Real-time planner in the operational space for the automatic handling of kinematic constraints

Corrado Guarino Lo Bianco† and Fabio Ghilardelli

***Abstract*—Planning problems in the operational space are characterized by implementation issues that do not occur in the joint space. For example, depending on the manipulator pose, relatively slow trajectories in the operational space could require unfeasible joint speeds, thus causing the degeneration of the system performances: Path tracking errors certainly increase but, in the worst situations, the manipulator must be stopped in order to prevent the system instability. This paper proposes a real-time planner in the operational space that is able to generate trajectories subject to dynamic constraints and devised according to the path-velocity decomposition approach. The feasibility is achieved by means of an automatic scaling system that, starting from a possibly unfeasible trajectory, modifies its longitudinal velocity in order to fulfill a given set of kinematic constraints, thus preserving an accurate path tracking. The scaling system promptly reacts to critical configurations through minimum-time transients. The proposed approach has been tested on an actual anthropomorphic manipulator by executing 6D trajectories.**

***Note to practitioners*—The accurate path tracking must be guaranteed especially when trajectories are planned in the operational space. Unfortunately, path tracking worsens every time system limits are exceeded. The trajectory generator proposed in this paper is specifically designed for non redundant manipulators and it is equipped with a scaling system that automatically modifies the speed of the end effector in order to guarantee an accurate path tracking. Several kinematic constraints are handled at the same time. Joint velocities are kept below the manufacturer's limits, while joint accelerations are bounded in order to achieve smooth movements. The system is also able to constrain the kinematics of the end effector. For example, in order to reduce the mechanical stress on the payload and to avoid the excitation of elastic modes, additional bounds on the velocities and accelerations of the end effector are considered and managed. The planner can also be used to generate minimum-time constrained trajectories in real-time. To this purpose, further constraints on the longitudinal velocities and accelerations have been introduced. Differently from alternative approaches, the proposed planning scheme does not require any interaction with the controller. This is an advantage, since controllers of industrial manipulators are typically not accessible or modifiable, while, in turn, proprietary trajectory planners can normally be replaced with ad-hoc implementations. The scaling system can be easily expanded in order to handle additional constraints. The trajectory smoothness, for example, can be improved by managing the jerk bounds, so that the ongoing research activity is currently focused on that target. In the same way, it could also be possible to handle some dynamic constraints, but this would impose the introduction of mutual interactions between the scaling system and the central control unit.**

† Corresponding author.

The authors are with the Dip. di Ing. dell'Informazione of the University of Parma, Italy. Email: {guarino,fghilardelli}@ce.unipr.it.

This work was supported in part by the Ateneo Italo Tedesco in the framework of a Vigoni project and by MIUR in the framework of a PRIN2010-11 project (MARIS - Marine Autonomous Interventions).

I. Introduction

Any industrial plant is affected by physical limits that could negatively interfere with the control loops. As a consequence, if high performances are desired, the existing constraints must be handled at a control level [1] or by synthesizing proper reference signals [2], [3]. In a robotic context, physical limits may be the reason of serious feasibility issues, especially when trajectories are planned in the operational space, since joint velocities can easily reach unfeasible amplitudes even in case of slow Cartesian motions. The problem is well known and different solutions have been proposed in the literature.

A historical control approach was proposed in [4], where, for a redundant manipulator, the available degrees of freedom were used to minimize joint velocities. Nevertheless, in [5] it was proved that such approach can not guarantee the avoidance of singular points and, consequently, the insurgence of kinematic problems, so that further strategies were successively proposed in order to increase the robustness of control algorithms (see, for instance, [6], [7]).

Kinematic bounds can be alternatively managed by acting on the trajectories. This approach is particularly useful when the manipulator controller can not be directly accessed. In case of redundant manipulators, the solution is typically found by using the available degrees of freedom that derive from the exploitation of the null space of the Jacobian matrix. For non-redundant manipulators, known approaches can be roughly divided in two categories. In the first, minor modifications of the planned path are allowed in order to avoid critical configurations, while, in the second, the trajectory is slowed down in order to preserve the accurate path tracking. Evidently, the choice of the most appropriate approach depends on the manipulator task. If the execution time is mandatory, the first strategy is preferred. For example, in [8], [9] minor orientation errors of the end effector are accepted in order to minimize the execution time of a constrained trajectory. Conversely, if deviations from the nominal path are not admissible, kinematic constraints must be handled with the second approach. The adopted techniques are typically based on the path-velocity decomposition paradigm [10], i.e., the trajectory is obtained by combining a path with a Longitudinal Time Law (LTL), and the feasibility is preserved by acting on this latter.

The robotics literature proposes many scaling approaches for the synthesis of optimal constrained trajectories. Several

performance indexes could be potentially considered. In [11], for example, the LTL is chosen by minimizing the energy consumption of a robotic system. More commonly, the scaling approach is used for the generation of time optimal trajectories [12]–[20]. Sometimes, the LTL is devised offline [21], while, in alternative approaches, it is generated online by means of Trajectory Scaling Systems (TSS). The robotics literature mainly focuses on planning problems in the configuration space, while a limited number of works concerns trajectories in the operational space. The offline approach proposed in [22] can be considered a precursor of the trajectory scaling methods in the Cartesian space: The proposed TSS was able to manage the motor torque limits by modulating the longitudinal speed of the end effector.

In many industrial applications, offline solutions do not represent an appropriate choice, since the path needs to be frequently changed. This aspect stimulated researches to study alternative algorithms, specifically designed to be executed online. Paper [23] proposes an interesting example of real-time strategy, where the scaling system is directly built inside the controller. More frequently, control and scaling systems are kept disjoint. That is what happens, e.g., in the "path governor" that was proposed in [24], where an optimal strategy was cyclically evaluated in order to guarantee the path tracking with a prescribed accuracy and bounded motor feeding voltages. The online solution was obtained by solving a complex optimization problem, so that the LTL was only updated 2-3 times per second.

In more recent approaches, the LTL is updated at frequencies that are close or even equal to those of the controller in order to respond more promptly to possible environmental changes. In case of problems in the operational space, it is still possible to distinguish solutions that are based on the path modification from those that act on the LTL. In the former, kinematic bounds are not explicitly defined, but joint velocities are generically kept limited by maintaining a proper distance from singular points. This result is achieved, in real time, in [25], on the basis of a measure of the system manipulability. Similarly, in [26] singular points are avoided by adding a correcting vector to the task velocity. Conversely, systems acting on the LTL are designed to explicitly fulfill a set of assigned kinematic bounds. For example, the real-time planning strategy proposed in [27] keeps joint velocities and accelerations below some given limits by means of a hybrid approach that is based on concepts which derive from both the path modification and the LTL scaling strategies. Primarily, a lightweight optimization, that is executed in real time, tries to slow down the LTL in order to follow the assigned path with the prescribed accuracy. If no feasible solution is found, then minor path violations are accepted in order to maintain joint speeds and accelerations within the assigned bounds.

The alternative strategy introduced in [28] scales the LTL in order to bound joint velocities. The scaling system returns time-optimal solutions, so that the delays that are introduced have marginal effects on the system productivity: At any time, at least one joint works at the maximum speed. This result is obtained by means of a sliding mode feedback scheme that modifies the nominal trajectory in real time.

The novel planner proposed in this paper is devised for non-redundant manipulators. The basic scheme, originally conceived in [19] and in [20] for planning problems in the configuration space, is here modified in order to handle trajectories in the operational space. To this purpose, the scaling mechanism has been redesigned. Its complexity is increased, since the relationships between the operational and the configuration space must now be taken into account. However, due to the proposed algorithmic formulation, computational burdens are still compatible with real-time implementations.

The new generator, if compared with alternative planners in the operational space, handles a larger number of constraints. In particular, the novel TSS simultaneously manages bounds on

- joint velocities and accelerations,
- linear velocities and accelerations of the tool frame,
- angular velocities and accelerations of the tool frame,
- longitudinal velocities and accelerations, i.e., velocities and accelerations along the path.

Each constraint is motivated by a different reason. Joint velocity bounds are due to the motors' speed limits, while joint accelerations are bounded in order to achieve smooth movements. The constraints on the tool frame are used to reduce the solicitations acting on the payload and to avoid the excitation of elastic modes. The limits on the longitudinal velocities and accelerations make it possible to convert the proposed trajectory generator, that is normally used to produce fixed-time trajectories, into a minimum-time planner.

The proposed TSS modifies the LTL by means of feasible minimum-time transients in order to promptly react to critical situations. This important characteristic has been tested with an actual 6D industrial manipulator, by generating and tracking trajectories that pass close to singular configurations. Indeed, in the neighborhood of singular points joint velocities increase very quickly, so that feasibility can be maintained only if the TSS is highly reactive.

