Time-Energy Optimal Path Tracking for Robots: a Numerically Efficient Optimization Approach

Diederik Verscheure*, Bram Demeulenaere*, Jan Swevers*, Joris De Schutter* and Moritz Diehl†
*Division PMA, Department of Mechanical Engineering
Katholieke Universiteit Leuven, Belgium
Email: diederik.verscheure@mech.kuleuven.be
†Division SCD, Department of Electrical Engineering
Katholieke Universiteit Leuven, Belgium

***Abstract*— This paper focuses on time-optimal and time-energy optimal path tracking, which are subproblems in optimal motion planning of robot systems. Through a nonlinear change of variables, the time-energy optimal path tracking problem is transformed here into a convex optimal control problem with a single state variable. 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. Application to a six-DOF KUKA 361 industrial robot carrying out a writing task illustrates the practicality of the new method.**

I. Introduction

Time-optimal and related motion planning is of significant importance for maximizing the productivity of robot systems. Instead of solving the motion planning problem directly in the system's state space, the decoupled approach [1] which solves the motion planning problem in two stages, is often preferred for its lower computational requirements. 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 [2], 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 [3]–[14]. To achieve less agressive use of the actuators, time-optimality is often traded off against other criteria such as energy [11].

The path tracking stage constitutes the focus of this paper. Methods for time-optimal and related path tracking problems subject to actuator constraints have been proposed in [3]–[15]. 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 $\dot{s}$ [3], [4], [6]. Hence, the multi-dimensional state space of a robotic manipulator can be reduced to a *two-dimensional state space*.

The first category of methods are indirect methods, which have have been proposed in [3], [4], [6] and subsequently refined in [7], [9], [11], [13]. These methods are based on numerical searches, and forward and backward integrations to determine the points along the trajectory where changes between the active actuator constraints occur and which unambiguously determine the solution of the path tracking problem.

The second category consists of dynamic programming methods [5], [6], [15], while the third category consists of direct transcription methods [10], [14]. In contrast with most of the indirect methods which consider only time-optimality, except [11] 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.

In this paper, the time-energy optimal path tracking problem, discussed in Sec. II, is transformed into a *convex* optimal control problem with a single state variable through a nonlinear change of variables introduced in Sec. III. Section IV shows that direct transcription [16] by simultaneous 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 V illustrates the practicality of the new method by means of an example of a six-DOF KUKA 361 industrial robot carrying out a writing task. Section VI contrasts the proposed solution method with the existing indirect methods [3], [4], [6], [7], [9], [11], [13], dynamic programming methods [5], [6], [15] and direct transcription methods [10], [14] 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 [17]

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

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 [6], 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 $s(t)$. Without loss of generality, it is assumed that the trajectory starts at $t = 0$, ends at $t = T$ and that $s(0) = 0 \leq s(t) \leq 1 = s(T)$. For the given path, the joint velocities and accelerations can be rewritten using the chain rule as

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

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

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 [3]

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

where

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

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

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

and where $\mathrm{sgn}(\dot{\mathbf{q}}(s))$ is replaced by $\mathrm{sgn}(\mathbf{q}'(s))$ using equation (2) and the fact that the velocity $\dot{s}$ along the path must be positive for a trajectory which is time-optimal.

Similarly as in [11], the time-energy optimal path tracking problem can be expressed as

$$\min_{T, s(\cdot), \boldsymbol{\tau}(\cdot)} T + \gamma \int_0^T \sum_{i=1}^{n} \frac{\tau_i(t)^2}{\overline{\tau}_i^2} dt, \tag{8}$$

$$\text{subject to } \boldsymbol{\tau}(t) = \mathbf{m}(s(t))\ddot{s}(t) + \mathbf{c}(s(t))\dot{s}(t)^2 + \mathbf{g}(s(t)), \tag{9}$$

$$s(0) = 0, \tag{10}$$

$$s(T) = 1, \tag{11}$$

$$\dot{s}(0) = \dot{s}_0, \tag{12}$$

$$\dot{s}(T) = \dot{s}_T, \tag{13}$$

$$\dot{s}(t) \geq 0, \tag{14}$$

$$\underline{\boldsymbol{\tau}}(s(t)) \leq \boldsymbol{\tau}(t) \leq \overline{\boldsymbol{\tau}}(s(t)), \quad \text{for } t \in [0, T], \tag{15}$$

