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

Frederik Debrouwere*, Wannes Van Loock*, Goele Pipeleers*, Quoc Tran Dinh†,
Moritz Diehl‡, Joris De Schutter* and Jan Swevers*
Frederik.Debrouwere@mech.kuleuven.be
* Department of Mechanical Engineering, Division PMA, KU Leuven, BE-3001 Heverlee, Belgium
† Laboratory for Information and Inference System, Ecole Polytechnique Federale de Lausanne, Switzerland.
‡ Department of Electrical Engineering, Division SCD, KU Leuven, BE-3001 Heverlee, Belgium

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 with simplified constraints a convex reformulation of this optimal control problem has been derived previously. However, many applications in robotics feature constraints such as velocity-dependent torque constraints or torque rate constraints that destroy the convexity. The present paper proposes an efficient sequential convex programming (SCP) approach to solve the corresponding non-convex optimal control problems by writing the non-convex constraints as a difference of convex (DC) functions, resulting in convex-concave constraints. We consider seven practical applications that fit into the proposed framework even when mutually combined, illustrating the flexibility and practicality of the proposed framework. Furthermore, numerical simulations for some typical applications illustrate the fast convergence of the proposed method in only a few SCP iterations, confirming the efficiency 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 [1]–[3], since the motion planning problem is difficult and highly complex to solve in its entirety [4], [5]. 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 [1]–[3], 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 [6], [7]. This guarantees the efficient computation of globally optimal solutions. However, this convex reformulation fails to address many practical applications, since torque rate constraints, velocity-dependent torque constraints, viscous friction effects or the inclusion of cutting forces at the tooltip, give rise to non-convex formulations. To account for such constraints we propose an extension of the convex framework of [6] using sequential convex programming (SCP), since many of these constraints can easily be written as a difference of two convex (DC) functions [8], [9]. In SCP the concave parts of the constraints are sequentially linearised while preserving the convex parts [10]–[12]. Fast convergence of this method in only three to six iterations is observed for typical applications in robotics, 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 [6] and illustrates the non-convex constraints that are considered in the present paper. Section III describes a DC decomposition of the non-convex constraints and reviews a sequential convex programming approach [10]–[12] to efficiently solve the non-convex optimal path following problem. To illustrate the practical use of the proposed SCP approach, seven typical applications in robotics that result in DC constraints are discussed in Section IV and combined in Section V. In these sections, many numerical examples illustrate the efficiency and practicality of the proposed framework.

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. $\boldsymbol{q}$, and matrices with an upper-case letter, e.g. M. $\boldsymbol{q}_i$ denotes the i-th element of $\boldsymbol{q}$. 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 velocity of a rigid body: ${}^{v}_{f}\boldsymbol{t}^{o}_{w} = (({}^{v}_{f}\boldsymbol{v}^{o}_{w})^T, ({}_{f}\boldsymbol{\omega}^{o}_{w})^T)^T$. Here ${}^{v}_{f}\boldsymbol{v}^{o}_{w}$ and ${}_{f}\boldsymbol{\omega}^{o}_{w}$ 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 $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are expressed.

The relation between the velocity twist and the joint velocity

$\dot{\boldsymbol{q}}$ is defined by

$$ {}^{v}_{f}\boldsymbol{t}^{o}_{w} = {}^{v}_{f}J^{o}_{w}(\boldsymbol{q})\,\dot{\boldsymbol{q}}, \tag{1}$$

where ${}^{v}_{f}J^{o}_{w}(\boldsymbol{q})$ is the geometric robot Jacobian [13], [14].

II. Problem formulation

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

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

where $\boldsymbol{\tau} \in \mathbb{R}^n$ 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 $\dot{\boldsymbol{q}}$, accounting for Coriolis and centrifugal effects and, $\boldsymbol{g}$ is a vector accounting for gravity and other position dependent torques.

Consider a prescribed geometric path $\boldsymbol{q}(s)$ 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

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

where $\boldsymbol{q}' = d\boldsymbol{q}(s)/ds$ and, $\boldsymbol{q}'' = d^2\boldsymbol{q}(s)/ds^2$. Substitution of the above equations in (2) projects the equations of motion onto the path [6]:

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

By introducing the variable

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

it was shown in [6] that the time-optimal path following problem is transformed into a *convex* optimization problem. Indeed, since $b(s) \geq 0$, the total motion time

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

is a convex function. Furthermore, since $b'(s) = 2\ddot{s}$, the torque constraints are linear in $b(s)$ and $b'(s)$. The time-optimal path following problem is then reformulated as

$$\begin{aligned} \underset{b(\cdot)}{\text{minimize}} \quad & \int_0^1 \frac{1}{\sqrt{b(s)}} ds \\ \text{subject to} \quad & b(0) = \dot{s}_0^2, b(1) = \dot{s}_T^2 \\ & b(s) \geq 0 \\ & \boldsymbol{h}(s, b(s)) \leq \boldsymbol{\rho}(s) \\ & \text{for } s \in [0, 1]. \end{aligned} \tag{6}$$

where $\boldsymbol{h}(s, b(s)) \leq \boldsymbol{\rho}(s)$ is interpreted as a componentwise inequality. To render the problem convex, $\boldsymbol{h}(s, b(s))$ is restricted to convex functions in the optimization variables.

In [6] some constraints are described that maintain the convexity of the problem. Torque constraints

$$\boldsymbol{\tau}_{-}(s) \leq \boldsymbol{\tau}(s) \leq \boldsymbol{\tau}_{+}(s), \tag{7}$$

are convex with $\boldsymbol{\rho}(s) = \left(\boldsymbol{\tau}_{+}(s)^T, -\boldsymbol{\tau}_{-}(s)^T\right)^T$ and $\boldsymbol{h}(s, b(s)) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \left(\boldsymbol{m}(s)\frac{b'(s)}{2} + \boldsymbol{c}(s)b(s) + \boldsymbol{g}(s)\right)$. Symmetric joint velocity constraints or joint acceleration constraints also keep the problem convex. Furthermore, the objective function can for instance be supplemented with thermal energy terms or integrals of the absolute value of the rate of change of the torque.

However, many practical constraints do not have a convex formulation. In practice, torque constraints are speed dependent [15] and not simple bounds as in (7). Also, to cope with the limited bandwidth of physical actuators, torque rate constraints should be added to the problem. Moreover, robotic manipulators are often used in milling or grinding applications. In this context, cutting forces at the end-effector must be taken into account. As will be detailed in Section IV, all these constraints are non-convex constraints of the form $\boldsymbol{h}(s, b(s)) \leq \boldsymbol{\rho}(s)$, where $\boldsymbol{h}(s, b(s)) : \mathbb{R}^2 \mapsto \mathbb{R}^l$ is a non-convex function of the form

$$\boldsymbol{h}_i(s, b(s)) = \boldsymbol{\phi}_i(s, b(s)) + \boldsymbol{\psi}_i(s)b(s)^{\boldsymbol{\kappa}_i},\; i = 1, \ldots, l, \tag{8}$$

here $\boldsymbol{\phi}(s, \cdot)$ are linear vector functions of the optimization variables, $\boldsymbol{\psi}(s)$ and $\boldsymbol{\kappa}$ are vectors independent of the optimization variables.

The convexity of (8) depends on the sign of $\boldsymbol{\psi}_i(s)$ and the value of $\boldsymbol{\kappa}_i$ (see Section III). If non-convex, the constraint can easily be decomposed as a difference of two convex functions (DC constraint). The following section illustrates this decomposition and reviews a sequential convex programming approach to efficiently solve optimization problem (6)-(8).

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 [6]. Here linear constraints maintain the relationship between the derivatives b', ... of b. Thus, from now on we work in a finite dimensional setting.

III. Sequential convex programming algorithm

Consider the optimization problem with DC (difference of two convex functions) constraints:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{u}_i(\boldsymbol{x}) - \boldsymbol{v}_i(\boldsymbol{x}) \leq 0,\; i = 1, \ldots, l \\ & \boldsymbol{x} \in \Omega, \end{aligned} \tag{9}$$

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

$$\mathcal{D} = \{\boldsymbol{x} \in \Omega \mid \boldsymbol{u}_i(\boldsymbol{x}) - \boldsymbol{v}_i(\boldsymbol{x}) \leq 0, i = 1, \ldots, l\}$$

and its relative interior by

$$\mathrm{ri}(\mathcal{D}) = \{\boldsymbol{x} \in \mathrm{ri}(\Omega) \mid \boldsymbol{u}_i(\boldsymbol{x}) - \boldsymbol{v}_i(\boldsymbol{x}) < 0, i = 1, \ldots, l\}.$$

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

Suppose that $\boldsymbol{x}^k \in \Omega$ is a given point, then the linearized problem of (9) around $\boldsymbol{x}^k$ is

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\boldsymbol{x}) + \tfrac{\beta}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|_2^2 \\ \text{subject to} \quad & \boldsymbol{u}_i(\boldsymbol{x}) - \boldsymbol{v}_i(\boldsymbol{x}^k) - \nabla_{\boldsymbol{x}}\boldsymbol{v}_i(\boldsymbol{x}^k)(\boldsymbol{x}-\boldsymbol{x}^k) \le 0, \\ & i=1,\dots,l, \\ & \boldsymbol{x}\in\Omega, \end{aligned} \tag{10}$$

where a regularization term with parameter β is added to the objective function. The following algorithm for solving (9) is proposed in [8], [10], [11].

