The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Trajectory scaling for a manipulator inverse dynamics control subject to generalized force derivative constraints

Corrado GUARINO LO BIANCO* and Oscar GERELLI

Abstract— **The behaviour of robotic manipulators is affected by the actuators dynamic limits. When such limits are not explicitly considered, manipulators performances rapidly decrease. In this paper, dynamic saturations are handled by means of a real-time technique based on a trajectory scaling method: whenever saturations occur, trajectories are automatically scaled by means of a dynamic filter in order to preserve an accurate path tracking. Commonly known scaling algorithms only consider the existence of torque saturations. In this paper, the strategy is enriched by also accounting for torque derivatives constraints. The solution proposed is suited to be used in conjunction with standard inverse dynamics controllers. The adopted methodology explicitly requires the realtime evaluation of the derivative of the manipulator inertia matrix. To this purpose, a novel efficient procedure is proposed.**

I. Introduction

Efficiency is an important issue in robotics. The time required to accomplish a given task has an evident impact on the productivity of any industrial process. For this reason, users ask for very demanding robot trajectories, which possibly conflict with the manipulators limits. In order to avoid potential problems deriving from the manipulator capabilities, kinematics limits are normally considered while planning trajectories, e.g., by bounding velocities, accelerations, and, when possible, jerks. Conversely, dynamic constraints are rarely taken into account owing to their difficult evaluation, so that system actuators could easily saturate, thus causing tracking losses. Typically, dynamic bounds are considered during trajectory planning: off-line optimizations are used to minimize the manipulator traveling times by introducing explicit constraints on the sole joint forces or torques, i.e., on the Generalized Forces (GF). Initial results for nonredundant manipulators have been proposed in [1], where a scaling factor is introduced to guarantee the feasibility of a planned trajectory. This approach has been subsequently extended to the case of robots in cooperative tasks, [2], and for manipulators with elastic joints, [3]. It is worth noting that off-line methods require an a priori knowledge of the exact robot dynamics, which in many practical situations can only be roughly estimated. For this reason, due to model uncertainties and/or external disturbances, trajectory tracking can still be lost. To obviate this drawback, scaling methods have been proposed in the past in order to online reshape planned trajectories and fulfill kinematic and/or dynamic limits. Initial solutions have been proposed in [4]–[6] for robot manipulators subject to torque limits, while in [7] a robust online extension of [1] has been described. Moreover, in [8], online adaptation schemes of the manipulator torque controller are introduced in order to track a given path with a prescribed tolerance. The strategy known as path-velocity decomposition [9] has been used in all the previously mentioned works. It is based on a two step approach: first, a desired path is planned and, then, the time law along the path is defined. The trajectory scaling method online modifies this second planning depending on the active dynamic constraints. The purpose is to preserve an accurate path tracking by dropping, if required due to dynamic saturations, the planned time law.

Although these methods improve tracking accuracy, they still do not consider existing physical constraints on the Generalized Force Derivatives (GFD). Also such constraints affect tracking performances: commonly used minimum-time solutions require rapidly changing GFs, which cannot be provided due to the actuators dynamic characteristics.

For this reason, new offline approaches have been recently proposed in order to plan optimal trajectories subject to constraints both on GFs and on GFDs, [10]–[12]. Evidently, these approaches require new methods for the online trajectory scaling. To this purpose, the original problem proposed by Dahl and Nielsen [4] has been recently revised [13] in order to consider the generation of trajectories subject to constraints both on GFs and on GFDs. More precisely, in [13] a scaling method based on a nonlinear filter was proposed. It was suited to be used with Feedforward Controllers with Position and Velocity feedback (FCPV).

In this paper, the topic is newly reconsidered in order to improve and extend the previous obtained results. First, a new scaling filter is proposed, which shows dead-beat convergence properties and, secondly, the control scheme is adapted to be used with an Inverse Dynamics Controller (IDC). The same methodology is assumed: dynamic constraints are online converted into equivalent kinematic constraints and an appropriate filter is then used to autonomously scale the trajectory in order to fulfill such constraints. The computational burden of the new control scheme is higher if compared with that of the controller proposed in [13], since it explicitly requires the online evaluation of more terms and, in particular, of the derivative of the manipulator inertia matrix. To this purpose the paper proposes an algorithm, which could also be used in other contexts, for the efficient evaluation of such matrix.

The paper is organized as follows. In §II the trajectory

* Corresponding author.

This research is partially supported by AER-TECH Lab of Regione Emilia-Romagna, Italy.

The authors are with the Dip. di Ing. dell'Informazione, University of Parma, I-43100 Parma, Italy, Tel. +39 0521 905752, Fax. +39 0521 905723, Email: {guarino,gerelli}@ce.unipr.it

scaling problem is formulated. The dynamic expressions which characterize the manipulator high order dynamics are proposed in §III. The control strategy and the method used for the online conversion of the dynamic limits into kinematic limits are described in §IV: they require the online evaluation of the derivative of the manipulator inertia matrix which is obtained by means of an efficient algorithm reported in the Appendix. The nonlinear dynamic filter for the online trajectories scaling is proposed in §V, while the overall control strategy is tested in §VI by means of an example case. §VII concludes the paper.

II. The trajectory scaling problem

According to the path-velocity decomposition, let us define the assigned path $\mathbf{q}_d$, expressed in the joint space, as follows

$$\begin{array}{rcl} \mathbf{f}:[0,u_f] & \rightarrow & \mathbb{R}^n \\ u & \rightarrow & \mathbf{q}_d := \mathbf{f}(u) \end{array} \tag{1}$$

where u is the scalar used to parametrize the curve, u_f is its final value, and n is the number of independent joints. A monotonically increasing time law is then needed in order to describe the movement along the curve

$$\begin{array}{rcl} u:[0,t_f] & \rightarrow & [0,u_f] \\ t & \rightarrow & u_d := u(t) \end{array} \tag{2}$$

where t_f is the total traveling time. Necessarily, $\dot{u}_d(t) > 0, \forall t \in [0,t_f]$.