where the torque lower bounds $\underline{\boldsymbol{\tau}}$ and upper bounds $\overline{\boldsymbol{\tau}}$ may depend on s, where γ is a suitably chosen factor to trade off time against energy optimality and where $\overline{\tau}_i$ for $i = 1 \ldots n$ are suitably chosen values, for example $\overline{\tau}_i = \max_{s \in [0,1]} \overline{\tau}_i(s)$. 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 [11], based on a reformulation, the time-energy optimal control problem is shown to be convex, such that it can be concluded that any local optimum of the problem is also globally optimal. This paper introduces, 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

$$T + \gamma \int_0^T \sum_{i=1}^{n} \frac{\tau_i(t)^2}{\overline{\tau}_i^2} dt = \int_0^1 \left[\frac{1}{\dot{s}} + \frac{\gamma}{\dot{s}} \sum_{i=1}^{n} \frac{\tau_i(s)^2}{\overline{\tau}_i^2} \right] ds. \tag{16}$$

Second, $a(s) = \ddot{s}$ and $b(s) = \dot{s}^2$ are introduced as optimization variables and supplemented with an additional constraint

$$b'(s) = 2a(s), \tag{17}$$

which follows from the observation that $\dot{b}(s) = b'(s)\dot{s}$, as well as $\dot{b}(s) = \frac{d(\dot{s}^2)}{dt} = 2\ddot{s}\dot{s} = 2a(s)\dot{s}$. While this nonlinear transformation is already recognized in [6], 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

$$\min_{a(\cdot), b(\cdot), \boldsymbol{\tau}(\cdot)} \int_0^1 \left[\frac{1}{\sqrt{b(s)}} + \frac{\gamma}{\sqrt{b(s)}} \sum_{i=1}^{n} \frac{\tau_i(s)^2}{\overline{\tau}_i^2} \right] ds, \tag{18}$$

$$\text{subject to } \boldsymbol{\tau}(s) = \mathbf{m}(s)a(s) + \mathbf{c}(s)b(s) + \mathbf{g}(s), \tag{19}$$

$$b(0) = \dot{s}_0^2, \tag{20}$$

$$b(1) = \dot{s}_T^2, \tag{21}$$

$$b'(s) = 2a(s), \tag{22}$$

$$b(s) \geq 0, \tag{23}$$

$$\underline{\boldsymbol{\tau}}(s) \leq \boldsymbol{\tau}(s) \leq \overline{\boldsymbol{\tau}}(s), \quad \text{for } s \in [0, 1]. \tag{24}$$

Problem (18)-(24) is convex since all constraints (19)-(24) are *linear*, while the objective function (18) is convex.

Problem (18)-(24) 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 (22) and subject to *linear* state dependent constraints (19), (23)-(24), as well as initial and terminal constraints (20) and (21) respectively.

In contrast with the formulation in [11], the reformulated problem (18)-(24) has *only one* differential state, while time does *not appear explicitly* in the formulation anymore. However, the true merit of the reformulation (18)-(24), is that, unlike the reformulation in [11], first, it is clear without proof that problem (18)-(24) 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*. For the sake of brevity, a list of possible objective functions and constraints is omitted and this paper discusses only time-energy optimal path tracking.

[1]For a path given in operational space coordinates, inverse kinematics techniques can be used to obtain the corresponding path in joint space coordinates [7], [17].

IV. Numerical solution

The optimal control problem (18)-(24) can be solved using dynamic programming [5], [6], [15], indirect methods [11], direct transcription [10], [14] or direct single or multiple shooting [18]. Section IV-A proposes a direct transcription method, which leads to a second-order cone program (SOCP) formulation in Sec. IV-B.

A. Direct transcription

The direct transcription method consists of reformulating the optimal control problem (18)-(24) 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 $k = 0 \dots K$. 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.

