Time-Optimal Path Tracking via Reachability Analysis

Hung Pham, Quang-Cuong Pham

Abstract— **Given a geometric path, the Time-Optimal Path Tracking problem consists in finding the control strategy to traverse the path time-optimally while regulating tracking errors. A simple yet effective approach to this problem is to decompose the controller into two components: (i) a path controller, which modulates the parameterization of the desired path in an online manner, yielding a reference trajectory; and (ii) a tracking controller, which takes the reference trajectory and outputs joint torques for tracking. However, there is one major difficulty: the path controller might not find any feasible reference trajectory that can be tracked by the tracking controller because of torque bounds. In turn, this results in degraded tracking performances. Here, we propose a new path controller that is guaranteed to find feasible reference trajectories by accounting for possible future perturbations. The main technical tool underlying the proposed controller is Reachability Analysis, a new method for analyzing path parameterization problems. Simulations show that the proposed controller outperforms existing methods.**

I. Introduction

Time-optimal motion planning and control along a predefined path are fundamental and important problems in robotics, motivated by many industrial applications, ranging from machining, to cutting, to welding, to painting, etc.

The *planning* problem is to find the Time-Optimal Path Parameterization (TOPP) of a path under kinematic and dynamic bounds. The underlying assumptions are that the robot is perfectly modeled, no perturbations during execution and no initial tracking errors. This problem has been extensively studied since the 1980's [1], see [2], [3] for recent reviews.

The *control* problem, which looks for a control strategy to time-optimally track the path while *accounting for model inaccuracies, perturbations and initial tracking errors*, is comparatively less well understood. We refer to this problem as the Time-Optimal Path Tracking problem, or "path tracking problem" in short.

A. Approaches to Time-Optimal Path Tracking

The first approach to the path tracking problem was proposed by Dahl and colleagues in the 1990's [4], [5]. Suppose that we are given a geometric path $\mathbf{p}(s)_{s\in[0,1]}$. In Dahl's approach, termed Online Scaling (OS), the path tracking controller is composed of two sub-controllers: a *path controller* and a *tracking controller*, see Fig. 1. The path controller generates a *path parameterization* $s(t)$ ("scaling") by controlling the path acceleration $\ddot{s}(t)$, from which it returns "online" a reference trajectory $(\mathbf{q}_d, \dot{\mathbf{q}}_d)$ via the relations

$$\mathbf{q}_d(t) = \mathbf{p}(s(t)),\ \dot{\mathbf{q}}_d(t) = \mathbf{p}'(s(t))\dot{s}(t).$$

The tracking controller then takes the reference trajectory $(\mathbf{q}_d, \dot{\mathbf{q}}_d)$ and generates the joint torques $\boldsymbol{\tau}$ to drive the current state to the reference state. In OS, the tracking controller is usually as a computed-torque tracking controller with fixed Proportional-Derivative (PD) gains. Thus, the problem is reduced to designing a path controller that can regulate the path tracking errors while tracking a reference parameterization or minimizing execution time.

There have been a number of developments to OS. In [6], the author proposed to use an observer to estimate the online constraints on the parameterization. In [7] and [8], OS was extended to handle manipulators with elastic joints or are subject to high-order dynamics such as torque rate or jerk.

Yet these developments neglect a fundamental problem: there is *no guarantee* for the path controller to find feasible controls at execution time. In fact, this issue is recognized in most of the papers devoted to OS. For example, in the original paper [4], the authors proposed to use the nominal control if there is no feasible control for the path controller. In a more recent work [7], the authors asserted that: "*since [the path control] bounds are online evaluated [. . .], it is not possible to guarantee [. . . that] a feasible solution exists [. . .]*". Yet, employing arbitrary substitute controls when no feasible control exists will generate large path tracking errors. For time-optimal path tracking, this issue of *infeasibility* is far from rare since, by Pontryagin Maximum Principle, time-optimality is associated with saturating torque bounds at almost every time instant.

A simpler approach to the path tracking problem can be found in [9]. The authors proposed to consider more conservative torque bounds at the planning stage, "reserving"

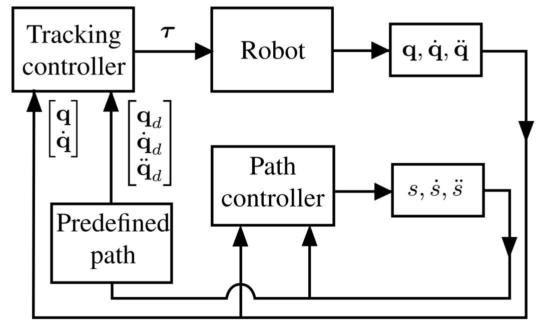

Fig. 1. Block diagram of an Online Scaling controller.

Hung Pham and Quang-Cuong Pham are with Air Traffic Management Research Institute (ATMRI) and Singapore Centre for 3D Printing (SC3DP), School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore (email: hungpham2511@gmail.com, cuong.pham@normalesup.org)

thereby some torques for tracking during execution. However, this approach is clearly sub-optimal.

