

# Time-scaling of trajectories for point-to-point robotic tasks

Javier Moreno-Valenzuela\*

Centro de Investigación y Desarrollo de Tecnología Digital, CITEDI-IPN, Av. del Parque 1310, Mesa de Otay,  
Tijuana, B.C., 22510, Mexico Electronic mail: moreno@citedi.mx

(Received 15 September 2004; accepted 14 September 2005)

---

## Abstract

In order to improve the tracking performance in the practical robot task of time-optimal point-to-point motion when torque limits and model uncertainty are present, in this paper is introduced an algorithm for on-line time-scaling of the desired trajectories, which is used along with a conventional trajectory tracking controller. Numerical examples for a two-degrees-of-freedom arm are provided, which illustrate the tracking accuracy of the proposed method. © 2006 ISA—The Instrumentation, Systems, and Automation Society.

*Keywords:* Point-to-point motion; Constrained input; Robot manipulator; Stability; Time-scaling

---

## 1. Introduction

Point-to-point tasks are often found in many robotic applications, such as parts handling, spot welding, and pick and place duties. In practice, it is important that the manipulator performs the point-to-point motion in minimum transversal time, thus reducing the production cost. At this stage, the nonlinear constraint of limited torque input should be used for planning the desired trajectory.

In practice, the robot motion is specified according to physical constraints, such as limited torque input. Thus, the computation of the desired trajectory is constrained to attain the physical limit of actuator saturations. See, e.g., [1,2] for a review of trajectory planning algorithms considering the robot model parameters and torque saturation. Besides, trajectories can also be planned irrespective of the estimated robot model, i.e., simply by using constraints of position, velocity, and acceleration at each time instant, see, e.g., [3,4]. Once the ref-

erence trajectory is specified, the task execution is achieved in real time by using a trajectory tracking controller. The trajectory tracking, control of manipulators has been well studied in the past 20 years; see, e.g., Abdallah *et al.* [5], Lewis *et al.* [6], Sciavicco and Siciliano [7], and references therein. Trajectory tracking control in joint space is specified through trajectories of the joint position depending on time, which encode the requested task to be performed by a robot arm. This way of formulating the robot manipulator control has given good results for many practical applications. Notwithstanding, if parametric errors in the model estimation are present, and considering that the desired trajectories require using the maximum torque, no margin to suppress the tracking error will be available. As a consequence, the manipulator deviates from the desired trajectory and poor execution of the task is obtained.

Slowing down of the desired trajectories has been an alternative to solve the problem of trajectory tracking control considering constrained torques and model uncertainties. Nevertheless, independently of the problem of the model distur-

---

\*E-mail: moreno@citedi.mx

bances, in the literature we can distinguish *off-line* and *on-line* algorithms to scale in time the desired nominal trajectories so that the torque input generated by the controller remains within the physical limits of the actuators.

Usually the *off-line* methods for time-scaling require knowing *a priori* the exact robot dynamics. Using this assumption, Hollerbach [8] introduced a technique for time-scaling of *off-line* planned trajectories. The main feature of this method is the introduction of a scaling factor to determine whether a planned trajectory is dynamically realizable given actuator torque limits and to bring it to one that is realizable. The method does not consider the practical situation in that only a rough estimation of the robot model is available, which is its main disadvantage. It is noteworthy that this approach has been extended to the cases of multiple robots in cooperative tasks [9] and robot manipulators with elastic joints [10].

On the other hand, to obviate the drawback of the assumption that the robot model must be exactly known, *on-line* methods for time-scaling of the desired trajectories have been developed. Namely, Dahl and Nielsen [11] proposed a control algorithm which consists in the tracking of a time-scaled trajectory obtained from a specified geometric path and a modified on-line *velocity profile*. The method considers an inner loop that slows down the generated reference trajectory when the torque input is saturated. In the work proposed by Arai *et al.* [12], a feedback control method for path tracking which takes the torque saturation into account is introduced. The method is based on controlling the components of a new coordinated system called path coordinates. However, this method uses the same internal feedback proposed by [11], which is based on the specification of a velocity profile function.

Motivated by the work of Hollerbach [8], i.e., the introduction in an explicit way of a time-scaling factor in the trajectory planning stage, we present in this paper a new optics for tracking control of robot manipulators under the practical situation-limited torques and model uncertainties. More specifically, we introduce an algorithm to complete time-optimal point-to-point tasks, which consists in using a trajectory tracking controller and an algorithm for on-line modification of the reference trajectories according to tracking errors and torque/acceleration saturation. The trajectory

modification is done by controlling this time-scaling factor, whose time evolution responds to the tracking errors and torque limits. The new method does not require the specification of a velocity profile, as in [11,12]. The proposed approach can be applied to perform a wide class of robotic tasks such as laser cutting, arc welding, glue dispensing, material removing from conveyor belts, and other tasks where accuracy is required to reach forward motion along the geometric path. Numerical simulations are provided to illustrate the feasibility of the proposed methodology.

It is worthwhile to notice that an important difference between the proposed approach and the one introduced by Hollerbach [8] is that the former is an *on-line* method while the latter is an *off-line* technique. Besides, the proposed scheme only requires a very simple estimation of the robot model, which is a significant advantage with respect to the assumption that the robot model must be exactly known as required in [8]. Due to the use of a time-scaling factor, the methodology proposed in this paper can be interpreted as the robust extension of the algorithm introduced by Hollerbach [8].

This paper is organized as follows. Robot modeling is discussed in Section 2. Section 3 is devoted to discussing the planning of a time-optimal point-to-point motion. In Section 4, we discuss the trajectory tracking approach to robot control. An algorithm for dynamic time-scaling of trajectories is proposed in Section 5. Simulation results are presented in Section 6. Finally, in Section 7 some concluding remarks are given.

## 2. Robot modeling

The dynamics in joint space of a serial-chain  $n$ -link robot manipulator, considering the presence of friction at the robot joints, can be written as [7]

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\dot{\mathbf{q}}) = \boldsymbol{\tau}, \quad (1)$$

where  $\mathbf{q}$  is the  $n \times 1$  vector of joint displacements,  $\dot{\mathbf{q}}$  is the  $n \times 1$  vector of joint velocities,  $\boldsymbol{\tau}$  is the  $n \times 1$  vector of applied torque inputs,  $M(\mathbf{q})$  is the  $n \times n$  symmetric positive-definite inertia matrix,  $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$  is the  $n \times 1$  vector of centripetal and Coriolis torques,  $\mathbf{g}(\mathbf{q})$  is the  $n \times 1$  vector of gravitational torques, and  $\mathbf{f}(\dot{\mathbf{q}})$  is the vector of forces and torques due to the viscous and Coulomb friction, which, for the  $i$ th element, satisfies

$$|f_i(\dot{q})| \leq f_{Ci} + f_{vi}|\dot{q}_i|,$$

where  $f_{Ci}$  and  $f_{vi}$  are positive constants, and  $i = 1, \dots, n$ .

