# Nonlinear Variable Structure Filter for the Online Trajectory Scaling

Oscar Gerelli and Corrado Guarino Lo Bianco, *Member, IEEE*

***Abstract*—Time efficiency and accurate path tracking represent two conflicting demands typical of robotic applications: Time efficiency induces one to plan extremely fast trajectories which can easily collide with the manipulator kinematic and dynamic constraints, thus causing a reduction of accuracy. To deal with this problem, several approaches can be found in the literature mainly based on the synthesis of dynamic filters used for the online trajectory scaling: A possibly unfeasible input trajectory is automatically scaled to fulfill given dynamic bounds. In this way, an accurate path tracking is guaranteed. This paper can be collocated in such a framework. A new discrete-time filter, with novel capabilities, is designed. Differently from other proposals, not only torque constraints are considered but also kinematic constraints are easily handled. Moreover, to preserve time efficiency, the new filter always attempts to recover any delay caused by the constraints.**

***Index Terms*—Industrial manipulators, kinematic and dynamic bounds, nonlinear tracking filters, path tracking, trajectory scaling, variable structure systems.**

## I. Introduction

MOTION control of industrial manipulators requires the generation of appropriate reference signals in order to improve the system performances in terms of precision and time efficiency. Great attention is commonly devoted to design trajectories able to minimize the time required to complete an assigned task. The fulfillment of this requirement is crucial in order to increase the production rate in industrial applications which are often limited by the robot performances rather than the process constraints.

In the last decades, several algorithms have been proposed for the optimal trajectory planning. Some of them evaluate the trajectory as a whole [1]–[4], while others take advantage of the path-velocity paradigm [5]–[7], and then introduce a scaling factor to guarantee the trajectory feasibility [8]–[10].

Unfortunately, traveling time minimization leads to an increment of the mechanical solicitations and, moreover, the actuators dynamic limits can easily be exceeded, thus causing a degeneration of the control performances. For example, when minimum-time trajectories are planned under dynamic constraints, there is always at least one joint working at its torque limits: External disturbances and robot unmodeled dynamics cannot be compensated by the controller, likely causing a path tracking lost.

Manuscript received June 15, 2008; revised March 9, 2009. First published April 7, 2009; current version published September 16, 2009. This work was supported in part by AER-TECH LAB, Emilia Romagna, Italy.

The authors are with the Dipartimento di Ingegneria dell'Informazione, University of Parma, 43039 Parma, Italy (e-mail: gerelli@ce.unipr.it; guarino@ce.unipr.it).

Digital Object Identifier 10.1109/TIE.2009.2018431

To overcome these possible problems, nominal trajectories are typically online modified by means of appropriate methods. In the framework of the path-velocity decomposition [5], a good path tracking is normally guaranteed by suitably scaling the longitudinal time-law (LTL) used to move the manipulator along the assigned path. Sometimes, the LTL is only scaled to preserve given kinematic bounds [11], [12], sometimes it is modified by accounting for dynamic limits [13]–[15]. When dynamic bounds are considered, they are first online converted into equivalent kinematics constraints on the longitudinal velocity profile. Then, appositely designed dynamic systems are used to "filter" the original LTL and generate output reference signals which fulfill the assigned bounds.

This paper nests in such framework. The proposed control scheme introduces several novelties with respect to similar approaches [13]–[15]. First of all, in addition to usual torque constraints, explicit limits on the maximum joint velocities are considered. Second, a different parametrization of the LTL is used. The time-law is commonly expressed as a function of the manipulator position along the path [13] by specifying the longitudinal velocity to be assumed in each point. In this paper, it is defined by means of a standard trajectory specified in the time domain. The difference is relevant. In the first case, time delays caused by the trajectory scaling mechanism cannot be recovered: As long as saturations cease, the system automatically assume the velocity planned for the current path position, so that time delays accumulate along the trajectory, thus reducing productivity. In the second case, any delay accumulated due to saturations is extinguished as soon as dynamic conditions make it possible: Efficiency is preserved and, at the same time, a good path tracking is achieved. As a further advantage, the longitudinal reference signal is generated in a natural way by means of standard planning methods.

The dynamic filter used to scale the LTL represents itself a novelty. Similar solutions were proposed in the past, dealing with the optimal filtering of rough reference signals for servosystems [16]–[18]. Unfortunately, none of them is suited to directly manage the asymmetric kinematic bounds which characterize the proposed robotic problem. For this reason, a new filter, featuring an improved control law, has been designed.

This paper is organized as follows. The robotic problem is posed in Section II. In the same section, it is shown how joint torque and velocity constraints can be converted into equivalent kinematic constraints. Such constraints are used to scale the LTL by means of a dynamic filter: The characteristics of the new discrete-time filter are discussed in Section III, while a detailed analysis of the filter convergence properties is reported

in Appendix I-A. The usefulness of the approach is investigated in Section IV by means of an example concerning a Cartesian manipulator: simulation results are proposed and analyzed. Finally, conclusions are drawn in Section V.

## II. Online Trajectory Scaling for Robotic Manipulators

The problem here investigated is similar to that described in [13], where an online strategy was proposed to scale the velocity profile of any given input trajectory such to fulfill known system constraints. To this purpose a two-level control scheme was designed. At the primary level, a standard feedback controller was adopted, tuned for disturbances rejection and good transient performances. At the secondary level, a dynamic filter was used to modify the nominal, and potentially rough, LTL in order to fulfill the manipulator torque constraints and track, at the best, a given path.

In this paper, the same two-level approach is used. The first level is represented by a standard computed torque controller, while a novel filter is used for the optimal trajectory scaling.

Some preliminary definitions can be useful for the discussion. The robot trajectory is defined according to the so-called path-velocity decomposition [5]. For this reason, the path to be followed is described in the joint space by means of a vectorial function $\mathbf{\Gamma}(x)$ defined as follows:

$$\begin{array}{ccc} \mathbf{\Gamma} : [0, x_f] & \rightarrow & \mathbb{R}^n \\ x & \rightarrow & \mathbf{q}_d := \mathbf{\Gamma}(x) \end{array} \tag{1}$$

where $x \in \mathbb{R}$ is the scalar which parametrizes the curve, while $n \in \mathbb{N}$ is the number of independent joints. Without any loss of generality, the path is assigned in the joint space.

A monotonically increasing LTL, defined in the time domain, is used to move the end effector along $\mathbf{\Gamma}(x)$. It is defined as follows:

$$\begin{array}{ccc} x : [0, t_f] & \rightarrow & [0, x_f] \\ t & \rightarrow & x_d := x(t) \end{array} \tag{2}$$

where $t_f$ is the total traveling time. Evidently, the overall robot trajectory is obtained by combining (1) and (2): $\mathbf{q}_d(t) := \mathbf{\Gamma}(x(t))$. By taking into account the chain differentiation rule, it is possible to evaluate the trajectory time derivatives

$$\dot{\mathbf{q}}_d = \mathbf{\Gamma}'(x)\dot{x} \tag{3}$$

$$\ddot{\mathbf{q}}_d = \mathbf{\Gamma}''(x)\dot{x}^2 + \mathbf{\Gamma}'(x)\ddot{x}. \tag{4}$$

Superscript $'$ indicates a differentiation with respect to $x$, e.g., $\mathbf{\Gamma}b'(x) = (d\mathbf{\Gamma}(x)/dx)$, while, as usual, dots indicate time derivatives, e.g., $\dot{x}(t) = (dx(t)/dt)$.

