A scaling algorithm for the generation of jerk-limited trajectories in the operational space[☆]

Corrado Guarino Lo Bianco[1], Fabio Ghilardelli

Dipartimento di Ingegneria dell'Informazione, University of Parma, Viale delle Scienze,181/A, Parma, Italy, email:{guarino, fghilardelli}@ce.unipr.it.

Abstract

Kinematic singularities represent a relevant problem for trajectories that are defined in the operational space. In case of industrial applications characterized by non-repetitive tasks, feasibility cannot be checked in advance, so that appropriate methods have been developed for the online management of otherwise critical situations. In this paper, a scaling scheme proposed in the past for the automatic handling of possibly unfeasible trajectories is revised in order to generate jerk-limited reference signals: close to critical points, trajectories are appropriately slowed down such to guarantee an accurate tracking of the assigned path in the operational space. The actual performances of the proposed system have been experimentally verified on a commercial manipulator by means of extensive tests.

Keywords: Anthropomorphic manipulator, real-time singularity management, wrist singularity, trajectory scaling, jerk constraints

1. Introduction

Trajectories for robotic manipulators are commonly planned by fulfilling proper optimality criteria. Several performance indexes can be considered to this purpose: energy consumption [1, 2], jerk minimization [3, 4] but, for performance reasons, the most commonly used index is undoubtedly the total

[1] Corresponding author

traveling time [5, 6]. Sometimes, hybrid performance indexes are considered [7, 8].

In case of repetitive tasks, optimal trajectories can be evaluated offline [9], but many applications require real-time planning capability. For this reason, several real-time planners have been recently proposed for the management of multi-axis problems in the configuration space. Among them, it is worth citing those proposed in [10, 11, 12] for the generation of jerk constrained trajectories or the recent algorithm proposed in [13] which is able to bound up to nth generic derivative of the output signal by using a recursive strategy. In all the previously mentioned techniques the path is not defined in advance, but it is the result of the planning phase.

In alternative contexts, paths are specifically imposed by the task – or by the users – through a curve $\mathbf{p}$ expressed as a function of a scalar variable s, defined curvilinear coordinate. Optimality is subsequently gained by properly planning a time law $s(t)$: the overall trajectory is given by the composite function $\mathbf{p}[s(t)]$. The advantage of this approach, originally proposed in [14] and known as path-velocity decomposition, is that it splits the original problem into two simpler sub-problems.

The path-velocity decomposition strategy can indifferently be used for the management of problems in the configuration space or in the operational space. The first solutions to the optimal, minimum-time planning problem, based on the path-velocity decomposition, were proposed in [15, 16, 17, 18] in an offline context and by considering velocity, acceleration, and torque constraints. Still in an offline context, in [19] and in [20] the problem was extended by also accounting for third order constraints concerning the actuator feeding voltages and the torque derivatives, respectively. The original problem has been successively re-elaborated by considering alternative approaches. For example, a sequential convex programming method is used in [21], while a strategy based on the Pontryagin Maximum principle is adopted in [22] for the generation of optimal trajectories subject to velocity, acceleration, and torque constraints. An interesting characteristic of the approach proposed in [22] is that, despite it can be classified as an offline method, its evaluation times are compatible with those of many real-time applications.

The planning strategy must necessarily change when considering systems which target is not known in advance – for example, in a pick-and-place application driven by a visual system – or which are moved by the users through a teaching device. In both cases the path and the time-law must be planned in real-time, thus raising possible feasibility issues that must be online managed through appropriate algorithms. The most common approach to the problem is based on the online scaling of $s(t)$. The first real-time strategies were proposed in [23, 24] for problems in the configuration space and by considering torque constrained systems. That early approach was affected by some drawbacks which limited its diffusion in industrial contexts. For example, the time law was specified in an unnatural way as $\dot{s}(s)$. A direct consequence of this choice is that delays introduced to preserve feasibility cannot be eliminated. Such drawback has been eliminated in two alternative approaches proposed in [25, 26].

Another interesting real-time planning approach has been proposed in [27] for a pure kinematic problem concerning a manipulator subject to jerk bounds. The trajectory scaling approach has been subsequently extended in [28, 29], still considering planning strategies in the configuration space, by simultaneously managing velocity, acceleration, jerk, torque and torque derivative constraints.

In [30, 31] the scaling problem has been extended to trajectories that are not defined through the path-velocity decomposition. Joint reference signals are online modified in order to fulfill a set of kinematic constraints in the configuration space by preserving, at the same time, the underlying trajectory path.

The previously cited approaches consider paths and/or constraints in the configuration space. The problem complicates when the path primitive is defined in the operational space and it becomes critical especially close to singular points, when given limits could be violated even in case of slow movements. Several applications in the operational space might benefit from the use of real-time scaling approaches. The most obvious ones concern manually operated manipulators that are moved in the Cartesian space through a teaching device like, e.g., in [32], or applications in which assigned paths must be modified in real time in order to avoid obstacles like, e.g., in [33]. In [32], close to a kinematic singularity, $s(t)$ is scaled down by means of an optimization procedure which preserves feasibility. The evaluation times of the optimization algorithm limit make the method suitable for slow motions or to manage a limited number of constraints. More bounds are considered in [33], but joint constraints are not handled in case of trajectories in the operational space.

This work proposes an algorithm – which core is represented by a recently devised nonlinear filter [34] – for the real-time scaling of trajectories in the operational space. Differently from any other real-time strategy proposed in

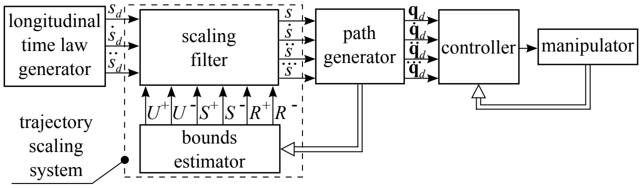

Figure 1: A trajectory scaling system is placed between the longitudinal time law generator and the path generator in order to achieve feasible trajectories. The scaling filter generates an output which is compatible with the assigned bounds: $\dot{s} \in [R^-, R^+]$, $\ddot{s} \in [S^-, S^+]$, and $\dddot{s} \in [U^-, U^+]$.

the literature, it is able to simultaneously manage constraints on velocities, accelerations, and jerks indifferently assigned in the configuration space – for the fulfillment of the joint actuators' limits – and in the operational space – in order to guarantee an adequate comfort of the payload and to reduce system vibrations. This result is achieved with an efficient real-time algorithm which computational time is close to 50 μs. Differently from the system proposed in [23, 24], it uses a time-law which is directly defined as a function of the time and eliminates delays accumulated to preserve feasibility. Finally, even if this work, for space reasons, will specifically focus on kinematic constraints, the paper will show how dynamic limits can be easily included in the scaling scheme.

This work is the natural prosecution of the one described in [35] for the management of the sole velocity and acceleration constraints, so that comparisons between the two strategies are proposed in the experimental section. Additional comparisons have been carried out with respect to the performances of the commercial controller that is sold together with the Comau Smart SiX 6.14 manipulator, for a total of more than 8000 trajectories passing close to a singular configuration. Results have shown that, with the aid of the scaling system, areas that were previously precluded to the motion become reachable.

The higher complexity of the novel algorithm with respect to the one proposed in [35] has raised problems correlated to the evaluation times. For example, the novel scaling system requires the analytical evaluation of the second derivative of the Jacobian matrix associated to the end-effector, which is defined, owing to path-velocity decomposition, as follows: $\mathbf{J}_T\{\mathbf{p}[s(t)]\}$. This operation, that could be potentially time consuming, represents the main problem which needs to be solved for the actual implementation of the jerk-constrained scaling system. For this reason, the paper proposes a novel algorithm which simultaneously evaluates the first and the second derivatives of $\mathbf{J}_T\{\mathbf{p}[s(t)]\}$ with a computational burden that is only slightly higher than the one required in [35] for the evaluation of the sole first derivative. The algorithm, despite it represents the main outcome of the paper, is described in Appendix A for readability reasons, while the paper is mainly focused on the description of the novel scaling system and, as previously announced, on the comparison of its performances with the ones that can be achieved with the acceleration-bounded version.

The paper is organized as follows. The real-time, constrained planning problem is formulated in Section 2. Section 3 shows how constraints in the operational and in the configuration space can be converted into equivalent constraints in the curvilinear-coordinate space. Experimental results are reported in Section 4. In the same section, the acceleration-constrained and the jerk-constrained scaling systems are compared in order to establish the most appropriate application fields for each of them. Final conclusions are drawn in Section 5.

Notation

The following notation is used along the paper

$\mathbf{q}$	generalized joint variables
$\mathbf{p}_i$	position of the origin of frame i w.r.t. frame 0
${}^0_i\mathbf{R}$	rotation matrix which describes the orientation of frame i w.r.t. frame 0
$\mathbf{v}_i$	linear velocity of frame i w.r.t. frame 0
$\boldsymbol{\omega}_i$	angular velocity of frame i w.r.t. frame 0
$\mathbf{a}_i$	linear acceleration of frame i w.r.t. frame 0
$\boldsymbol{\alpha}_i$	angular acceleration of frame i w.r.t. frame 0
$\mathbf{j}_i$	linear jerk of frame i w.r.t. frame 0
$\boldsymbol{\iota}_i$	angular jerk of frame i w.r.t. frame 0
$\mathbf{J}_i$	jacobian matrix associated to frame i

Subscript $i = T$ indicates quantities which refer to the tool frame.

Generalized velocities, accelerations, and jerks associated to any generic frame i are respectively defined as follows: $\widetilde{\mathbf{v}}_i := [\mathbf{v}_i^T \; \boldsymbol{\omega}_i^T]^T$, $\widetilde{\mathbf{a}}_i := [\mathbf{a}_i^T \; \boldsymbol{\alpha}_i^T]^T$, and $\widetilde{\mathbf{j}}_i := [\mathbf{j}_i^T \; \boldsymbol{\iota}_i^T]^T$.

2. Problem formulation

In many robotic contexts, trajectories are defined according to the path-velocity decomposition, i.e., they are obtained by specifying a path, which is function of the so-called curvilinear coordinate s, and a Longitudinal Time Law (LTL) $s(t)$, which associates s to time. The path is typically specified through two different functions: the first one, i.e., $\mathbf{p}_T(s)$, expresses the position of tool frame T w.r.t. an inertial frame which typically coincides with frame 0, while the second one, i.e., ${}^{0}_{T}\mathbf{R}(s)$, expresses its orientation.

The main assumption of this work is that trajectories are provided to the system without any preliminary investigation, so that they could potentially violate the physical constraints of the system or they could be insufficiently smooth for the considered task. The resulting, undesired effects can be mitigated by adopting a Trajectory Scaling System (TSS) like the one shown in Figure 1, which automatically modifies the user-provided LTL, i.e., $s_d(t)$, in order to fulfill a given set of constraints and to preserve the assigned path. As shown in Figure 1, the scaling system does not require any feedback from the controller. The design solution adopted in this paper is clearly different from the ones used in other works (see e.g. [28]) where feedback signals coming from the controller were exploited in order to mitigate undesired behaviors caused by model uncertainties and external disturbances. The change in the design approach is motivated by the necessity of enlarging the number of potential users: schemes which exploit feedbacks from the controllers can be hardly implemented, since controllers of commercial manipulators are typically "closed", i.e., their internal structure is unknown, so that it is practically impossible to obtain any feedback from them.

The approach to the problem is clearly the same already considered in [35], with the addition of the jerk constraints that have been explicitly introduced in order to increase the motion smoothness. The proposed scaling strategy handles constraints that can be indifferently assigned in the configuration and/or in the operational space. More in detail, limits in the configuration space involve joint velocities, accelerations and jerks and can be summarized as follows

$$\underline{\dot{\mathbf{q}}} \leq \dot{\mathbf{q}} \leq \overline{\dot{\mathbf{q}}}\,, \tag{1}$$

$$\underline{\ddot{\mathbf{q}}} \leq \ddot{\mathbf{q}} \leq \overline{\ddot{\mathbf{q}}}\,, \tag{2}$$

$$\underline{\dddot{\mathbf{q}}} \leq \dddot{\mathbf{q}} \leq \overline{\dddot{\mathbf{q}}}\,, \tag{3}$$

where $\underline{\dot{\mathbf{q}}}$, $\underline{\ddot{\mathbf{q}}}$, $\underline{\dddot{\mathbf{q}}}$, $\overline{\dot{\mathbf{q}}}$, $\overline{\ddot{\mathbf{q}}}$, and $\overline{\dddot{\mathbf{q}}}$ are proper upper and lower bounds. Velocity limits normally depend on the maximum motors' speeds and reduction gears' speeds. Acceleration bounds are correlated to the maximum motor torques, while jerk constraints are used to limit the system wear. The presence of jerk bounds guarantees that generated trajectories will be continuous together with their first and second time derivatives, i.e., $\mathbf{q} \in \mathcal{C}^2$.

