2013 American Control Conference (ACC)
Washington, DC, USA, June 17-19, 2013

On the Time-optimal Trajectory Planning and Control of Robotic Manipulators Along Predefined Paths†

Pedro Reynoso-Mora[1], Wenjie Chen[1], and Masayoshi Tomizuka[1]

Abstract— **In this paper we study the problem of time-optimal trajectory planning and control of robotic manipulators along predefined paths. An algorithm that generates dynamically feasible time-optimal trajectories and controls is presented, which considers the complete dynamic model with both Coulomb and viscous friction. Even though the effects of viscous friction for fast motions become more significant than Coulomb friction, in previous formulations viscous friction was ignored. We propose a formulation that naturally leads to a convex relaxation which solves exactly the original non-convex formulation. In order to numerically solve the proposed formulation, a discretization scheme is also developed. Through simulation and experimental studies on the resulting tracking errors, applied torques, and accelerometer readings for a 6-axis manipulator, we emphasize the significance of penalizing a measure of total jerk. The importance of imposing acceleration constraints at the initial and final transitions of the trajectory is also studied.**

I. INTRODUCTION

Time-optimality in robotic manipulators is crucial for maximizing robot productivity. In many industrial applications of robotic arms, such as palletizing and pick-and-place, an operator specifies a collision-free path that the robot must follow in order to accomplish a particular task. This path specification is usually done through a so-called teach-pendant or through a path planning algorithm [1]. Once the path has been specified, it is important to find out how to move the robot along that path in the shortest time dynamically feasible; first successful approaches to tackle this problem date back to the 1980s [2], [3], [4]. In the approaches [2], [3], search algorithms were proposed that find switching points in the so-called velocity limit curve. At these switching points, instant changes from maximum acceleration to maximum deceleration and vice versa must occur. In [4], additional properties of the velocity limit curve were pointed out, which allow for simplification in the computation of the switching points.

Over the past decades, a prominent approach to numerical optimal control has emerged [5], in which the original optimal control problem is transcribed directly to a non-linear optimization problem. In the specific context of time-optimal trajectories and controls for robotic manipulators along predefined paths, a modern formulation that uses tools from convex optimization was proposed in [6]. Even though the effects of viscous friction become noticeably more significant than Coulomb friction, none of the existing formulations take into account the complete dynamic model. In this paper, we propose a formulation that incorporates the complete dynamic model, which results in a non-convex optimal control problem. It is then shown how to relax the referred non-convex formulation so as to obtain a convex optimal control problem. Because of the way in which both, the non-convex and the convex formulations are constructed, it turns out the convex relaxation solves exactly the original non-convex problem.

Aided by simulations and experimental studies of the tracking errors, applied torques, and accelerometer readings on a 6-axis manipulator, we point out the importance of: (i) penalizing a measure of total jerk, and (ii) imposing acceleration constraints at the initial and final transitions of the trajectory. Likewise, a discretization scheme is presented that allows a straightforward incorporation of these acceleration constraints. Thus, optimal trajectories and controls that are more realistic than the purely time-optimal solutions are generated, at the cost of modest increase in the traversal time. The optimal trajectories and controls are termed dynamically feasible since the complete dynamic model is incorporated into the formulation. It is thus verified that the solutions produced by the algorithm are indeed dynamically feasible with respect to the complete dynamic model.

In Section II the problem formulation as a mathematical optimization is presented. Then in Section III a discretization scheme is presented that allows to obtain a numerical solution. In Section IV an application to a 6-axis industrial manipulator is presented; from simulations on a realistic simulator, it is argued that pure time-optimal solutions need to be slightly modified before being implemented on the real robot. Then, in Section V, specific details for incorporating acceleration constraints and for penalizing a measure of total jerk are presented; experimental results are also presented to report on the benefits of the proposed methodology. Conclusions are presented in Section VI.

II. PROBLEM FORMULATION

In order to generate trajectories and open-loop controls that are dynamically feasible, the dynamic model should take into account the most significant effects present in the physical system. For an $n-$DOF manipulator, the equations of motion take the following standard form [7]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{D}_{\mathrm{v}}\dot{\mathbf{q}} + \mathbf{F}_C\,\mathrm{sgn}(\dot{\mathbf{q}}) = \boldsymbol{\tau}, \quad (1)$$

†The first author acknowledges UC MEXUS-CONACYT for providing him with a fellowship to support his PhD studies. Also, the financial and technical supports from FANUC Corporation are acknowledged.

[1]Department of Mechanical Engineering, University of California at Berkeley, 94720 CA, USA, {preynoso, wjchen, tomizuka}@me.berkeley.edu

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ are the joint-space positions, velocities, and accelerations, respectively; $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of applied joint torques; $\mathbf{M}(\mathbf{q}), \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ are the inertia and Coriolis/centrifugal matrices, respectively; $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ represents the vector of gravitational torques; diagonal matrices $\mathbf{D}_{\mathrm{v}}, \mathbf{F}_{\mathrm{C}} \in \mathbb{R}^{n \times n}$ represent the coefficients of viscous and Coulomb friction, respectively.

A. Mathematical Formulation

Assume that the geometric path, along which the robot is required to move, has been determined using for example a teach-pendant or a path planning algorithm [1]. Then the goal is to determine how to move along that path in the shortest time possible. The reference trajectory is denoted by $\mathbf{q}_{\mathrm{d}} \in \mathbb{R}^n$, it is then desired to obtain a 4-tuple $(\mathbf{q}_{\mathrm{d}}, \dot{\mathbf{q}}_{\mathrm{d}}, \ddot{\mathbf{q}}_{\mathrm{d}}, \boldsymbol{\tau}_{\mathrm{d}})$ that is dynamically feasible with respect to (1) and that guaranties motion in the shortest time possible. In this paper, the geometric path is represented by $\mathbf{h}(s)$, where s is a monotonically increasing parameter $s(t) \in [0, 1]$. It is required that $\mathbf{h}(s)$ be twice continuously differentiable in s.

