Practical Time-Optimal Trajectory Planning for Robots: a Convex Optimization Approach

Diederik Verscheure*, Bram Demeulenaere†, Jan Swevers‡, Joris De Schutter§, and Moritz Diehl¶

***Abstract*—This paper focuses on time-optimal path-constrained trajectory planning, a subproblem in time-optimal motion planning of robot systems. Through a nonlinear change of variables, the time-optimal trajectory planning is transformed here into a convex optimal control problem with a single state. Various convexity-preserving extensions are introduced, resulting in a versatile approach for optimal trajectory planning. A direct transcription method is presented that reduces finding the globally optimal trajectory to solving a second-order cone program using robust numerical algorithms that are freely available. Validation against known examples and application to a more complex example illustrate the versatility and practicality of the new method.**

***Index Terms*—time-optimal trajectory planning, time-optimal control, convex optimal control, path tracking, trajectory planning, convex optimization, second-order cone program, direct transcription.**

I. INTRODUCTION

TIME-OPTIMAL motion planning is of significant importance for maximizing the productivity of robot systems. Solving the motion planning problem in its entirety, however, is in general a highly complex and difficult task [1]–[6]. Therefore, instead of solving the entire motion planning problem directly in the system's state space, which is called the direct approach [5], the decoupled approach [1], [5], [7]–[9] is often preferred for its lower computational requirements. The decoupled approach solves the motion planning problem in two stages. In the first *path planning* stage, a high-level planner determines a geometric path thereby accounting for task specifications, obstacle avoidance and other high-level – usually geometric – aspects [1], [8], [9], but ignoring lower-level – often dynamic – aspects such as the dynamics of the robotic manipulator. In the subsequent *path tracking* or *trajectory planning* stage, a time-optimal trajectory along the geometric path is determined, whereby the manipulator dynamics and actuator constraints are taken into account [7], [10]–[23].

The trajectory planning stage constitutes the focus of this paper. Time optimality along a predefined path implies realizing as high as possible a velocity along this path, without violating actuator constraints. To this end, the optimal trajectory should exploit the actuators' maximum acceleration and deceleration ability [10], [13], such that for every point along the path, at least one actuator saturates [15].

Methods for time-optimal robot trajectory planning subject to actuator constraints have been proposed in [7], [10]–[24]. While these optimal control methods can roughly be divided into three categories, most exploit that motion along a predefined path can be described by a single path coordinate s and its time derivative $\dot{s}$ [7], [10], [13]. Hence, the multi-dimensional state space of a robotic manipulator can be reduced to a *two-dimensional state space*. The $(s, \dot{s})$ curve, sometimes referred to as the *switching curve* [10], [14], unambiguously determines the solution of the time-optimal trajectory planning problem.

The first category of methods are indirect methods, which have have been proposed in [7], [10], [13] and subsequently refined in [14], [16], [19], [21], [22]. For the sake of brevity, only [14] is discussed here, which reports to be at least an order of magnitude faster than [7], [10], [13]. This method conducts one-dimensional numerical searches over s to exhaustively determine all characteristic switching points, that is, points where changes between the active actuator constraints can occur. Based on these points, the switching curve, and hence the solution to the overall planning problem, is found numerically through a procedure that is based on forward and backward integrations.

The second category consists of dynamic programming methods [11], [13], [24], while the third category consists of direct transcription methods [18], [23]. In contrast with most of the indirect methods, except [19] which considers time-energy optimality, the methods in the second and third category are able to take into account more general constraints and objective functions, such that time-optimality can be traded-off against other criteria such as energy, leading to less aggressive use of the actuators.

In this paper, the basic time-optimal trajectory planning problem, discussed in Sec. II, is transformed into a *convex* optimal control problem with a single state through a nonlinear change of variables introduced in Sec. III. Various convexity-preserving extensions, resulting in a versatile approach for optimal trajectory planning that goes beyond mere time-optimality, are presented in Sec. IV. Section V shows that direct transcription [6], [25]–[27] by simultaneous

D. Verscheure, B. Demeulenaere, J. Swevers and J. De Schutter are with the Division PMA, Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium.

M. Diehl is with the Division SCD, Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Belgium, e-mail: moritz.diehl@esat.kuleuven.be.

All authors are part of the Optimization in Engineering Center (OPTEC), Katholieke Universiteit Leuven.

Bram Demeulenaere is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen). The authors gratefully acknowledge the financial support by K.U.Leuven's Concerted Research Action GOA/05/10, K.U.Leuven's CoE EF/05/006 Optimization in Engineering Center (OPTEC), the Prof. R. Snoeys Foundation, and the Belgian Program on Interuniversity Poles of Attraction IAP VI/4 DYSCO (Dynamic Systems, Control and Optimization) initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture.

Manuscript received November 1, 2007; revised November 1, 2007.

discretization of the states and the controls, results in a reliable and very efficient method to numerically solve the optimal control problem based on second-order cone programming. Section VI subsequently validates this numerical method against the examples introduced in [13]. Section VII illustrates the practicality and versatility of the generalized problem formulation through a more advanced example of a six-DOF KUKA 361 industrial robot carrying out a writing task. Section VIII contrasts the proposed solution method with the existing indirect methods [7], [10], [13], [14], [16], [19], [21], [22], dynamic programming methods [11], [13], [24] and direct transcription methods [18], [23] and identifies aspects of future work.

II. Original problem formulation

The equations of motion of an n-DOF robotic manipulator with joint angles $\mathbf{q} \in \mathcal{R}^n$, can be written as a function of the applied joint torques $\boldsymbol{\tau} \in \mathcal{R}^n$ as [28]

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_s(\mathbf{q})\mathrm{sgn}(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}), \quad (1)$$

where $\mathbf{M}(\mathbf{q}) \in \mathcal{R}^{n\times n}$ is a positive definite mass matrix and $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathcal{R}^{n\times n}$ is a matrix accounting for Coriolis and centrifugal effects, which is *linear* in the joint velocities, $\mathbf{F}_s(\mathbf{q}) \in \mathcal{R}^{n\times n}$ is a matrix of Coulomb friction torques, which can be joint angle dependent, while $\mathbf{G}(\mathbf{q}) \in \mathcal{R}^n$ denotes the vector accounting for gravity and other joint angle dependent torques. In this paper, similarly as in [13], viscous friction is not considered.

Consider a path $\mathbf{q}(s)$, given in joint space coordinates[1], as a function of a scalar path coordinate s. The path coordinate determines the spatial geometry of the path, whereas the trajectory's time dependency follows from the relation $s(t)$ between the path coordinate s and time t. Without loss of generality, it is assumed that the trajectory starts at $t = 0$, ends at $t = T$ and that $s(0) = 0 \le s(t) \le 1 = s(T)$. In addition, since this paper considers *time-optimal* trajectory planning or related problems, it is assumed that $\dot{s}(t) \ge 0$ everywhere and $\dot{s}(t) > 0$ *almost* everywhere for $t \in [0, T]$.