Let  $\mathcal{T}$  denotes the torque space, defined as

$$\mathcal{T} = \{\tau \in \mathbb{R}^n: \tau_i^{\min} < \tau_i < \tau_i^{\max}\}, \quad i = 1, \dots, n, \quad (2)$$

with  $\tau_i^{\max}$  the maximum torque input and  $\tau_i^{\min}$  the minimum torque input for the  $i$ th joint. It is assumed that

$$f_{Ci} + \bar{k}_{gi} < \tau_i^{\max}, \quad i = 1, \dots, n, \quad (3)$$

where

$$|g_i(q)| < \bar{k}_{gi},$$

$g_i(q)$  is the  $i$ th element of the vector  $g(q)$ . Assumption (3) ensures that any joint position  $q$  is reachable.

In general, taking into account the nonlinear robot model (1) for control and motion planning may be a difficult task. See, e.g., [1] for a reference on time-optimal trajectory planning algorithms considering the nonlinear model (1) subject to constrained torques  $\tau \in \mathcal{T}$ .

On the other hand, carrying out an estimation of the nonlinear robot model (1) can be demanding work, especially for high degree-of-freedom robots. Thus, a very simple estimation of the robot model (1) is the decoupled second-order plant

$$\hat{m}_i \ddot{q}_i = \tau_i, \quad i = 1, \dots, n, \quad (4)$$

where  $\hat{m}_i$  denotes an estimation of the inertia corresponding to the  $i$ th link, and  $q_i$  and  $\tau_i$  are the robot joint position and torque input, respectively. In this paper, model (4) is referred to as the *nominal robot model*. In practice, designing a motion controller and motion trajectories for the nominal model (4) is easier than for the general nonlinear model (1), because we are interpreting the robot system as a set of decoupled integrators.

### 3. Planning of point-to-point motions

With an academic point of view, in this section we provide the explicit derivation of a time-optimal class of point-to-point motions. The studied class of motions is encoded by the path

$$q_d(s) = \begin{bmatrix} a_1 s + b_1 \\ \vdots \\ a_n s + b_n \end{bmatrix}, \quad s \in [s_0, s_f], \quad (5)$$

where  $0 < s_0 < s_f$ ,  $a_1, \dots, a_n$ , and  $b_1, \dots, b_n$  are constants, and  $s$  is called the path parameter. Let us notice that any path of the form (5) can be planned using  $0 < s_0 < s_f$ . This assumption will be required in the proposed algorithm of time-scaling of trajectories. The task is completely determined specifying the path parameter  $s$  as a function of time. The function  $s(t)$  must satisfy either a cost function or restrictions in position, velocity, and acceleration at the beginning and end of the motion. We are interested in minimal time execution—cost function—subject to limited torque input.

The optimal point-to-point motion problem can be stated as follows: *Minimize the performance measure—cost function—*

$$J = \int_{t_0}^{t_f} 1 dt = t_f - t_0,$$

where  $t_0$  is the initial time and  $t_f$  is the final time, subject to the nominal robot dynamics (4), the position constraints

$$q_i = q_{di}(s) = a_i s + b_i, \quad s_0 \leq s \leq s_f,$$

the boundary conditions

$$\begin{aligned} s(t_0) &= s_0, & s(t_f) &= s_f, \\ \dot{s}(t_0) &= 0, & \dot{s}(t_f) &= 0, \end{aligned} \quad (6)$$

and the actuator constraint  $\tau_i \in \mathcal{T}$ , where the torque space  $\mathcal{T}$  has been defined in (2),

$$\mathcal{T} = \{\tau_i \in \mathbb{R}: \tau_i^{\min} < \tau_i < \tau_i^{\max}\}, \quad i = 1, \dots, n,$$

A physical meaning of the boundary conditions (6) in the optimal point-to-point motion problem stated above is that the robot should start and finish the motion with velocity zero. Hence a *time-optimal control* in the torque space (2), i.e.,  $\tau_i \in \mathcal{T}$ , should drive the robot dynamics (4) from the initial state

$$\begin{bmatrix} q_i(t_0) \\ \dot{q}_i(t_0) \end{bmatrix} = \begin{bmatrix} a_i s(t_0) + b_i \\ a_i \dot{s}(t_0) \end{bmatrix} \quad (7)$$

to the final state

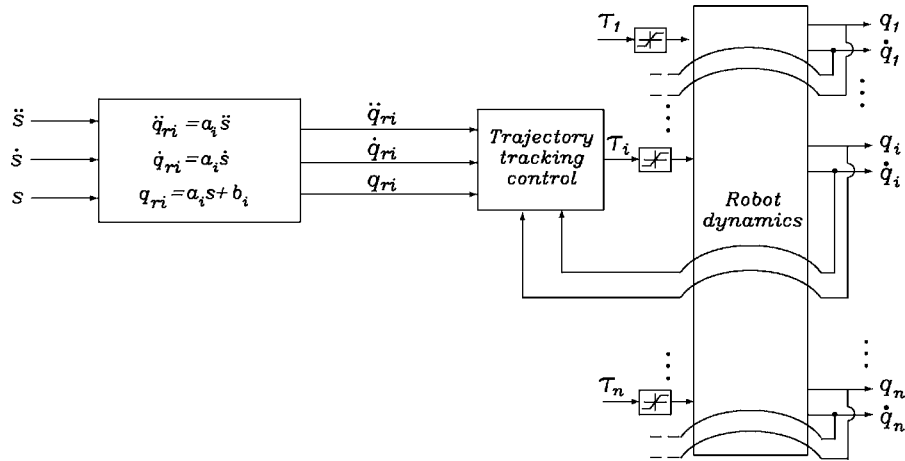


Fig. 1. Block diagram of the trajectory tracking control.

$$\begin{bmatrix} q_i(t_f) \\ \dot{q}_i(t_f) \end{bmatrix} = \begin{bmatrix} a_i s(t_f) + b_i \\ a_i \dot{s}(t_f) \end{bmatrix} \quad (8)$$

in minimum transversal time  $J=t_f-t_0$ .