Recently, some authors considered the full optimal control problem, whose state is $(\mathbf{q}, \dot{\mathbf{q}}, s, \dot{s})$ and control is $(\boldsymbol{\tau}, \ddot{s})$, and applied Nonlinear Model Predictive Control (NMPC) [10], [11]. While NMPC can account for hard constraints on state and control, it has some limitations. First, ensuring stability is still non-trivial [12]. For instance, in [10], to achieve stability, the path tracking NMPC controller requires *hand-designed* terminal sets. Second, the time-optimality objective is challenging since it is non-convex in the time domain [13].

B. Contribution and organization of the paper

To guarantee that the path controller will always find feasible controls requires a certain level of foresight: one needs to take into account *all possible perturbations along the path*. In this paper, we build on the recent formulation of TOPP by Reachability Analysis [14] to provide such foresight. Specifically, we compute sets of robust controllable states [1] that guarantee the existence of feasible controls for bounded tracking errors. From these sets, a class of path tracking controllers that have exponential stability and feasiblity guarantees is identified. The time-optimal controller is then found straightforwardly.

The rest of the paper is organized as follows. Section II provides the background on the path tracking problem and the path tracking controller. Section III presents the main contributions. Section IV reports experimental results, demonstrating the effectiveness of the proposed approach. Finally, Section V delivers concluding remarks and sketches directions for future research.

C. Notation

We adopt the following conventions. Vectors are denoted by bold letters: $\mathbf{x}$. The i-th component of a vector is denoted using subscript i: x_i. A vector quantity at stage j is denoted by bold letters with subscript j: $\mathbf{x}_j$, its i-th component is denoted by adding a second subscript i: x_{ji}. Define function $\phi(s, s^d; \mathbf{p}) := [\mathbf{p}(s)^\top, \mathbf{p}'(s)^\top s^d]^\top$, argument $\mathbf{p}$ will be neglected if clear from context. If $\mathbf{x}, \mathbf{y}$ are two vectors, $(\mathbf{x}, \mathbf{y})$ denote the concatenated vector $(\mathbf{x}^\top, \mathbf{y}^\top)^\top$. Values of differential quantities, such as $\dot{\mathbf{q}}$, have superscript d: $\mathbf{q}_0^d$.

II. Background: Path Tracking problem and controllers

A. Path Tracking problem

Path tracking is the problem of designing a controller to make the robot's joint positions follow a path parameterization of a predefined path. The path parameterization is not fixed but is generated by the controller in an online manner.

Specifically, we consider a n-dof manipulator with the dynamic equation

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{q}}^\top \mathbf{C}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{h}(\mathbf{q}) = \boldsymbol{\tau}, \tag{1}$$

[1]These are parameterization states, which are defined as squared path velocities. In Section III, precise definitions are given.

where $\mathbf{q} \in \mathbb{R}^n$ and $\boldsymbol{\tau} \in \mathbb{R}^n$ denote the vectors of joint positions and joint torques; $\mathbf{M}, \mathbf{C}, \mathbf{h}$ are appropriate functions. The joint torques are bounded:

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max}. \tag{2}$$

A geometric path is a twice-differentiable function $\mathbf{p}(s)_{s\in[0,1]} \in \mathbb{R}^n$. A *path parameterization* is a twice-differentiable non-decreasing function $s(t)_{t\in[0,T]} \in [0,1]$. Path parameterizations are also subject to terminal velocity constraints of the form $\dot{s}(T) \in \mathbb{I}_{\mathrm{end}}$, where $\mathbb{I}_{\mathrm{end}}$ is called the terminal set. To generate the parameterization, one directly controls the path acceleration. Let u denote the path parameterization control: $\ddot{s} = u$.

The state-space equation of the *coupled system* consisting of the manipulator and the path parameterization reads

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{q}\\ \dot{\mathbf{q}}\\ s\\ \dot{s}\end{bmatrix} = \begin{bmatrix}\dot{\mathbf{q}}\\ \mathbf{M}(\mathbf{q})^{-1}(\boldsymbol{\tau} - \dot{\mathbf{q}}^\top\mathbf{C}(\mathbf{q})\dot{\mathbf{q}} - \mathbf{h}(\mathbf{q}))\\ \dot{s}\\ u\end{bmatrix}. \tag{3}$$

Let $\mathbf{y}$ denote the state of the coupled system $[\mathbf{q}, \dot{\mathbf{q}}, s, \dot{s}]$, Eq. (3) can be written concisely as

$$\dot{\mathbf{y}} = f(\mathbf{y}) + g(\mathbf{y})\begin{bmatrix}\boldsymbol{\tau}\\ u\end{bmatrix}.$$

Consider a control law $[\boldsymbol{\tau}, u] = \alpha(\mathbf{y})$ that always satisfies the torque bounds, one obtains the autonomous dynamics

$$\dot{\mathbf{y}} = f(\mathbf{y}) + g(\mathbf{y})\alpha(\mathbf{y}) = \hat{f}(\mathbf{y}). \tag{4}$$