It is readily shown from the equality constraint $\mathbf{q}_{\mathrm{d}} = \mathbf{h}(s)$, that $\dot{\mathbf{q}}_{\mathrm{d}} = \mathbf{h}'(s)\dot{s}$ and $\ddot{\mathbf{q}}_{\mathrm{d}} = \mathbf{h}''(s)\dot{s}^2 + \mathbf{h}'(s)\ddot{s}$, where $\mathbf{h}'(s)$ and $\mathbf{h}''(s)$ denote the first and second derivatives of $\mathbf{h}(s)$ with respect to s, respectively. Since the 4-tuple $(\mathbf{q}_{\mathrm{d}}, \dot{\mathbf{q}}_{\mathrm{d}}, \ddot{\mathbf{q}}_{\mathrm{d}}, \boldsymbol{\tau}_{\mathrm{d}})$ must be dynamically feasible with respect to (1), the expressions for $\dot{\mathbf{q}}_{\mathrm{d}}$ and $\ddot{\mathbf{q}}_{\mathrm{d}}$ are substituted into (1). Using standard properties of dynamic model (1), and after some algebraic manipulations, it is obtained:

$$\boldsymbol{\tau}_{\mathrm{d}} = \mathbf{a}_1(s)\ddot{s} + \mathbf{a}_2(s)\dot{s}^2 + \mathbf{a}_3(s)\dot{s} + \mathbf{a}_4(s), \tag{2}$$

where $\mathbf{a}_i(s) \in \mathbb{R}^n$, $i = 1, \ldots, 4$, are defined as:

$$\begin{aligned}
\mathbf{a}_1(s) &:= \mathbf{M}\left(\mathbf{h}(s)\right)\mathbf{h}'(s) \\
\mathbf{a}_2(s) &:= \mathbf{M}\left(\mathbf{h}(s)\right)\mathbf{h}''(s) + \mathbf{C}\left(\mathbf{h}(s), \mathbf{h}'(s)\right)\mathbf{h}'(s) \\
\mathbf{a}_3(s) &:= \mathbf{D}_{\mathrm{v}}\mathbf{h}'(s) \\
\mathbf{a}_4(s) &:= \mathbf{F}_{\mathrm{C}}\,\mathrm{sgn}\left(\mathbf{h}'(s)\right) + \mathbf{g}\left(\mathbf{h}(s)\right).
\end{aligned}$$

Note that $\mathbf{a}_i(s)$, $i = 1, \ldots, 4$, are entirely known since $\mathbf{h}(s)$, $\mathbf{h}'(s)$, and $\mathbf{h}''(s)$ are known. The unknowns in parametrization (2) are $\ddot{s}$, $\dot{s}^2$, $\dot{s}$, and $\boldsymbol{\tau}_{\mathrm{d}}$. Consider therefore defining $a(s) := \ddot{s}$, $b(s) := \dot{s}^2$, $c(s) := \dot{s}$, and $\boldsymbol{\tau}_{\mathrm{d}}(s)$, which are to be determined as functions of s. Since a one-to-one relationship between s and t will be enforced (i.e., $\dot{s} > 0$), finding the unknowns as functions of s implies that they can be unambiguously recovered as functions of t.

In this manner, $\boldsymbol{\tau}_{\mathrm{d}}(s)$ has a simple affine parametrization in $a(s)$, $b(s)$, and $c(s)$, namely, $\boldsymbol{\tau}_{\mathrm{d}}(s) = \mathbf{a}_1(s)a(s) + \mathbf{a}_2(s)b(s) + \mathbf{a}_3(s)c(s) + \mathbf{a}_4(s)$. Likewise, with the definitions of $a(s)$, $b(s)$, and $c(s)$, two additional constraints must be incorporated: (i) $\dot{b}(s) = b'(s)\dot{s} = 2\dot{s}\ddot{s} \Leftrightarrow b'(s) = 2a(s)$ if $\dot{s} > 0$, (ii) $c(s) = \sqrt{b(s)}$. On the other hand, the total traversal time is denoted by t_f and can be rewritten in a different manner, using the fact that $\dot{s} > 0$:

$$t_f = \int_0^{t_f} \mathrm{d}t = \int_0^1 \left(\frac{\mathrm{d}s}{\mathrm{d}t}\right)^{-1} \mathrm{d}s = \int_0^1 \frac{1}{c(s)}\,\mathrm{d}s. \tag{3}$$

It is important to consider the case when the initial and final pseudo-speeds are zero, i.e., $\dot{s}_0 = \dot{s}_f = 0$. In that case, it is clear that the objective functional (3) is unbounded above. In order to overcome this limitation, the integral in (3) is defined instead in the interval $[0_\alpha, 1_\alpha]$, where $0_\alpha = 0_+$ and $1_\alpha = 1_-$ will be formally defined in Section III.

Consider therefore the following optimization problem:

$$\begin{aligned}
\underset{a(s), b(s), c(s), \boldsymbol{\tau}_{\mathrm{d}}(s)}{\text{minimize}} \quad & \int_{0_\alpha}^{1_\alpha} \frac{1}{c(s)}\,\mathrm{d}s \\
\text{subject to} \quad & b(0) = \dot{s}_0^2,\ b(1) = \dot{s}_f^2 \\
& c(0) = \dot{s}_0,\ c(1) = \dot{s}_f \\
& \boldsymbol{\tau}_{\mathrm{d}}(s) = \mathbf{a}_1(s)a(s) + \mathbf{a}_2(s)b(s) \\
& \qquad\qquad + \mathbf{a}_3(s)c(s) + \mathbf{a}_4(s) \\
& \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau}_{\mathrm{d}}(s) \leq \overline{\boldsymbol{\tau}} \\
& \forall s \in [0, 1] \\
& b'(s) = 2a(s),\ c(s) = \sqrt{b(s)} \\
& b(s), c(s) > 0 \\
& \forall s \in [0_\alpha, 1_\alpha]
\end{aligned} \tag{4}$$

where $\underline{\boldsymbol{\tau}}, \overline{\boldsymbol{\tau}} \in \mathbb{R}^n$ represent the lower and upper bounds on the torques, and $\dot{s}_0, \dot{s}_f$ represent the initial and final pseudo-speeds along the path, respectively.

It can be argued that formulation (4) is a non-convex optimal control problem [5], [6], [8]. The non-convexity of (4) is due to the non-linear equality constraint $c(s) = \sqrt{b(s)}$ $\forall s \in [0, 1]$, which shows up because of viscous friction. For fast motions, viscous friction has a significant effect in mechanical systems. Therefore instead of ignoring this term, a convex relaxation that solves exactly the non-convex formulation (4) is proposed.