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. Based on this assumption and from equation (22), it follows that $a(s)$ must at least be piecewise constant and from equation (19), it follows that $\tau_i(s)$ is then in general piecewise nonlinear. From these observations, it is natural to assign b^k on the grid points s^k. In other words,

$$b(s) = b^k + \left(\frac{b^{k+1} - b^k}{s^{k+1} - s^k}\right)(s - s^k), \tag{25}$$

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 at $s^{k+1/2} = (s^k + s^{k+1})/2$. 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 integral (18) can be approximated as

$$\begin{aligned}\int_0^1 \left[\frac{1}{\sqrt{b(s)}} + \frac{\gamma}{\sqrt{b(s)}} \sum_{i=1}^n \frac{\tau_i(s)^2}{\overline{\tau}_i^2}\right] ds \\ \approx \sum_{k=0}^{K-1} \left[1 + \gamma \sum_{i=1}^n \frac{(\tau_i^k)^2}{\overline{\tau}_i^2}\right] \int_{s^k}^{s^{k+1}} \frac{1}{\sqrt{b(s)}} ds.\end{aligned} \tag{26}$$

To handle integrable singularities, that is, where $b(s) = 0$, $1/\sqrt{b(s)}$ is treated separately. Using equation (25) to calculate the integral in equation (26) analytically, the right-hand side of equation (26) can be rewritten as

$$\sum_{k=0}^{K-1} \left[1 + \gamma \sum_{i=1}^n \frac{(\tau_i^k)^2}{\overline{\tau}_i^2}\right] \frac{2\Delta s^k}{\sqrt{b^{k+1}} + \sqrt{b^k}}, \tag{27}$$

where $\Delta s^k = s^{k+1} - s^k$. After introducing the shorthand notation $b^{k+1/2} = (b^k + b^{k+1})/2$, problem (18)-(24) is rewritten in discretized form as a large scale optimization problem

$$\min_{a^k, b^k, \boldsymbol{\tau}^k} \sum_{k=0}^{K-1} \frac{2\Delta s^k (1 + \gamma \sum_{i=1}^n (\tau_i^k)^2/\overline{\tau}_i^2)}{\sqrt{b^{k+1}} + \sqrt{b^k}}, \tag{28}$$

$$\text{subject to } \boldsymbol{\tau}^k = \mathbf{m}(s^{k+1/2}) a^k + \mathbf{c}(s^{k+1/2}) b^{k+1/2} + \mathbf{g}(s^{k+1/2}), \tag{29}$$

$$b^0 = \dot{s}_0^2, \tag{30}$$

$$b^K = \dot{s}_T^2, \tag{31}$$

$$(b^{k+1} - b^k) = 2a^k \Delta s^k, \tag{32}$$

$$b^k \geq 0 \text{ and } b^K \geq 0, \tag{33}$$

$$\underline{\boldsymbol{\tau}}(s^{k+1/2}) \leq \boldsymbol{\tau}^k \leq \overline{\boldsymbol{\tau}}(s^{k+1/2}), \tag{34}$$

$$\text{for } k = 0 \dots K-1.$$

Due to the convexity of problem (28)-(34), any local optimum is also globally optimal such that the problem may be solved using any general purpose nonlinear solver. However, by rewriting problem (28)-(34) 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 [19]

$$\min_{\mathbf{x}} \ \mathbf{f}^T \mathbf{x}, \tag{35}$$

$$\text{subject to } \mathbf{F}\mathbf{x} = \mathbf{g}, \tag{36}$$

$$\|\mathbf{M}_j \mathbf{x} + \mathbf{n}_j\|_2 \leq \mathbf{p}_j^T \mathbf{x} + q_j, \quad \text{for } j = 1 \dots m. \tag{37}$$

Reformulating problem (28)-(34) in this form requires a number of steps. First, (28)-(34) can be reformulated as an equivalent problem with a linear objective function, by introducing variables d^k for $k = 0 \dots K-1$, such that objective function (28) can be rewritten as

$$\sum_{k=0}^{K-1} 2\Delta s^k d^k. \tag{38}$$

It is then necessary to augment problem (28)-(34) with the inequality constraints

$$\frac{(1 + \gamma \sum_{i=1}^n (\tau_i^k)^2/\overline{\tau}_i^2)}{\sqrt{b^{k+1}} + \sqrt{b^k}} \leq d^k, \text{ for } k = 0 \dots K-1, \tag{39}$$

