# Complete and Time-Optimal Path-Constrained Trajectory Planning With Torque and Velocity Constraints: Theory and Applications

Peiyao Shen, Xuebo Zhang, *Member, IEEE*, and Yongchun Fang, *Senior Member, IEEE*

***Abstract*—Time optimality and completeness are desired properties for robot trajectory planning, which ensure that robotic systems work reliably and efficiently. However, there is great difficulty to design planning algorithms that simultaneously guarantee time optimality and completeness. In this paper, a complete, time-optimal and real-time algorithm for path-constrained trajectory planning is proposed under both torque and velocity constraints. The time optimality and completeness of the proposed algorithm are theoretically proven with rigorous mathematical analysis. Moreover, the proposed approach is general and applicable on different robotic systems, and two examples are given on a two-link manipulator and an active-caster-based omnidirectional wheeled mobile robot. Comparative simulation and experimental results are provided to demonstrate the superior performance of the proposed algorithm.**

***Index Terms*—Completeness, manipulator, omnidirectional wheeled mobile robot (OWMR), time-optimal trajectory planning.**

## I. Introduction

PLANNING without predefined paths is a complex task [1]–[5]. Therefore, the decoupled planning, also called “path-velocity decomposition” [6], gains popularity for its lower computational complexity. The decoupled planning divides the motion planning into two stages. The first stage uses path planning methods to generate a geometric path satisfying high-level constraints, including obstacle avoidance, curvature, etc. The second stage assigns a feasible or optimal velocity profile to the obtained path from the first stage, under various constraints including velocity [7]–[12], acceleration [7]–[9], [11], jerk [13], force [5], [14], [15], torque [5], [10], [12], [14], [16]–[22], torque rate [17], etc. In this paper, we will focus on the second stage to study the time-optimal path-constrained trajectory planning (TOPCTP).

Time optimality is important to increase the working efficiency of robots, which implies to assign as high as possible path velocity $\dot{s}$ along the given path. To this end, the work in [5] proposes an offline approach with cable tension and motor torque constraints for cable-driven parallel mechanisms. The works in [14], [23], and [24] use dynamic programming techniques on the grid plane $(s, \dot{s})$ to guarantee the optimality of generated velocity profiles. With convex optimization techniques, a time-optimal velocity profile along the given path, is also obtained in [7], [16], [25], and [26]. The work in [17] proposes to generate a smooth and time-optimal velocity profile with torque and torque rate constraints along specified paths. In task space, quintic polynomials are used in [27] to yield optimal trajectories. Alternately, the proposed method [13] obtains a near time-optimal velocity profile with jerk constraints in real time. The work in [28] provides a simultaneous design of an online planner and a control system to generate and track feasible trajectories. Recently, a planning trajectory is obtained for an industrial biaxial gantry by considering acceleration and velocity constraints [29]–[31]. It is highly computationally efficient since the optimal solution is analytical, and thus shows the obvious advantages comparing with traditional numerical trajectory planning schemes. The work in [8] proposes an online planner in the operational space to generate a feasible velocity profile with kinematic and dynamic constraints. Among these works, the numerical integration (NI) method [18], [19] is able to generate the time-optimal velocity profile along the given path in a real-time manner [22], which attracts significant attention.

The original version of NI [18], [19] aims to generate a time-optimal velocity profile along a specified path in the presence of only torque constraints for industrial manipulators. Based on Pontryagin maximum principle, TOPCTP possesses a bang–bang structure of torque inputs, thus, the key problem is the computation of switch points, which is successfully solved by NI. The accelerating and decelerating curves, integrated from those switch points, constitute the time-optimal velocity profile. The works in [20]–[22], and [32] provide the detection methods for three types of switch points: tangent, discontinuity, and zero-inertia points. In addition to manipulators [9], [15], NI is also applied to spacecrafts [33] and humanoid robots [34]. In real applications, velocity constraints should also be considered. Under torque/acceleration and velocity constraints, the works in

Manuscript received April 25, 2017; revised August 21, 2017 and January 19, 2018; accepted February 3, 2018. Date of publication February 28, 2018; date of current version April 16, 2018. Recommended by Technical Editor Y. Guo. This work was supported by the National Natural Science Foundation of China under Grant 61573195 and Grant U1613210. *(Corresponding author: Xuebo Zhang.)*

The authors are with the Institute of Robotics and Automatic Information System, Nankai University, Tianjin 300071, China, and also with the Tianjin Key Laboratory of Intelligent Robotics, Nankai University, Tianjin 300071, China (e-mail: 1120140100@mail.nankai.edu.cn; zhangxuebo@nankai.edu.cn; fangyc@nankai.edu.cn).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2018.2810828

[10]–[12] indicate that the original NI method [18] that only considers torque constraints may fail to output a feasible velocity profile. The works in [10] and [11] independently suggest their methods to generate feasible velocity profiles, yet time optimality and completeness proofs of these methods have not been reported in existing literatures. In robot motion planning, the completeness of an algorithm guarantees that a feasible solution is returned when one exists, otherwise failure is reported in a finite time [35]. If a planning algorithm is not complete, then this algorithm may return failure for a specific problem that actually has feasible solutions, or it may not return failure in a finite time for a problem that has no feasible solutions. Thus, an incomplete planning algorithm may seriously restrict the work capability of robots in complex environments, even leading to unsafe motions, such as a missing grasping [36] and crossroad collision [37]. It is important and highly demanded to design a complete and time-optimal trajectory planning algorithm.

For the TOPCTP problem, this paper proposes a complete, time-optimal, and real-time algorithm under both torque and velocity constraints (CTOA-TV), along with rigorous mathematical proofs on completeness and optimality. In particular, the proposed approach introduces a novel concept of “switch arc” on the part of the speed limit curve determined by velocity constraints, which presents an appealing property for decelerating curves with rigorous mathematical proof. Then, these switch arcs are used together with switch points to generate time-optimal velocity profiles. With the aid of these switch arcs, it is proven that under both torque and velocity constraints, the proposed approach can guarantee both completeness and time optimality. Several simulation and experimental results on a two-link manipulator and an active-caster-based omnidirectional wheeled mobile robot (OWMR) are provided to verify the proposed approach and related properties. The major contributions of this paper are as follows.

1) A novel concept of “switch arc” is defined on the part of the speed limit curve decided by velocity constraints, and a property of decelerating curves from switch arcs is discovered and proven. Thus, a real-time, complete, and TOPCTP algorithm is proposed.
2) Time optimality and completeness of the proposed algorithm are shown with rigorous mathematical proofs. These theoretical results, lacked in the previous works, make the proposed algorithm more rigorous and help users to construct a more safe motion planning system that could report a time-optimal solution or a failure in a real-time manner.
3) The proposed algorithm is general and it can be applied on different robotic systems to generate time-optimal velocity profiles along specified paths. This is demonstrated by simulation and experiments on two example systems: two-link manipulators and active-caster-based OWMRs.

This paper is organized as follows. Section II introduces related works of NI under both torque and velocity constraints. In Section III, the switch arc is newly defined, and the CTOA-TV using switch arcs is proposed to generate time-optimal trajectories under torque and velocity constraints. The mathematical proofs on completeness and optimality of CTOA-TV are provided in Section IV. The detailed application examples on manipulators and OWMRs are provided in Section V. Simulation and experimental results are provided in Section VI. Some conclusions are given in Section VII.

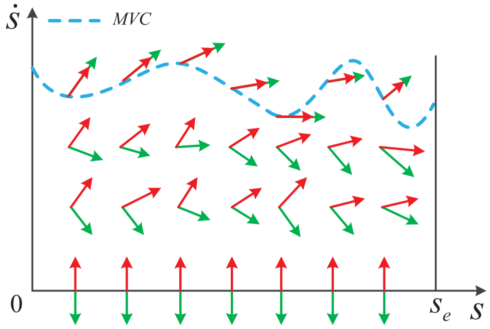


Fig. 1. Upper and lower limits of path acceleration $\ddot{s}$ on the plane $(s, \dot{s})$: red arrows represent upper limits and green arrows represent lower limits. The symbol $s_e$ is the total length of a specified path.

## II. Related Works on NI with Torque and Velocity Constraints

### A. NI with Torque Constraints

In original NI [18], [19] with only torque constraints, along a specified path, the robot model is parameterized by $s(t)$ instead of time $t$, with $s$ being the path coordinate on the given path. Accordingly, the time-optimal trajectory planning is simplified as a planning problem in two dimensional path parameterization space $(s, \dot{s})$, with $\dot{s} = ds/dt$ being the path velocity. Then, torque constraints of robotic systems are formulated as an inequality in terms of $s, \dot{s}, \ddot{s}$ as follows:

$$\alpha(s, \dot{s}) \leq \ddot{s} \leq \beta(s, \dot{s}) \tag{1}$$

where $\ddot{s}$ is the path acceleration, $\alpha(s, \dot{s})$ and $\beta(s, \dot{s})$ denote the minimum path acceleration and maximum path acceleration, respectively. The path acceleration $\ddot{s} = \alpha(s, \dot{s})$ or $\ddot{s} = \beta(s, \dot{s})$ indicates that some of actuators touch torque bounds. The more detailed description of (1) can be found in [12]. On the phase plane $(s, \dot{s})$, $\alpha(s, \dot{s})$, and $\beta(s, \dot{s})$ are depicted as a pair of arrows as shown in Fig. 1. The cyan dash line represents maximum velocity curve (MVC), where all pairs of arrows close. The core of NI is the computation of switch points on MVC, which include tangent, discontinuity and zero-inertia points [20]–[22], [32]. From the starting point $(s = 0, \dot{s} = \dot{s}_0)$, terminal point $(s = s_e, \dot{s} = \dot{s}_e)$ and these switch points, accelerating curves are integrated forward with maximum path acceleration $\beta(s, \dot{s})$ (red solid curves $\beta_1, \beta_2$ in Fig. 2), and decelerating curves are integrated backward with minimum path acceleration $\alpha(s, \dot{s})$ (green solid curves $\alpha_1, \alpha_2$ in Fig. 2). Finally, these accelerating and decelerating curves constitute a time-optimal velocity profile on the plane $(s, \dot{s})$. In this paper, the terms “minimum path acceleration” and “maximum path deceleration” have the same meaning of the lower bound for path acceleration $\ddot{s}$. In addition, as lower and upper bounds of $\ddot{s}$, $\alpha(s, \dot{s})$, and $\beta(s, \dot{s})$ may possess the same sign, which has been mentioned in [18] and [19].