B. Convex Relaxation

Since $b(s), c(s) > 0$ for all $s \in [0_\alpha, 1_\alpha]$, the following chain of equivalences is true:

$$\begin{aligned}
\sqrt{b(s)} = c(s) &\Leftrightarrow \frac{1}{\sqrt{b(s)}} = \frac{1}{c(s)} \\
&\Leftrightarrow \frac{1}{\sqrt{b(s)}} \leq \frac{1}{c(s)},\ \frac{1}{c(s)} \leq \frac{1}{\sqrt{b(s)}} \\
&\Leftrightarrow c(s)^2 - b(s) \leq 0,\ \ -c(s)^2 + b(s) \leq 0,
\end{aligned}$$

where the inequality constraint $c(s)^2 - b(s) \leq 0$ is convex since $f(b, c) = c^2 - b$ is a convex function of b, c [8]. On the other hand, $-c(s)^2 + b(s) \leq 0$ is a concave inequality constraint. Having concave inequality constraints in an optimization problem implies that the problem is non-convex. It is then proposed to drop the concave constraint, and replace the equality constraint $c(s) = \sqrt{b(s)}$ in (4) with the convex inequality constraint $1/\sqrt{b(s)} \leq 1/c(s)$.

Because the functional in problem (4) is minimized and since $c(s) > 0$, at optimum the inequality $1/\sqrt{b(s)} \leq 1/c(s)$ must be active, i.e., $1/\sqrt{b^\star(s)} = 1/c^\star(s)$ $\forall s \in [0_\alpha, 1_\alpha]$. This entails that for all $s \in [0, 1]$ the solution $a^\star(s)$, $b^\star(s)$, $c^\star(s)$, $\boldsymbol{\tau}_{\mathrm{d}}^\star(s)$ to the convex relaxation solves exactly the non-convex

problem (4). In order to obtain a convenient formulation of the convex relaxation, note that $\forall s \in [0_\alpha, 1_\alpha]$ [9]:

$$\frac{1}{\sqrt{b(s)}} \leq \frac{1}{c(s)} \Leftrightarrow c(s)^2 \leq b(s) \cdot 1 \Leftrightarrow \left\| \begin{bmatrix} 2c(s) \\ b(s)-1 \end{bmatrix} \right\|_2 \leq b(s)+1.$$

Likewise, in order to have a linear objective functional, assume $\exists d(s) > 0$ satisfying $d(s) \geq 1/c(s)\ \forall s \in [0_\alpha, 1_\alpha]$, then note that:

$$d(s) \geq \frac{1}{c(s)} \Leftrightarrow 1 \leq c(s)d(s) \Leftrightarrow \left\| \begin{bmatrix} 2 \\ c(s)-d(s) \end{bmatrix} \right\|_2 \leq c(s)+d(s).$$

Therefore, the relaxed problem results in:

$$\begin{aligned} \underset{a(s),b(s),c(s),\boldsymbol{\tau}_\mathrm{d}(s),d(s)}{\text{minimize}} \quad & \int_{0_\alpha}^{1_\alpha} d(s)\ \mathrm{d}s \\ \text{subject to} \quad & b(0) = \dot{s}_0^2,\ b(1) = \dot{s}_f^2 \\ & c(0) = \dot{s}_0,\ c(1) = \dot{s}_f \\ & \boldsymbol{\tau}_\mathrm{d}(s) = \mathbf{a}_1(s)a(s) + \mathbf{a}_2(s)b(s) \\ & \qquad + \mathbf{a}_3(s)c(s) + \mathbf{a}_4(s) \\ & \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau}_\mathrm{d}(s) \leq \overline{\boldsymbol{\tau}} \\ & \forall s \in [0,1] \\ & b'(s) = 2a(s) \\ & \left\| \begin{bmatrix} 2c(s) \\ b(s)-1 \end{bmatrix} \right\|_2 \leq b(s)+1 \\ & \left\| \begin{bmatrix} 2 \\ c(s)-d(s) \end{bmatrix} \right\|_2 \leq c(s)+d(s) \\ & b(s), c(s) > 0 \\ & \forall s \in [0_\alpha, 1_\alpha], \end{aligned} \tag{5}$$

which is a convex optimization problem.

III. PROBLEM DISCRETIZATION

In order to obtain a numerical solution to problem (5), a discretization scheme should be developed. As discussed later in Section V, it is important to impose acceleration constraints, so that smooth acceleration transitions from/to zero are guaranteed at the initial/final points of the trajectory. Unlike the approach in [6], where a mid grid had to be defined since $a(s)$ was assumed piecewise constant, we follow the approach to solve optimal control problems presented in [10]. This approach, known as cubic collocation at Lobatto points, will allow us to incorporate the required acceleration constraints in a straightforward manner.

A. Cubic Collocation at Lobatto Points

First, a grid is created by discretizing the path parameter s into N points, $s_1 = 0 < s_2 < \cdots < s_N = 1$. Then, it is assumed that $b(s)$ and $a(s)$ are piecewise cubic and piecewise linear, respectively. Finally, the optimization variables are defined in the following manner: $a_1 = a(s_1), \ldots, a_N = a(s_N)$, $b_1 = b(s_1), \ldots, b_N = b(s_N)$, $c_1 = c(s_1), \ldots, c_N = c(s_N)$, $d_1 = d(s_1), \ldots, d_N = d(s_N)$, $\boldsymbol{\tau}^1 = \boldsymbol{\tau}_\mathrm{d}(s_1), \ldots, \boldsymbol{\tau}^N = \boldsymbol{\tau}_\mathrm{d}(s_N)$, which represent the functions $a(s), b(s), c(s), d(s)$, and $\boldsymbol{\tau}_\mathrm{d}(s)$ evaluated at the grid points $s_1 < s_2 < \cdots < s_N$.

The pseudo-acceleration $a(s)$ in (5) is chosen as piecewise linear, i.e.,

$$a(s) = a_j + (a_{j+1} - a_j)\left(\frac{s - s_j}{s_{j+1} - s_j}\right), \quad s \in [s_j, s_{j+1}] \tag{6}$$

$j = 1, 2, \ldots, N-1$, from which it is noticed that indeed $a(s_j) = a_j$ and $a(s_{j+1}) = a_{j+1}$. The squared pseudo-speed $b(s)$ is chosen as piecewise cubic, i.e.,

$$b(s) = \sum_{k=0}^{3} \beta_{j,k} \left(\frac{s - s_j}{s_{j+1} - s_j}\right)^k, \quad s \in [s_j, s_{j+1}] \tag{7}$$

$j = 1, 2, \ldots, N-1$, where the polynomial coefficients $\beta_{j,0}, \beta_{j,1}, \beta_{j,2}$, and $\beta_{j,3}$ need to be determined explicitly. The above representation for $b(s)$ must satisfy $b(s_j) = b_j$, $b(s_{j+1}) = b_{j+1}$, $b'(s_j) = 2a(s_j)$, and $b'(s_{j+1}) = 2a(s_{j+1})$, which gives 4 equations for the four unknowns $\beta_{j,0}, \beta_{j,1}, \beta_{j,2}$, and $\beta_{j,3}$. After solving, it is obtained:

$$\begin{aligned} \beta_{j,0} &= b_j, \quad \beta_{j,1} = 2\Delta s_j a_j \\ \beta_{j,2} &= 3(b_{j+1} - b_j) - 2\Delta s_j(a_{j+1} + 2a_j) \\ \beta_{j,3} &= 2\Delta s_j(a_{j+1} + a_j) - 2(b_{j+1} - b_j), \end{aligned}$$

where $\Delta s_j := s_{j+1} - s_j$, $j = 1, 2, \ldots, N-1$.