Sequential convex programming algorithm

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

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

1) Solve the convex problem (10) to obtain a solution $\boldsymbol{x}^{k+1}$.
2) If $\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\| \le \varepsilon$ for a given tolerance $\varepsilon > 0$ then terminate. Otherwise, set $k = k + 1$ and go to step 1.

Note that when the initial point $\boldsymbol{x}^0 \in \mathrm{ri}(\mathcal{D})$, then the algorithm generates a sequence of points $\boldsymbol{x}^k$ 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 (9) [11]. The regularisation parameter β ensures a downhill search direction of the algorithm, however it is not always needed and can be chosen small depending on the gradient of the objective function [11].

To apply the algorithm to problem (6)-(8), we need to determine a DC decomposition of

$$\boldsymbol{\phi}_i(s, b(s)) + \boldsymbol{\psi}_i(s) b(s)^{\boldsymbol{\kappa}_i} \le \boldsymbol{\rho}_i(s),\ i = 1, \dots, l, \tag{11}$$

by examining its convexity, which depends on the values of $\boldsymbol{\psi}_i(s)$ and $\boldsymbol{\kappa}_i$. Since $b(s) \ge 0$, (11) is non-convex whenever

$$\begin{aligned} \boldsymbol{\psi}_i(s) > 0 \ \text{ and } \ 0 < \boldsymbol{\kappa}_i < 1, \ \text{ or} \\ \boldsymbol{\psi}_i(s) < 0 \ \text{ and } \ \boldsymbol{\kappa}_i > 1 \cup \boldsymbol{\kappa}_i < 0. \end{aligned}$$

In the case that (11) is non-convex, an obvious DC decomposition as in (9) is

$$\boldsymbol{u}_i(b(s)) = \boldsymbol{\phi}_i(s, b(s)) - \boldsymbol{\rho}_i(s) \text{ and } \boldsymbol{v}_i(b(s)) = -\boldsymbol{\psi}_i(s) b(s)^{\boldsymbol{\kappa}_i}.$$

This is only one of the several possible choices of DC decompositions. The convergence of the algorithm is faster when the curvature of the concave part is small [11], hence, it is worth to find a DC decomposition that has small curvature in the concave part, as the one given above.

As we are interested in time-optimal solutions, the stopping criterion of the SCP algorithm may be changed to:

$$\|T^{k+1} - T^k\| \le \varepsilon,$$

where T^k denotes the optimal travel time for iteration k. This stopping condition allows exiting the SCP algorithm earlier, since the convergence of T^k is observed to be faster than the convergence of $b^k(s)$, as will be illustrated in Section IV.

In the case of the considered path following problem (6) the variable $\boldsymbol{x}$ represents the discrete values $\boldsymbol{b}$ of $b(s)$ on the grid points of s. The initial point $\boldsymbol{x}^0(s) = \boldsymbol{b}^0$ of the SCP algorithm is chosen as a parabola in s with a free parameter similar to [7]. This free parameter is changed until $b^0(s)$ satisfies the constraints of the optimization problem for all s. The regularisation parameter β is tuned experimentally and it is observed that it does not influence the convergence much.

IV. Applications

In this section, typical robotic applications are described that fit in problem formulation (6)-(8). Here we show that these non-convex constraints can easily be transformed into convex-concave constraints of the form (8). Numerical results show that in practice only a few SCP iterations are necessary to solve the corresponding time-optimal path following problem, illustrating the efficiency of the proposed framework.

The applications we consider result in the addition of new constraints to the framework of [6] or the adaptation of existing constraints. Some of these applications involve constraints on, or extensions to, the robot dynamics as defined in [6] and (2). Therefore, we redefine the actuator torque

$$\tilde{\boldsymbol{\tau}} = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathrm{g}(\boldsymbol{q}) \tag{12}$$

to be the torque following from the nominal robot dynamics as used in [6], and $\boldsymbol{\tau}$ to be the total actuator torques, including some extensions to the robot dynamics resulting from additional forces and robot loads.

The SCP algorithm is implemented in MATLAB using the free high-level optimization modeling tool YALMIP [16] on an Intel Core i3 CPU running at a 2.53GHz Windows machine. All numerical examples illustrating the applications are considering a seven degree of freedom robot executing a line trajectory with starting point $(-0.5, -0.4, 0.2)$ and end point $(0.5, -0.4, 0.2)$, expressed in the base frame of the robot, while the end effector frame is aligned with the base frame of the robot with its z-axis pointing downwards. The kinematic and dynamic parameters of the robot are given in the Appendix. The optimization problem is discretized over one hundred samples.

A. *Affine velocity-dependent torque constraint*

In this section we consider affine joint velocity-dependent constraints on $\tilde{\boldsymbol{\tau}}$ (12). It will be shown below that these constraints can be used to impose velocity-dependent constraints on the total actuator torque $\boldsymbol{\tau}$ (to account for viscous joint friction). Joint velocity-dependent torque constraints, together with a torque constraint and an artificial joint velocity constraint, describe the interior of a polytope for every joint $i = 1..n$:

$$\tilde{\boldsymbol{\tau}}_{-,i}(s) \le \tilde{\boldsymbol{\tau}}_i(s) \le \tilde{\boldsymbol{\tau}}_{+,i}(s), \tag{13a}$$

$$\boldsymbol{f}^i(s)\tilde{\boldsymbol{\tau}}_i(s) + \boldsymbol{h}^i(s)\dot{\boldsymbol{q}}_i(s) \le \boldsymbol{p}^i(s), \tag{13b}$$

$$\dot{\boldsymbol{q}}_{-,i}(s) \le \dot{\boldsymbol{q}}_i(s) \le \dot{\boldsymbol{q}}_{+,i}(s), \tag{13c}$$

where $\boldsymbol{f}^i(s) \in \mathbb{R}^{m\times 1}$, $\boldsymbol{h}^i(s) \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{p}^i(s) \in \mathbb{R}^{m\times 1}$ are the vectors that define the velocity-dependent torque constraint for joint i.

An example of a polytope of the form (13a)-(13b) is given in Figure 1.A. For n joints these n polytope constraints are expressed compactly as:

$$\begin{aligned} \tilde{\boldsymbol{\tau}}_-(s) \leq \tilde{\boldsymbol{\tau}}(s) &\leq \tilde{\boldsymbol{\tau}}_+(s), \\ T(s)\tilde{\boldsymbol{\tau}}(s) + Q(s)\dot{\boldsymbol{q}}(s) &\leq \boldsymbol{p}(s), \\ \dot{\boldsymbol{q}}_-(s) \leq \dot{\boldsymbol{q}}(s) &\leq \dot{\boldsymbol{q}}_+(s), \end{aligned}$$

with

$$T(s) = \begin{bmatrix} \boldsymbol{f}^1(s) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{f}^2(s) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{f}^n(s) \end{bmatrix} \in \mathbb{R}^{m\cdot n \times n},$$

$$Q(s) = \begin{bmatrix} \boldsymbol{h}^1(s) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{h}^2(s) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{h}^n(s) \end{bmatrix} \in \mathbb{R}^{m\cdot n \times n},$$

and $\boldsymbol{p}(s) = \left(\boldsymbol{p}^1(s)^T, \boldsymbol{p}^2(s)^T, \cdots, \boldsymbol{p}^n(s)^T\right)^T \in \mathbb{R}^{m\cdot n\times 1}$. Substituting the expression for the nominal actuator torques $\tilde{\boldsymbol{\tau}}(s)$ and (3) allows us to rewrite the polytope constraint as:

$$\tilde{\boldsymbol{\tau}}_-(s) \leq \left(\boldsymbol{m}(s)\frac{b'(s)}{2} + \boldsymbol{c}(s)b(s) + \boldsymbol{g}(s)\right) \leq \tilde{\boldsymbol{\tau}}_+(s), \quad (14a)$$

$$T(s)\left(\boldsymbol{m}(s)\frac{b'(s)}{2} + \boldsymbol{c}(s)b(s) + \boldsymbol{g}(s)\right) + Q(s)\boldsymbol{q}'(s)\sqrt{b(s)} \leq \boldsymbol{p}(s), \quad (14b)$$

$$\dot{\boldsymbol{q}}_-(s) \leq \boldsymbol{q}'(s)\sqrt{b(s)} \leq \dot{\boldsymbol{q}}_+(s). \quad (14c)$$

Note the convexity of (14a). By carefully squaring (14c), it becomes linear in $b(s)$ and thus convex. Suppose $\dot{\boldsymbol{q}}_{+,i} > 0$ and $\dot{\boldsymbol{q}}_{-,i} < 0$ then we can square $\boldsymbol{q}'_i\sqrt{b} \leq \dot{\boldsymbol{q}}_{+,i}$ if $\boldsymbol{q}'_i > 0$ ($\dot{\boldsymbol{q}}_{-,i} \leq \boldsymbol{q}'_i\sqrt{b}$ can be neglected since $b > 0$, $\dot{\boldsymbol{q}}_{-,i} < 0$ and $\boldsymbol{q}'_i > 0$) and square $-\dot{\boldsymbol{q}}_{-,i} \geq -\boldsymbol{q}'_i\sqrt{b}$ if $\boldsymbol{q}'_i < 0$. Note furthermore the non-convexity of the joint velocity-depending torque constraint (14b) due to $\sqrt{b}$. The following application shows how this non-convexity is eliminated in the literature with conservative approximations and shows how the proposed SCP approach results in a non-conservative optimization problem.