Second, to obtain an SOCP, constraints (39) can be replaced by two equivalent constraints by introducing variables c^k for $k = 0 \dots K$ as

$$\frac{(1 + \gamma \sum_{i=1}^n (\tau_i^k)^2/\overline{\tau}_i^2)}{c^{k+1} + c^k} \leq d^k, \text{ for } k = 0 \dots K-1, \tag{40}$$

$$c^k \leq \sqrt{b^k}, \text{ for } k = 0 \dots K. \tag{41}$$

Inequalities (40) and (41) can then be rewritten as two second-order cone constraints (47) and (48) of the general form (37),

thus yielding an SOCP in standard form

$$\min_{a^k,b^k,\boldsymbol{\tau}^k,c^k,d^k,\mathbf{e}^k} \sum_{k=0}^{K-1} 2\Delta s^k d^k, \tag{42}$$

$$\text{subject to } \boldsymbol{\tau}^k = \mathbf{m}(s^{k+1/2})a^k + \mathbf{c}(s^{k+1/2})b^{k+1/2} + \mathbf{g}(s^{k+1/2}), \tag{43}$$

$$b^0 = \dot{s}_0^2 \text{ and } b^K = \dot{s}_T^2, \tag{44}$$

$$(b^{k+1} - b^k) = 2a^k\Delta s^k, \tag{45}$$

$$\underline{\boldsymbol{\tau}}(s^{k+1/2}) \leq \boldsymbol{\tau}^k \leq \overline{\boldsymbol{\tau}}(s^{k+1/2}), \tag{46}$$

$$\left\| \begin{array}{c} 2 \\ 2\sqrt{\gamma}\tau_1^k/\overline{\tau}_1 \\ \cdots \\ 2\sqrt{\gamma}\tau_n^k/\overline{\tau}_n \\ c^{k+1}+c^k-d^k \end{array} \right\|_2 \leq c^{k+1}+c^k+d^k, \tag{47}$$

for $k = 0 \ldots K-1$, and

$$\left\| \begin{array}{c} 2c^k \\ b^k - 1 \end{array} \right\|_2 \leq b^k + 1, \tag{48}$$

$$b^k \geq 0, \tag{49}$$

for $k = 0 \ldots K$.

where a^k, $\boldsymbol{\tau}^k$, d^k are defined for $k = 0 \ldots K-1$ and b^k, c^k for $k = 0 \ldots K$. While the formulation as an SOCP is not straightforward and requires auxiliary variables c^k and d^k, it allows us to use mature solvers [20] that exploit the specific SOCP structure very efficiently and which have a *much better worst-case complexity* than solvers for more general types of convex programs, such as semidefinite programs [21].

The formulation (42)-(49) can be implemented directly and very easily using the free high-level optimization modeling tool YALMIP [22], which also allows us to test various solvers.

V. Numerical Example: Writing Task

While it is verified that the method in Sec. IV is able to reproduce results such as in [6], it is applied to a more complex example involving a six-DOF manipulator, to illustrate the practicality of the method. Sec. V-A discusses the manipulator and the path, while the results are presented in Sec. V-B.

A. Introduction

In the context of programming by human demonstration [23], 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. Time-optimal path tracking can speed up these types of tasks considerably.

Considered here is a six-DOF KUKA 361 industrial manipulator carrying out a complex writing task (Fig. 1). The objective is to write a text on a plane parallel to the XY-plane (Fig. 1), 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. 2, where the value of the path coordinate $s \in [0, 1]$ along the path is

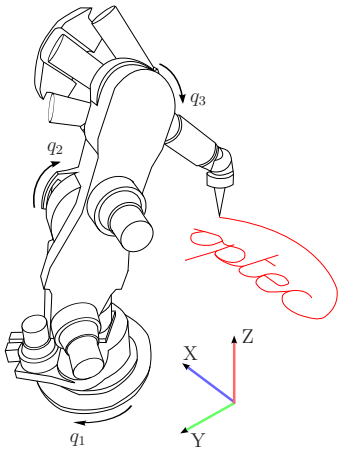

Fig. 1. A six-DOF KUKA 361 industrial manipulator performing a writing task.

shown in steps of 0.05. This type of path features long smooth segments as well as sharp edges 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.