For notational convenience, the time dependency of the path coordinate s and its derivatives is omitted wherever possible. For the given path, the joint velocities and accelerations can be rewritten using the chain rule as

$$\dot{\mathbf{q}}(s) = \mathbf{q}'(s)\dot{s}, \quad (2)$$

$$\ddot{\mathbf{q}}(s) = \mathbf{q}'(s)\ddot{s} + \mathbf{q}''(s)\dot{s}^2, \quad (3)$$

where $\dot{s} = \frac{ds}{dt}$, $\ddot{s} = \frac{d^2s}{dt^2}$, $\mathbf{q}'(s) = \frac{\partial \mathbf{q}(s)}{\partial s}$ and $\mathbf{q}''(s) = \frac{\partial^2 \mathbf{q}(s)}{\partial s^2}$. Substituting $\dot{\mathbf{q}}(s)$ and $\ddot{\mathbf{q}}(s)$ based on (2)-(3) results in the following expression for the equations of motion [10]

$$\boldsymbol{\tau}(s) = \mathbf{m}(s)\ddot{s} + \mathbf{c}(s)\dot{s}^2 + \mathbf{g}(s), \quad (4)$$

where

$$\mathbf{m}(s) = \mathbf{M}(\mathbf{q}(s))\mathbf{q}'(s), \quad (5)$$

$$\mathbf{c}(s) = \mathbf{M}(\mathbf{q}(s))\mathbf{q}''(s) + \mathbf{C}(\mathbf{q}(s), \mathbf{q}'(s))\mathbf{q}'(s), \quad (6)$$

$$\mathbf{g}(s) = \mathbf{F}_s(\mathbf{q}(s))\mathrm{sgn}(\mathbf{q}'(s)) + \mathbf{G}(\mathbf{q}(s)), \quad (7)$$

and where $\mathrm{sgn}(\dot{\mathbf{q}}(s))$ is replaced by $\mathrm{sgn}(\mathbf{q}'(s))$ using equation (2) and the assumption that $\dot{s} > 0$ almost everywhere.

Similarly as in [7], [10], [13], the time-optimal trajectory planning problem for the robotic manipulator subject to lower and upper bounds on the torques, can be expressed as

$$\min_{T, s(\cdot), \boldsymbol{\tau}(\cdot)} T, \quad (8)$$

$$\begin{aligned} \text{subject to } \boldsymbol{\tau}(t) = \ & \mathbf{m}(s(t))\ddot{s}(t) \\ & + \mathbf{c}(s(t))\dot{s}(t)^2 + \mathbf{g}(s(t)), && (9) \\ & s(0) = 0, && (10) \\ & s(T) = 1, && (11) \\ & \dot{s}(0) = \dot{s}_0, && (12) \\ & \dot{s}(T) = \dot{s}_T, && (13) \\ & \dot{s}(t) \ge 0, && (14) \\ & \underline{\boldsymbol{\tau}}(s(t)) \le \boldsymbol{\tau}(t) \le \overline{\boldsymbol{\tau}}(s(t)), && (15) \\ & \text{for } t \in [0, T], \end{aligned}$$

where the torque lower bounds $\underline{\boldsymbol{\tau}}$ and upper bounds $\overline{\boldsymbol{\tau}}$ may depend on s. In most cases, $\dot{s}_0$ and $\dot{s}_T$ can be taken equal to 0.

III. Reformulation as a convex optimal control problem

From equations (8)-(15), it is not obvious to decide whether any local solution to the problem is also *globally time-optimal*. In [19], the time-energy optimal control problem, which features (8)-(15) as a special case, is reformulated as an optimal control problem with *linear* system dynamics, differential state $(s, \dot{s})^T$ and control input $\ddot{s}$, subject to *nonlinear* state dependent control constraints. Subsequently, the Hamiltonian is shown to be convex with respect to the control input, which allows to conclude that any local optimum of the problem is also globally optimal. This paper provides, thanks to the use of a nonlinear change of variables, an appreciably different reformulation with a number of attractive properties.

First, by changing the integration variable from t to s, the objective function (8) is rewritten as

$$T = \int_0^T 1 dt = \int_{s(0)}^{s(T)} \frac{1}{\dot{s}} ds = \int_0^1 \frac{1}{\dot{s}} ds. \quad (16)$$

Second,

$$a(s) = \ddot{s}, \quad (17)$$

$$b(s) = \dot{s}^2 \quad (18)$$

are introduced as optimization variables and supplemented with an additional constraint

$$b'(s) = 2a(s), \quad (19)$$

which follows from the observation that

$$\dot{b}(s) = b'(s)\dot{s}, \quad (20)$$

as well as

$$\dot{b}(s) = \frac{d(\dot{s}^2)}{dt} = 2\ddot{s}\dot{s} = 2a(s)\dot{s}. \quad (21)$$

[1]For a path given in operational space coordinates, inverse kinematics techniques can be used to obtain the corresponding path in joint space coordinates [14], [28].

While the nonlinear transformation (17)-(18) is already recognized in [13], where it is used for quadrature purposes and for a geometric characterization of the admissable area of motion, this paper instead uses the transformed variables *directly as the optimization variables*, such that problem (8)-(15) can be reformulated as a convex problem

$$\min_{a(\cdot),b(\cdot),\boldsymbol{\tau}(\cdot)} \int_0^1 \frac{1}{\sqrt{b(s)}} ds, \tag{22}$$

$$\text{subject to } \boldsymbol{\tau}(s) = \mathbf{m}(s)a(s) + \mathbf{c}(s)b(s) + \mathbf{g}(s), \tag{23}$$

$$b(0) = \dot{s}_0^2, \tag{24}$$

$$b(1) = \dot{s}_T^2, \tag{25}$$

$$b'(s) = 2a(s), \tag{26}$$

$$b(s) \geq 0, \tag{27}$$

$$\underline{\boldsymbol{\tau}}(s) \leq \boldsymbol{\tau}(s) \leq \overline{\boldsymbol{\tau}}(s), \tag{28}$$

$$\text{for } s \in [0,1].$$

Problem (22)-(28) is convex since all constraints (23)-(28) are *linear*, while the objective function (22) is convex.

Problem (22)-(28) can be regarded as an optimal control problem in differential algebraic form (DAE), with pseudo-time s, control input $a(s)$, differential state $b(s)$, algebraic states $\boldsymbol{\tau}(s)$, *linear* system dynamics (26) and subject to *linear* state dependent constraints (23), (27)-(28), as well as initial and terminal constraints (24) and (25) respectively.

In contrast with the reformulation in [19], the reformulated problem (22)-(28) has *only one* differential state, while the algebraic states can easily be eliminated using equation (23). Moreover, time does *not appear explicitly* in the formulation anymore. However, the true merit of the reformulation (22)-(28), is that, unlike the reformulation in [19], first, it is clear without proof that problem (22)-(28) is convex, and second, it becomes very easy to devise additional objective functions and inequality constraints that can be incorporated, such that the resulting optimal control problem is *still convex*, as illustrated in Secs. IV-A and IV-B. In addition, due to the particular structure of the reformulated problem, it is shown in Sec. V how a numerical solution can be obtained very efficiently and reliably using a direct transcription method [6], [25]–[27].

IV. Generalized convex optimal control problem

In Secs. IV-A and IV-B, a number of practical objective functions and constraints are given, which can be incorporated to yield a more general, yet still convex optimal control problem as summarized in Sec. IV-C.

A. Objective functions

In addition to time, other objectives can also be incorporated to design a desirable relation between the path coordinate s and time t, such that time-optimality is traded-off against other criteria.

1) Thermal energy: The integral of the square of the torque of joint i is

$$\int_0^T \tau_i(s)^2 dt = \int_0^1 \frac{\tau_i(s)^2}{\dot{s}} ds = \int_0^1 \frac{\tau_i(s)^2}{\sqrt{b(s)}} ds. \tag{29}$$

This objective function is related to the thermal energy generated by actuator i. It can easily be shown [29] that $x^2/\sqrt{y}$ is a convex function of (x, y), for $y \geq 0$.