1) Actuator torque characteristic: In [6] the actuator torque bounds are assumed to be constant, which follows from an armature current constraint to prevent overheating of the (e.g. brushless DC) actuator. If we want to fully utilize the actuator performance in the optimization problem we also have to consider a bound on the voltage that drives the actuator [17]. This voltage constraint will become important at high velocities, which are likely in time-optimal motions. As the produced counter-EMF (counter-electromotive force) is proportional to the joint angular velocity, the voltage constraint approximately imposes a linear decrease of the available actuator torque as a function of the joint angular velocity. Figure 1.A shows a typical (linearly approximated) torque-speed characteristic for a brushless DC-motor in four quadrants, which follows form the current (I) and voltage (II) constraints.

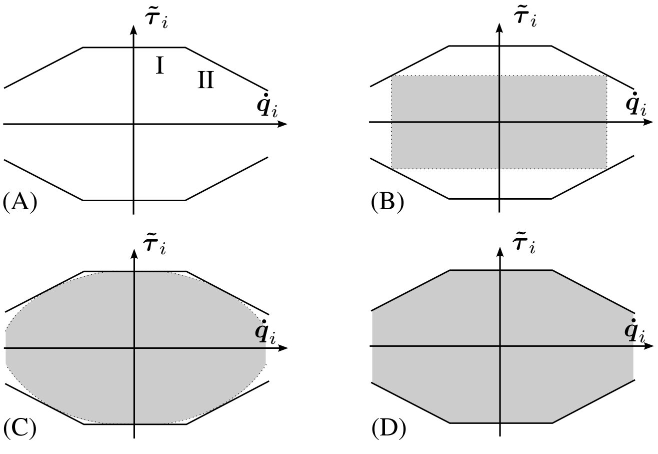

$\tilde{\boldsymbol{\tau}}_i$
I
II
$\dot{\boldsymbol{q}}_i$
(A)
$\tilde{\boldsymbol{\tau}}_i$
$\dot{\boldsymbol{q}}_i$
(B)
$\tilde{\boldsymbol{\tau}}_i$
$\dot{\boldsymbol{q}}_i$
(C)
$\tilde{\boldsymbol{\tau}}_i$
$\dot{\boldsymbol{q}}_i$
(D)

Fig. 1. (A) Linear approximation of the torque-velocity characteristic, (B) conservative constraints in the convex framework of [6], (C) conservative convex approximation of the polytope from [15] and (D) the SCP approach with unchanged polytope.

The entire torque-velocity characteristic in the four quadrants for joint i are then expressed as:

$$\text{I. current bound:} \quad \tilde{\boldsymbol{\tau}}_{-,i} \leq \tilde{\boldsymbol{\tau}}_i \leq \tilde{\boldsymbol{\tau}}_{+,i}, \quad (15)$$

$$\text{II. voltage bound:} \quad \boldsymbol{f}_v^i\tilde{\boldsymbol{\tau}}_i + \boldsymbol{h}_v^i\dot{\boldsymbol{q}}_i \leq \boldsymbol{p}_v^i, \quad (16)$$

for $i = 1..n$, where $\boldsymbol{f}_v^i \in \mathbb{R}^{4\times 1}$, $\boldsymbol{h}_v^i \in \mathbb{R}^{4\times 1}$ and $\boldsymbol{p}_v^i \in \mathbb{R}^{4\times 1}$ determine the voltage bound in four quadrants for every joint. These constraints can be written in the form of (13) with

$$\boldsymbol{f}^i(s) = \boldsymbol{f}_v^i,\ \boldsymbol{h}^i(s) = \boldsymbol{h}_v^i \text{ and } \boldsymbol{p}^i(s) = \boldsymbol{p}_v^i.$$

The convex framework of [6] considers constant, velocity-independent, actuator constraints, and hence avoids this non-convexity, yielding torque bounds which describe the largest rectangular polytope of the form (13a)-(13c) inside the polytope (13). This requires an artificial joint-velocity constraint. As can be seen in Figure 1.B, these simplifications induce conservatism over the whole velocity range. In [15] the authors make a conservative convex approximation of the non-convex constraint (13b): $\tilde{\boldsymbol{f}}^i(s)\tilde{\boldsymbol{\tau}}_i + \tilde{\boldsymbol{h}}^i(s)\dot{\boldsymbol{q}}_i^2 \leq \tilde{\boldsymbol{p}}^i(s)$ (see Figure 1.C). Hence substitution of (3) results in a term $\tilde{\boldsymbol{h}}^i(s)(\boldsymbol{q}'_i)^2 b(s)$ which in turn results in a convex, but conservative (mainly at high velocities), optimization problem. Since both simplifications lead to suboptimal actuator use due to the conservative approximations, we reformulate (14b) as a DC constraint of the form (11) with

$$\begin{gathered} \boldsymbol{\phi}(s, b(s)) = T(s)\left(\boldsymbol{m}(s)\frac{b'(s)}{2} + \boldsymbol{c}(s)b(s) + \boldsymbol{g}(s)\right), \\ \boldsymbol{\psi}(s) = Q(s)\boldsymbol{q}'(s),\ \boldsymbol{\rho}(s) = \boldsymbol{p}(s) \text{ and } \boldsymbol{\kappa} = \frac{1}{2}\mathbf{1}_n. \end{gathered} \quad (17)$$

Note that there is no need for an artificial joint-velocity constraint (14c). In constrast to the methods in [6] and [15], the proposed SCP approach results in optimal actuator use as can be seen from Figure 1.D and the following example.

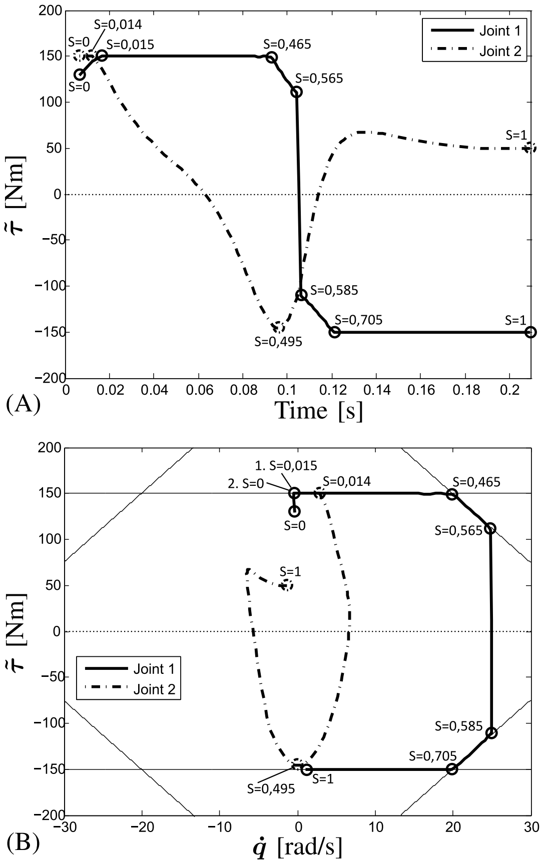


Fig. 2. Actuator torque, as the result of the optimization, in function of time (A) and joint speed (B).

The results of a numerical simulation, for a seven dof robotic manipulator executing a line trajectory, are given in Figure 2. The values used for the voltage constraint vectors $\boldsymbol{f}_v^i$, $\boldsymbol{h}_v^i$ and $\boldsymbol{p}_v^i$ are given in the Appendix. A solution was obtained with sufficient precision ($\varepsilon = 10^{-8}$) in five iterations. The inner convex problem is solved in ± 1.5 s, which is similar to the results presented in [6]. Only the actuator torque of joint 1 and 2 are shown, since the other torques are small in comparison and do not hit any bound. The actuator torque is plotted as a function of time in 2.A and as a function of the joint speed in 2.B, where the velocity-dependent torque constraints are shown in thin lines. The optimal motion is of the bang-bang type, since at any s, there is always one actuator where the torque hits one of the bounds, which is to be expected for a time-optimal motion.

Figure 3 illustrates the convergence of the SCP algorithm. Here the normalised optimal motion time ($\Delta T/T^*$ with $\Delta T = T^{k+1} - T^k$ and T^* the solution of the SCP algorithm) and the normalised residual ($||\Delta \boldsymbol{b}||/||\boldsymbol{b}^*||$ with $\Delta \boldsymbol{b} = \boldsymbol{b}^{k+1} - \boldsymbol{b}^k$) are shown for each iteration. After the second iteration the accuracy is already high.

