Citation	<p>Debrouwere, F., Van Loock, W., Pipeleers, G., Tran Dinh, Q., Diehl, M., De Schutter, J., Swevers, J. (2013) Time-Optimal Path Following for Robots with Trajectory Jerk Constraints using Sequential Convex Programming 2013 IEEE International Conference on Robotics and Automation (ICRA). IEEE International Conference on Robotics and Automation. Karlsruhe, Germany, 6-10 May 2013 (pp. 1908-1913)</p>
Archived version	Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher
Published version	http://dx.doi.org/10.1109/ICRA.2013.6630831
Publisher homepage	*
Author contact	<p>E-mail: frederik.debrouwere@kuleuven.be Phone number: +32 16 32 92 64</p>
IR	https://lirias.kuleuven.be/handle/123456789/358308

(article begins on next page)

Time-Optimal Path Following for Robots with Trajectory Jerk Constraints using Sequential Convex Programming

Frederik Debrouwere*, Wannes Van Loock*, Goele Pipeleers*, Quoc Tran Dinh[†], Moritz Diehl[†],
Joris De Schutter*, Jan Swevers*

Abstract—Time-optimal path following considers the problem of moving along a predetermined geometric path in minimum time. In the case of a robotic manipulator a convex reformulation of this optimal control problem has been derived previously [1]. However, the bang-bang nature of the time-optimal trajectories results in near-infinite jerks in joint space and operational (Cartesian) space. For systems with unmodeled flexibilities, this usually results in excitation of the resonant frequencies, hence in unwanted vibrations and acceleration peaks, contributing to a tracking error. These vibrations can be reduced by imposing jerk constraints on the trajectory [2]. However, these jerk constraints destroy the convexity of the time-optimal control problem. The present paper proposes an efficient sequential convex programming (SCP) approach to solve the corresponding non-convex optimal control problem by writing the non-convex jerk constraints as a difference of convex (DC) functions. We illustrate the developed approach by means of experiments with a seven d.o.f. robot. Furthermore, numerical simulations illustrate the fast convergence of the proposed method in only a few SCP iterations, confirming the efficiency and practicality of the proposed framework.

I. INTRODUCTION

Path following deals with the problem of following a geometric path without any preassigned timing information. Many industrial robot tasks, such as welding, glueing, laser cutting and milling can be cast as path following problems. In addition, path following is often considered to be the low level stage in a decoupled motion planning approach [3], [4]. First, a high level planner determines a geometric path ignoring the system dynamics but taking into account geometric path constraints. Second, an optimal trajectory along the geometric path is determined that accounts for the system dynamics and limitations. Since the dynamics along a geometric path can be described by a scalar path coordinate s and its time derivatives [3], [4], the decoupled approach simplifies the motion planning problem to great extent. Recently it was shown that the path following problem for a robotic manipulator with simplified constraints can be cast as a *convex* optimization problem [1]. This guarantees the efficient computation of globally optimal solutions.

However, the bang-bang nature of the time-optimal trajectories results in near-infinite jerks in joint space and operational (Cartesian) space. For systems with flexibilities, this usually results in excitation of the resonant frequencies, hence in unwanted vibrations and acceleration peaks. These

unwanted vibrations contribute to the tracking error of the control system that executes the time-optimal trajectory. Moreover, in most applications the acceleration peaks and vibrations are unwanted, e.g. when moving non-fixed bodies or when moving fragile objects [5]. These vibrations can either be reduced by including the flexibilities in the system model, which results in a strong non-convex optimization problem. Furthermore, the flexibilities are hard to identify. A more indirect, empirical, approach to reduce the vibrations consists of imposing jerk constraints [2]. These jerk constraints also have a positive effect on the tracking error, since the resulting trajectory will be smoother. However, these jerk constraints introduce a mild non-convexity into the time-optimal control problem. To account for such constraints we propose an extension of the convex framework of [1] using sequential convex programming (SCP), since the jerk constraints can easily be written as a difference of two convex (DC) functions [6], [7]. In SCP the concave parts of the constraints are sequentially linearised while preserving the convex parts [8]–[10]. Fast convergence of this method in only three iterations is observed for the examples discussed in this paper, making it a highly efficient framework.