The approach proposed in this paper also handles constraints in the operational space in order to reduce the mechanical solicitations acting on the payload. Given proper upper and lower bounds $\underline{\mathbf{v}}_T$, $\overline{\mathbf{v}}_T$, $\underline{\mathbf{a}}_T$, $\overline{\mathbf{a}}_T$, $\underline{\mathbf{j}}_T$, and $\overline{\mathbf{j}}_T$ for the generalized velocities, accelerations, and jerks of the tool frame, trajectories must be scaled such to guarantee that the following inequalities are satisfied

$$\underline{\mathbf{v}}_T \le \widetilde{\mathbf{v}}_T \le \overline{\mathbf{v}}_T \,, \tag{4}$$

$$\underline{\mathbf{a}}_T \le \widetilde{\mathbf{a}}_T \le \overline{\mathbf{a}}_T \,, \tag{5}$$

$$\underline{\mathbf{j}}_T \le \widetilde{\mathbf{j}}_T \le \overline{\mathbf{j}}_T \,. \tag{6}$$

A last set of constraints directly acts on the LTL. To this purpose, the following additional limits have been considered

$$\underline{\dot{s}} \le \dot{s} \le \overline{\dot{s}} \,, \tag{7}$$

$$\underline{\ddot{s}} \le \ddot{s} \le \overline{\ddot{s}} \,, \tag{8}$$

$$\underline{\dddot{s}} \le \dddot{s} \le \overline{\dddot{s}} \,, \tag{9}$$

where $\underline{\dot{s}}$, $\overline{\dot{s}}$, $\underline{\ddot{s}}$, $\overline{\ddot{s}}$, $\underline{\dddot{s}}$, and $\overline{\dddot{s}}$ are given bounds.

The problem that is solved in the paper can be formulated as follows:

Problem 1. *Given a nominal trajectory, planned according to the path-velocity decomposition approach, smoothly scale its LTL $s(t)$, by means of minimum-time transients, such that (1)–(9) simultaneously hold and $\mathbf{q} \in \mathcal{C}^2$. Moreover, delays accumulated to preserve feasibility must be eliminated in non-critical configurations through minimum-time transients and compatibly with the given bounds.*

The problem is solved by first converting all constraints defined in the operational and in the configuration spaces, i.e., (1)–(6), into analogous constraints in the curvilinear-coordinate space, i.e., into a form that is similar to the one proposed in (7)–(9) for $s(t)$. Such transformation evidently simplifies the problem, since it can then be handled by operating within a single domain.

The conversion is accomplished by the bound estimator block shown in Figure 1, which online evaluates equivalent lower and upper bounds $\dot{s}(t)$, $\ddot{s}(t)$, and $\dddot{s}(t)$. The nominal time-law $s_d(t)$ is then modified by a nonlinear filter which generates a feasible output signal $s(t) \in \mathcal{C}^2$. The filter characteristics and its performances have been widely described in [34]. Its main characteristics can be summarized here for the reader convenience: if $s_d(t)$ is feasible, then $s(t) = s_d(t)$, otherwise $s(t)$ is generated such to be the best feasible approximation of $s_d(t)$. Another relevant property of the filter is that, compatibly with the given bounds, it always attempts to eliminate, by means of minimum-time transients, any possible delay between $s_d(t)$ and $s(t)$. Readers that are interested to a detailed description of the nonlinear filter can refer to [34].

3. Evaluation of the equivalent bounds for the longitudinal time-law

As seen in Section 2, the first problem which needs to be dealt with concerns the conversion of the constraints originally defined in the configuration space and in the operational space into equivalent constraints in the curvilinear-coordinate space. In order to achieve this result, the trajectory in the operational space needs to be converted into an analogous one in the joint space. More precisely, given a LTL, defined through a function $s(t)$ of class $\mathcal{C}^2$, and a path given by the pair $\mathbf{p}_T(s)$, ${}^{0}_{T}\mathbf{R}(s)$, still composed by functions of class $\mathcal{C}^2$, the first step of the procedure requires the derivation of the following closed-form equations: $\mathbf{q}(s)$, $\dot{\mathbf{q}}(s, \dot{s})$, $\ddot{\mathbf{q}}(s, \dot{s}, \ddot{s})$, and $\dddot{\mathbf{q}}(s, \dot{s}, \ddot{s}, \dddot{s})$. The evaluation of $\mathbf{q}(s)$ is straightforward, since it requires the solution of an inverse kinematics problem. The analytic synthesis of its derivatives is more complex, since it is based on the evaluation of the Jacobian matrix and of its first two time derivatives. It is known, indeed, that generalized velocities, accelerations and jerks can be evaluated according to the following equations

$$\widetilde{\mathbf{v}}_T = \mathbf{J}_T\,\dot{\mathbf{q}}\;, \tag{10}$$

$$\widetilde{\mathbf{a}}_T = \dot{\mathbf{J}}_T\,\dot{\mathbf{q}} + \mathbf{J}_T\,\ddot{\mathbf{q}}\;, \tag{11}$$

$$\widetilde{\mathbf{j}}_T = \ddot{\mathbf{J}}_T\,\dot{\mathbf{q}} + 2\dot{\mathbf{J}}_T\,\ddot{\mathbf{q}} + \mathbf{J}_T\,\dddot{\mathbf{q}}\;, \tag{12}$$

which can be clearly rearranged as follows

$$
\begin{aligned}
\dot{\mathbf{q}} &= \mathbf{J}_T^{-1}\,\widetilde{\mathbf{v}}_T\ , && (13)\\
\ddot{\mathbf{q}} &= \mathbf{J}_T^{-1}(\widetilde{\mathbf{a}}_T - \dot{\mathbf{J}}_T\,\dot{\mathbf{q}}) = \mathbf{J}_T^{-1}(\widetilde{\mathbf{a}}_T - \dot{\mathbf{J}}_T\,\mathbf{J}_T^{-1}\,\widetilde{\mathbf{v}})\ , && (14)\\
\dddot{\mathbf{q}} &= \mathbf{J}_T^{-1}(\widetilde{\mathbf{j}}_T - \ddot{\mathbf{J}}_T\,\dot{\mathbf{q}} - 2\dot{\mathbf{J}}_T\,\ddot{\mathbf{q}}) && \\
&= \mathbf{J}_T^{-1}\left[\widetilde{\mathbf{j}}_T - \ddot{\mathbf{J}}_T\,\mathbf{J}_T^{-1}\,\widetilde{\mathbf{v}}_T - 2\dot{\mathbf{J}}_T\,\mathbf{J}_T^{-1}(\widetilde{\mathbf{a}}_T - \dot{\mathbf{J}}_T\,\mathbf{J}_T^{-1}\,\widetilde{\mathbf{v}}_T)\right]\ . && (15)
\end{aligned}
$$

Closed form representations for $\dot{\mathbf{q}}(s,\dot{s})$, $\ddot{\mathbf{q}}(s,\dot{s},\ddot{s})$, and $\dddot{\mathbf{q}}(s,\dot{s},\ddot{s},\dddot{s})$ can thus be obtained by expressing all terms in (13)–(15) as functions of $s(t)$ and its derivatives. Such operation, already considered in [20] for an offline approach, raises efficiency concerns in view of real-time implementations since evaluation times become relevant. The computation of terms $\widetilde{\mathbf{v}}_T$, $\widetilde{\mathbf{a}}_T$, and $\widetilde{\mathbf{j}}_T$ does not pose particular problems. In particular, by extending to $\widetilde{\mathbf{j}}_T$ the procedure already seen in [35] for $\widetilde{\mathbf{v}}_T$ and $\widetilde{\mathbf{a}}_T$, it is possible to derive the following equations starting from the knowledge of $\mathbf{p}_T(s)$ and ${}^{0}_{T}\mathbf{R}(s)$

$$
\begin{aligned}
\widetilde{\mathbf{v}}_T(s,\dot{s}) &= \widetilde{\mathbf{v}}_T(s)\,\dot{s}\ , && (16)\\
\widetilde{\mathbf{a}}_T(s,\dot{s},\ddot{s}) &= \widetilde{\mathbf{v}}'_T(s)\,\dot{s}^2 + \widetilde{\mathbf{v}}_T(s)\,\ddot{s}\ , && (17)\\
\widetilde{\mathbf{j}}(s,\dot{s},\ddot{s},\dddot{s}) &= \widetilde{\mathbf{v}}''_T(s)\,\dot{s}^3 + 3\,\widetilde{\mathbf{v}}'_T(s)\,\dot{s}\ddot{s} + \widetilde{\mathbf{v}}_T(s)\,\dddot{s}\ , && (18)
\end{aligned}
$$

where $\widetilde{\mathbf{v}}'_T(s) := [d\widetilde{\mathbf{v}}_T(s)]/[ds]$ and $\widetilde{\mathbf{v}}''_T(s) := [d^2\widetilde{\mathbf{v}}_T(s)]/[ds^2]$.

The synthesis of closed form equations for $\mathbf{J}(s)$, $\dot{\mathbf{J}}(s,\dot{s})$, and $\ddot{\mathbf{J}}(s,\dot{s},\ddot{s})$ is more critical since, depending on the algorithmic implementation, the computational burden could be excessive. To this purpose, the procedure devised in [35] for the computation of $\dot{\mathbf{J}}(s,\dot{s})$ has been revised in order to simultaneously provide $\ddot{\mathbf{J}}(s,\dot{s},\ddot{s})$ with a minor additional burden, so that, while the sole first derivative is obtained in 11.3 μs, the combined evaluation of the two derivatives only requires 14.1 μs (Intel Core2 Duo E8400 @3GHz). For readability reasons, the algorithm is described in Appendix A. Its outcome is represented by the following analytic functions

$$
\begin{aligned}
\dot{\mathbf{J}}_T(s,\dot{s}) &= \mathbf{J}'_T(s)\,\dot{s}\ , && (19)\\
\ddot{\mathbf{J}}_T(s,\dot{s},\ddot{s}) &= \mathbf{J}''_T(s)\,\dot{s}^2 + \mathbf{J}'_T(s)\,\ddot{s}\ . && (20)
\end{aligned}
$$

Equations (16)–(20) permit rewriting (13)–(15) in function of the curvilinear coordinate and of its derivatives. In particular, a few algebraic manipulations lead to the following equations

$$\dot{\mathbf{q}}(s,\dot{s}) = \mathbf{a}(s)\,\dot{s}\ , \tag{21}$$
$$\ddot{\mathbf{q}}(s,\dot{s},\ddot{s}) = \mathbf{b}(s)\,\dot{s}^2 + \mathbf{a}(s)\,\ddot{s}\ , \tag{22}$$
$$\dddot{\mathbf{q}}(s,\dot{s},\ddot{s},\dddot{s}) = \mathbf{c}(s)\,\dot{s}^3 + 3\,\mathbf{b}(s)\,\dot{s}\ddot{s} + \mathbf{a}(s)\,\dddot{s}\ , \tag{23}$$

where

$$\mathbf{a}(s) := \mathbf{J}_T^{-1}(s)\,\widetilde{\mathbf{v}}_T(s)\ , \tag{24}$$
$$\mathbf{b}(s) := \mathbf{J}_T^{-1}(s)[\widetilde{\mathbf{v}}_T'(s) - \mathbf{J}_T'(s)\,\mathbf{a}(s)]\ , \tag{25}$$
$$\mathbf{c}(s) := \mathbf{J}_T^{-1}(s)[\widetilde{\mathbf{v}}_T''(s) - \mathbf{J}_T''(s)\,\mathbf{a}(s) - 2\mathbf{J}_T'(s)\,\mathbf{b}(s)]\ . \tag{26}$$

Equations (16)–(18) and (21)–(26) can now be used to convert the original constraints in the operational space and in the configuration space into equivalent constraints in the curvilinear-coordinate space. Let us consider, for example, constraints (1)–(3) in the joint space. Their scalar components can be written as follows ($i = 1, 2, \dots, N$)

$$\underline{\dot{q}}_i \le \dot{q}_i \le \overline{\dot{q}}_i\,, \tag{27}$$
$$\underline{\ddot{q}}_i \le \ddot{q}_i \le \overline{\ddot{q}}_i\,, \tag{28}$$
$$\underline{\dddot{q}}_i \le \dddot{q}_i \le \overline{\dddot{q}}_i\ . \tag{29}$$

