# Efficient computation of inverse dynamics and feedback linearization for VSA-based robots

Gabriele Buondonno and Alessandro De Luca

***Abstract*—We develop a recursive numerical algorithm to compute the inverse dynamics of robot manipulators with an arbitrary number of joints, driven by Variable Stiffness Actuation (VSA) of the antagonistic type. The algorithm is based on Newton-Euler dynamic equations, generalized up to the fourth differential order to account for the compliant transmissions, combined with the decentralized nonlinear dynamics of the variable stiffness actuators at each joint. A variant of the algorithm can be used also for implementing a feedback linearization control law for the accurate tracking of desired link and stiffness trajectories. As in its simpler versions, the algorithm does not require dynamic modeling in symbolic form, does not use numerical approximations, grows linearly in complexity with the number of joints, and is suitable for on-line feedforward and real-time feedback control. A Matlab/C code is made available.**

***Index Terms*—Flexible Robots; Compliant Joint/Mechanism; Direct/Inverse Dynamics Formulation; Motion Control of Manipulators; Feedback Linearization**

## I. Introduction

THE use of compliant transmission elements in lightweight robotic devices is one of the current trends in research and applications, motivated by the (mutually non-exclusive) goals of realizing more natural robot behaviors, improving safety in human-robot interaction, optimizing energy consumption, or being capable of explosive tasks with limited actuation torque [1], [2]. The explicit introduction of (constant) elasticity at the joints, once considered only a parasitic feature in industrial robots, provides a first response to these objectives, as soon as its effects are modeled and controlled. In fact, by suitable model-based control designs [3] it is possible to obtain a desired compliant behavior of the robot in response to contact forces, without giving away accuracy in the execution of reference trajectories, provided these are sufficiently smooth.

Even beyond this, one can achieve simultaneous control of joint stiffness and link motion by taking inspiration from bio-mechanical analogies. For this, two motors are used at each robot joint, connected though nonlinear compliant transmissions to the driven link. In the last ten years or so, there has been a consistent activity in the design of Variable Stiffness Actuation (VSA) [4]. For single-degree-of-freedom (single-dof) devices, the two motors work either in antagonistic mode, collaborating in a similar way to both control objectives (as in the VSA-II of the University of Pisa [5] and the MACCEPA [6]) or in serial mode, where the smaller, secondary motor is used only to modulate stiffness on line (as in the DLR FVJ device [7] and IIT AWAS series [8]). These elementary devices, suitably combined, are currently paving the way to complete VSA-based manipulators and humanoids, sometimes considered the robots of the next generation. Multi-dof examples include the DLR Hand Arm System (HASy) [9], the IIT CompAct arm [10], and the VSA CubeBot modular series from Pisa, now commercialized as low-cost qbmove kits [11]. Notably, a large class of these single-dof or multi-dof VSA-based robots with complex nonlinear dynamics turns out to satisfy, under mild conditions, the necessary and sufficient conditions for feedback linearization and input-output decoupling [12], [13], a result that generalizes from the rigid case and from the case of robots with finite, but constant joint stiffness [14].

Together with these technological developments, a need arises to develop exact and efficient tools for addressing a version of the inverse dynamics problem for VSA-based robots with an arbitrary large number of joints, namely computing the motor torque commands that smoothly execute desired link and stiffness trajectories in nominal conditions. Similarly, we would like to be able to evaluate in real time also the complex but best performing feedback linearization control law using state measurements obtained on line to track in a stable way such trajectories. Moreover, it would be convenient if both tasks did not require the analytic development of a dynamic model in symbolic form.

In a recent paper [15], we designed a solution algorithm, that we called EJ-NEA (Elastic Joints Newton-Euler Algorithm), that works for robots with elastic joints (of *constant* stiffness) as a direct extension of the well-known recursive Newton-Euler Algorithm (NEA) for rigid robots [16]. For generic VSA-based robots with antagonistic actuation, we propose here a generalized version called Variable Stiffness Actuation Newton-Euler Algorithm (VSA-NEA).