The paper is organized as follows. Section II reviews the convex reformulation of the time-optimal path following problem of [1] and illustrates the convexity of joint and Cartesian velocity and acceleration constraints. Section III describes the jerk constraints in joint space and operational space and formulates them in the same non-convex form. Section IV describes a DC decomposition of the non-convex constraints and reviews a sequential convex programming approach [8]–[10] to efficiently solve the non-convex optimal path following problem. To illustrate the practical use of the proposed SCP approach, a numerical examples and experiments are given in Section V.

Throughout the paper we will use the following shorthand notations for the derivatives of a function $f(s(t))$: $\dot{f} = \frac{df}{dt}$, $\ddot{f} = \frac{d^2f}{dt^2}$, $\dddot{f} = \frac{d^3f}{dt^3}$, $f' = \frac{\partial f}{\partial s}$, $f'' = \frac{\partial^2 f}{\partial s^2}$ and $f''' = \frac{\partial^3 f}{\partial s^3}$ where t indicates time and s the path coordinate. Furthermore, we indicate scalars with a lower-case letter, e.g. n , vectors with a bold lower-case letter, e.g. $\mathbf{q}$, and matrices with an upper-case letter, e.g. M . $\mathbf{q}_i$ denotes the i -th element of $\mathbf{q}$. To indicate an all-ones vector of size n we use $\mathbf{1}_n \in \mathbb{R}^{n \times 1}$. We use the following definition for the velocity twist, capturing the translational and rotational

*F. Debrouwere, W. Van Loock, G. Pipeleers, J. De Schutter and J. Swevers are with the Division PMA, Department of Mechanical Engineering, KU Leuven, Belgium, e-mail: <firstname>.<lastname>@mech.kuleuven.be

[†]Q. Tran Dinh and M. Diehl are with the Division SCD, Department of Electrical Engineering, KU Leuven, Belgium

velocity of a rigid body:

$${}^v_t_w^o = \left(({}^v_v^o)^T, ({}_f\omega_w^o)^T \right)^T.$$

Here ${}^v_v^o$ and ${}_f\omega_w^o$ represent the translational and rotational velocity of the body, represented by a frame $\{o\}$ with respect to the world, represented by a frame $\{w\}$. Index v is the reference point on the body used to express the translational velocity, while $\{f\}$ represents the frame in which the coordinates of the v and ω are expressed. The relation between the velocity twist and the joint velocity $\dot{q}$ is defined by

$${}^v_t_w^o = {}^vJ_w^o(q) \dot{q}, \quad (1)$$

where ${}^vJ_w^o(q)$ is the geometric robot Jacobian [11].

II. PROBLEM FORMULATION

Consider a robotic manipulator with n degrees of freedom and joint angles $q \in \mathbb{R}^n$. The equations of motion are given by

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q), \quad (2)$$

where $\tau \in \mathbb{R}^n$ are the joint torques, $M \in \mathbb{R}^{n \times n}$ is the mass matrix, $C \in \mathbb{R}^{n \times n}$ is a matrix, linear in $\dot{q}$, accounting for Coriolis and centrifugal effects and, g is a vector accounting for gravity and other position dependent torques.

Consider a prescribed geometric path $q(s)$ as a function of a scalar path coordinate s , given in joint space coordinates. The time dependence of the path is determined through $s(t)$. Without loss of generality it is assumed that the trajectory starts at $t = 0$, ends at $t = T$ and, $0 = s(0) \leq s(t) \leq s(T) = 1$. It is furthermore assumed that we always move forward along the path, i.e. $\dot{s}(t) \geq 0, \forall t \in [0, T]$.

Using the chain-rule we rewrite joint velocities and accelerations as

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

where $q' = dq(s)/ds$ and, $q'' = d^2q(s)/ds^2$. Substitution of the above equations in (2) projects the equations of motion onto the path [1]:

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

By introducing the variable

$$b(s) = \dot{s}^2, \quad (5)$$

it was shown in [1] that the time-optimal path following problem is transformed into a *convex* optimization problem. Indeed, since $b(s) \geq 0$, the total motion time

$$T = \int_0^T 1dt = \int_0^1 \frac{1}{\dot{s}} ds = \int_0^1 \frac{1}{\sqrt{b(s)}} ds,$$

is a convex function. Furthermore, since $b'(s) = 2\dot{s}$, the torque constraints are linear in $b(s)$ and $b'(s)$. The time-

optimal path following problem is then reformulated as

$$\begin{aligned} & \underset{b(\cdot)}{\text{minimize}} && \int_0^1 \frac{1}{\sqrt{b(s)}} ds \\ & \text{subject to} && b(0) = \dot{s}_0^2, b(1) = \dot{s}_T^2 \\ & && b(s) \geq 0 \\ & && \tau_- \leq m(s) \frac{b'(s)}{2} + c(s)b(s) + g(s) \leq \tau_+ \\ & && \text{for } s \in [0, 1]. \end{aligned} \quad (6)$$

In [1], [5] many constraints are described that maintain the convexity of the problem, such as velocity constraints and acceleration constraints.

A. Velocity constraints

Constraints on the joint velocity $\dot{q} = q'(s)\dot{s}$ and Cartesian velocity ${}^v_t_w^o = {}^vJ_w^o(q)q'(s)\dot{s}$ can be written as a linear function of the optimization variables (5) by carefully squaring these constraints. Suppose the joint velocity upper bound $\dot{q}_{+,i}(s) > 0$ and the joint velocity lower bound $\dot{q}_{-,i}(s) < 0$, which is typically the case, we can square $q'_i(s)\sqrt{b(s)} \leq \dot{q}_{+,i}(s)$ if $q'_i(s) > 0$, yielding $q'_i(s)^2 b(s) \leq \dot{q}_{+,i}(s)^2$, and square $-\dot{q}_{-,i}(s) \geq -q'_i(s)\sqrt{b(s)}$ if $q'_i(s) < 0$, yielding $(-\dot{q}_{-,i}(s))^2 \geq (-q'_i(s))^2 b(s)$. The same holds for the Cartesian velocity.

B. Acceleration constraints

The joint acceleration can also be written as a linear function of the optimization variables. By substituting (5) and $2\ddot{s} = b'(s)$ into (3) we get

$$\ddot{q}(s) = q'(s) \frac{1}{2} b'(s) + q''(s)b(s)$$

Furthermore, by applying the chain rule to (1) and $\dot{J}$:

$$\begin{aligned} {}^v_t_w^o(s) &= {}^vJ_w^o(s)\ddot{q}(s) + {}^vJ_w^o(s)\dot{q}(s) \text{ and} \\ {}^vJ_w^o(s) &= ({}^vJ_w^o)'(s)\dot{s}, \end{aligned} \quad (7)$$

and by using (3), the Cartesian acceleration twist of a body, represented by a frame $\{o\}$ with respect to the world, represented by a frame $\{w\}$, can be written as a linear function of the optimization variables [5]

$${}^v_t_w^o = {}^vJ_{a_w}^o(s) \frac{1}{2} b'(s) + {}^vJ_{b_w}^o(s) b(s), \quad (8)$$

with

$$\begin{aligned} {}^vJ_{a_w}^o(s) &= {}^vJ_w^o(q)q', \\ {}^vJ_{b_w}^o(s) &= {}^vJ_w^o(q)q'' + ({}^vJ_w^o)'(q)q'. \end{aligned}$$

Hence, a constraints on the joint and Cartesian velocity and acceleration can be added to the optimization problem, preserving its convexity. From here on we omit the indices of $\dot{t}$ and J for notational simplicity.

III. ADDING JERK CONSTRAINTS

We now introduce the joint and Cartesian jerk constraints into the optimization problem. We show that both constraints can be written into the form

$$\phi(s, b(s)) + \psi(s)b(s)\kappa \leq \rho(s), \quad (9)$$

where $\phi(s, \cdot)$ is a linear vector functions of the optimization variables, $\psi(s)$ and κ are vectors independent of the optimization variables and $b(s)\kappa = (\dots, b(s)\kappa_i, \dots)^T$. The convexity of (9) depends on the sign of $\psi_i(s)$ and the value of κ_i . If non-convex, the constraint can easily be decomposed as a difference of two convex functions (DC constraint). Section IV illustrates this decomposition and reviews a sequential convex programming approach to efficiently solve optimization problem (6)-(9).