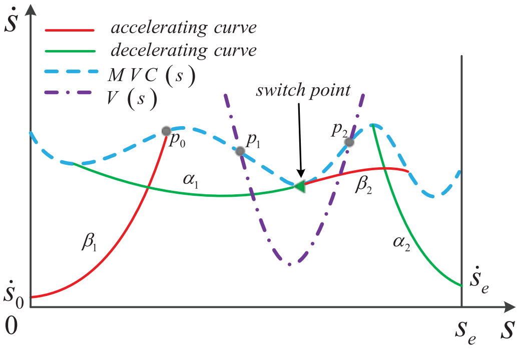


Fig. 2. Failure case: when velocity constraints are considered, the original version of NI with torque constraints fails to output a feasible velocity profile. The symbols $\dot{s}_0, \dot{s}_e$ are, respectively, starting path velocity and terminal path velocity. The cyan dash line represents the limit curve with torque constraints, and the purple dash-dot line is the limit curve with velocity constraints.

## B. NI With Torque and Velocity Constraints

The original NI generates a time-optimal velocity profile with only torque constraints. Yet, when velocity constraints are also considered, it may fail to output a feasible trajectory [10]–[12]. The concrete failure conditions are provided with detailed mathematical proofs in [12]. Specifically, due to velocity constraints $V(s)$, the MVC is altered as

$$\text{MVC}^*(s) = \min\left(\text{MVC}\,(s), V(s)\right), \quad s \in [0, s_e] \tag{2}$$

where $s_e$ is the total length of the given path. In addition, the part of MVC$^*$, which is different from MVC, is described as

$$\text{MVC}^\dagger(s) = \{\text{MVC}^*(s)|\text{MVC}^*(s)<\text{MVC}(s), s \in [0, s_e]\}. \tag{3}$$

For instance, the purple dash-dot curve $\widehat{p_1p_2}$ is MVC$^\dagger$ in Fig. 2. It is shown that, switch points are nonexistent on MVC$^\dagger$, thus it breaks the intersection between the accelerating and decelerating curves, which results in failure of trajectory planning tasks for the original NI with only torque constraints.

In order to generate feasible velocity profiles under torque/acceleration and velocity constraints, the works in [10] and [11] provide some solutions. The trap regions on the plane $(s, \dot{s})$ are discovered in the method [10]. On this basis, the method generates feasible velocity profiles under torque and velocity constraints, and it has been implemented in the codes [22]. The method in [11] requires *two* searching stages to generate feasible velocity profiles: first, it searches switch points along the given path and outputs one feasible velocity profile that only satisfies acceleration constraints; second, taking into account the obtained feasible velocity profile from the first search, together with the velocity constraint curve, it searches velocity characteristic points along the given path again, then a feasible velocity profile is generated. Recently, the work in [38] provides the probabilistic completeness proofs of the "AVP-RRT" method with torque constraints for a kinodynamic motion planning problem, wherein the trajectory planning part uses the approach in [10], yet the completeness of the method [10] is not guaranteed.

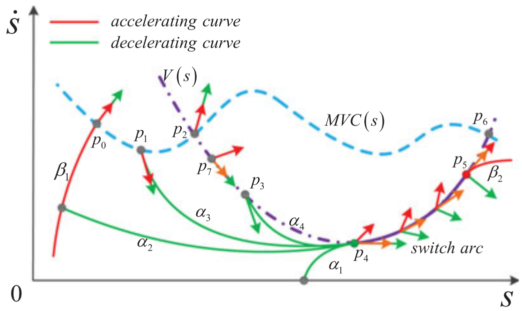


Fig. 3. From the intersection point $p_0$ of $\beta_1$ and MVC$^*$, the switch arc $\widehat{p_4p_5}$ is the first one found among switch points and switch arcs forward along MVC$^*$. The curves $\alpha_1, \alpha_2, \alpha_3$, and$\alpha_4$ are all possible decelerating curves from $p_4$. The cyan dash and purple dash-dot curves are, respectively, MVC$(s)$ with only torque constraints and $V(s)$ with only velocity constraints. The slope of red arrows represents maximum path acceleration $\beta(s, \dot{s})$. The slope of green arrows represents minimum path acceleration $\alpha(s, \dot{s})$. The slope of orange arrows represents path acceleration $\ddot{s}$ of MVC$^\dagger$ (purple dash-dot curve $\widehat{p_2p_6}$).

From the above-mentioned analysis, there is still a lack of a complete and TOPCTP algorithm under both torque and velocity constraints, which is explored in this paper.

# III. Complete and Time-Optimal Trajectory Planning

In this section, a novel concept of "switch arc" is defined on the part of the curve MVC$^\dagger(s)$. With the aid of an appealing property of integrated backward decelerating curves from switch arcs, the CTOA-TV is proposed, which could guarantee the completeness and time optimality.

## A. Switch Arc

*Definition 1:* A switch arc is a continuous segment of MVC$^\dagger$ (3) that satisfies path acceleration constraints (1).

Satisfying path acceleration constraints (1) indicates that, for any point lying on a switch arc, the slope $\ddot{s}$ of the tangent line to MVC$^\dagger$ is between the slopes $[\alpha(s, \dot{s}), \beta(s, \dot{s})]$. As shown in Fig. 3, orange arrows tangent to MVC$^\dagger$ are between green and red arrows on the switch arc (purple solid $\widehat{p_4p_5}$).

It is noted that many switch arcs may exist on the curve MVC$^\dagger$. In addition, the switch arc may degenerate into a single point at which path acceleration constraints are guaranteed.

*Property 1:* From the intersection point of one accelerating curve and MVC$^*$, if a switch arc is the first one found among switch points and switch arcs forward along MVC$^*$, then from its left endpoint, the integrated backward decelerating curve intersects one of lines: the previous accelerating curve, $\dot{s} = 0$, and $s = 0$ with $\dot{s}<\dot{s}_0$.

*Proof:* As shown in Fig. 3, from the intersection point $p_0$ of $\beta_1$ and MVC$^*$, the switch arc $\widehat{p_4p_5}$ is the first one found among switch points and switch arcs forward along MVC$^*$. Then, from the left endpoint $p_4$ of $\widehat{p_4p_5}$, all possible integrated backward decelerating curves are described as follows.

*Case 1:* The decelerating curve $(\alpha_1)$ intersects the line $\dot{s} = 0$, or the line $s = 0$ with $\dot{s}<\dot{s}_0$.

*Case 2:* The decelerating curve ($\alpha_2$) intersects the previous accelerating curve.

*Case 3:* The decelerating curve ($\alpha_3$ or $\alpha_4$) intersects MVC$^*$.

In the following, we will prove that Case 3 cannot occur by contradiction. Assume that Case 3 occurs. Then, the integrated backward decelerating curve $\alpha^*$ from $p_4$ is one of $\alpha_3, \alpha_4$.

When $\alpha^*$ is $\alpha_3$, there exists at least one switch point on $\widehat{p_0p_1}$ due to the continuity of the slope of red and green arrows [38]. It contradicts with the condition that the switch arc $\widehat{p_4p_5}$ is first found along MVC$^*$ from $p_0$.

When $\alpha^*$ is $\alpha_4$, the green arrow at $p_3$ is below MVC$^\dagger$ (the purple dash-dot curve $\widehat{p_2p_6}$). The red and green arrows at $p_2$ exceed MVC, otherwise, there exists at least one switch point on $\widehat{p_0p_2}$, which contradicts with the given condition. Based on the definition of MVC$^\dagger$ in (2), MVC$^\dagger$ is below MVC, which indicates that the red and green arrows exceed MVC$^\dagger$ at $p_2$. Since the slope of green arrows is continuous, there exists at least one point $p_7$ on $\widehat{p_2p_3}$, where the green arrow is tangent to MVC$^\dagger$. Moreover, red and green arrows cannot close at $p_7$ as shown in Fig. 1, which indicates that the slope of the orange tangent arrow to MVC$^\dagger$ satisfies the inequality constraints (1) at $p_7$. Thus, $p_7$ is the point of another switch arc on MVC$^\dagger$, which contradicts with the condition that the switch arc $\widehat{p_4p_5}$ is first found along MVC$^*$ from $p_0$. Note that $\alpha^*$ may hit MVC$^*$ at $p_4$ immediately, and its analysis is same with $\alpha_4$.

From the above-mentioned analysis, this assumption is invalid, and Case 3 cannot occur. Therefore, Case 1 and Case 2 possibly occur for the integrated backward decelerating curve from the left endpoint of the switch arc. ■

*Remark 1:* Property 1 still holds, if the first found switch arc degenerates into one point where (1) is guaranteed. In addition, if the terminal point ($s = s_e, \dot{s} = \dot{s}_e$) or a switch point on MVC$^\dagger$ is first found instead of switch arcs, Property 1 also holds. The corresponding proof is similar with the above-mentioned procedure. In addition, it has been proven that from switch points on MVC with only torque constraints, integrated backward decelerating curves cannot intersect MVC in [38]. ■

## B. Algorithm Procedure

Based on the switch arc in *Definition 1*, detailed procedures of the proposed approach are given in Algorithm 1. The output arguments include "failure" and "solution," wherein the failure indicates that there is no physically feasible velocity profile for TOPCTP, while the solution consisting of accelerating curves, decelerating curves, and switch arcs, represents a time-optimal velocity profile under torque and velocity constraints. The subfunctions are described as.