The paper is organized as follows. The planning problem is formulated in Section II. Section III shows how the assigned constraints can be converted into bounds for the LTL. The experimental results acquired with a 6 degrees of freedom industrial manipulator are commented in Section IV. Conclusions are drawn in Section V.

II. Problem formulation

Given a trajectory in the operational space and a set of kinematic constraints, the TSS proposed in this paper appropriately modifies the longitudinal speed in order to preserve the path tracking. To this aim, trajectories are planned according to the path-velocity decomposition paradigm [10], an approach that is suited for planning scenarios in the operational space. Practically, trajectories are obtained as a combination of a path in the operational space and a LTL. By defining s as the Euclidean distance from the beginning of the path, measured along the path itself, the LTL is specified by assigning the following function

$$\begin{array}{rcl} s : [0, t_f] & \rightarrow & \mathbb{R}^+ \\ t & \rightarrow & s := s(t) \end{array} , \tag{1}$$

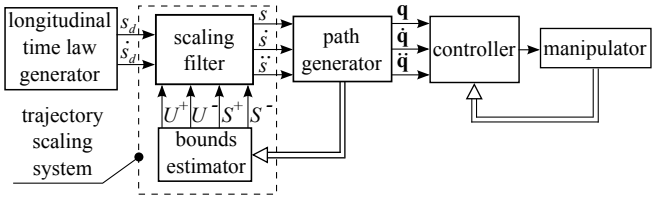

Fig. 1. The overall manipulator control scheme. The dashed box surrounds the trajectory scaling system.

where t_f is the total traveling time. Along the paper, term "path" will indicate the combination of the positions and the orientations that are assumed by the tool frame or by the wrist frame during the motion. Positions are specified through a function $\mathbf{p}_T(s)$ that is defined as follows

$$\begin{array}{rcl} \mathbf{p}_T : [0, s_f] & \rightarrow & \mathbb{R}^3 \\ s & \rightarrow & \mathbf{p}_T := \mathbf{p}_T(s) \end{array} , \tag{2}$$

where $s_f = s(t_f)$ is the path length. Orientations can be specified in two different ways, depending on the planning strategy. They can be expressed, e.g., by the following function

$$\begin{array}{rcl} \mathbf{\Phi}_T : [0, s_f] & \rightarrow & \mathbb{R}^3 \\ s & \rightarrow & \mathbf{\Phi}_T := \mathbf{\Phi}_T(s) \end{array} , \tag{3}$$

where $\mathbf{\Phi}_T$ indicates an orientation that is expressed through any minimal representation. Alternatively, orientations could also be described by means of a rotation matrix

$$\begin{array}{rcl} {}^0_T\mathbf{R} : [0, s_f] & \rightarrow & \mathbb{R}^9 \\ s & \rightarrow & {}^0_T\mathbf{R} := {}^0_T\mathbf{R}(s) \end{array} . \tag{4}$$

Depending on the selected planning scheme, the trajectory is thus given by functions $\mathbf{p}_T(t) = \mathbf{p}_T[s(t)]$ and $\mathbf{\Phi}_T(t) = \mathbf{\Phi}_T[s(t)]$ or, alternatively, by $\mathbf{p}_T(t) = \mathbf{p}_T[s(t)]$ and ${}^0_T\mathbf{R}(t) = {}^0_T\mathbf{R}[s(t)]$. Position and orientations are measured with respect to an inertial frame. Without any loss of generality, it will coincide with the manipulator 0 frame.

From the knowledge of the path in the operational space, it is possible to obtain an analogous path $\mathbf{q}_T(t)$ in the joint space by solving an inverse kinematic problem. $\mathbf{q}_T(t)$ is the reference signal for the manipulator controller.

The TSS shown in Fig. 1 is a dynamic system that is inserted between the LTL generator and the path generator: It prevents constraint violations by properly reducing longitudinal speeds and accelerations. An advantage of this scaling technique is immediately evident: The TSS does not require data from the manipulator controller, which internal structure can even be unknown.

The scheme of the TSS is similar to those that have been proposed in [19], [20] for problems in the configuration space. It is made of a general purpose, nonlinear scaling filter and a bounds estimator. The first modifies the nominal LTL, that is expressed through function $s_d(t)$, in order to keep its first and second time derivatives within given bounds, i.e., the filter output is a signal $s(t)$ that fulfills the following inequalities

$$S^- \leq \dot{s}(t) \leq S^+ , \tag{5}$$

$$U^- \leq \ddot{s}(t) \leq U^+ , \tag{6}$$

where S^+, S^-, U^+, and U^- are freely assignable limits. The scaling filter has been widely described in [19] and in [29], so that the interested reader can refer to those papers for details. It is worth to mention its main characteristics: Output signal $s(t)$ tracks $s_d(t)$ if this latter is feasible, otherwise $s(t)$ becomes the best possible feasible approximation of $s_d(t)$. Moreover, $s(t)$ always hangs in minimum time, compatibly with the imposed constraints, any feasible $s_d(t)$. This property can be used, as shown in Section IV, to generate minimum-time trajectories.

The second block of the TSS is the bounds estimator, that converts the kinematic constraints, which affect the manipulator, into equivalent limits for the LTL. It is totally different from analogous implementations that have been proposed in [19], [20], [29], since it must handle problems that are defined in the operational space. Its computational complexity is necessarily higher, because the required conversions involve the existing relationships between the configuration and the operational spaces. Nevertheless, the proposed formulation, as shown in Section IV, is perfectly compatible with a real-time implementation. The bounds estimator handles several different constraints. For example, velocities and accelerations of the end effector must be kept within some given limits. Thus, by defining the desired generalized velocity of the tool frame as

$$\overline{\mathbf{v}}_T := \left[\frac{\mathbf{v}_T}{\boldsymbol{\omega}_T}\right] = [\overline{v}_{T_1}\ \overline{v}_{T_2}\ \overline{v}_{T_3}\ \overline{v}_{T_4}\ \overline{v}_{T_5}\ \overline{v}_{T_6}]^T \in \mathbb{R}^6 \tag{7}$$

where $\mathbf{v}_T$ and $\boldsymbol{\omega}_T$ respectively represent the desired linear and angular velocities in the operational space, and by indicating with

$$\overline{\mathbf{a}}_T := \left[\frac{\mathbf{a}_T}{\boldsymbol{\alpha}_T}\right] = [\overline{a}_{T_1}\ \overline{a}_{T_2}\ \overline{a}_{T_3}\ \overline{a}_{T_4}\ \overline{a}_{T_5}\ \overline{a}_{T_6}]^T \in \mathbb{R}^6 \tag{8}$$

the desired generalized acceleration, where $\mathbf{a}_T$ and $\boldsymbol{\alpha}_T$ respectively indicate the desired linear and angular accelerations, the following constraints must be simultaneously satisfied ($i = 1, 2, \ldots, 6$)

$$\overline{v}_i^- \leq \overline{v}_{T_i} \leq \overline{v}_i^+ , \tag{9}$$

$$\overline{a}_i^- \leq \overline{a}_{T_i} \leq \overline{a}_i^+ . \tag{10}$$

Constraint vectors $\overline{\mathbf{v}}^- := [\overline{v}_1^-\ \overline{v}_2^-\ \overline{v}_3^-\ \overline{v}_4^-\ \overline{v}_5^-\ \overline{v}_6^-]^T$, $\overline{\mathbf{v}}^+ := [\overline{v}_1^+\ \overline{v}_2^+\ \overline{v}_3^+\ \overline{v}_4^+\ \overline{v}_5^+\ \overline{v}_6^+]^T$, $\overline{\mathbf{a}}^- := [\overline{a}_1^-\ \overline{a}_2^-\ \overline{a}_3^-\ \overline{a}_4^-\ \overline{a}_5^-\ \overline{a}_6^-]^T$, and $\overline{\mathbf{a}}^+ := [\overline{a}_1^+\ \overline{a}_2^+\ \overline{a}_3^+\ \overline{a}_4^+\ \overline{a}_5^+\ \overline{a}_6^+]^T$ are assigned by the users depending on the desired motion smoothness.