2) Integral of the absolute value of the rate of change of the torque: The integral of the absolute value of the rate of change of the torque of joint i is

$$\int_0^T |\dot{\tau}_i(s)|\, dt = \int_0^1 \frac{|\tau_i'(s)\dot{s}|}{\dot{s}} ds = \int_0^1 |\tau_i'(s)|\, ds, \tag{30}$$

since $\dot{s} \geq 0$ for $t \in [0, T]$. This objective function is convex, since $|x|$ is a convex function of x. While this objective function has no straightforward physical interpretation, incorporating this term reduces the number and magnitude of torque changes, which can be particularly useful for some cases as explained in Sec. VI.

B. Inequality constraints

In addition to torque constraints, other constraints may also be useful.

1) Velocity constraints: It is possible to incorporate velocity limits, which may be inspired by task specifications, by imposing symmetric lower bounds $-\overline{\dot{q}}_i(s)$ and upper bounds $\overline{\dot{q}}_i(s)$ on the velocity of joint i as follows

$$\begin{aligned} &-\overline{\dot{q}}_i(s) \leq \dot{q}_i(s) \leq \overline{\dot{q}}_i(s), \\ &\Leftrightarrow (\dot{q}_i(s))^2 = (q_i'(s)\dot{s})^2 = (q_i'(s))^2 b(s) \leq \left(\overline{\dot{q}}_i(s)\right)^2, \end{aligned} \tag{31}$$

or by imposing symmetric lower bounds and upper bounds on the translational components (v_x, v_y, v_z) or the rotational components $(\omega_x, \omega_y, \omega_z)$ of the operational space velocity. For example

$$\begin{aligned} &-\overline{v}_x(s) \leq v_x(s) \leq \overline{v}_x(s), \\ &\Leftrightarrow (v_x(s))^2 \leq (\overline{v}_x(s))^2. \end{aligned} \tag{32}$$

The operational space velocity components are related to the joint velocities by the robot Jacobian [28] $\mathbf{J}(\mathbf{q}) = (J_{ij}(\mathbf{q}))$ as follows

$$(v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^T = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}. \tag{33}$$

Hence, equation (32) can be rewritten as

$$\begin{aligned} (v_x(s))^2 &= \left(\sum_{i=1}^n J_{1i}(\mathbf{q}(s))\dot{q}_i(s)\right)^2 = \left(\sum_{i=1}^n J_{1i}(\mathbf{q}(s))q_i'(s)\dot{s}\right)^2 \\ &= \left(\sum_{i=1}^n J_{1i}(\mathbf{q}(s))q_i'(s)\right)^2 b(s) \leq (\overline{v}_x(s))^2. \end{aligned} \tag{34}$$

Note that the bounds defined by equations (31) and (34) can easily be rewritten in the following form

$$b(s) \leq \overline{b}(s), \tag{35}$$

and that hence, they can be interpreted directly as upper bounds on $b(s)$. Therefore, it is also sufficient to consider only the most restrictive upper bound for each $s \in [0, 1]$.

2) Acceleration constraints: It is also possible to incorporate lower bounds $\underline{\ddot{q}}_i(s)$ and upper bounds $\overline{\ddot{q}}_i(s)$ on the acceleration of joint i as follows

$$\begin{aligned}
&\underline{\ddot{q}}_i(s) \leq \ddot{q}_i(s) \leq \overline{\ddot{q}}_i(s), \\
&\Leftrightarrow \underline{\ddot{q}}_i(s) \leq q_i'(s)\ddot{s} + q_i''(s)\dot{s}^2 \leq \overline{\ddot{q}}_i(s), \\
&\Leftrightarrow \underline{\ddot{q}}_i(s) \leq q_i'(s)a(s) + q_i''(s)b(s) \leq \overline{\ddot{q}}_i(s). \qquad (36)
\end{aligned}$$

Similar as for velocity constraints, acceleration constraints can also be applied to the components of the operational space acceleration. It can be shown that both joint space and operational space acceleration constraints can be rewritten in the following form

$$\underline{f}(s) \leq f(s)a(s) + h(s)b(s) \leq \overline{f}(s). \qquad (37)$$

3) Rate of torque change constraints: It can be shown that constraints on the rate of change of the torque $\dot{\tau}_i(s)$, which are also useful to impose, are not convex and are therefore less attractive from an optimization point of view.

C. Generalized problem formulation

Combining the objective functions (8), (29) and (30) in an affine way[2] and incorporating the constraints (9), (31), (32) yields the following *generalized* optimal control problem

$$\begin{aligned}
\min_{a(\cdot),b(\cdot),\boldsymbol{\tau}(\cdot)} \int_0^1 \Bigg[\frac{1}{\sqrt{b(s)}} + \frac{\gamma_1}{\sqrt{b(s)}} \left(\sum_{i=1}^{n} \frac{\tau_i(s)^2}{\overline{\tau}_i^2} \right) \\
+\gamma_2 \left(\sum_{i=1}^{n} \frac{|\tau_i'(s)|}{|\overline{\tau}_i|} \right) \Bigg] ds, \qquad (38)
\end{aligned}$$

$$\begin{aligned}
\text{subject to } \boldsymbol{\tau}(s) &= \mathbf{m}(s)a(s) + \mathbf{c}(s)b(s) + \mathbf{g}(s), && (39)\\
b(0) &= \dot{s}_0^2, && (40)\\
b(1) &= \dot{s}_T^2, && (41)\\
b'(s) &= 2a(s), && (42)\\
b(s) &\geq 0, && (43)\\
b(s) &\leq \overline{b}(s), && (44)\\
\underline{\mathbf{f}}(s) \leq \mathbf{f}(s)a(s) &+ \mathbf{h}(s)b(s) \leq \overline{\mathbf{f}}(s), && (45)\\
\underline{\boldsymbol{\tau}}(s) \leq \boldsymbol{\tau}(s) &\leq \overline{\boldsymbol{\tau}}(s), && (46)\\
\text{for } s &\in [0,1],
\end{aligned}$$

where $\overline{\tau}_i$ for $i = 1 \ldots n$ are suitably chosen values, for example $\overline{\tau}_i = \max_{s\in[0,1]} \overline{\tau}_i(s)$. This generalized optimal control problem is convex, due to the convexity of the objective function and the inequality constraints, and the linearity of the system dynamics and the equality constraints.

V. Numerical solution

There are three ways to solve the generalized optimal control problem (38)-(46), namely dynamic programming as adopted in [11], [13], [24], indirect methods [19], or direct transcription as adopted in [18], [23] or direct single or multiple shooting [30]. In this section, a direct transcription method is proposed in Sec. V-A, which leads to a second-order cone program (SOCP) formulation in Sec. V-B.

A. Direct transcription

The direct transcription method consists of reformulating the optimal control problem (38)-(46) as a large sparse optimization problem. To this end, first, the path coordinate s is discretized on $[0, 1]$, which leads to $K + 1$ grid points $s^0 = 0 \leq s^k \leq 1 = s^K$, for $k = 0 \ldots K$. Second, the functions $b(s)$, $a(s)$ and $\tau_i(s)$ are modeled, by introducing a finite number of variables b^k, a^k, τ_i^k, which represent evaluations of these respective functions on the grid points or in between. The choice of the number of variables and the choice of the points at which they are evaluated characterize different direct transcription methods.