Using the path (5), a minimum-time trajectory  $q_d[s(t)]$  can be planned when the joint actuator torque is limited. In order to show that, let us notice that the robot dynamics (4) is constrained to the position specified by the path (5), whereby we can write

$$\hat{m}_i a_i \ddot{s} = \tau_i. \quad (9)$$

Therefore, the admissible values of  $\ddot{s}$  that satisfy the time-optimal criterion can be computed by finding the maximum and minimum admissible  $\ddot{s}$  on the torque space (2). Thus, the path acceleration  $\ddot{s}$  is constrained by

$$\frac{\tau_i^{\min}}{\hat{m}_i a_i} \leq \ddot{s} \leq \frac{\tau_i^{\max}}{\hat{m}_i a_i}, \quad i = 1, \dots, n. \quad (10)$$

Moreover, we can write system (9) in state variables

$$\frac{d}{dt} \begin{bmatrix} s \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 \\ \ddot{s} \end{bmatrix}, \quad (11)$$

where  $\ddot{s}$  should satisfy inequality (10). By invoking the Pontryagin maximum principle [13], it is possible to show that for system (11)—subject to the boundary constraints (6)—the transversal time  $J=t_0-t_f$  is minimized using

$$\ddot{s} = \begin{cases} \ddot{s}^{\max}, & t_0 \leq t \leq t_s, \\ \ddot{s}^{\min}, & t_s < t \leq t_f, \end{cases} \quad (12)$$

where  $t_s$  is the switching time, and

$$\ddot{s}^{\max} = \min_i \left\{ \frac{\tau_i^{\max}}{\hat{m}_i a_i} \right\}, \quad (13)$$

$$\ddot{s}^{\min} = \max_i \left\{ \frac{\tau_i^{\min}}{\hat{m}_i a_i} \right\}. \quad (14)$$

It is worthwhile to note that input  $\ddot{s}$  in (12) for system (11) satisfies the constraint (10).

Thus, the controls in the torque space (2) that solve the optimal point-to-point problem stated before are obtained from Eqs. (9) and (12),

$$\tau_i = \begin{cases} \tau_i^{\sup} & \text{for all } t_0 \leq t \leq t_s, \\ \tau_i^{\inf} & \text{for all } t_s < t \leq t_f, \end{cases} \quad (15)$$

where  $\tau_i^{\sup}$  and  $\tau_i^{\inf}$  are defined as follows:

$$\tau_i^{\sup} = \hat{m}_i a_i \left[ \min_j \left\{ \frac{\tau_j^{\max}}{\hat{m}_j a_j} \right\} \right],$$

$$\tau_i^{\inf} = \hat{m}_i a_i \left[ \max_j \left\{ \frac{\tau_j^{\min}}{\hat{m}_j a_j} \right\} \right],$$

with  $j=1, \dots, n$ , and  $t_s$  is the switching time.

Using controls (12) in the system (11), we can find the time evolution of the path parameter  $s(t)$ ,

$$s(t) = \begin{cases} \ddot{s}_{\max} \frac{[t-t_0]^2}{2} + s_0, & t_0 \leq t \leq t_s, \\ \ddot{s}_{\min} \frac{[t-t_s]^2}{2} + \dot{s}_s[t-t_s] + s_s, & t_s < t \leq t_f, \end{cases} \quad (16)$$

where  $s_s = s(t_s)$  and  $\dot{s}_s = \dot{s}(t_s)$ .

With boundary conditions (6) and Eq. (16), it is possible to obtain the following nonlinear equation system:

$$s_f = \ddot{s}_{\min} \frac{[t_f-t_s]^2}{2} + \dot{s}_s[t_f-t_s] + s_s, \quad (17)$$

$$\dot{s}_s = \ddot{s}_{\max}[t_s-t_0], \quad (18)$$

$$0 = \ddot{s}_{\min}[t_f-t_s] + \dot{s}_s, \quad (19)$$

$$s_s = \ddot{s}_{\max} \frac{[t_s-t_0]^2}{2} + s_0. \quad (20)$$

By solving the equation system (17)–(20), it is possible to obtain the numerical values of  $t_s$ ,  $t_f$ ,  $s(t_s) = s_s$ , and  $\dot{s}(t_s) = \dot{s}_s$ .

In this way, applying the time-optimal controls (15) into the system (4), we find that  $q_i(t) = q_{ri}(t)$ , where

$$q_{ri}(t) = a_i s(t) + b_i, \quad (21)$$

with  $s(t)$  given by (16). The function  $q_{ri}(t)$  in (21) is called the nominal trajectory. In conclusion, the path (5) and path parameter  $s(t)$  in (16) encode the task of time-optimal point-to-point motion under torque input constraints.

The realization of the point-to-point task encoded by the desired path  $q_d(t)$  in (5) and the

time-optimal path parameter  $s(t)$  in (16) is carried out using a trajectory tracking controller. In the next section this subject is discussed.

#### 4. Trajectory tracking control

Trajectory tracking control is the classical approach to solving a point-to-point task specified by the nominal trajectory (21). Let us consider the following trajectory tracking controller:

$$\tau_i = \hat{m}_i \ddot{q}_{ri} + k_{vi} \dot{e}_{pi} + k_{pi} e_{pi}, \quad (22)$$

where  $i=1, \dots, n$ ,  $k_{vi}$  and  $k_{pi}$  are positive constants, and

$$e_{pi} = q_{ri} - q_i = a_i s + b_i - q_i,$$

$$\dot{e}_{pi} = \dot{q}_{ri} - \dot{q}_i = a_i \dot{s} - \dot{q}_i.$$

Fig. 1 shows a block diagram of the decentralized implementation of the trajectory tracking controller (22).

Let us suppose that no torque saturation is present and the robot dynamics is governed by a set of  $n$  decoupled integrators, i.e., by the nominal model (4). Then, the control law (22) leads to the closed-loop system,

$$\hat{m}_i \ddot{e}_{pi} + k_{vi} \dot{e}_{pi} + k_{pi} e_{pi} = 0, \quad i = 1, \dots, n, \quad (23)$$

which is an exponentially stable system because  $k_{vi}, k_{pi} > 0$ . However, Eq. (23) does not represent the real closed-loop system because the control law (22) is only based on the estimation of the link inertia  $\hat{m}_i$ , hence the uncompensated dynamics can be interpreted as disturbances. By using the robot dynamics (1), the robust trajectory tracking control system (22) leads to the system

$$\frac{d}{dt} \begin{bmatrix} \mathbf{e}_p \\ \dot{\mathbf{e}}_p \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{e}}_p \\ \mathbf{M}(\mathbf{q})^{-1} [-K_v \dot{\mathbf{e}}_p - K_p \mathbf{e}_p + [\mathbf{M}(\mathbf{q}) - \hat{\mathbf{M}}] \ddot{\mathbf{q}}_r(t) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\dot{\mathbf{q}})] \end{bmatrix}, \quad (24)$$

where  $\mathbf{e}_p = [e_{p1}, \dots, e_{pn}]^T$ ,  $\mathbf{q}_r(t) = \mathbf{q}_d[s(t)] = \{q_{d1}[s(t)], \dots, q_{dn}[s(t)]\}^T$ ,  $\hat{\mathbf{M}} = \text{diag}\{\hat{m}_1, \dots, \hat{m}_n\}$ ,  $K_v = \text{diag}\{k_{v1}, \dots, k_{vn}\}$ , and  $K_p = \text{diag}\{k_{p1}, \dots, k_{pn}\}$ . Uniform boundedness of the signals  $[\mathbf{e}_p(t)^T \dot{\mathbf{e}}_p(t)^T]^T$  can be shown follow-

ing the results described in either Chap. 4 in [6] or Theorem 1 in [14].