Consider a serial link rigid-body manipulator, whose dynamic is described by

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{F}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{5}$$

where $\mathbf{q} := [q_1, q_2, \ldots, q_n]^{\mathrm{T}} \in \mathbb{R}^n$ is the vector of the joint variables, $\boldsymbol{\tau} := [\tau_1, \tau_2, \ldots, \tau_n]^{\mathrm{T}} \in \mathbb{R}^n$ is the vector of the applied torques, $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the symmetric positive-definite inertia matrix, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ is the vector of Coriolis and centripetal forces, $\mathbf{F}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ describes the viscous friction, and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ is the vector of gravitational forces. The manipulator is subject to dynamic and kinematic constraints. More precisely, maximum admissible torques are bounded, so that it is possible to write

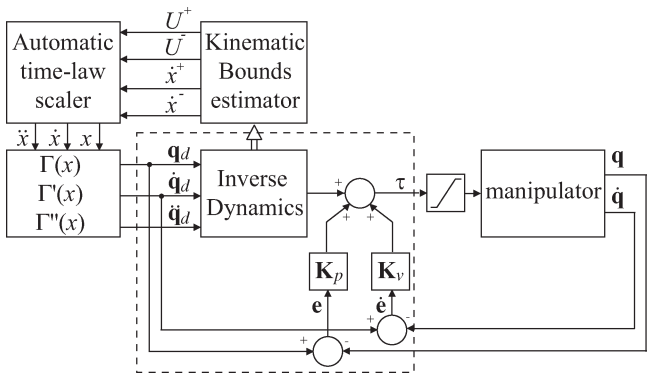


Fig. 1. Proposed control scheme. (Dashed box) A standard computed torque controller is used to drive the manipulator, in conjunction with the ATLS proposed in this paper.

$$\underline{\tau}_i \leq \tau_i \leq \overline{\tau}_i, \qquad i = 1, 2, \ldots, n \tag{6}$$

where $\underline{\tau}_i \in \mathbb{R}^-$ and $\overline{\tau}_i \in \mathbb{R}^+$ represent the lower and the upper bounds of the $i$th joint torque. Analogously, maximum joint velocities are bounded, i.e.,

$$\underline{\dot{q}}_i \leq \dot{q}_i \leq \overline{\dot{q}}_i, \qquad i = 1, 2, \ldots, n \tag{7}$$

where $\underline{\dot{q}}_i \in \mathbb{R}^-$ and $\overline{\dot{q}}_i \in \mathbb{R}^+$ represent the lower and the upper limits of the $i$th joint velocity.

Owing to (6) and (7), the following tracking problem can be defined.

*Problem 1:* Given a manipulator described by (5) and a desired trajectory (1) and (2), design a control law to achieve the best possible path tracking compatibly with torque constraints (6) and joint velocity constraints (7).

The control scheme proposed to deal with Problem 1 is shown in Fig. 1. As anticipated, it is based on a computed torque controller. The controller output is saturated to account for (6), while the manipulator internal dynamics has been modified in order to introduce the effects of (7). If an improper trajectory is used to drive the torque controller, saturations cause a drastic degeneration of the tracking performances. The proposed trajectory scaling system guarantees that reference signals used to drive the controller are always feasible. It is based on a three-stage scheme. The first stage is a kinematic bounds estimator (KBE) which online converts, on the basis of the manipulator current state of motion, bounds (6) and (7) into equivalent kinematics constraints for $x(t)$ or, better, for $\dot{x}(t)$ and $\ddot{x}(t)$. Such constraints are then used by the automatic time-law scaler (ATLS) to modify any given nominal, but possibly unfeasible, reference signal $r(t)$ and generate a feasible time-law $x(t)$. Finally, $x(t)$ is converted by the third stage into a feasible trajectory by means of (1), (3), and (4). Equations required for the KBE are devised in the following, while the ATLS is described in Section III.

The output signal of a computed torque controller is evaluated according to

$$\boldsymbol{\tau}=\mathbf{H}(\mathbf{q}_d)\ddot{\mathbf{q}}_d+\mathbf{c}(\mathbf{q}_d,\dot{\mathbf{q}}_d)+\mathbf{F}(\mathbf{q}_d)\dot{\mathbf{q}}_d+\mathbf{g}(\mathbf{q}_d)+\mathbf{K}_p^{\mathrm{T}}\mathbf{e}+\mathbf{K}_v^{\mathrm{T}}\dot{\mathbf{e}} \tag{8}$$

where $\mathbf{K}_p, \mathbf{K}_v \in \mathbb{R}^n$ are the controller feedback gains and where $\mathbf{e} := \mathbf{q}_d - \mathbf{q}$ is the reference tracking error, while $\dot{\mathbf{e}} := \dot{\mathbf{q}}_d - \dot{\mathbf{q}}$ is its first derivative. Due to (1), (3), and (4), it is always possible to rewrite (8) as follows:

$$\boldsymbol{\tau} = \mathbf{b}_1(x)\ddot{x} + \mathbf{b}_2(x,\dot{x}) \tag{9}$$

where

$$\mathbf{b}_1(x) := \mathbf{H}\left(\boldsymbol{\Gamma}(x)\right)\boldsymbol{\Gamma}'(x) \tag{10}$$

$$\begin{aligned}\mathbf{b}_2(x,\dot{x},\mathbf{q},\dot{\mathbf{q}}) :=& \mathbf{H}\left(\boldsymbol{\Gamma}(x)\right)\boldsymbol{\Gamma}''(x)\dot{x}^2 + \mathbf{c}\left(\boldsymbol{\Gamma}(x),\boldsymbol{\Gamma}'(x)\dot{x}\right)\\ &+ \mathbf{F}\left(\boldsymbol{\Gamma}(x)\right)\boldsymbol{\Gamma}'(x)\dot{x} + \mathbf{g}\left(\boldsymbol{\Gamma}(x)\right)\\ &+ \mathbf{K}_p^{\mathrm{T}}\mathbf{e} + \mathbf{K}_v^{\mathrm{T}}\dot{\mathbf{e}}.\end{aligned} \tag{11}$$

Let us define $[\,b_{1,1} \quad b_{1,2} \quad \cdots \quad b_{1,n}\,]^{\mathrm{T}} := \mathbf{b}_1(x)$ and $[\,b_{2,1} \quad b_{2,2} \quad \cdots \quad b_{2,n}\,]^{\mathrm{T}} := \mathbf{b}_2(x,\dot{x},\mathbf{q},\dot{\mathbf{q}})$. Due to (9), constraints (6) can be rewritten as follows:

$$\underline{\tau}_i \le b_{1,i}\,\ddot{x} + b_{2,i} \le \overline{\tau}_i, \qquad i = 1,2,\ldots,n. \tag{12}$$

By reorganizing (12), it is possible to convert $\underline{\tau}_i$ and $\overline{\tau}_i$ into equivalent bounds $\psi_i$ and $\phi_i$ on $\ddot{x}$. In particular, given the current status of motion $(x,\dot{x})$, for each joint $i = 1,2,\ldots,n$, it is necessary to guarantee that $\ddot{x} \in [\psi_i, \phi_i]$, where

$$\phi_i = \begin{cases} \frac{\overline{\tau}_i - b_{2,i}}{b_{1,i}}, & \text{if } b_{1,i} > 0 \\ \frac{\underline{\tau}_i - b_{2,i}}{b_{1,i}}, & \text{if } b_{1,i} < 0 \\ \infty, & \text{if } b_{1,i} = 0 \end{cases}$$

$$\psi_i = \begin{cases} \frac{\underline{\tau}_i - b_{2,i}}{b_{1,i}}, & \text{if } b_{1,i} > 0 \\ \frac{\overline{\tau}_i - b_{2,i}}{b_{1,i}}, & \text{if } b_{1,i} < 0 \\ -\infty, & \text{if } b_{1,i} = 0. \end{cases}$$

Since bounds $\phi_i$ and $\psi_i$ must be simultaneously fulfilled $\forall i = 1,2,\ldots,n$, any feasible acceleration $\ddot{x}$ must belongs to the range $[U^-, U^+]$, where

$$U^+ := \min_{i=1,\ldots,n}\{\phi_i\} \quad U^- := \max_{i=1,\ldots,n}\{\psi_i\}. \tag{13}$$

Named $\Gamma'_i(x)$ the $i$ element of $\Gamma b'(x)$, due to (3), it is possible to rewrite (7) as follows:

$$\underline{\dot{q}}_i \le \Gamma'_i(x)\dot{x} \le \dot{\overline{q}}_i, \qquad i = 1,2,\ldots,n. \tag{14}$$