1) $accProfile(p)$ calculates one accelerating curve by forward direction integral with maximum path acceleration $\beta(s, \dot{s})$ from $p$ or the right endpoint of $p$ if $p$ is a switch arc, until the accelerating curve hits one of lines: MVC$^*$, $\dot{s} = 0$ and $s = s_e$. The hitting point is returned.
2) $decProfile(p)$ calculates one decelerating curve by backward direction integral with minimum path acceleration $\alpha(s, \dot{s})$ from $p$ or the left endpoint of $p$ if $p$ is a switch arc, until the decelerating curve hits one of lines: $\dot{s} = 0$, $s = 0$ with $\dot{s} < \dot{s}_0$, and the previously generated accelerating curve. The hitting point is returned.
3) $searchSwitch(h, \text{MVC}^*)$ searches forward the first switch point or switch arc along MVC$^*$ from the point $h$. If some switch points or switch arcs are found, then the first detected one (the most left one searching from $h$) is returned, otherwise NULL is returned.

**Algorithm 1:** A Complete and Time-Optimal Algorithm with Torque and Velocity Constraints (CTOA-TV).

```
**Input:** $s_e$, $\dot{s}_0$, $\dot{s}_e$, MVC$^*$.
**Output:** a solution or failure.
1:  $p \leftarrow (0, \dot{s}_0)$
2:  $h \leftarrow accProfile(p)$
3:  **if** $h$ is in one of lines: $\dot{s} = 0$, $s = s_e$ with $\dot{s} < \dot{s}_e$ **then**
4:      **return** failure
5:  **else if** $h$ is in the line $s = s_e$ with $\dot{s} \geq \dot{s}_e$ **then**
6:      $p \leftarrow (s_e, \dot{s}_e)$ and **go to 17**
7:  **else if** $h$ is in a switch arc **then**
8:      $p \leftarrow$ the right endpoint of the switch arc, and **go to 2**
9:  **else**
10:     $s \leftarrow searchSwitch(h, \text{MVC}^*)$
11:     **if** $s$ is NULL **then**
12:         $p \leftarrow (s_e, \dot{s}_e)$
13:     **else**
14:         $p \leftarrow s$
15:     **end if**
16: **end if**
17: $h \leftarrow decProfile(p)$
18: **if** $h$ is in one of lines: $\dot{s} = 0$, $s = 0$ with $\dot{s} < \dot{s}_0$ **then**
19:     **return** failure
20: **else if** $p$ is $(s_e, \dot{s}_e)$ **then**
21:     **return** solution
22: **else**
23:     **go to 2**
24: **end if**
```

*Remark 2:* In C++ implementation of the proposed approach, three subfunctions are called in sequence to do simple numerical computation along the given path that is divided into $n$ pieces. Suppose there exists a constant $T$ such that the computational time of each subfunction at each discrete point less than or equal to $T$. The point sets $P_{\text{acc}}, P_{\text{dec}}, P_{\text{switch}}$ consist of discrete points where subfunctions $accProfile, decProfile, switchSearch$ are recalled, respectively. $accProfile$ and $searchSwitch$ both do computation *forward* along the given path. Moreover, the starting point of $P_{\text{acc}}$ is the terminal point of $P_{\text{switch}}$ in term of the line 2–10 in Algorithm 1. Thus, $P_{\text{acc}}$ and $P_{\text{switch}}$ have no overlapped parts. However, $decProfile$ does computation *backward* along the given path, thus $P_{\text{dec}}$ may be overlapped with $P_{\text{acc}}$ or $P_{\text{switch}}$. From the above-mentioned analysis, a solution or failure will be returned by the proposed approach in a fast and finite computational time $\delta \cdot n \cdot T$ with $\delta \in [1, 2]$. In addition, the lines 18 and 19 in Algorithm 1 show that Case 1 in proof of Property 1 is one sufficient condition for the failure of TOPCTP. ■

## IV. Time Optimality and Completeness

In this section, based on Property 1 of switch arcs, it is proven that the generated velocity profiles by the proposed approach are time-optimal. Then, we give sufficient and necessary conditions with mathematical proofs for the proposed approach to generate time-optimal velocity profiles under both torque and velocity constraints. Furthermore, it is proven that under torque and velocity constraints, the proposed approach is complete for TOPCTP.

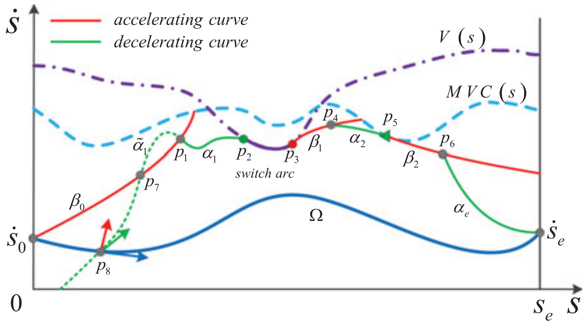


Fig. 4. Proposed approach outputs a time-optimal velocity profile under torque and velocity constraints, which consists of red solid curves ($\beta_0, \beta_1, \beta_2$), green solid curves ($\alpha_1, \alpha_2, \alpha_e$), and the purple solid curve $\widehat{p_2 p_3}$. The cyan dash and purple dash-dot curves are, respectively, MVC($s$) with only torque constraints and $V(s)$ with only velocity constraints. Note that the green dot curve $\tilde{\alpha}_1$ and blue solid curve $\Omega$ are only used in the proofs of Theorem 3 and necessity in Theorem 2. $\tilde{\alpha}_1$ represents the backward extended part of $\alpha_1$, and $\Omega$ represents one possible time-optimal velocity profile for TOPCTP under torque and velocity constraints.

### A. Time Optimality

Under torque and velocity constraints, the obtained solution by Algorithm 1 consists of switch arcs, accelerating curves, and decelerating curves. In this section, the following theorem is presented to prove the optimality of the solution.

*Theorem 1:* Under both torque and velocity constraints, the solution generated by the proposed CTOA-TV in Algorithm 1 is time-optimal.

*Proof:* This theorem is proven by contradiction. It is assumed that on the plane $(s, \dot{s})$ the solution $\Phi(s)$ generated by CTOA-TV is not time-optimal from the starting point $(0, \dot{s}_0)$ to the terminal point $(s_e, \dot{s}_e)$, while $\Psi(s)$ is the optimal trajectory. Then, along the given path, there exists at least one $s = s^*$ where $\Psi(s^*) > \Phi(s^*)$ holds. If $\Phi(s)$ is accelerating at $s^*$, then in order to reach $(s^*, \Psi(s^*))$, $\Psi(s)$ must be accelerating before $s^*$ with the path acceleration $\beta'(s, \dot{s})$ that is greater than $\beta(s, \dot{s})$ in (1). It contradicts with the definition of maximum path acceleration $\beta(s, \dot{s})$. If $\Phi(s)$ is decelerating at $s^*$, then in order to leave $(s^*, \Psi(s^*))$ for $(s_e, \Psi(s_e))$, $\Psi(s)$ must be decelerating after $s^*$ with the path acceleration $\alpha'(s, \dot{s})$ that is less than $\alpha(s, \dot{s})$ in (1). It contradicts with the definition of minimum path acceleration $\alpha(s, \dot{s})$. If $\Phi(s)$ is in a switch arc at $s^*$, then $\Psi(s)$ is beyond MVC* at $s^*$. It contradicts with torque and velocity constraints represented by MVC*. Therefore, this assumption is invalid, and $\Phi(s)$ is time-optimal. ■

### B. Feasibility Conditions and Completeness

We first present Theorem 2 to show working conditions of the proposed approach under torque and velocity constraints. On the basis of Theorem 2, we give the detailed proof on completeness of this approach in Theorem 3.

*Theorem 2:* The sufficient and necessary conditions of the proposed CTOA-TV in Algorithm 1 to generate a time-optimal velocity profile under torque and velocity constraints are given as follows.

$C_1$: Starting from $(s = 0, \dot{s} = \dot{s}_0)$, switch points and right endpoints of switch arcs, all integrated forward accelerating curves do not hit the line $\dot{s} = 0$ and the line $s = s_e$ with $\dot{s} < \dot{s}_e$.

$C_2$: Starting from $(s = s_e, \dot{s} = \dot{s}_e)$, switch points and left endpoints of switch arcs, all integrated backward decelerating curves do not hit the line $\dot{s} = 0$ and the line $s = 0$ with $\dot{s} < \dot{s}_0$.

*Proof:* (*Sufficiency*) In order to prove sufficiency, we analyze procedures of CTOA-TV when both conditions $C_1, C_2$ hold, and it is shown that this algorithm outputs a time-optimal velocity profile $\Phi(s)$ with torque and velocity constraints. According to $C_1$, from the starting point $(s = 0, \dot{s} = \dot{s}_0)$, the integrated forward accelerating curve in $accProfile$ hits either MVC* or the line $s = s_e$ with $\dot{s} \geq \dot{s}_e$. If the line $s = s_e$ with $\dot{s} \geq \dot{s}_e$ is hit, then $decProfile$ is called. In term of $C_2$ and Property 1, the integrated backward decelerating curve from the terminal point $(s = s_e, \dot{s} = \dot{s}_e)$ intersects the previously generated accelerating curve, and the algorithm outputs $\Phi(s)$. If the curve MVC* is hit, then the hitting point is within a switch arc or not. For this hitting point being in the switch arc, starting from the right endpoint of the switch arc, one new accelerating curve is integrated forward in $accProfile$. For this hitting point being not in the switch arc, $searchSwitch$ is called to search forward the first one among switch points and switch arcs along MVC* from the hitting point. When the first one is found, $decProfile$ is called to generate one decelerating curve that intersects the previously generated accelerating curve due to $C_2$ and Property 1, and $accProfile$ is called to generate one new accelerating curve. When no one is found, the integrated backward decelerating curve from $(s = s_e, \dot{s} = \dot{s}_e)$ in $decProfile$ intersects the previously generated accelerating curve to output $\Phi(s)$. For instance, Fig. 4 shows one case of the above-mentioned procedures, wherein $\Phi(s)$ consists of the switch arc $\widehat{p_2 p_3}$, accelerating curves $(\beta_0, \beta_1, \beta_2)$ and decelerating curves $(\alpha_1, \alpha_2, \alpha_e)$. From the above-mentioned analysis, sufficiency is proven.