Equations (27)–(29) must evidently be simultaneously satisfied for any $i = 1, 2, \dots, N$. By virtue of (21)–(23), it is possible to rewrite (27)–(29) as follows

$$\underline{\dot{q}}_i \le a_i(s)\,\dot{s} \le \overline{\dot{q}}_i\,, \tag{30}$$
$$\underline{\ddot{q}}_i \le a_i(s)\,\ddot{s} + b_i(s)\,\dot{s}^2 \le \overline{\ddot{q}}_i\,, \tag{31}$$
$$\underline{\dddot{q}}_i \le a_i(s)\,\dddot{s} + 3\,b_i(s)\,\dot{s}\ddot{s} + c_i(s)\,\dot{s}^3 \le \overline{\dddot{q}}_i\,. \tag{32}$$

Equations (30)–(32) can be rearranged in order to convert the original joint bounds into equivalent constraints for the curvilinear coordinate. For example, joint i fulfills (30) if $\dot{s} \in \left[\underline{\zeta}_i, \overline{\zeta}_i\right]$ where $\underline{\zeta}_i$ and $\overline{\zeta}_i$ are given in Table 1. The same inclusion must simultaneously apply for all joints, so that velocity constraint (1) is satisfied only if $\dot{s} \in \bigcap_{i=1}^N \left[\underline{\zeta}_i, \overline{\zeta}_i\right]$. Similar considerations apply for the acceleration and for the jerk constraints, thus equations (31) and (32) are satisfied, and in turn also (2) and (3), only if $\ddot{s} \in \bigcup_{i=1}^N [\underline{\mu}_i, \overline{\mu}_i]$ and

Table 1: Equivalent longitudinal bounds.

	$a_i > 0$	$a_i < 0$	$a_i = 0$
$\overline{\sigma}_i$	$(\dddot{\overline{q}}_i - c_i\dot{s}^3 - 3b_i\dot{s}\ddot{s})/a_i$	$(\dddot{\underline{q}}_i - c_i\dot{s}^3 - 3b_i\dot{s}\ddot{s})/a_i$	∞
$\underline{\sigma}_i$	$(\dddot{\underline{q}}_i - c_i\dot{s}^3 - 3b_i\dot{s}\ddot{s})/a_i$	$(\dddot{\overline{q}}_i - c_i\dot{s}^3 - 3b_i\dot{s}\ddot{s})/a_i$	$-\infty$
$\overline{\mu}_i$	$(\ddot{\overline{q}}_i - b_i\dot{s}^2)/a_i$	$(\ddot{\underline{q}}_i - b_i\dot{s}^2)/a_i$	∞
$\underline{\mu}_i$	$(\ddot{\underline{q}}_i - b_i\dot{s}^2/)a_i$	$(\ddot{\overline{q}}_i - b_i\dot{s}^2)/a_i$	$-\infty$
$\overline{\zeta}_i$	$\dot{\overline{q}}_i/a_i$	$\dot{\underline{q}}_i/a_i$	∞
$\underline{\zeta}_i$	$\dot{\underline{q}}_i/a_i$	$\dot{\overline{q}}_i/a_i$	$-\infty$

	$\widetilde{v}_{T_k} > 0$	$\widetilde{v}_{T_k} < 0$	$\widetilde{v}_{T_k} = 0$
$\overline{\gamma}_k$	$(\overline{j}_k - \widetilde{v}''_{T_k}\dot{s}^3 - 3\widetilde{v}'_{T_k}\dot{s}\ddot{s})/\widetilde{v}_{T_k}$	$(\underline{j}_k - \widetilde{v}''_{T_k}\dot{s}^3 - 3\widetilde{v}'_{T_k}\dot{s}\ddot{s})/\widetilde{v}_{T_k}$	∞
$\underline{\gamma}_k$	$(\underline{j}_k - \widetilde{v}''_{T_k}\dot{s}^3 - 3\widetilde{v}'_{T_k}\dot{s}\ddot{s})/\widetilde{v}_{T_k}$	$(\overline{j}_k - \widetilde{v}''_{T_k}\dot{s}^3 - 3\widetilde{v}'_{T_k}\dot{s}\ddot{s})/\widetilde{v}_{T_k}$	$-\infty$
$\overline{\beta}_k$	$(\overline{a}_k - \widetilde{v}'_{T_k}\dot{s}^2)/\widetilde{v}_{T_k}$	$(\underline{a}_k - \widetilde{v}'_{T_k}\dot{s}^2)/\widetilde{v}_{T_k}$	∞
$\underline{\beta}_k$	$(\underline{a}_k - \widetilde{v}'_{T_k}\dot{s}^2/)\widetilde{v}_{T_k}$	$(\overline{a}_k - \widetilde{v}'_{T_k}\dot{s}^2)/\widetilde{v}_{T_k}$	$-\infty$
$\overline{\alpha}_k$	$\overline{v}_k/\widetilde{v}_{T_k}$	$\underline{v}_k/\widetilde{v}_{T_k}$	∞
$\underline{\alpha}_k$	$\underline{v}_k/\widetilde{v}_{T_k}$	$\overline{v}_k/\widetilde{v}_{T_k}$	$-\infty$

$\dddot{s} \in \bigcup_{i=1}^{N} [\underline{\sigma}_i, \overline{\sigma}_i]$, where terms $\underline{\mu}_i$, $\overline{\mu}_i$, $\underline{\sigma}_i$, and $\overline{\sigma}_i$ can be evaluated through the expressions provided in Table 1.

A similar procedure can be used to transform the limits in the operational space. Equations (4)–(6) make it possible to write ($k = 1, 2, \ldots, 6$)

$$\underline{v}_k \leq \widetilde{v}_{T_k} \leq \overline{v}_k \ , \tag{33}$$

$$\underline{a}_k \leq \widetilde{a}_{T_k} \leq \overline{a}_k \ , \tag{34}$$

$$\underline{j}_k \leq \widetilde{j}_{T_k} \leq \overline{j}_k \ , \tag{35}$$

which, by virtue of (16)–(18), become

$$\underline{v}_k \leq \widetilde{v}_{T_k}(s)\, \dot{s} \leq \overline{v}_k \ , \tag{36}$$

$$\underline{a}_k \leq \widetilde{v}_{T_k}(s)\, \ddot{s} + \widetilde{v}'_{T_k}(s)\, \dot{s}^2 \leq \overline{a}_k \ , \tag{37}$$

$$\underline{j}_k \leq \widetilde{v}_{T_k}(s)\, \dddot{s} + 3\, \widetilde{v}'_{T_k}(s)\, \dot{s}\ddot{s} + \widetilde{v}''_{T_k}(s)\, \dot{s}^3 \leq \overline{j}_k \ . \tag{38}$$

The structure of (36)–(38) is evidently the same of (30)–(32), so that it is possible to assert that the trajectory is feasible if $\dot{s} \in \bigcap_{k=1}^{6} [\underline{\alpha}_k, \overline{\alpha}_k]$, $\ddot{s} \in \bigcap_{k=1}^{6} \left[\underline{\beta}_k, \overline{\beta}_k\right]$, and $\dddot{s} \in \bigcap_{k=1}^{6} \left[\underline{\gamma}_k, \overline{\gamma}_k\right]$, whose lower and upper bounds can be evaluated through the expressions given in Table 1.

Finally, constraints (7)–(9) are directly defined in the curvilinear-coordinate space, thus they do not require any transformation.

In conclusion, time law $s(t)$ is feasible if the following conditions are simultaneously fulfilled

$$\dot{s} \in [R^-, R^+] \,, \tag{39}$$

$$\ddot{s} \in [S^-, S^+] \,, \tag{40}$$

$$\dddot{s} \in [U^-, U^+] \,, \tag{41}$$

where ($i = 1, 2, \ldots, N$, $k = 1, 2, \ldots, 6$)

$$R^- := \max_{i,k}\{\underline{\alpha}_k, \underline{\zeta}_i, \underline{\dot{s}}\}, \quad R^+ := \min_{i,k}\{\overline{\alpha}_k, \overline{\zeta}_i, \overline{\dot{s}}\}, \tag{42}$$

$$S^- := \max_{i,k}\{\underline{\beta}_k, \underline{\mu}_i, \underline{\ddot{s}}\}, \quad S^+ := \min_{i,k}\{\overline{\beta}_k, \overline{\mu}_i, \overline{\ddot{s}}\}, \tag{43}$$

$$U^- := \max_{i,k}\{\underline{\gamma}_k, \underline{\sigma}_i, \underline{\dddot{s}}\}, \quad U^+ := \min_{i,k}\{\overline{\gamma}_k, \overline{\sigma}_i, \overline{\dddot{s}}\}. \tag{44}$$

Remark 1. *For conciseness reasons, this work only considers kinematic constraints, but the approach can be easily extended in order to include more* *limits. For example, dynamic constraints can be accounted for by recalling that generalized torques* $\boldsymbol{\tau}$ *can be expressed as follows*

$$\boldsymbol{\tau}(s) = \mathbf{d}(s)\,\dot{s}^2 + \mathbf{e}(s)\,\ddot{s} + \mathbf{g}(s)\ , \tag{45}$$

where $\mathbf{d}(s)$*,* $\mathbf{e}(s)$*, and* $\mathbf{g}(s)$ *are proper functions derived from the dynamic model and from (21)–(26) (see [17, 20, 21]). The structure of (45) is similar to the one of (22), so that the classic torque constraint*

$$\underline{\boldsymbol{\tau}} \le \boldsymbol{\tau} \le \overline{\boldsymbol{\tau}}$$

can be converted, with obvious manipulations, into further bounds on $\ddot{s}$ *to be added to (43). Analogously, torque-derivative constraints generate additional limits for (44) (see [20]).*

The knowledge of the equivalent bounds represents the key point for the solution of *Problem 1*. Indeed, as explained in Section 2, once the equivalent constraints are known, the feasibility is maintained by modifying the LTL through the nonlinear filter proposed in [34]. Roughly speaking, the filter approximates any possibly unfeasible $s_d(t)$ with a signal $s(t)$ which represents its best feasible approximation. The nominal LTL is changed by means of minimum-time transients. Evidently, this implicitly implies that anytime $s(t)$ is feasible the filter gains the tracking condition, i.e., $s(t) = s_d(t)$, in minimum time and with feasible transients, thus eliminating any possible delay that has been accumulated to preserve feasibility.

The scaling mechanism proposed in this paper is subject to the same feasibility issues already pointed out in [29] for trajectories in the configuration space. The problem can be summarized as follows: depending on the status of motion of the system, interval $[S^-, S^+]$ could become very small. Obviously, when this happens the dynamics of the speed signal is strongly limited, so that such signal can only be slowly changed. Unfortunately, the velocity bounds, i.e., R^- and R^+, are not constant: if they change too rapidly the feasibility is lost since the available dynamics is not sufficient to permit fast speed changes. A similar problem arises for the acceleration signal when interval $[U^-, U^+]$ vanishes. In [29] a solution was provided for trajectories in the configuration space: roughly speaking, if $[S^-, S^+]$ and/or $[U^-, U^+]$ become too small, it is possible to invert such trend by reducing the longitudinal speed before any problem could occur. This result was achieved by properly downscaling velocity bound R^+ (more details on that strategy can be found in [29]). Unfortunately, preliminary tests have revealed that such technique is not suited for trajectories in the operational space since acceleration and jerk feasibility intervals are normally well open along the whole trajectory, then they suddenly collapse to zero when the system is very close to the critical point. Practically, the available time is not sufficient for any reaction: the bound scaling method proposed in [29] can only be used in uncritical zones to manage reference signals characterized by too demanding longitudinal speeds.