By means of (14) it is then possible to convert the bounds on $\dot{q}_i$ into equivalent bounds $\dot{x}^-$ and $\dot{x}^+$ on $\dot{x}$. First of all, since negative velocities along the path are not acceptable, it is necessary to assign $\dot{x}^- := 0$. Upper bound $\dot{x}^+$ is then evaluated according to the following expressions:

$$\rho_i = \begin{cases} \frac{\dot{\overline{q}}_i}{\Gamma'_i(x)}, & \text{if } \Gamma'_i(x) > 0 \\ \frac{\underline{\dot{q}}_i}{\Gamma'_i(x)}, & \text{if } \Gamma'_i(x) < 0 \\ \infty, & \text{if } \Gamma'_i(x) = 0. \end{cases}$$

Velocity $\dot{x}$ is feasible only if it lies in the interval $[\dot{x}^-, \dot{x}^+]$, where

$$\dot{x}^+ := \min_{i=1,\ldots,n}\{\rho_i\}, \qquad \dot{x}^- := 0. \tag{15}$$

It is important to point out that the solution of Problem 1 could possibly not exist depending on the current status of motion [6]. More precisely, depending on the current values of $x$ and $\dot{x}$, and assigned a set of torque bounds $\underline{\tau}_i$, $\overline{\tau}_i$ and velocity bounds $\underline{\dot{q}}_i$, $\dot{\overline{q}}_i$, it could not exist any $\ddot{x}$ such that (12) is fulfilled. In this case, independently from the adopted scaling method, path tracking is lost. A feasible solution certainly exists until condition $U^+ \ge U^-$ is satisfied.

## III. Automatic Time-Law Scaler

The ATLS is composed by a reference LTL generator, whose output is a signal $r(t)$, and a dynamic nonlinear discrete-time filter, which appropriately modifies $r(t)$ in order to fulfill (13) and (15). In particular, if $r(t)$ is feasible the filter output is $x(t) = r(t)$, otherwise $r(t)$ is abandoned and an appropriate $x(t)$, which fulfill the constraints, is generated.

Similar dynamic filters have been successfully used in the past to generate smooth set points for motion control systems [16]–[18]. None of them can be used in this context since they are not able to handle asymmetric bounds on the acceleration like those deriving from (13). Since manipulator controllers are discrete-time implemented, a sampled time filter is proposed to solve the following design problem.

*Problem 2:* Design a nonlinear discrete-time filter whose output $x(t)$ tracks "at best" a given reference signal $r(t)$ by fulfilling the following requirements.

1) The first and second time derivatives of $x$ must be bounded

$$\dot{x}^- \le \dot{x} \le \dot{x}^+, \qquad U^- \le \ddot{x} \le U^+ \tag{16}$$

where $\dot{x}^-$, $\dot{x}^+ \in \mathbb{R}$, $U^+ \in \mathbb{R}^+$ and $U^- \in \mathbb{R}^-$.
2) Bounds (16) can be time varying and can also change during transients.
3) If (16) is not satisfied owing to the filter initial conditions or to a sudden change of the bounds, $\ddot{x}$ must be forced in a single step within the given limits, while $\dot{x}$ must reach the assigned bounds in minimum time.
4) When a reference signal $r$ satisfying (16) is applied, the tracking condition $x = r$ is reached in minimum time and without overshoot.
5) When a discontinuous reference signal is applied (or the reference signal has time derivatives larger than the bound values), the tracking is lost. As soon as the reference signal newly satisfies (16), tracking is achieved in minimum time.
6) The time derivatives $\dot{x}$ and $\ddot{x}$ of the bounded output must be available for the generation of feedforward actions.

Problem 2 is an optimal minimum-time tracking problem subject to bounded dynamics. Its solution is based on a chain

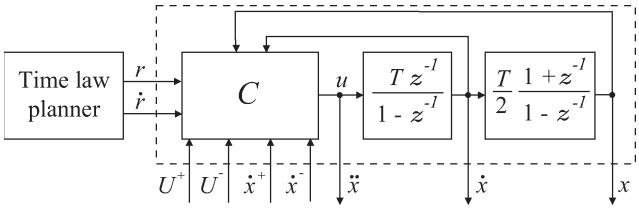


Fig. 2. Optimal bounded-dynamics trajectory tracker.

of two integrators like that shown in Fig. 2, whose dynamic equation is

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} u_k \tag{17}$$

where $T$ is the system sampling time, $(x, \dot{x})$ is the internal state, while $u$ is the control command of the integrator chain. Subscript $k$ indicates the sample number, so that $u_k$ represents the command signal at time $t_k = kT$.

The integrators are driven by an algebraic discrete-time nonlinear controller $C$ designed by means of variable structure control techniques [19]. In order to fulfill the requirements of Problem 2, the following control law $C$ is proposed:

$$C: \quad u_k := \begin{cases} U^- \text{sat}(\sigma_k), & \text{if } \sigma_k \geq 0 \\ -U^+ \text{sat}(\sigma_k), & \text{if } \sigma_k < 0 \end{cases} \tag{18}$$

$$\sigma_k := \dot{z}_k - \dot{\tilde{z}}_k \tag{19}$$

where $\dot{z}_k$ and $\dot{\tilde{z}}_k$ are evaluated by means of the following expressions:

$$\dot{z}^+ := -\frac{\dot{x}^+ - \dot{r}_k}{TU^-} \tag{20}$$

$$z^+ := -\lceil \dot{z}^+ \rceil \left[ \dot{z}^+ - \frac{\lceil \dot{z}^+ \rceil - 1}{2} \right] \tag{21}$$

$$\dot{z}^- := \frac{\dot{x}^- - \dot{r}_k}{TU^+} \tag{22}$$

$$z^- := \lceil -\dot{z}^- \rceil \left[ -\dot{z}^- - \frac{\lceil -\dot{z}^- \rceil - 1}{2} \right] \tag{23}$$

$$[\alpha \ \beta] := \begin{cases} [U^+ \quad U^-], & \text{if } \frac{y_k}{T} + \frac{\dot{y}_k}{2} > 0 \\ [U^- \quad U^+], & \text{if } \frac{y_k}{T} + \frac{\dot{y}_k}{2} \leq 0 \end{cases} \tag{24}$$

$$z_k := \frac{1}{T\alpha} \left| \frac{y_k}{T} + \frac{\dot{y}_k}{2} \right| \tag{25}$$

$$\gamma_k := \begin{cases} z^+, & \text{if } z_k < z^+ \\ z_k & \text{if } z^+ \leq z_k \leq z^- \\ z^-, & \text{if } z_k > z^- \end{cases} \tag{26}$$

$$m_k := \left\lfloor \frac{1 + \sqrt{1 + 8\,|\gamma_k|}}{2} \right\rfloor \tag{27}$$

$$\dot{\tilde{z}}_k := -\frac{\gamma_k}{m_k} - \frac{m_k - 1}{2} \text{sgn}(\gamma_k) \tag{28}$$

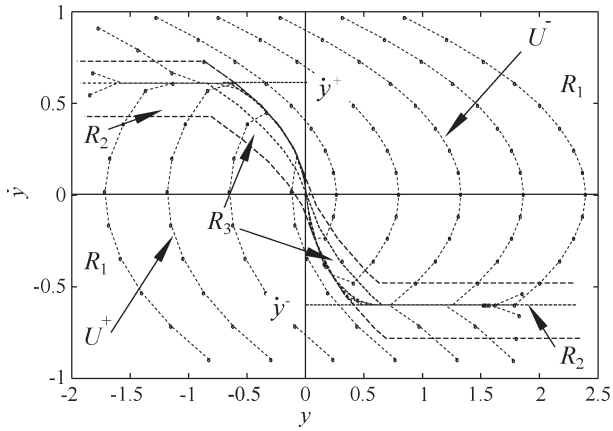


Fig. 3. $(y, \dot{y})$ phase plane.