The approximating functions of $a(s)$ and $b(s)$ in (6)-(7) must satisfy $b'(s) = 2a(s)$ at the grid points s_j, $j = 1, \ldots, N$, and at the middle points $\bar{s}_j := (s_j + s_{j+1})/2$, $j = 1, \ldots, N-1$. The coefficients $\beta_{j,0}, \beta_{j,1}, \beta_{j,2}$, and $\beta_{j,3}$ given above already guarantee fulfillment of the constraints at grid points s_j. Therefore, the only constraints that need to be incorporated result from requiring $b'(\bar{s}_j) = 2a(\bar{s}_j)$, $j = 1, \ldots, N-1$. After standard algebraic simplifications, it is shown that this is equivalent to:

$$b_{j+1} - b_j = \Delta s_j(a_{j+1} + a_j), \quad j = 1, \ldots, N-1. \tag{8}$$

Note that enforcing these constraints implies that the coefficients $\beta_{j,3}$ above are zero, which means that actually $b(s)$ is piecewise quadratic. Since $c(s)^2 = b(s)$, it is then reasonable to assume $c(s)$ is piecewise linear.

Define $0_\alpha := (1-\alpha)s_1 + \alpha s_2$ and $1_\alpha := (1-\alpha)s_N + \alpha s_{N-1}$, with $\alpha > 0$ being a small adjustable parameter. Since $d(s)$ is assumed piecewise linear, the objective functional is discretized as follows:

$$\begin{aligned} \int_{0_\alpha}^{1_\alpha} d(s)\ \mathrm{d}s &= \int_{0_\alpha}^{s_2} d(s)\ \mathrm{d}s + \sum_{k=2}^{N-2} \int_{s_k}^{s_{k+1}} d(s)\ \mathrm{d}s + \int_{s_{N-1}}^{1_\alpha} d(s)\ \mathrm{d}s \\ &\approx \frac{1}{2}\left[(1-\alpha)\Delta s_1(d(0_\alpha) + d_2) \right. \\ &\quad + \sum_{k=2}^{N-2} \Delta s_k(d_k + d_{k+1}) \\ &\quad \left. + (1-\alpha)\Delta s_{N-1}(d_{N-1} + d(1_\alpha))\right], \end{aligned} \tag{9}$$

where $d(0_\alpha) = (1-\alpha)d_1 + \alpha d_2$ and $d(1_\alpha) = \alpha d_{N-1} + (1-\alpha)d_N$. The constraints in problem (5) are discretized by simply being evaluated at the grid points s_j, $j = 1, \ldots, N$. For those constraints that are not defined at $s_1 = 0$ and $s_N = 1$, they are evaluated at 0_α and 1_α instead. Therefore, the following convex optimization problem is obtained:

$$
\begin{aligned}
\underset{a_k, b_k, c_k, \boldsymbol{\tau}^k, d_k}{\text{minimize}} \quad & \frac{1}{2}\left[(1-\alpha)\Delta s_1 (d(0_\alpha) + d_2)\right. \\
& + \sum_{k=2}^{N-2} \Delta s_k (d_k + d_{k+1}) \\
& \left. + (1-\alpha)\Delta s_{N-1}(d_{N-1} + d(1_\alpha))\right] \\
\text{subject to} \quad & b_1 = \dot{s}_0^2, \; b_N = \dot{s}_f^2 \\
& c_1 = \dot{s}_0, \; c_N = \dot{s}_f \\
& \boldsymbol{\tau}^k = \mathbf{a}_1(s_k) a_k + \mathbf{a}_2(s_k) b_k \\
& \qquad + \mathbf{a}_3(s_k) c_k + \mathbf{a}_4(s_k) \\
& \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau}^k \leq \overline{\boldsymbol{\tau}} \\
& \left\| \begin{bmatrix} 2c_k \\ b_k - 1 \end{bmatrix} \right\|_2 \leq b_k + 1 \\
& \text{for } k = 1, \ldots, N \\
& b_{j+1} - b_j = \Delta s_j (a_{j+1} + a_j) \\
& \text{for } j = 1, \ldots, N-1 \\
& b_l > 0, \; c_l > 0 \\
& \left\| \begin{bmatrix} 2 \\ c_l - d_l \end{bmatrix} \right\|_2 \leq c_l + d_l \qquad (10) \\
& \text{for } l = 2, \ldots, N-1 \\
& \left\| \begin{bmatrix} 2 \\ c(0_\alpha) - d(0_\alpha) \end{bmatrix} \right\|_2 \leq c(0_\alpha) + d(0_\alpha) \\
& \left\| \begin{bmatrix} 2 \\ c(1_\alpha) - d(1_\alpha) \end{bmatrix} \right\|_2 \leq c(1_\alpha) + d(1_\alpha)
\end{aligned}
$$

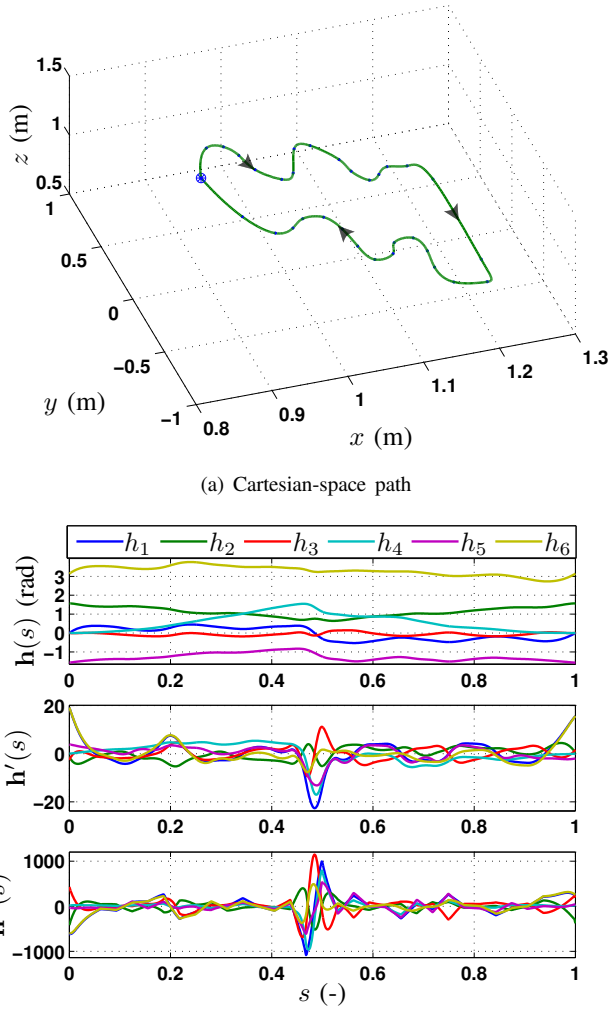
where $c(0_\alpha) = (1-\alpha)c_1 + \alpha c_2$ and $c(1_\alpha) = \alpha c_{N-1} + (1-\alpha)c_N$. The discretized problem (10) is a Second Order Cone Program (SOCP), which is a special class of convex optimization problems [6], [8], [9]. This optimization problem is readily coded and solved using CVX, which is a MATLAB® software for discipline convex optimization.