Bearing in mind (1) and (2), the following expressions hold due to the chain differentiation rule

$$\dot{\mathbf{q}}_d = \mathbf{f}(u)^{'}\dot{u}\,, \tag{3}$$

$$\ddot{\mathbf{q}}_d = \mathbf{f}(u)^{''}\dot{u}^2 + \mathbf{f}(u)^{'}\ddot{u}\,, \tag{4}$$

$$\dddot{\mathbf{q}}_d = \mathbf{f}(u)^{'''}\dot{u}^3 + 3\mathbf{f}(u)^{''}\dot{u}\ddot{u} + \mathbf{f}(u)^{'}\dddot{u}\,. \tag{5}$$

where superscript $'$ indicates a differentiation with respect to u, e.g., $\mathbf{f}(u)^{'} = \frac{d\mathbf{f}(u)}{du}$, while, as usual, dots indicate time derivatives, e.g., $\dot{u}(t) = \frac{du(t)}{dt}$.

The trajectory scaling problem can be summarized as follows: given a trajectory, assigned according to the path-velocity decomposition, automatically scale its velocity profile such that path tracking is not lost even if a dynamic saturation occurs. Practically this implies that trajectory tracking is occasionally lost if dynamic limits are reached, but, in any case, path tracking is maintained. Usually [4], trajectory scaling only considers GF constraints, i.e., given for each joint $k = 1,2,\ldots,n$ an upper bound $\overline{\tau}_k$ and a lower bound $\underline{\tau}_k$, the trajectory is online modified such that the controlled GF, i.e., τ_k, is bounded between assigned limits

$$\underline{\tau}_k \le \tau_k \le \overline{\tau}_k\,. \tag{6}$$

This paper continues the discussion started in [13]. More precisely, the problem is deepened by also considering bounds on the GFD $\dot{\tau}_k, k = 1,2,\ldots,n$

$$\underline{\dot{\tau}}_k \le \dot{\tau}_k \le \overline{\dot{\tau}}_k\,, \tag{7}$$

where $\underline{\dot{\tau}}_k$ and $\overline{\dot{\tau}}_k$ represent the lower and the upper bounds on the k-th joint GFD. Differently from [13], the time scaling methodology is used in association with an inverse dynamics controller.

III. Manipulators high order dynamics

The time scaling procedure proposed in the next section is based on the exact knowledge of the dynamic stresses acting on each joint. To this purpose, it is essential to represent GFs and GFDs by means of closed form equations. Generalized forces $\boldsymbol{\tau}$ can be evaluated by means of the classic inverse dynamics equation,

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\mathbf{q},\dot{\mathbf{q}})\,. \tag{8}$$

As usual, $\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}} \in \mathbb{R}^n$ indicate the joint variables and their first and second time derivatives, $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{nxn}$ is the symmetric positive definite inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{nxn}$ is the matrix of centripetal and Coriolis terms, $\mathbf{g} \in \mathbb{R}^n$ is the vector of the gravity forces, and $\mathbf{f}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^n$ is the vector of the friction forces.

Equation (8) can also be posed in scalar form by explicitly writing each single term of $\boldsymbol{\tau} := [\tau_1\ \tau_2\ \cdots, \tau_n]^T$

$$\tau_k = \sum_{j=1}^{n} h_{kj}(\mathbf{q})\,\ddot{q}_j + \sum_{j=1}^{n} c_{kj}(\mathbf{q})\,\dot{q}_j + g_k(\mathbf{q}) + f_k(\mathbf{q},\dot{\mathbf{q}})\,, \tag{9}$$

where

$$c_{kj}(\mathbf{q},\dot{\mathbf{q}}) := \sum_{i=1}^{n} c_{ijk}(\mathbf{q})\,\dot{q}_i\,, \tag{10}$$

being c_{ijk} the so-called Christoffel symbols of the first kind.

Closed form expressions are also needed for the evaluation of the manipulator GFDs: they will be used to check constraint (7). By differentiating (9) with respect to time, and taking also into account (10), for each joint it is possible to write ($k = 1,2,\ldots,n$)

$$\begin{aligned} \dot{\tau}_k = {} & \sum_{j=1}^{n} \dot{h}_{kj}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{q}_j + \sum_{j=1}^{n} h_{kj}(\mathbf{q})\,\dddot{q}_j + \\ & \sum_{j=1}^{n} d_{kj}(\mathbf{q},\dot{\mathbf{q}})\,\dot{q}_j + 2\sum_{j=1}^{n} c_{kj}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{q}_j + \\ & \sum_{j=1}^{n} b_{kj}(\mathbf{q},\dot{\mathbf{q}})\,\dot{q}_j + \sum_{j=1}^{n} e_{kj}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{q}_j\,. \end{aligned} \tag{11}$$

where

$$\dot{h}_{kj}(\mathbf{q},\dot{\mathbf{q}}) := \sum_{i=1}^{n} \frac{\partial h_{kj}(\mathbf{q})}{\partial q_i}\,\dot{q}_i\,, \tag{12}$$

$$d_{kj}(\mathbf{q},\dot{\mathbf{q}}) := \sum_{i=1}^{n}\sum_{l=1}^{n} \frac{\partial c_{ijk}(\mathbf{q})}{\partial q_l}\,\dot{q}_l\,\dot{q}_i\,, \tag{13}$$