(*Necessity*) We prove necessity by contradiction. CTOA-TV outputs a time-optimal velocity profile $\Phi(s)$ with torque and velocity constraints. This velocity profile satisfies path acceleration constraints (1), and it consists of accelerating curves, decelerating curves, and switch arcs. It is assumed that one of conditions $C_1, C_2$ does not hold. If $C_1$ is violated, then in order to hit the line $\dot{s} = 0$ or the line $s = s_e$ with $\dot{s} < \dot{s}_e$, there exists at least one integrated forward accelerating curve from $(s = 0, \dot{s} = \dot{s}_0)$, switch points and right endpoints of switch arcs, which intersects $\Phi(s)$. If $C_2$ is violated, then in order to hit the line $\dot{s} = 0$ or the line $s = 0$ with $\dot{s} < \dot{s}_0$, there exists at least one integrated backward decelerating curve from $(s = s_e, \dot{s} = \dot{s}_e)$, switch points and left endpoints of switch arcs, which intersects $\Phi(s)$. At these intersection points, the path acceleration of $\Phi(s)$ violates path constraints (1), which contradicts with $\Phi(s)$ satisfying (1). Therefore, this assumption is invalid, and both conditions $C_1, C_2$ hold. For instance, as shown in Fig. 4, under torque and velocity constraints, this algorithm outputs a time-optimal velocity profile $\beta_0 - \alpha_1 - \widehat{p_2 p_3} - \beta_1 - \alpha_2 - \beta_2 - \alpha_e$. Starting from $p_2$, the integrated backward decelerating curve $\tilde{\alpha_1}$ (the backward extended part of $\alpha_1$) intersects $\beta_0$ at $p_7$ if $C_2$ is violated. At $p_7$, the slope of the accelerating curve $\beta_0$ is less than the slope of the decelerating curve $\tilde{\alpha_1}$, which indicates that (1) is violated. From the above-mentioned analysis, necessity is proven. In summary, this theorem is proven. ∎

*Remark 3:* The conditions $C_1, C_2$ in Theorem 2 have straightforward physical interpretations. The $C_1$ indicates that along the given path, the given $\dot{s}_0$ is too slow to reach the given $\dot{s}_e$ despite of maximum path acceleration $\beta(s, \dot{s})$. The $C_2$ indicates that along the given path, the given $\dot{s}_0$ is too fast to reach the given $\dot{s}_e$ despite of minimum path acceleration $\alpha(s, \dot{s})$. ∎

Based on Theorem 2, the completeness of the proposed approach for TOPCTP is proven as follows.

*Theorem 3:* Under torque and velocity constraints, the proposed CTOA-TV in Algorithm 1 is complete for TOPCTP.

*Proof:* In robot motion planning, an algorithm is complete for a problem when the following two conditions hold [35]:

1) a solution is returned if one exists; and
2) failure is returned in a finite time otherwise.

As shown in Fig. 4, under torque and velocity constraints, for TOPCTP, if a time-optimal velocity profile $\Omega$ exists, then this profile satisfies path acceleration constraints (1), and $\Omega$ is a continuous curve between $(0, \dot{s}_0)$ and $(s_e, \dot{s}_e)$ that is enclosed by the lines: MVC$^*$, $s = 0$, $s = s_e$, and $\dot{s} = 0$. All of switch points, switch arcs, starting point $(0, \dot{s}_0)$, and terminal point $(s_e, \dot{s}_e)$ are on $\Omega$ or above $\Omega$. It is assumed that one of conditions $C_1, C_2$ does not hold. If $C_1$ is violated, then, starting from $(s = 0, \dot{s} = \dot{s}_0)$, switch points and right endpoints of switch arcs, there exists at least one integrated forward accelerating curve that intersects $\Omega$, in order to hit one of lines below $\Omega$: $\dot{s} = 0$, and $s = s_e$ with $\dot{s} < \dot{s}_e$. If $C_2$ is violated, then starting from $(s = s_e, \dot{s} = \dot{s}_e)$, switch points and left endpoints of switch arcs, there exists at least one integrated backward decelerating curve that intersects $\Omega$, in order to hit one of lines below $\Omega$: $\dot{s} = 0$, and $s = 0$ with $\dot{s} < \dot{s}_0$. At these intersection points, the path acceleration of $\Omega$ violates path acceleration constraints (1), which indicates that the assumption is invalid and both conditions $C_1, C_2$ hold if a solution exists for TOPCTP. Based on sufficiency in Theorem 2, a time-optimal velocity profile is returned by CTOA-TV. From these analysis, the condition 1) holds.

As described in Remark 2, a time-optimal velocity profile or failure is obtained by CTOA-TV in a finite time. Therefore, if there is no feasible velocity profile for TOPCTP under torque and velocity constraints, then CTOA-TV will return failure in a finite time. The condition 2) holds. ∎

## V. Applications

This section provides two application examples of the proposed approach on a two-link manipulator and an active-caster-based OWMR.

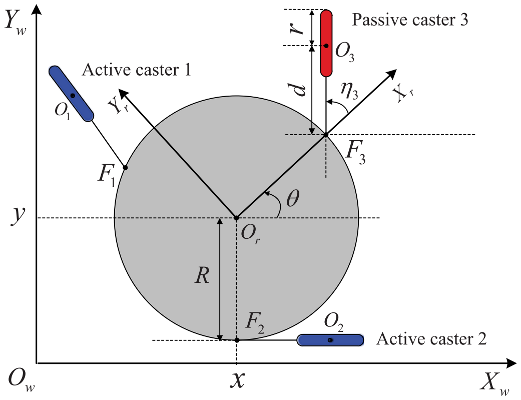


Fig. 5. OWMR with two blue active casters and one red passive caster.

### A. Omnidirectional Wheeled Mobile Robot

Omnidirectional wheeled mobile robots with active casters possess high degree of maneuverability [39], [40]. In Fig. 5, the vehicle has three casters: one passive caster and two active casters. $\mathcal{F}(X_r O_r Y_r)$ and $\mathcal{F}(X_w O_w Y_w)$ represent the robot coordinate system and reference coordinate system, respectively. $R \in \mathbb{R}^+$ represents the radius of the robot frame. $r \in \mathbb{R}^+$ stands for the radius of three wheels. $d \in \mathbb{R}^+$ stands for the length of the steering link $F_i O_i$, $i \in [1, 3]$, wherein $O_i$ is the center point of the wheel. $x \in \mathbb{R}$ and $y \in \mathbb{R}$ represent the position of $O_r$ in $\mathcal{F}_w$, with $\theta \in \mathbb{S}$ being the robot orientation. The angles of three casters $\eta_i \in \mathbb{S}$ are the angles from $O_r F_i$ to $F_i O_i$, $i \in [1, 3]$, with anticlockwise being positive.

The kinematic model of the vehicle [39] is described as

$$\boldsymbol{\omega} = J\dot{\boldsymbol{\xi}} \tag{4}$$

$$\boldsymbol{a} = \dot{\boldsymbol{\omega}} \tag{5}$$

wherein the vector $\boldsymbol{\xi} \in \mathbb{R}^3$ is the pose of the vehicle, and the $4 \times 1$ vectors $\boldsymbol{\omega}, \boldsymbol{a}$ are the velocity and acceleration of active casters, respectively. Other variables are described as

$$\boldsymbol{\xi} = [x \; y \; \theta]^{\mathrm{T}}$$

$$J = [\boldsymbol{J}_1 \;\; \boldsymbol{J}_2 \;\; \boldsymbol{J}_3 \;\; \boldsymbol{J}_4]^{\mathrm{T}}$$

$$\boldsymbol{J}_1 =$$

$$\left[ \frac{-\cos(\theta + \eta_1 + 2\pi/3)}{r} \quad \frac{-\sin(\theta + \eta_1 + 2\pi/3)}{r} \quad \frac{-R\sin(\eta_1)}{r} \right]^{\mathrm{T}}$$

$$\boldsymbol{J}_2 =$$

$$\left[ \frac{\sin(\theta + \eta_1 + 2\pi/3)}{d} \quad \frac{-\cos(\theta + \eta_1 + 2\pi/3)}{d} \quad \frac{-d - R\cos(\eta_1)}{d} \right]^{\mathrm{T}}$$

$$\boldsymbol{J}_3 =$$

$$\left[ \frac{-\cos(\theta + \eta_2 - 2\pi/3)}{r} \quad \frac{-\sin(\theta + \eta_2 - 2\pi/3)}{r} \quad \frac{-R\sin(\eta_2)}{r} \right]^{\mathrm{T}}$$

$$\boldsymbol{J}_4 =$$

$$\left[ \frac{\sin(\theta + \eta_2 - 2\pi/3)}{d} \quad \frac{-\cos(\theta + \eta_2 - 2\pi/3)}{d} \quad \frac{-d - R\cos(\eta_2)}{d} \right]^{\mathrm{T}}$$

where $\boldsymbol{J}_i \in \mathbb{R}^3, i \in [1, 4]$ constitute the matrix $J \in \mathbb{R}^{4 \times 3}$.