Since the resulting time-optimal path following problem has an infinite number of optimization variables and an infinite number of constraints, it is discretized by adopting the direct transcription method from [1].

A. Joint jerk constraints

We can write the joint jerk by applying the chain rule to (3)

$$\ddot{q} = q''' \dot{s}^3 + 3q'' \ddot{s} \dot{s} + q' \ddot{s}. \quad (10)$$

Furthermore, applying the chain rule to (5) results in:

$$\dot{s} = \sqrt{b(s)}, \ddot{s} = \frac{1}{2}b'(s), \ddot{s} = \frac{1}{2}b''(s)\sqrt{b(s)}. \quad (11)$$

By factoring out $\sqrt{b(s)} \geq 0$, we can write the joint jerk constraint $\ddot{q}_-(s) \leq \ddot{q}(s) \leq \ddot{q}_+(s)$ as (9) with $\kappa = -\frac{1}{2}\mathbf{1}_{2n}$, $\rho(s) = \mathbf{0}$ and

$$\phi_q(s, b(s)) = q'''(s)b(s) + \frac{3}{2}q''(s)b'(s) + \frac{1}{2}q'(s)b''(s),$$

$$\phi(s, b(s)) = \begin{pmatrix} -\phi_q(s, b(s)) \\ \phi_q(s, b(s)) \end{pmatrix} \text{ and } \psi(s) = \begin{pmatrix} \ddot{q}_-(s) \\ -\ddot{q}_+(s) \end{pmatrix}.$$

B. Cartesian (operational space) jerk constraints

By applying the chain rule to (7) we can write the Cartesian jerk as

$$\ddot{t}(s) = \ddot{J}(s)\dot{q}(s) + 2\dot{J}(s)\ddot{q}(s) + J(s)\ddot{\ddot{q}}(s)$$

with $\ddot{J}(s) = J''(s)\dot{s}^2 + J'(s)\ddot{s}$. By using (3), (10) and (11) and by factoring out $\sqrt{b(s)} \geq 0$ we can write the Cartesian jerk constraint $\ddot{t}_-(s) \leq \ddot{t}(s) \leq \ddot{t}_+(s)$ in the form of (9) with $\kappa = -\frac{1}{2}\mathbf{1}_{2n}$, $\rho(s) = \mathbf{0}$ and

$$\phi_t(s, b(s)) = (J''(s)q'(s) + J'(s)q''(s) + J(s)q'''(s))b(s) + \frac{3}{2}(J'(s)q'(s) + J(s)q''(s))b'(s) + \frac{1}{2}J(s)q'(s)b''(s),$$

$$\phi(s, b(s)) = \begin{pmatrix} -\phi_t(s, b(s)) \\ \phi_t(s, b(s)) \end{pmatrix} \text{ and } \psi(s) = \begin{pmatrix} \ddot{t}_-(s) \\ -\ddot{t}_+(s) \end{pmatrix}.$$

IV. SCP ALGORITHM AND DC DECOMPOSITION

A. Sequential convex programming algorithm

Consider the DC optimization problem:

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} && f(x) \\ & \text{subject to} && u_i(x) - v_i(x) \leq 0, \quad i = 1, \dots, l \\ & && x \in \Omega, \end{aligned} \quad (12)$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex, $\Omega \subseteq \mathbb{R}^n$ is a nonempty, closed convex set, and u_i and v_i are convex functions. The feasible set is denoted by

$$\mathcal{D} = \{x \in \Omega \mid u_i(x) - v_i(x) \leq 0, i = 1, \dots, l\}$$

and its relative interior by

$$\text{ri}(\mathcal{D}) = \{x \in \text{ri}(\Omega) \mid u_i(x) - v_i(x) < 0, i = 1, \dots, l\}.$$

The main idea of the algorithm is to iteratively linearise the non-convex part of the DC inequality constraint, transforming the problem into a convex optimization problem, which can be solved efficiently.

Suppose that $x^k \in \Omega$ is a given point, then the linearized problem of (12) around x^k is

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} && f(x) + \frac{\beta}{2}\|x - x^k\|_2^2 \\ & \text{subject to} && u_i(x) - v_i(x^k) - \nabla_x v_i(x^k)(x - x^k) \leq 0, \\ & && i = 1, \dots, l, \\ & && x \in \Omega, \end{aligned} \quad (13)$$

where a regularization term with parameter β is added to the objective function. The following algorithm for solving (12) is proposed in [9].