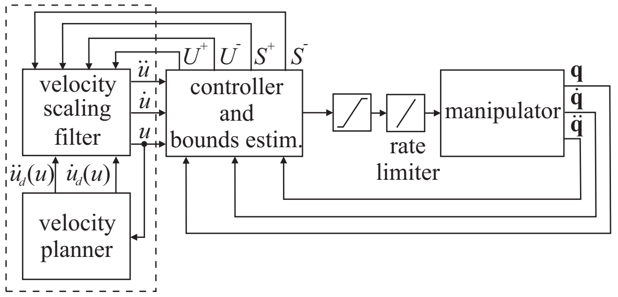
$$b_{kj}(\mathbf{q},\dot{\mathbf{q}}) := \frac{\partial g_k(\mathbf{q})}{\partial q_j} + \frac{\partial f_k(\mathbf{q},\dot{\mathbf{q}})}{\partial q_j}\,, \tag{14}$$

$$e_{kj}(\mathbf{q},\dot{\mathbf{q}}) := \frac{\partial f_k(\mathbf{q},\dot{\mathbf{q}})}{\partial \dot{q}_j}\,, \tag{15}$$

Equation (11) can be posed into the following compact form

$$\begin{aligned} \dot{\boldsymbol{\tau}} = {} & \dot{\mathbf{H}}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q})\,\dddot{\mathbf{q}} + \mathbf{D}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + 2\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{\mathbf{q}} + \\ & \mathbf{B}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{E}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{\mathbf{q}}\,. \end{aligned} \tag{16}$$

The first two terms represent the components of the GFD which are due to the system inertia. In the same way, the second two terms are due to the Coriolis and centripetal components, while the last two terms refer to the gravity and the friction effects.



velocity scaling filter
controller and bounds estim.
manipulator
rate limiter
velocity planner
U^+ U^- S^+ S^-
$\ddot{u}$ $\dot{u}$ u
$\ddot{u}_d(u)$ $\dot{u}_d(u)$
$\mathbf{q}$ $\dot{\mathbf{q}}$ $\ddot{\mathbf{q}}$

Fig. 1. The manipulator control scheme. The dashed box surrounds the automatic trajectory scaler.

IV. THE CONTROL STRATEGY

As anticipated in the introduction, the manipulator is driven by an inverse dynamics controller. The overall control scheme is shown in Fig. 1. As known [14], an inverse dynamics controller is described by the following dynamic equation

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{H}(\mathbf{q})\,\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\mathbf{q},\dot{\mathbf{q}}) \\ &\quad + \mathbf{k}_p^T\,\mathbf{e} + \mathbf{k}_v^T\,\dot{\mathbf{e}}\,, \qquad (17)\end{aligned}$$

where $\mathbf{e} := \mathbf{q}_d - \mathbf{q}$ and $\dot{\mathbf{e}} := \dot{\mathbf{q}}_d - \dot{\mathbf{q}}$ respectively represent the tracking errors and their first derivatives; $\mathbf{k}_p \in \mathbb{R}^n$ and $\mathbf{k}_v \in \mathbb{R}^n$ are the gain vectors of the feedback action. Convergence properties of controller (17) are well investigated [14], so that they will be not discussed in the following.

The controller equation can be reparametrized by means of (1)–(4) leading to

$$\boldsymbol{\tau}(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}}) = \mathbf{b}_1(u;\mathbf{q})\ddot{u} + \mathbf{b}_2(u,\dot{u};\mathbf{q},\dot{\mathbf{q}})\,, \qquad (18)$$

where

$$\begin{aligned}\mathbf{b}_1(u;\mathbf{q}) &:= \mathbf{H}(\mathbf{q})\,\mathbf{f}(u)'\,, \qquad (19)\\ \mathbf{b}_2(u,\dot{u};\mathbf{q},\dot{\mathbf{q}}) &:= \mathbf{H}(\mathbf{q})\,\mathbf{f}(u)''\,\dot{u}^2 + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\mathbf{q},\dot{\mathbf{q}}) \\ &\quad + \mathbf{k}_p^T\,\mathbf{e} + \mathbf{k}_v^T\,\dot{\mathbf{e}}\,. \qquad (20)\end{aligned}$$

Terms $\mathbf{b}_1(u;\mathbf{q}) = [b_{1,1}, b_{1,2}, \ldots, b_{1,n}]^T$ and $\mathbf{b}_2(u,\dot{u};\mathbf{q},\dot{\mathbf{q}}) = [b_{2,1}, b_{2,2}, \ldots, b_{2,n}]^T$ need to be online evaluated. To this purpose, efficient strategies are required to calculate the inertia matrix $\mathbf{H}(\mathbf{q})$ as well as the Coriolis, the centripetal, the friction, and the gravity terms, i.e., $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\mathbf{q},\dot{\mathbf{q}})$. Solutions to this problem are proposed in the literature [15], mainly based on an appropriate use of the recursive Newton-Euler algorithm [16]. It is evident from (18) that GFs depend on $u,\dot{u},\ddot{u}$. It is thus necessary to supply a proper reference signal u in order to fulfill constraints (6). This is the target of the two blocks shown inside the dashed rectangle of Fig. 1. The first block provides a velocity reference signal $\dot{u}_d$ and an acceleration reference signal $\ddot{u}_d$, which could be possibly unfeasible with respect to the dynamic constraints. Such signals are subsequently scaled by the velocity scaling filter in order to achieve a feasible signal $\dot{u}$.