Notwithstanding, the torque provided by the actuators is always limited and time-optimal robotic tasks are given in terms of planned trajectories that

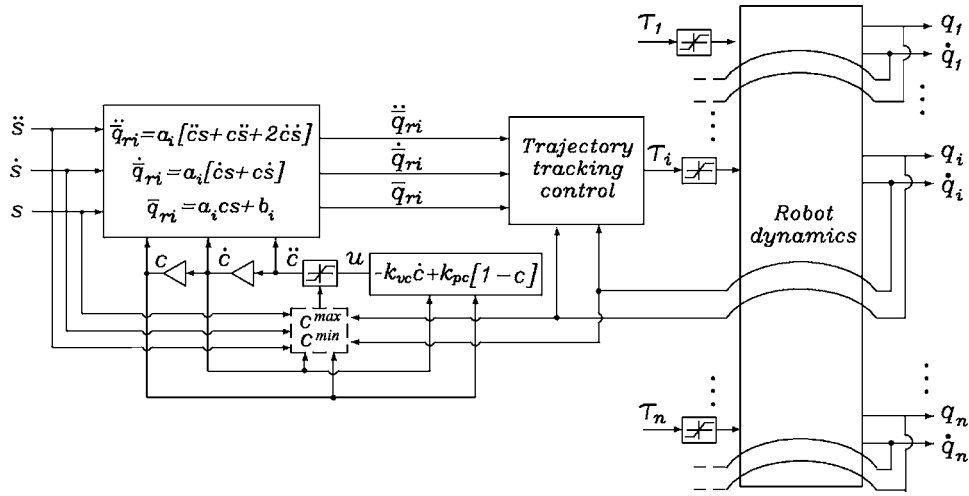


Fig. 2. Block diagram of the proposed trajectory time-scaling algorithm.

use the maximum available torque. Therefore, there is no room for extra torque of the feedback control action for compensating the model disturbances, and large tracking errors would be present. In consequence, in the condition of saturated torque, the closed-loop system (24) is no longer valid.

An approach of dynamic time-scaling of trajectories can be used to keep the produced torques in the physical actuator limits and to improve the tracking accuracy.

## 5. Dynamic time-scaling of trajectories

Let us define a new path parameter in the following way:

$$\sigma(t) = c(t)s(t), \quad (25)$$

where  $c(t)$  is a scalar function called the *time-scaling factor*. The desired timed trajectory is obtained using the new path parameter (25) into the path (5) as a function of time,

$$\bar{q}_{ri}(t) = q_{di}[\sigma(t)] = q_{di}[c(t)s(t)]. \quad (26)$$

The signal  $c(t)$  in (26) is a time-scaling factor of the nominal trajectory  $q_{ri}(t)$  in (21). The introduction of a time-scaling factor  $c(t)$  is also used in the nonrobust algorithm proposed in [8]. The nominal—desired—value of the time scaling factor  $c(t)$  is 1. It is easy to see that when no time-scaling is present, the time evolution of  $\sigma(t)$  is identical to  $s(t)$ , and  $\bar{q}_{ri}(t) = q_{ri}(t)$ , where  $q_{ri}(t)$  is given by (21). The effect of the time-scaling factor

$c(t)$  is that the trajectory  $\bar{q}_{ri}(t)$  is sped up for  $c(t) > 1$ , while for  $c(t) < 1$  the movement is slowed down. Thus, movement speed can be dynamically changed to compensate errors in the resulting tracking performance of trajectory  $\bar{q}_{ri}(t)$  in (26).

The time derivative of (25) is given as follows:

$$\dot{\sigma}(t) = \dot{c}(t)s(t) + c(t)\dot{s}(t),$$

and the path acceleration  $\ddot{\sigma}$  is

$$\ddot{\sigma}(t) = \ddot{c}(t)s(t) + \gamma(c(t), \dot{c}(t), \dot{s}(t), \ddot{s}(t)), \quad (27)$$

where

$$\gamma(c(t), \dot{c}(t), \dot{s}(t), \ddot{s}(t)) = 2\dot{c}(t)\dot{s}(t) + c(t)\ddot{s}(t).$$

The signal  $\ddot{c}(t)$  in (27) is called *time-scaling acceleration*.

Let us consider the trajectory tracking controller given by Eq. (22) with desired trajectory  $\bar{q}_{ri}(t)$  in (26). This trajectory tracking controller can be written in terms of the new parameter  $\sigma$  as follows:

$$\tau_i = \hat{m}_i a_i \ddot{\sigma} + k_{vi} \dot{e}_{pi} + k_{pi} e_{pi}, \quad (28)$$

where

$$e_{pi} = \bar{q}_{ri}(t) - q_i = q_{di}[c(t)s(t)] - q_i, \quad (29)$$

$$\dot{e}_{pi} = \dot{\bar{q}}_{ri}(t) - \dot{q}_i = \dot{q}_{di}[c(t)s(t)] - \dot{q}_i. \quad (30)$$

The path acceleration  $\ddot{\sigma}$  is constrained by the saturation,

$$\ddot{\sigma}_i^{\min} \leq \ddot{\sigma}(t) \leq \ddot{\sigma}_i^{\max},$$

where

$$\ddot{\sigma}_i^{\min} = \frac{\tau_i^{\min} - k_{vi}\dot{e}_{pi} - k_{pi}e_{pi}}{\hat{m}_i a_i},$$

$$\ddot{\sigma}_i^{\max} = \frac{\tau_i^{\max} - k_{vi}\dot{e}_{pi} - k_{pi}e_{pi}}{\hat{m}_i a_i}.$$

It is easy to see that  $\ddot{\sigma}(t)$  in Eq. (27) depends on  $\ddot{c}(t)$ . In this way, instead of constraining the path acceleration  $\ddot{\sigma}(t)$ , it is possible to constrain the scaling acceleration  $\ddot{c}(t)$  using the limits of torque input for each joint  $i$ ,

$$\ddot{c}_i^{\min} \leq \ddot{c} \leq \ddot{c}_i^{\max},$$

where

$$\ddot{c}_i^{\min} = \left[ \frac{\tau_i^{\min} - k_{vi}\dot{e}_{pi} - k_{pi}e_{pi}}{\hat{m}_i a_i} - \gamma(c, \dot{c}, \dot{s}, \ddot{s}) \right] \frac{1}{s},$$

$$\ddot{c}_i^{\max} = \left[ \frac{\tau_i^{\max} - k_{vi}\dot{e}_{pi} - k_{pi}e_{pi}}{\hat{m}_i a_i} - \gamma(c, \dot{c}, \dot{s}, \ddot{s}) \right] \frac{1}{s}.$$

Let us remark that for computing the values of  $\ddot{c}_i^{\min}$  and  $\ddot{c}_i^{\max}$ , the assumption that  $s_0 > 0$  is required to avoid  $\ddot{c}_i^{\min}$  and  $\ddot{c}_i^{\max}$  becoming undetermined.