Using the convex framework of [6] the constant torque constraint is chosen conservatively ($\tilde{\boldsymbol{\tau}}_+ = 120Nm$) and results in an optimal motion time of $T^*_{\text{convex}} = 0.265s$. Including the speed dependence of the torque constraint results in an optimal motion time of $T^*_{\text{SCP}} = 0.236s$, which is 12% faster for this example thanks to the full use of the actuator's capacity.

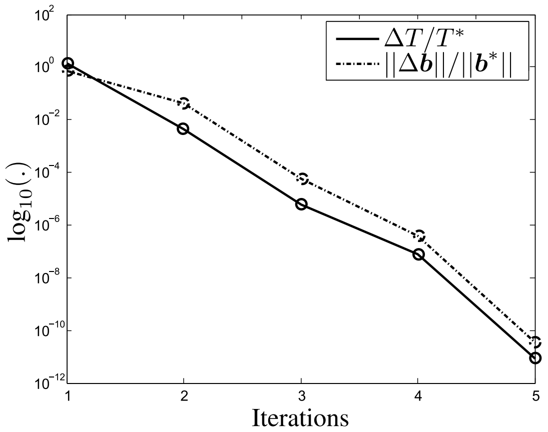


Fig. 3. Convergence of the SCP algorithm.

2) Viscous friction in the joints: Viscous friction in the joints results in an extension of the robot dynamics given as in (2):

$$\boldsymbol{\tau} = M(\boldsymbol{q})\ddot{\boldsymbol{q}}+C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}+\boldsymbol{g}(\boldsymbol{q})+B(\boldsymbol{q})\dot{\boldsymbol{q}} = \tilde{\boldsymbol{\tau}}+B(\boldsymbol{q})\dot{\boldsymbol{q}}, \quad (18)$$

where $B(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is a diagonal matrix accounting for viscous friction in the joints. Hence, viscous friction introduces an affine velocity dependence, and results in non-convex constraints

$$\boldsymbol{\tau}_-(s) \le \tilde{\boldsymbol{\tau}} + B(s)\boldsymbol{q}(s)'\sqrt{b(s)} \le \boldsymbol{\tau}_+(s).$$

These constraints describe an open polytope in the four quadrants of the $\tilde{\boldsymbol{\tau}}_i$-$\dot{\boldsymbol{q}}_i$ plane for every joint i, of the form (14) where we only have (14b), with:

$$\boldsymbol{f}^i(s) = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \ \boldsymbol{h}^i(s) = B_{ii}(s)\boldsymbol{f}^i \text{ and } \boldsymbol{p}^i(s) = \begin{pmatrix} \boldsymbol{\tau}_{+,i}(s) \\ -\boldsymbol{\tau}_{-,i}(s) \end{pmatrix},$$

where $B_{ii}(s)$ indicates the i-th diagonal element of $B(s)$.

Due to this non-convexity, viscous friction in the joints is either neglected in [6] or it is included in the torque bounds in [6] and approximated by constant, velocity-independent, torque bounds $\tilde{\boldsymbol{\tau}}_- \le \tilde{\boldsymbol{\tau}} \le \tilde{\boldsymbol{\tau}}_+$ which ensure that $\boldsymbol{\tau}_- - B(\boldsymbol{q})\dot{\boldsymbol{q}} \le \tilde{\boldsymbol{\tau}} \le \boldsymbol{\tau}_+ - B(\boldsymbol{q})\dot{\boldsymbol{q}}$ for some artificial velocity range between $\dot{\boldsymbol{q}}_-$ and $\dot{\boldsymbol{q}}_+$, introducing an artificial constraint (14c). As can be seen from Figure 4.A, the larger the viscous friction, the more conservative these bounds will be.

In this paper, we include the viscous friction by treating the introduced velocity-dependence as a polytope of the form (14) and by using the proposed SCP approach with the DC decomposition of (14b) as in (17). This results in optimal actuator use (see Figure 4.B).

B. Forces at the tool tip proportional to the path velocity

Assume the robot is using a tool for interaction with the environment. Obviously, the tool contact forces influence the robot dynamics. This section focusses on the effect of tool contact forces that are a function of the path velocity $\dot{s}$ onto the robot dynamics and their DC decomposition. Figure 5

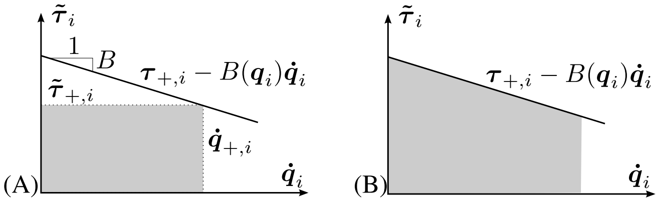


Fig. 4. Incorporation of the viscous friction into the available actuator torque. In the convex framework this results in a conservative torque bound (A), however in the non-convex SCP framework the actuator is used at full capacity (B).

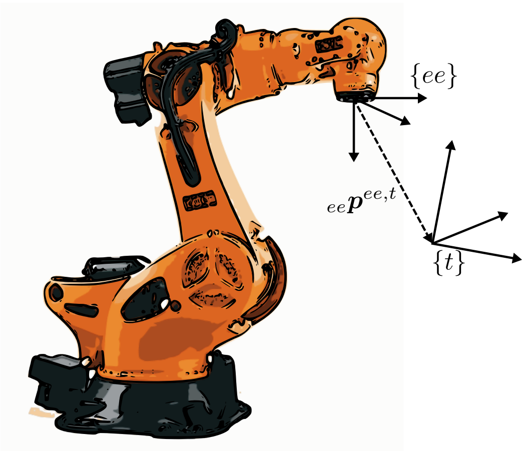


Fig. 5. Schematic of a serial robot holding a tool with tool frame $\{t\}$ with orientation ${}^{t}_{ee}R$ and position ${}_{ee}\boldsymbol{p}^{ee,t}$ with respect to the end effector frame.

illustrates the notation used. The tool frame $\{t\}$ has orientation ${}^{t}_{ee}R = {}^{ee}_{t}R^T$ and position ${}_{ee}\boldsymbol{p}^{ee,t}$ with respect to the end effector frame $\{ee\}$.

The actuator torque equation in [6] only takes into account the torque produced by the motion of the links of the robot, so when there is no contact with the environment. However, the tool contact forces ${}_{t}\boldsymbol{f}^{t} \in \mathbb{R}^{3\times 1}$ and contact torques ${}_{t}\boldsymbol{\tau}^{t} \in \mathbb{R}^{3\times 1}$ (with reference point at the tool tip), both expressed in the tool tip frame, stacked in the contact wrench ${}_{t}\boldsymbol{w}^{t} = \left(\left({}_{t}\boldsymbol{f}^{t}\right)^T, \left({}_{t}\boldsymbol{\tau}^{t}\right)^T\right)^T \in \mathbb{R}^{6\times 1}$, introduce extra actuator torques $W(s){}_{t}\boldsymbol{w}^{t}(s)$ [13]. The robot dynamics are now given by:

$$\boldsymbol{\tau} = \tilde{\boldsymbol{\tau}} + W(s)\ {}_{t}\boldsymbol{w}^{t}(s,\dot{s}), \tag{19}$$

where $W(s) = N^{-1}\ \left({}^{ee}_{ee}J^{ee}_{bs}(s)\right)^T\ {}^{t}_{ee}S(s) \in \mathbb{R}^{n\times 6}$. ${}^{ee}_{ee}J^{ee}_{bs}$ is the body-fixed[1] representation [14] of the geometrical Jacobian, ${}^{t}_{ee}S(s) = \begin{bmatrix} {}^{t}_{ee}R & 0_{3\times 3} \\ {}_{ee}\boldsymbol{p}^{ee,t} \times {}^{t}_{ee}R & {}^{t}_{ee}R \end{bmatrix}$ the wrench (screw) transformation matrix between the tool frame $\{t\}$ and the end effector frame $\{ee\}$. N is the diagonal actuator-joint transmission ratio matrix.

In this section we are interested in contact wrenches that are a function of the path velocity $\dot{s}$ and can be written as

$${}_{t}\boldsymbol{w}^{t}(s,\dot{s}) = \boldsymbol{w}_c(s)\dot{s}^{2\kappa} = \boldsymbol{w}_c(s)b(s)^{\kappa}. \tag{20}$$

Physically this means that we are interested in forces and torques that are proportional to a power of the velocity along the path. These forces are determined by the desired path following task. Below, two applications (cutting forces and viscous friction at the tool tip) are given that result in this type of contact wrench.