Details on the filter will be given in the next sections. It is important to highlight that it is designed such to automatically scale the velocity profile in order to fulfill proper constraints. To this purpose, inequalities (6) and (7) must be preliminary converted into equivalent bounds on $\ddot{u}$ and $\dddot{u}$. Due to (18), for each joint k it is possible to write $\tau_k = b_{1,k}\ddot{u} + b_{2,k}$: constraints (6) are satisfied by imposing

$$\underline{\tau}_k \le b_{1,k}\ddot{u} + b_{2,k} \le \overline{\tau}_k, \quad k = 1,2,\ldots,n\,. \qquad (21)$$

Necessarily, this implies that the feasibility of $\ddot{u}$ is guaranteed if $\ddot{u} \in \bigcap_{k=1}^{n} [\beta_k\,,\alpha_k]$, with

$$\alpha_k = \begin{cases} \frac{\overline{\tau}_k - b_{2,k}}{b_{1,k}}, \text{if } b_{1,k} > 0 \\ \frac{\underline{\tau}_k - b_{2,k}}{b_{1,k}}, \text{if } b_{1,k} < 0 \\ \infty, \quad \text{if } b_{1,k} = 0 \end{cases} \text{and } \beta_k = \begin{cases} \frac{\underline{\tau}_k - b_{2,k}}{b_{1,k}}, \text{if } b_{1,k} > 0 \\ \frac{\overline{\tau}_k - b_{2,k}}{b_{1,k}}, \text{if } b_{1,k} < 0 \\ -\infty, \quad \text{if } b_{1,k} = 0 \end{cases} \qquad (22)$$

or, equivalently, if $\ddot{u} \in [S^-\,, S^+]$ where

$$S^- := \max_{k=1,\ldots,n} \{\beta_k\}\,, \quad S^+ := \min_{k=1,\ldots,n} \{\alpha_k\}\,. \qquad (23)$$

Depending on the manipulator status of motion it could happen that $S^- > S^+$: in this case there does not exist any feasible interval for $\ddot{u}$ and the control will be lost with certainty.

A similar problem must be solved in order to guarantee that also (7) are satisfied. To this aim, constraints on the maximum GFDs must be converted into an equivalent limit on the maximum admissible jerk along the path. Bearing in mind (16), it follows that the derivative of (17) can be written as

$$\begin{aligned}\dot{\boldsymbol{\tau}} &= \dot{\mathbf{H}}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{\mathbf{q}}_d + \mathbf{H}(\mathbf{q})\,\dddot{\mathbf{q}}_d + \mathbf{D}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + 2\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{\mathbf{q}} \\ &\quad + \mathbf{B}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{E}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{\mathbf{q}} + \mathbf{k}_p^T\,\dot{\mathbf{e}} + \mathbf{k}_v^T\,\ddot{\mathbf{e}}\,. \qquad (24)\end{aligned}$$

Also this expression can be parametrized in function of the curvilinear coordinate u by means of (1)–(5), so that, after simple manipulations, it is possible to write

$$\dot{\boldsymbol{\tau}} = \mathbf{b}_1(u;\mathbf{q})\dddot{u} + \widetilde{\mathbf{b}}_2(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\,, \qquad (25)$$

where

$$\begin{aligned}\mathbf{b}_1(u;\mathbf{q}) &:= \mathbf{H}(\mathbf{q})\,\mathbf{f}(u)'\,, \qquad (26)\\ \widetilde{\mathbf{b}}_2(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) &:= \dot{\mathbf{H}}(\mathbf{q},\dot{\mathbf{q}})\,[\mathbf{f}(u)''\,\dot{u}^2 + \mathbf{f}(u)'\,\ddot{u}] \\ &\quad + \mathbf{H}(\mathbf{q})\,[\mathbf{f}(u)'''\,\dot{u}^3 + 3\mathbf{f}(u)''\,\dot{u}\ddot{u}] \\ &\quad + \mathbf{D}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + 2\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} \\ &\quad + \mathbf{E}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{\mathbf{q}} + \mathbf{k}_p^T\,\dot{\mathbf{e}} + \mathbf{k}_v^T\,\ddot{\mathbf{e}}\,. \qquad (27)\end{aligned}$$

Term $\mathbf{b}_1(u;\mathbf{q})$ is the same computed in (19). Therefore only $\widetilde{\mathbf{b}}_2(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ needs to be evaluated. From Equation (27) it follows that the knowledge of the inertia matrix derivative, i.e., $\dot{\mathbf{H}}(\mathbf{q},\dot{\mathbf{q}})$, is required. This give us the opportunity to propose a new method for the online evaluation of $\dot{\mathbf{H}}(\mathbf{q},\dot{\mathbf{q}})$. The new algorithm is described in the appendix so that, from now on, both $\widetilde{\mathbf{b}}_2(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ and $\dot{\mathbf{H}}(\mathbf{q},\dot{\mathbf{q}})$ are assumed to be known.

In order to satisfy the requirements on the GFDs, it is necessary to guarantee that

$$\underline{\dot{\tau}}_k \le b_{1,k}\dddot{u} + \widetilde{b}_{2,k} \le \overline{\dot{\tau}}_k, \quad k = 1,2,\ldots,n\,. \qquad (28)$$

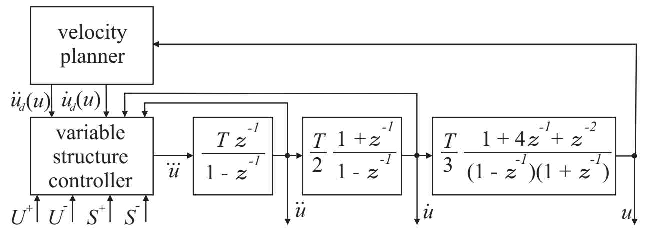

velocity planner
$\ddot{u}_d(u)$
$\dot{u}_d(u)$
variable structure controller
U^+
U^-
S^+
S^-
$\dddot{u}$
$\frac{T z^{-1}}{1 - z^{-1}}$
$\ddot{u}$
$\frac{T}{2}\frac{1 + z^{-1}}{1 - z^{-1}}$
$\dot{u}$
$\frac{T}{3}\frac{1 + 4z^{-1} + z^{-2}}{(1 - z^{-1})(1 + z^{-1})}$
u

Fig. 2. A detail of the automatic trajectory scaler composed by a velocity planner and a nonlinear, discrete-time, scaling filter.