For this reason, further investigations have been driven toward other directions. Techniques based on the analysis of the singular values of the Jacobian matrix associated to the end effector have given good results, but due to their computational burden the final choice has fallen on an alternative method. The singularities of a 6R anthropomorphic manipulator can be predicted by means of simple checks on the joint variables. For example, anytime the manipulator approaches a wrist singularity, the fifth joint variable, i.e., q_5, goes to zero. In that case, bounds R^- and R^+ can be scaled as follows

$$\left[\widetilde{R}^-,\ \widetilde{R}^+\right] = \begin{cases} \left[R^-,\ \psi R^+\right], & \text{if } \dot{s} \geq 0 \\ \left[\psi R^-,\ R^+\right], & \text{otherwise} \end{cases}, \tag{46}$$

where ψ is given by

$$\psi = \begin{cases} \text{sat}\left[1 + \frac{|\dot{s}|}{v_R}\left(\frac{|q_5|}{\overline{q}_5} - 1\right)\right], & \text{if } |q_5| < \overline{q}_5 \\ 1, & \text{otherwise} \end{cases}. \tag{47}$$

Function $\text{sat}(\cdot)$ saturates its output between 0 and 1.

Threshold $\overline{q}_5$ is chosen such to guarantee a reasonably anticipated reaction of the system, while v_R coincides with the maximum longitudinal speed that can be admitted for the considered application: more incisive reactions are provided as $\dot{s}$ approaches v_R. The new bounds, i.e., $\widetilde{R}^-$ and $\widetilde{R}^+$, replace R^- and R^+, so that (39) becomes

$$\dot{s} \in \left[\widetilde{R}^-, \widetilde{R}^+\right] .$$

It is important to mention that analogous detection methods, still based on the use of appropriate joint variables (more precisely on q_2 and on q_3), can be adopted to manage elbow or shoulder singularities, i.e., the target of the proposed scaling strategy is not specifically represented by wrist singularities.

It is worth highlighting that in this paper, for coherence with the real-time specification, any decision is taken on the basis of the current status of the system, so that scaling term ψ is evaluated on the basis of the current value of q_5. In order to increase the method robustness, further studies, in which such requirement has been partially dropped, are being currently carried out. Practically, by preliminary inspecting the path a few steps ahead, looking for critical configurations, it is possible to almost eliminate the possibility to be entrapped in unfeasible configurations.

4. Experimental results

The experimental tests proposed in this section have been mainly conceived to compare the performances of the Jerk Constrained TSS (JC-TSS) with those that can be achieved with its previous release proposed in [35], i.e., with the Acceleration Constrained TSS (AC-TSS) and with those achievable with the commercial controller of a Comau Smart Six 6-1.4 manipulator. To this purpose, the trajectory planner of the commercial controller has been initially replaced with a novel reference generator, alternatively equipped with one of the two TSSs. Conversely, joint motors have been driven with the control loops of the commercial controller. The trajectory updating time is equal to $2 \cdot 10^{-3}$ s.

The first comparisons concern the execution of a linear trajectory from $\mathbf{p}_A = [0.25\ 0.83\ 1.095]^T$ to $\mathbf{p}_B = [-0.25\ 0.83\ 1.095]^T$, which passes close to a wrist singularity located at $\overline{\mathbf{p}} = [0\ 0.83\ 1.07765]^T$ and that, consequently, requires unfeasible velocities for joints 4 and 6.

The nominal LTL is given by a step signal defined as follows

$$s_d(t) := \begin{cases} 0\text{ m} & t = 0 \\ 0.5\text{ m} & t > 0 \end{cases},$$

$\dot{s}_d(t) = 0$ ms^{-1}, $\ddot{s}_d(t) = 0$ ms^{-2}, and $\dddot{s}_d(t) = 0$ ms^{-3}. Both planners react to such signal by generating almost minimum-time trajectories which are compatible with the user defined limits. For the problem at hand, the upper and lower bounds for (1)–(6) are reported in Table 2, while the limits for (7)–(9) are given by $\underline{\dot{s}} = -0.4$ ms^{-1}, $\overline{\dot{s}} = 0.4$ ms^{-1}, $\underline{\ddot{s}} = -15$ ms^{-2}, $\overline{\ddot{s}} = 15$ ms^{-2}, $\underline{\dddot{s}} = -1000$ ms^{-3}, $\overline{\dddot{s}} = 1000$ ms^{-3} (jerk bounds only apply to the JC-TSS). As explained in Section 3 the assigned bounds are online converted into equivalent limits for the LTL through (39)–(44) and Table 1, with the

Table 2: Kinematic limits that have been used for the experiments. Jerk bounds only apply to the JC-TSS.

$\underline{\dot{\mathbf{q}}}$ (rad s^{-1})	$[-10\ -10\ -10\ -7.5\ -10\ -9]^T$
$\overline{\dot{\mathbf{q}}}$ (rad s^{-1})	$[10\ 10\ 10\ 7.5\ 10\ 9]^T$
$\underline{\ddot{\mathbf{q}}}$ (rad s^{-2})	$[-100\ -100\ -100\ -100\ -100\ -100]^T$
$\overline{\ddot{\mathbf{q}}}$ (rad s^{-2})	$[100\ 100\ 100\ 100\ 100\ 100]^T$
$\underline{\dddot{\mathbf{q}}}$ (rad s^{-3})	$[-4000\ -4000\ -4000\ -4000\ -4000\ -4000]^T$
$\overline{\dddot{\mathbf{q}}}$ (rad s^{-3})	$[4000\ 4000\ 4000\ 4000\ 4000\ 4000]^T$
$\underline{\mathbf{v}}_T$ (m s^{-1})	$[-0.4\ -0.4\ -0.4\ -10\ -10\ -10]^T$
$\overline{\mathbf{v}}_T$ (m s^{-1})	$[0.4\ 0.4\ 0.4\ 10\ 10\ 10]^T$
$\underline{\mathbf{a}}_T$ (m s^{-2})	$[-5\ -5\ -5\ -100\ -100\ -100]^T$
$\overline{\mathbf{a}}_T$ (m s^{-2})	$[5\ 5\ 5\ 100\ 100\ 100]^T$
$\underline{\mathbf{j}}_T$ (m s^{-3})	$[-400\ -400\ -400\ -1000\ -1000\ -1000]^T$
$\overline{\mathbf{j}}_T$ (m s^{-3})	$[400\ 400\ 400\ 1000\ 1000\ 1000]^T$

sole exception of R^- that has been forced to 0 in order to prevent backward movements.

The output of the AC-TSS is shown in Fig. 2. Since transients are minimum-time, the acceleration suddenly assumes its maximum value and the velocity increases until its upper limit is reached. During this phase the motion is essentially limited by the constraints in the operational space. Fig. 3 shows the same transient achieved with the JC-TSS. The motion is still minimum-time and the cruising speed is reached with a bang-zero-bang jerk signal. Transient time is evidently longer due to the imposed continuity on the acceleration.

In the neighborhood of the singular point, bounds on joint velocities and accelerations become dominant, so that the longitudinal speed is reduced in order to maintain the feasibility. Figs. 2 and 3 make it possible to appreciate the different behaviors of the two scaling systems. In particular, while the AC-TSS shows an evident chattering on the acceleration, the JC-TSS admits much smoother transients. Fig. 3 is also useful to understand the importance of the downscaling action on R^+, obtained by means of (46) and (47) and by assuming $v_R = 0.4$ ms^{-1} and $\overline{q}_5 = 0.1$ rad. The JC-TSS, indeed, owing to the acceleration continuity, normally reacts less promptly than the AC-TSS to sudden changes of the velocity limits and, consequently, constraint violations could occur more easily. Equation (46) anticipates critical situations by reducing in advance upper bound R^+. It is worth to mention that also the AC-TSS could benefit from the use of the same downscaling strategy since, as shown in Fig. 2, it can be subject to minor constraint violations.

The fulfillment of the equivalent constraints implies that original bounds (1)–(6) are satisfied as well. In order to prove this assertion, the reference signals generated by the two planners for joints 4 and 6, i.e., the most solicited joints, are compared in Fig. 4 with the corresponding user-defined bounds. Evidently, both scaling systems guarantee that the assigned constraints are fulfilled, even if minor violations can be detected for the AC-TSS (see the dash-dotted areas in Fig. 4): they are caused by the above mentioned violations of the equivalent bounds. Another difference between the two TSSs can be noticed by observing the acceleration signals: the AC-TSS shows an evident chattering, which can be potentially troublesome depending on the characteristics of the manipulator controller.

As known, the characteristics of reference signals affect the controller performances. It is possible to get an idea of the different behaviors that can be achieved when using each one of the two TSSs by considering the trajectory tracking errors in the operational space. To this purpose, the Cartesian reference signal has been compared with the actual position of the end-effector, derived from the joint encoders through the direct kinematics. Fig. 5 shows the tracking errors for the z axis, i.e., along the vertical axis, while tracking errors along the x and the y axes have not been reported being negligible. At the beginning and at the end of each trajectory the two TSSs show similar performances, while differences become significant in the neighborhood of the singularity: the smoother accelerations which characterize the JC-TSS transients guarantee better control loop performances and, in turn, tracking errors of reduced amplitude. The numerical comparisons proposed in Table 3 further confirm that smaller tracking errors can be achieved by means of the JC-TSS. It is worth mentioning that data shown in Fig. 5 and in Table 3 are consistent since the manipulator is equipped with 24 bit encoders and has a repeatability equal to $5 \cdot 10^{-5}$ m.

The solicitations acting on the payload have been verified by means of a further experiment. The end-effector of the manipulator has been equipped with an accelerometer in order to measure the structural vibrations. Solicitations along the y and the z axes are negligible, so that they have not

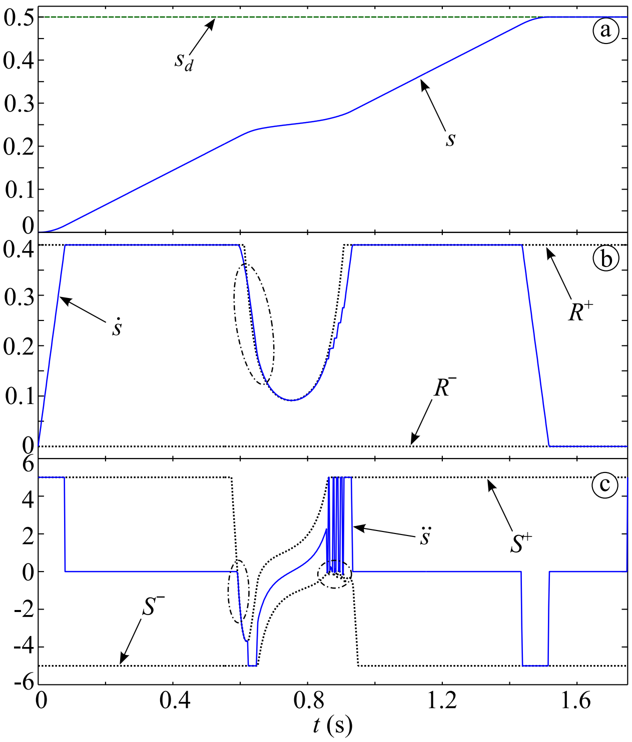

Figure 2: Outputs of the AC-TSS compared with the equivalent longitudinal bounds: (a) Longitudinal time law $s(t)$; (b) Longitudinal velocity $\dot{s}(t)$; (c) Longitudinal acceleration $\ddot{s}(t)$. Dash-dotted lines point out minor constraints violations.

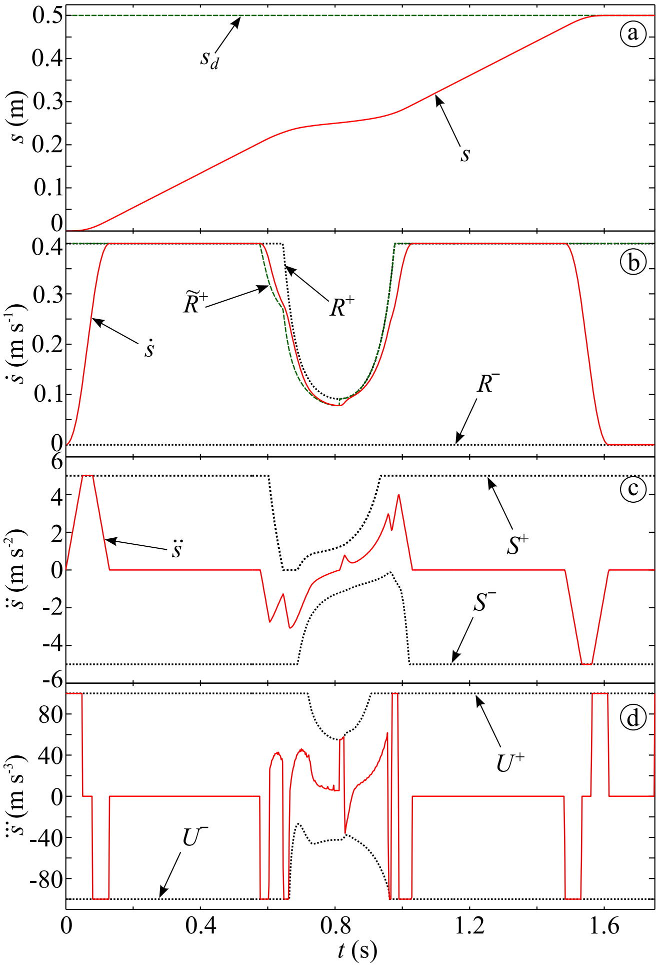

Figure 3: Outputs of the JC-TSS compared with the equivalent longitudinal bounds: (a) Longitudinal time law $s(t)$; (b) Longitudinal velocity $\dot{s}(t)$; (c) Longitudinal acceleration $\ddot{s}(t)$; (d) Longitudinal jerk $\dddot{s}(t)$.