Sequential convex programming algorithm

Initialization: Choose a $\beta > 0$ and find an initial point $x^0 \in \text{ri}(\mathcal{D})$. Set $k = 0$.

Iteration k : For $k = 0, 1, \dots$ do

- 1) Solve the convex problem (13) to obtain a solution x^{k+1} .
 - 2) If $\|x^{k+1} - x^k\| \leq \varepsilon$ for a given tolerance $\varepsilon > 0$ then terminate. Otherwise, set $k = k + 1$ and go to step 1.
-

Note that when the initial point $x^0 \in \text{ri}(\mathcal{D})$, then the algorithm generates a sequence of points x^k which also belongs to $\mathcal{D}$. Geometrically, the algorithm can be seen as an inner approximation method. Under fairly mild assumptions, which in practice come down to the problem being well posed, it can be shown that the SCP algorithm converges to a stationary point of (12) [9].

B. DC decomposition

To apply the algorithm to problem (6)-(9), we need to determine a DC decomposition of (9) by examining its

Fig. 1. Robot set-up with accelerometer.

convexity, which depends on the values of $\psi_i(s)$ and κ_i . Since $b(s) \geq 0$, (9) is nonconvex whenever

$$\begin{aligned} \psi_i(s) > 0 \text{ and } 0 < \kappa_i < 1, \text{ or} \\ \psi_i(s) < 0 \text{ and } \kappa_i > 1 \cup \kappa_i < 0. \end{aligned}$$

In the case of joint and Cartesian jerk $\kappa_i = -\frac{1}{2}$. Typically $\ddot{\mathbf{q}}_-(s) \leq 0$, $\dot{\mathbf{t}}_-(s) \leq 0$, $\ddot{\mathbf{q}}_+(s) \geq 0$ and $\dot{\mathbf{t}}_+(s) \geq 0$, hence $\psi_i(s) \leq 0$ and (9) is non-convex. An obvious DC decomposition as in (12) is

$$\mathbf{u}_i(b(s)) = \phi_i(s, b(s)) - \rho_i(s) \text{ and } \mathbf{v}_i(b(s)) = -\psi_i(s)b(s)^{\kappa_i}.$$

V. NUMERICAL EXAMPLES AND EXPERIMENTS

In this section we illustrate the proposed framework with numerical simulations and experiments. For the experiments, an off-line calculated motion profile is applied to the robot. We show how Cartesian jerk constraints result in lower acceleration peaks and less tracking error with only a minimal loss in time-optimality. For the experiments we use a seven degree of freedom KUKA LWR executing a linear trajectory along the x-axis of the base frame. The orientation of the end-effector remains equal to the orientation of the base frame during the whole motion. The robot is controlled using OROCOS [12] and for the measurements of the end effector acceleration we use a HBM B12/500 accelerometer. The SCP algorithm is implemented directly and very easily using the free high-level optimization modeling tool YALMIP [13] on an Intel Core i3 CPU running at a 2.53GHz Windows machine.

We consider transporting a non-fixed rigid body, placed on a plate held by the end effector of the robot. Figure 1 shows a picture of the set-up. To prevent the object from tipping over or sliding away, the Cartesian acceleration has to be constrained. For a rectangular object with width w and height h , the object will tip over if $|\dot{\mathbf{t}}_x| > \frac{w}{h}g$ and slip if $|\dot{\mathbf{t}}_x| > \mu g$, with g the gravitational acceleration and μ the static friction coefficient. Here we have ensured the friction to be big enough so that the tipping constraint

Fig. 2. Desired versus measured acceleration for a linear Cartesian trajectory without jerk constraint.

Fig. 3. Trade-off between time-optimality and normalized measure acceleration peak for several jerk constraints.

is dominant. The geometry of the object implies that the Cartesian acceleration has to be constrained to 5 m/s^2 . We show that in the absence of a jerk constraint, the acceleration peaks due to unmodeled flexibilities result in the object tipping over. Including jerk constraints result in a motion with some loss in time-optimality but with less acceleration peaks, which leads to the object not tipping over.