Substitution into (19) and substitution of (12) and (3) allows us to write the joint torques as

$$\boldsymbol{\tau}(s) = \boldsymbol{m}(s)\frac{b'(s)}{2} + \boldsymbol{c}(s)b(s) + \boldsymbol{g}(s) + W(s)\boldsymbol{w_c}(s)b(s)^{\kappa},$$

Then, the actuator torque constraints can be rewritten in the form of (11) with

$$\boldsymbol{\phi}(s, b(s)) = \begin{pmatrix} -\boldsymbol{m}(s)\frac{b'(s)}{2} - \boldsymbol{c}(s)b(s) - \boldsymbol{g}(s) \\ \boldsymbol{m}(s)\frac{b'(s)}{2} + \boldsymbol{c}(s)b(s) + \boldsymbol{g}(s) \end{pmatrix},$$

$$\boldsymbol{\psi}(s) = \begin{pmatrix} -W(s)\boldsymbol{w_c}(s) \\ W(s)\boldsymbol{w_c}(s) \end{pmatrix},\ \boldsymbol{\rho}(s) = \begin{pmatrix} -\boldsymbol{\tau}_{-}(s) \\ \boldsymbol{\tau}_{+}(s) \end{pmatrix} \text{ and } \boldsymbol{\kappa} = \kappa\boldsymbol{1}_{2n}.$$

1) Cutting forces at the tool tip: In milling or grinding the (contact) cutting forces are given by [18]:

$${}_{t}\boldsymbol{w}^{t}(s,\dot{s}) = \boldsymbol{c_1}r(s,\dot{s})^{\lambda},$$

where $r(s,\dot{s}) = \boldsymbol{c_2}\,{}^{t}_{t}\boldsymbol{v}^{t}_{bs}(s,\dot{s})$ is the feed rate which is tangential to the geometric trajectory, $\boldsymbol{c_1} \in \mathbb{R}^{6\times 1}$ is a vector depending on cutting conditions and the parameter λ represents the influence of the cutting depth. In the definition of the feed rate r, ${}^{t}_{t}\boldsymbol{v}^{t}_{bs}(s,\dot{s})$ is the translational speed of the origin of $\{t\}$ on the tool tip with respect to the base frame, expressed in the tool frame. In this application we align one of the axes of the tool frame with the direction of the feed rate r, where this alignment is described by $\boldsymbol{c_2} \in \mathbb{R}^{1\times 3}$, so the tool frame is aligned with the task. Hence the forces parallel and orthogonal to the feed rate can be expressed easily in the tool frame by means of $\boldsymbol{c_1}$. This also implies that $r(s,\dot{s}) \geq 0$ for $s \in [0,1]$. We now assume that the orientation between the tool frame $\{t\}$ and the end effector frame $\{ee\}$ does not change so that the end effector frame is also aligned with the feed rate. This is taken care of during path planning. Hence we get a constant ${}^{t}_{ee}R$.

Now, using some velocity manipulations[2] we can write ${}^{t}_{t}\boldsymbol{v}^{t}_{bs}(s,\dot{s}) = S_v\ {}^{ee}_{ee}\boldsymbol{t}^{ee}_{bs}(s,\dot{s})$ with $S_v = \begin{bmatrix} {}^{t}_{ee}R & \left({}^{t}_{ee}R\left(-{}_{ee}\boldsymbol{p}^{ee,t}\right)\right) \times {}^{t}_{ee}R \end{bmatrix}$. Then by using (1) and (3) the tool velocity can be written as

$${}^{t}_{t}\boldsymbol{v}^{t}_{bs}(s,\dot{s}) = \boldsymbol{\nu}(s)\dot{s}, \tag{22}$$

with $\boldsymbol{\nu}(s) = S_v\ {}^{ee}_{ee}J^{ee}_{bs}(s)\boldsymbol{q'}(s)$. This allows us to write the contact wrench as in (20) with[3]

$$\boldsymbol{w_c}(s) = \boldsymbol{c}_1(\boldsymbol{c}_2\boldsymbol{\nu}(s))^{\lambda} \text{ and } \kappa = \frac{\lambda}{2}.$$

[1] This Jacobian describes the motion of the end effector frame $\{ee\}$ with respect to the robot base frame $\{bs\}$, with reference point for the translational velocity in the origin of $\{ee\}$, and expressed in $\{ee\}$.

[2] Since ${}^{t}_{t}\boldsymbol{t}^{t}_{bs} = {}^{ee}_{t}S\ {}^{ee}_{ee}\boldsymbol{t}^{t}_{bs}$ with the screw transformation matrix ${}^{ee}_{t}S = \begin{bmatrix} {}^{ee}_{t}R & {}_{t}\boldsymbol{p}^{t}_{ee} \times {}^{ee}_{t}R \\ \boldsymbol{0} & {}^{ee}_{t}R \end{bmatrix}^T$, ${}_{t}\boldsymbol{p}^{t}_{ee} = {}^{ee}_{t}R\ \left(-{}_{ee}\boldsymbol{p}^{ee,t}\right)$ and ${}^{ee}_{ee}\boldsymbol{t}^{t}_{ee} + {}^{ee}_{ee}\boldsymbol{t}^{ee}_{bs} = {}^{ee}_{ee}\boldsymbol{t}^{t}_{bs}$ where ${}^{ee}_{ee}\boldsymbol{t}^{t}_{ee} = 0$ since the tool frame does not move with respect to the end-effector frame.

[3] We have that $r(s,\dot{s}) \geq 0$ and $\dot{s} > 0$ [6] for $s \in [0,1]$, hence the scalar $\boldsymbol{c}_2\boldsymbol{\nu}(s) \geq 0$ for $s \in [0,1]$

2) Viscous friction at the tool tip: In this paragraph we consider time-optimal tasks that include viscous friction at the tool tip, e.g. a robot performing a writing task on a whiteboard with a marker [6]. We consider the contact to be a point and to lie at the origin of $\{t\}$. Therefore there will only be a contact force, which can be written as ${}_t\boldsymbol{f}^t(s,\dot{s}) = B_t(s)\ {}^t_t\boldsymbol{v}^t_{bs}(s,\dot{s})$ where $B_t(s) \in \mathbb{R}^{3\times 3}$ is a matrix accounting for viscous friction. Again, we assume the orientation of the tool frame with respect to the end effector frame to be constant. Hence, the end effector frame is aligned with the task. Using (22) we can write the contact wrench as in (20) with

$$\boldsymbol{w_c}(s) = \begin{pmatrix} B_t(s)\boldsymbol{\nu}(s) \\ \mathbf{0} \end{pmatrix} \text{ and } \kappa = \frac{1}{2}. \tag{23}$$

Note on Coulomb friction:
Coulomb friction at the tool tip will depend on the normal force the tool exerts on the object the robot tool is in contact with, but also on the velocity of the tool which determines the direction of the friction force. Hence the Coulomb friction force can be writen as ${}_t\boldsymbol{f}^t \sim \text{sign}({}^t_t\boldsymbol{v}^t_{bs})$. By using (22) we can write ${}_t\boldsymbol{f}^t \sim \text{sign}\left(\boldsymbol{\nu}(s)\right)$ since $\dot{s} \geq 0$. Hence Coulomb friction is independent of $b(s)$ and results only in an extra term in $\boldsymbol{g}(s)$ since $\dot{s}$ has been eliminated from the contact force. Hence, similar to static friction in the joints [6], static friction at the tool tip preserves the convexity of the optimization problem.

C. Torque rate constraints

Due to the bang-bang nature of time-optimal trajectories, the framework in [6] will result in near infinitely fast torque changes, and hence near infinite torque rates. However, physical actuators cannot realise these infinite torque rates due to electromechanical effects. To cope with this limitation, a torque rate constraint is introduced. Also, adding torque rate constraints results in smoother actuator loads [19]–[21].

The torque rate is found by taking the time derivative of the actuator torque equation (4)

$$\begin{aligned}\dot{\boldsymbol{\tau}} &= \frac{d\left(\boldsymbol{m}(s)\ddot{s} + \boldsymbol{c}(s)\dot{s}^2 + \boldsymbol{g}(s)\right)}{dt}, \\ &= \boldsymbol{m}'(s)\dot{s}\ddot{s} + \boldsymbol{m}(s)\dddot{s} + \boldsymbol{c}'(s)\dot{s}\dot{s}^2 + 2\boldsymbol{c}(s)\dot{s}\ddot{s} + \boldsymbol{g}'(s)\dot{s}\end{aligned}$$

which is constrained by $\dot{\boldsymbol{\tau}}_-$ and $\dot{\boldsymbol{\tau}}_+$. Applying the chain rule results in:

$$\dot{s} = \sqrt{b(s)}, \ddot{s} = \frac{1}{2}b'(s), \dddot{s} = \frac{1}{2}b''(s)\sqrt{b(s)}. \tag{24}$$

This is substituted in the expression of the torque rate. Now note that in all terms of $\dot{\boldsymbol{\tau}}$ the expression $\sqrt{b(s)}$ can be factored out. This allows us to write the torque rate constraint as in (11) with $\boldsymbol{\kappa} = -\frac{1}{2}\mathbf{1}_{2n}$, $\boldsymbol{\rho}(s) = \mathbf{0}$ and

$$\begin{aligned}\boldsymbol{\phi}(s, b(s)) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \Big(&\boldsymbol{g}'(s) + \boldsymbol{c}'(s)b(s) \\ &+ \left(\boldsymbol{c}(s) + \tfrac{1}{2}\boldsymbol{m}'(s)\right)b'(s) + \tfrac{1}{2}\boldsymbol{m}(s)b''(s)\Big),\end{aligned}$$

$$\boldsymbol{\psi}(s) = \begin{pmatrix} \dot{\boldsymbol{\tau}}_-(s) \\ -\dot{\boldsymbol{\tau}}_+(s) \end{pmatrix}.$$