Notice that by solving problem (10), the optimal functions $a^\star(s), b^\star(s), c^\star(s), d^\star(s)$, and $\boldsymbol{\tau}_{\mathrm{d}}^\star(s)$, evaluated at $s_1 = 0 < s_2 < \cdots < s_N = 1$, are obtained. The optimal 4-tuple $(\mathbf{q}_{\mathrm{d}}^\star(t), \dot{\mathbf{q}}_{\mathrm{d}}^\star(t), \ddot{\mathbf{q}}_{\mathrm{d}}^\star(t), \boldsymbol{\tau}_{\mathrm{d}}^\star(t))$ is simply recovered from $\dot{\mathbf{q}}_{\mathrm{d}}^\star = \mathbf{h}'(s)c^\star(s)$, $\ddot{\mathbf{q}}_{\mathrm{d}}^\star = \mathbf{h}''(s)b^\star(s) + \mathbf{h}'(s)a^\star(s)$. Likewise, the corresponding time $t(s)$ can be recovered from $d(s)$ as follows: initialize $t(s_1) = 0$, then for $k = 2, \ldots, N$, compute:

$$
t(s_k) = t(s_{k-1}) + \frac{1}{2}\Delta s_{k-1}(d_{k-1} + d_k). \qquad (11)
$$

IV. APPLICATION TO A 6-AXIS MANIPULATOR

The time-optimal 4-tuple $(\mathbf{q}_{\mathrm{d}}^\star(t), \dot{\mathbf{q}}_{\mathrm{d}}^\star(t), \ddot{\mathbf{q}}_{\mathrm{d}}^\star(t), \boldsymbol{\tau}_{\mathrm{d}}^\star(t))$ is generated for a 6-axis industrial manipulator, namely, FANUC M-16iB. A non-trivial path with non-constant orientation of the end effector is designed. The Cartesian-space path is shown in Fig. 1(a), where the initial and final points are marked with '*' and 'o', respectively. Both the position and orientation (Euler angles) are interpolated using cubic splines to satisfy the differentiability requirement on $\mathbf{h}(s)$.



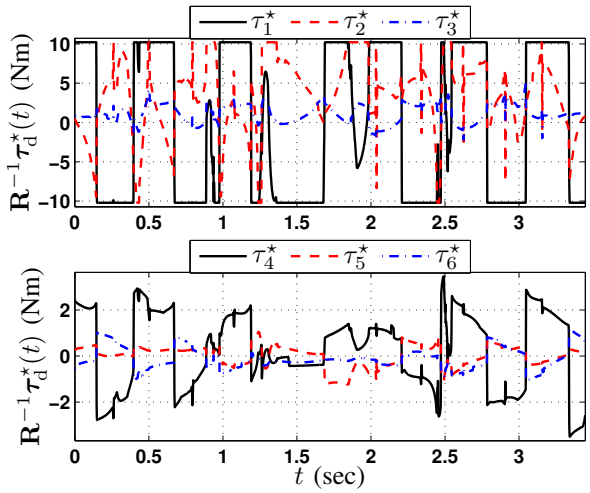
(a) Cartesian-space path

(b) Joint-space path

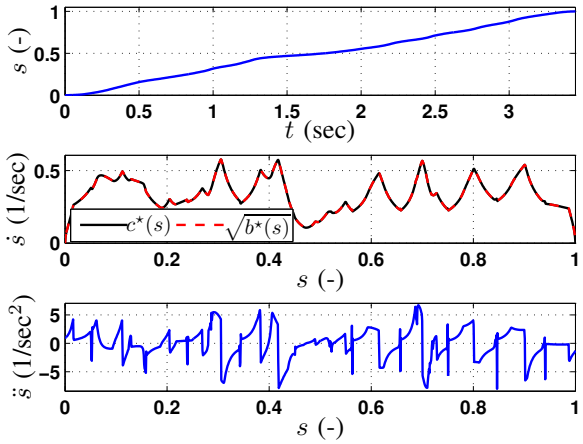
Fig. 1. Non-trivial path used to test the algorithm

A total of $N = 1200$ points are used to discretize the parameter s. The Robotics Toolbox for MATLAB® is used to carry out kinematics and dynamics computations [11]. In Fig. 1(b), the corresponding joint-space path $\mathbf{h}(s)$ together with $\mathbf{h}'(s)$ and $\mathbf{h}''(s)$ are presented.

The initial and final speeds are enforced to be zero, i.e., $\dot{s}_0 = \dot{s}_f = 0$. The torque constraints $\overline{\boldsymbol{\tau}}$ are computed as follows: for the jth motor, $\overline{\tau}_j = (1/4) r_j k_j i_j^{\max}$, where r_j is the gear ratio, k_j is the torque-current constant, and $i_j^{\max}$ the maximum current. Therefore, $\overline{\boldsymbol{\tau}} = (1782.4 \;\; 1789.7 \;\; 1647.2 \;\; 97.2 \;\; 108.5 \;\; 79.1)^\top$ Nm, which represents the maximum torques at the link-side, i.e., after the reducers. The corresponding maximum torques at the motor-side, i.e., before the reducers, are $\overline{\mathbf{u}} = \mathbf{R}^{-1}\overline{\boldsymbol{\tau}} = (10.21 \;\; 10.21 \;\; 8.60 \;\; 4.30 \;\; 1.58 \;\; 1.58)^\top$ Nm, where $\mathbf{R}$ is a



(a) Time-optimal torques in motor-side, $\mathbf{R}^{-1}\boldsymbol{\tau}_\mathrm{d}^\star(t)$