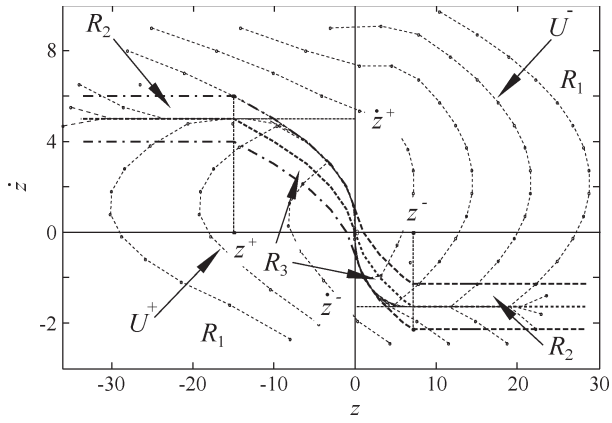


Fig. 4. $(z, \dot{z})$ phase plane.

$$\dot{z}_k := \begin{cases} \frac{\dot{y}_k}{T|\alpha|}, & \text{if } \left[ \left( z_k \geq 0 \ \& \ \frac{\dot{y}_k}{T|\alpha|} \leq \dot{\tilde{z}}_k \right) \right. \\ & \left. \text{or} \left( z_k < 0 \ \& \ \frac{\dot{y}_k}{T|\alpha|} \geq \dot{\tilde{z}}_k \right) \right] \\ \frac{\dot{y}_k}{T|\beta|} + \left( \frac{m_k - 1}{2} + \frac{|\gamma_k|}{m_k} \right) & \\ \quad \times \frac{\alpha + \beta}{|\beta|}, & \text{otherwise} \end{cases} \tag{29, 30}$$

and where $r_k$ is the sampled reference signal, $\dot{r}_k$ is the corresponding discrete-time derivative, $y_k := x_k - r_k$ is the filter tracking error, $\dot{y}_k := \dot{x}_k - \dot{r}_k$ is the filter velocity error. Functions $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ respectively provide the ceil and the floor of their arguments, provides the upper integer of its argument, while $\text{sat}(\cdot)$ saturates its argument to $\pm 1$. Signals $r_k$ and $\dot{r}_k$ are assumed to be known. Moreover, $\dot{r}_k$ is supposed to be piecewise constant.

The filter behavior is summarized in the following with the help of Figs. 3 and 4. The interested reader can find the analytic demonstrations of the filter convergence properties in Appendix I-A.

The objective of controller $C$ is to force the system state $(y, \dot{y})$ toward the origin of the phase plane since this implies, according to the definition of $y$ and $\dot{y}$, that a perfect tracking of $r$ is achieved. This result must be obtained in minimum time and by satisfying, if possible, the given constraints on the maximum velocity and acceleration. To this purpose, any point in the $(y, \dot{y})$-plane is transformed into an equivalent one in the

$(z, \dot{z})$-space by means of (20)–(30). It is possible to verify that such mapping is bijective, and the origins of the two spaces coincide. As a consequence, tracking is achieved if controller $C$ is able to force the state $(z, \dot{z})$ and, in turn, $(y, \dot{y})$ toward the origin. The two constants $\dot{z}^+$ and $\dot{z}^-$ represent the transformed values in the $(z, \dot{z})$-plane of velocity constraints $\dot{x}^+$ and $\dot{x}^-$. Analogously, $\dot{y}^+$ and $\dot{y}^-$ represent the transformed values in the $(y, \dot{y})$-plane of the same constraints.

From a practical point of view, control law (18)–(30) creates a sliding surface in the phase plane, whose equation, due to (19), is clearly given by (28). The sliding surface has been planned such that it monotonically decreases when $z \in [z^+, z^-]$, while it becomes constant and equal to $\dot{\tilde{z}} = \dot{z}^+$ if $z \leq z^+$ or $\dot{\tilde{z}} = \dot{z}^-$ if $z \geq z^-$. Sliding surface $\dot{\tilde{z}}$ is surrounded by a boundary layer (BL): If the filter state is outside such BL, the command signal is $u = U^+$ or $u = U^-$, otherwise $u$ lies in the interval $[U^-, U^+]$. In this way, being $\ddot{x} = u$, the constraint on the maximum acceleration is automatically fulfilled.

Fig. 3 shows some system trajectories in the $(y, \dot{y})$-plane by considering different starting conditions, while Fig. 4 shows the same trajectories in the transformed $(z, \dot{z})$-plane. From the two figures, it is possible to deduce that the origin of the $(z, \dot{z})$-plane is reached in two steps: The system state is first driven toward $\dot{\tilde{z}}$, then it slides along such surface by pointing to the origin. When outside the BL (region $R_1$), transients are obtained by applying the maximum command signal: The BL is reached with certainty and in minimum time, as demonstrated in Appendix I-B. Control law $C$ guarantees that the BL cannot be crossed: as soon as the system state reaches region $R_2$, with a single step, it is forced to the sliding surface and, then, it slides toward the origin with command signal $u = 0$ (see Appendix I-C). Finally, $(z, \dot{z})$ enters in region $R_3$ and, again with a single step, it is forced to the frontier of the BL: The origin is reached by applying the maximum command signal and with a deadbeat behavior (see Appendix I-D). Apart from the two single-step transients from $R_1$ to $R_2$ and from $R_2$ to $R_3$, the command signal is always $u \in \{U^-, 0, U^+\}$, i.e., the controller has a bang-zero-bang behavior.

From Figs. 3 and 4, it can be evinced that if the constraint on the maximum velocity is violated, e.g., $\dot{z} \notin [\dot{z}^-, \dot{z}^+]$ for a sudden change of the given bounds, the system is forced within the new bounds by applying the maximum control action, i.e., in minimum time as required by point 3) of Problem 2.

Some applicative remarks can be useful in order to adopt the filter for actual applications. Difference $U^+ - U^-$ depends on the distance of filter state $x, \dot{x}$ from the boundary of the feasible region: If the difference tends to zero, the state is approaching the frontier. Under this condition, due to the limitedness of available dynamics (remember that $\ddot{x} \in [U^-, U^+]$), boundary could be possibly crossed, thus causing a torque constraints violation. This situation should be avoided. Owing to the characteristics of the proposed filter, several solutions are possible, all of them based on an appropriate reduction of $\dot{x}^+$. Indeed, interval $[U^-, U^+]$ enlarges as soon as current $\dot{x}$ is reduced, so that feasibility can be more easily maintained.

Filter stability requires $\dot{z}^+ \in \mathbb{R}^+$ and $\dot{z}^- \in \mathbb{R}^-$ (see Appendix I-A), so that $\dot{r}$ must necessarily fulfill the following condition owing to (20) and (22):

$$\dot{x}^- \leq \dot{r} \leq \dot{x}^+. \tag{31}$$

Inequality (31) cannot be guaranteed *a priori* since $\dot{x}^+$ is continuously modified by the KBE. The solution to this problem is straightforward, since it is sufficient to saturate $\dot{r}$ within $[\dot{x}^-, \dot{x}^+]$: tracking is lost, but the filter remains stable, and $r(t)$ is newly hooked as soon as $\dot{r}$ returns inside interval $[\dot{x}^-, \dot{x}^+]$.

A few words must be spent on the computational overhead of the proposed scaling method. Fig. 1 shows that it only requires two additional blocks with respect to traditional computed torque control schemes, more precisely the KBE and the ATLS. The KBE online returns $U^-$, $U^+$, $\dot{x}^-$, and $\dot{x}^+$ by means of (13) and (15) which are based on terms already evaluated inside the computed torque controller: The additional cost is negligible. In addition, the ATLS has a marginal cost. Its outputs $x$, $\dot{x}$, $\ddot{x}$ are evaluated at the sampling instants by executing the sequence of algebraic equations (18)–(30), and by updating the internal state of the two integrators (17): The computational burden is irrelevant if compared with that required by the computed torque controller. Summarizing, the advantages deriving from the scaling method can be gained at an almost zero cost.