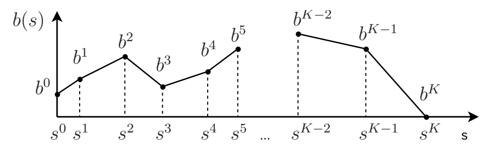

Fig. 1. $b(s)$ is piecewise linear for $s \in [0, 1]$.

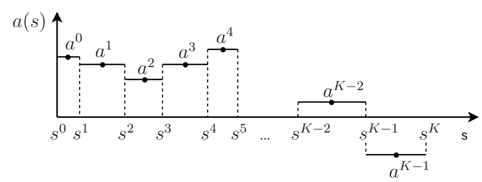

Fig. 2. $a(s)$ is piecewise constant for $s \in [0, 1]$.

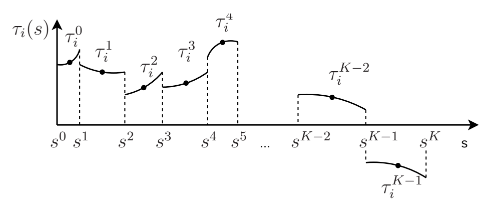

Fig. 3. $\tau_i(s)$ is piecewise nonlinear for $s \in [0, 1]$.

This paper proposes just one possibility. Since the velocity of a robotic manipulator cannot change discontinuously, it is assumed that $b(s)$ which is related to the velocity, is at least piecewise linear (Fig. 1). Based on this assumption and from equation (26), it follows that $a(s)$ must at least be piecewise constant (Fig. 2) and from equation (23), it follows that $\tau_i(s)$ is then in general piecewise nonlinear (Fig. 3). From these observations, it is natural to assign b^k on the grid points s^k. In other words,

$$b(s) = b^k + \left(\frac{b^{k+1} - b^k}{s^{k+1} - s^k} \right) (s - s^k), \qquad (47)$$

for $s \in [s^k, s^{k+1}]$ and hence, $b(s^k) = b^k$. a^k and τ_i^k are evaluated in the middle between the grid points s^k, namely

[2]Strictly speaking, the integral of the square of the torque for each actuator should be weighted with a factor which is proportional to the product of the motor resistance, the square of the motor gearing constant and the inverse of the square of the motor constant, as in [11]. Since these quantities are not always readily available, this paper adopts $\frac{1}{\overline{\tau}_i^2}$ instead.

on $s^{k+1/2} = (s^k + s^{k+1})/2$ (Figs. 2 and 3). After introducing the variables $a^k = a(s^{k+1/2})$ and $\boldsymbol{\tau}^k = \boldsymbol{\tau}(s^{k+1/2})$, for $k = 0 \dots K-1$ and given the fact that $b(s)$ is piecewise linear, the first two terms of the integral (38) can be approximated as

$$\begin{aligned}
&\int_0^1 \left[\frac{1}{\sqrt{b(s)}} + \frac{\gamma_1}{\sqrt{b(s)}}\left(\sum_{i=1}^n \frac{\tau_i(s)^2}{\overline{\tau}_i^2}\right)\right] ds \\
&= \sum_{k=0}^{K-1} \int_{s^k}^{s^{k+1}} \left[\frac{1}{\sqrt{b(s)}} + \frac{\gamma_1}{\sqrt{b(s)}}\left(\sum_{i=1}^n \frac{\tau_i(s)^2}{\overline{\tau}_i^2}\right)\right] ds \\
&\approx \sum_{k=0}^{K-1} \left[1 + \gamma_1 \left(\sum_{i=1}^n \frac{(\tau_i^k)^2}{\overline{\tau}_i^2}\right)\right] \int_{s^k}^{s^{k+1}} \frac{1}{\sqrt{b(s)}} ds. \quad (48)
\end{aligned}$$

To handle integrable singularities, that is, where $b(s) = 0$, $1/\sqrt{b(s)}$ is treated separately. Using equation (47) to calculate the integral in equation (48) analytically, the right-hand side of equation (48) can be rewritten as

$$\sum_{k=0}^{K-1} \left[1 + \gamma_1 \left(\sum_{i=1}^n \frac{(\tau_i^k)^2}{\overline{\tau}_i^2}\right)\right] \frac{2\Delta s^k}{\sqrt{b^{k+1}} + \sqrt{b^k}}, \quad (49)$$

where $\Delta s^k = s^{k+1} - s^k$. The third term of the integral (38) can be approximated for each i as

$$\int_0^1 \gamma_2 \frac{|\tau_i'(s)|}{|\overline{\tau}_i|} ds \approx \gamma_2 \sum_{k=1}^{K-1} \frac{|\Delta \tau_i^k|}{|\overline{\tau}_i|}, \quad (50)$$

where $\Delta\tau_i^k = \tau_i^k - \tau_i^{k-1}$ for $k = 1 \dots K-1$. After introducing the shorthand notation $b^{k+1/2} = (b^k + b^{k+1})/2$, the problem (38)-(46) can be rewritten in discretized form as a large scale optimization problem

$$\begin{aligned}
\min_{a^k, b^k, \boldsymbol{\tau}^k} \; & \sum_{k=0}^{K-1} \frac{2\Delta s^k (1 + \gamma_1 \sum_{i=1}^n (\tau_i^k)^2/\overline{\tau}_i^2)}{\sqrt{b^{k+1}} + \sqrt{b^k}}, \\
& + \gamma_2 \sum_{k=1}^{K-1} (\sum_{i=1}^n |\Delta\tau_i^k| / |\overline{\tau}_i|), \quad (51) \\
\text{subject to } & \boldsymbol{\tau}^k = \mathbf{m}(s^{k+1/2}) a^k \\
& + \mathbf{c}(s^{k+1/2}) b^{k+1/2} + \mathbf{g}(s^{k+1/2}), \quad (52) \\
& b^0 = \dot{s}_0^2, \quad (53) \\
& b^K = \dot{s}_T^2, \quad (54) \\
& (b^{k+1} - b^k) = 2a^k \Delta s^k, \quad (55) \\
& b^k \geq 0 \text{ and } b^K \geq 0, \quad (56) \\
& b^k \leq \overline{b}(s^k) \text{ and } b^K \leq \overline{b}(s^K), \quad (57) \\
& \underline{\mathbf{f}}(s^{k+1/2}) \leq \mathbf{f}(s^{k+1/2}) a^k + \mathbf{h}(s^{k+1/2}) b^{k+1/2}, \quad (58) \\
& \mathbf{f}(s^{k+1/2}) a^k + \mathbf{h}(s^{k+1/2}) b^{k+1/2} \leq \overline{\mathbf{f}}(s^{k+1/2}), \quad (59) \\
& \underline{\boldsymbol{\tau}}(s^{k+1/2}) \leq \boldsymbol{\tau}^k \leq \overline{\boldsymbol{\tau}}(s^{k+1/2}), \quad (60) \\
& \text{for } k = 0 \dots K-1.
\end{aligned}$$

Due to the convexity of problem (51)-(60), any local optimum is also globally optimal. Hence, the problem may be solved using any general purpose nonlinear solver. However, by rewriting problem (51)-(60) as a second-order cone program (SOCP), it can be solved even more efficiently, using a dedicated solver for these types of problems.