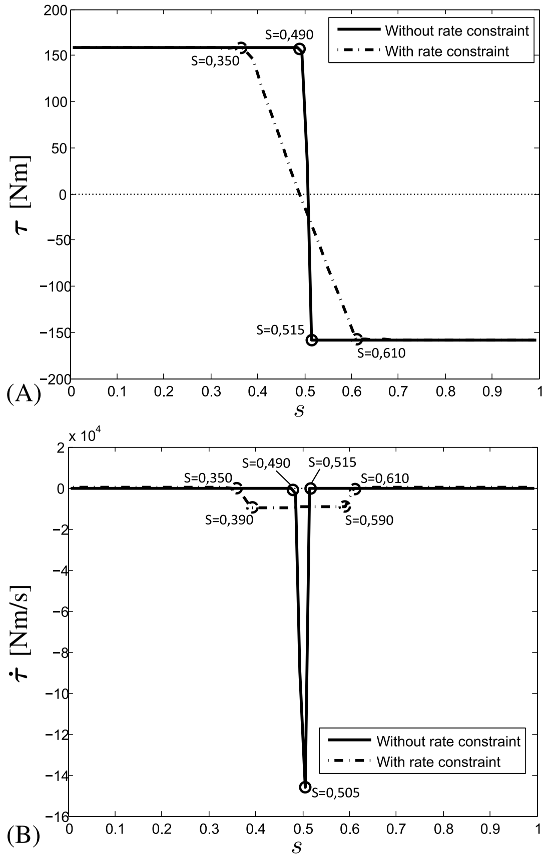


Fig. 6. Torque (A) and torque rate (B) of joint 1, with and without torque rate constraint.

Figure 6 shows the torque (Figure 6.A) and torque rate (Figure 6.B) of one joint of a seven degree of freedom robotic manipulator executing a line trajectory. The torque is constrained between -158 Nm and 158 Nm. The results of the optimization problem with torque rate constraint are shown in dashed lines and without in full lines. Without torque rate constraint the torque rate goes up to $1.5 \cdot 10^5$ Nm/s in contrast to a torque rate constraint of $1 \cdot 10^4$ Nm/s. A solution was obtained in five iterations. The inner convex problem is solved in ±1.5 s, which is similar to the results presented in [6].

D. Jerk constraints

Due to the bang-bang nature of time-optimal trajectories, the framework in [6] may also result in near-infinite jerks in joint space and operational (Cartesian) space. For systems with unmodeled flexibilities, this usually results in high acceleration peaks and unwanted oscillations. These vibrations can be reduced by imposing jerk constraints [21]. Jerks in joint space and Cartesian space are treated separately.

1) Joint jerk: By using the chain rule and (24), and by factoring out $\sqrt{b(s)}$, we can write the joint jerk constraint $\dddot{\boldsymbol{q}}_-(s) \leq \dddot{\boldsymbol{q}} = \frac{d^2\dot{\boldsymbol{q}}}{dt^2} \leq \dddot{\boldsymbol{q}}_+$ as in (11) with $\boldsymbol{\kappa} = -\frac{1}{2}\mathbf{1}_{2n}$,

$\boldsymbol{\rho}(s) = \mathbf{0}$ and

$$\boldsymbol{\phi}(s, b(s)) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \bigg(\boldsymbol{q}'''(s) b(s) + \tfrac{3}{2} \boldsymbol{q}''(s) b'(s) + \tfrac{1}{2} \boldsymbol{q}'(s) b''(s) \bigg),$$

$$\boldsymbol{\psi}(s) = \begin{pmatrix} \ddot{\boldsymbol{q}}_-(s) \\ -\ddot{\boldsymbol{q}}_+(s) \end{pmatrix}.$$

2) Cartesian jerk: The Cartesian jerk ${}^{v}_{f}\ddot{\boldsymbol{t}}^{o}_{w} = \frac{d^2\left({}^{v}_{f}\boldsymbol{t}^{o}_{w}\right)}{dt^2}$, the second derivative of the twist, represents the jerk, of the frame $\{o\}$ with respect to the reference point v, with respect to the frame $\{w\}$, expressed in the frame $\{f\}$. For notational simplicity we neglect the indices from here on. Similarly, by elaborating $\ddot{\boldsymbol{t}}$, using (1) and (24) and factoring out $\sqrt{b(s)}$, we can write a Cartesian jerk constraint as in (11) with $\boldsymbol{\kappa} = -\frac{1}{2}\mathbf{1}_{2n}$, $\boldsymbol{\rho}(s) = \mathbf{0}$ and

$$\boldsymbol{\phi}(s, b(s)) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \bigg(\left(J''(s)\boldsymbol{q}'(s) + J'(s)\boldsymbol{q}''(s) + J(s)\boldsymbol{q}'''(s)\right) b(s) + \tfrac{3}{2} \left(J'(s)\boldsymbol{q}'(s) + J(s)\boldsymbol{q}''(s)\right) b'(s) + \tfrac{1}{2} J(s)\boldsymbol{q}'(s) b''(s) \bigg),$$

$$\boldsymbol{\psi}(s) = \begin{pmatrix} \ddot{\boldsymbol{t}}_-(s) \\ -\ddot{\boldsymbol{t}}_+(s) \end{pmatrix}.$$

V. Combining constraints and physical effects

This section focusses on pairwise combinations of the constraints and physical effects resulting from the applications given in Section IV. In several cases both of the discussed physical effects lead to two independent constraints which can be stacked into a constraint of the form (11) (see Section V-A). In several other cases the physical effects from both applications influence each other leading to different constraints than when treated individually. These constraints either fit in the proposed form (11) (see SectionV-B) or not. In the latter case an efficient DC decomposition may be hard to find. E.g. when including a torque rate constraint (Section IV-C) in the optimization while considering viscous friction in the joints (Section IV-A2), the viscous friction term in the actuator torque (18) introduces extra terms in the torque rate constraint

$$\dot{\boldsymbol{\tau}}_- \leq \bigg(\boldsymbol{g}'(s) + \boldsymbol{c}'(s)b(s) + (B\boldsymbol{q}')'b(s)^{\frac{1}{2}} + (B\boldsymbol{q}')b'(s)b(s)^{-\frac{1}{2}} + \left(\boldsymbol{c}(s) + \tfrac{1}{2}\boldsymbol{m}'(s)\right)b'(s) + \tfrac{1}{2}\boldsymbol{m}(\boldsymbol{s})b''(s) \bigg) b(s)^{\frac{1}{2}} \leq \dot{\boldsymbol{\tau}}_+,$$

which is not of the form (11), hence, it does not fit into the scope of this paper.

Table I lists whether the combination of two applications leads to a constraint of the form (11) (indicated by ✓) or not (indicated by ✗). The constraint that follows from a specific application is indicated by its section letter from IV. From Table I it can also be concluded whether a combination of more than two constraints leads to a constraint of the form (11), e.g. we can combine A.1 with A.2, A.2 with D.1, and A.1 with D.1, hence we can combine A.1 with A.2 with D.1. Two combinations of DC constraints are illustrated below.

	A.1	A.2	B.1	B.2	C	D.1	D.2
A.1	-						
A.2	✓	-					
B.1	✗	✗	-				
B.2	✓	✓	✗	-			
C	✓	✗	✗	✗	-		
D.1	✓	✓	✓	✓	✓	-	
D.2	✓	✓	✓	✓	✓	✓	-

TABLE I
APPLICATION COMBINATION THAT LEADS TO A CONSTRAINT OF THE FORM (11) THAT ALLOWS FOR A SIMPLE DC DECOMPOSITION. THE TABLE LISTINGS REFER TO THE CORRESPONDING SUBSECTIONS OF SECTION IV.

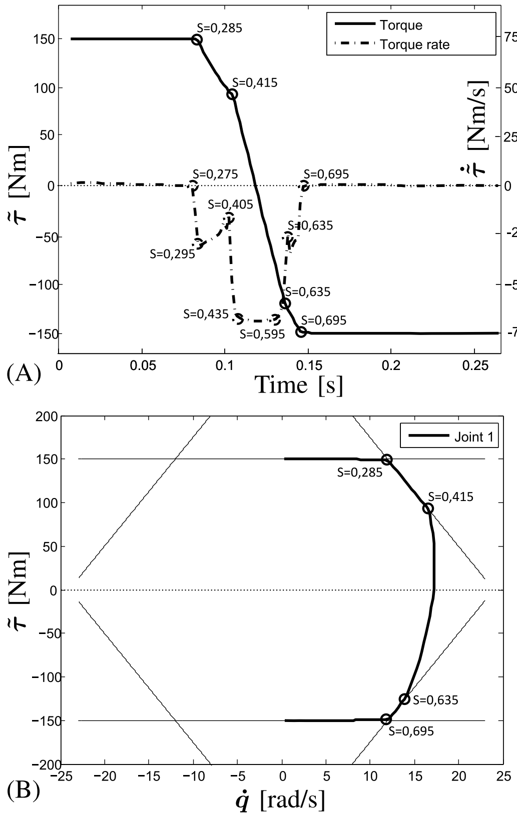


Fig. 7. Actuator torque as the result of the optimization in function of time (A) and joint speed (B), with torque rate constraint.