We say that $\mathbf{y}(t)_{t\in[0,T]} \in \mathbb{R}^{2n+2}$ is a *solution* of Eq. (4) if

$$\dot{\mathbf{y}}(t) = \hat{f}(\mathbf{y}(t)),\ y_{2n+1}(t) \in [0,1],\ \forall t \in [0,T].$$

It is a *feasible solution* if additionally,

$$\begin{aligned} &y_{2n+2}(t) \geq 0,\ \forall t \in [0,T],\\ &y_{2n+1}(T) = 1, y_{2n+2}(T) \in \mathbb{I}_{\mathrm{end}}. \end{aligned}$$

It is a solution *with initial value* $\mathbf{y}_0$ if $\mathbf{y}(0) = \mathbf{y}_0$.

The coupled system is *stable* at $(s_0, s_0^d) \in [0,1] \times [0,\infty]$ if for any $R > 0$, there exist $r > 0$ such that if $\|(\mathbf{q}_0, \mathbf{q}_0^d) - \phi(s_0, s_0^d; \mathbf{p})\|_2 \leq r$, the solution $\mathbf{y} =: (\mathbf{q}, \dot{\mathbf{q}}, s, \dot{s})$ with initial value $(\mathbf{q}_0, \mathbf{q}_0^d, s_0, s_0^d)$ exists and

$$\|(\mathbf{q}(t), \dot{\mathbf{q}}(t)) - \phi(s(t), \dot{s}(t))\|_2 < R,\ \forall t \in [0,T].$$

The coupled system is *exponentially stable* at $(s_0, s_0^d) \in [0,1]\times[0,\infty]$ if it is stable and there exist $r_e > 0$ such that if $\|(\mathbf{q}_0, \mathbf{q}_0^d) - \phi(s_0, s_0^d; \mathbf{p})\|_2 \leq r_e$, the solution $\mathbf{y} = (\mathbf{q}, \dot{\mathbf{q}}, s, \dot{s})$ with initial value $(\mathbf{q}_0, \mathbf{q}_0^d, s_0, s_0^d)$ exists and satisfies

$$\|(\mathbf{q}(t), \dot{\mathbf{q}}(t)) - \phi(s(t), \dot{s}(t))\|_2 < Ke^{-\lambda t}, 0 \leq t \leq T,$$

for some positive real numbers K, λ.

B. Path Tracking controllers

A path tracking controller consists of a path controller, which controls the path acceleration to generate a desired joint trajectory $\mathbf{q}_d(t) := \mathbf{p}(s(t))$, and a tracking controller that controls the joint torques to track the desired joint trajectory.

A common control objective is to track a predefined reference path parameterization. Here we consider the time-optimal objective, which is to traverse the path as fast as possible.

Similar to the paper [4] and subsequent developments [5], [7], we employ the computed-torque trajectory tracking scheme for the tracking controller. This scheme implements the following control law

$$\begin{aligned}\boldsymbol{\tau} =& \mathbf{M}(\mathbf{q})[\ddot{\mathbf{q}}_d + \mathbf{K}_p\mathbf{e} + \mathbf{K}_d\dot{\mathbf{e}}] \\ &+ \dot{\mathbf{q}}^\top\mathbf{C}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{h}(\mathbf{q}),\end{aligned} \tag{5}$$

where $\mathbf{e}$ denote the joint positions error vector, defined as $\mathbf{e} := \mathbf{q}_d(t) - \mathbf{q}(t)$, $\mathbf{K}_p$ and $\mathbf{K}_d$ are the PD gain matrices. The vector $(\mathbf{e}, \dot{\mathbf{e}})$ is called the tracking error. The first and second time derivatives of the desired joint trajectory, which are used in (5), are given by

$$\begin{aligned}\dot{\mathbf{q}}_d(t) &= \mathbf{p}'(s(t))\dot{s}(t),\\ \ddot{\mathbf{q}}_d(t) &= \mathbf{p}'(s(t))\ddot{s}(t) + \mathbf{p}''(s(t))\dot{s}(t)^2.\end{aligned}$$

Rearranging Eq. (5), one obtains a formula for joint torques:

$$\boldsymbol{\tau} = \hat{\mathbf{a}}(\mathbf{y})\ddot{s} + \hat{\mathbf{b}}(\mathbf{y})\dot{s}^2 + \hat{\mathbf{c}}(\mathbf{y}), \tag{6}$$

where

$$\begin{aligned}\hat{\mathbf{a}}(\mathbf{y}) &= \mathbf{M}(\mathbf{p}(s) + \mathbf{e})\mathbf{p}'(s),\\ \hat{\mathbf{b}}(\mathbf{y}) &= \mathbf{M}(\mathbf{p}(s) + \mathbf{e})\mathbf{p}''(s) + \mathbf{p}'(s)^\top\mathbf{C}(\mathbf{p}(s) + \mathbf{e})\mathbf{p}'(s),\\ \hat{\mathbf{c}}(\mathbf{y}) &= \mathbf{M}(\mathbf{p}(s) + \mathbf{e})[\mathbf{K}_p\mathbf{e} + \mathbf{K}_d\dot{\mathbf{e}}]\\ &\quad + 2\dot{\mathbf{e}}^\top\mathbf{C}(\mathbf{p}(s) + \mathbf{e})\mathbf{p}'(s)\dot{s} + \mathbf{h}(\mathbf{p}(s) + \mathbf{e}).\end{aligned}$$