In this second case, feasibility is achieved if $\dddot{u} \in \bigcap_{k=1}^{n} [\delta_k \, , \gamma_k]$, with

$$\gamma_k = \begin{cases} \frac{\dot{\overline{\tau}}_k - \widetilde{b}_{2,k}}{b_{1,k}}, \text{if } b_{1,k} > 0 \\ \frac{\dot{\underline{\tau}}_k - \widetilde{b}_{2,k}}{b_{1,k}}, \text{if } b_{1,k} < 0 \\ \infty, \quad \text{if } b_{1,k} = 0 \end{cases} \text{ and } \delta_k = \begin{cases} \frac{\dot{\underline{\tau}}_k - \widetilde{b}_{2,k}}{b_{1,k}}, \text{if } b_{1,k} > 0 \\ \frac{\dot{\overline{\tau}}_k - \widetilde{b}_{2,k}}{b_{1,k}}, \text{if } b_{1,k} < 0 \\ -\infty, \quad \text{if } b_{1,k} = 0 \end{cases} \tag{29}$$

or, equivalently, if $\dddot{u} \in [U^- \, , U^+]$ where

$$U^+ := \min_{k=1,\ldots,n} \{\gamma_k\} \, , \quad U^- := \max_{k=1,\ldots,n} \{\delta_k\} \, . \tag{30}$$

Again, configurations such that $U^- > U^+$ could arise, thus indicating that the manipulator status is located inside an area where feasibility cannot be achieved. In this case, a feasible solution to the problem does not exist and dynamic limits are violated with certainty. A more detailed discussion on the solution feasibility can be found in [13].

V. THE TRAJECTORY SCALING FILTER

The trajectory scaler is implemented according to the scheme shown in Fig. 2. It was early anticipated that it is composed by two basic elements. The first is a velocity planner which provides a velocity reference signal $\dot{u}_d$ parametrized in function of the longitudinal coordinate u: given any point u along the curve its corresponding reference velocity is $\dot{u}_d(u)$. Function $\dot{u}_d(u)$ is supplied by the users and could be unfeasible with respect to (6) and (7). The second part of the system is the dynamic nonlinear filter which automatically scales $\dot{u}_d(u)$ in order to fulfill constraints (23) and (30) and, in turn, (6) and (7). It is made by a chain of three integrators driven by an algebraic nonlinear controller designed by means of variable structure techniques [17]. A discrete-time implementation has been considered. In the following, subscript i is used to denote sampled variables, so that $\dot{u}_{d_i}$ corresponds to the reference signal $\dot{u}_d$ acquired at time $t_i = iT$, where T is the sampling period.

The discrete state-space model of the integration chain is given by the following equation

$$\begin{bmatrix} u_{i+1} \\ \dot{u}_{i+1} \\ \ddot{u}_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ \dot{u}_i \\ \ddot{u}_i \end{bmatrix} + \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix} \dddot{u}_i \, . \tag{31}$$

It is driven by the following nonlinear controller C

$$C: \quad \dddot{u}_i := \begin{cases} U^- \text{sat}(\sigma_i) & \text{if } \sigma_i \geq 0 \\ -U^+ \text{sat}(\sigma_i) & \text{if } \sigma_i < 0 \end{cases} \tag{32}$$

$$\sigma_i := \dot{z}_i - \dot{\tilde{z}}_i \, . \tag{33}$$

The two terms $\dot{z}_i$ and $\dot{\tilde{z}}_i$ are evaluated by means of the following algebraic equations

$$\dot{z}^+ := -\frac{S^+ - \ddot{u}_{d_i}}{TU^-} \, , \tag{34}$$

$$z^+ := -\lceil \dot{z}^+ \rceil \left[\dot{z}^+ - \frac{\lceil \dot{z}^+ \rceil - 1}{2} \right] , \tag{35}$$

$$\dot{z}^- := \frac{S^- - \ddot{u}_{d_i}}{TU^+} \, , \tag{36}$$

$$z^- := \lceil -\dot{z}^- \rceil \left[-\dot{z}^- - \frac{\lceil -\dot{z}^- \rceil - 1}{2} \right] , \tag{37}$$

$$[\alpha \ \beta] := \begin{cases} [U^+ \ U^-] & \text{if } \frac{\dot{y}_i}{T} + \frac{\ddot{y}_i}{2} > 0 \\ [U^- \ U^+] & \text{if } \frac{\dot{y}_i}{T} + \frac{\ddot{y}_i}{2} \leq 0 \end{cases} , \tag{38}$$

$$z_i := \frac{1}{T\alpha} \left| \frac{\dot{y}_i}{T} + \frac{\ddot{y}_i}{2} \right| , \tag{39}$$

$$\gamma_i := \begin{cases} z^+ & \text{if } z_i < z^+ \\ z_i & \text{if } z^+ \leq z_i \leq z^- \\ z^- & \text{if } z_i > z^- \end{cases} , \tag{40}$$

$$m_i := \text{Int} \left[\frac{1 + \sqrt{1 + 8 \, |\gamma_i|}}{2} \right] , \tag{41}$$

$$\dot{\tilde{z}}_i := -\frac{\gamma_i}{m_i} - \frac{m_i - 1}{2} \text{sgn}(\gamma_i) \, , \tag{42}$$

$$\dot{z}_i := \begin{cases} \frac{\ddot{y}_i}{T\,|\alpha|} & \text{if } \left[(z_i \geq 0 \;\&\; \frac{\ddot{y}_i}{T\,|\alpha|} \leq \dot{\tilde{z}}_i) \right. \quad (43) \\ & \left. \quad \text{or } (z_i < 0 \;\&\; \frac{\ddot{y}_i}{T\,|\alpha|} \geq \dot{\tilde{z}}_i) \right] \\ \frac{\ddot{y}_i}{T\,|\beta|} + \left(\frac{m_i - 1}{2} + \frac{|\gamma_i|}{m_i} \right) \frac{\alpha + \beta}{|\beta|} & \text{otherwise} \, . \end{cases} \tag{44}$$

Evidently, $\ddot{u}_{d_i}$ is the discrete-time derivative of the velocity reference signal, $\dot{y}_i := \dot{u}_i - \dot{u}_{d_i}$ is the filter velocity error, $\ddot{y}_i := \ddot{u}_i - \ddot{u}_{d_i}$ is the filter acceleration error. Function $\lceil \cdot \rceil$ provides the upper integer of its argument, while sat$(\cdot)$ saturates its argument to ± 1. Signals $\dot{u}_{d_i}$ and $\ddot{u}_{d_i}$ are assumed to be known and $\ddot{u}_{d_i}$ is supposed to be piece-wise constant.