B. Second-order cone program formulation

An SOCP has the following standard form [29]

$$\begin{aligned}
\min_{\mathbf{x}} \; & \mathbf{f}^T \mathbf{x}, \quad (61) \\
\text{subject to } & \mathbf{F}\mathbf{x} = \mathbf{g}, \quad (62) \\
& \|\mathbf{M}_j \mathbf{x} + \mathbf{n}_j\|_2 \leq \mathbf{p}_j^T \mathbf{x} + q_j, \quad (63) \\
& \text{for } j = 1 \dots m.
\end{aligned}$$

Reformulating problem (51)-(60) in this form requires a number of steps. First, (51)-(60) can be reformulated as an equivalent problem with a linear objective function, by introducing variables d^k for $k = 0 \dots K-1$ and $\mathbf{e}^k \in \mathcal{R}^n$ for $k = 1 \dots K-1$, such that equation (51) can be rewritten as

$$\sum_{k=0}^{K-1} 2\Delta s^k d^k + \gamma_2 \sum_{k=1}^{K-1} \mathbf{1}^T \mathbf{e}^k, \quad (64)$$

where $\mathbf{1} \in \mathcal{R}^n$ is a vector with all elements equal to 1. It is then necessary to augment problem (51)-(60) with the inequality constraints

$$\frac{(1 + \gamma_1 \sum_{i=1}^n (\tau_i^k)^2/\overline{\tau}_i^2)}{\sqrt{b^{k+1}} + \sqrt{b^k}} \leq d^k, \text{ for } k = 0 \dots K-1, \quad (65)$$

and

$$-\mathbf{e}^k \leq \begin{pmatrix} \Delta\tau_1^k / |\overline{\tau}_1| \\ \dots \\ \Delta\tau_n^k / |\overline{\tau}_n| \end{pmatrix} \leq \mathbf{e}^k, \text{ for } k = 1 \dots K-1. \quad (66)$$

Second, to obtain an SOCP, constraints (65) can be replaced by two equivalent constraints by introducing variables c^k for $k = 0 \dots K$ as

$$\begin{aligned}
\frac{(1 + \gamma_1 \sum_{i=1}^n (\tau_i^k)^2/\overline{\tau}_i^2)}{c^{k+1} + c^k} \leq d^k, & \text{ for } k = 0 \dots K-1, \quad (67) \\
c^k \leq \sqrt{b^k}, & \text{ for } k = 0 \dots K. \quad (68)
\end{aligned}$$

Inequalities (67) and (68) can now be rewritten as two second-order cone constraints of the general form (63) as

$$\begin{aligned}
& \frac{(1 + \gamma_1 \sum_{i=1}^n (\tau_i^k)^2/\overline{\tau}_i^2)}{c^{k+1} + c^k} \leq d^k, \quad (69) \\
\Leftrightarrow & \left\| \begin{matrix} 2 \\ 2\sqrt{\gamma_1}\tau_1^k/\overline{\tau}_1 \\ \dots \\ 2\sqrt{\gamma_1}\tau_n^k/\overline{\tau}_n \\ c^{k+1} + c^k - d^k \end{matrix} \right\|_2 \leq c^{k+1} + c^k + d^k, \quad (70) \\
& \text{for } k = 0 \dots K-1, \quad (71)
\end{aligned}$$

and

$$\begin{aligned}
& c^k \leq \sqrt{b^k} \Leftrightarrow \left\| \begin{matrix} 2c^k \\ b^k - 1 \end{matrix} \right\|_2 \leq b^k + 1, \quad (72) \\
& \text{for } k = 0 \dots K. \quad (73)
\end{aligned}$$

Summarizing, an SOCP is obtained in standard form

$$\min_{a^k, b^k, \boldsymbol{\tau}^k, c^k, d^k, \mathbf{e}^k} \sum_{k=0}^{K-1} 2\Delta s^k d^k + \gamma_2 \sum_{k=1}^{K-1} \mathbf{1}^T \mathbf{e}^k, \tag{74}$$

$$\text{subject to } \boldsymbol{\tau}^k = \mathbf{m}(s^{k+1/2}) a^k + \mathbf{c}(s^{k+1/2}) b^{k+1/2} + \mathbf{g}(s^{k+1/2}), \tag{75}$$

$$b^0 = \dot{s}_0^2, \tag{76}$$

$$b^K = \dot{s}_T^2, \tag{77}$$

$$(b^{k+1} - b^k) = 2a^k \Delta s^k, \tag{78}$$

$$\underline{\boldsymbol{\tau}}(s^{k+1/2}) \leq \boldsymbol{\tau}^k \leq \overline{\boldsymbol{\tau}}(s^{k+1/2}), \tag{79}$$

$$\left\| \begin{matrix} 2 \\ 2\sqrt{\gamma_1} \tau_1^k / \overline{\tau}_1 \\ \dots \\ 2\sqrt{\gamma_1} \tau_n^k / \overline{\tau}_n \\ c^{k+1} + c^k - d^k \end{matrix} \right\|_2 \leq c^{k+1} + c^k + d^k, \tag{80}$$

$$\underline{\mathbf{f}}(s^{k+1/2}) \leq \mathbf{f}(s^{k+1/2}) a^k + \mathbf{h}(s^{k+1/2}) b^{k+1/2}, \tag{81}$$

$$\mathbf{f}(s^{k+1/2}) a^k + \mathbf{h}(s^{k+1/2}) b^{k+1/2} \leq \overline{\mathbf{f}}(s^{k+1/2}), \tag{82}$$

for $k = 0 \dots K-1$ and

$$\left\| \begin{matrix} 2c^k \\ b^k - 1 \end{matrix} \right\|_2 \leq b^k + 1, \tag{83}$$

$$b^k \geq 0, \tag{84}$$

$$b^k \leq \overline{b}(s^k), \tag{85}$$

for $k = 0 \dots K$ and

$$-\mathbf{e}^k \leq \begin{pmatrix} \Delta \tau_1^k / |\overline{\tau}_1| \\ \dots \\ \Delta \tau_n^k / |\overline{\tau}_n| \end{pmatrix} \leq \mathbf{e}^k \tag{86}$$

for $k = 1 \dots K-1$.

where a^k, $\boldsymbol{\tau}^k$, d^k are defined for $k = 0 \dots K-1$, b^k, c^k for $k = 0 \dots K$, and $\mathbf{e}^k$ for $k = 1 \dots K-1$. While the reformulation as an SOCP is not straightforward and requires the introduction of the auxiliary variables c^k, d^k and $\mathbf{e}^k$, it allows to use mature solvers [31] based on interior point methods [32], [33], that exploit the specific SOCP structure very efficiently. These dedicated methods have a *much better worst-case complexity* than methods for more general types of convex programs, such as semidefinite programs [34].

The formulation (74)-(86) can be implemented directly and very easily using the free high-level optimization modeling tool YALMIP [35], which also allows to test various solvers. Once a solution is obtained for the variables b^k, the relation $s(t)$ between the path coordinate and time can be constructed from the inverse relation $t(s)$, which is calculated as

$$t(s) = \int_0^s \frac{1}{\sqrt{b(u)}} du. \tag{87}$$

VI. NUMERICAL BENCHMARK

To illustrate the validity of the method discussed in Sec. V, it is applied to a double parabola and a circular path for a three-DOF elbow manipulator as in [13]. Using YALMIP as a modeling tool [35] and the free optimization software SeDuMi [31] which uses a primal-dual interior point method [32], the method presented in Sec. V is used to determine the time-optimal solutions for the parabolic and circular paths, with $K = 299$.