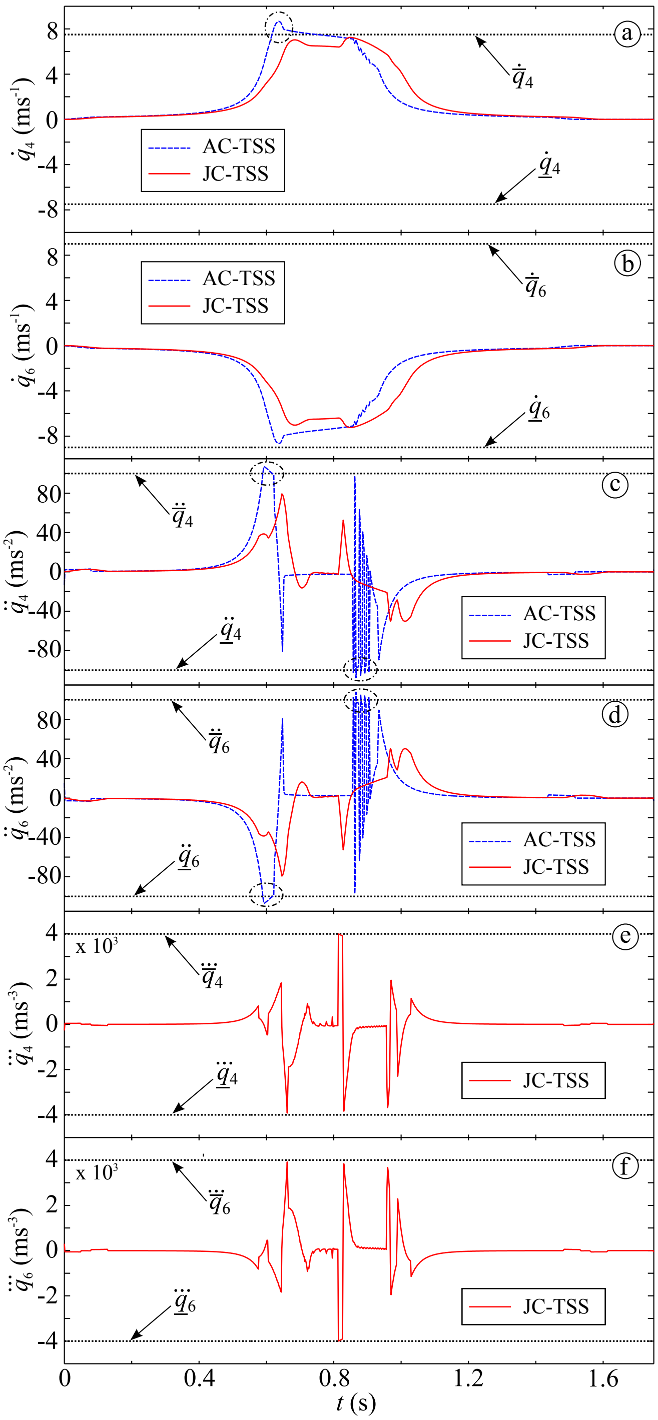

Figure 4: Joint reference signals generated by the AC-TSS and by the JC-TSS for joints 4 and 6: (a) and (b) Joint velocities; (c) and (d) Joint accelerations; (e) and (f) Joint jerks.

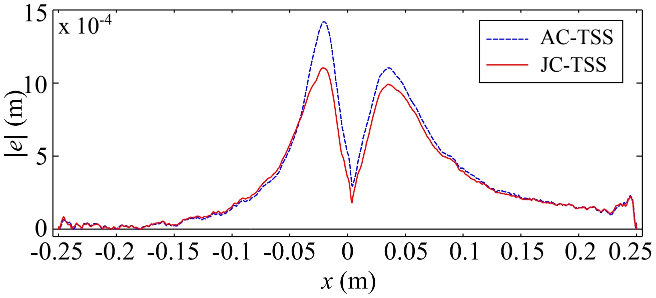

Figure 5: Trajectory tracking errors along the x axis.

Table 3: Statistics on trajectory tracking errors.

	AC-TSS	JC-TSS
e_{max} (m)	$1.44 \cdot 10^{-3}$	$1.10 \cdot 10^{-3}$
$E\,[\,\lvert e\rvert\,]$ (m)	$3.20 \cdot 10^{-4}$	$3.01 \cdot 10^{-4}$
variance (m^2)	$13.14 \cdot 10^{-8}$	$9.61 \cdot 10^{-8}$
e_{rms} (m)	$4.84 \cdot 10^{-4}$	$4.32 \cdot 10^{-4}$
$\int_T \lvert e\rvert$ (m)	$2.86 \cdot 10^{-1}$	$2.69 \cdot 10^{-1}$

been reported. From Fig. 6 it is possible to evince that differences between the two TSSs are generally negligible, but they become evident close to the singularity, where the AC-TSS shows much more evident oscillations in the range 12–16 Hz. This detail is also confirmed by Fig. 7, which compares the spectra of the signals acquired in the neighborhood of the singularity. In particular, it highlights that, in the range of frequencies previously mentioned, the AC-TSS produces resonant harmonics with higher peaks.

The last set of experiments has been conceived to verify how close to a singular point it is possible to pass with the aid of the two scaling systems. In order to provide realistic operating conditions, the robot starts moving immediately after it receives the path parameters: any decision is taken by the system during the motion, i.e., in real time. An initial trajectory is executed in the xz-plane sufficiently far from the singularity. If feasibility is not violated, a parallel one, 10^{-3} m closer to the singularity, is generated. The approaching process continues until feasibility is lost: the minimum distance from the singularity of the last feasible trajectory is acquired and then reported in Fig. 8. The procedure is then repeated for another set of parallel trajectories still lying in the xz-plane, but with a different slope m. The supplementary video file shows two approaching procedures for $m = 0$

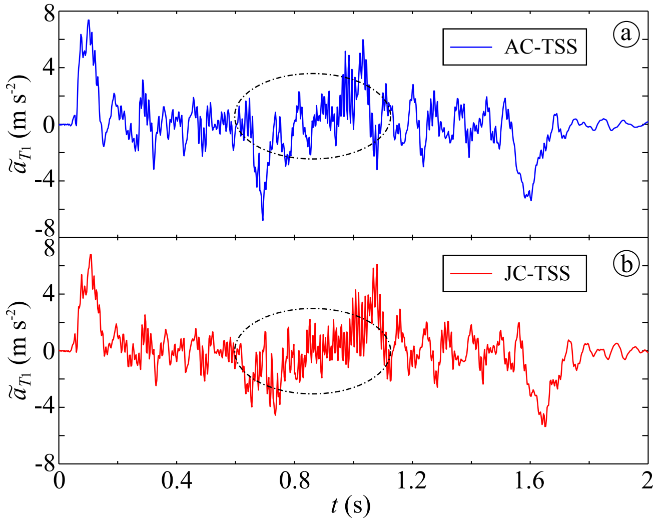

Figure 6: Accelerations measured along the x axis. During the trajectory scaling phase, the AC-TSS excites self-oscillations in the range from 12 to 20 Hz (see the area surrounded by the dash-dotted lines).

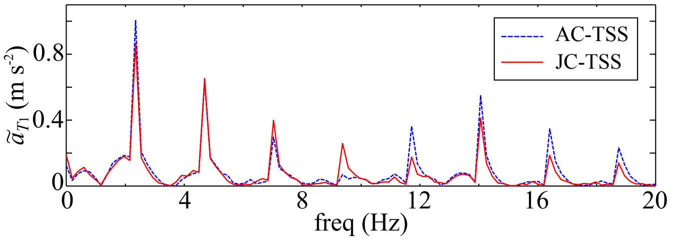

Figure 7: Frequency spectra of the acceleration signals (x component) in the area close the singularity, i.e., during the periods in which the two TSSs modify the trajectory. The peaks of the AC-TSS are generally higher than the ones of the JC-TSS.

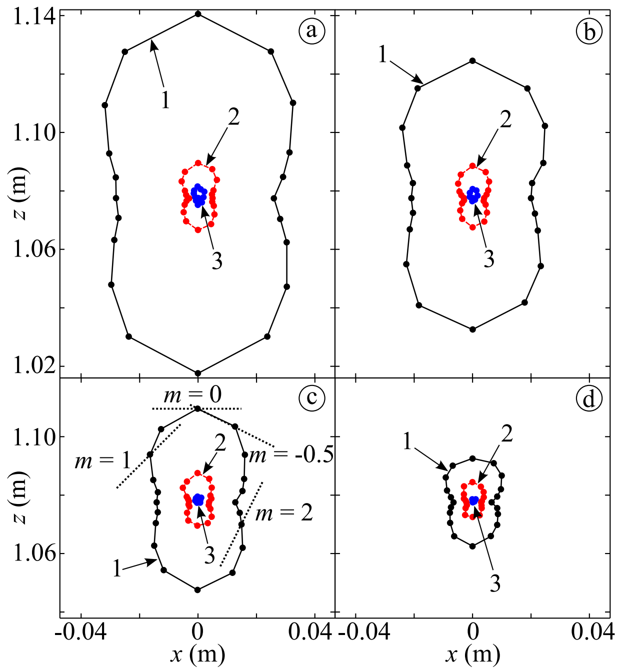

Figure 8: Minimum distance points from singularity achieved by executing linear segments and for different slopes m. Points have been obtained (1) without any TSS, (2) with the JC-TSS, and (3) with the AC-TSS. Experiments refer to different longitudinal speeds: (a) 0.4 m s^{-1}; (b) 0.3 m s^{-1}; (c) 0.2 m s^{-1}; (d) 0.1 m s^{-1}.

and for $m = 0.5$ obtained by means of the JC-TSS (in order to shorten the video length, the distance between two parallel paths has been posed equal to 10^{-2} m). The experiment has been executed by using 1) the commercial planner, 2) the JC-TSS, and 3) the AC-TSS. All tests have been repeated for different longitudinal speeds for a total of more than 8000 real-time trajectories. Fig. 8 summarizes the achieved results. The use of the two TSSs evidently permits closer approaches to the singularity. Improvements are especially evident at the highest speeds. Notice that the AC-TSS, owing to the discontinuity of the acceleration signal, reacts more promptly to critical configurations, so that feasibility is preserved even in areas that cannot be reached with the aid of the JC-TSS. Conversely, the JC-TSS can modify the assigned trajectories by means of very smooth transients, as can be evinced from the supplementary video file. Independently from the approaching direction, Cartesian tracking errors always assume shapes and maximum values that are very similar to the ones shown in Fig. 5. Such behavior can be easily justified. Close to the singularity, in all the 20 cases joints 4 and 6 reach, and maintain for a significant time-interval, their respective maximum speeds. Tracking errors are proportional to the joint speeds, so that their maximum values in the joint space are practically the same in all the experiments, while minor differences can be verified in the Cartesian space owing to the nonlinear relationships existing between the two spaces.

The two scaling methods have been also compared in terms of algorithmic efficiency by using an Intel Core2 Duo E8400 @3GHz processor. Indeed, differently from the common feeling, the CPU time still represents a precious resource, so that novel algorithms can be embedded in existing control architectures only if their execution times are short and known. Such characteristics are common to both planners. The AC-TSS does not use any iterative procedure, so that its evaluation time is practically constant: for the six degrees of freedom manipulator used in the experiments the whole planning system shown in Fig. 1 is executed in $3.48 \cdot 10^{-5}$ s. Evaluation times slightly increase for the JC-TSS, since its scaling filter uses an iterative procedure. However, its overall computational time is almost constant being equal, on average, to $4.98 \cdot 10^{-5}$ s with a standard deviation equal to $2.797 \cdot 10^{-6}$ s. The computational burden of both systems is compatible with the sampling times of many real-time applications, even considering processors that are less powerful than the one used for the experiments. It is worth to highlight that, despite the JC-TSS is much more complex than its precursor, their computational times are comparable. This is possible thanks to the efficient procedure used for the evaluation of $\dot{\mathbf{J}}_T(s, \dot{s})$ and of $\ddot{\mathbf{J}}_T(s, \dot{s}, \ddot{s})$.