Bounds on joint velocities, i.e., $\dot{q}_k$, and accelerations, i.e., $\ddot{q}_k$, must also be taken into account. It has been previously pointed out that trajectories in the operational space could produce joint velocities and accelerations that are beyond the actuators capabilities. In order to prevent path tracking problems the following limits must be fulfilled

$$\dot{q}_k^- \leq \dot{q}_k \leq \dot{q}_k^+ , \tag{11}$$

$$\ddot{q}_k^- \leq \ddot{q}_k \leq \ddot{q}_k^+ , \tag{12}$$

where $k = 1, 2, \ldots, N$, while N is equal to the number of independent joints. Bounds $\dot{\mathbf{q}}^- := [\dot{q}_1^-\ \dot{q}_2^-\ \ldots\ \dot{q}_N^-]^T$ and $\dot{\mathbf{q}}^+ := [\dot{q}_1^+\ \dot{q}_2^+\ \ldots\ \dot{q}_N^+]^T$ typically coincide with the motor speed limits, while $\ddot{\mathbf{q}}^- := [\ddot{q}_1^-\ \ddot{q}_2^-\ \ldots\ \ddot{q}_N^-]^T$ and $\ddot{\mathbf{q}}^+ := [\ddot{q}_1^+\ \ddot{q}_2^+\ \ldots\ \ddot{q}_N^+]^T$ are used to reduce the mechanical stress:

Limited accelerations make it possible to drastically reduce the excitation of oscillatory modes.

Longitudinal velocities and accelerations along the path can be bounded, by imposing

$$\xi^- \leq \dot{s}(t) \leq \xi^+ \,, \tag{13}$$

$$\zeta^- \leq \ddot{s}(t) \leq \zeta^+ \,, \tag{14}$$

where ξ^-, ξ^+, ζ^-, and ζ^+ are user defined limits.

In next Section III, it will be shown how constraints (9)–(12) can be converted into equivalent bounds S^-, S^+, U^-, and U^+ for the LTL. No adaptation is required for constraints (13) and (14), that can be immediately applied to (5) and (6).

III. Evaluation of the equivalent longitudinal bounds

As known, velocities and accelerations of the end effector can be evaluated as follows

$$\overline{\mathbf{v}}_T = \mathbf{J}_T \, \dot{\mathbf{q}} \,, \tag{15}$$

$$\overline{\mathbf{a}}_T = \dot{\mathbf{J}}_T \, \dot{\mathbf{q}} + \mathbf{J}_T \, \ddot{\mathbf{q}} \,, \tag{16}$$

where $\mathbf{J}_T = \mathbf{J}_T(\mathbf{q})$ is the system geometric Jacobian, while $\dot{\mathbf{J}}_T = \dot{\mathbf{J}}_T(\mathbf{q})$ is its first time derivative.

If $\mathbf{J}_T$ is not singular, from (15) it immediately descends that

$$\dot{\mathbf{q}} = \mathbf{J}_T^{-1} \, \overline{\mathbf{v}}_T \,, \tag{17}$$

while, by manipulating (16), it is possible to write

$$\ddot{\mathbf{q}} = \mathbf{J}_T^{-1}(\overline{\mathbf{a}}_T - \dot{\mathbf{J}}_T \, \dot{\mathbf{q}}) = \mathbf{J}_T^{-1}(\overline{\mathbf{a}}_T - \dot{\mathbf{J}}_T \, \mathbf{J}_T^{-1} \, \overline{\mathbf{v}}_T) \,. \tag{18}$$

Equations (17) and (18) are expressed in function of variables of the configuration space and are instrumental for the evaluation of the equivalent longitudinal bounds, provided that they could be posed in function of s.

To this purpose, path $[\mathbf{p}_T(s), {}^0_T\mathbf{R}(s)]$ is first converted into an equivalent path $\mathbf{q}_T(s)$ in the configuration space, which is subsequently used to evaluate

$$\mathbf{J}_T(s) = \mathbf{J}[\mathbf{q}(s)]$$

and

$$\mathbf{J}_T^{-1}(s) = \mathbf{J}^{-1}[\mathbf{q}(s)] \,.$$

Linear velocity $\mathbf{v}_T$ can be obtained from $\mathbf{p}_T(s)$ by means of the chain differentiation rule, i.e.,

$$\mathbf{v}_T(s, \dot{s}) = \frac{d\mathbf{p}_T(s)}{ds} \, \frac{ds}{dt} = \frac{d\mathbf{p}_T(s)}{ds} \, \dot{s} := \mathbf{v}_T(s) \, \dot{s} \,. \tag{19}$$

A similar result can also be obtained for $\boldsymbol{\omega}_T := [\omega_x \;\; \omega_y \;\; \omega_z]^T$. The differentiation rule of rotational matrices, i.e., ${}^0_T\dot{\mathbf{R}} = \mathbf{S}(\boldsymbol{\omega}_T) \, {}^0_T\mathbf{R}$, where $\mathbf{S}(\boldsymbol{\omega}_T)$ is a skew symmetric matrix that is defined as follows

$$\mathbf{S}(\boldsymbol{\omega}_T) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \,,$$

makes it possible to write

$$\mathbf{S}[\boldsymbol{\omega}_T(s, \dot{s})] = {}^0_T\dot{\mathbf{R}} \, {}^0_T\mathbf{R}^T = \frac{d\left[{}^0_T\mathbf{R}(s)\right]}{ds} \, {}^0_T\mathbf{R}^T(s) \, \dot{s} \,. \tag{20}$$

Owing to (20), $\boldsymbol{\omega}_T(s, \dot{s})$ can evidently be written as follows

$$\boldsymbol{\omega}_T(s, \dot{s}) = \boldsymbol{\omega}_T(s) \, \dot{s} \,. \tag{21}$$

Bearing in mind (19) and (21), (7) can be posed into the following form

$$\overline{\mathbf{v}}_T(s, \dot{s}) = \widetilde{\mathbf{v}}_T(s) \, \dot{s} \,, \tag{22}$$

so that its first derivative with respect to time is given by

$$\overline{\mathbf{a}}_T(s, \dot{s}) = \widetilde{\mathbf{v}}'_T(s) \, \dot{s}^2 + \widetilde{\mathbf{v}}_T(s) \, \ddot{s} \tag{23}$$

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

By means of (22), (17) can be written as follows

$$\dot{\mathbf{q}}(s, \dot{s}) = \mathbf{a}(s) \, \dot{s} \,, \tag{24}$$

where

$$\mathbf{a}(s) := \mathbf{J}_T^{-1}(s) \, \widetilde{\mathbf{v}}_T(s) \,. \tag{25}$$

As shown in [30], $\widetilde{\mathbf{v}}_T(s)$ can be obtained from the knowledge of the path equations, so that, given s, it is certainly possible to evaluate $\mathbf{a}(s)$. Consequently, in the following, $\mathbf{a}(s)$ is supposed to be known.

A procedure for the evaluation of $\dot{\mathbf{J}}_T$ is proposed in Appendix A, where it is also shown that $\dot{\mathbf{J}}_T$ assumes the following structure

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

Thus, bearing also in mind (22) and (23), (18) can be written as follows

$$\ddot{\mathbf{q}}(s, \dot{s}, \ddot{s}) = \mathbf{a}(s) \, \ddot{s} + \mathbf{b}(s) \, \dot{s}^2 \,, \tag{26}$$

where $\mathbf{a}(s)$ is given by (25), while

$$\mathbf{b}(s) := \mathbf{J}_T^{-1}(s) \, [\widetilde{\mathbf{v}}'_T(s) - \mathbf{J}'_T(s) \, \mathbf{J}_T^{-1}(s) \, \widetilde{\mathbf{v}}_T(s)] \,. \tag{27}$$

Since $\widetilde{\mathbf{v}}'_T(s)$ can be analytically obtained from the path equations, all terms that are required for the evaluation of $\mathbf{b}(s)$ are known.

Constraint equations (11) and (12) can be written, due to (24) and (26), in the following form ($k = 1, 2, \ldots, N$)

$$\dot{q}_k^- \leq a_k(s) \, \dot{s} \leq \dot{q}_k^+ \,, \tag{28}$$

$$\ddot{q}_k^- \leq b_k(s) \, \dot{s}^2 + a_k(s) \, \ddot{s} \leq \ddot{q}_k^+ \,, \tag{29}$$

where $a_k(s)$ and $b_k(s)$ respectively indicate the components of vectors $\mathbf{a}(s) := [a_1(s) \, a_2(s) \, \ldots \, a_N(s)]^T$ and $\mathbf{b}(s) := [b_1(s) \, b_2(s) \, \ldots \, b_N(s)]^T$. Inequalities (28) and (29) can, then, be converted into equivalent bounds for $\dot{s}$ and $\ddot{s}$. In particular, (28) is fulfilled if $\dot{s}$ satisfies the following expression $\dot{s} \in \bigcap_{k=1}^{N} [0 \,, \eta_k]$, where

$$\eta_k = \begin{cases} \frac{\dot{q}_k^+}{a_k(s)}, & \text{if } a_k(s) > 0 \\ \frac{\dot{q}_k^-}{a_k(s)}, & \text{if } a_k(s) < 0 \\ \infty, & \text{if } a_k(s) = 0 \end{cases} \,. \tag{30}$$

Notice that the lower bound on $\dot{s}$ has been posed equal to zero in order to avoid backward movements.