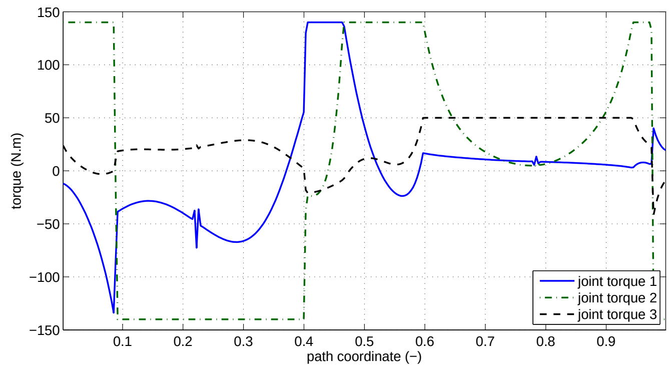

Fig. 4. The joint torques $\boldsymbol{\tau}$ as a function of the path coordinate s for the parabolic path for $\gamma_1 = 0$ and $\gamma_2 = 0$.

The resulting torques $\boldsymbol{\tau}$ as a function of the path coordinate s are shown in Fig. 4 for the parabola path and for $\gamma_1 = \gamma_2 = 0$. These torques are equal to those produced in [13] apart from numerical jitter around $s = 0.22$ and $s = 0.78$. This jitter is caused by singularities which are specific to the combination of manipulator and path [16]. In a singular point, one of the elements of the vector $\mathbf{m}(s)$ defined by equation (5) is equal to zero [16], and the solution to the time-optimal trajectory is not *uniquely* defined [14]. A very effective way to remove this jitter, which *hardly affects* the overall solution, is to add a small amount of regularization to *penalize torque jumps*, or in other words, choose $\gamma_2 > 0$. The resulting torques $\boldsymbol{\tau}$ as a function of the path coordinate s for the parabola path *with regularization* ($\gamma_2 = 10^{-6}$) are shown in Fig. 5. Similar results are obtained for the circular path.

The regularization illustrates very well the flexibility of the generalized problem formulation and the proposed solution method. Unwanted behavior is easily eliminated using an appropriate choice for the objective function. The effectiveness of this approach was also validated for the example presented in [16], which focusses in detail on the problem of singular points and singular arcs. Furthermore, the efficiency of the proposed solution method is evident from the solver times reported by YALMIP, namely 0.98 s for the parabola path without regularization, 1.29 s for the parabola path with regularization and 1.24 s for the circular path with regularization on an Intel Pentium 4 CPU running at 3.60 GHz. If $\boldsymbol{\tau}^k$ are eliminated as optimization variables, the solver times decrease to 0.74 s, 0.93 s and 0.89 s respectively [36].

VII. NUMERICAL EXAMPLE: WRITING TASK

To illustrate the versatility and practicality of the method discussed in Sec. V, it is applied to a more complex example involving a six-DOF manipulator. Sec. VII-A discusses the manipulator and the path, while the results are presented in Sec. VII-B.

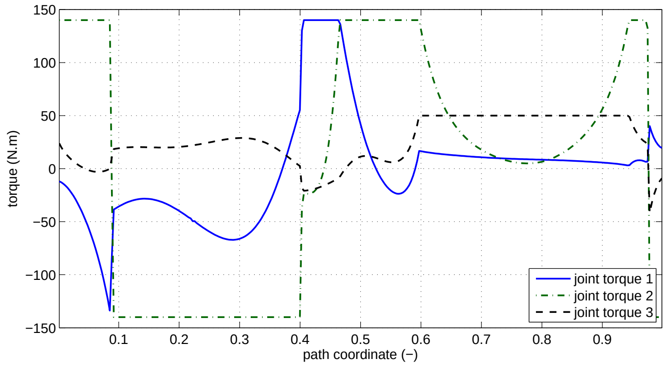

Fig. 5. The joint torques $\boldsymbol{\tau}$ as a function of the path coordinate s for the parabolic path for $\gamma_1 = 0$ and $\gamma_2 = 10^{-6}$.

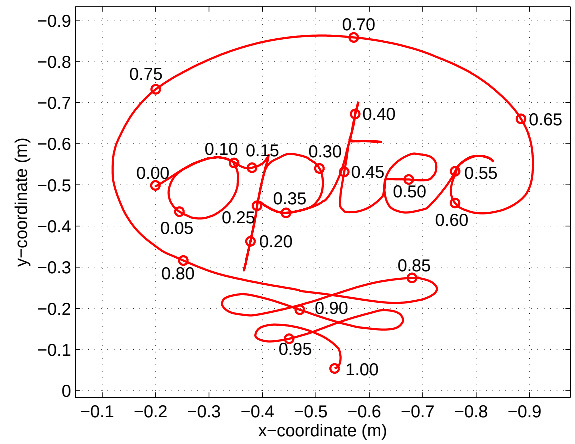

Fig. 7. The path in the plane parallel to the XY-plane.

A. Introduction

In the context of programming by human demonstration [37], it may be desirable for a manipulator to track a human-generated path, but not necessary or even undesirable to enforce the path-time relation established during the demonstration. In fact, time-optimal trajectory planning may speed up these types of tasks considerably.

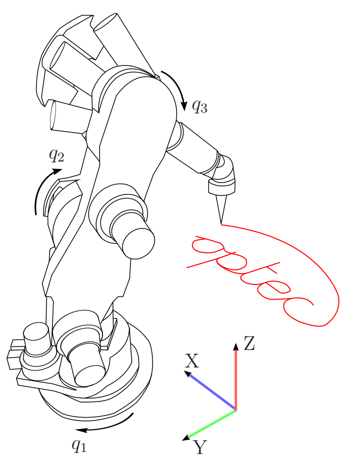

Fig. 6. A six-DOF KUKA 361 industrial manipulator performing a writing task.

Considered here is a six-DOF KUKA 361 industrial manipulator carrying out a complex writing task (Fig. 6). The objective is to write a text on a plane parallel to the XY-plane (Fig. 6), while keeping the end-effector oriented in the negative Z-direction at a height of $z = 1.2$ m. The end-effector path parallel to the XY-plane is shown separately in Fig. 7, where the value of the path coordinate $s \in [0, 1]$ along the path is shown in steps of 0.05. This type of path features long smooth segments as well as sharp edges at the transitions between different character segments. Therefore, a lot of switching is expected to realize the time-optimal trajectory, such that the grid on which the problem is solved needs to be sufficiently fine.

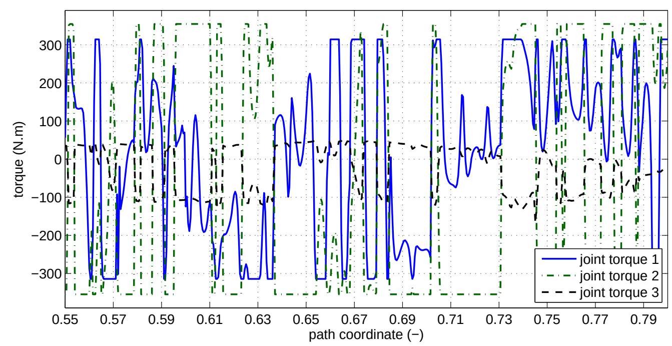

Fig. 8. The joint torques $\boldsymbol{\tau}$ as a function of the path coordinate s for the first three axes for the path shown in Fig. 7 and for $\gamma_1 = 0$ and $\gamma_2 = 0$.