5. Final discussion and conclusions.

The validation tests proposed in Section 4 make it possible to assert that the performances of two TSSs are, in any case, better than those achievable with the commercial controller, since closer approaches to the singularities are permitted. The mutual comparison between the two TSSs permits drawing further conclusions. The behaviors of the two scaling systems are clearly different for high speeds and close to singular points: the AC-TSS is generally characterized by a higher reactivity, while the JC-TSS induces lower mechanical stresses on the structure and causes smaller tracking errors and vibrations. Performances become similar at low speeds. This behavior is mainly justified by the characteristics of the position controller of the Comau Smart SiX manipulator, which only accepts reference signals given by joint angles and by their derivatives: the acceleration signals provided by the two TSSs are not used, so that the acceleration discontinuities, which are typical of the AC-TSS, produce almost negligible effects on the system. In case of inverse dynamics controllers or, more in general, of controllers which use the acceleration signal for feedforward actions, the situation could totally change: discontinuous accelerations are indeed converted into discontinuous signals for the inner torque controller which, in turn, excite the elastic modes, thus generating undesired vibrations. Oscillatory behaviors can thus be better mitigated by the JC-TSS, which intrinsically bounds the variability of the acceleration signal.

In conclusion, the JC-TSS represents the most promising strategy when a smooth approach to singularities is required, while the AC-TSS is still the best solution when trajectories must pass very close to singularities.

Appendix A. An efficient evaluation of $\ddot{\mathbf{J}}_T(s, \dot{s}, \ddot{s})$

Let us draw some preliminary considerations that are instrumental for the synthesis of the jacobian derivatives. As known, for a manipulator whose frames are assigned according to the modified Denavit-Hartenberg procedure, the jacobian matrix associated to tool frame T – which is located on the Nth link and, consequently, it is rigidly connected to frame N – has the following structure

$$\mathbf{J}_T = \left[\begin{array}{c} \mathbf{J}_{v_T} \\ \hline \mathbf{J}_{\omega_T} \end{array}\right] = \left[\begin{array}{c|c|c|c} \mathbf{j}_T^{v_1} & \mathbf{j}_T^{v_2} & \cdots & \mathbf{j}_T^{v_N} \\ \hline \mathbf{j}_T^{\omega_1} & \mathbf{j}_T^{\omega_2} & \cdots & \mathbf{j}_T^{\omega_N} \end{array}\right] ,$$

where $(k = 1, 2, \ldots, N)$

$$\mathbf{j}_T^{v_k} = \begin{cases} \hat{\mathbf{z}}_k & \text{prismatic joint} \\ \hat{\mathbf{z}}_k \times [\mathbf{p}_T - \mathbf{p}_k] & \text{revolute joint} \end{cases}, \tag{A.1}$$

$$\mathbf{j}_T^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \hat{\mathbf{z}}_k & \text{revolute joint} \end{cases}, \tag{A.2}$$

and where $\mathbf{p}_T$ and $\mathbf{p}_k$ are, respectively, the positions of the tool frame and of the kth frame w.r.t. frame 0. $\hat{\mathbf{z}}_k$ is the $\hat{\mathbf{z}}$ unit vector of frame k, described w.r.t. frame 0 and can be evaluated as follows: $\hat{\mathbf{z}}_k = {}^0_k\mathbf{R}\,\hat{\mathbf{z}}^*$, where $\hat{\mathbf{z}}^* = [0\ 0\ 1]^T$.

Similarly, the Jacobian matrix of the generic ith link frame has the following structure

$$\mathbf{J}_i := \left[\begin{array}{c} \mathbf{J}_{v_i} \\ \hline \mathbf{J}_{\omega_i} \end{array}\right] = \left[\begin{array}{c|c|c|c|c|c|c} \mathbf{j}_i^{v_1} & \mathbf{j}_i^{v_2} & \cdots & \mathbf{j}_i^{v_i} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \mathbf{j}_i^{\omega_1} & \mathbf{j}_i^{\omega_2} & \cdots & \mathbf{j}_i^{\omega_i} & \mathbf{0} & \cdots & \mathbf{0} \end{array}\right],$$

where $(k = 1, 2, \ldots, i)$

$$\mathbf{j}_i^{v_k} = \begin{cases} \hat{\mathbf{z}}_k & \text{prismatic joint} \\ \hat{\mathbf{z}}_k \times [\mathbf{p}_i - \mathbf{p}_k] & \text{revolute joint} \end{cases}, \tag{A.3}$$

$$\mathbf{j}_i^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \hat{\mathbf{z}}_k & \text{revolute joint} \end{cases}. \tag{A.4}$$

Evidently, by comparing (A.2) with (A.4) it is immediately possible to conclude that, for any element $\mathbf{j}_i^{\omega_k}$ which is different from zero, $\mathbf{j}_i^{\omega_k} = \mathbf{j}_T^{\omega_k}$. Conversely, $\mathbf{j}_i^{v_k} = \mathbf{j}_T^{v_k}$ only holds, according to (A.1) and (A.3), for prismatic joints and still considering elements $\mathbf{j}_i^{v_k}$ which are different from zero.

Similar expressions can also be proposed for the evaluation of the jacobian derivatives. For example, $\dot{\mathbf{J}}_T$ has the following structure

$$\dot{\mathbf{J}}_T := \left[\begin{array}{c} \dot{\mathbf{J}}_{v_T} \\ \hline \dot{\mathbf{J}}_{\omega_T} \end{array}\right] = \left[\begin{array}{c|c|c|c} \dot{\mathbf{j}}_T^{v_1} & \dot{\mathbf{j}}_T^{v_2} & \cdots & \dot{\mathbf{j}}_T^{v_N} \\ \hline \dot{\mathbf{j}}_T^{\omega_1} & \dot{\mathbf{j}}_T^{\omega_2} & \cdots & \dot{\mathbf{j}}_T^{\omega_N} \end{array}\right],$$

where $(k = 1, 2, \ldots, N)$

$$\dot{\mathbf{j}}_T^{v_k} = \begin{cases} \dot{\hat{\mathbf{z}}}_k & \text{prismatic joint} \\ \dot{\hat{\mathbf{z}}}_k \times [\mathbf{p}_T - \mathbf{p}_k] + \hat{\mathbf{z}}_k \times [\mathbf{v}_T - \mathbf{v}_k] & \text{revolute joint} \end{cases}, \tag{A.5}$$

$$\dot{\mathbf{j}}_T^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \dot{\hat{\mathbf{z}}}_k & \text{revolute joint} \end{cases} . \tag{A.6}$$

The analogous matrix $\dot{\mathbf{J}}_i$, associated to the generic ith frame, admits the following, representation

$$\dot{\mathbf{J}}_i := \left[\begin{array}{c} \dot{\mathbf{J}}_{v_i} \\ \hline \dot{\mathbf{J}}_{\omega_i} \end{array} \right] = \left[\begin{array}{c|c|c|c|c|c|c} \dot{\mathbf{j}}_i^{v_1} & \dot{\mathbf{j}}_i^{v_2} & \cdots & \dot{\mathbf{j}}_i^{v_i} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \dot{\mathbf{j}}_i^{\omega_1} & \dot{\mathbf{j}}_i^{\omega_2} & \cdots & \dot{\mathbf{j}}_i^{\omega_i} & \mathbf{0} & \cdots & \mathbf{0} \end{array} \right] ,$$

where $(k = 1, 2, \ldots, i)$

$$\dot{\mathbf{j}}_i^{v_k} = \begin{cases} \dot{\hat{\mathbf{z}}}_k & \text{prismatic joint} \\ \dot{\hat{\mathbf{z}}}_k \times [\mathbf{p}_i - \mathbf{p}_k] + \hat{\mathbf{z}}_k \times [\mathbf{v}_i - \mathbf{v}_k] & \text{revolute joint} \end{cases} ,$$

$$\dot{\mathbf{j}}_i^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \dot{\hat{\mathbf{z}}}_k & \text{revolute joint} \end{cases} .$$

The considerations that were made for the Jacobian matrix also apply to its first derivative, so that for the elements of $\dot{\mathbf{J}}_i$ that are different from 0, it is always possible to assume $\dot{\mathbf{j}}_i^{\omega_k} = \dot{\mathbf{j}}_T^{\omega_k}$, while $\dot{\mathbf{j}}_i^{v_k} = \dot{\mathbf{j}}_T^{v_k}$ is only true for prismatic joints.

Finally, $\ddot{\mathbf{J}}_T$ can be written as follows

$$\ddot{\mathbf{J}}_T := \left[\begin{array}{c} \ddot{\mathbf{J}}_{v_T} \\ \hline \ddot{\mathbf{J}}_{\omega_T} \end{array} \right] = \left[\begin{array}{c|c|c|c} \ddot{\mathbf{j}}_T^{v_1} & \ddot{\mathbf{j}}_T^{v_2} & \cdots & \ddot{\mathbf{j}}_T^{v_N} \\ \hline \ddot{\mathbf{j}}_T^{\omega_1} & \ddot{\mathbf{j}}_T^{\omega_2} & \cdots & \ddot{\mathbf{j}}_T^{\omega_N} \end{array} \right] ,$$

where $(k = 1, 2, \ldots, N)$

$$\ddot{\mathbf{j}}_T^{v_k} = \begin{cases} \dot{\hat{\mathbf{z}}}_k & \text{prismatic joint} \\ \ddot{\hat{\mathbf{z}}}_k \times [\mathbf{p}_T - \mathbf{p}_k] + 2\, \dot{\hat{\mathbf{z}}}_k \times [\mathbf{v}_T - \mathbf{v}_k] & \\ \qquad + \hat{\mathbf{z}}_k \times [\mathbf{a}_T - \mathbf{a}_k] & \text{revolute joint} \end{cases} ,$$

$$\ddot{\mathbf{j}}_T^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \ddot{\hat{\mathbf{z}}}_k & \text{revolute joint} \end{cases} ,$$

and, analogously, $\ddot{\mathbf{J}}_i$ has the following representation

$$\ddot{\mathbf{J}}_i := \left[\begin{array}{c} \ddot{\mathbf{J}}_{v_i} \\ \hline \ddot{\mathbf{J}}_{\omega_i} \end{array} \right] = \left[\begin{array}{c|c|c|c|c|c|c} \ddot{\mathbf{j}}_i^{v_1} & \ddot{\mathbf{j}}_i^{v_2} & \cdots & \ddot{\mathbf{j}}_i^{v_i} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \ddot{\mathbf{j}}_i^{\omega_1} & \ddot{\mathbf{j}}_i^{\omega_2} & \cdots & \ddot{\mathbf{j}}_i^{\omega_i} & \mathbf{0} & \cdots & \mathbf{0} \end{array} \right] ,$$

where $(k = 1, 2, \ldots, i)$

$$\ddot{\mathbf{j}}_i^{v_k} = \begin{cases} \dot{\hat{\mathbf{z}}}_k & \text{prismatic joint} \\ \ddot{\hat{\mathbf{z}}}_k \times [\mathbf{p}_i - \mathbf{p}_k] + 2\,\dot{\hat{\mathbf{z}}}_k \times [\mathbf{v}_i - \mathbf{v}_k] + \hat{\mathbf{z}}_k \times [\mathbf{a}_i - \mathbf{a}_k] \\ & \text{revolute joint} \end{cases} \tag{A.7}$$

$$\ddot{\mathbf{j}}_i^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \ddot{\hat{\mathbf{z}}}_k & \text{revolute joint} \end{cases} \quad . \tag{A.8}$$

Again, for any element of $\ddot{\mathbf{J}}_i$ which is different from zero we have $\ddot{\mathbf{j}}_i^{\omega_k} = \ddot{\mathbf{j}}_T^{\omega_k}$, while $\ddot{\mathbf{j}}_i^{v_k} = \ddot{\mathbf{j}}_T^{v_k}$ only applies for prismatic joints.