In the same way, joints accelerations fulfill (29) if $\ddot{s}$ belongs to interval $\ddot{s} \in \bigcap_{k=1}^{N} [\mu_k \, , \lambda_k]$, with

$$\lambda_k = \begin{cases} \frac{\ddot{q}_k^+ - b_k(s)\,\dot{s}^2}{a_k(s)}, & \text{if } a_k(s) > 0 \\ \frac{\ddot{q}_k^- - b_k(s)\,\dot{s}^2}{a_k(s)}, & \text{if } a_k(s) < 0 \\ \infty, & \text{if } a_k(s) = 0 \end{cases} , \tag{31}$$

and

$$\mu_k = \begin{cases} \frac{\ddot{q}_k^- - b_k(s)\,\dot{s}^2}{a_k(s)}, & \text{if } a_k(s) > 0 \\ \frac{\ddot{q}_k^+ - b_k(s)\,\dot{s}^2}{a_k(s)}, & \text{if } a_k(s) < 0 \\ -\infty, & \text{if } a_k(s) = 0 \end{cases} . \tag{32}$$

Constraints (9) and (10) can be similarly handled. Due to (22) and (23), it is possible to rewrite (9) and (10) as follows ($i = 1, 2, \ldots, 6$)

$$\overline{v}_i^- \leq \widetilde{v}_{T_i}(s)\,\dot{s} \leq \overline{v}_i^+ \, , \tag{33}$$

$$\overline{a}_i^- \leq \widetilde{v}'_{T_i}(s)\,\dot{s}^2 + \widetilde{v}_{T_i}(s)\,\ddot{s} \leq \overline{a}_i^+ \, , \tag{34}$$

where $\widetilde{v}_{T_i}(s)$ and $\widetilde{v}'_{T_i}(s)$ respectively indicate the components of vectors $\mathbf{v}_T(s) := [\widetilde{v}_{T_1}(s)\, \widetilde{v}_{T_2}(s)\, \ldots\, \widetilde{v}_{T_6}(s)]^T$ and $\mathbf{v}'_T(s) := [\widetilde{v}'_{T_1}(s)\, \widetilde{v}'_{T_2}(s)\, \ldots\, \widetilde{v}'_{T_6}(s)]^T$. Consequently, the constraints in the operational space are fulfilled if $\dot{s} \in \bigcap_{i=1}^{6} [0 \, , \rho_i]$, where

$$\rho_i = \begin{cases} \frac{\overline{v}_i^+}{\widetilde{v}_{T_i}(s)}, & \text{if } \widetilde{v}_{T_i}(s) > 0 \\ \frac{\overline{v}_i^-}{\widetilde{v}_{T_i}(s)}, & \text{if } \widetilde{v}_{T_i}(s) < 0 \\ \infty, & \text{if } \widetilde{v}_{T_i}(s) = 0 \end{cases} , \tag{35}$$

and if $\ddot{s} \in \bigcap_{i=1}^{6} [\gamma_i \, , \delta_i]$, where

$$\delta_i = \begin{cases} \frac{\overline{a}_i^+ - \widetilde{v}'_{T_i}(s)\,\dot{s}^2}{\widetilde{v}_{T_i}(s)}, & \text{if } \widetilde{v}_{T_i}(s) > 0 \\ \frac{\overline{a}_i^- - \widetilde{v}'_{T_i}(s)\,\dot{s}^2}{\widetilde{v}_{T_i}(s)}, & \text{if } \widetilde{v}_{T_i}(s) < 0 \\ \infty, & \text{if } \widetilde{v}_{T_i}(s) = 0 \end{cases} , \tag{36}$$

and

$$\gamma_i = \begin{cases} \frac{\overline{a}_i^- - \widetilde{v}'_{T_i}(s)\,\dot{s}^2}{\widetilde{v}_{T_i}(s)}, & \text{if } \widetilde{v}_{T_i}(s) > 0 \\ \frac{\overline{a}_i^+ - \widetilde{v}'_{T_i}(s)\,\dot{s}^2}{\widetilde{v}_{T_i}(s)}, & \text{if } \widetilde{v}_{T_i}(s) < 0 \\ -\infty, & \text{if } \widetilde{v}_{T_i}(s) = 0 \end{cases} . \tag{37}$$

Bearing in mind all previously mentioned conditions, it is finally possible to assert that constraints (9)–(14) are certainly fulfilled, if inequalities (5) and (6) are satisfied, with ($k = 1, 2, \ldots, N,\ i = 1, 2, \ldots, 6$)

$$S^- := 0 \, , \quad S^+ := \min_{k,i} \{\eta_k, \rho_i, \xi^+\} \, , \tag{38}$$

$$U^- := \max_{k,i} \{\mu_k, \gamma_i, \zeta^-\} \, , \quad U^+ := \min_{k,i} \{\lambda_k, \delta_i, \zeta^+\} \, . \tag{39}$$

As formerly asserted, the fulfillment of conditions (5) and (6) is delegated to the scaling filter that online modifies the LTL. It is important to point out the mutual correlations existing between the scaling filter and the bounds estimator. The first online modifies the LTL, in order to fulfill the given longitudinal bounds. The second, in turn, continuously changes the longitudinal constraints by means of (30)–(37), thus influencing the filter behavior. This practically implies that the equivalents bounds on the longitudinal velocities and accelerations are not known a priori but, conversely, they are evaluated in real time, depending on the action of the scaling filter.

Another relevant aspect that is important to highlight is that the planning problem admits solutions only if the feasible region is not empty, i.e., only if $S^- \leq S^+$ and $U^- \leq U^+$. Condition $S^- \leq S^+$ is banally satisfied because of (30) and (35), but, conversely, (31), (32), (36), and (37) do not give any certainty concerning the fulfillment of condition $U^- \leq U^+$. Fortunately, in many practical cases, the onset of problems is avoided due to the action of the filter. Let us consider, for example, a situation in which the trajectory passes close to a singular point. In that point, the interval between U^- and U^+ critically tends to zero. In the same point also S^+ suddenly decreases and, consequently, the filter scales down $\dot{s}$. This, in turn, causes an increment of the distance between U^- and U^+ through of (31), (32), (36), and (37): Condition $U^- \leq U^+$ is preserved if such expansion dominates the collapsing effect. Far from singular points, a similar result can be achieved by means of strategies like those proposed in [29], [31].

For the sake of completeness, it is worth to mention that the same scaling approach can also be adopted if the orientation of the end effector is alternatively specified by means of a minimal representation and, consequently, if the points in the operational space are expressed by vector $\overline{\mathbf{x}}_T := [\mathbf{p}_T^T \; \mathbf{\Phi}_T^T]^T$. In that case, the desired tool velocity can be described as follows

$$\dot{\overline{\mathbf{x}}}_T := \begin{bmatrix} \mathbf{v}_T \\ \hline \dot{\mathbf{\Phi}}_T \end{bmatrix} ,$$

where $\dot{\mathbf{\Phi}}_T$ is the time derivative of $\mathbf{\Phi}_T$.

Equations

$$\dot{\overline{\mathbf{x}}}_T = \overline{\mathbf{J}}_T\, \dot{\mathbf{q}} \, , \tag{40}$$

$$\ddot{\overline{\mathbf{x}}}_T = \dot{\overline{\mathbf{J}}}_T\, \dot{\mathbf{q}} + \overline{\mathbf{J}}_T\, \ddot{\mathbf{q}} \, , \tag{41}$$

where $\overline{\mathbf{J}}_T = \overline{\mathbf{J}}_T(\mathbf{q})$ is the analytic Jacobian of the system, have the same structure of (15) and (16), so that the scaling method that has been previously proposed can also be adopted in situations where the path in the operational space is specified by means of $\overline{\mathbf{x}}_T(s)$: In such alternative scenario, the analytic Jacobian takes the place of the geometric Jacobian.