The exact characterization of the filter is beyond the scopes of this paper due to space limitation, but it is important to point-out its main characteristics. Its output signal $\dot{u}_i$ perfectly tracks $\dot{u}_{d_i}$ until this latter fulfill constraints (23) and (30). If these conditions are not satisfied tracking is voluntarily lost in order to guarantee that $\dot{u}_i$ does not exceed the given limits. The filter is similar to that proposed in [13]: as soon as $\dot{u}_{d_i}$ becomes feasible, its tracking is newly gained without overshoot and in minimum time, but, and this is the novelty, the reference signal is now hanged with a deadbeat approach.

VI. A TEST CASE

The control strategy has been simulated by considering a RP planar manipulator characterized by the dynamic parameters reported in Table I. The reference path is an ellipse

TABLE I
ROBOT INERTIAL PARAMETERS

Link	Mass	Center of gravity			Inertia			Friction
q	m (Kg)	x(m)	y(m)	z(m)	$I_{xx}(Kg.m^2)$	$I_{yy}(Kg.m^2)$	$I_{zz}(Kg.m^2)$	$B(N.s/rad)$
θ_1	23.90	0	0.10	0	2.521	1.671	1.358	1.5e-3
d_2	3.88	0	-0.30	0	0.336	0.336	0.026	2.8e-3

parametrized as follows

$$\mathbf{f}(u) = \begin{bmatrix} \theta_1 \\ d_2 \end{bmatrix} := \begin{bmatrix} \text{Atan2}(0.8\sin u, 0.4\cos u) \\ \sqrt{0.4^2\cos^2 u + 0.8^2 \sin^2 u} \end{bmatrix}, u \in [0, 2\pi]. \tag{45}$$

The following tuning parameters have been selected for the controller: $\mathbf{k}_p = [500\ 400]^T$, $\mathbf{k}_v = [10\ 60]^T$. The velocity reference signal is shown in Fig. 3a by means of a dashed line. It is parametrized as follows

$$\dot{u}_d(u) = \begin{cases} -7.6(u-0.5)^2+2 & 0 \le u < 0.5 \\ 2 & 0.5 \le u < 1.8 \\ 1 & 1.8 \le u < 3.6 \\ -5.56(u-3.9)^2+1.5 & 3.6 \le u < 3.9 \\ 1.5 & 3.9 \le u \end{cases} \tag{46}$$

The corresponding reference acceleration is computed by considering the chain differentiation rule.

For the considered robot, dynamic saturations are assumed active on both joints. In particular, we suppose that $\tau_1, \tau_2 \in [-13, 13]$, $\dot{\tau}_1 \in [-200, 200]$, and $\dot{\tau}_2 \in [-150, 150]$. Reference signal (46) is unfeasible with respect to such dynamic constraints and for this reason it must be filtered in order to obtain a new feasible signal $\dot{u}$.

The system behavior can be understood with the help of Fig. 3b and Fig. 3c. Dashed lines correspond to upper and lower bounds on $\ddot{u}$ and $\dddot{u}$ evaluated by means of (23) and (30): the time scaling system generates an output signal $\dot{u}$ whose first and second derivatives fulfill the imposed constraints. A comparison between the original $\dot{u}_d$ and $\dot{u}$ is shown in Fig. 3a. The feasibility of the generated profile is proven by Fig. 4: $\boldsymbol{\tau}$ and $\dot{\boldsymbol{\tau}}$ always satisfy the given constraints.

The overall accuracy of the controller is verified by measuring the path tracking error defined as the Euclidean distance, expressed in function of u, between the manipulator tool frame and the reference path. Fig. 4e compares the errors detected with and without the filter: the maximum error without the filter is equal to 3.770e-2 m, while it decreases to 6.946e-4 m when the filter is used. The analysis of the tracking tolerance is beyond the scope of this paper: for the proposed filter it is currently not possible to predict or assign a desired path tracking accuracy. Adaptations of the techniques proposed in [8] could be used to this purpose.

VII. Conclusions

When saturations on GFDs are neglected path tracking can be easily lost. The use of appropriate online trajectory scaling methods can solve this issue. In the paper, a novel technique to be used with inverse dynamics controllers has been proposed. It is based on a dynamic filter which automatically modifies reference trajectories in order to preserve a correct path tracking. The new scaling method requires the online evaluation of the derivative of the manipulator inertia matrix: to this purpose an efficient algorithm has been formulated.