B. Results

The method presented in Sec. V results in a minimal trajectory duration of 11.3 s ($K = 1999$). The corresponding torques $\boldsymbol{\tau}$ for the first[3] three axes are shown as a function of the path coordinate s in Fig. 8 ($s = 0.55 \ldots 0.8$). Figure 8 indicates that a lot of switching is necessary and that the torque for axis two, which is heavily loaded by gravity, is saturated for a considerable part along the path and therefore is the main limitation on the acceleration and deceleration ability of the manipulator.

Figure 9 shows the joint velocities for the first three axes for $s = 0.55 \ldots 0.8$. Although interpretation is in general difficult, it can be intuitively understood that a low velocity is required for the sharp edge of the letter “c” at $s \approx 0.57$, while the subsequent long smooth arc can be executed at a much higher velocity. Despite the relative non-smoothness of the path, sharp segments of the path pose no problem to the solution method. By solving the problem for $K = 999$, 1999 and 2999, it is verified that the obtained solution is, aside from sampling effects, grid-independent.

When solving the problem on an Intel Pentium 4 CPU running at 3.60 GHz and with $\boldsymbol{\tau}^k$ eliminated as optimization

[3]The torques for the last three axes are omitted, because the first three axes are more heavily loaded.

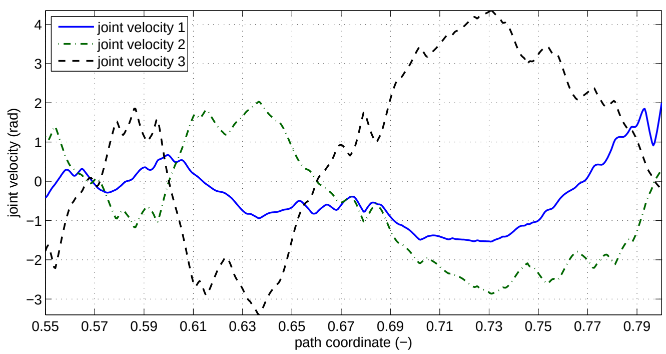

Fig. 9. The joint velocities $\dot{\mathbf{q}}$ as a function of the path coordinate s for the first three axes for the path shown in Fig. 7 and for $\gamma_1 = 0$ and $\gamma_2 = 0$.

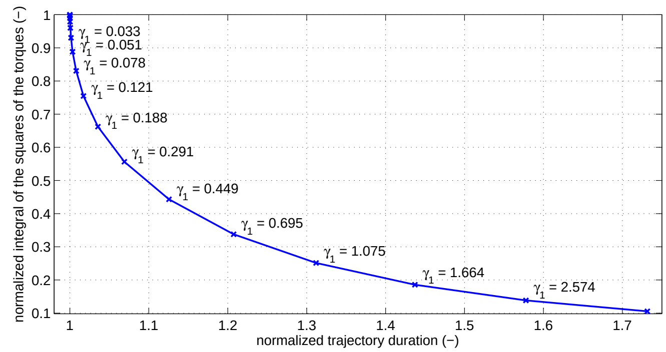

Fig. 10. The normalized trajectory duration versus the normalized integral of the squares of the torques.

variables, YALMIP reports a solver time of 2.87 s. To allow the reader to perform own optimization studies or to allow a fair comparison with other algorithms, a downloadable Matlab implementation has been made available [36], as well as a software-rendered video showing the manipulator carrying out the trajectory.

C. Time-optimality versus energy-optimality

The very limited solver times allow to calculate the solution of (74)-(86) for a large number of different weighting factors γ_1 and γ_2, in order to investigate their effect on the time-optimality of the solution. Here, time-optimality and energy-optimality are traded-off by varying γ_1. One important motivation for having a nonzero $\gamma_1 > 0$ is to limit the rate of change of the torques [23], such that the actuators can better handle the torque demand. Other approaches to limit the rate of change of the torques consist of directly imposing upper and lower bounds on this rate of change [17], [23], or choosing a parameterization for the pseudo-acceleration $\ddot{s}$ that is at least piecewise continuous [23]. Another important motivation for choosing $\gamma_1 > 0$, on the other hand, is to limit the thermal energy dissipated by the actuators, so as to prevent actuator overheating if the task is carried out repeatedly [38].

Figure 10 shows the relation between the trajectory duration T and the thermal actuator energy $\sum_{i=1}^{n} \int_0^T (\tau_i(t)^2/\overline{\tau}_i^2)dt$ if γ_1 is varied between 0 and $10^{0.6}$, while keeping $\gamma_2 = 0$. Both objectives are normalized through division by their values for $\gamma_1 = 0$. The resulting trade-off curve reveals that a 10% increase in trajectory duration, results in a spectacular 50% reduction of the thermal energy dissipated by the actuators, while a further increase in trajectory duration to 20% results in only 65% reduction of the thermal energy. These results, as well as the steep trade-off close to $\gamma_1 = 0$, numerically quantify the engineering intuition that true time-optimality comes at a significant energy cost.

Fig. 10 illustrates very well the use and the versatility of the generalized optimal control formulation (38)-(46) and the usefulness of an *efficient numerical algorithm* to investigate the effect of trading off time-optimality against other criteria.

VIII. DISCUSSION

The convex reformulation and accompanying SOCP-based solution method presented in Secs. III and V, are very appealing from both a theoretical and a numerical point of view: the global optimum of time-energy optimal planning is guaranteed to be found within a few CPU seconds of computation time.

None of the methods which consider more than just time-optimality [11], [13], [18], [23], [24], except [19], can provide a theoretical guarantee of finding the global optimum. The theoretical merit of our formulation with respect to [19]. however, is that first, the proof of global optimality is extremely simple as it follows directly from the convexity of the optimal control problem. Secondly, the formulation allows to easily devise convexity-preserving extensions, as illustrated in Sec. IV-A and IV-B.

With respect to numerical efficiency, it is difficult to compare our method to the other methods [7], [10], [11], [13], [14], [16], [18], [19], [21]–[24]: unfortunately, to the best of the authors' knowledge, no detailed solution times and actual implementations are publicly available for most of the methods, some of which were published nearly two decades ago. Hence, it is not possible to assess whether these methods also result in the same global optimum for the complex example of Sec. VII, nor whether they give rise to shorter computational times. Even if these methods were faster, the authors would perceive this as only a minor shortcoming of the present method, since the computational times reported here are sufficiently short to be practical in academic and industrial practice. In order to make future numerical benchmarking possible and to allow use by practitioners, the authors have provided a freely downloadable Matlab implementation under the GNU public license [36].

Two other key aspects of the presented method are the ease of implementation and flexibility. With regard to flexibility, the indirect methods [7], [10], [13], [14], [16], [21], [22], except [19] which considers time-energy optimality, can only take into account time-optimality and torque constraints and therefore do not feature the same flexibility as any of the dynamic programming [11], [13], [24] and direct transcription methods [18], [19], [23]. The latter two categories of methods do not restrict the choice of constraints nor objective function, except that the objective is generally assumed to be an affine combination of trajectory duration with other criteria. The price to be paid is that the global optimum is not guaranteed to be found. Our method is more restricted in that sense, since only a limited, yet, in our opinion, sufficiently rich set of objective functions and constraints can be handled. In fact, this limited set constitutes the very *essence of the efficiency* of the presented solution method, and, rather than perceiving it as restrictive, it can also be seen as a *theoretical foundation* to evaluate the tractability of more general formulations.