The limits of the time-scaling acceleration  $\ddot{c}$  are given by

$$\ddot{c}^{\max} = \min_i \{ \ddot{c}_i^{\max} \}, \quad (31)$$

$$\ddot{c}^{\min} = \max_i \{ \ddot{c}_i^{\min} \}. \quad (32)$$

The values of  $\ddot{c}^{\max}$  and  $\ddot{c}^{\min}$  depend on  $s, \dot{s}, \ddot{s}, c, \dot{c}$ , and measured signals  $q_i, \dot{q}_i$ . The bounding limits of the time-scaling acceleration in (31) and (32) provide a way of modifying the reference trajectory (26), so that the torque limits are satisfied.

Considering the limits (31) and (32), we should design an internal feedback to drive the time-scaling factor  $c(t)$  in a proper way. The proposed internal feedback is given by

$$\frac{d}{dt}c = \dot{c}, \quad (33)$$

$$\frac{d}{dt}\dot{c} = \text{sat}(u; \dot{c}^{\min}, \dot{c}^{\max}), \quad (34)$$

with initial conditions  $c(t_0)=1$ , and  $\dot{c}(t_0)=0$ , saturation function

$$\text{sat}(u; \dot{c}^{\min}, \dot{c}^{\max}) = \begin{cases} u & \text{for all } \dot{c}^{\min} \leq u \leq \dot{c}^{\max}, \\ \dot{c}^{\min} & \text{for all } u < \dot{c}^{\min}, \\ \dot{c}^{\max} & \text{for all } u > \dot{c}^{\max}, \end{cases}$$

and

$$u = -k_{vc}\dot{c} + k_{pc}[1 - c]. \quad (35)$$

In Eq. (35),  $k_{vc}$  and  $k_{pc}$  are positive constants. If the saturation function is not active, system (34) satisfies

$$\ddot{c} + k_{vc}\dot{c} + k_{pc}[c - 1] = 0,$$

which is a stable system and  $c(t)$  converges to one exponentially as time increases. In this way, if the resulting reference trajectory  $\bar{q}_{ri}[c(t)s(t)]$  is inadmissible in the sense of unacceptable torques, then it will be modified by the internal feedback (33) and (34), limiting the time-scaling acceleration  $\ddot{c}(t)$ . Fig. 2 depicts a block diagram of the decentralized controller and time-scaling algorithm (28) and (33)–(35).

It is worthwhile to note that the proposed time-scaling method is independent of the definition of path parameter  $s(t)$ . Other definitions of the path parameter  $s(t)$  considering different characteristics in its time evolution can be used along the lines of the proposed time algorithm. See, e.g., [4,3] for other planning algorithms of  $s(t)$ .

Using the discussed technique with observations made before, a point-to-point task can be performed by the following steps:

**Step 1.** Specify the initial and final configuration and the desired path  $q_d(s)$  following the structure of Eq. (5) and using the assumption  $s_f > s_0 > 0$ .

**Step 2.** Using the torque limits and the estimations of the link inertias  $\hat{m}_i$ , compute the path parameter  $s(t)$  as discussed in the procedure of Section 3.

**Step 3.** Program the tracking controller using the  $\bar{q}_d[c(t)s(t)]$  as desired trajectory and Eqs. (28)–(30).

**Step 4.** Carry out the calculations of the maxi-

imum and minimum time-scaling acceleration,  $\ddot{c}_{\max}$  and  $\ddot{c}_{\min}$ , respectively, following Eqs. (31) and (32).

**Step 5.** Implement Eqs. (33)–(35), which provide the update of the time-scaling factor  $c(t)$  used in the desired trajectory  $\bar{q}_d[c(t)s(t)]$ .

$$\frac{d}{dt} \begin{bmatrix} \mathbf{e}_p \\ \mathbf{e}_v \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{e}}_p \\ M(\mathbf{q})^{-1}[-K_v \dot{\mathbf{e}}_p - K_p \mathbf{e}_p + [M(\mathbf{q}) - \hat{M}] \ddot{\bar{\mathbf{q}}}_r(t) + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\dot{\mathbf{q}})] \end{bmatrix},$$

where  $\mathbf{e}_p = [e_{p1}, \dots, e_{pn}]^T$ ,  $e_{pi}$  defined in Eq. (29),

$$\begin{aligned} \ddot{\bar{\mathbf{q}}}_r(t) &= \frac{d}{dt} \mathbf{q}_d[c(t)s(t)] \\ &= \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} [\ddot{c}(t)s(t) + 2\dot{c}(t)\dot{s}(t) + c(t)\ddot{s}(t)] \end{aligned}$$

with  $c(t)$ ,  $\dot{c}(t)$ , and  $\ddot{c}(t)$  obtained through the internal feedback (33)–(35),  $K_v = \text{diag}\{k_{v1}, \dots, k_{vn}\}$ , and  $K_p = \text{diag}\{k_{p1}, \dots, k_{pn}\}$ . Because all the conditions described in either Chap. 4 in [6] or Theorem 1 in [14] are satisfied, we can claim the uniform boundedness of  $[\mathbf{e}_p(t)^T \dot{\mathbf{e}}_p(t)^T]^T$ . Thus, the performance can be improved by selecting properly the control gains  $K_v = \text{diag}\{k_{v1}, \dots, k_{vn}\}$  and  $K_p = \text{diag}\{k_{p1}, \dots, k_{pn}\}$ . In general, a high numerical value of  $K_p$  is a good choice.

## 6. Simulation results

We have evaluated in simulations the performance of the proposed scheme. We have used the model of the two-link planar robot described in [15,16], which is given in the Appendix for self-contents of this paper. Let us remark that the manipulator utilized for simulation tests is a real system built on direct-drive technology. This has the feature that the robot dynamics is really nonlinear, which is important from the control point of view.

For the implementation of the controllers and planning of the trajectories, we chose

$$\hat{m}_1 = 2.0 \text{ [Nm sec}^2/\text{rad]},$$

We have provided a method to slow down the desired trajectories  $\bar{q}_{ri}(t) = q_{di}[c(t)s(t)]$  so that the generated torque (28) will be into the torque space  $\mathcal{T}$ . The position and velocity tracking error dynamics can be written as

$$\hat{m}_2 = 0.1 \text{ [Nm sec}^2/\text{rad]},$$

as estimations of the link inertias.