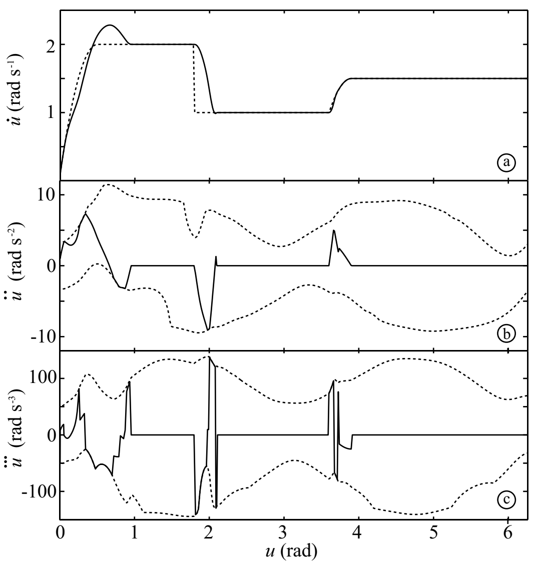


Fig. 3. a) Velocity reference signal $\dot{u}_d$ (dashed line) compared with the filter output $\dot{u}$ (solid line); b) and c) acceleration and jerks bounds online evaluated (dashed line) compared with the filter output $\ddot{u}$ and $\dddot{u}$ (solid line)

References

[1] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," *Journal of Dynamic Systems, Measurement, and Control*, vol. 106, pp. 102–106, 1984.

[2] S. Moon and S. Ahmad, "Time scaling of cooperative multirobot trajectories," *IEEE Transaction System Man and Cybernetics*, vol. 21, no. 4, pp. 900–908, 1991.

[3] A. De Luca and R. Farina, "Dynamic scaling of trajectories for robots with elastic joints," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Washington, DC, May 2002, pp. 2436–2442.

[4] O. Dahl and L. Nielsen, "Torque-limited path following by on-line trajectory time scaling," *IEEE Trans. on Robotics and Automation*, vol. 6, no. 5, pp. 554–561, 1990.

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

[6] H. Arai, K. Tanie, and S. Tachi, "Path tracking control of a manipulator considering torque saturation," *IEEE Trans. on Industrial Electronics*, vol. 41, pp. 25–31, February 1994.

[7] J. Moreno-Valenzuela, "Time-scaling of trajectories for point-to-point robotic tasks," *ISA Transaction*, vol. 45, pp. 407–418, July 2006.

[8] J. Kieffer, A. Cahill, and M. James, "Robust and accurate time-optimal path-tracking control for robot manipulators," *IEEE Trans. on Robotics and Automation*, vol. 13, no. 6, pp. 880–890, 1997.

[9] K. Kant and S. Zucker, "Toward efficient trajectory planning: the path-velocity decomposition," *Int. J. of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.

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

[11] S. Macfarlane and E. A. Croft, "Jerk-bounded manipulator trajectory planning: design for real-time applications," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 1, pp. 42–52, 2003.

[12] A. Gasparetto and V. Zanotto, "A technique of time-jerk optimal planning of robot trajectories," *Robotics and Computer-Integrated Manufacturing*, vol. 24, pp. 415–426, 2008.

[13] O. Gerelli and C. Guarino Lo Bianco, "Real-time path-tracking control of robotic manipulators with bounded torques and torque-derivatives," in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2008*, Nice, France, Sept. 2008, pp. 532–537.

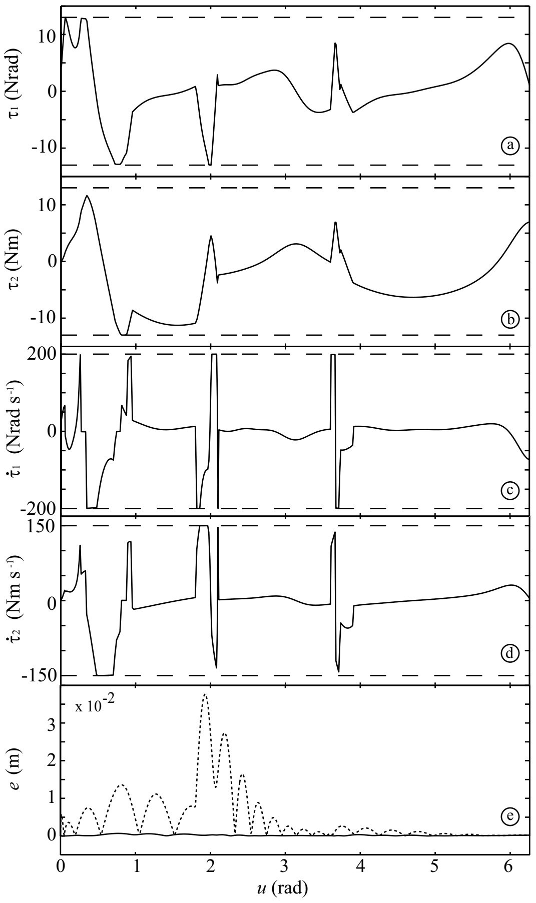


Fig. 4. a)-d) Generalized forces and their derivatives for the two joints; e) the path tracking error without (dashed line) and with (solid line) the velocity scaling filter.

[14] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*, ser. Advanced Textbooks in Control and Signal Processing. Berlin, Germany: Springer-Verlag, 2000.

[15] M. Walker and D. Orin, "Efficient dynamic computer simulation of robotic mechanism," *J. of Dynamic Systems, Measurement, Control*, no. 104, pp. 205–211, 1982.

[16] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-line computational scheme for mechanical manipulators," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 69–76, 1980.

[17] V. Utkin, "Variable structure systems with sliding modes," *IEEE Transactions on Automatic Control*, vol. 22, pp. 212–222, 1977.

[18] C. Guarino Lo Bianco and E. Fantini, "A recursive newton-euler approach for the evaluation of generalized forces derivatives," in *IEEE 12th Int. Conf. on Methods and Models in Automation and Robotics*, Miedzyzdroje, Poland, August 2006, pp. 739–744.