(b) Pseudo-speed and pseudo-acceleration

Fig. 2. Numerical time-optimal solutions

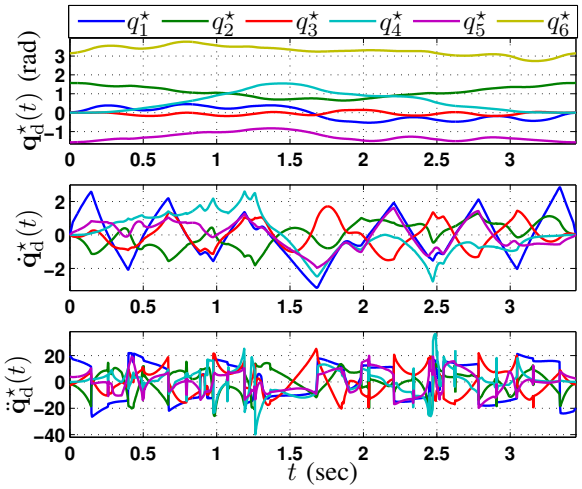


Fig. 3. Time-optimal trajectory $(\mathbf{q}_\mathrm{d}^\star(t), \dot{\mathbf{q}}_\mathrm{d}^\star(t), \ddot{\mathbf{q}}_\mathrm{d}^\star(t))$

diagonal matrix with the gear ratios.

The time-optimal solutions are presented in Fig. 2, with a total traversal time $t_f = 3.447$ seconds. Notice that the optimal torques in Fig. 2(a) are reported in the motor-side, namely, $\mathbf{R}^{-1}\boldsymbol{\tau}_\mathrm{d}^\star$. These torques feature bang-bang behavior, i.e., there is always one actuator that saturates. Likewise, in Fig. 2(b), both $c^\star(s)$ and $\sqrt{b^\star(s)}$ are plotted together, from which it is observed that $c^\star(s) = \sqrt{b^\star(s)}\ \forall s \in [0, 1]$. This confirms that indeed, the proposed convex relaxation (5) solves exactly the original non-convex problem (4).

The time-optimal trajectory $(\mathbf{q}_\mathrm{d}^\star(t), \dot{\mathbf{q}}_\mathrm{d}^\star(t), \ddot{\mathbf{q}}_\mathrm{d}^\star(t))$ is presented in Fig. 3. The dynamic feasibility of the 4-tuple $(\mathbf{q}_\mathrm{d}^\star(t), \dot{\mathbf{q}}_\mathrm{d}^\star(t), \ddot{\mathbf{q}}_\mathrm{d}^\star(t), \boldsymbol{\tau}_\mathrm{d}^\star(t))$ is verified by utilizing $(\mathbf{q}_\mathrm{d}^\star(t), \dot{\mathbf{q}}_\mathrm{d}^\star(t), \ddot{\mathbf{q}}_\mathrm{d}^\star(t))$ to compute the torques $\boldsymbol{\tau}(t)^\mathrm{feas}$ using the dynamic model (1); the resulting $\boldsymbol{\tau}(t)^\mathrm{feas}$ is then compared against $\boldsymbol{\tau}_\mathrm{d}^\star(t)$. It turns out, in all cases, $\boldsymbol{\tau}_\mathrm{d}^\star(t) - \boldsymbol{\tau}(t)^\mathrm{feas} = \mathbf{0}\ \forall t \in [0, t_f]$, which means that our algorithm generates optimal 4-tuples that are exactly dynamically feasible with respect to the dynamic model (1).

A. Simulation of Time-optimal Solution

In order to study the effects of implementing the time-optimal 4-tuple $(\mathbf{q}_\mathrm{d}^\star(t), \dot{\mathbf{q}}_\mathrm{d}^\star(t), \ddot{\mathbf{q}}_\mathrm{d}^\star(t), \boldsymbol{\tau}_\mathrm{d}^\star(t))$, simulations are carried out. A Robot Simulator in MATLAB® that uses Simulink® and SimMechanics™ has been developed in our research group. This Simulator incorporates real robot dynamic effects that are not considered in the dynamic model (1), for instance, joint flexibility due to indirect drives. The link-side and motor-side positions are denoted by $\mathbf{q}$ and $\boldsymbol{\theta}$, respectively. The control law in the motor-side:

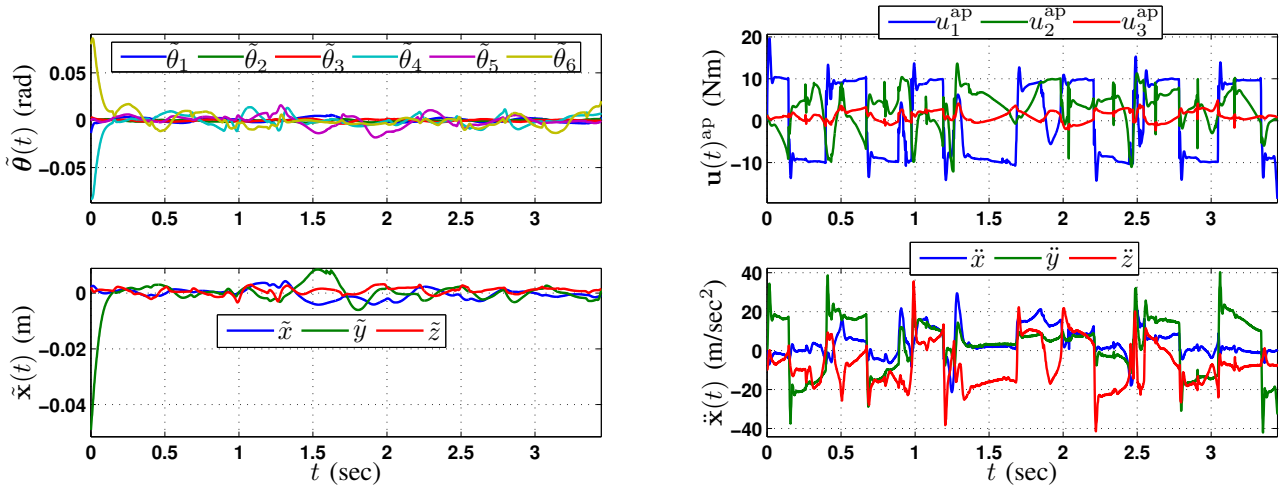
$$\mathbf{u} = \mathbf{u}(t)^\mathrm{ff} + \mathbf{K}_\mathrm{P}\tilde{\boldsymbol{\theta}} + \mathbf{K}_\mathrm{V}\dot{\tilde{\boldsymbol{\theta}}} + \mathbf{K}_\mathrm{I}\int_0^t \tilde{\boldsymbol{\theta}}(v)\ \mathrm{d}v, \qquad (12)$$

where $\mathbf{u}(t)^\mathrm{ff}$ is the motor-side feedforward torque, i.e., $\mathbf{u}(t)^\mathrm{ff} = \mathbf{R}^{-1}\boldsymbol{\tau}_\mathrm{d}^\star(t)$, and $\tilde{\boldsymbol{\theta}}(t) := \boldsymbol{\theta}_\mathrm{d}(t) - \boldsymbol{\theta}(t)$ is the motor-side tracking error. The feedback gains $\mathbf{K}_\mathrm{P}$, $\mathbf{K}_\mathrm{V}$, and $\mathbf{K}_\mathrm{I}$ are constant for both simulations and experiments. The resulting tracking errors $\tilde{\boldsymbol{\theta}}$ and $\tilde{\mathbf{x}}$, are shown in Fig. 4(a).