The expressions that have been proposed for the evaluation of the Jacobians and of their derivatives can be used every time trajectories are defined in the time domain. The planning scheme that is used in this paper assumes that trajectories are assigned according to the path-velocity decomposition. Consequently, alternative computational methods must be developed and, since Jacobians must be evaluated in real time, they must be efficient. The solution originally proposed in [35] for the evaluation of $\mathbf{J}_T(s)$ and $\dot{\mathbf{J}}_T(s, \dot{s})$ is briefly summarized in the following and it is then extended in order to efficiently compute $\ddot{\mathbf{J}}_T(s, \dot{s}, \ddot{s})$.

The terms of $\mathbf{J}_T(s)$ can be immediately derived from (A.1) and (A.2), so that for prismatic joints it is possible to assume

$$\mathbf{j}_T^{\omega_k}(s) := \mathbf{0} \ , \tag{A.9}$$

$$\mathbf{j}_T^{v_k}(s) := \hat{\mathbf{z}}_k(s) \ , \tag{A.10}$$

while for revolute joints we have

$$\mathbf{j}_T^{\omega_k}(s) := \hat{\mathbf{z}}_k(s) \ , \tag{A.11}$$

$$\mathbf{j}_T^{v_k}(s) := \mathbf{j}_T^{\omega_k}(s) \times \Delta_k(s) \ , \tag{A.12}$$

where $\hat{\mathbf{z}}_k(s) := {}^0_k\mathbf{R}(s)\,\hat{\mathbf{z}}^*$ and

$$\Delta_k(s) := \mathbf{p}_T(s) - \mathbf{p}_k(s) \ . \tag{A.13}$$

Evidently, all the terms that are required for the evaluation of $\mathbf{J}_T(s)$ can be directly obtained from the path definition, i.e., from the knowledge of $\mathbf{p}_T(s)$ and ${}^0_T\mathbf{R}(s)$.

The synthesis of the terms which compose $\dot{\mathbf{J}}_T$ requires some more steps (details have been omitted for conciseness, but can be found in [35]). In particular, it is possible to prove that the following equation applies for revolute joints

$$\dot{\mathbf{z}}_k(s,\dot{s}) = \dot{s}\,\mathbf{j}'^{\omega_k}_T(s) = \dot{s}[\mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}^{\omega_k}_T(s)]\ , \tag{A.14}$$

while for prismatic joints we have

$$\dot{\mathbf{z}}_k(s,\dot{s}) = \dot{s}\,\mathbf{j}'^{v_k}_T(s) = \dot{s}[\mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}^{v_k}_T(s)]\ . \tag{A.15}$$

Moreover, it descends from (10) that the tool velocity can be obtained from

$$\mathbf{v}_T(s,\dot{s}) := \mathbf{J}_{v_T}(s)\,\dot{\mathbf{q}}(s,\dot{s})\,, \tag{A.16}$$

and that, analogously, joint velocities are given by

$$\mathbf{v}_k(s,\dot{s}) := \mathbf{J}_{v_k}(s)\,\dot{\mathbf{q}}(s,\dot{s})\,. \tag{A.17}$$

Closed form equations for $\dot{\mathbf{j}}^{\omega_k}_T(s,\dot{s})$ and $\dot{\mathbf{j}}^{v_k}_T(s,\dot{s})$ can be obtained from (A.5) and (A.6) by considering (21), (A.11), and (A.13)–(A.17). In particular, after a few algebraic manipulations, it is possible to write

$$\dot{\mathbf{j}}^{\omega_k}_T(s,\dot{s}) := \dot{s}\,\mathbf{j}'^{\omega_k}_T(s)\,, \tag{A.18}$$

$$\dot{\mathbf{j}}^{v_k}_T(s,\dot{s}) := \dot{s}\,\mathbf{j}'^{v_k}_T(s)\,, \tag{A.19}$$

where, for prismatic joints,

$$\mathbf{j}'^{\omega_k}_T(s) := \mathbf{0}\,, \tag{A.20}$$

$$\mathbf{j}'^{v_k}_T(s) := \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}^{v_k}_T(s)\,, \tag{A.21}$$

while, for revolute joints,

$$\mathbf{j}'^{\omega_k}_T(s) := \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}^{\omega_k}_T(s)\,, \tag{A.22}$$

$$\begin{aligned}\mathbf{j}'^{v_k}_T(s) := {} & \mathbf{j}'^{\omega_k}_T(s) \times \Delta_k(s)\\ & + \mathbf{j}^{\omega_k}_T(s) \times [\mathbf{J}_{v_T}(s) - \mathbf{J}_{v_k}(s)]\mathbf{a}(s)\,.\end{aligned} \tag{A.23}$$

It is important to mention that, because of the structure of (A.18) and (A.19), the derivative of the jacobian matrix can always be posed in the following form

$$\dot{\mathbf{J}}_T(s,\dot{s}) = \mathbf{J}'_T(s)\,\dot{s}\ ,$$

with

$$\mathbf{J}'_T(s) := \left[\begin{array}{c} \mathbf{J}'_{v_T}(s) \\ \hline \mathbf{J}'_{\omega_T}(s) \end{array}\right] = \left[\begin{array}{c|c|c|c} \mathbf{j}'^{v_1}_T(s) & \mathbf{j}'^{v_2}_T(s) & \cdots & \mathbf{j}'^{v_N}_T(s) \\ \hline \mathbf{j}'^{\omega_1}_T(s) & \mathbf{j}'^{\omega_2}_T(s) & \cdots & \mathbf{j}'^{\omega_N}_T(s) \end{array}\right] .$$

The same property evidently applies for the Jacobians of the intermediate frames, so that it is licit to assume for any generic frame i that

$$\dot{\mathbf{J}}_{v_i}(s,\dot{s}) = \mathbf{J}'_{v_i}(s)\,\dot{s} \,, \tag{A.24}$$

$$\dot{\mathbf{J}}_{\omega_i}(s,\dot{s}) = \mathbf{J}'_{\omega_i}(s)\,\dot{s} \,. \tag{A.25}$$

The trajectory scaling method proposed in Section 2 is based on the knowledge of $\ddot{\mathbf{J}}_T(s,\dot{s},\ddot{s})$, which terms can by obtained from (A.7) and (A.8). According to (A.8), for prismatic joints $\ddot{\mathbf{j}}_T^{\omega_k} = \mathbf{0}$, while for revolute joints $\ddot{\mathbf{j}}_T^{\omega_k} = \ddot{\hat{\mathbf{z}}}_k$. Let us focus our attention on the evaluation of $\ddot{\hat{\mathbf{z}}}_k$. In [35] it has been shown that

$$\dot{\hat{\mathbf{z}}}_k = \boldsymbol{\omega}_k \times \hat{\mathbf{z}}_k \,.$$

Its time derivative is evidently given by

$$\ddot{\hat{\mathbf{z}}}_k = \boldsymbol{\alpha}_k \times \hat{\mathbf{z}}_k + \boldsymbol{\omega}_k \times \dot{\hat{\mathbf{z}}}_k = \boldsymbol{\alpha}_k \times \hat{\mathbf{z}}_k + \boldsymbol{\omega}_k \times (\boldsymbol{\omega}_k \times \hat{\mathbf{z}}_k) \,. \tag{A.26}$$

From (10) it is possible to evince that

$$\boldsymbol{\omega}_k = \mathbf{J}_{\omega_k}\,\dot{\mathbf{q}} \,. \tag{A.27}$$

The differentiation of (A.27) leads to the following expression

$$\boldsymbol{\alpha}_k = \dot{\mathbf{J}}_{\omega_k}\,\dot{\mathbf{q}} + \mathbf{J}_{\omega_k}\,\ddot{\mathbf{q}} \,. \tag{A.28}$$

By substituting (A.27) and (A.28) into (A.26) it is possible to write

$$\ddot{\hat{\mathbf{z}}}_k = (\dot{\mathbf{J}}_{\omega_k}\,\dot{\mathbf{q}} + \mathbf{J}_{\omega_k}\,\ddot{\mathbf{q}}) \times \hat{\mathbf{z}}_k + \mathbf{J}_{\omega_k}\,\dot{\mathbf{q}} \times (\mathbf{J}_{\omega_k}\,\dot{\mathbf{q}} \times \hat{\mathbf{z}}_k) \,. \tag{A.29}$$

Equation (A.29) needs to be posed in function of s and of its time derivatives. Closed form expressions for $\dot{\mathbf{q}}(s,\dot{s})$ and $\ddot{\mathbf{q}}(s,\dot{s},\ddot{s})$ are given by (21) and (22), while $\dot{\mathbf{J}}_{\omega_k}(s,\dot{s})$ is given by (A.25), and $\mathbf{J}_{\omega_k}(s)$ is a matrix composed by terms $\hat{\mathbf{z}}_k(s)$ that, in turn, are given by (A.11). By considering such substitutions, (A.29) can be rewritten as follows

$$\begin{aligned} \ddot{\hat{\mathbf{z}}}_k(s,\dot{s},\ddot{s}) = \Big\{ & [\mathbf{J}'_{\omega_k}(s)\,\mathbf{a}(s) + \mathbf{J}_{\omega_k}(s)\,\mathbf{b}(s)] \times \mathbf{j}_T^{\omega_k}(s) \\ & + \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times [\mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}_T^{\omega_k}(s)] \Big\}\,\dot{s}^2 \\ & + \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}_T^{\omega_k}(s)\,\ddot{s} \,, \end{aligned}$$

and, consequently, for revolute joints generic term $\ddot{\mathbf{j}}_T^{\omega_k} = \ddot{\hat{\mathbf{z}}}_k$ is given by

$$\ddot{\mathbf{j}}_T^{\omega_k}(s,\dot{s},\ddot{s}) := \mathbf{j}_T''^{\omega_k}(s)\,\dot{s}^2 + \mathbf{j}_T'^{\omega_k}(s)\,\ddot{s}\ , \tag{A.30}$$

where $\mathbf{j}_T'^{\omega_k}(s)$ is defined according to (A.18), while

$$\begin{aligned}\mathbf{j}_T''^{\omega_k}(s) := {} & [\mathbf{J}'_{\omega_k}(s)\,\mathbf{a}(s) + \mathbf{J}_{\omega_k}(s)\,\mathbf{b}(s)] \times \mathbf{j}_T^{\omega_k}(s) \\ & + \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}_T'^{\omega_k}(s)\ . \end{aligned} \tag{A.31}$$

The evaluation of terms $\ddot{\mathbf{j}}_T^{v_k}(s,\dot{s},\ddot{s})$ requires a similar procedure. For prismatic joints, because of (A.7), term $\ddot{\mathbf{j}}_T^{v_k}(s,\dot{s},\ddot{s})$ still coincides with $\ddot{\hat{\mathbf{z}}}_k$, so that it can be derived from (A.29) by applying the same substitutions. The sole difference is represented by term $\hat{\mathbf{z}}_k$ that, according to (A.10), now coincides with $\mathbf{j}_T^{v_k}$. Consequently, it is possible to assert that

$$\ddot{\mathbf{j}}_T^{v_k}(s,\dot{s},\ddot{s}) = \mathbf{j}_T''^{v_k}(s)\,\dot{s}^2 + \mathbf{j}_T'^{v_k}(s)\,\ddot{s}\ ,$$

where $\mathbf{j}_T'^{v_k}(s)$ is given by (A.21), while

$$\begin{aligned}\mathbf{j}_T''^{v_k}(s) := {} & [\mathbf{J}'_{\omega_k}(s)\,\mathbf{a}(s) + \mathbf{J}_{\omega_k}(s)\,\mathbf{b}(s)] \times \mathbf{j}_T^{v_k}(s) \\ & + \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}_T'^{v_k}(s)\ . \end{aligned} \tag{A.32}$$

Equation (A.7) is also the starting point for the evaluation of terms $\ddot{\mathbf{j}}_T^{v_k}(s,\dot{s},\ddot{s})$ for revolute joints. The linear velocity and acceleration of any frame k, including tool frame T, are given by

$$\begin{aligned}\mathbf{v}_k &= \mathbf{J}_{v_k}\,\dot{\mathbf{q}}\ , \\ \mathbf{a}_k &= \dot{\mathbf{J}}_{v_k}\,\dot{\mathbf{q}} + \mathbf{J}_{v_k}\,\ddot{\mathbf{q}}\ ,\end{aligned}$$

so that the second equation of (A.7) can be rewritten as follows

$$\begin{aligned}\ddot{\mathbf{j}}_T^{v_k} = {} & \ddot{\hat{\mathbf{z}}}_k \times [\mathbf{p}_T - \mathbf{p}_k] + 2\,\dot{\hat{\mathbf{z}}}_k \times [\mathbf{J}_{v_T} - \mathbf{J}_{v_k}]\,\dot{\mathbf{q}} \\ & + \hat{\mathbf{z}}_k \times [(\dot{\mathbf{J}}_{v_T} - \dot{\mathbf{J}}_{v_k})\,\dot{\mathbf{q}} + (\mathbf{J}_{v_T} - \mathbf{J}_{v_k})\,\ddot{\mathbf{q}}]\end{aligned}$$