## IV. Simulation Results

The effectiveness of the proposed filter is verified by simulating the control of a two-link Cartesian planar robot. The two orthogonal prismatic joints are characterized by the following dynamic parameters: $m_1 = 23.90$ kg, $m_2 = 3.88$ kg, $B_1 = 1.5e-3\ \mathrm{N \cdot m^{-1} \cdot s}$, and $B_2 = 2.8e-3\ \mathrm{N \cdot m^{-1} \cdot s}$, where $m_1$ and $m_2$ are the link masses while $B_1$ and $B_2$ represent the joint friction coefficients. The trajectory filter and the torque controller sampling times are equal to $1e-3$ s, while the solution of the continuous robot dynamics has been obtained with the Runge–Kutta method for the ODEs [20] by adopting a variable integration step superiorly constrained to $1e-4$ s.

The manipulator path is an ellipsoid represented by means of a curve in the joint space parametrized with respect to angle $x \in [0, 2\pi]$ rad, i.e., $\mathbf{q}_d(x) = \mathbf{\Gamma}(x) := [\Gamma_1(x) \quad \Gamma_2(x)]^{\mathrm{T}}$, where

$$\begin{cases} \Gamma_1(x) := 0.4\,(1 - \cos(x)) \\ \Gamma_2(x) := 0.8 \sin(x). \end{cases}$$

The trajectory is completely defined once a time-law $r(t)$ is assigned. Function $r(t)$ has been chosen such to be too demanding with respect to the robot velocity constraints

$$r(t) := \begin{cases} \frac{\pi}{12} t^2, & 0 \leq t \leq 2 \\ \frac{\pi}{3}(t-1), & 2 \leq t \leq 6 \\ \frac{\pi}{6}(t+4), & 6 \leq t \leq 8. \end{cases} \tag{32}$$

Corresponding $\dot{r}(t)$ is obtained straightforward.

Simulations are carried out by considering joint velocities and torques constrained between the following bounds: $|\dot{q}_i| \leq 0.65\ \mathrm{m \cdot s^{-1}}$ and $|\tau_i| \leq 15$ N, $i = 1, 2$. The feedback controller gains are equal to $\mathbf{K}_p = [200 \quad 200]^{\mathrm{T}}$ and $\mathbf{K}_v = [60 \quad 60]^{\mathrm{T}}$.

Fig. 5 shows what happens when the scaling filter is not used. In particular, Fig. 5(b) and (c) shows that both $\dot{q}_2$ and

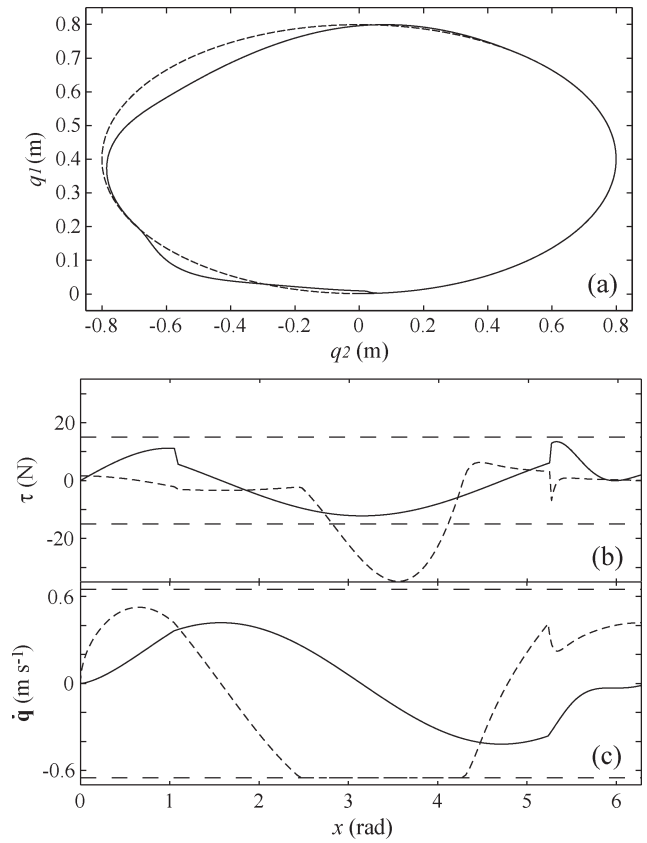


Fig. 5. Tracking performances of a standard torque controller due to joint velocity and torque saturations. (a) (Dashed line) Reference robot path compared with the (solid line) actual robot path. (b) Joint torques (solid line) $\tau_1$ and (dashed line) $\tau_2$. (c) Joint velocities (solid line) $\dot{q}_1$ and (dashed line) $\dot{q}_2$.

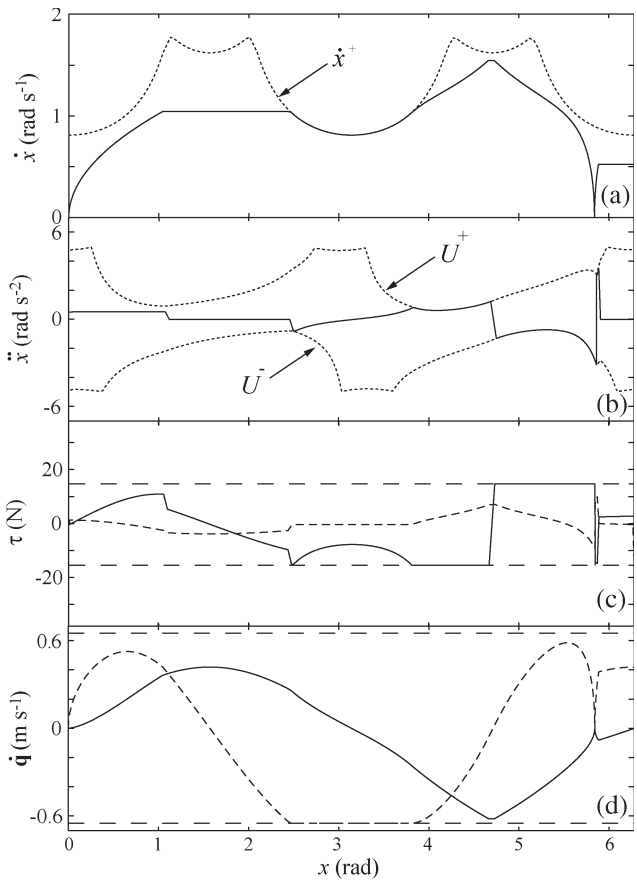


Fig. 6. Simulation results using the nonlinear trajectory scaling filter. (a) (Solid line) Longitudinal velocity $\dot{x}$ and (dotted line) online evaluated velocity bound $\dot{x}^+$. (b) (Solid line) Longitudinal acceleration $\ddot{x}$ and (dotted lines) online acceleration bounds $U^+$ and $U^-$. (c) Joint torques (solid line) $\tau_1$ and (dashed line) $\tau_2$. (d) Joint velocities (solid line) $\dot{q}_1$ and (dashed line) $\dot{q}_2$.

$\tau_2$ cross the assigned limits and, consequently, path tracking is lost, as shown in Fig. 5(a): The maximum error is equal to $e_{\max} = \max_{x\in[0,2\pi]}\{\|\mathbf{e}\|\} = 0.1619$ m.

By adopting the proposed filter, the situation neatly improves. Curves shown in Fig. 6(c) and (d) exactly coincide with those of Fig. 5(b) and (c) until the saturation on $q_2$ occurs, i.e., at the beginning the filter acts as a pass-through and $r = x$. Then, the filter starts scaling the reference signal to guarantee that assigned bounds are touched but not crossed. As a result, path-tracking error reduces of several order of magnitude with respect to the previous case: $e_{\max} = \max_{x\in[0,2\pi]}\{\|\mathbf{e}\|\} = 1.824e-4$ m.

One relevant filter feature is that it always tries to hook in minimum time its input reference trajectory compatibly with the assigned constraints. Analogous approaches (see, e.g., [13]) do not own this characteristic, since they preserve path tracking but do not take care of delays introduced into the original time-law. Fig. 7 compares reference signal $r$ with filter output $x$: The time instant when tracking is lost is evident, as well as the moment when tracking is newly gained. Since $r$ must be reached in minimum time, the filter assumes the bang-bang behavior shown in Fig. 6(c) and (d): Until $r$ is not newly gained, there is always one torque or velocity constraint which is active. The asymmetry of $U^+$, $U^-$ is clearly shown in Fig. 6(b) and justifies the use of the novel filter proposed in this paper. Fig. 6(a), together with Fig. 6(b), also prove that, as desired, both $\dot{x}(t)$ and $\ddot{x}(t)$ fulfill bounds $U^+$, $U^-$, $\dot{x}^+$, and $\dot{x}^-$.