IV. Experimental results

The TSS has been tested on a Comau Smart Six 6-1.4 manipulator controlled by means of a C4GOpen operating system, that can bypass, totally or partially, the commercial control unit. In particular, the here proposed tests have been executed by disabling the internal trajectory generator, that has been replaced by the novel planner, while the original control loops have been maintained. Trajectories have been generated by means of an external PC, operated with an RTAI real time system running at 2e-3 s. The effectiveness and the promptness of the TSS have been tested by executing trajectories that pass very close to singular points and that strongly solicit joints 4 and 6: Emergency stops, issued by the manipulator

controller, can only be avoided by keeping the joint speeds within ± 12 rad s^{-1}. Critical situations arise very quickly, so that feasibility is maintained only if the TSS is sufficiently reactive.

Two different path primitives have been considered: a linear segment and a circular segment. The linear path joins $\mathbf{p}_A = [0.650\ 0.830\ 1.120]^T$ to $\mathbf{p}_B = [-0.200\ 0.830\ 1.120]^T$ and it is generated by assuming $\mathbf{\Phi}_A = \mathbf{\Phi}_B = [\pi/2\ 0\ \pi/2]^T$. The circular path goes from $\mathbf{p}_A = [0.650\ 0.830\ 0.810]^T$ to $\mathbf{p}_B = [-0.144\ 0.830\ 0.879]^T$ and it is centered in $\mathbf{p}_C = [0.250\ 0.830\ 0.810]^T$. Evidently, the motion does not admit Cartesian components along the Y direction. The initial tool orientation, that is expressed through the Roll-Pitch-Yaw (RPY) notation, is $\mathbf{\Phi}_A = [\pi/2\ \pi/2\ \pi/2]^T$, while the final orientation is $\mathbf{\Phi}_B = [\pi/2\ 13\pi/9\ \pi/2]^T$. The required functions $\mathbf{p}_T(s)$, $\widetilde{\mathbf{v}}_T(s)$, and $\widetilde{\mathbf{v}}'_T(s)$ are generated according to the procedure that is proposed in [30].

In order to highlight an interesting feature of the TSS reference, signal s_T has been assumed constant and equal to the path lengths (0.850 m and 1.1868 m respectively): The TSS automatically generates, due to the capabilities of the adopted nonlinear filter [29], a minimum-time trajectory which fulfills the given longitudinal constraints on the velocity and the acceleration. The crucial point is that this result is achieved by means of a planning algorithm that is totally executed in real time.

In the first test experiments, that have been indicated in the attached video as Experiments 1 and 3, the sole bounds ξ^+, ζ^-, and ζ^+ have been activated, i.e.,

$$S^- := 0\,, \quad S^+ := \xi^+\,,$$

$$U^- := \zeta^-\,, \quad U^+ := \zeta^+\,,$$

where $\zeta^+ = 2.5$ m s^{-2}, $\zeta^- = -2.5$ m s^{-2}, $\xi^+ = 0.4239$ m s^{-1}. Limits ξ^+, ζ^+, and ζ^- have been selected such to guarantee smooth transients. Since the given trajectories pass very close to singular configurations, the velocity limits of joints 4 and 6, which have not been taken into account for the evaluation of the equivalent longitudinal bounds, are violated, so that the manipulator controller, as shown in the video, executes an emergency stop by activating the joint brakes.

In the second set of experiments, the constraints on the joint velocities and accelerations have been taken into account, i.e.,

$$S^- := 0\,, \quad S^+ := \min_k\{\eta_k, \xi^+\}\,,$$

$$U^- := \max_k\{\mu_k, \zeta^-\}\,, \quad U^+ := \min_k\{\lambda_k, \zeta^+\}\,.$$

Bounds η_k, μ_k, and λ_k are obtained from (30), (31) and (32) by assuming $\dot{\mathbf{q}}^- := [-8\ -8\ -8\ -12\ -12\ -12]^T$, $\dot{\mathbf{q}}^+ := [8\ 8\ 8\ 12\ 12\ 12]^T$, $\ddot{\mathbf{q}}^- := [-100\ -100\ -100\ -100\ -100\ -100]^T$, and $\ddot{\mathbf{q}}^+ := [100\ 100\ 100\ 100\ 100\ 100]^T$.

Figures 2b and 2c, which refer to the generation of the circular segment, compare signals $\dot{s}$ and $\ddot{s}$, that are generated by the filter, with S^+, S^-, U^+, and U^-: Equivalent longitudinal bounds are fulfilled and, as it is expected in case of optimal solutions, there is always one active constraint. Dash-dotted lines surround the zones in which the TSS modifies the nominal time law. According to the theory, the fulfillment of the equivalent longitudinal constraints guarantees that velocities and accelerations of critical joints 4 and 6, as proved in Fig. 3, are kept within the assigned bounds, so that the trajectory is correctly executed.

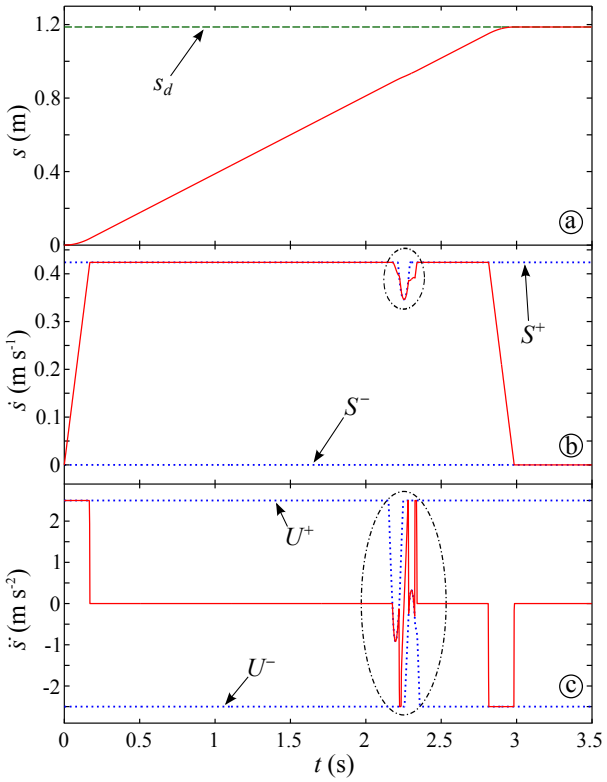

Fig. 2. Time law used for the generation of a circular paths. Dash-dotted lines highlight the action of the scaling system. a) rough reference s_T (dashed line) compared with s (solid line); b) Signal $\dot{s}$ (solid line) compared with the equivalent longitudinal bounds (dotted lines); c) Signal $\ddot{s}$ (solid line) compared with the equivalent longitudinal bounds (dotted lines).

Similar performances have been verified for the linear segment: Fig. 4 shows the trends of s, $\dot{s}$, and $\ddot{s}$, while Fig. 5 reports velocities and accelerations for joints 4 and 6.

The last experiments, that are indicated as Experiment 2 and Experiment 4 in the attached video, have been executed by activating all the constraints, i.e., S^+, S^-, U^+, and U^- have been evaluated by means of (38) and (39). Additional bounds ρ_i, γ_i, and δ_i have been obtained from (35), (36) and (37) by imposing $\overline{\mathbf{v}}^- := [-0.4\ -0.4\ -0.4\ -1\ -1\ -1]^T$, $\overline{\mathbf{v}}^+ := [0.4\ 0.4\ 0.4\ 1\ 1\ 1]^T$, $\overline{\mathbf{a}}^- := [-2\ -2\ -2\ -10\ -10\ -10]^T$, $\overline{\mathbf{a}}^+ := [2\ 2\ 2\ 10\ 10\ 10]^T$. Fig. 6 shows the results for the circular segment. Because of the additional constraints, bounds are different with respect to those shown in Fig. 2. Joint velocity and acceleration constraints are still satisfied, but now the fulfillment of the longitudinal constraints also implies that the assigned Cartesian limits are fulfilled. This is proved by Fig. 7, in which the velocity and acceleration components along the ZX-plane are shown. The Y component has been neglected because it is always equal to zero. A similar response has been obtained for the linear trajectory, but the corresponding results have been omitted for the sake of brevity.

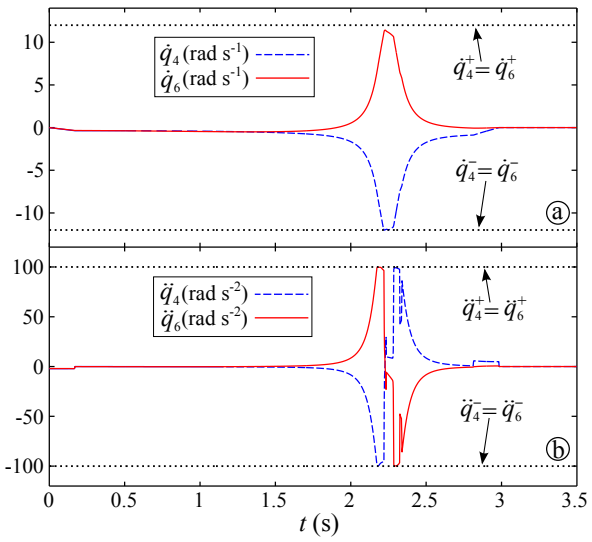

Fig. 3. a) Velocities and b) accelerations of joints 4 and 6 that are obtained during the generation of a circular path, compared with the assigned bounds (dotted lines).