Along the specified path, the pose of the vehicle is represented as $\boldsymbol{\xi}(s)$. Taking the time derivative of $\boldsymbol{\xi}(s)$ yields that

$$\dot{\boldsymbol{\xi}} = \boldsymbol{\xi}_s \dot{s} \tag{6}$$

$$\boldsymbol{\xi}_s = [x_s \ y_s \ \theta_s]^{\mathrm{T}} \tag{7}$$

where $x_s = dx/ds, y_s = dy/ds$, and $\theta_s = d\theta/ds$.

Substituting (6) and (7) into (4) and (5), we obtain

$$\boldsymbol{\omega} = J\boldsymbol{\xi}_s \dot{s} \tag{8}$$

$$\boldsymbol{a} = J\boldsymbol{\xi}_s \ddot{s} + (J^{\dagger}\boldsymbol{\xi}_s + J\boldsymbol{\xi}_{ss})\dot{s}^2 \tag{9}$$

where $\boldsymbol{\xi}_{ss} = \partial\boldsymbol{\xi}_s/\partial s$, and the $4 \times 3$ matrix $J^{\dagger} = \partial J/\partial s$.

During the motion along the specified path, due to nonholonomic constraints of each active caster, there are only two possible angles for each active caster: $\eta_i$ or $\eta_i + \pi$. The angles $\eta_1, \eta_2$ are computed as

$$\begin{aligned} \eta_1 = \operatorname{atan2}(&-y_s - R\theta_s \cos(\theta + 2\pi/3), -x_s \\ &+ R\theta_s \sin(\theta + 2\pi/3)) - \theta - 2\pi/3 \end{aligned} \tag{10}$$

$$\begin{aligned} \eta_2 = \operatorname{atan2}(&-y_s - R\theta_s \cos(\theta - 2\pi/3), -x_s \\ &+ R\theta_s \sin(\theta - 2\pi/3)) - \theta + 2\pi/3. \end{aligned} \tag{11}$$

Because of the stabilizability of active casters in the motion [40], the angles $\eta_1, \eta_2$ are chosen. In addition, initial angles of active casters at the starting pose with $s = 0$ are arbitrary. However, for convenience, we suppose that the initial angles of active casters equal to $\eta_1, \eta_2$ in (10) and (11) with $s = 0$. Velocity and acceleration constraints of active casters are given as

$$-\boldsymbol{\omega}_{\max} \leq \boldsymbol{\omega} \leq \boldsymbol{\omega}_{\max} \tag{12}$$

$$-\boldsymbol{a}_{\max} \leq \boldsymbol{a} \leq \boldsymbol{a}_{\max} \tag{13}$$

where $\boldsymbol{\omega}_{\max}, \boldsymbol{a}_{\max} \in \mathbb{R}^4$. In order to satisfy acceleration constraints, substituting (9) into (13) yields that

$$\boldsymbol{A}(s)\ddot{s} + \boldsymbol{B}(s)\dot{s}^2 + \boldsymbol{C}(s) \leq \boldsymbol{0} \tag{14}$$

with

$$\begin{aligned} \boldsymbol{A}(s) &= [(J\boldsymbol{\xi}_s)^{\mathrm{T}} \ \ (-J\boldsymbol{\xi}_s)^{\mathrm{T}}]^{\mathrm{T}} \\ \boldsymbol{B}(s) &= [(J^{\dagger}\boldsymbol{\xi}_s + J\boldsymbol{\xi}_{ss})^{\mathrm{T}} \ \ (-J^{\dagger}\boldsymbol{\xi}_s - J\boldsymbol{\xi}_{ss})^{\mathrm{T}}]^{\mathrm{T}} \\ \boldsymbol{C}(s) &= [-\boldsymbol{a}_{\max}^{\mathrm{T}} \ \ -\boldsymbol{a}_{\max}^{\mathrm{T}}]^{\mathrm{T}}. \end{aligned}$$

Then, according to (14), on the plane $(s, \dot{s})$, the limits of path acceleration $\ddot{s}$ are computed as

$$\alpha(s, \dot{s}) = \max\left\{\frac{-B_i(s)\dot{s}^2 - C_i(s)}{A_i(s)} \middle| A_i(s){<}0, i \in [1, 8]\right\}$$

$$\beta(s, \dot{s}) = \min\left\{\frac{-B_i(s)\dot{s}^2 - C_i(s)}{A_i(s)} \middle| A_i(s) > 0, i \in [1, 8]\right\}.$$

In terms of $\alpha(s, \dot{s})$ and $\beta(s, \dot{s})$, the velocity limit curve with acceleration constraints is computed as

$$\mathrm{MVC}(s) = \min\{\dot{s} \geq 0 | \alpha(s, \dot{s}) = \beta(s, \dot{s})\}. \tag{15}$$

In order to satisfy velocity constraints, substituting (8) into (12) yields that

$$\boldsymbol{A}(s)\dot{s} + \boldsymbol{D}(s) \leq \boldsymbol{0} \tag{16}$$

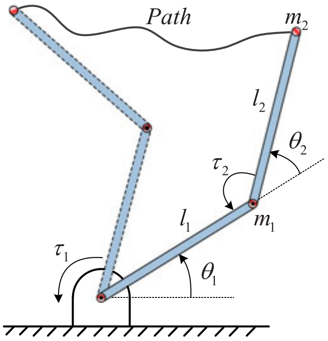


Fig. 6. End-effector motion of a two-link manipulator along a given path.

with

$$\begin{aligned} \boldsymbol{A}(s) &= [(J\boldsymbol{\xi}_s)^{\mathrm{T}} \ \ (-J\boldsymbol{\xi}_s)^{\mathrm{T}}]^{\mathrm{T}} \\ \boldsymbol{D}(s) &= [-\boldsymbol{\omega}_{\max}^{\mathrm{T}} \ \ -\boldsymbol{\omega}_{\max}^{\mathrm{T}}]^{\mathrm{T}}. \end{aligned}$$

According to (16), the velocity limit curve with velocity constraints is computed as

$$V(s) = \min\{-D_i(s)/A_i(s) | A_i(s) > 0, i \in [1, 8]\}. \tag{17}$$

Finally, substituting (15) and (17) into (2) yields the MVC*. With these $\alpha(s, \dot{s}), \beta(s, \dot{s})$, and MVC*, the proposed approach is applied to the OWMR to generate a time-optimal velocity profile under both acceleration and velocity constraints.

## *B. Two-Link Manipulator*

Fig. 6 shows that a two-link manipulator end effector moves along a given path. The $l_1$ and $l_2$ represent the link length. The $\theta_1$ and $\theta_2$ represent the joint angle. For simplicity, it is assumed that the endpoint of each link is its mass point, and they are represented as $m_1$ and $m_2$. The $\tau_1$ and $\tau_2$ represent the joint torque.

The dynamic equation of the manipulator is described as

$$\boldsymbol{\tau} = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) \tag{18}$$

where

$$\begin{aligned} \boldsymbol{\tau} &= [\tau_1 \ \tau_2]^{\mathrm{T}} \\ \boldsymbol{q} &= [\theta_1 \ \theta_2]^{\mathrm{T}} \\ M(\boldsymbol{q}) &= \begin{bmatrix} \Gamma + l_1 l_2 m_2 \cos(\theta_2) + l_1^2(m_1 + m_2) & \Gamma \\ \Gamma & l_2^2 m_2 \end{bmatrix} \\ \boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \begin{bmatrix} -m_2 l_1 l_2 \sin(\theta_2)\dot{\theta}_2^2 - 2m_2 l_1 l_2 \sin(\theta_2)\dot{\theta}_1\dot{\theta}_2 \\ m_2 l_1 l_2 \sin(\theta_2)\dot{\theta}_1^2 \end{bmatrix} \\ \boldsymbol{G}(\boldsymbol{q}) &= \begin{bmatrix} m_2 l_2 g \cos(\theta_1 + \theta_2) + (m_1 + m_2) l_1 g \cos(\theta_1) \\ m_2 l_2 g \cos(\theta_1 + \theta_2) \end{bmatrix} \end{aligned}$$

wherein $\Gamma{=}l_2^2 m_2{+}l_1 l_2 m_2 \cos(\theta_2)$, $g$ is the gravitational acceleration.

For the joints of the manipulator, torque and angular velocity constraints are set as

$$-\boldsymbol{\tau}_{\max} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max} \tag{19}$$

$$-\dot{\boldsymbol{q}}_{\max} \leq \dot{\boldsymbol{q}} \leq \dot{\boldsymbol{q}}_{\max} \tag{20}$$

wherein $\dot{\boldsymbol{q}}_{\max} \in \mathbb{R}^2$ is a constant vector representing the angular velocity boundary, and $\boldsymbol{\tau}_{\max} \in \mathbb{R}^2$ is a constant vector representing the torque boundary.

With inverse kinematic techniques [32], vectors $\boldsymbol{q}, \dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ are represented as

$$\boldsymbol{q} = \boldsymbol{q}(s) \tag{21}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{q}_s \dot{s} \tag{22}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{q}_s \ddot{s} + \boldsymbol{q}_{ss} \dot{s}^2 \tag{23}$$

wherein $\boldsymbol{q}_s = \partial \boldsymbol{q}/\partial s$ and $\boldsymbol{q}_{ss} = \partial \boldsymbol{q}_s/\partial s$. Then, we substitute (18) and (21)–(23) into (19)–(20), and do similar computational procedures with OWMR to obtain the path acceleration limits $\alpha(s, \dot{s}), \beta(s, \dot{s})$, and the MVC$^*$ of manipulators. With these $\alpha(s, \dot{s}), \beta(s, \dot{s})$, MVC$^*$, the proposed approach generates a time-optimal velocity profile for the two-link manipulator under both torque and velocity constraints. In addition, interested readers can follow the same procedures to implement the proposed approach on a six-DOF manipulator.