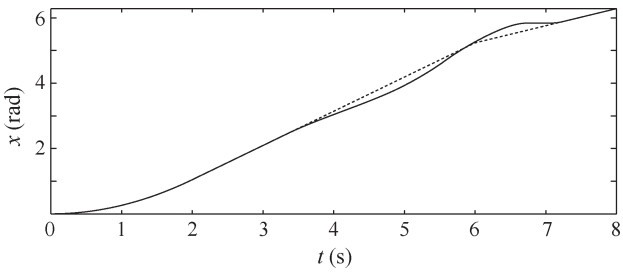


Fig. 7. Simulation results using the nonlinear trajectory scaling filter. Comparison between (solid line) $x(t)$ and (dashed line) $r(t)$.

The approach robustness has been tested by adding a white noise on $\mathbf{q}$ and $\dot{\mathbf{q}}$ in order to emulate a sensor disturbance. The simulation results shown in Fig. 8 refer to a very consistent additive noise: Its "peak to peak" amplitude is approximatively equal to $2e-2$ m for $\mathbf{q}$, and to $8e-2$ m $\cdot$ s$^{-1}$ for $\dot{\mathbf{q}}$. Clearly, such disturbances have a strong impact on the torque command due to the feedback action, as can be evinced by comparing Fig. 8(a) with Fig. 5(b), which report two situations where the filter is disabled. Evidently, path tracking is lost in both cases.

It is very interesting to observe what happens when the filter is enabled. As shown in Fig. 9, filter bounds $U^+$, $U^-$, $\dot{x}^+$, and $\dot{x}^-$ are affected by disturbances due to (13) and (14). However, the same figures highlight that filter outputs, i.e., $\ddot{x}$

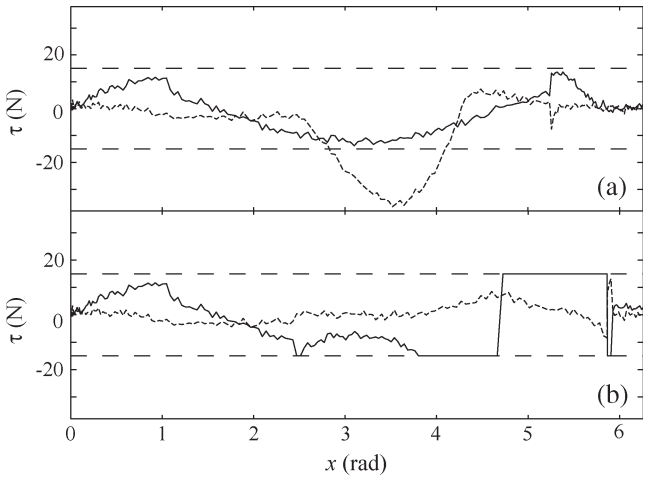


Fig. 8. Comparison of the torque profiles in presence of white additive noise on the robot measurement data. (a) When the trajectory filter not activated. (b) With the trajectory filter active.

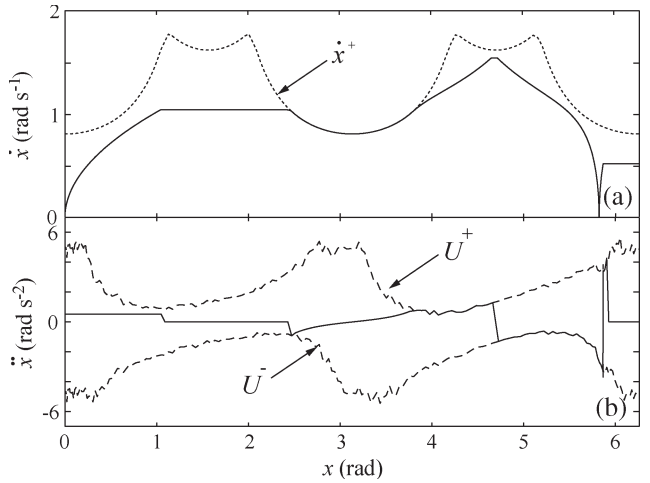


Fig. 9. Simulation results using the proposed trajectory filter in presence of disturbances. (a) (Solid line) Longitudinal velocity $\dot{x}$ and (dotted line) velocity bound $\dot{x}^+$. (b) (Solid line) Longitudinal acceleration $\ddot{x}$ and (dotted lines) acceleration bounds $U^+$ and $U^-$.

and $\dot{x}$, are only marginally influenced by noise: Practically, almost no further disturbances are introduced on $\boldsymbol{\tau}$, as proven by Fig. 8 where $\boldsymbol{\tau}$ signals obtained with or without the filter are compared. In general, measurement noise does not affect the behavior and the stability of the scaling filter as long as condition (31) is satisfied and $U^+ \geq U^-$. Owing to the noise, tracking errors evidently increase with respect to the ideal situation, but they are almost completely due to the feedback action. In particular, the worst case path-tracking error is equal to $e_{\max} = \max_{x\in[0,2\pi]}\{\|\mathbf{e}\|\} = 1.079e-2$ m.

## V. Conclusion

In this paper, a discrete-time filter has been proposed for the online automatic scaling of robot trajectories. By means of the new filter, it is possible to modify any input trajectory in order to fulfill given requirements on the maximum joint torques and velocities. The approach requires minor adaptations of a standard manipulator controller, since the desired result is obtained by simply inserting the new filter between a possible unfeasible reference signal and the controller itself. Simulation results demonstrate that path-tracking performances neatly improve and, simultaneously, the velocity reference is followed at best, compatibly with the manipulator constraints.

## Appendix I

In the following, the filter stability is proved. Simultaneously, some relevant properties are highlighted. The discussion reported hereafter will analyze a system evolution starting from a point $(z, \dot{z})$ located in the left plane of the $(z, \dot{z})$-space, i.e., such that $z \leq 0$. An analogous discussion holds when $z > 0$: The corresponding demonstrations are omitted for conciseness.

### A. General Properties

It is easy to verify that, when $z \leq 0$, (24) returns $[\alpha\ \beta] := [U^-\quad U^+]$, so that (25)–(30) simplify as follows:

$$z_k := -\frac{1}{TU^-}\left(\frac{y_k}{T} + \frac{\dot{y}_k}{2}\right) \tag{33}$$

$$\gamma_k := \begin{cases} z^+, & \text{if } z_k < z^+ \\ z_k, & \text{if } z^+ \leq z_k \leq 0 \end{cases} \tag{34}$$

$$m_k := \left\lfloor \frac{1 + \sqrt{1 - 8\,|\gamma_k|}}{2} \right\rfloor \tag{35}$$

$$\tilde{\dot{z}}_k := -\frac{\gamma_k}{m_k} + \frac{m_{k-1}}{2} \tag{36}$$

$$\dot{z}_k := \begin{cases} -\frac{\dot{y}_k}{TU^-}, & \text{if } \left(-\frac{\dot{y}_k}{TU^-} \geq \tilde{\dot{z}}_k\right) \\ \frac{\dot{y}_k}{TU^+} + \left(\frac{m_k - 1}{2} - \frac{\gamma_k}{m_k}\right) & \\ \quad \times \frac{U^+ + U^-}{U^+}, & \text{if } \left(-\frac{\dot{y}_k}{TU^-} < \tilde{\dot{z}}_k\right). \end{cases} \tag{37, 38}$$

The following two properties have general validity and will be used in the last part of the section to prove the system stability.