## II. Background

For a rigid robot consisting of an open kinematic chain with $N$ moving links and $N$ joints, the dynamic model is

$$(\boldsymbol{M}(\boldsymbol{q})+\boldsymbol{B})\,\ddot{\boldsymbol{q}}+\boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}})=\boldsymbol{\tau}, \tag{1}$$

where $\boldsymbol{q}\in\mathbb{R}^N$ is the link position, $\boldsymbol{\tau}\in\mathbb{R}^N$ is the motor torque, $\boldsymbol{M}(\boldsymbol{q})>0$ is the inertia matrix of the robot links, $\boldsymbol{B}>0$ is the constant diagonal matrix with the drive inertia moments (as reflected through the reduction ratios), and $\boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}})=\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}})+\boldsymbol{g}(\boldsymbol{q})+\boldsymbol{D}\dot{\boldsymbol{q}}$ contains centrifugal, Coriolis,

Manuscript received: August 31, 2015; Revised November 21, 2015; Accepted January 16, 2016.

This paper was recommended for publication by Editor Kevin Lynch upon evaluation of the Associate Editor and Reviewers' comments. This work is supported by the European Commission, within the FP7 ICT-287513 SAPHARI project (www.saphari.eu).

The authors are with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto 25, 00185 Roma, Italy (e-mail: buondonno@diag.uniroma1.it; deluca@diag.uniroma1.it).

Digital Object Identifier (DOI): see top of this page.

and gravitational terms, as well as viscous friction (as a dissipative term, with diagonal $\boldsymbol{D} \geq 0$).

Given a twice-differentiable trajectory $\boldsymbol{q}_d(t)$, we compute the torques needed to execute the motion (inverse dynamics problem) in an efficient numerical way, especially when $N$ is large, using the recursive Newton-Euler Algorithm (NEA)

$$\boldsymbol{\tau}_d = \text{NEA}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d) = (\boldsymbol{M}(\boldsymbol{q}_d)+\boldsymbol{B})\,\ddot{\boldsymbol{q}}_d + \boldsymbol{n}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d). \quad (2)$$

To achieve feedback linearization (and input-output decoupling), it is sufficient to let

$$\boldsymbol{\tau} = \text{NEA}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{a}) = (\boldsymbol{M}(\boldsymbol{q})+\boldsymbol{B})\,\boldsymbol{a} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \quad (3)$$

where $\boldsymbol{a} = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{K}_P(\boldsymbol{q}_d - \boldsymbol{q})$, with diagonal matrices $\boldsymbol{K}_P > 0$ and $\boldsymbol{K}_D > 0$, for trajectory stabilization.

In the presence of joint elasticity, the link positions are distinct from the motor positions. A robot with $N$ elastic joints (of constant stiffness, and with one motor per joint) will need $2N$ generalized coordinates $\boldsymbol{\Theta} = (\boldsymbol{q}^T\ \boldsymbol{\theta}_m^T)^T \in \mathbb{R}^{2N}$, where $\boldsymbol{\theta}_m \in \mathbb{R}^N$ is the motor position after the reduction gears. Under the commonly used modelling assumptions introduced by Spong [14], the dynamic model is

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\Sigma}(\boldsymbol{\theta}_m - \boldsymbol{q}) \quad (4a)$$

$$\boldsymbol{B}\ddot{\boldsymbol{\theta}}_m + \boldsymbol{D}_m\dot{\boldsymbol{\theta}}_m + \boldsymbol{\Sigma}(\boldsymbol{\theta}_m - \boldsymbol{q}) = \boldsymbol{\tau} \quad (4b)$$

where $\boldsymbol{\Sigma} > 0$ is the joint stiffness matrix and $\boldsymbol{D}_m > 0$ is the matrix of motor viscous coefficients, both diagonal. The elastic torque $\boldsymbol{\tau}_e = \boldsymbol{\Sigma}(\boldsymbol{\theta}_m - \boldsymbol{q})$ couples the link equation (4a) and the motor equation (4b). A Newton-Euler algorithm for computing inverse dynamics and feedback linearization control of elastic joint robots modeled by (4a–4b) has been introduced in [15].