We observe that if the tracking error is zero, the coefficients $\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}$ depend only on the path position s and not on the path velocity $\dot{s}$. Indeed in this case, the coefficients reduce to

$$\begin{aligned}\mathbf{a}(s) &= \mathbf{M}(\mathbf{p}(s))\mathbf{p}'(s),\\ \mathbf{b}(s) &= \mathbf{M}(\mathbf{p}(s))\mathbf{p}''(s) + \mathbf{p}'(s)^\top\mathbf{C}(\mathbf{p}(s))\mathbf{p}'(s),\\ \mathbf{c}(s) &= \mathbf{h}(\mathbf{p}(s)).\end{aligned}$$

We call $\mathbf{a}, \mathbf{b}, \mathbf{c}$ the nominal coefficients. Additionally, by inspection, we see that the coefficients $\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}$ are continuous with respect to the tracking error $\mathbf{e}, \dot{\mathbf{e}}$.

It follows from the definition of continuous functions that for any pair $(s, s^d) \in [0, 1] \times [0, \infty]$ and any positive number R there exists $r > 0$ such that for all $i \in [1 \dots n]$

$$\left\| \begin{bmatrix}\mathbf{e}\\ \dot{\mathbf{e}}\end{bmatrix} \right\|_2 < r \implies \left\| \begin{bmatrix}\hat{a}_i(\mathbf{q}, \dot{\mathbf{q}}, s, s^d) - a_i(s)\\ \hat{b}_i(\mathbf{q}, \dot{\mathbf{q}}, s, s^d) - b_i(s)\\ \hat{c}_i(\mathbf{q}, \dot{\mathbf{q}}, s, s^d) - c_i(s)\end{bmatrix} \right\|_2 < R. \tag{7}$$

This result implies that for tracking errors with sufficiently small magnitude, the coefficients $\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}$ vary around the nominal coefficients $\mathbf{a}, \mathbf{b}, \mathbf{c}$. Furthermore, since the path velocity can always be assumed to be bounded, we can strengthen this result: there exists $\bar{r} > 0$ such that Eq. (7) holds for any pair (s, s^d).

C. Difficulties with designing path controllers

The fundamental difficulty with designing path controllers is that the coefficients $\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}$ are only available online. Hence, it is non-trivial to avoid situations in which there is no path acceleration that satisfies Eq. (6). A consequence of this infeasibility is that exponential convergence of the actual joint trajectory to the desired joint trajectory is not guaranteed because of saturating torque bounds. This difficulty also renders reaching the terminal set challenging.

III. Solving the Time-Optimal Path Tracking Problem

A. Exponential stability with robust feasible control laws

In the last section, it was shown that if tracking errors have small magnitude, the coefficients of Eq. (6) vary around the nominal coefficients. Motivating by this observation, we introduce the notion of *robust feasible* control laws. Note that in this section, *control laws* refer to control laws for selecting path parameterization controls u, not the coupled control $[\boldsymbol{\tau}, u]$.

Specifically, a *control law* is a function that computes the path acceleration from the current path position, the current path velocity and the coefficients:

$$u(t) = \pi(s(t), \dot{s}(t), \hat{\mathbf{a}}(t), \hat{\mathbf{b}}(t), \hat{\mathbf{c}}(t)).$$

The control law π is *robust feasible* at the pair (s_0, s_0^d) if there exists $R > 0$ such that for any set of coefficients

$$\begin{aligned}&\hat{\mathbf{a}}(t) := \mathbf{a}(s(t)) + \boldsymbol{\Delta}_a(t), \hat{\mathbf{b}}(t) := \mathbf{b}(s(t)) + \boldsymbol{\Delta}_b(t),\\ &\hat{\mathbf{c}}(t) := \mathbf{b}(s(t)) + \boldsymbol{\Delta}_c(t),\end{aligned}$$

where the perturbations $\boldsymbol{\Delta}_a, \boldsymbol{\Delta}_b, \boldsymbol{\Delta}_c$ are arbitrary continuous functions satisfying

$$\|(\Delta_{a,i}(t), \Delta_{b,i}(t), \Delta_{c,i}(t))\|_2 < R, \forall t, i \in [1, \dots, n], \tag{8}$$

the generated parameterization $s(t)$ is feasible, that is

$$s(0) = s_0, \dot{s}(0) = s_0^d, \tag{9}$$

$$\exists T, s(T) = 1, \dot{s}(t) \in \mathbb{I}_{\text{end}}, \tag{10}$$

$$\forall t \in [0, T], \dot{s}(t) \geq 0 \tag{11}$$

$$\forall t \in [0, T], \text{Eq. (8) holds.} \tag{12}$$

The following result shows that a robust feasible control law ensures exponentially stable path tracking.