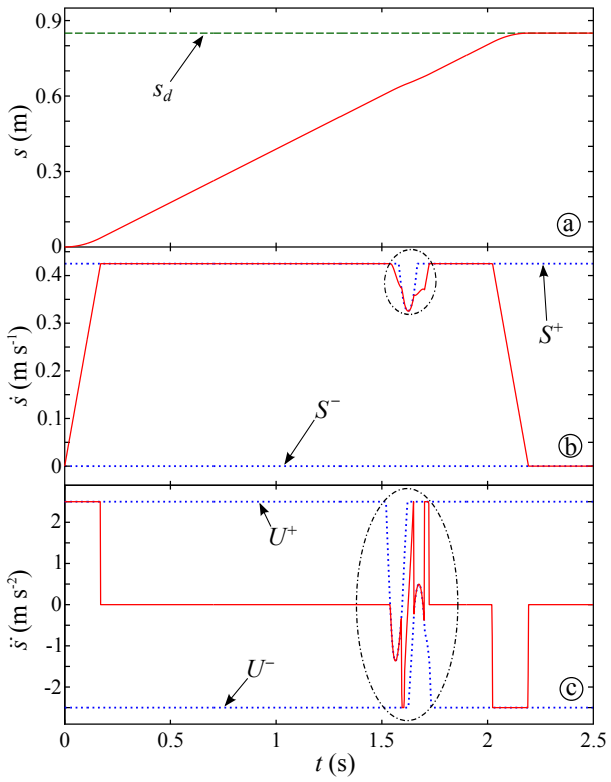

Fig. 4. Time law used for the generation of a linear path. Dash-dotted lines highlight the action of the scaling system. a) rough reference s_T (dashed line) compared with s (solid line); b) Signal $\dot{s}$ (solid line) compared with the equivalent longitudinal bounds (dotted lines); c) Signal $\ddot{s}$ (solid line) compared with the equivalent longitudinal bounds (dotted lines)

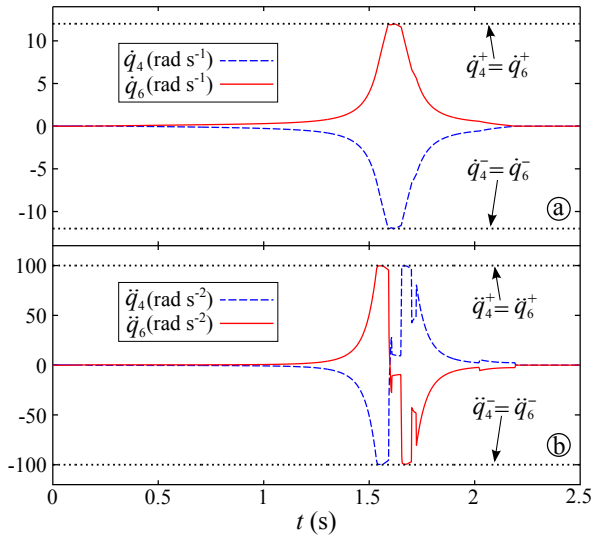

Fig. 5. a) Velocities and b) accelerations of joints 4 and 6 that are obtained during the generation of a linear path, compared with the assigned bounds (dotted lines).

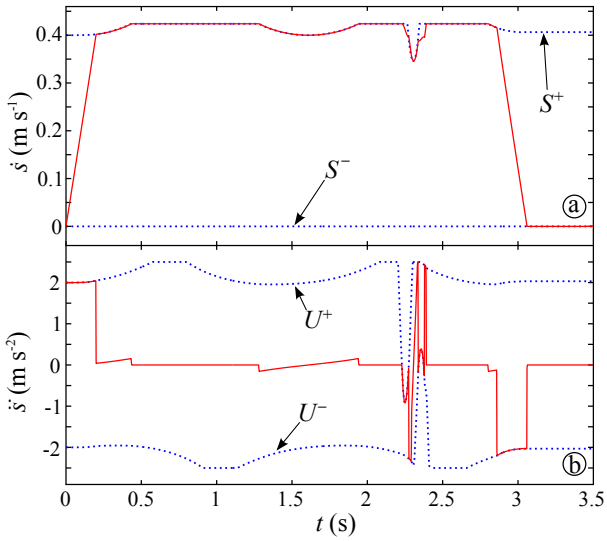

Fig. 6. Time law used for the generation of a circular path when constraints on the joint speeds and on the operational speeds are simultaneously activated. a) Signal $\dot{s}$ (solid line) compared with the equivalent longitudinal bounds (dotted lines); b) Signal $\ddot{s}$ (solid line) compared with the equivalent longitudinal bounds (dotted lines)

The performances of the TSS can also be appreciated by means of Figure 8, which reports the path tracking errors for all the experiments of the attached video. Results relative to Experiment 1 and Experiment 3 respectively end for $s = 0.8998$ m and for $s = 0.6318$ m, due to the activation of the brakes. Path tracking errors of Experiments 1 and 2 are quite similar until the trajectory is far from the singular point, but they suddenly differ in its neighborhood: the tracking error of Experiment 1 is clearly diverging when brakes are activated. Maximum errors of Experiment 2 only depend on the performances of the manipulator controller, which accuracy is inversely proportional to the joint speeds. The TSS can evidently be tuned such to guarantee higher precisions

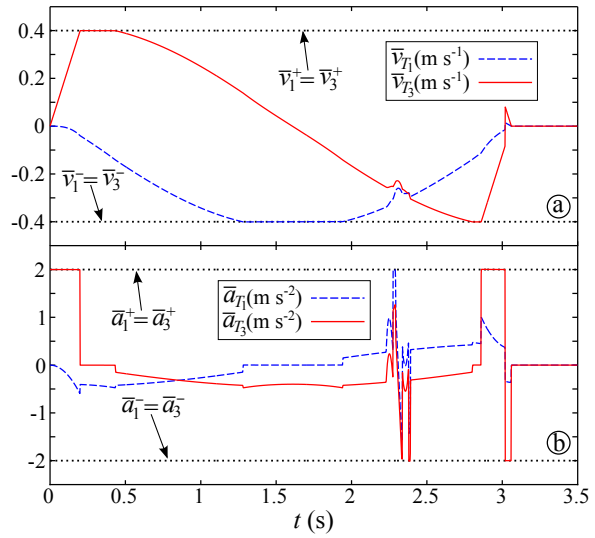

Fig. 7. Generation of a circular path: Components of a) linear velocities and b) accelerations along the X and the Z axes compared with the respective bounds (dotted lines).

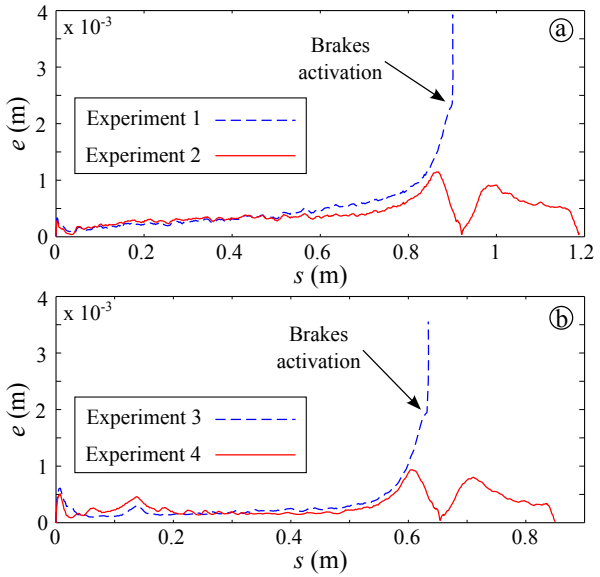

Fig. 8. Path tracking errors: a) circular path; b) linear path.

in the vicinity of the singularity by simply imposing more stringent limits on the joint speeds. Similar considerations hold for Experiments 3 and 4.

The slow motion videos highlight the tracking performances in the neighborhood of the singularity: Despite the LTL scaling, vibrations are avoided due to the smoothness of the involved transients.