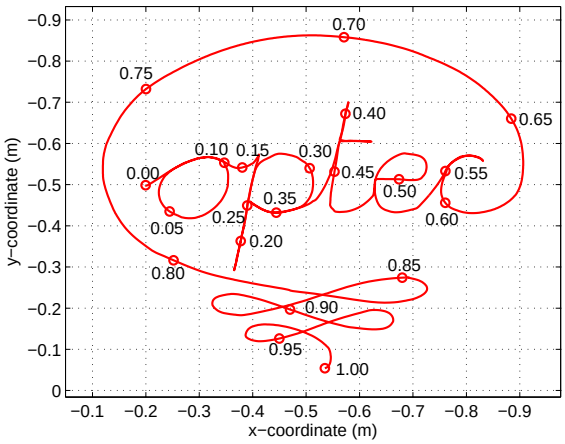

Fig. 2. The path in the plane parallel to the XY-plane.

B. Time-optimal path tracking results

Considering at first only time-optimality ($\gamma = 0$), the method in Sec. IV yields a trajectory duration of 11.3 s ($K = 1999$), while the corresponding torques $\boldsymbol{\tau}$ for the first[2] three axes are shown as a function of the path coordinate s in Fig. 3 ($s = 0.55 \ldots 0.8$). Figure 3 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. Figure 4 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

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

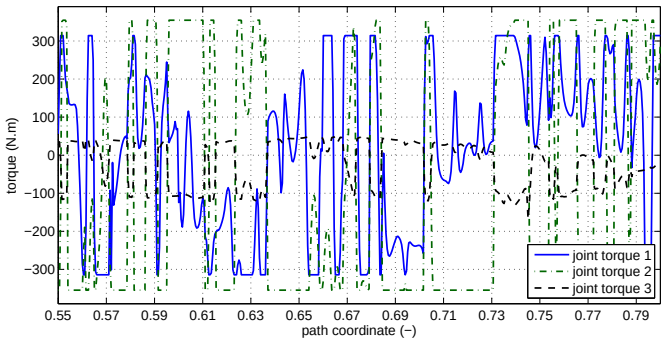

Fig. 3. The joint torques $\boldsymbol{\tau}$ as a function of the path coordinate s for the first three axes for the path shown in Fig. 2 and for $\gamma = 0$.

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.

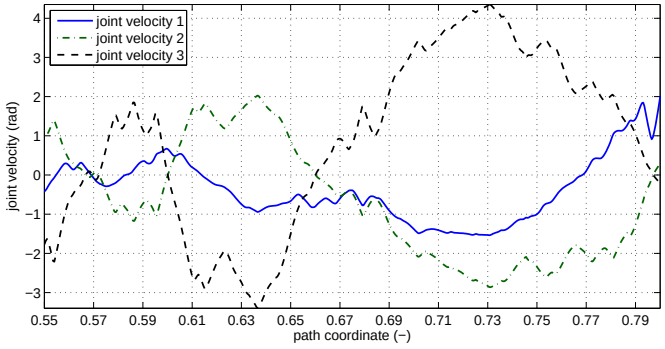

Fig. 4. The joint velocities $\dot{\mathbf{q}}$ as a function of the path coordinate s for the first three axes for the path shown in Fig. 2 and for $\gamma = 0$.

When solving the problem on an Intel Pentium 4 CPU running at 3.60 GHz and with $\boldsymbol{\tau}^k$ and a^k eliminated as optimization variables, YALMIP reports a solver time of 2.87 s. To allow the reader to perform own optimization studies and to allow a fair comparison with other algorithms, a downloadable Matlab implementation has been made available [24], as well as a software-rendered video showing the manipulator carrying out the trajectory.

C. Time-optimality versus energy-optimality

The limited solver times allow us to calculate the solution of (42)-(49) for a large number of different weighting factors γ to investigate the effect on the time-optimality of the solution. Here, time-optimality and energy-optimality are traded-off by varying γ. Two important motivations for having a nonzero $\gamma > 0$ is to limit the rate of change of the torques [14], such that the actuators can better handle the torque demand and to limit the thermal energy dissipated by the actuators, so as to prevent actuator overheating if the task is carried out repeatedly [25].