The motor-side applied torques, $\mathbf{u}(t)^\mathrm{ap}$, are presented in Fig. 4(b); for readability only first three joints a presented. Notice that the applied torques $\mathbf{u}(t)^\mathrm{ap}$, which are computed with control law (12), feature large peaks at those instants when the feedforward torques $\mathbf{u}(t)^\mathrm{ff}$ change suddenly. This effect is also evident from the accelerometer readings. Due to the non-zero initial and final accelerations that time-optimal solutions feature, the initial and final tracking errors are comparatively large. These effects are seldom pointed out in any of the listed references that deal with time-optimal trajectories and controls.

V. IMPOSING ACCELERATION CONSTRAINTS AND PENALIZING TOTAL JERK

It is clear from Fig. 4 that time optimality alone leads to degradation of the system performance, nonetheless, time-



(a) Motor-side and Cartesian tracking errors

(b) Applied torques and accelerometer readings

Fig. 4. Simulation results for the time-optimal solution

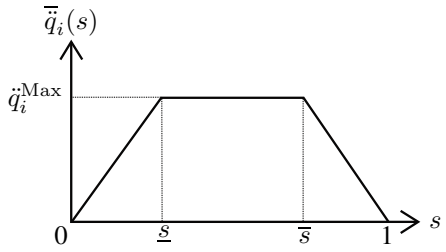


Fig. 5. Profile of joint-space acceleration constraints

optimal trajectories and controls are important for increase of robot productivity. Therefore, it is still desirable to consider problem (10), and incorporate acceleration constraints that guarantee smooth acceleration growth (resp. decay) from zero (resp. to zero) at the the initial (resp. final) point of the trajectory. Additionally, a term that penalizes a measure of total jerk will prove to be useful.

A. Acceleration Constraints

Consider imposing joint-space acceleration constraints with the profile shown in Fig. 5, for each $\overline{\ddot{q}}_i(s)$, $i = 1, \ldots, n$, where $\ddot{q}_i^{\mathrm{Max}}$ represents the maximum acceleration at intermediate points. In vector form, $-\overline{\ddot{\mathbf{q}}}(s) \leq \ddot{\mathbf{q}}_{\mathrm{d}}(s) \leq \overline{\ddot{\mathbf{q}}}(s)$, which is readily discretized as:

$$-\overline{\ddot{\mathbf{q}}}(s_k) \leq \mathbf{h}''(s_k)b_k + \mathbf{h}'(s_k)a_k \leq \overline{\ddot{\mathbf{q}}}(s_k) \tag{13}$$

for $k = 1, \ldots, N$. Inequality constraints (13) are therefore incorporated to problem (10).

B. Penalizing a Measure of Total Jerk

We are interested in $\dddot{\mathbf{q}}$, but the forthcoming derivation follows similar lines for $\dot{\boldsymbol{\tau}}$ (see [6]).

$$\begin{aligned}\lambda \int_0^{t_f} \|\dddot{\mathbf{q}}\|_1 \, \mathrm{d}t &= \lambda \sum_{i=1}^{n} \int_0^{t_f} |\dddot{q}_i| \, \mathrm{d}t \\ &= \lambda \sum_{i=1}^{n} \int_0^{t_f} \left| \frac{\mathrm{d}\ddot{q}_i}{\mathrm{d}t} \right| \mathrm{d}t \\ &= \lambda \sum_{i=1}^{n} \int_0^{1} \left| \frac{\mathrm{d}\ddot{q}_i}{\mathrm{d}s} \right| \mathrm{d}s \\ &\approx \lambda \sum_{i=1}^{n} \sum_{j=1}^{N-1} |\ddot{q}_i(s_{j+1}) - \ddot{q}_i(s_j)| \\ &\propto \lambda \sum_{i=1}^{n} \sum_{j=1}^{N-1} \frac{|\ddot{q}_i(s_{j+1}) - \ddot{q}_i(s_j)|}{\ddot{q}_i^{\mathrm{Max}}}. \end{aligned} \tag{14}$$

By introducing the slack variables e_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, N-1$, such that $|\ddot{q}_i(s_{j+1}) - \ddot{q}_i(s_j)| \leq \ddot{q}_i^{\mathrm{Max}} e_{ij}$, (14) can be replaced with the linear objective function

$$J_{\mathrm{jerk}} = \lambda \sum_{i=1}^{n} \sum_{j=1}^{N-1} e_{ij}, \tag{15}$$

which is incorporated into the objective of problem (10) to trade off the traversal time. The constraints $|\ddot{q}_i(s_{j+1}) - \ddot{q}_i(s_j)| \leq \ddot{q}_i^{\mathrm{Max}} e_{ij}$, are expressed compactly by defining $\mathbf{e}_j := (e_{1j} \; e_{2j} \; \cdots \; e_{nj})^\top \in \mathbb{R}^n$. Therefore,

$$\begin{aligned}-\mathbf{e}_j * \ddot{\mathbf{q}}^{\mathrm{Max}} \leq \mathbf{h}''(s_{j+1})b_{j+1} + \mathbf{h}'(s_{j+1})a_{j+1} \\ - \mathbf{h}''(s_j)b_j - \mathbf{h}'(s_j)a_j \leq \mathbf{e}_j * \ddot{\mathbf{q}}^{\mathrm{Max}},\end{aligned}$$

for $j = 1, \ldots, N-1$, where $\mathbf{e}_j * \ddot{\mathbf{q}}^{\mathrm{Max}}$ means element-wise multiplication. These are the final constraints that need to be incorporated to problem (10).