A. Actuator characteristic and torque rate constraints

In a typical application in robotics, we have velocity-dependent torque bounds due to the actuator voltage constraints and torque rate constraints due to the limited bandwidth of the actuators. Combining both constraints involves stacking the $\boldsymbol{\phi}(s, b(s))$, $\boldsymbol{\psi}(s)$, $\boldsymbol{\rho}(s)$ and $\boldsymbol{\kappa}$ from IV-A.1 and IV-C. This results in a constraint of the form (11) with

$$\boldsymbol{\phi}(s, b(s)) = \begin{pmatrix} \boldsymbol{\phi}_{A.1}(s, b(s)) \\ \boldsymbol{\phi}_{C}(s, b(s)) \end{pmatrix}, \ \boldsymbol{\psi}(s) = \begin{pmatrix} \boldsymbol{\psi}_{A.1}(s) \\ \boldsymbol{\psi}_{C}(s) \end{pmatrix},$$

$$\boldsymbol{\rho}(s) = \begin{pmatrix} \boldsymbol{\rho}_{A.1}(s) \\ \boldsymbol{\rho}_{C}(s) \end{pmatrix} \text{ and } \boldsymbol{\kappa} = \begin{pmatrix} \boldsymbol{\kappa}_{A.1} \\ \boldsymbol{\kappa}_{C} \end{pmatrix}$$

The example of IV-A is revisited with an added torque rate constraint (±7000 Nm/s). To illustrate that both constraints

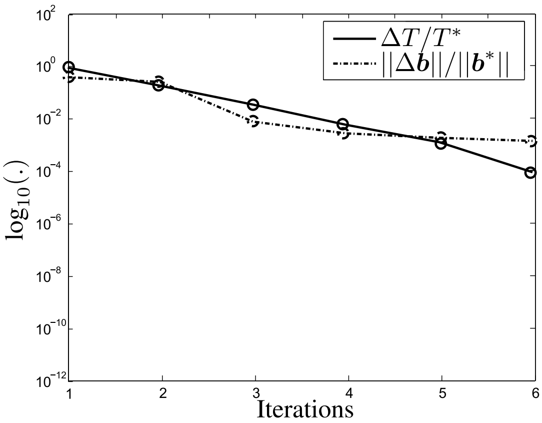


Fig. 8. Convergence of the SCP algorithm.

become active, the voltage constraints are chosen slightly differently. The results for the first joint are given in figure 7. Figure 7.A shows the actuator torque and torque rate as a function of time. Figure 7.B illustrates the actuator torque as a function of the joint speed and visualizes the velocity dependent torque constraints. Figure 8 illustrates the convergence of the SCP algorithm. Here the normalised optimal motion time ($\Delta T/T^{*}$ with $\Delta T = T^{k+1} - T^{k}$ and T^{*} the solution of the SCP algorithm) and the normalised residual ($||\Delta \boldsymbol{b}||/||\boldsymbol{b}^{*}||$ with $\Delta \boldsymbol{b} = \boldsymbol{b}^{k+1} - \boldsymbol{b}^{k}$) are shown for each iteration.

A solution with sufficient precision ($\varepsilon = 10^{-4}$) was obtained in six iterations. The inner convex problem is solved in ± 2 s. Compared to the example of IV-A more iterations are needed. As to be expected for a time-optimal control problem, the motion is of the bang-bang type. This can be seen from Figure 7, since for all s, always one constraint is active (neglecting discretisation effects). From $s = 0$ to $s = 0.285$ and from $s = 0.695$ to $s = 1$ the torque (current) constraint is active, from $s = 0.285$ to $s = 0.415$ and from $s = 0.635$ to $s = 0.695$ the voltage constraint is active and from $s = 0.435$ to $s = 0.595$ the torque rate constraint is active.

B. Actuator characteristic and viscous friction in the joints

Other applications involve viscous friction in the joints and velocity-dependent torque bounds due to an actuator voltage constraint. Including both in the optimization problem leads to a closed polytope for every joint of the form (13), as can be seen in Figure 9.B, with

$$\boldsymbol{f}^{i}(s) = \begin{pmatrix} 1 \\ -1 \\ \boldsymbol{f}_{v}^{i} \end{pmatrix}, \; \boldsymbol{h}^{i}(s) = \begin{pmatrix} 0 \\ 0 \\ \boldsymbol{h}_{v}^{i} \end{pmatrix} + B_{ii}(s)\boldsymbol{f}^{i}(s)$$

$$\text{and } \boldsymbol{p}^{i}(s) = \begin{pmatrix} \boldsymbol{\tau}_{+,i}(s) \\ -\boldsymbol{\tau}_{-,i}(s) \\ \boldsymbol{p}_{v}^{i} \end{pmatrix}.$$

The example of IV-A is revisited with viscous friction in the joints (1 N m s/rad). The results for joint 1 and 2 are given in figure 9. Figure 9.A shows the actuator torque and torque rate as a function of time. Figure 7.B illustrates the actuator torque as a function of the joint speed, where the polytope is also shown in thin lines. A solution with sufficient precision ($\varepsilon = 10^{-4}$) is obtained in five iterations. The inner convex problem is solved in ± 1.5 s.

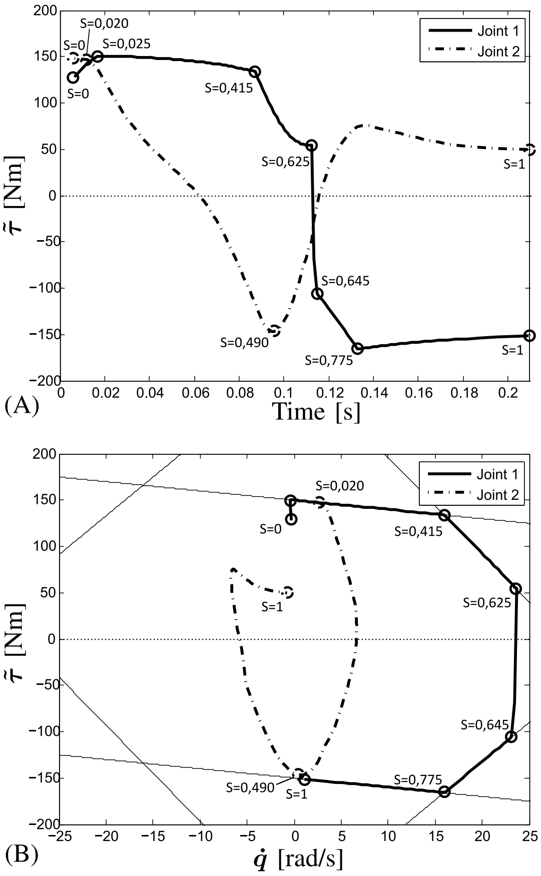


Fig. 9. Actuator torque, as the result of the optimization, in function of time (A) and joint speed (B).

VI. Discussion

The SCP approach given in Section III 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)-(8) is attained. Furthermore, the solution is found with high accuracy in only a few iterations, which is important from a practical point of view. All numerical examples given here were performed off-line, but thanks to its high efficiency, the proposed SCP approach has potential to be used in on-line applications similar to [7].

Furthermore, the SCP framework proposed in this paper counters the major downsides of the convex framework [6]. In [6] the torque rates, velocity-dependence of the actuator torque bounds and interaction with the environment, which cannot be neglected in practical applications, cannot be included due to the non-convex nature of the resulting constraints. In the proposed SCP approach we can easily include these constraints, as shown in the previous sections. Regarding the velocity-dependence of the actuator torque bounds, the proposed approach results in an optimal use of the actuators, since the torque characteristic is not approximated by a conservative

link	α	a	θ	d
1	0	0	0	0.31
2	π/2	0	0	0
3	-π/2	0	0	0.4
4	-π/2	0	0	0
5	π/2	0	0	0.39
6	π/2	0	0	0
7	-π/2	0	0	0

TABLE II
Denavit-Hartenberg coordinates for each joint.

link	(I_{xx}, I_{yy}, I_{zz}) [kg m^2]	(r_x, r_y, r_z) [m]	$\tau_{+,-}$ [Nm]
1	(0, 0, 0.0115)	(0, 0, 0)	± 158
2	(-0.5472, 0, -0.54)	(0, -0.3, -0.0039)	± 158
3	(0.0064, 0, 0.0108)	(0, -0.0016, 0)	± 90
4	(-1.044, 0, -1.040)	(0, 0.5217, 0)	± 90
5	(0.0037, 0, 0.006)	(0, 0.012, 0)	± 90
6	(0.001, 0, 0.0036)	(0, 0.0081, 0)	± 27
7	(0.001, 0, 0.12)·10^{-3}	(0, 0, 0)	± 27

TABLE III
Dynamic parameters for each link

convex characteristic as in [15], or the torque bounds are not chosen conservatively as in [6]. Hence, the actuators are used at full capacity. The downside of the proposed approach is a larger computational time by a factor of three to six due to the SCP iterations.