Evaluation times are compatible with the real-time requirement. Indeed, the scaling filter, the nonlinear bounds estimator, and the path generator are executed in 30e-6 s on a PC Intel Core 2 Duo E8400 @3.00GHz, i.e., the computational burden is evidently marginal and compatible with the evaluation capabilities of modern controllers.

V. Conclusions

The proposed trajectory planner is able to handle some critical situations that could arise when trajectories are defined in the operational space. In particular, it uses a TSS that modifies, by means of minimum time transients, the nominal LTL in order to fulfill a given set of kinematic limits and to guarantee the accurate path tracking.

The new generator, owing to its moderate computational burden and due to the lack of interactions with the manipulator controller, can be easily implemented on currently available commercial units. It can generate any kind of path primitive that can be parametrized in function of the curvilinear coordinate.

The evaluation of the equivalent longitudinal bounds requires the knowledge of the inverse Jacobian matrix, so that the TSS is not suited to manage trajectories that directly cross singular points. Such trajectories can be handled by admitting minor path violations: Possible approaches to the problem are currently under investigation.

Another advantage of the novel TSS is represented by its flexibility. The number and the type of constraints that can be simultaneously handled is not limited to those that have been analyzed in the paper. Conversely, it can manage any constraint that can be converted into equivalent bounds for the LTL. This possibility will be used in the future, e.g., to improve the motion smoothness by also constraining jerks or, if the manipulator controller is accessible and modifiable, joint torques. However, the experimental evidence has proved that the proposed trajectory generator can guarantee sufficiently smooth motions, so that the additional complexity, that is required to handle further constraints, is justified only in case of specific applications.

References

[1] M. Mashaei and B. Lennartson, "Sustainable and robust control of cyclic pallet systems," *IEEE Trans. on Autom. Sci. and Eng.*, vol. 10, no. 4, pp. 916–927, 2013.

[2] H. Sun, Z. Hou, and D. Li, "Coordinated iterative learning control schemes for train trajectory tracking with overspeed protection," *IEEE Trans. on Autom. Sci. and Eng.*, vol. 10, no. 2, pp. 323–333, 2013.

[3] U. Schaper, E. Arnold, O. Sawodny, and K. Schneider, "Constrained real-time model-predictive reference trajectory planning for rotary cranes," in *IEEE/ASME Int. Conf. on Adv. Intel. Mech. (AIM13)*, 2013, pp. 680–685.

[4] D. Whitney, "Resolved motion rate control of manipulators and human prostheses ," *IEEE Trans. on Man-Mach. Sys.*, vol. MMS-10, no. 2, pp. 47–53, Jun. 1969.

[5] J. Baillieul, J. Hollerbach, and R. Brockett, "Programming and control of kinematically redundant manipulators," in *Proc. 23rd IEEE Conf. Decision Contr., CDC84*, Las Vegas, NV, Dec. 1984, pp. 768–774.

[6] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *Int. J. on Robot. Res.*, vol. 10, no. 4, pp. 410–425, 1991.

[7] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators ," *IEEE Tran. on Rob. and Autom.*, vol. 13, no. 3, pp. 398–410, 1997.

[8] P. From and J. Gravdahl, "A Real-Time Algorithm for Determining the Optimal Paint Gun Orientation in Spray Paint Applications," *IEEE Trans. on Autom. Science and Eng.*, vol. 7, no. 4, pp. 803–816, 2010.

[9] P. From, J. Gunnar, and J. Gravdahl, "Optimal Paint Gun Orientation in Spray Paint Applications #x2014;Experimental Results," *IEEE Trans. on Autom. Science and Eng.*, vol. 8, no. 2, pp. 438–442, Apr. 2011.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] Q. Zhang and S.-R. Li, "Efficient computation of smooth minimum time trajectory for CNC machining," *The Int. J. of Adv. Manuf. Tech.*, vol. 68, no. 1-4, pp. 683–692, 2013.

[22] 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.

[23] H. Arai, K. Tanie, and S. Tachi, "Path Tracking Control of a Manipulator Considering Torque Saturation," *IEEE Trans Ind Electron*, vol. 41, no. 1, pp. 25–31, February 1994.

[24] A. Bemporad, T.-J. Tarn, and N. Xi, "Predictive path parameterization for constrained robot control ," *IEEE Trans. on Contr. Sys. Tech.*, vol. 7, no. 6, pp. 648–656, Nov. 1999.

[25] G. Marani, J. Kim, J. Yuh, and W. Chung, "A real-time approach for singularity avoidance in resolved motion rate control of robotic manipulators ," in *Proc. 2002 IEEE Int. Conf. on Rob. & Autom., ICRA 2002*, Washington, DC, May 2002, pp. 1973–1978.

[26] C. Qiu, Q. Cao, and S. Miao, "An on-line task modification method for singularity avoidance of robot manipulators," *Robotica*, vol. 27, pp. 539–546, 2009.

[27] G. Antonelli, S. Chiaverini, and G. Fusco, "A new on-line algorithm for inverse kinematics of robot manipulators ensuring path tracking capability under joint limits," *IEEE Trans Robot Automat*, vol. 19, no. 1, pp. 162–167, Feb. 2003.

[28] F. Garelli, L. Gracia, A. Sala, and P. Albertos, "Sliding mode speed auto-regulation technique for robotic tracking ," *Robotics and Autonomous Systems*, vol. 59, pp. 519–529, 2011.

[29] C. Guarino Lo Bianco and F. Wahl, "A novel second order filter for the real-time trajectory scaling," in *IEEE Int. Conf. on Rob. and Autom., ICRA 2011*, Shanghai, China, May 2011, pp. 5813–5818.

[30] L. Sciavicco, B. Siciliano, L. Villani, and G. Oriolo, *Robotics: Modelling, planning and Control*, ser. Advanced Textbooks in Control and Signal Processing. Berlin, Germany: Springer-Verlag, 2011.

[31] 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.

APPENDIX

The estimation of the equivalent bounds for the scaling filter requires the online evaluation of the derivative of the Jacobian matrix expressed in function of curvilinear coordinate s. An efficient approach is proposed in the following.

As known, the Jacobian matrix associated to the end effector of any robotic manipulator which reference frames have been selected according to the modified Denavit-Hartenberg procedure can be expressed as follows

$$\mathbf{J}_T(\mathbf{q}) := \left[\begin{array}{c|c|c|c} \mathbf{j}_T^{v_1}(\mathbf{q}) & \mathbf{j}_T^{v_2}(\mathbf{q}) & \cdots & \mathbf{j}_T^{v_N}(\mathbf{q}) \\ \hline \mathbf{j}_T^{\omega_1}(\mathbf{q}) & \mathbf{j}_T^{\omega_2}(\mathbf{q}) & \cdots & \mathbf{j}_T^{\omega_N}(\mathbf{q}) \end{array}\right] , \qquad (42)$$

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

$$\mathbf{j}_T^{v_k}(\mathbf{q}) = \left\{ \begin{array}{ll} \hat{\mathbf{z}}_k(\mathbf{q}) & \text{prismatic joint} \\ \hat{\mathbf{z}}_k(\mathbf{q}) \times \overbrace{[\mathbf{p}_T(\mathbf{q}) - \mathbf{p}_k(\mathbf{q})]}^{\Delta_k(\mathbf{q})} & \text{revolute joint} \end{array} \right. , \qquad (43)$$

$$\mathbf{j}_T^{\omega_k}(\mathbf{q}) = \left\{ \begin{array}{ll} \mathbf{0} & \text{prismatic joint} \\ \hat{\mathbf{z}}_k(\mathbf{q}) & \text{revolute joint} \end{array} \right. , \qquad (44)$$

and where N is the number of independent joints, $\hat{\mathbf{z}}_k(\mathbf{q}) = {}^0_k\mathbf{R}(\mathbf{q})\,\hat{\mathbf{z}}^*$ (with $\hat{\mathbf{z}}^* := [0\ 0\ 1]^T$) is the $\hat{\mathbf{z}}$ unit vector of the kth reference frame, $\mathbf{p}_k(\mathbf{q})$ indicates the origin of the kth frame, and, finally, $\mathbf{p}_T(\mathbf{q})$ is the origin of the frame that must follow the assigned trajectory, i.e., frame N or the tool frame. All vectors are described with respect to an inertial system that is typically located on the manipulator base.