With regard to ease of implementation, it is clear from the discussion in Sec. I that the indirect methods [7], [10], [13], [14], [16], [19], [21], [22] are procedural in nature and require implementation of a number of substeps, all of which require the choice of a method, tolerances and stopping criteria. The dynamic programming approaches [11], [13], [24] are also procedural in nature although their implementation involves less numerical choices since they are decision-based rather than gradient-based. However, because of their sequential nature and large amount of variables, an efficient implementation requires careful thought. Direct transcription methods [18], [23] are more straightforward to implement because they are based on solving nonlinear programs for which, theoretically, any nonlinear solver can be used. Numerical efficiency, however, requires an in-depth numerical analysis of the problem structure. Conversely, the method in Sec. V requires only the choice of a transcription scheme, and the main difficulty lies in enforcing the SOCP structure in the resulting program. The *actual implementation* is, however, very easy. Moreover, SOCP solvers such as SeDuMi [31], are freely available.

A first part of future work will focus on improving the experimental applicability of the time-optimal trajectory method and consists of mitigating three key issues. First, the limited bandwidth of any physical actuator implies that infinitely fast torque jumps cannot be realized and therefore, it will be investigated how constraints on the rate of changes on the torques can be efficiently taken into account in the presented method. Second, the torques calculated by the time-optimal trajectory planning algorithm are purely feedforward and need to be implemented in conjunction with a feedback controller. To prevent the feedback controller from becoming unstable, it is necessary to impose bounds on the actuator torques which are lower than the actually allowable actuator torques. This consideration is especially important in the presence of significant modeling errors. Third, it will be investigated how viscous and other joint velocity dependent friction phenomena can be accommodated for in the presented method. Finally, the practical applicability of the presented method will be demonstrated experimentally. A second part of future work will focus on the application of the presented method to path-constrained trajectory planning problems which consider, in addition to time-optimality, minimization of reaction forces at the robot base.

Acknowledgment

Bram Demeulenaere is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen). The authors gratefully acknowledge the financial support by K.U.Leuven's Concerted Research Action GOA/05/10, K.U.Leuven's CoE EF/05/006 Optimization in Engineering Center (OPTEC), the Prof. R. Snoeys Foundation, and the Belgian Program on Interuniversity Poles of Attraction IAP VI/4 DYSCO (Dynamic Systems, Control and Optimization) initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture.

References

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

[2] O. Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of Operations Research*, vol. 37, 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," *Journal of Optimization Theory and Applications*, vol. 84, no. 3, pp. 589–616, 1995.

[4] V. Schulz, "Reduced SQP methods for large-scale optimal control problems in DAE with application to path planning problems for satellite mounted robots," Ph.D. dissertation, Universität Heidelberg, 1996.

[5] H. Choset, W. Burgard, S. Hutchinson, G. Kantor, L. E. Kavraki, K. Lynch, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, June 2005.

[6] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast Motions in Biomechanics and Robotics Optimization and Feedback Control*, ser. Lecture Notes in Control and Information Sciences, K. Mombaur, Ed. Springer-Verlag, 2006, pp. 65–94.

[7] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, June 1985.

[8] Z. Shiller and S. Dubowsky, "Robot path planning with obstacles, actuator, gripper and payload constraints," *The International Journal of Robotics Research*, vol. 8, no. 6, pp. 3–18, 1989.

[9] ——, "On computing the global time-optimal motions of robotic manipulators in the presence of obstacles," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 786–797, 1991.

[10] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[11] K. G. Shin and N. D. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Transactions on Automatic Control*, vol. AC-31, no. 6, pp. 491–500, 1986.

[12] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, San Fransisco, CA, 1986, pp. 1399–1405.

[13] ——, "A concept for manipulator trajectory planning," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 2, pp. 115–123, 1987.

[14] J.-J. E. Slotine and H. S. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 118–124, February 1989.

[15] Y. Chen and A. Desrochers, "Structure of minimum-time control law for robotic manipulators withconstrained paths," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, 1989, pp. 971–976 vol.2.

[16] Z. Shiller and H.-H. Lu, "Computation of path constrained time optimal motions with dynamic singularities," *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, vol. 114, pp. 34–40, March 1992.

[17] M. Tarkiainen and Z. Shiller, "Time optimal motions of manipulators with actuator dynamics," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Atlanta, GA, 1993, pp. 725–730 vol.2.

[18] J. T. Betts and W. P. Huffman, "Path-constrained trajectory optimization using sparse sequential quadratic programming," *Journal of Guidance, Control and Dynamics*, vol. 16, no. 1, pp. 59–68, 1993.

[19] Z. Shiller, "Time-energy optimal control of articulated systems with geometricpath constraints," in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, 1994, pp. 2680–2685.

[20] ——, "On singular time-optimal control along specified paths," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 561–566, 1994.

[21] H. X. Phu, H. G. Bock, and J. Schlöder, "Extremal solutions of some constrained control problems," *Optimization*, vol. 35, no. 4, pp. 345–355, 1995.

[22] ——, "The method of orienting curves and its application for manipulator trajectory planning," *Numerical Functional Analysis and Optimization*, vol. 18, pp. 213–225, 1997.

[23] D. Constantinescu and E. A. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *Journal of Robotic Systems*, vol. 17, no. 5, pp. 233–249, 2000.

[24] S. Singh and M. C. Leu, "Optimal trajectory generation for robotic manipulators using dynamic programming," *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, vol. 109, no. 2, pp. 88–96, 1987.

[25] L. Biegler, "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation," *Computers and Chemical Engineering*, vol. 8, pp. 243–248, 1984.

[26] O. Stryk, "Numerical solution of optimal control problems by direct collocation," in *Optimal Control: Calculus of Variations, Optimal Control Theory and Numerical Methods*. Bulirsch et al., 1993, vol. 129.

[27] J. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*. Philadelphia: SIAM, 2001.

[28] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*. McGraw-Hill, 1996.

[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004. [Online]. Available: http://www.ee.ucla.edu/∼vandenbe/cvxbook.html

[30] H. Bock and K. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," in *Proceedings 9th IFAC World Congress Budapest*. Pergamon Press, 1984, pp. 243–247. [Online]. Available: http://www.iwr.uni-heidelberg.de/groups/agbock/FILES/Bock1984.pdf

[31] J. Sturm, "Using sedumi: a matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999.

[32] S. Wright, *Primal-Dual Interior-Point Methods*. Philadelphia: SIAM Publications, 1997.

[33] Y.-J. Kuo and H. Mittelmann, "Interior point methods for second-order cone programming and OR applications," *Computational Optimization and Applications*, vol. 28, no. 3, pp. 255–285, 2004.

[34] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second order cone programming," *Linear Algebra and its Applications*, vol. 284, pp. 193–228, 1998.

[35] J. Löfberg, *YALMIP, Yet another LMI parser.* Sweden: University of Linköping, 2001, available: http://www.control.isy.liu.se/∼johanl.

[36] D. Verscheure, "Practical time-optimal trajectory planning for robots: a convex optimization approach." [Online]. Available: http://homes.esat.kuleuven.be/∼optec/software/timeopt/

[37] W. Meeussen, "Compliant robot motion: from path planning or human demonstration to force controlled task execution," Ph.D. dissertation, Katholieke Universiteit Leuven, Department of Mechanical Engineering, Dec. 2006.

[38] M. Guilbert, P.-B. Wieber, and L. Joly, "Optimal trajectory generation for manipulator robots under thermal constraints," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.