The task consists in performing a point-to-point motion from the home position  $[q_1(t_0) \ q_2(t_0)]^T = [0 \ 0]^T$  to the final position  $[q_1(t_f) \ q_2(t_f)]^T = [\pi/4 \ \pi/4]^T$  [rad] in minimum time. The torque limits are  $\tau_1^{\max} = 45$ ,  $\tau_1^{\min} = -45$ ,  $\tau_2^{\max} = 4$ , and  $\tau_2^{\min} = -4$ .

The requested task is encoded by the following desired path:

$$\mathbf{q}_d(s) = \begin{bmatrix} s - 1 \\ s - 1 \end{bmatrix}, \quad s \in [1, 1 + \pi/4],$$

and by the time-optimal path parameter

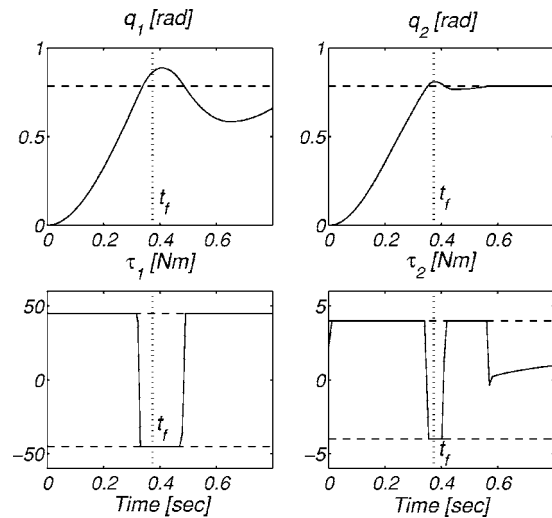


Fig. 3. Trajectory tracking control.

$$s(t) = \begin{cases} 11.25t^2 + 1, & 0 < t \leq t_s, \\ -11.25[t - t_s]^2 + 4.2[t - t_s] + 1 + \pi/8, & t_s < t \leq t_f, \end{cases} \quad (36)$$

where  $t_s=0.1868$  [sec] and  $t_f=0.3736$  [sec], which was computed using Eqs. (17) and (18) with  $t_0=0$ .

### 6.1. Tracking control

We carried out a simulation with the trajectory tracking controller given by Eq. (22). The used gains were

$$\begin{aligned} k_{p1} &= 4000, & k_{p2} &= 4000, \\ k_{v1} &= 10, & k_{v2} &= 10. \end{aligned} \quad (37)$$

The reason to use high proportional gain  $k_{pi}$  is to reject as much as possible the unmodeled dynamics and then obtain a stiff response. In other words, high proportional gain allows that the ultimate bound of the signals  $e_{pi}(t)$  and  $\dot{e}_{pi}(t)$ ,  $i = 1, 2$ , be small ([6], Chap. 4). It is noteworthy that this only holds under the assumption that the actuators provide unlimited torque.

The resulting numerical simulation is shown in Fig. 3, where we depict the time evolution of the joint positions and the applied torques. It is observed that the control inputs  $\tau_1$  and  $\tau_2$  are saturated at the same time, which is not natural in minimum time controls [12].

### 6.2. Dahl and Nielsen [9] scheme

Dahl and Nielsen [11] introduced a technique to scale in time the reference trajectories. This technique is outlined here,

$$\tau_i = \hat{m}_i a_i \ddot{\sigma} + k_{vi} \dot{e}_{pi} + k_{pi} e_{pi}, \quad (38)$$

where  $e_{pi} = q_{di}(\sigma) - q_i$ ,  $\dot{e}_{pi} = (d/dt)q_{di}(\sigma) - \dot{q}_i$ , and  $\sigma$  is obtained from the internal feedback

$$\frac{d}{dt}\sigma = \dot{\sigma}, \quad (39)$$

$$\frac{d}{dt}\dot{\sigma} = \text{sat}(u, \ddot{\sigma}^{\min}, \ddot{\sigma}^{\max}), \quad (40)$$

$$u = v_2(\sigma) + \frac{\alpha}{2}[v_1(\sigma)^2 - \dot{\sigma}^2], \quad (41)$$

with  $\alpha > 0$  and

$$\ddot{\sigma}^{\min} = \max_i \left\{ \frac{\tau_i^{\min} - k_{vi} \dot{e}_{pi} - k_{pi} e_{pi}}{\hat{m}_i a_i} \right\},$$

$$\ddot{\sigma}^{\max} = \min_i \left\{ \frac{\tau_i^{\max} - k_{vi} \dot{e}_{pi} - k_{pi} e_{pi}}{\hat{m}_i a_i} \right\},$$

$$v_1(\sigma) = \begin{cases} \sqrt{2\ddot{s}_{\max}[\sigma - s_0]}, & s_0 \leq \sigma \leq s_s, \\ \sqrt{2\ddot{s}_{\min}[\sigma - s_f] + \dot{s}_s^2}, & s_s < \sigma \leq s_f, \\ 0 & \text{elsewhere,} \end{cases}$$

$$v_2(\sigma) = \begin{cases} \dot{s}_{\max}, & s_0 \leq \sigma \leq s_s, \\ \dot{s}_{\min}, & s_s < \sigma \leq s_f, \\ 0 & \text{elsewhere.} \end{cases} \quad (42)$$

The initial conditions for the system (39)–(41) are  $\sigma(0) = s_0$  and  $\dot{\sigma}(0) = v_1(s_0) = 0$ . The function  $v_1(\sigma)$  in (42) is denoted *velocity profile*. Further details on the rationale behind the algorithm (38)–(42) can be found in [11]. Note that the parameters involved in the velocity profile function (42) can be obtained through the path parameter  $s(t)$  in (36), which are for the simulation example  $s_s = 1.39$ ,  $\dot{s}_s = 4.2$ ,  $s_f = 1.78$ ,  $\ddot{s}_{\max} = 22.5$ , and  $\ddot{s}_{\min} = -22.5$ . The simulation was done using the numerical value of the control gains  $k_{vi}$  and  $k_{pi}$  in (37) for the torque input (38). In addition, we set  $\alpha = 5000$  in the internal feedback (39) and (40). Fig. 4 shows the time evolution of the joint positions and the applied torques. Smooth response of the joint positions is observed and no simultaneous saturation of the torque inputs  $\tau_1$  and  $\tau_2$  is achieved.