Proposition 1. *Consider a path tracking controller with (path acceleration) control law π. If π is robust feasible at (s_0, s_0^d) then the coupled system is exponentially stable at (s_0, s_0^d).*

Proof. Let R denote the bound on the magnitude of the perturbations such that π is robust feasible. Select $\bar{r} > 0$ such that Eq. (7) holds for R being the scalar bound in Eq. (19) and for all (s, s^d). Select $(\mathbf{q}_0, \mathbf{q}_0^d)$ such that the norm of the initial tracking error is less than $\bar{r}$.

Suppose we remove the torque bounds, the computed-torque tracking controller is exponentially stable. Thus, the tracking error converges exponentially to zero and its norm remains smaller than $\bar{r}$. Again using Eq. (7), it follows that the norm of the perturbations is always smaller than R.

Since π is robust feasible at (s_0, s_0^d) for R being the upper bound on the magnitude of the perturbations, the resulting parameterization $s(t)$ is feasible. It follows that the torque bounds are always satisfied. Therefore, the coupled path tracking system is exponentially stable at (s_0, s_0^d) according to the definition given in Section II-A. □

Notice that the definition of robust feasible control laws does not require a specific R. In fact, as seen in the proof of Proposition 1, continuity of the coefficients guarantees the existence of $\bar{r}$ such that Eq. (8) holds for any value of R. It can be observed that $\bar{r}$ is the radius of a ball lying inside the region of attraction of the path tracking controller.

B. Characterizing robust feasible control laws

We now provide a characterization of robust feasible control laws. This development follows and extends the analysis of the Time-Optimal Path Parameterization problem in [14].

Discretize the interval $[0, 1]$ into $N + 1$ stages

$$0 =: s_0, s_1, \ldots, s_N := 1.$$

Define the *state* x_i and the *control* u_i as the squared velocity at s_i and the constant acceleration over $[s_i, s_{i+1}]$. One obtains the transition function

$$x_{i+1} = f_i(x_i, u_i) := x_i + 2\Delta_i u_i, \tag{13}$$

where $\Delta_i := s_{i+1} - s_i$. We say that u_i "steers" x_i to x_{i+1}. See [14] for a derivation of Eq. (13).

At each stage, there are n pairs of torque bounds. The j-th pair of torque bounds at stage i is

$$\tau_{\min,j} \le \tau_{ij} = \hat{a}_{ij} u_i + \hat{b}_{ij} x_i + \hat{c}_{ij} \le \tau_{\max,j}. \tag{14}$$

The coefficients $\hat{a}_{ij}, \hat{b}_{ij}, \hat{c}_{ij}$ are assumed to vary around known nominal coefficients a_{ij}, b_{ij}, c_{ij}:

$$\hat{a}_{ij} = a_{ij} + \Delta_{a,ij},\ \hat{b}_{ij} = b_{ij} + \Delta_{b,ij},\ \hat{c}_{ij} = c_{ij} + \Delta_{c,ij}, \tag{15}$$

$$\|(\Delta_{a,ij}, \Delta_{b,ij}, \Delta_{c,ij})\|_2 \le R. \tag{16}$$

The terms $(\Delta_{a,ij}, \Delta_{b,ij}, \Delta_{c,ij})$ are also called the perturbations. R is a parameter that be tuned to account for the magnitude of initial tracking errors. See the discussion at the end of Section III-A for more details. A control is feasible if it satisfies the constraints and is robust feasible if it satisfies all realizations of the constraints. Finally, the terminal velocity constraint is transformed to

$$x_N \in X_f := \{x : \sqrt{x} \in \mathbb{I}_{\text{end}}\}.$$

We say that a state is *robust controllable* at stage i if there exists a sequence of robust feasible controls that steers it to $\mathcal{X}_f$. The set of robust controllable states at stage i is called the *i-stage robust controllable set* $\mathcal{K}_i$.

In this discrete reformulation, the control law becomes a function π that maps the stage index, the state and the constraint coefficients $(i, x_i, \hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i, \hat{\mathbf{c}}_i)$ to a control. Similarly, a control law is *robust feasible* at (i, x_i) if it steers x_i to the terminal set from stage i for any realization of the constraints with feasible controls.

We now give a characterization of robust feasible control laws.