## VI. Simulation and Experimental Results

In order to verify the proposed approach and related properties, this section provides simulation and experimental results on a two-link manipulator and an active-caster-based OWMR, comparing with the original NI approach [18]. Both simulation and experiment are conducted on an Intel i3 processor at 2.93 GHz. This section chooses $n$th order Bézier curves as the position paths

$$\boldsymbol{P}(\lambda) = \sum_{i=0}^{n} \binom{n}{i} (1-\lambda)^{n-i} \lambda^i \boldsymbol{P}_i \tag{24}$$

where $\binom{n}{i}$ is the binomial coefficient, $\lambda \in [0, 1]$ is the path parameter, and $\boldsymbol{P}_i = [x_i \ y_i]^{\mathrm{T}}$, $i \in [0, n]$ is the path control point. Note that the scalars $\lambda$ and $s$ obey a nonlinear scaling relation. For the choice of the given path, in fact, any smooth paths from existing path planning algorithms can be used in the proposed approach. Without loss of generality, we use Bézier curves because of its simplicity, closed-form expression, and successful applications in existing path planners [41], [42].

### A. Simulation Results on OWMR

In this simulation, the center point $O_r$ of robot moves along a cubic Bézier curve with $n = 3$ in (24). The robot orientation is specified as $\theta(s) = \pi \sin(s)$. The starting and terminal path velocities are set as $\dot{s}_0 = 0.2$ and $\dot{s}_e = 0.4$, respectively. Each element of velocity constraints $\boldsymbol{\omega}_{\max}$ in (12) is set as 18 rad/s and each element of acceleration constraints $\boldsymbol{a}_{\max}$ in (13) is set as 20 rad/s$^2$. In Figs. 7 and 8, cyan dash curves represent MVC$(s)$ with acceleration constraints, and purple dash-dot curves represent $V(s)$ with velocity constraints. According to (2), the MVC is obtained as MVC$^*$, which is represented as the boundary between gray and blank regions.

Simulation results of original NI [18] in Fig. 7 shows that the original NI fails to output a feasible velocity profile along the specified path under both acceleration and velocity constraints.

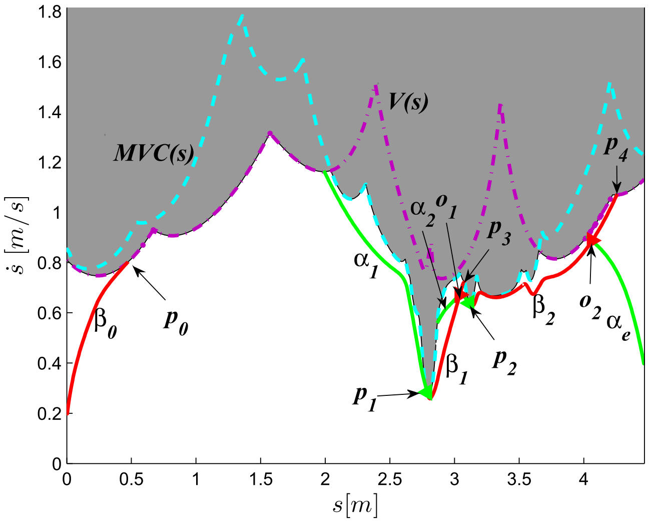


Fig. 7. OWMR: the original NI method fails to output a feasible velocity profile with constraints $\boldsymbol{\omega}_{\max}$ and $\boldsymbol{a}_{\max}$.

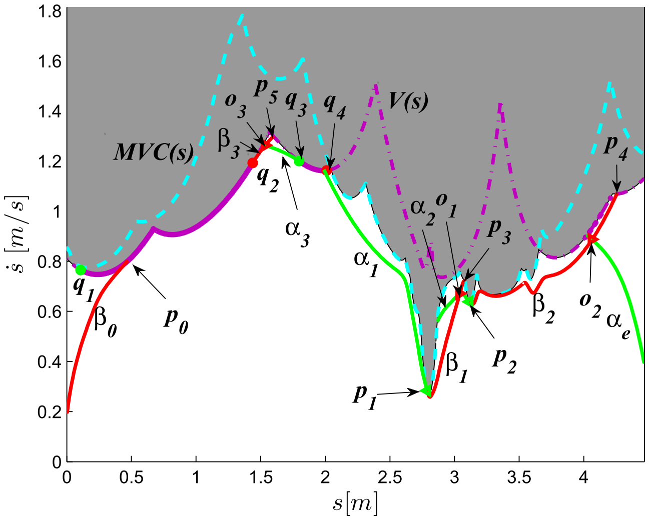


Fig. 8. OWMR: the proposed approach outputs a time-optimal velocity profile with constraints $\boldsymbol{\omega}_{\max}$ and $\boldsymbol{a}_{\max}$.

In Fig. 7, when the accelerating curve $\beta_0$ hits MVC$^*$ at $p_0$, it cannot intersect either the decelerating curve $\alpha_1$ from the switch point $p_1$, the decelerating curve $\alpha_2$ from the switch point $p_2$, or the decelerating curve $\alpha_e$ from the point $(s = s_e, \dot{s} = \dot{s}_e)$. Therefore, the original NI fails to output a feasible velocity profile.

The time cost of the proposed CTOA-TV is 35 ms, which shows good real-time performance. With the aid of switch arcs (purple solid curves: $\widehat{q_1 q_2}$ and $\widehat{q_3 q_4}$ in Fig. 8), the proposed approach generates a time-optimal velocity profile along the specified path under both velocity and acceleration constraints. When the accelerating curve $\beta_0$ hits the switch arc $\widehat{q_1 q_2}$, starting from $q_2$, the integrated forward accelerating curve $\beta_3$ hits MVC$^*$ at $p_5$. Starting from $p_5$, the switch arc $\widehat{q_3 q_4}$ is first found along MVC$^*$. From the point $q_3$, the integrated backward decelerating

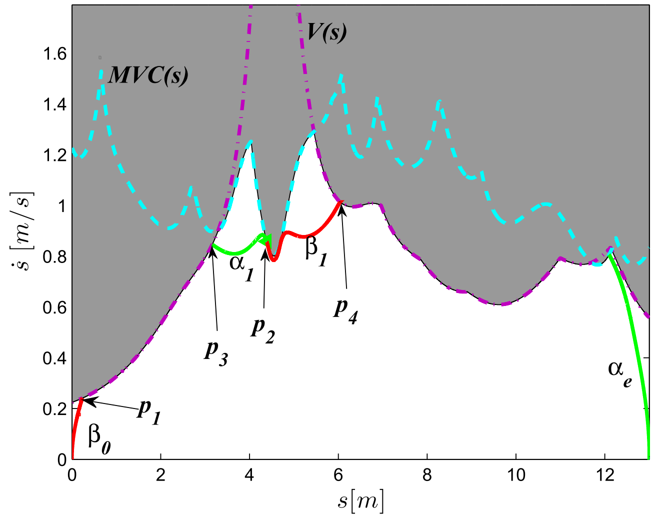


Fig. 9. Manipulator: the original NI method fails to output a feasible velocity profile with constraints $\dot{\boldsymbol{q}}_{\max}$ and $\boldsymbol{\tau}_{\max}$.

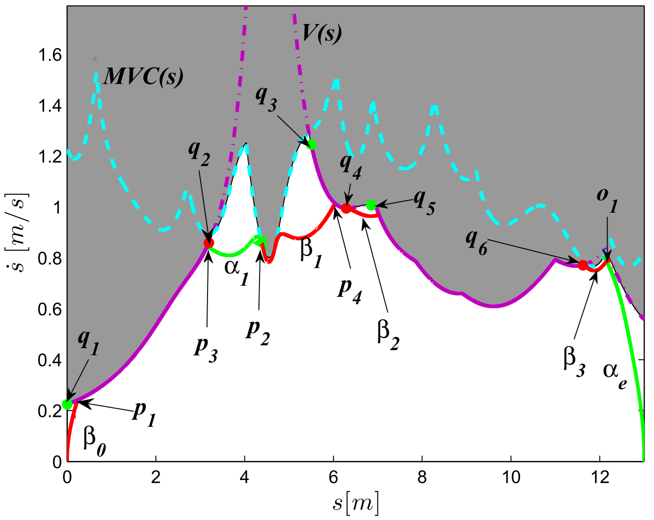


Fig. 10. Manipulator: the proposed approach outputs a time-optimal velocity profile with constraints $\dot{\boldsymbol{q}}_{\max}$ and $\boldsymbol{\tau}_{\max}$.

curve $\alpha_3$ intersects $\beta_3$ at $o_3$, which verifies Property 1. Starting from $q_4$, the integrated forward accelerating curve hits MVC* at $q_4$ immediately. From $q_4$, the switch point $p_1$ is first found along MVC*. The integrated backward decelerating curve $\alpha_1$ from $p_1$ intersects $\widehat{q_3q_4}$. Then, the integrated forward accelerating curve $\beta_1$ from $p_1$ hits MVC* at $p_3$. Staring from $p_3$, the switch point $p_2$ is first found along MVC*. Starting from $p_2$, the integrated backward decelerating curve $\alpha_2$ intersects $\beta_1$ at $o_1$, and the integrated forward accelerating curve $\beta_2$ hits MVC* at $p_4$. From $p_4$, no switch point and switch arc is detected, therefore starting from the point $(s = s_e, \dot{s} = \dot{s}_e)$, the decelerating curve $\alpha_e$ is integrated backward and intersects $\beta_2$ at $o_2$. Finally, the proposed approach generates a time-optimal velocity profile: $\beta_0$-$\widehat{q_1q_2}$-$\beta_3$-$\alpha_3$-$\widehat{q_3q_4}$-$\alpha_1$-$\beta_1$-$\alpha_2$-$\beta_2$-$\alpha_e$. These results on OWMR verify Theorem 2 and Theorem 3.