First, we apply the optimal trajectory with a Cartesian acceleration constraint and without Cartesian jerk constraint. In this case, the robot motion (with $T^* = 1.19\text{s}$) results in high acceleration peaks due to unmodeled flexibilities. The desired and measured end effector acceleration along the x-axis are given in Figure 2. Due to the acceleration peaks (up to 9.2 m/s^2) the object will tip over.

To reduce the acceleration peaks and vibrations we include a jerk constraint into the optimization problem. The value of this constraint is chosen empirically to reduce the effect of the unmodeled flexibilities on the vibrations and acceleration peaks, such that the object will not tip over. Figure 3 shows the trade-off between time-optimality and acceleration peak

Fig. 4. Desired versus measured acceleration for a linear Cartesian trajectory with jerk constraint (50 m/s^3).

for seven jerk constraints between $[\infty, 50] \text{ m/s}^3$ for this particular example. The normalized measured acceleration peak (acceleration divided by the value for no jerk constraint or $\ddot{t}_x = \infty$) and normalized jerk constraint (jerk divided by the first considered finite jerk constraint, here 500 m/s^3) are given in function of the normalized time (time divided by the value for no jerk constraint). One can see (indicated by the square on Figure 3) that for only a minimal loss in time-optimality (e.g. 1.5%) the acceleration peak is already reduced a lot (e.g. 28%). Furthermore, the acceleration peak only drops under the desired value of 5 m/s^2 for a jerk constraint of 50 m/s^3 .

A solution of the SCP problem, discretized with grid size $K = 100$ of s , with Cartesian jerk constraint (50 m/s^3) was obtained with sufficient precision ($\varepsilon = 10^{-6}$) in four iterations for which we have an optimal motion time of $T^* = 1.38\text{s}$. The inner convex problem is solved in 1.17s , which is similar to the results presented in [1]. The initial point of the SCP algorithm is chosen as a parabola in $b(s)$ [14]. One could also choose to use the result of the optimization problem without jerk constraint as the initial value of the SCP algorithm, however, this does not speed up the convergence and requires an extra solution of a convex problem. Here $\beta = 10^{-6}$ is tuned experimentally and it is observed that it does not influence the convergence much. The desired and measured end effector acceleration along the x-axis are given in Figure 4. Figure 5 shows the desired jerk in the x-direction in function of time. The unmodeled flexibilities still result in accelerations larger than desired but it can be seen that the measured acceleration lies between -5 m/s^2 and 5 m/s^2 , hence the object will not tip over. Furthermore, there are less vibrations.

Figure 6 shows some frames from of a video of the experiment. In the upper part of the frame the robot executes the time-optimal trajectory without jerk constraint. In the lower part the robot executes the time-optimal trajectory with jerk constraint. It can be seen that both motions are very similar, but slower for the case with jerk constraints, and the

Fig. 5. Desired jerk constrained to 50 m/s^3 .

object does not fall off when the jerk constraint is included into the optimization.

To confirm that the jerk constraint is essential to prevent the object from tipping over, and hence, the observed result is not only due to the longer motion time, we can make a remark on the difference in motion time between both cases. Suppose we stretch the trajectory for the case with no jerk constraints, such that the motion time is equal to the case with jerk constraints, the acceleration peaks would still be too large (6.7 m/s^2) due to infinite jerks, hence the object would still tip over.

We have applied the trajectories as a desired joint position for feedback controllers on the joint position. Due to a limited bandwidth of the feedback controllers there will be a tracking error. An additional benefit of trajectories with jerk constraints is that they give rise to a reduced tracking error. For this experiment we see that the RMS (Root Mean Square) value has dropped from $8.29 \cdot 10^{-3}$ for the trajectory without jerk constraint to $5.02 \cdot 10^{-3}$ for the trajectory with jerk constraint, which is a decrease of nearly 40%. Hence, including jerk constraints improves the tracking behaviour of the control system.

To illustrate the efficiency of the SCP algorithm the convergence is given in Figure 7 shows the convergence of the SCP algorithm for this particular example. Here the normalised optimal motion time ($\Delta T/T^*$ with $\Delta T = T^k - T^{k-1}$ and T^* the solution of the SCP algorithm) and the normalised residual ($\|\Delta b\|/\|b^*\|$ with $\Delta b = b^k - b^{k-1}$) are shown for each iteration k .