Proposition 2. *For any state in $\mathcal{K}_i$ and any realization of the constraints, there exists at least one feasible control that steers that state to $\mathcal{K}_{i+1}$. If at any stage i, a control law steers states in $\mathcal{K}_i$ to $\mathcal{K}_{i+1}$, it is robust feasible at all robust controllable states at all stages.*

This characterization of robust feasible control laws is only useful if one can compute the robust controllable sets. To do so, we first introduce the notion of the robust one-step set. Given a target set $\mathbb{I} \subseteq \mathbb{R}$, the i-stage *robust one-step set* $\mathcal{Q}_i(\mathbb{I})$ is the set of states such that at each state, there is a robust feasible control that steers it to $\mathbb{I}$.

Proposition 3. *The i-stage robust controllable sets, for $i \in [0, \ldots, N]$, can be computed recursively by*

$$\mathcal{K}_N = X_f, \quad \mathcal{K}_i = \mathcal{Q}_i(\mathcal{K}_{i+1}). \tag{17}$$

A proof of this statement is omitted due to space constraints. Interested readers can refer to [14] for the proof of a similar result. We can now give a proof of Proposition 2 below.

Proof of Proposition 2. Let x_i be a state in $\mathcal{K}_i$. Since x_i is robust controllable, there exists a sequence of controls $(u_i, \ldots, u_{N-1})$ that are robust feasible and the resulting sequence of states $(x_{i+1}, \ldots, x_N)$ satisfies $x_N \in \mathcal{X}_N$. Observe that this implies x_{i+1} is robust controllable, and hence, u_i is a robust feasible control that steer x_i to $\mathcal{K}_{i+1}$.

Consider a control law that steers states in $\mathcal{K}_i$ for any stage i to $\mathcal{K}_{i+1}$. It is clear that this control law steers any robust controllable states to $\mathcal{K}_N$. Since $\mathcal{K}_N = \mathcal{X}_f$, see (3), the control law is robust feasible. □

A class of convex sets that can be handled quite efficiently is the class of *Conic-Quadratic* representable (CQr) sets [15]. A set of vectors $\boldsymbol{\epsilon}$ is CQr if it is defined by finitely many conic-quadratic constraints

$$\left\| \mathbf{D}_i \begin{bmatrix} \boldsymbol{\epsilon} \\ \boldsymbol{\nu} \end{bmatrix} - \mathbf{d}_i \right\|_2 \le \mathbf{p}_i^\top \begin{bmatrix} \boldsymbol{\epsilon} \\ \boldsymbol{\nu} \end{bmatrix} - q_i,\ i \in [1, \ldots, k].$$

Proposition 4. *If $\mathbb{I}$ is an interval, the set of state and robust feasible control pairs (x, u) that satisfies $x + 2\Delta_i u \in \mathbb{I}$ is a CQr. Furthermore, $\mathcal{Q}_i(\mathbb{I})$ is an interval.*

Indeed, from Eq. (14) and Eq. (15), the j-th joint torque is given by

$$\tau_{ij} = \begin{bmatrix} a_{ij} & b_{ij} & c_{ij} \end{bmatrix} \begin{bmatrix} u_i \\ x_i \\ 1 \end{bmatrix} + \begin{bmatrix} \Delta_{a,ij} & \Delta_{b,ij} & \Delta_{c,ij} \end{bmatrix} \begin{bmatrix} u_i \\ x_i \\ 1 \end{bmatrix}.$$

Since the norm of the perturbation is bounded, see Eq. (16), one obtains the inequality

$$\tau_{ij} \le \begin{bmatrix} a_{ij} & b_{ij} & c_{ij} \end{bmatrix} \begin{bmatrix} u_i \\ x_i \\ 1 \end{bmatrix} + R \left\| \begin{bmatrix} u_i \\ x_i \\ 1 \end{bmatrix} \right\|_2 . \quad (18)$$

It is clear that if and only if the right-hand side is not greater than $\tau_{\max,j}$, the pair (u_i, x_i) satisfies all realizations of this constraint. One obtains the conic-quadratic constraint

$$R \left\| \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_i \\ x_i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\|_2 \le - \begin{bmatrix} a_{ij} & b_{ij} \end{bmatrix} \begin{bmatrix} u_i \\ x_i \end{bmatrix} - c_i + \tau_{\max,j} . \quad (19)$$

Note that the lower bound can be handled in a similar way. Instead of finding the upper bound of τ_{ij}, one derives the lower bound

$$\tau_{ij} \ge \begin{bmatrix} a_{ij} & b_{ij} & c_{ij} \end{bmatrix} \begin{bmatrix} u_i \\ x_i \\ 1 \end{bmatrix} - R \left\| \begin{bmatrix} u_i \\ x_i \\ 1 \end{bmatrix} \right\|_2 . \quad (20)$$

By requiring the right-hand side to be greater than or equal to $\tau_{\min,j}$, one obtains another conic-quadratic constraint.

Finally, if $\mathbb{I}$ is an interval, the constraint $x + 2\Delta_i u \in \mathbb{I}$ is equivalent to two linear inequalities, which are clearly CQr. $\mathcal{Q}_i(\mathbb{I})$ being an interval is a simple corollary.