C. Experimental Results

We generate optimal 4-tuples $(\mathbf{q}_{\mathrm{d}}^\star(t), \dot{\mathbf{q}}_{\mathrm{d}}^\star(t), \ddot{\mathbf{q}}_{\mathrm{d}}^\star(t), \boldsymbol{\tau}_{\mathrm{d}}^\star(t))$ for the 6-axis manipulator FANUC M-16iB. Here we present the results for $\lambda = 0.02$, and for the acceleration constraint parameters $\underline{s} = 0.02$, $\overline{s} = 0.98$, and $\ddot{\mathbf{q}}^{\mathrm{Max}} = (60 \; 60 \; 60 \; 30 \; 30 \; 30)^\top$ rad/sec^2. Some representative variables that the algorithm generates are shown in Fig. 6. Notice that exact zero acceleration is indeed enforced at the beginning/end of the trajectory, with a smooth growth/decay. Also notice that at the intermediate points sudden changes

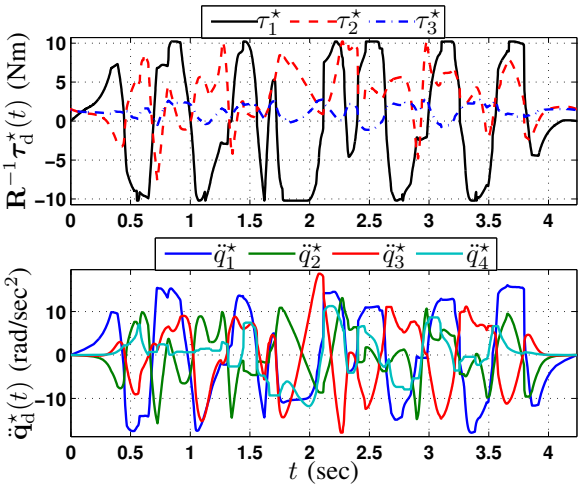


Fig. 6. Representative variables from algorithm results for $\lambda = 0.02$

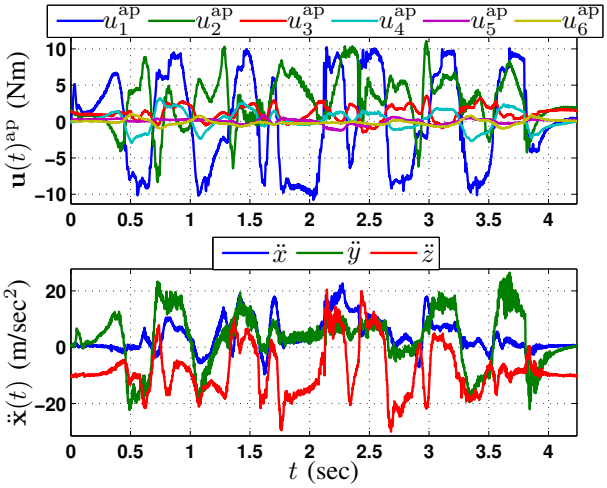


Fig. 7. Experimental applied torques and accelerometer readings

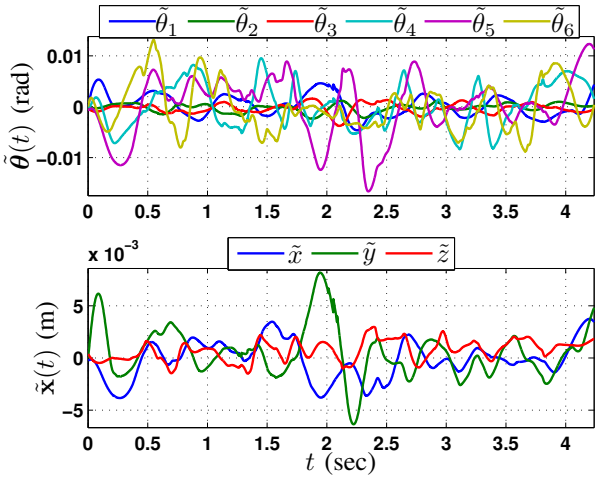


Fig. 8. Experimental motor-side and Cartesian-space tracking errors

are eliminated. These benefits come at the cost of a modest increase in the traversal time, i.e., $t_f = 4.238$ seconds, which means that this solution is slower than the purely time-optimal by only 0.791 seconds. Nonetheless, the benefits in terms of performance become a crucial factor to justify our development.

Experimental results are presented in Figs. 7 and 8. Note that the applied torques $\mathbf{u}(t)^{\mathrm{ap}}$ are close to the feedforward torques $\mathbf{u}(t)^{\mathrm{ff}}$ in Fig. 6, and therefore $\mathbf{u}(t)^{\mathrm{ap}}$ do not exceed the torque limits. Also, the accelerometer readings of Fig. 7 should be compared against the ones in Fig. 4(b). The corresponding motor-side and Cartesian-space tracking errors are presented in Fig. 8, all of which are comparatively better than the ones in Fig. 4(a). Our methodology therefore generates the fastest solutions that can actually be implemented in the real system, without degrading its performance.

VI. CONCLUSIONS

An algorithm that generates optimal trajectories and controls was studied. Initially, pure time-optimality was considered. Then, acceleration constraints and penalization of a measure of total jerk were incorporated, both of which proved useful from real experiments on a 6-axis industrial manipulator. In all cases, the resulting optimal trajectories and controls are always dynamically feasible with respect to the complete dynamic model (1), which brings a modest theoretical and practical extension to existing algorithms.

REFERENCES

[1] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[2] 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.

[3] 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, 1985.

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

[5] J. T. Betts, *Practical methods for optimal control using nonlinear programming*, ser. Advances in Design and Control. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 2001, vol. 3.

[6] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: a convex optimization approach," *IEEE Transactions on Automatic Control*, 2008.

[7] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 1st ed., ser. Advanced Textbooks in Control and Signal Processing. Springer-Verlag, 2009.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. Cambridge University Press, 2004, [Online]. Available: http://www.stanford.edu/ boyd/cvxbook/.

[9] L. I, L. Vandenberghe, H. Lebret, and S. Boyd, "Applications of second-order cone programming," *Linear Algebra and its Applications*, pp. 193–228, 1998, [Online]. Available: http://www.stanford.edu/ boyd/papers/socp.html.

[10] O. V. Stryk, "Numerical solution of optimal control problems by direct collocation," 1993. [Online]. Available: http://citeseer.ist.psu.edu/69756.html; http://www-m2.mathematik.tu-muenchen.de/ stryk/paper/1991-dircol.ps.gz

[11] P. Corke, "A robotics toolbox for MATLAB," *IEEE Robotics and Automation Magazine*, vol. 3, no. 1, pp. 24–32, March 1996.