VI. DISCUSSION

The SCP approach given in section IV is very appealing from a theoretical and practical point of view. Theoretically it ensures that a stationary point of the non-convex optimal path following problem (6)-(9) is attained. Furthermore, the solution is found with high accuracy in only a few iterations, which is important from a practical point of view. The numerical examples given here were performed off-line, but

Fig. 6. Frames taken from a video of the experiment

thanks to its high efficiency, the proposed SCP approach has potential to be used in on-line applications.

To counter the near-infinite acceleration jumps resulting from the convex framework from [1] we have introduced non-convex jerk constraints. For robotic manipulators with unmodeled flexibilities the proposed SCP approach results in the efficient calculation of a solution in only four iterations and an effective reduction of the acceleration peaks (50% for this particular example) and the tracking error (40% for this particular example) with only minimal loss in time-optimality (15% for this particular example).

Also, the SCP approach can easily be implemented. The algorithm can be built around the convex optimal path following framework from [1] with minimal extra effort. The DC decomposition of (9) involves only a few lines of code.

ACKNOWLEDGMENT

This work benefits from K.U.Leuven-BOF PFV/10/002 Center-of-Excellence Optimization in Engineering (OPTEC), the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office (DYSCO), the European research project EMBOCON FP7-ICT-2009-4 248940, project G.0377.09 of the Research Foundation Flanders (FWO) Vlaanderen, and K.U.Leuven's Concerted

Fig. 7. Convergence of the SCP algorithm for $\varepsilon = 10^{-6}$.

Research Action GOA/10/11. Goele Pipeleers is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen).

REFERENCES

- [1] D. Verschuere, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *Automatic Control, IEEE Transactions on*, vol. 54, pp. 2318–2327, oct. 2009.
- [2] K. J. Kyriakopoulos and Saridis, "Minimum jerk path generation," in *Proc. IEEE International Conference Robotics and Automation*, pp. 364–369, 1988.
- [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. Shin and N. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *Automatic Control, IEEE Transactions on*, vol. 30, no. 6, pp. 531–541, 1985.
- [5] F. Debruere, W. Van Loock, G. Pipeleers, M. Diehl, J. Swevers, and J. De Schutter, "Time-optimal robot path following with cartesian acceleration constraints: a convex optimization approach," in *Mechatronics Forum, Linz, 2012*.
- [6] R. Horst and N. Thoai, "Dc programming: Overview," *J. Optim. Theory Appl.*, vol. 103(1), pp. 1–43, 1999.
- [7] D. Pham and H. L. Thi, "A DC optimization algorithms for solving the trust region subproblem," *SIAM J. Optimization*, vol. 8, pp. 476–507, 1998.
- [8] Q. Tran-Dinh, C. Savorgnan, and M. Diehl, "Adjoint-based predictor-corrector sequential convex programming for parametric nonlinear optimization," *SIAM J. Optimization*, (accepted).
- [9] T. D. Quoc, S. Gumussoy, W. Michiels, and M. Diehl, "Combining convex-concave decompositions and linearization approaches for solving bmis, with application to static output feedback," *IEEE Transactions on Automatic Control*, (in print).
- [10] B. K. Sriperumbudur and G. R. G. Lanckriet, "On the convergence of the concave-convex procedure," in *NIPS'09*, pp. 1759–1767, 2009.
- [11] H. Bruyninckx and J. D. Schutter, "Symbolic differentiation of the velocity mapping for a serial kinematic chain," *Mechanism and Machine Theory*, vol. 31, no. 2, pp. 135–148, 1996.
- [12] H. Bruyninckx, P. Soetens, and B. Koninckx, "The real-time motion control core of the Orocos project," in *IEEE International Conference on Robotics and Automation*, pp. 2766–2771, 2003.
- [13] J. Löfberg, "Yalmip: A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, (Taipei, Taiwan), 2004.
- [14] D. Verschuere, M. Diehl, J. De Schutter, and J. Swevers, "On-line time-optimal path tracking for robots," in *Proceedings of the 2009 IEEE international conference on Robotics and Automation, ICRA'09*, (Piscataway, NJ, USA), pp. 610–616, IEEE Press, 2009.