From Proposition 4, one can formulate a pair of conic-quadratic optimization programs to compute the robust one-step set for any given target set. One program maximizes x while one program minimizes. Computing the robust controllable sets is then possible with Proposition 3.

C. A control law for time-optimal path tracking

Proposition 2 can be used to identify a class of control laws that are robust feasible, which, by Proposition 1 are exponentially stable. What is then the control law that realizes the shortest traversal time in this class? We give the following conjecture.

Conjecture 1. *In the class of control laws that for all i steers states in $\mathcal{K}_i$ to $\mathcal{K}_{i+1}$, the control law that always chooses* the greatest feasible controls *is time-optimal.*

We current do not have a proof of this conjecture. Regardless, in our experiments, the conjecture is verified by comparing the traversal time with the duration of the time-optimal path parameterization.

IV. Experimental results

We simulated a 6-axis robotic arm and controlled it to track a geometric path $\mathbf{p}(s)_{s\in[0,1]}$ with zero terminal velocity constraint. The torques bounds are

$$\boldsymbol{\tau}_{\max} = -\boldsymbol{\tau}_{\min} = [120., 280., 280., 120., 80., 80.](\mathrm{Nm}).$$

Fig. 2 visualizes the swinging motion. Initially the robot was at rest and had an initial joint positions error with magnitude $0.1\,\mathrm{rad}$. Forward dynamic computations were performed using OpenRAVE [16] and the `dopri5` solver. We sampled joint torques at sample time $1\,\mathrm{ms}$.

We implemented the time-optimal path tracking controller conjectured in Section III-C, with the bounds on the norm of the perturbations R set uniformly to 0.5. The number of discretization step N was set to 100.

Computing the robust controllable sets excluding computations of the coefficients took $120\,\mathrm{ms}$. We solved the conic-quadratic programs using the Python interface of `ECOS` [17]. Note that computing the coefficients involves evaluating the inverse dynamics twice per stage [2], which had a total running time of $40\,\mathrm{ms}$. Online computations of the controls $(\boldsymbol{\tau}, u)$ took $0.50\,\mathrm{ms}$ per time step. All computations were done on a single core of a laptop at $3.800\,\mathrm{GHz}$.

We compared our controller, called the Time-Optimal Path Tracking controller (TOPT), with the Online Scaling controller (OS) in [5] and the Computed-Torque Trajectory Tracking controller (TT) in [18]. The OS controller tracked the time-optimal path parameterization of the given geometric path, while the TT controller tracked the time-optimal *trajectory*.

TABLE I
TRACKING DURATION AND MAX POSITION ERRORS

	TOPT	OS	TT
Max pos. err. (rad)	0.10	0.491	0.493
Tracking dur. (sec)	1.021	1.017	1.017

We observe that the TT controller was incapable of handling the initial position error: Fig. 3A shows that position errors increased quickly reaching a maximum norm of $0.49\,\mathrm{rad}$ before stabilizing.

The OS controller was only able to regulate position errors during the initial segment. At $s \approx 0.18$, position errors increased sharply. See Fig. 3A. We note that at this instance, the OS controller was not able to find any *feasible* path acceleration. This event can be observed in Fig. 3B as a sharp spike on the generated parameterization.

The TOPT controller did not show any of the above problems. The joint positions converged quickly to zero. The total tracking duration of the TOPT controller was slightly higher than the *optimal duration*: about 1% longer. See Table I for the durations.

Finally, we observed that the parameterization generated by the TOPT controller differed from the parameterization generated by the OS controller mostly *during decelerating path segments*. See for instance the path position interval $s \in [0.05, 0.15]$ in Fig. 3. Specifically, it can be seen that the TOPT controller "slowed down" in order to stay within the robust controllable sets. This helped the TOPT controller avoids the infeasibility at $s = 0.18$.

V. Conclusion

In this paper, we considered the Time-Optimal Path Tracking problem: given a geometric path, find the control strategy to traverse the path time-optimally while regulating tracking errors. We have introduced the Time-Optimal Path Tracking controller and shown that the controller outperforms existing

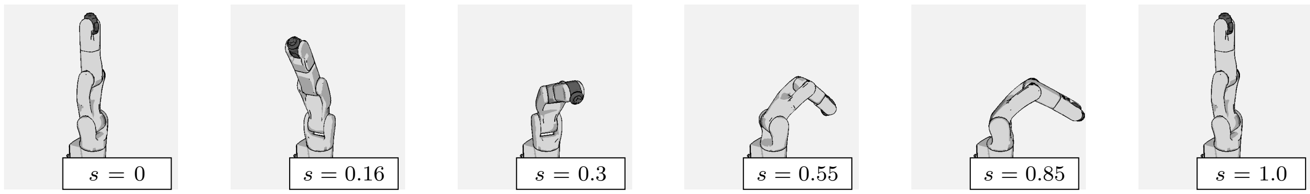

Fig. 2. The swinging motion used in the experiment.