APPENDIX
EVALUATION OF $\widetilde{\mathbf{b}}_2(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$

Vector $\widetilde{\mathbf{b}}_2(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ must be evaluated in real time, so that an efficient algorithm must be formulated. Most of the terms in (27) have a negligible computational burden, since known efficient techniques can be used for their evaluation. The only two terms which require attention are the derivative of the inertia matrix, i.e., $\dot{\mathbf{H}}(\mathbf{q},\dot{\mathbf{q}})$, and the derivatives of the Coriolis, centripetal, gravity and friction forces, i.e., $\mathbf{D}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}}+2\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{\mathbf{q}}+\mathbf{B}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}}+\mathbf{E}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{\mathbf{q}}$. The evaluation of this last term is straightforward once $\dot{\mathbf{H}}(\mathbf{q},\dot{\mathbf{q}})$ is known: it can be obtained by evaluating (16) for $\dddot{\mathbf{q}}=0$.

Coefficients of $\dot{\mathbf{H}}(\mathbf{q},\dot{\mathbf{q}})$ can be obtained with a two step approach. The first step determines terms $c_{kj}(\mathbf{q},\dot{\mathbf{q}})$ of the Coriolis/centripetal matrix $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$, then the second step devises terms $\dot{h}_{kj}(\mathbf{q},\dot{\mathbf{q}})$ of $\dot{\mathbf{H}}(\mathbf{q},\dot{\mathbf{q}})$.

Let us indicate the unit vectors of a standard orthonormal base as $\mathbf{e}_j\in\mathbb{R}^n, j=1,2,\ldots,n$: the j-th component of $\mathbf{e}_j$ is equal to one, the other elements are equal to zero. In the following, friction and gravity coefficients are always set equal to zero, so that (9) and (11) simplify as follows

$$\begin{aligned}\tau_k &= \sum_{j=1}^{n} h_{kj}(\mathbf{q})\,\ddot{q}_j+\sum_{j=1}^{n}\sum_{i=1}^{n} c_{ijk}(\mathbf{q})\,\dot{q}_i\,\dot{q}_j\,, \qquad (47)\\ \dot{\tau}_k &= \sum_{j=1}^{n} \dot{h}_{kj}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{q}_j+\sum_{j=1}^{n} h_{kj}(\mathbf{q})\,\dddot{q}_j+\\ &\quad \sum_{j=1}^{n} d_{kj}(\mathbf{q},\dot{\mathbf{q}})\,\dot{q}_j+2\sum_{j=1}^{n} c_{kj}(\mathbf{q},\dot{\mathbf{q}})\,\ddot{q}_j\,. \qquad (48)\end{aligned}$$

For efficiency reasons, generalized forces τ_k can be evaluated by means of standard recursive Newton-Euler methods [16], while their derivatives $\dot{\tau}_k$ can be obtained through a recently devised extended Newton-Euler approach [18].

As a first step, the Newton-Euler algorithm is invoked n times with $\ddot{\mathbf{q}}=0$, $\dot{\mathbf{q}}=\mathbf{e}_j; j=1,2,\ldots,n$. It is evident from (47) that, under these conditions, the recursive algorithm returns all the Christoffel symbols which have the same first two indexes, i.e.,

$$y_{kj}(\mathbf{q}):=\tau_k=c_{jjk}(\mathbf{q})\,. \qquad (49)$$

Subsequently, the inverse dynamics is newly evaluated with $\ddot{\mathbf{q}}=0$, $\dot{\mathbf{q}}=\mathbf{e}_j+\mathbf{e}_i; i,j=1,2,\ldots,n; i\neq j$. It is easy to verify that this time (47) returns

$$\widetilde{y}_{ijk}(\mathbf{q}):=\tau_k=c_{jjk}(\mathbf{q})+c_{jik}(\mathbf{q})+c_{ijk}(\mathbf{q})+c_{iik}(\mathbf{q})\,. \qquad (50)$$

Since $c_{jik}(\mathbf{q})=c_{ijk}(\mathbf{q})$, and remembering that terms $c_{jjk}(\mathbf{q})=y_{kj}$ have already been computed, it is possible to reorganize (50) and infer that

$$c_{ijk}(\mathbf{q})=\frac{\widetilde{y}_{ijk}(\mathbf{q})-y_{kj}(\mathbf{q})-y_{ki}(\mathbf{q})}{2}\,. \qquad (51)$$

Once all Christoffel symbols $c_{ijk}(\mathbf{q})$ have been evaluated, it is possible to get elements $c_{kj}(\mathbf{q},\dot{\mathbf{q}})$ of matrix $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ by means of (10).

The second step of the procedure is based on the use of the extended Newton-Euler algorithm [18]. If we assume $\dddot{\mathbf{q}}=0$, $\ddot{\mathbf{q}}=\mathbf{e}_j; j=1,2,\ldots,n$, (48) returns

$$w_{kj}(\mathbf{q},\dot{\mathbf{q}}):=\dot{\tau}_k=\dot{h}_{kj}(\mathbf{q},\dot{\mathbf{q}})+2c_{kj}(\mathbf{q},\dot{\mathbf{q}})+\sum_{j=1}^{n} d_{kj}(\mathbf{q},\dot{\mathbf{q}})\dot{q}_j. \qquad (52)$$

Analogously, by assuming $\dddot{\mathbf{q}}=0$, $\ddot{\mathbf{q}}=0$, from (48) it descends that

$$\widetilde{w}_{kj}(\mathbf{q},\dot{\mathbf{q}}):=\dot{\tau}_k=\sum_{j=1}^{n} d_{kj}(\mathbf{q},\dot{\mathbf{q}})\,\dot{q}_j\,. \qquad (53)$$

By rearranging (52) and considering (53), we finally obtain the elements of matrix $\dot{\mathbf{H}}(\mathbf{q},\dot{\mathbf{q}})$

$$\dot{h}_{kj}(\mathbf{q},\dot{\mathbf{q}})=w_{kj}(\mathbf{q},\dot{\mathbf{q}})-\widetilde{w}_{kj}(\mathbf{q},\dot{\mathbf{q}})-2c_{kj}(\mathbf{q},\dot{\mathbf{q}})\,. \qquad (54)$$