### 6.3. Proposed scheme

The proposed scheme of time-scaling of trajectories (28) and (33)–(35) was simulated using the gains (37), which was done to keep a similar con-

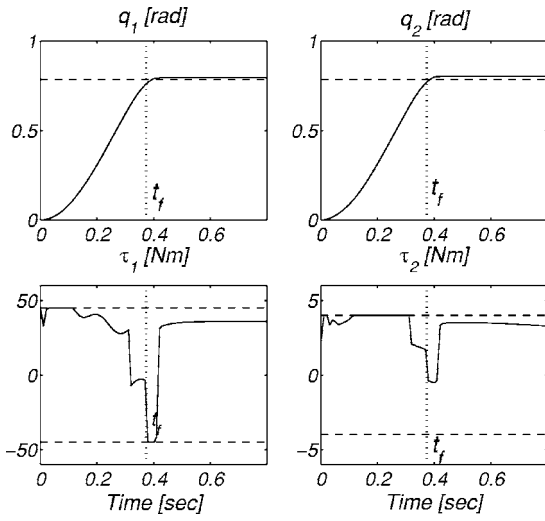


Fig. 4. Dahl and Nielsen scheme.

dition in simulations of the classical tracking control and the Dahl and Nielsen method. Besides, we selected

$$k_{vc} = 100 \quad \text{and} \quad k_{pc} = 2500,$$

which were chosen to obtain a critically damped system in the internal feedback (35), i.e.,  $k_{vc} = 2\sqrt{k_{pc}}$ .

Fig. 5 shows the time history of the robot joint positions and the applied torques. It is observed that at the planned minimum time  $t_f$ , the tracking error is very small. Moreover, the time evolution of the applied torques is well-behaved since no

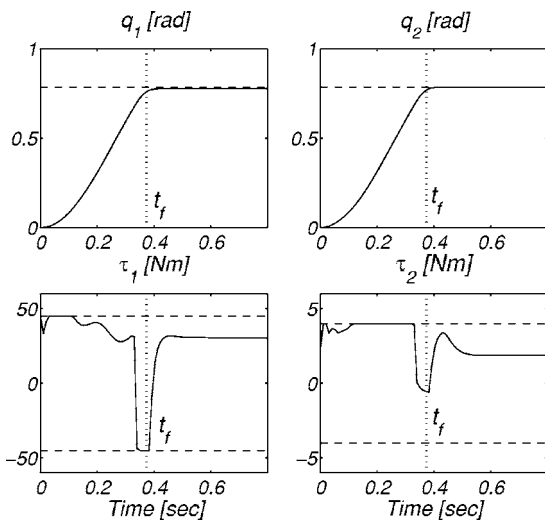


Fig. 5. Proposed scheme of dynamic time scaling of trajectories.

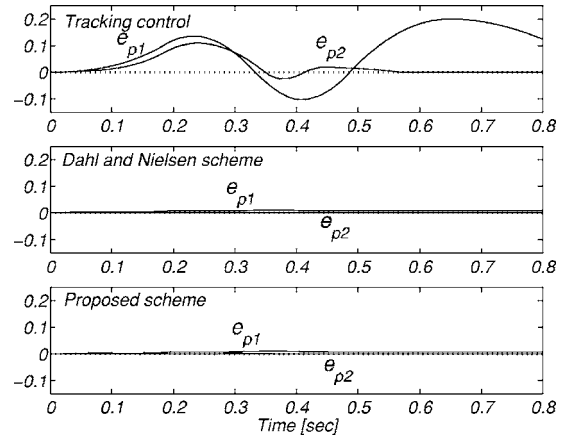


Fig. 6. Tracking errors in the three approaches.

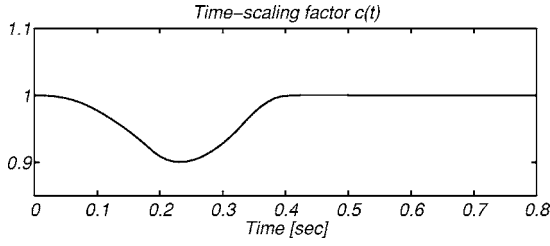
multiple saturations are presented, like in the simulation of the Dahl and Nielsen method. Inspection of Figs. 4 and 5 shows that no important differences in the performance of the Dahl and Nielsen scheme with respect to the proposed scheme are obtained.

#### 6.4. Discussions

Visual examination of Figs. 3–5 is not suitable for quantitative description of the closed-loop performance, but tracking errors can be a useful performance index.

Tracking errors for the three approaches are depicted in Fig. 6. For the trajectory tracking control (22) it is observed that the bigger tracking error peak for  $e_{p1}(t)$  it is 0.2 [rad], while for  $e_{p2}(t)$  it is 0.11 [rad]. For the values of the gains in the internal feedback used in the Dahl and Nielsen scheme and the proposed scheme of time-scaling of trajectories, the same numerical value of bigger tracking error peak is obtained. Specifically, for  $e_{p1}(t)$  it is 0.01 [rad], while for  $e_{p2}(t)$  it is 0.001 [rad]. The improvement of the performance using either the Dahl and Nielsen scheme or the proposed scheme with respect to the classical trajectory tracking control is 95% and 99.1% for joint 1 and 2, respectively. It is important to remark that in the Dahl and Nielsen scheme, the definition of the tracking error signal definition is  $e_{pi} = q_{di}(\sigma) - q_i$ , where  $\sigma$  obtained from (39)–(41), and in the proposed scheme  $e_{pi} = q_{di}(cs) - q_i$ , with  $c$  obtained from (33)–(35).

The tracking errors are bigger in the trajectory tracking approach because the time-optimal con-


 Fig. 7. Time evolution of the time-scaling factor  $c(t)$ .

trols require more torque than is available. In particular, the proposed scheme gives a better result than the classical approach of tracking control (22), because it rejects the unmodeled dynamics slowing down the reference trajectory  $q_{ri}(t) = q_{di}[c(t)s(t)]$ . Finally, in Fig. 7 is presented the time evolution of the time-scaling factor  $c(t)$ .

## 7. Summary

This paper addressed the problem of time-optimal trajectory tracking of point-to-point robotic tasks under constrained torque inputs. Assuming a nominal robot system consisting in  $n$  decoupled double integrators, we propose an algorithm for the time-scaling of a minimum time trajectory. The algorithm is based on a time-scaling factor that changes dynamically in accordance with to the torque limits and tracking errors. As shown by simulation results, the proposed approach has better performance than the classical approach of trajectory tracking control, while no important differences in the performance with respect to the Dahl and Nielsen scheme were found.