Figure 5 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 is varied between 0 and $10^{0.6}$. Both objectives are normalized through division by their values for $\gamma = 0$. 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 (labeled (1) in Fig. 5), while a further increase in trajectory duration to 20% results in only 65% reduction of the thermal energy (labeled (2) in Fig. 5). These results, as well as the steep trade-off close to $\gamma = 0$, numerically quantify the engineering intuition that true time-optimality comes at a significant energy cost. Fig. 5 illustrates very well the use and the practicality of an *efficient numerical algorithm* to investigate the effect of trading off time-optimality against other criteria such as energy.

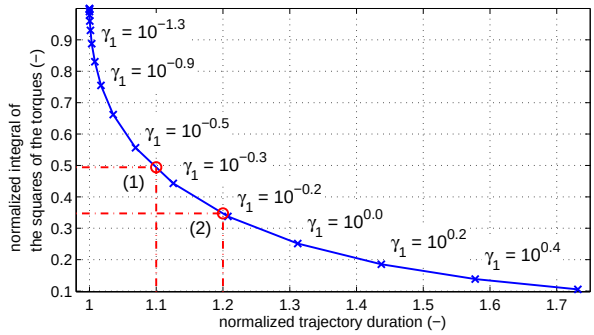

Fig. 5. The normalized trajectory duration versus the normalized integral of the squares of the torques.

VI. Discussion

The reformulation and SOCP-based solution method presented in Secs. III and IV 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.

Unfortunately, it is difficult to compare the numerical efficiency of our method to that of other methods [3]–[7], [9]–[11], [13]–[15], since, to the best of the authors' knowledge, no detailed solution times and actual implementations are publicly available for most of these methods. However, even if these methods were faster, the authors would perceive this as only a minor shortcoming of the presented method, since the computational times are sufficiently short to be practical in academic and industrial practice.

Two other key aspects of the presented method are the flexibility and ease of implementation. With regard to flexibility, the indirect methods [3], [4], [6], [7], [9], [13], except [11], can only take into account time-optimality and actuator constraints, and therefore do not feature the same flexibility as the dynamic programming [5], [6], [15] and direct transcription methods [10], [11], [14]. The latter two categories of methods do not restrict the choice of constraints

nor objective function. The price to be paid is that the global optimum is not guaranteed to be found. Although a list of possible objective functions and constraints is omitted, our method is more restricted and can only handle a limited set of objective functions and constraints. However, this limited set also constitutes the very *essence of the efficiency* of the presented method.

With regard to ease of implementation, the indirect methods [3], [4], [6], [7], [9], [11], [13] 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 [5], [6], [15] are also procedural in nature although their implementation involves less numerical choices. Direct transcription methods [10], [14] are more straightforward to implement, but a numerically efficient implementation requires an in-depth numerical analysis. Conversely, the method in Sec. IV 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, while the efficiency of the implementation follows from the use of an SOCP solver.

Future work will focus on improving the experimental applicability of the time-optimal path tracking method and consists of mitigating two key issues. First, the limited bandwidth of any physical actuator implies that infinitely fast torque jumps cannot be realized. Therefore, it will be investigated how the presented method can be extended to efficiently take into account constraints on the rate of change of the torques. Second, it will be investigated how viscous and other joint velocity dependent friction phenomena can be accommodated for in the presented method.

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] 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.

[2] Z. Shiller and S. Dubowsky, "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.

[3] 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.

[4] 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.

[5] ——, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Transactions on Automatic Control*, vol. AC-31, no. 6, pp. 491–500, 1986.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] H. X. Phu, H. G. Bock, and J. Schlöder, "The method of orienting curves and its application for manipulator trajectory planning," *Numerical Functional Analysis and Optimization*, vol. 18, pp. 213–225, 1997.

[14] 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.

[15] 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.

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

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

[18] 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

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

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

[21] 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.

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

[23] 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.

[24] D. Verscheure, "Practical time-optimal trajectory planning for robots: a convex optimization approach." [Online]. Available: http://people.mech.kuleuven.be/∼dversche/timeopt/timeopt.html

[25] 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.