## B. Simulation Results on Manipulator

In this simulation, the two-link manipulator end effector moves along a fifth-order Bézier curve with $n = 5$ in (24) as shown in Fig. 6. Each element of torque constraints $\boldsymbol{\tau}_{\max}$ in (19) is set as 5 Nm, and each element of velocity constraints $\dot{\boldsymbol{q}}_{\max}$ in (20) is set as 30 rad/s. The starting and terminal path velocities equal to zero. In Figs. 9 and 10, cyan dash curves represent MVC$(s)$ with torque constraints, and purple dash-dot curves represent $V(s)$ with velocity constraints. According to (2), the MVC* is represented as the boundary between gray and blank regions.

Fig. 9 shows that accelerating curves $\beta_0, \beta_1$ and decelerating curves $\alpha_1, \alpha_e$ generated by the original NI do not constitute a feasible velocity profile under both torque and velocity constraints. However, with the aid of switch arcs (purple solid curves: $\widehat{q_1q_2}$, $\widehat{q_3q_4}$ and $\widehat{q_5q_6}$ in Fig. 10), the proposed approach generates a time-optimal velocity profile $\beta_0$-$\widehat{q_1q_2}$-$\alpha_1$-$\beta_1$-$\widehat{q_3q_4}$-$\beta_2$-$\widehat{q_5q_6}$-$\beta_3$-$\alpha_e$. The corresponding computational time is 31 ms. These simulation results on manipulators verify Theorem 2 and Theorem 3 of the proposed approach again.

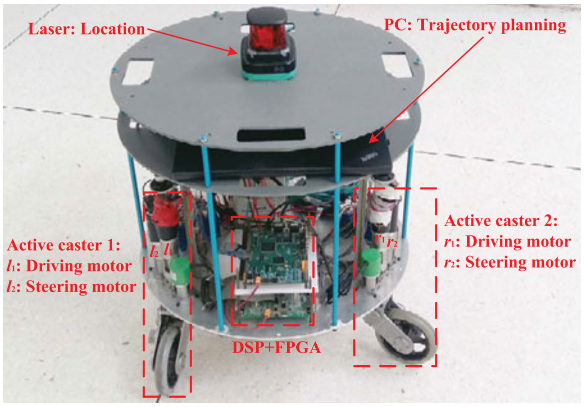


Fig. 11. Experimental platform “NK-OMNI I” [39].

## C. Experimental Results

This section provides experimental results of the proposed approach on an active-caster-based OWMR “NK-OMNI I” [39] in Fig. 11. The robot center point $O_r$ moves along a cubic Bézier curve (see the blue solid curve in Fig. 12), and the robot orientation is set as $\theta = 0$. Each element of velocity constraints $\boldsymbol{\omega}_{\max}$ in (12) is set as 6 rad/s and each element of acceleration constraints $\boldsymbol{a}_{\max}$ in (13) is set as 1 rad/s$^2$. The starting and terminal path velocities equal to zero.

The experimental results show that with the aid of switch arcs defined in Section III-A, the proposed approach generates a time-optimal velocity profile under acceleration and velocity constraints, which consists of one red accelerating curve $\beta_1$, one purple switch arc $\widehat{q_1q_2}$, and one green decelerating curve $\alpha_e$ in Fig. 13. It is worth mentioning that the proposed approach costs 39 ms to output the time-optimal solution, which shows that the proposed approach can be applied in real-time planning tasks.

With a simple proportional-integral-derivative controller, the robot tracks this time-optimal velocity profile, and the tracking errors go to zero in Fig. 14. The velocity and acceleration of active casters are shown in Figs. 15 and 16, which satisfy the

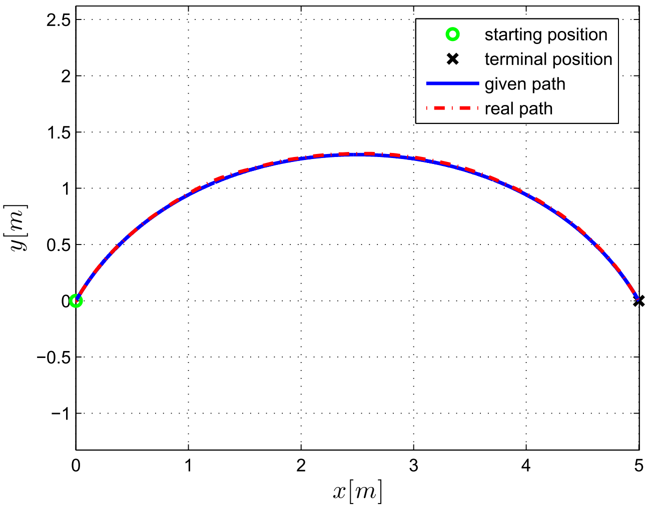


Fig. 12. Experimental path: the blue solid line represents the given path, and the red dash-dot line represents the obtained path from the laser positioning equipment.

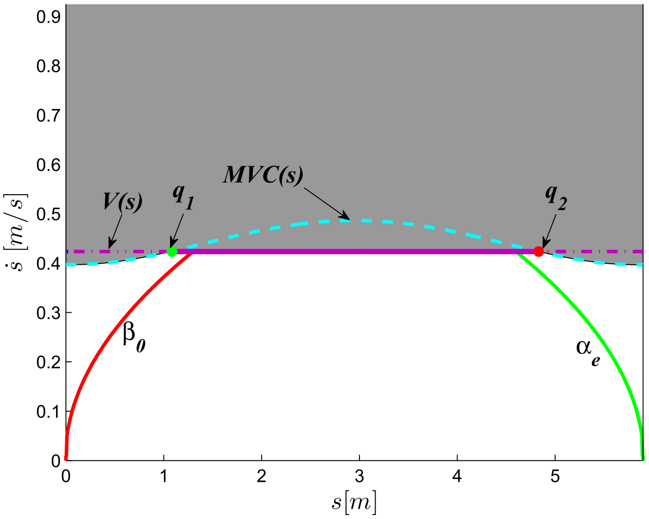


Fig. 13. Proposed approach with constraints $\boldsymbol{\omega}_{\max}$ and $\boldsymbol{a}_{\max}$, generates a time-optimal velocity profile: $\beta_0$-$\widehat{q_1q_2}$-$\alpha_e$.

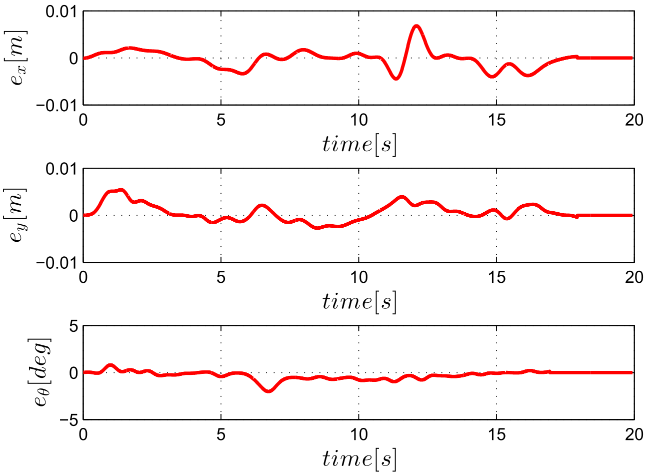


Fig. 14. Tracking errors for $x, y,$ and $\theta$.

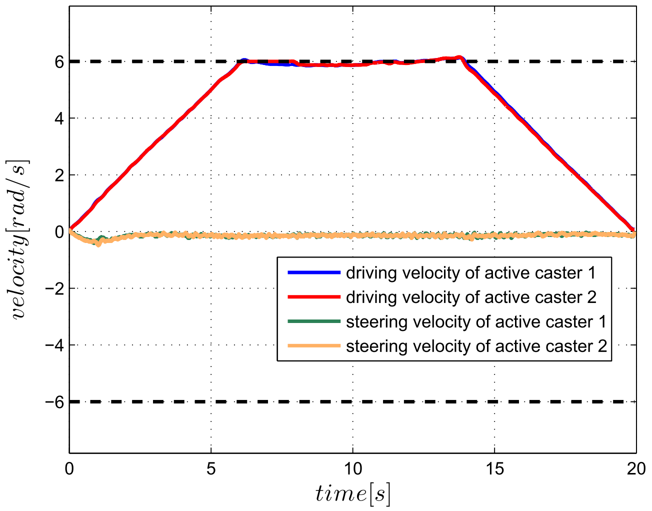


Fig. 15. Velocities of active casters: dark dash lines represent velocity constraints $\boldsymbol{\omega}_{\max}$ and other solid lines represent driving and steering velocities of active casters.

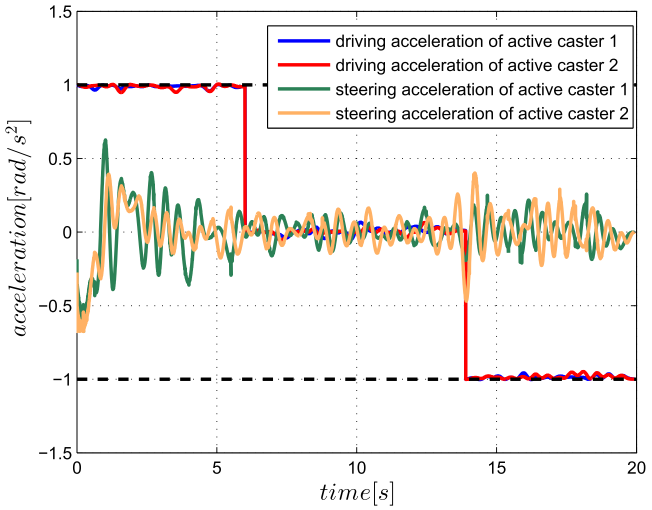


Fig. 16. Accelerations of active casters: dark dash lines represent acceleration constraints $\boldsymbol{a}_{\max}$ and other solid lines represent driving and steering accelerations of active casters.

corresponding constraints. When the robot tracks the accelerating curve $\beta_0$, the driving accelerations of active casters equal to 1 rad/s$^2$. When the robot tracks $\widehat{q_1q_2}$, the driving velocities of active casters are limited by 6 rad/s, and the corresponding driving accelerations are close to zero. When the robot tracks the decelerating curve $\alpha_e$, the driving accelerations of active casters equal to $-1$ rad/s$^2$. From the above-mentioned analysis, some active casters touch either velocity bounds or acceleration bounds, thus the robot achieves time-optimal trajectory planning while satisfying both acceleration and velocity constraints.