## Appendix

With reference to the symbols listed in Table 1

Table 1  
 Parameters of the manipulator.

	notation	value	unit
Length link 1	$l_1$	0.45	m
Length link 2	$l_2$	0.45	m
Link (1) center of gravity	$l_{c1}$	0.091	m
Link (2) center of gravity	$l_{c2}$	0.048	m
Mass link 1	$m_1$	23.90	kg
Mass link 2	$m_2$	3.88	kg
Inertia link 1	$I_1$	1.27	kg m <sup>2</sup>
Inertia link 2	$I_2$	0.09	kg m <sup>2</sup>
Gravity acceleration	$g$	9.8	m/sec <sup>2</sup>

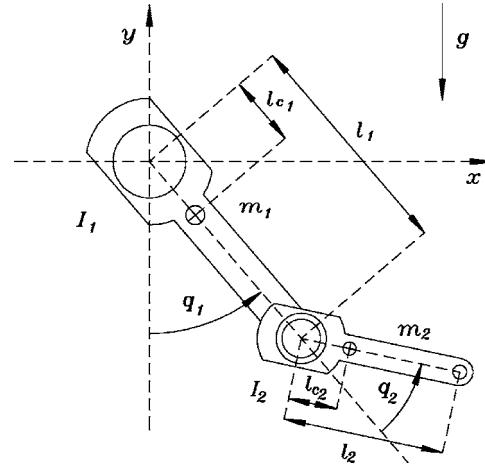


Fig. 8. Robot manipulator.

and Fig. 8, the entries of the experimental robot arm described in [15,16] are presented below.

The elements  $M_{ij}(\mathbf{q})(i, j=1, 2)$  of the inertia matrix  $M(\mathbf{q})$  are

$$M_{11}(\mathbf{q}) = m_1 l_{c1}^2 + m_2 [l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)] + I_1 + I_2,$$

$$M_{12}(\mathbf{q}) = m_2 [l_{c2}^2 + l_1 l_{c2} \cos(q_2)] + I_2,$$

$$M_{21}(\mathbf{q}) = m_2 [l_{c2}^2 + l_1 l_{c2} \cos(q_2)] + I_2,$$

$$M_{22}(\mathbf{q}) = m_2 l_{c2}^2 + I_2.$$

The elements  $C_{ij}(\mathbf{q}, \dot{\mathbf{q}})(i, j=1, 2)$  from the centrifugal and Coriolis matrix  $C(\mathbf{q}, \dot{\mathbf{q}})$  are

$$C_{11}(\mathbf{q}, \dot{\mathbf{q}}) = -m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2,$$

$$C_{12}(\mathbf{q}, \dot{\mathbf{q}}) = -m_2 l_1 l_{c2} \sin(q_2) [\dot{q}_1 + \dot{q}_2],$$

$$C_{21}(\mathbf{q}, \dot{\mathbf{q}}) = m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1,$$

$$C_{22}(\mathbf{q}, \dot{\mathbf{q}}) = 0.$$

The entries of the gravitational torque vector  $\mathbf{g}(\mathbf{q})$  are given by

$$g_1(\mathbf{q}) = [m_1 l_{c1} + m_2 l_1] g \sin(q_1) + m_2 l_{c2} g \sin(q_1 + q_2),$$

$$g_2(\mathbf{q}) = m_2 l_{c2} g \sin(q_1 + q_2).$$

Finally, entries of the vector of torques due to the viscous and Coulomb friction  $\mathbf{f}(\mathbf{q})$  are given by

$$f_1(\dot{q}) = b_1\dot{q}_1 + f_{C1} \operatorname{sgn}(\dot{q}_1),$$

$$f_2(\dot{q}) = b_2\dot{q}_2 + f_{C2} \operatorname{sgn}(\dot{q}_2),$$

where the coefficients are  $b_1=2.288$  [N m sec],  $f_{C1}=7.5$  [N m],  $b_2=0.175$  [N m sec], and  $f_{C2}=1.7$  [N m].

## References

- [1] Pfeiffer, E. and Johanni, R., A concept for manipulator trajectory planning. *IEEE J. Rob. Autom.* **RA-3**, 115–123 (1987).
- [2] Bobrow, J., Dubowsky, S., and Gibson, J., Time-optimal control of robotic manipulators along specified paths. *Int. J. Robot. Res.* **4**(3), 3–17 (1985).
- [3] Cao, B., Dodds, G. I., and Irwin, G. W., Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots. *IEE Proc.: Control Theory Appl.* **44**(5), 467–475 (1997).
- [4] Macfarlane, S. and Croft, E. A., Jerk bounded manipulator trajectory planning: Design for real-time applications. *IEEE Trans. Rob. Autom.* **19**(1), 42–52 (2003).
- [5] Abdallah, C., Dawson, D., Dorato, P., and Jamshidi, M., Survey of robust control for rigid robots. *IEEE Control Syst. Mag.* **11**(2), 24–30 (1991).
- [6] Lewis, F. L., Abdallah, C. T., and Dawson, D., *Control of Robot Manipulators*. Macmillan, New York, 1993.
- [7] Sciavicco, L. and Siciliano, B., *Modeling and Control of Robot Manipulators*. Springer, London, 2000.
- [8] Hollerbach, J. M., Dynamic scaling of manipulator trajectories. *J. Dyn. Syst., Meas., Control* **106**, 102–106 (1984).
- [9] Moon, S. B. and Ahmad, S., Time scaling of cooperative multirobot trajectories. *IEEE Trans. Syst. Man Cybern.* **21**(4), 900–908 (1991).
- [10] De Luca, A. and Farina, R., Dynamic scaling of trajectories for robots with elastic joints. *Proc. of the IEEE ICRA*, Washington, May, 2436–2442 (2002).
- [11] Dahl, O. and Nielsen, L., Torque-limited path following by on-line trajectory time scaling. *IEEE Trans. Rob. Autom.* **6**(5), 554–561 (1990).
- [12] Arai, H., Tanie, K., and Tachi, S., Path tracking control of a manipulator considering torque saturation. *IEEE Trans. Ind. Electron.* **41**(1), 25–31 (1994).
- [13] Lewis, F. L. and Syrmos, V. L., *Optimal Control*. John Wiley and Sons, New York, 1995.
- [14] Wen, J. T., A unified perspective on robot control: The energy Lyapunov function approach. *Int. J. Adapt. Control Signal Process.* **4**(6), 487–500 (1990).
- [15] Reyes, F. and Kelly, R., Evaluation of identification schemes on a direct drive robot. *Robotica* **15**, 563–571 (1997).
- [16] Reyes, F. and Kelly, R., Experimental evaluation of model-based controllers on a direct-drive robot arm. *Mechatronics* **11**, 267–282 (2001).

**Javier Moreno-Valenzuela** was born in Culiacán, Mexico, in 1974. He received the B.Sc. degree in Electronics Engineering from the Instituto Tecnológico de Culiacán, Mexico, in 1997, and the Ph.D. degree in Automatic Control from the Centro de Investigación Científica y de Educación Superior de Ensenada, CICESE, Mexico, in 2002. He was an Associate Researcher at the Centro de Investigación y Desarrollo de Tecnología Digital del IPN, CITEDI-IPN, Tijuana, Mexico from 2002 to 2004 and a Postdoctoral Fellow at the Université de Liège, Belgium, from 2004 to 2005. Currently, he is at CITEDI-IPN Research Center. His main research interests are control of electromechanical systems.