*Property 1:* For any point $(z_k, \dot{z}_k)$ lying inside the BL, the filter command signal is given by

$$u_k := -\frac{\dot{y}_k}{T} + \left(\frac{\gamma_k}{m_k} - \frac{m_k - 1}{2}\right) U^-. \tag{39}$$

*Proof:* Potentially, two different control laws could apply inside the BL due to (37) and (38). Suppose that

$$-\frac{\dot{y}_k}{TU^-} = \tilde{\dot{z}}_k. \tag{40}$$

According to (37), it immediately follows that $\dot{z}_k = -(\dot{y}_k/TU^-) = \tilde{\dot{z}}_k$, so that, due to (19), it is possible to conclude that, when (40) holds, the considered point is lying on the sliding surface. Practically, (37) and (38) are two alternative mappings that can be used depending on the position of the considered point with respect to the sliding surface.

Now, hypothesize that $-(\dot{y}_k/TU^-) < \tilde{\dot{z}}_k$, and, equivalently, that $\sigma_k < 0$. Since $(z_k, \dot{z}_k)$ is located inside the BL but below the sliding surface, it is possible to write

$$u_k = -U^+\sigma_k = -U^+(\dot{z}_k - \tilde{\dot{z}}_k).$$

Equation (39) is easily obtained after few algebraic manipulations by means of (36) and (38).

Similarly, when $-(\dot{y}_k/TU^-) \geq \dot{\tilde{z}}_k$ or, equivalently, when $\sigma_k \geq 0$, the control law becomes

$$u_k = U^-\sigma_k = U^-(\dot{z}_k - \dot{\tilde{z}}_k).$$

Again, (39) is immediately obtained by considering (36) and (37). ■

*Property 2:* Given any point $(z_k, \dot{z}_k)$ lying within the BL, controller $C$ generates a new point such that

$$z_{k+1} = z_k + \dot{\tilde{z}}_k. \tag{41}$$

Moreover, the following condition holds:

$$\operatorname{sgn}(z_k) = \operatorname{sgn}(z_{k+1}). \tag{42}$$

*Proof:* Being $\dot{r}_k$ piecewise constant, and assuming that

$$\dot{r}_k = \frac{r_{k+1} - r_k}{T}$$

the discrete-time evolution (17) is converted into an equivalent one in the $(y, \dot{y})$-plane defined as follows:

$$\begin{bmatrix} y_{k+1} \\ \dot{y}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_k \\ \dot{y}_k \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} u_k. \tag{43}$$

The evolution of system (43), obtained by applying command signal (39), can be converted, by means of (33), into the following updating law for $z$:

$$z_{k+1} = z_k + \frac{m_k(m_k - 1) - 2\gamma_k}{2m_k}. \tag{44}$$

From (35), it descends that, when $z_k \leq 0$, the following inequality is verified:

$$z_k \leq \gamma_k \leq -m_k + 1 \tag{45}$$

so that (44) implies

$$z_{k+1} \leq \frac{(m_k - 1)(2 - m_k)}{2m_k}. \tag{46}$$

Due to definition (35), we have that $m_k \in \mathbb{N} \setminus 0$. As a consequence, it is possible to deduce form (46) that $z_{k+1} \leq 0$, thus (42) holds.

Equation (41) immediately descends from (44) by considering (36). ■

Property 2 practically asserts that any point within the BL cannot abandon the left plane $z \leq 0$. Properties 1 and 2 generically apply to any point within the BL.

### B. Behavior Inside Region $R1$

*Proposition 1:* Given any starting point $(z, \dot{z})$ lying inside region $R1$, the BL which surrounds sliding surface $\dot{\tilde{z}}$ is reached in minimum time and in a finite number of steps.

*Proof:* The proof is straightforward since from (18) and (19), it follows that above the sliding surface, we have $u_k = U^-$, while below $u_k = U^+$. Due to (43), it is possible to conclude that $\dot{y}$ monotonically decreases above $\dot{\tilde{z}}$ while it monotonically increases below $\dot{\tilde{z}}$: owing to the shape of the sliding surface region $R2$ or, alternatively, region $R3$ are certainly reached after a finite number of steps (see also Fig. 3). ■

### C. Behavior Inside Region $R2$

*Proposition 2:* Given any point $(z_k, \dot{z}_k)$ lying within the BL and with $z_k < z^+$, controller $C$ generates a command signal such that the system evolves as follows:

$$\begin{bmatrix} z_{k+1} \\ \dot{z}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_k \\ \dot{z}_k \end{bmatrix} + \begin{bmatrix} \dot{z}^+ \\ \dot{z}^+ \end{bmatrix}. \tag{47}$$

*Proof:* Since $z_k < z^+$, due to (34), we can write $\gamma_k = z^+$, so that (35) and (36) become constant and can be rewritten as follows:

$$m^+ := \left\lfloor \frac{1 + \sqrt{1 - 8z^+}}{2} \right\rfloor \tag{48}$$

$$\dot{\tilde{z}}_k := -\frac{z^+}{m^+} + \frac{m^+ - 1}{2}. \tag{49}$$

Due to (43) and (49), it is possible to write

$$-\frac{\dot{y}_{k+1}}{TU^-} = \frac{m^+ - 1}{2} - \frac{z^+}{m^+} = \dot{\tilde{z}}_k. \tag{50}$$

It was early anticipated that the sliding surface has been designed such that $\dot{\tilde{z}} = \dot{z}^+$ when $z < z^+$, so that from (50), it descends

$$-\frac{\dot{y}_{k+1}}{TU^-} = \dot{z}^+. \tag{51}$$

Owing to the shape of the sliding surface (see also Fig. 4), it is possible to assert that, in any case, $\dot{\tilde{z}}_{k+1} \leq \dot{z}^+$. Thus, from (51), it follows:

$$-\frac{\dot{y}_{k+1}}{TU^-} \geq \dot{\tilde{z}}_{k+1}. \tag{52}$$

Equation (52) indicates that $\dot{z}_{k+1}$ must be evaluated according to (37) and, consequently, bearing in mind (51), we finally obtain, as desired

$$\dot{z}_{k+1} = -\frac{\dot{y}_{k+1}}{TU^-} = \dot{z}^+. \tag{53}$$

The expression for $z_{k+1}$ is obtained straightforward by means of (41) and taking into account that $\dot{\tilde{z}}_k = \dot{z}^+$. ■

*Remark 1:* Equation (47), implies that when the system state enters into the BL and $z < z^+$, $\dot{z}$ is forced to the sliding surface $\dot{z}^+$ with a single step and there it remains. Moreover, being $\dot{z}^+ \geq 0$, coordinate $z$ increases, i.e., the state slides to the right. Necessarily, after a finite number of steps, it reaches region $R3$.

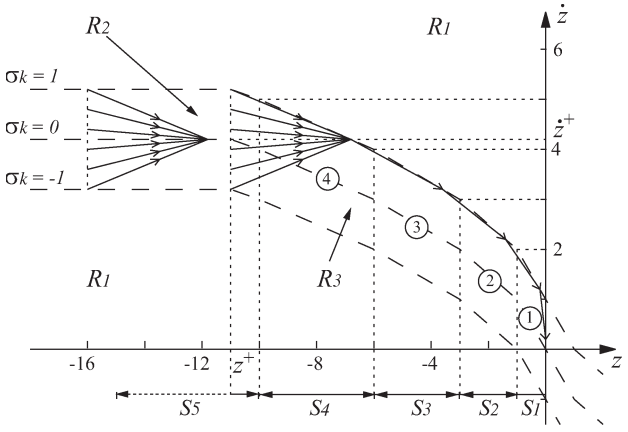


Fig. 10. Phase-plane in the $(z, \dot{z})$-plane: details in the vicinity of the origin. Circled numbers indicate the corresponding value of $m$.

### D. Behavior Inside Region $R3$

It is clear that, after a finite number of steps, the system reaches the BL of the region $z^+ \leq z \leq 0$ directly from $R1$ or, alternatively, from $R2$. The following discussion is devoted to demonstrate that the system state cannot abandon the BL, and it must move toward the origin of the $(z, \dot{z})$-space.

Due to (34), we can assume $\gamma_k = z_k$ when $z^+ \leq z \leq 0$.