## VII. Conclusion

We have proposed a complete, time-optimal and real-time algorithm for path-constrained trajectory planning under both torque and velocity constraints. The contribution is both theoretically and practically important. Theoretically, with rigorous mathematical proofs, we show the completeness and time optimality of the presented algorithm, which guarantee safe and efficient motion of robotic systems in the presence of both torque and velocity constraints. Practically, the presented algorithm is general and it is implemented on different robotic systems to obtain time-optimal trajectories in real time. Comparative simulation and experimental results on manipulators and OWMRs demonstrate the superior performance of the presented algorithm.

## References

[1] C. Lin, P. Chang, and J. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *IEEE Trans. Automat. Control*, vol. AC-28, no. 12, pp. 1066–1074, Dec. 1983.

[2] O. Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Ann. Oper. Res.*, vol. 37, no. 1, pp. 357–373, 1992.

[3] M. Steinbach, H. Bock, and R. Longman, "Time optimal extension and retraction of robots: Numerical analysis of the switching structure," *J. Optim. Theory Appl.*, vol. 84, no. 3, pp. 589–616, 1995.

[4] K. Kim and B. Kim, "Minimum-time trajectory for three-wheeled omnidirectional mobile robots following a bounded-curvature path with a referenced heading profile," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 800–808, Aug. 2011.

[5] E. Barnett and C. Gosselin, "Time-optimal trajectory planning of cable-driven parallel mechanisms for fully-specified paths with G1 discontinuities," *J. Dyn. Syst. Meas. Control*, vol. 137, no. 7, pp. 603–617, 2015.

[6] K. Kant and S. Zucker, "Toward efficient trajectory planning: The velocity decomposition," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 72–89, 1986.

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

[8] C. Bianco and F. Ghilardelli, "Real-time planner in the operational space for the automatic handling of kinematic constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 730–739, Jul. 2014.

[9] T. Kunz and M. Stilman, "Time-optimal trajectory generation for path following with bounded acceleration and velocity," in *Proc. Robot., Sci. Syst.*, pp. 9–13, 2012.

[10] L. Žlajpah, "On time optimal path control of manipulators with bounded joint velocities and torques," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1996, pp. 1572–1577.

[11] F. Lamiraux and J. Laumond, "From paths to trajectories for multi-body mobile robots," in *Proc. 5th Int. Symp. Exp. Robot.*, 1998, pp. 301–309.

[12] P. Shen, X. Zhang, and Y. Fang, "Essential properties of numerical integration for time-optimal trajectory planning along a specified path," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 888–895, Apr. 2017.

[13] S. Macfarlane and E. Croft, "Jerk-bounded manipulator trajectory planning: Design for real-time applications," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 42–52, Feb. 2003.

[14] K. Shin and N. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Trans. Automat. Control*, vol. 31, no. 6, pp. 491–500, Jun. 1986.

[15] S. Behzadipour and A. Khajepour, "Time-optimal trajectory planning in cable-based manipulators," *IEEE Trans. Robot.*, vol. 22, no. 3, pp. 559–563, Jun. 2006.

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

[17] D. Constantinescu and E. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *J. Robot. Syst.*, vol. 17, no. 5, pp. 233–249, 2000.

[18] K. Shin and N. Mckay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Automatic Control*, vol. AC-30, no. 6, pp. 531–541, Jun. 1985.

[19] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 3–17, 1985.

[20] J. Slotine and H. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 118–124, Feb. 1989.

[21] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE J. Robot. Autom.*, vol. 3, no. 2, pp. 115–123, Apr. 1987.

[22] Q. Pham, "A general, fast, and robust implementation of the time-optimal path parameterization algorithm," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1533–1540, 2014.

[23] K. Shin and N. McKay, "Selection of near-minimum time geometric paths for robotic manipulators," *IEEE Trans. Automat. Control*, vol. AC-31, no. 6, pp. 501–511, Jun. 1986.

[24] S. Singh and M. Leu, "Optimal trajectory generation for robotic manipulators using dynamic programming," *J. Dyn. Syst. Meas. Control*, vol. 109, no. 2, pp. 88–96, 1987.

[25] F. Debrouwere *et al.*, "Time-optimal path following for robots with convex-concave constraints using sequential convex programming," *IEEE Trans. Robot.*, vol. 29, no. 6, pp. 1485–1495, Dec. 2013.

[26] A. Singh and K. Krishna, "A class of non-linear time scaling functions for smooth time optimal control along specified paths," in *Proc. 2015 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 5809–5816.

[27] D. Hujic, E. Croft, G. Zak, R. Fenton, J. Mills, and B. Benhabib, "The robotic interception of moving objects in industrial settings: Strategy development and experiment," *IEEE/ASME Trans. Mechatronics*, vol. 3, no. 3, pp. 225–239, Sep. 1998.

[28] A. Matsushita and T. Tsuchiya, "Control system design with on-line planning for a desired signal and its application to robot manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 3, no. 2, pp. 149–152, Jun. 1998.

[29] M. Yuan, Z. Chen, B. Yao, and X. Zhu, "Time optimal contouring control of industrial biaxial gantry: A highly efficient analytical solution of trajectory planning," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 1, pp. 247–257, Feb. 2017.

[30] Z. Chen, B. Yao, and Q. Wang, "$\mu$-synthesis-based adaptive robust control of linear motor driven stages with high-frequency dynamics: A case study," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 3, pp. 1482–1490, Jun. 2015.

[31] Z. Chen, B. Yao, and Q. Wang, "Accurate motion control of linear motors with adaptive robust compensation of nonlinear electromagnetic field effect," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 3, pp. 1122–1129, Jun. 2013.

[32] Z. Shiller, "On singular time-optimal control along specified paths," *IEEE Trans. Robot. Autom.*, vol. 10, no. 4, pp. 561–566, Aug. 1994.

[33] H. Nguyen and Q. Pham, "Time-optimal path parameterization of rigid-body motions: Applications to spacecraft reorientation," *J. Guid. Control Dyn.*, vol. 39, no. 7, pp. 1667–1671, 2016.

[34] Q. Pham and O. Stasse, "Time-optimal path parameterization for redundantly actuated robots: A numerical integration approach," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 6, pp. 3257–3263, 2015.

[35] K. Goldberg, "Completeness in robot motion planning," in *Proc. Workshop Algorithmic Found. Robot.*, San Francisco, CA, USA, pp. 419–429, 1994.

[36] M. Dogar, A. Spielberg, S. Baker, and D. Rus, "Multi-robot grasp planning for sequential assembly operations," in *Proc. 2015 IEEE Int. Conf. Robot. Autom.*, 2015, pp. 193–200.

[37] E. Masehian and M. Amin-Naseri, "Mobile robot online motion planning using generalized Voronoi graphs," *J. Ind. Eng.*, vol. 4, no. 5, pp. 1–15, 2010.

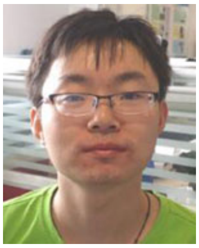
[38] Q. Pham, S. Caron, P. Lertkultanon, and Y. Nakamura, "Admissible velocity propagation: Beyond quasi-static path planning for high-dimensional robots," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 44–67, 2017.

[39] P. Shen, Y. Fang, and X. Zhang, "Trajectory planning of omnidirectional mobile robots with active casters: Multi-motor coordination and singularity avoidance," in *Proc. 2015 IEEE Int. Conf. Cyber Technol. Autom., Control Intell. Syst.*, 2015, pp. 151–156.

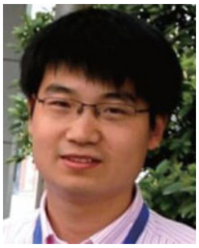
[40] G. Campion, G. Bastin, and B. D'Andréa-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 1, pp. 47–62, Feb. 1996.

[41] Y. Chen, Y. Cai, J. Zheng, and D. Thalmann, "Accurate and efficient approximation of clothoids using Bézier curves for path planning," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1242–1247, Oct. 2017.

[42] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 561–568, Jun. 2010.



**Peiyao Shen** received the B.S. degree in intelligence science and technology from Xidian University, Xi'an, China, in 2014. He is currently working toward the Ph.D. degree at the Institute of Robotics and Automatic Information Systems, Nankai University, Tianjin, China. His research interests include motion planning of omnidirectional wheeled mobile robots.



**Xuebo Zhang** (M'12) received the B.Eng. degree in automation from Tianjin University, Tianjin, China, in 2002, and the Ph.D. degree in control theory and control engineering from Nankai University, Nankai, China, in 2011.

He is currently an Associate Professor with the Institute of Robotics and Automatic Information Systems and also the Tianjin Key Laboratory of Intelligent Robotics, Nankai University. His research interests include mobile robotics, motion planning, and visual servoing.

**Yongchun Fang** (S'00–M'02–SM'08) received the B.S. and M.S. degrees in control theory and applications from Zhejiang University, Hangzhou, China, in 1996 and 1999, respectively, and the Ph.D. degree in electrical engineering from Clemson University, Clemson, SC, USA, in 2002.

From 2002 to 2003, he was a Postdoctoral Fellow with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, USA. He is currently a Professor with the Institute of Robotics and Automatic Information Systems, Nankai University, Tianjin, China. His research interests include nonlinear control, visual servoing, control of underactuated systems, and AFM-based nano-systems.

Dr. Fang is an Associate Editor of the *ASME Journal of Dynamic Systems, Measurement, and Control* and *Journal of Control Theory and Applications*.