If the manipulator is moving along an assigned path that is parametrized through a curvilinear coordinate s, it is possible to compute, for any given s, the corresponding position $\mathbf{q}(s)$ in the joint space by solving an inverse kinematic problem, then, by means of (42), it is also possible to evaluate $\mathbf{J}_T(s) = \mathbf{J}_T[\mathbf{q}(s)]$. Thus, in the following, $\mathbf{j}_T^{v_k}(s)$, $\mathbf{j}_T^{\omega_k}(s)$, $\Delta_k(s) := \mathbf{p}_T[\mathbf{q}(s)] - \mathbf{p}_k[\mathbf{q}(s)]$, and, consequently $\mathbf{J}_T(s)$, are supposed to be known.

The derivative of the Jacobian matrix is derived by differentiating (43) and (44). It can be represented as follows (dependency on $\mathbf{q}$ has been dropped for conciseness)

$$\dot{\mathbf{J}}_T := \left[\begin{array}{c} \dot{\mathbf{J}}_{v_T} \\ \hline \dot{\mathbf{J}}_{\omega_T} \end{array}\right] = \left[\begin{array}{c|c|c|c} \dot{\mathbf{j}}_T^{v_1} & \dot{\mathbf{j}}_T^{v_2} & \cdots & \dot{\mathbf{j}}_T^{v_N} \\ \hline \dot{\mathbf{j}}_T^{\omega_1} & \dot{\mathbf{j}}_T^{\omega_2} & \cdots & \dot{\mathbf{j}}_T^{\omega_N} \end{array}\right] , \qquad (45)$$

where

$$\dot{\mathbf{j}}_T^{v_k} = \left\{ \begin{array}{ll} \dot{\hat{\mathbf{z}}}_k & \text{prismatic joint} \\ \dot{\hat{\mathbf{z}}}_k \times [\mathbf{p}_T - \mathbf{p}_k] + \hat{\mathbf{z}}_k \times [\mathbf{v}_T - \mathbf{v}_k] & \text{revolute joint} \end{array} \right. \qquad (46)$$

$$\dot{\mathbf{j}}_T^{\omega_k}(\mathbf{q}) = \left\{ \begin{array}{ll} \mathbf{0} & \text{prismatic joint} \\ \dot{\hat{\mathbf{z}}}_k & \text{revolute joint} \end{array} \right. . \qquad (47)$$

Term $\dot{\hat{\mathbf{z}}}_k$ can be obtained as follows

$$\dot{\hat{\mathbf{z}}}_k = \frac{d}{dt}\left({}^0_k\mathbf{R}\,\hat{\mathbf{z}}^*\right) = \frac{d}{dt}\left({}^0_k\mathbf{R}\right)\,\hat{\mathbf{z}}^* = \mathbf{S}(\boldsymbol{\omega}_k)\,{}^0_k\mathbf{R}\,\hat{\mathbf{z}}^* = \boldsymbol{\omega}_k \times \hat{\mathbf{z}}_k \ , \qquad (48)$$

where $\boldsymbol{\omega}_k$ is the angular velocity of the kth reference frame. It is known that any generic $\boldsymbol{\omega}_i$, $i = 1, 2, \ldots, N$ can be evaluated by means of an appositely devised Jacobian matrix [30] according to the following equation

$$\boldsymbol{\omega}_i = \sum_{k=1}^{i} \mathbf{j}_i^{\omega_k}\,\dot{q}_k = \underbrace{\left[\begin{array}{c|c|c|c|c|c} \mathbf{j}_i^{\omega_1} & \mathbf{j}_i^{\omega_2} & \cdots & \mathbf{j}_i^{\omega_i} & \mathbf{0} & \cdots & \mathbf{0} \end{array}\right]}_{\mathbf{J}_{\omega_i}} \dot{\mathbf{q}} \qquad (49)$$

where terms $\mathbf{j}_i^{\omega_k}$ coincide with terms $\mathbf{j}_T^{\omega_k}$ of $\mathbf{J}_T(\mathbf{q})$.

In the same way, the linear velocity of the ith reference frame can be expressed as follows

$$\mathbf{v}_i = \sum_{k=1}^{i} \mathbf{j}_i^{v_k}\, \dot{q}_k = \underbrace{\left[\; \mathbf{j}_i^{v_1} \;\middle|\; \mathbf{j}_i^{v_2} \;\middle|\; \cdots \;\middle|\; \mathbf{j}_i^{v_i} \;\middle|\; \mathbf{0} \;\middle|\; \cdots \;\middle|\; \mathbf{0} \;\right]}_{\mathbf{J}_{v_i}} \dot{\mathbf{q}}\,. \tag{50}$$

Terms $\mathbf{j}_i^{v_k}$ of $\mathbf{J}_{v_k}$ coincide with terms $\mathbf{j}_T^{v_k}$ of $\mathbf{J}_T$ only in case of prismatic joints. In general, they can be evaluated by means of the following expression

$$\mathbf{j}_i^{v_k}(\mathbf{q}) = \begin{cases} \hat{\mathbf{z}}_k(\mathbf{q}) & \text{prismatic joint} \\ \hat{\mathbf{z}}_k(\mathbf{q}) \times [\mathbf{p}_i(\mathbf{q}) - \mathbf{p}_k(\mathbf{q})] & \text{revolute joint} \end{cases}. \tag{51}$$

Equations (46) and (47), with the aid of (48), (49), and (50), can be rewritten in the following form

$$\dot{\mathbf{j}}_T^{v_k}(\mathbf{q}) = \begin{cases} \mathbf{J}_{\omega_k}\, \dot{\mathbf{q}} \times \hat{\mathbf{z}}_k & \text{prismatic joint} \\ (\mathbf{J}_{\omega_k}\, \dot{\mathbf{q}} \times \hat{\mathbf{z}}_k) \times [\mathbf{p}_T - \mathbf{p}_k] + \hat{\mathbf{z}}_k \times [\mathbf{J}_{v_T} - \mathbf{J}_{v_k}]\, \dot{\mathbf{q}} & \text{revolute joint} \end{cases} \tag{52}$$

$$\dot{\mathbf{j}}_T^{\omega_k}(\mathbf{q}) = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \mathbf{J}_{\omega_k}\, \dot{\mathbf{q}} \times \hat{\mathbf{z}}_k & \text{revolute joint} \end{cases}. \tag{53}$$

Equations (52) and (53) can finally be posed in function of the longitudinal coordinate s. The resulting expressions must be computationally efficient, since the algorithm is executed in real time. Fortunately, many terms in (52) and in (53) are common to other expressions of the TSS, so that they need to be computed only once. For example, according to (24), we can write $\dot{\mathbf{q}}(s) = \mathbf{a}(s)\, \dot{s}$, where $\mathbf{a}(s)$ is a term that appears in many part of the algorithm. Analogously, $\hat{\mathbf{z}}_k$ is also a terms of $\mathbf{J}_T(s)$: For revolute joints $\hat{\mathbf{z}}_k(s) = \mathbf{j}_T^{\omega_k}(s)$, while $\hat{\mathbf{z}}_k(s) = \mathbf{j}_T^{v_k}(s)$ for prismatic joints. The same happens for $\Delta_k(s) = \mathbf{p}_T(s) - \mathbf{p}_k(s)$, while the columns of $\mathbf{J}_{\omega_k}(s)$ are common to $\mathbf{J}_{\omega_T}(s)$.

After a few algebraic manipulations it is thus possible to rewrite (52) and (53) as follows

$$\dot{\mathbf{j}}_T^{v_k}(s, \dot{s}) = \dot{s} \begin{cases} \mathbf{J}_{\omega_k}(s)\, \mathbf{a}(s) \times \mathbf{j}_T^{v_k}(s) & \text{prismatic joint} \\ [\mathbf{J}_{\omega_k}(s)\, \mathbf{a}(s) \times \mathbf{j}_T^{\omega_k}(s)] \times \Delta_k(s) & \\ \quad + \mathbf{j}_T^{\omega_k}(s) \times [\mathbf{J}_{v_T}(s) - \mathbf{J}_{v_k}(s)]\, \mathbf{a}(s) & \\ & \text{revolute joint} \end{cases}, \tag{54}$$

$$\dot{\mathbf{j}}_T^{\omega_k}(s, \dot{s}) = \dot{s} \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \mathbf{J}_{\omega_k}(s)\, \mathbf{a}(s) \times \mathbf{j}_T^{\omega_k}(s) & \text{revolute joint} \end{cases}. \tag{55}$$

The sole term $\mathbf{J}_{v_k}(s)$ actually needs to be evaluated from scratch. Consequently, it is possible to conclude that a minor additional burden is required for the evaluation of $\dot{\mathbf{J}}_T(s, \dot{s})$.

Equations (54) and (55) also reveal that the derivative of the Jacobian matrix can always be expressed as follows

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