The analytic expression of $\ddot{\mathbf{j}}_T^{v_k}(s,\dot{s},\ddot{s})$ is obtained by applying the following substitutions: $\ddot{\hat{\mathbf{z}}}_k(s,\dot{s},\ddot{s})$ is given by (A.30), $\mathbf{p}_T(s) - \mathbf{p}_k(s)$ coincides with (A.13), $\dot{\hat{\mathbf{z}}}_k(s,\dot{s})$ is given by (A.14), $\dot{\mathbf{q}}(s,\dot{s})$ is given by (21), $\ddot{\mathbf{q}}(s,\dot{s},\ddot{s})$ is given by (22), and, finally, $\dot{\mathbf{J}}_{v_k}$ and $\dot{\mathbf{J}}_{v_T}$ can be evaluated by means of (A.24). As a consequence, $\ddot{\mathbf{j}}_T^{v_k}(s,\dot{s},\ddot{s})$ assumes the following expression

$$\ddot{\mathbf{j}}_T^{v_k}(s,\dot{s},\ddot{s}) = \mathbf{j}_T''^{v_k}(s)\,\dot{s}^2 + \mathbf{j}_T'^{v_k}(s)\ddot{s}\ . \tag{A.33}$$

Table A.4: Equations that are required for the evaluation of the second time derivative of the jacobian matrix.

	$\mathbf{j}''^{v_k}_T(s)$	$\mathbf{j}'^{v_k}_T(s)$	$\mathbf{j}''^{\omega_k}_T(s)$	$\mathbf{j}'^{\omega_k}_T(s)$
prismatic	(A.32)	(A.21)	$\mathbf{0}$	$\mathbf{0}$
revolute	(A.34)	(A.23)	(A.31)	(A.22)

where $\mathbf{j}'^{v_k}_T(s)$ is given by (A.23), while

$$\begin{aligned}\mathbf{j}''^{v_k}_T(s) = {} & \mathbf{j}''^{\omega_k}_T(s) \times \Delta_k(s) \\ & + 2\,\mathbf{j}'^{\omega_k}_T(s) \times \left[\mathbf{J}_{v_T}(s) - \mathbf{J}_{v_k}(s)\right] \mathbf{a}(s) \\ & + \mathbf{j}^{\omega_k}_T(s) \times \Big\{\left[\mathbf{J}'_{v_T}(s) - \mathbf{J}'_{v_k}(s)\right] \mathbf{a}(s) \\ & + \left[\mathbf{J}_{v_T}(s) - \mathbf{J}_{v_k}(s)\right] \mathbf{b}(s)\Big\}\ . \end{aligned} \tag{A.34}$$

It is thus clear that, independently from the type of joint, terms $\ddot{\mathbf{j}}^{v_k}_T(s, \dot{s}, \ddot{s})$ and $\ddot{\mathbf{j}}^{\omega_k}_T(s, \dot{s}, \ddot{s})$ always admit the following structures

$$\ddot{\mathbf{j}}^{v_k}_T(s, \dot{s}, \ddot{s}) = \mathbf{j}''^{v_k}_T(s)\,\dot{s}^2 + \mathbf{j}'^{v_k}_T(s)\,\ddot{s}\ , \tag{A.35}$$

$$\ddot{\mathbf{j}}^{\omega_k}_T(s, \dot{s}, \ddot{s}) = \mathbf{j}''^{\omega_k}_T(s)\,\dot{s}^2 + \mathbf{j}'^{\omega_k}_T(s)\,\ddot{s}\ , \tag{A.36}$$

whose terms are defined according to Table A.4. Consequently, the second time derivative of the Jacobian matrix can always be written as in (20).

It is important to highlight that many terms of $\ddot{\mathbf{J}}_T(s, \dot{s}, \ddot{s})$ coincide with analogous terms of $\dot{\mathbf{J}}_T(s, \dot{s})$, so that they need to be computed only once. For example, many terms of (A.33) also appear in (A.23), so that the additional computational burden that is required for the evaluation of $\ddot{\mathbf{j}}^{v_k}_T(s, \dot{s}, \ddot{s})$ is quite limited. This is a relevant property, since $\ddot{\mathbf{J}}_T(s, \dot{s}, \ddot{s})$ is evaluated at each sample time: its efficient evaluation has represented a key-point for the synthesis of the TSS.

References

[1] O. Wigstrom, B. Lennartson, A. Vergnano, and C. Breitholtz, "High-Level Scheduling of Energy Optimal Trajectories," *IEEE Trans. on Autom. Sci. and Eng.*, vol. 10, no. 1, pp. 57–64, 2013.

[2] Y. Wang, Y. Zhao, S. Bortoff, and K. Ueda, "A Real-Time Energy-Optimal Trajectory Generation Method for a Servomotor System," *IEEE Trans. on Ind. Electr.*, 2014, in press.

[3] A. Piazzi and A. Visioli, "Global minimum-jerk trajectory planning of robot manipulators," *IEEE Trans. on Ind. Electr.*, vol. 47, no. 1, pp. 140–149, feb 2000.

[4] P. Boscariol and A. Gasparetto, "Model-based trajectory planning for flexible-link mechanisms with bounded jerk," *Rob. and Comp.-Integ. Manuf.*, vol. 29, no. 4, pp. 90–99, 2013.

[5] I. Pietsch, M. Krefft, O. Becker, C. Bier, and J. Hesselbach, "How to reach the dynamic limits of parallel robots? An autonomous control approach," *IEEE Trans. on Autom. Science and Eng.*, vol. 2, no. 4, pp. 369–380, Oct. 2005.

[6] X. Zhang, Y. Fang, and N. Sun, "Minimum-Time Trajectory Planning for Underactuated Overhead Crane Systems With State and Control Constraints," *IEEE Trans. on Ind. Electr.*, vol. 61, no. 12, pp. 6915–6925, Dec 2014.

[7] A. Gasparetto and V. Zanotto, "A technique of time-jerk optimal planning of robot trajectories," *Rob. and Comp.-Integ. Manuf.*, vol. 24, pp. 415–426, 2008.

[8] A. Gasparetto, A. Lanzutti, R. Vidoni, and V. Zanotto, "Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning," *Rob. and Comp.-Int. Manuf.*, vol. 28, no. 2, pp. 164–181, 2012.

[9] H. Liu, X. Lai, and W. Wu, "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints," *Rob. and Comp.-Int. Manuf.*, vol. 29, no. 2, pp. 309–317, 2013.

[10] X. Broquère, D. Sidobre, and I. Herrera-Aguilar, "Soft motion trajectory planner for service manipulator robot," in *IEEE/RSJ Int. Conf. on Int. Rob. and Sys., IROS 08*, 2008, pp. 2808–2813.

[11] R. Haschke, E. Weitnauer, and H. Ritter, "On-line planning of timeoptimal, jerk-limited trajectories," in *IEEE/RSJ Int. Conf. on Int. Rob. and Sys., IROS 08*, 2008, pp. 3248–3253.

[12] T. Kröger and F. M. Wahl, "On-line trajectory generation: basic concepts for instantaneous reactions to unforeseen events," *IEEE Trans. on Rob.*, vol. 26, no. 1, pp. 94–111, Feb. 2010.

[13] B. Ezair, T. Tassa, and Z. Shiller, "Planning high order trajectories with general initial and final conditions and asymmetric bounds," *Int. J. of Rob. Res.*, vol. 33, no. 6, pp. 898–916, May 2014.

[14] K. Kant and S. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 72–89, 1986.

[15] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," *J Dyn Sys Meas Control*, vol. 106, no. 1, pp. 102–106, 1984.

[16] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, Jun. 1985.

[17] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotics manipulators along specified paths," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 3–17, 1985.

[18] J.-J. Slotine and H. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Trans. on Rob. and Autom.*, vol. 5, no. 1, pp. 118–124, Feb 1989.

[19] M. Tarkiainen and Z. Shiller, "Time optimal motions of manipulators with actuator dynamics," in *IEEE Int. Conf. on Rob. and Autom., ICRA93*, May 1993, pp. 725–730 vol.2.

[20] D. Constantinescu and E. A. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *J. Robot. Syst.*, vol. 17, no. 5, pp. 233–249, 2000.

[21] F. Debrouwere, W. Van Loock, G. Pipeleers, Q. Dinh, M. Diehl, J. De Schutter, and J. Swevers, "Time-Optimal Path Following for Robots With Convex-Concave Constraints Using Sequential Convex Programming," *IEEE Trans. on Rob.*, vol. 29, no. 6, pp. 1485–1495, Dec 2013.

[22] Q.-C. Pham, “A General, Fast, and Robust Implementation of the Time-Optimal Path Parameterization Algorithm,” *IEEE Trans. on Rob.*, vol. 30, no. 6, pp. 1533–1540, Dec 2014.

[23] O. Dahl and L. Nielsen, “Torque-limited path following by online trajectory time scaling,” *IEEE Trans Robot Automat*, vol. 6, no. 5, pp. 554–561, 1990.

[24] O. Dahl, “Path-constrained robot control with limited torques-experimental evaluation,” *IEEE Trans Robot Automat*, vol. 10, no. 5, pp. 658–669, 1994.

[25] J. Moreno-Valenzuela and E. Oronzco-Manríquez, “A new approach to motion control of torque-constrained manipulators by using time-scaling of reference trajectories,” *J Mech Sci Technol*, vol. 23, no. 12, pp. 3221–3231, Dec. 2009.

[26] O. Gerelli and C. Guarino Lo Bianco, “Nonlinear variable structure filter for the online trajectory scaling,” *IEEE Trans. on Ind. Electr.*, vol. 56, no. 10, pp. 3921–3930, Oct. 2009.

[27] S. Macfarlane and E. A. Croft, “Jerk-bounded manipulator trajectory planning: design for real-time applications,” *IEEE Trans. on Rob. and Autom.*, vol. 19, no. 1, pp. 42–52, 2003.

[28] C. Guarino Lo Bianco and O. Gerelli, “Online trajectory scaling for manipulators subject to high-order kinematic and dynamic constraints,” *IEEE Trans. on Rob.*, vol. 27, no. 6, pp. 1144–1152, Dec. 2011.

[29] C. Guarino Lo Bianco and F. Ghilardelli, “Techniques to preserve the stability of a trajectory scaling algorithm,” in *IEEE Int. Conf. on Rob. and Autom. (ICRA2013)*, 2013, pp. 870–876.

[30] F. Lange and M. Suppa, “Predictive path-accurate scaling of a sensor-based defined trajectory,” in *IEEE Int. Conf. on Rob. and Autom, ICRA'14*, May 2014, pp. 754–759.

[31] ——, “Trajectory generation for immediate path-accurate jerk-limited stopping of industrial robots,” in *IEEE Int. Conf. on Rob. and Autom., ICRA 2015*, May 2015, pp. 2021–2026.

[32] G. Schreiber, M. Otter, and G. Hirzinger, "Solving the singularity problem of non-redundant manipulators by constraint optimization," in *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys., IROS'99*, vol. 3, 1999, pp. 1482–1488.

[33] R. Zhao, D. Sidobre, and W. He, "Online via-points trajectory generation for reactive manipulations," in *IEEE/ASME Int. Conf. on Adv. Intell. Mech., AIM'14*, July 2014, pp. 1243–1248.

[34] C. Guarino Lo Bianco and F. Ghilardelli, "A Discrete-Time Filter for the Generation of Signals With Asymmetric and Variable Bounds on Velocity, Acceleration, and Jerk," *IEEE Trans. on Ind. Electr.*, vol. 61, no. 8, pp. 4115–4125, Aug 2014.

[35] ——, "Real-Time Planner in the Operational Space for the Automatic Handling of Kinematic Constraints," *IEEE Trans. on Autom. Sci. and Eng.*, vol. 11, no. 3, pp. 730–739, 2014.