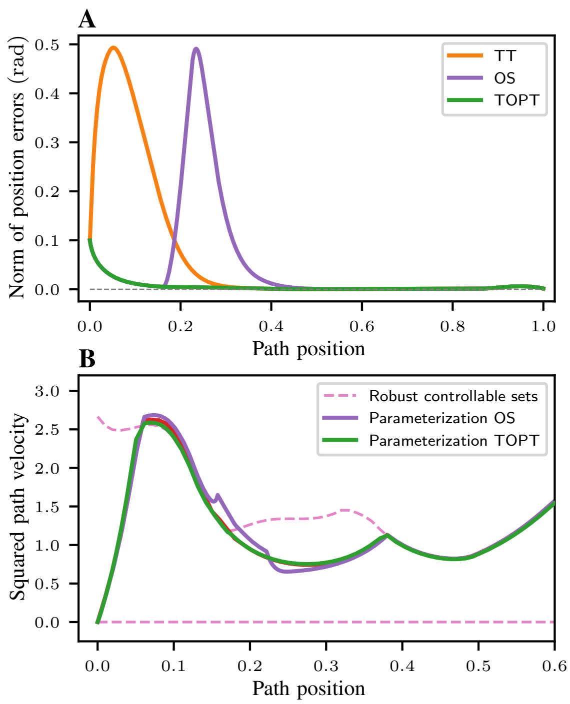

Fig. 3. **A**: Norms of joint position errors of three controllers: TOPT, OS and TT. **B**: The path position-squared path velocity space showing parameterizations (solid lines) generated online by the TOPT and OS controllers, the robust controllable sets (dashed lines).

methods. The key innovation is the use of robust controllable sets, which intuitively define the sets of "safe" path parameterizations that can be tracked while accounting for possible variations of the coefficients. The technique used in this paper is Reachability Analysis, a new method for analyzing path parameterization problems [14].

Several matters have been left for future investigations. Important questions include how to evaluate and optimize the region of attraction of path tracking controllers. Another direction is extending the approach to handle industrial manipulators with position or velocity interfaces and to account for higher-ordered constraints such as joint jerk bounds.

Acknowledgment

This work was partially supported by grant ATMRI:2014-R6-PHAM (awarded by NTU and the Civil Aviation Authority of Singapore) and by the Medium-Sized Centre funding scheme (awarded by the National Research Foundation, Prime Minister's Office, Singapore).

REFERENCES

[1] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The international journal of robotics research*, vol. 4, no. 3, pp. 3–17, 1985.

[2] Q.-C. Pham, "A General, Fast, and Robust Implementation of the Time-Optimal Path Parameterization Algorithm," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1533–1540, dec 2014.

[3] Q.-C. Pham and O. Stasse, "Time-Optimal Path Parameterization for Redundantly Actuated Robots: A Numerical Integration Approach," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 3257–3263, dec 2015.

[4] O. Dahl and L. Nielsen, "Torque-limited path following by online trajectory time scaling," *Robotics and Automation, IEEE Transactions on*, vol. 6, no. 5, pp. 554–561, 1990.

[5] O. Dahl, "Path-constrained robot control with limited torques-experimental evaluation," *IEEE transactions on robotics and automation*, vol. 10, no. 5, pp. 658–669, 1994.

[6] H. Arai, K. Tanie, and S. Tachi, "Path tracking control of a manipulator considering torque saturation," *IEEE Transactions on Industrial Electronics*, vol. 41, no. 1, pp. 25–30, 1994.

[7] C. G. L. Bianco and O. Gerelli, "Online trajectory scaling for manipulators subject to high-order kinematic and dynamic constraints," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1144–1152, 2011.

[8] O. Gerelli and C. G. lo Bianco, "Real-time Path Tracking Control of Robotic Manipulators with Bounded Torques and Torque-derivatives," *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2008*, 2008.

[9] J. Kieffer, A. J. Cahill, and M. R. James, "Robust and accurate time-optimal path-tracking control for robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 6, pp. 880–890, 1997.

[10] T. Faulwasser and R. Findeisen, "Nonlinear model predictive control for constrained output path following," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 1026–1039, 2016.

[11] T. Faulwasser, T. Weber, P. Zometa, and R. Findeisen, "Implementation of nonlinear model predictive path-following control for an industrial robot," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1505–1511, 2017.

[12] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[13] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Practical time-optimal trajectory planning for robots: a convex optimization approach," *IEEE Transactions on Automatic Control*, 2008.

[14] H. Pham and Q.-C. Pham, "A New Approach to Time-Optimal Path Parameterization based on Reachability Analysis," *IEEE Transactions on Robotics*, 2018. [Online]. Available: https://arxiv.org/abs/1707.07239http://arxiv.org/abs/1707.07239

[15] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM, 2001.

[16] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, August 2010.

[17] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *European Control Conference (ECC)*, 2013, pp. 3071–3076.

[18] M. W. Spong and M. Vidyasagar, *Robot dynamics and control*. John Wiley & Sons, 2008.