Also, the SCP approach can easily be implemented. The algorithm can be built around the convex optimal path following framework from [6] with minimal extra effort. The DC decomposition (11) involves only a few lines of code.

In this paper, we have discussed only one DC decomposition for one particular form of non-convex constraints. In general, one can find multiple DC decompositions. From the given numerical simulations, we can conclude that the presented DC decomposition results in an efficient algorithm.

A drawback of the proposed SCP method is that not all items in Table I are checked. Thus, not all applications can be combined into a constraint of the form (11). Although in some cases we can still make an approximation that results in a simple DC decomposition as in (11), for example, by neglecting viscous friction effects in torque rate constraints while including them in the torque constraints. Furthermore, e.g. in the case of viscous friction effects in torque rate constraints (see Section V), it is possible to find a DC decomposition although it does not fit into the form we discuss in this paper.

As a general conclusion, the proposed SCP method with DC constraints offers an efficient solution to many practical non-convex optimal path following problems for robotic manipulators.

Appendix

Dynamic model of the robot: In this appendix we include the dynamic model of the considered robot. The kinematics of the robot are described using Denavit-Hartenberg coordinates [13]. Table II and III give the Denavit-Hartenberg coordinates and dynamic parameters, respectively, for each joint. Here I_{xx}, I_{yy} and I_{zz} represent the inertia [13] of each link, r_x, r_y and r_z represent the location of the center of gravity in the link frame [13] of each link. The mass m of each link is assumed to be 2 kg for each link.

Voltage bound: The used voltage bound in the numerical simulations are given by the following vectors for each joint i:

$$\boldsymbol{f}_v^i = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}, \boldsymbol{h}_v^i = \begin{pmatrix} -\frac{1}{16}\tilde{\boldsymbol{\tau}}_{+,i} \\ \frac{1}{16}\tilde{\boldsymbol{\tau}}_{+,i} \\ \frac{1}{16}\tilde{\boldsymbol{\tau}}_{-,i} \\ -\frac{1}{16}\tilde{\boldsymbol{\tau}}_{-,i} \end{pmatrix}, \boldsymbol{p}_v^i = \begin{pmatrix} 2\tilde{\boldsymbol{\tau}}_{+,i} \\ 2\tilde{\boldsymbol{\tau}}_{+,i} \\ 2\tilde{\boldsymbol{\tau}}_{-,i} \\ 2\tilde{\boldsymbol{\tau}}_{-,i} \end{pmatrix}.$$

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, ERC HIGHWIND (259 166), project IWT-SBO 80032 (LECOPRO) of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen), project G.0377.09 of the Research Foundation Flanders (FWO Vlaanderen), and K.U.Leuvens Concerted Research Action GOA/10/11. Goele Pipeleers is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen).

References

[1] 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.

[2] 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.

[3] W. Van Loock, S. Bellens, G. Pipeleers, J. De Schutter, and J. Swevers, "Time-optimal parking and flying: Solving path following problems efficiently," in *IEEE International Conference on Mechatronics, Vicenza, 27-28 February, 1 March*, 2013.

[4] O. von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of Operations Research*, vol. 37, pp. 357–373, 1992.

[5] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast Direct Multiple Shooting Algorithms for Optimal Robot Control," in *Fast Motions in Biomechanics and Robotics*, (Heidelberg, Allemagne), 2005.

[6] D. Verscheure, 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.

[7] D. Verscheure, 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.

[8] R. Horst and N. Thoai, "Dc programming: Overview," *J. Optim. Theory Appl.*, vol. 103(1), pp. 1–43, 1999.

[9] 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.

[10] Q. Tran-Dinh, C. Savorgnan, and M. Diehl, "Adjoint-based predictor-corrector sequential convex programming for parametric nonlinear optimization.," *SIAM J. Optimization*, (accepted).

[11] 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*, vol. 57, no. 6, pp. 1377–1390, 2012.

[12] B. K. Sriperumbudur and G. R. G. Lanckriet, "On the convergence of the concave-convex procedure," in *NIPS'09*, pp. 1759–1767, 2009.

[13] M. W. Spong, *Robot Dynamics and Control*. New York, NY, USA: John Wiley & Sons, Inc., 1st ed., 1989.

[14] 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.

[15] T. Ardeshiri, M. Norrlöf, J. Löfberg, and A. Hansson, "Convex optimization approach for time-optimal path tracking of robots with speed dependent constraints," in *Preprints of the 18th IFAC World Congress*, pp. 14648–14653, 2011.

[16] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, (Taipei, Taiwan), 2004.

[17] H. W. Beaty and J. L. Kirtley, *Electric Motor Handbook*. McGraw-Hill Handbooks. McGraw-Hill Book Company, 1998.

[18] J. P. Kruth, *Productietechnieken en -systemen*. ACCO Leuven, Belgium (Dutch), 2012.

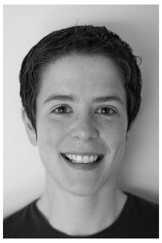
[19] D. Constantinescu and E. A. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *Journal of Robotic Systems*, vol. 17, no. 5, pp. 233–249, 2000.

[20] S. Macfarlane and E. Croft, "Jerk-bounded manipulator trajectory planning: design for real-time applications," *Robotics and Automation, IEEE Transactions on*, vol. 19, pp. 42 – 52, feb 2003.

[21] K. J. Kyriakopoulos and Saridis, "Minimum jerk path generation," in *Proc. IEEE International Conference Robotics and Automation*, pp. 364–369, 1988.

Frederik Debrouwere received the M.Sc. degree in mechanical engineering, in 2011, from the Katholieke Universiteit Leuven (KU Leuven), Belgium, where he is currently working towards the Ph.D. degree. His research interests include optimal motion control of mechatronic systems with a focus on robotic manipulators.

Wannes Van Loock received the M.Sc. degree in mechanical engineering, in 2009, from the Katholieke Universiteit Leuven (KU Leuven), Belgium, where he is currently working towards the Ph.D. degree. His research interests include optimal motion control of mechatronic systems.



Goele Pipeleers received her M.Sc. degree in mechanical engineering and her Ph.D. degree in mechanical engineering from the Katholieke Universiteit Leuven, Belgium, in 2004 and 2009, respectively. Currently she is a Post-doctoral Fellow of the Research Foundation-Flanders affiliated with the KU Leuven, Department of Mechanical Engineering, Division of Production Engineering, Machine Design and Automation (PMA). She has been a visiting scholar at the Colorado School of Mines and at the University of California Los Angeles. Her research interests include convex optimization, optimal and robust control, and their applications in mechatronics.

Quoc Tran Dinh obtained his B.S. degree in Applied Mathematics and Informatics, and M.S. degree in Computer Science from Vietnam National University, Hanoi, in 2001 and 2004, respectively. In 2012, he received the Ph.D degree in Electrical Engineering at the Department of Electrical Engineering and Optimization in Engineering Center (OPTEC), Katholieke Universiteit Leuven, under the supervision of Prof. Moritz Diehl. Currently, he is a postdoctoral researcher at the Laboratory for Information and Inference System, Ecole Polytechnique Federale de Lausanne, Switzerland. His research focuses on methods for convex optimization, sequential convex programming, parametric optimization, optimization in machine learning, and methods for variational inequality and equilibrium problems.

Moritz Diehl received the Ph.D. degree from the Interdisciplinary Center for Scientic Computing (IWR), Heidelberg University, Heidelberg, Germany, in 2001. Since 2006, he has been a Professor with the University of Leuven (K.U. Leuven), Belgium, and Principal Investigator of K.U. Leuvens Optimization in Engineering Center OPTEC. His research is centered around embedded optimization algorithms for use in model predictive control, real-time optimization, and moving horizon estimation. His general interests are in structure exploitation for optimization in engineering, convex optimization, dynamic optimization. He works on real-world applications of optimization and control in mechatronics, robotics, sustainable energy, and chemical engineering.

Joris De Schutter received the M.Sc. degree in mechanical engineering from the Katholieke Universiteit Leuven, Belgium, in 1980, the M.Sc. degree from the Massachusetts Institute of Technology, in 1981, and the Ph.D. degree in mechanical engineering, also from KU Leuven, in 1986. Following work as a control systems engineer in industry, in 1986, he became a lecturer in the Department of Mechanical Engineering, KU Leuven, where he has been a full professor since 1995. He teaches courses in kinematics and dynamics of machinery, control, robotics and optimization. His research interests include sensor-based robot control and programming, optimal motion control of mechatronic systems, and modeling and simulation of human motion.

Jan Swevers received his M.Sc. degree in electrical engineering and the Ph.D. degree in mechanical engineering from the Katholieke Universiteit Leuven (KU Leuven), Belgium, in 1986 and 1992, respectively. He is since 1995 a professor in the Department of Mechanical Engineering, Division Production Engineering, Machine Design and Automation (PMA), of KU Leuven. His research interests include modeling, identification, control and optimization of mechatronics systems. http://www.mech.kuleuven.be/meco