*Property 3:* Assume that at step $k$, system state $(z_k, \dot{z}_k)$ is lying within the BL, with $z^+ \leq z_k \leq 0$, and it is characterized by $m_k$. The new state $(z_{k+1}, \dot{z}_{k+1})$ generated by controller $C$ satisfies the following equality:

$$m_{k+1} = m_k - 1.$$

*Proof:* It is possible to rearrange (44) as follows:

$$z_{k+1} = (m_k - 1)\left(\frac{z_k}{m_k} + \frac{1}{2}\right). \tag{54}$$

It is worth noting that (35) induces a partition along the $z$-axis. In particular, owing to (35), associated with any $m \in \mathbb{N} \setminus 0$, there is an interval $S_m$ in $z$ defined as follows (see also Fig. 10):

$$S_m := \left\{ z : -\frac{(m+1)m}{2} < z \leq -\frac{m(m-1)}{2} \right\}. \tag{55}$$

Now, hypothesize that current $z_k$ is contained in $S_{m_k}$, i.e., $z_k \in S_{m_k}$. By taking into account (55) and (54), and by defining

$$m_{k+1} := m_k - 1 \tag{56}$$

the following equation holds:

$$-\frac{(m_{k+1}+1)m_{k+1}}{2} < z_{k+1} \leq -\frac{m_{k+1}(m_{k+1}-1)}{2}. \tag{57}$$

By comparing (57) with (55), it is immediately possible to conclude that $z_{k+1} \in S_{m_{k+1}}$, where $m_{k+1}$ is defined by (56). ■

*Property 4:* Given any point $(z_k, \dot{z}_k)$ lying within the BL, with $z^+ \leq z_k \leq 0$, the new point $(z_{k+1}, \dot{z}_{k+1})$ generated by controller $C$ is located on the upper frontier of the BL.

*Proof:* Due to *Properties 2* and *3*, it is possible to assert that $z_k < z_{k+1} \leq 0$, so that the position of the sliding surface corresponding to $z_{k+1}$ can be certainly written, according to (36) and due to (54), as follows:

$$\dot{\tilde{z}}_{k+1} = -\frac{z_k}{m_k} + \frac{m_k - 3}{2}. \tag{58}$$

Bearing in mind (39) and (43), it is possible to assert that

$$-\frac{\dot{y}_{k+1}}{TU^-} = -\frac{z_k}{m_k} + \frac{m_k - 1}{2}. \tag{59}$$

By comparing (58) with (59), it is possible to conclude that $\dot{z}_{k+1}$ must be evaluated by means of (37). Consequently

$$\dot{z}_{k+1} = -\frac{z_k}{m_k} + \frac{m_k - 1}{2}. \tag{60}$$

The position of the new point with respect to the sliding surface is

$$\sigma_{k+1} = \dot{z}_{k+1} - \dot{\tilde{z}}_{k+1} = 1 \tag{61}$$

i.e., it exactly lies on the upper frontier of the BL. ■

Previous properties are used in the following to prove the stability of the filter controller.

*Proposition 3:* Given any starting point $(z_k, \dot{z}_k)$ lying within the BL, with $z^+ \leq z_k \leq 0$, controller $C$ forces the system trajectory toward the origin of the $(z, \dot{z})$-plane in minimum time and with a deadbeat dynamics.

*Proof:* According to (54) and (60), the system evolution only depends on the current $z_k$ and $m_k$. Owing to Property 3, $m$ decreases at each step until it reaches the value $m = 1$. When it happens, owing to (54) and (60), we have $z_{k+1} = 0$ and $\dot{z}_{k+1} = -z_k$. It is easy to verify by means of (35) that $m_{k+1}$ will be still equal to one, so that at the next step (54) and (60) return $z_{k+2} = 0$ and $\dot{z}_{k+2} = 0$: the origin of the $(z, \dot{z})$-plane is reached with a deadbeat behavior. It is important to note that, due to Property 4, once the system reaches the BL, it is forced in a single step toward the frontier of the BL itself. The same Property 4 makes it possible to assert that during the subsequent steps the system does not abandon such frontier, so that the evolution toward the origin is obtained by applying the maximum control command $u_k = U^-$, i.e., in minimum time. ■

## REFERENCES

[1] C.-S. Lin, P.-R. Chang, and J. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *IEEE Trans. Autom. Control*, vol. AC-28, no. 12, pp. 1066–1074, Dec. 1983.

[2] A. De Luca, L. Lanari, and G. Oriolo, "A sensitivity approach to optimal spline robot trajectories," *Automatica*, vol. 27, no. 3, pp. 535–539, May 1991.

[3] C. Guarino Lo Bianco and A. Piazzi, "Minimum-time trajectory planning of mechanical manipulators under dynamic constraints," *Int. J. Control*, vol. 75, no. 13, pp. 967–980, 2002.

[4] S. Kim and F. Park, "Fast robot motion generation using principal components: Framework and algorithms," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2506–2516, Jun. 2008.

[5] 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.
[6] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Autom. Control*, vol. AC-30, no. 6, pp. 531–541, Jun. 1985.
[7] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 554–561, 1985.
[8] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," *Trans. ASME, J. Dyn. Syst. Meas. Control*, vol. 106, no. 1, pp. 102–106, 1984.
[9] S. Moon and S. Ahmad, "Time scaling of cooperative multirobot trajectories," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 4, pp. 900–908, Jul./Aug. 1991.
[10] A. De Luca and R. Farina, "Dynamic scaling of trajectories for robots with elastic joints," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, May 2002, pp. 2436–2442.
[11] T. Sugie, K. Fujimoto, and Y. Kito, "Obstacle avoidance of manipulators with rate constraint," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 168–174, Feb. 2003.
[12] G. Antonelli, S. Chiaverini, and G. Fusco, "A new online algorithm for inverse kinematics of robot manipulators ensuring path tracking capabilities," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 162–167, Feb. 2003.
[13] O. Dahl and L. Nielsen, "Torque-limited path following by on-line trajectory time scaling," *IEEE Trans. Robot. Autom.*, vol. 6, no. 5, pp. 554–561, Oct. 1990.
[14] O. Dahl, "Path-constrained robot control with limited torques-experimental evaluation," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 658–669, Oct. 1994.
[15] 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, Feb. 1994.
[16] R. Zanasi, C. Guarino Lo Bianco, and A. Tonielli, "Nonlinear filters for the generation of smooth trajectories," *Automatica*, vol. 36, no. 3, pp. 439–448, Mar. 2000.
[17] C. Guarino Lo Bianco and R. Zanasi, "Smooth profile generation for a tile printing machine," *IEEE Trans. Ind. Electron.*, vol. 50, no. 3, pp. 471–477, Jun. 2003.
[18] X. Wei, J. Wang, and Z. Yang, "Robust smooth-trajectory control of nonlinear servo systems based on neural networks," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 208–217, Feb. 2007.
[19] V. Utkin, "Variable structure systems with sliding modes," *IEEE Trans. Autom. Control*, vol. AC-22, no. 2, pp. 212–222, Apr. 1977.
[20] J. R. Dormand and P. J. Prince, "A family of embedded Runge–Kutta formulae," *J. Comput. Appl. Math.*, vol. 6, pp. 19–26, 1980.

**Oscar Gerelli** received the Laurea degree (with honors) in computer engineering and the Ph.D. degree in control system engineering from the University of Parma, Parma, Italy, in 2005 and 2009, respectively.

In 2007, he spent two quarters with the University of Twente, Enschede, The Netherlands, as a Visiting Student. He is currently with the Dipartimento di Ingegneria dell'Informazione, University of Parma. He is involved in research concerning mobile and industrial robotics. In particular, he is mainly interested in topics concerning the smooth and optimal profile generation for motion control and in optimal control algorithms for linear and nonlinear systems.

**Corrado Guarino Lo Bianco** (M'05) received the Laurea degree (with honors) in electronic engineering and the Ph.D. degree in control system engineering from the University of Bologna, Bologna, Italy, in 1989 and 1994, respectively.

Currently, he is with the Dipartimento di Ingegneria dell'Informazione, University of Parma, Parma, Italy, as an Aggregate Professor in industrial robotics. He is involved in research concerning mobile and industrial robotics. In particular, he is mainly interested in topics concerning the smooth and optimal trajectory generation, robot kinematics and dynamics, and robot control.