;26(2):561–98.
[33] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. In: Autonomous robot vehicles. Springer; 1986, p. 396–404.
[34] Gracia L, Garelli F, Sala A. Reactive sliding-mode algorithm for collision avoidance in robotic systems. IEEE Trans Control Syst Technol 2013;21(6):2391–9.
[35] Ames AD, Coogan S, Egerstedt M, Notomista G, Sreenath K, Tabuada P. Control barrier functions: Theory and applications. In: 2019 18th European control conference. IEEE; 2019, p. 3420–31.
[36] Liu C, Tomizuka M. Control in a safe set: Addressing safety in human-robot interactions. In: Dynamic systems and control conference, Vol. 46209. American Society of Mechanical Engineers; 2014, p. V003T42A003.
[37] Hu C, Ou T, Chang H, Zhu Y, Zhu L. Deep gru neural network prediction and feedforward compensation for precision multiaxis motion control systems. IEEE/ASME Trans Mechatronics 2020;25(3):1377–88.
[38] Wang W, Hu C, Zhou K, Wang Z. Time parameter mapping and contour error precompensation for multi-axis input shaping. IEEE Trans Indus Inform 2022. 1-1, http://dx.doi.org/10.1109/TII.2022.3158960.
[39] Ames AD, Grizzle JW, Tabuada P. Control barrier function based quadratic programs with application to adaptive cruise control. In: 53rd IEEE conference on decision and control. IEEE; 2014, p. 6271–8.
[40] Guo H, Shen C, Zhang H, Chen H, Jia R. Simultaneous trajectory planning and tracking using an MPC method for cyber-physical systems: A case study of obstacle avoidance for an intelligent vehicle. IEEE Trans Ind Inf 2018;14(9):4273–83.