## III. VSA-BASED ROBOTS

### A. Modeling

In robots having Variable Stiffness Actuation, *two* actuators are present at each joint to command both link position and joint stiffness. This will require $3N$ generalized coordinates $\boldsymbol{\Theta} = (\boldsymbol{q}^T\ \boldsymbol{\theta}_{m1}^T\ \boldsymbol{\theta}_{m2}^T)^T \in \mathbb{R}^{3N}$, where $\boldsymbol{\theta}_{m1}$ and $\boldsymbol{\theta}_{m2}$ are now the motor positions after the reduction. Many different VSA designs exist, the most common being the decoupled agonistic-antagonistic setup. Defining the transmission deflections $\boldsymbol{\phi}_1 = \boldsymbol{\theta}_{m1} - \boldsymbol{q}$ and $\boldsymbol{\phi}_2 = \boldsymbol{\theta}_{m2} - \boldsymbol{q}$, under assumptions as in [14], the dynamic model becomes

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau}_{e1}(\boldsymbol{\phi}_1) + \boldsymbol{\tau}_{e2}(\boldsymbol{\phi}_2) \quad (5a)$$

$$\boldsymbol{B}_1\,\ddot{\boldsymbol{\theta}}_{m1} + \boldsymbol{D}_{m1}\,\dot{\boldsymbol{\theta}}_{m1} + \boldsymbol{\tau}_{e1}(\boldsymbol{\phi}_1) = \boldsymbol{\tau}_1 \quad (5b)$$

$$\boldsymbol{B}_2\,\ddot{\boldsymbol{\theta}}_{m2} + \boldsymbol{D}_{m2}\,\dot{\boldsymbol{\theta}}_{m2} + \boldsymbol{\tau}_{e2}(\boldsymbol{\phi}_2) = \boldsymbol{\tau}_2, \quad (5c)$$

where $\boldsymbol{\tau}_{e1}$ and $\boldsymbol{\tau}_{e2}$ are the flexibility torques, which are *nonlinear* functions of $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$, decoupled for each joint. The $6N$-dimensional state of the system is given by $(\boldsymbol{q}, \boldsymbol{\theta}_{m1}, \boldsymbol{\theta}_{m2}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{\theta}}_{m1}, \dot{\boldsymbol{\theta}}_{m2})$. The total flexibility torque acting on the links is

$$\boldsymbol{\tau}_e = \boldsymbol{\tau}_{e1}(\boldsymbol{\phi}_1) + \boldsymbol{\tau}_{e2}(\boldsymbol{\phi}_2) \quad (6)$$

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau}_e \quad (7)$$

An analytic formulation of $\boldsymbol{\tau}_{e1}$ and $\boldsymbol{\tau}_{e2}$ can be obtained from the expression of the total elastic energy $U_e$ of the system:

$$\boldsymbol{\tau}_{ek} = \left(\frac{\partial U_e}{\partial \boldsymbol{\phi}_k}\right)^T, \quad k = 1, 2, \qquad \boldsymbol{\tau}_e = -\left(\frac{\partial U_e}{\partial \boldsymbol{q}}\right)^T. \quad (8)$$

The stiffness matrices are physically defined as

$$\boldsymbol{\Sigma}_k(\boldsymbol{\phi}_k) = \frac{\partial \boldsymbol{\tau}_{ek}}{\partial \boldsymbol{\phi}_k}, \quad k = 1, 2, \qquad \boldsymbol{\Sigma}(\boldsymbol{\Theta}) = -\frac{\partial \boldsymbol{\tau}_e}{\partial \boldsymbol{q}}. \quad (9)$$

Since the motors dynamics are decoupled, these are diagonal matrices, with diagonals elements that can be arranged in the $N$-vectors $\boldsymbol{\sigma}_1$, $\boldsymbol{\sigma}_2$, and $\boldsymbol{\sigma}$. From (6) and (9), it follows

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 \quad (10)$$

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_1(\boldsymbol{\phi}_1) + \boldsymbol{\sigma}_2(\boldsymbol{\phi}_2) \quad (11)$$

### B. Inverse dynamics

Given a desired four-times differentiable link trajectory $\boldsymbol{q}_d(t)$ and a desired twice-differentiable stiffness trajectory $\boldsymbol{\sigma}_d(t)$ (with $\dddot{\boldsymbol{q}}_d$ and $\ddot{\boldsymbol{\sigma}}_d$ possibly piecewise continuous), we can compute the nominal input torques $\boldsymbol{\tau}_{d1}(t)$ and $\boldsymbol{\tau}_{d2}(t)$ that realize this combined task trajectory. Both reference trajectories $\boldsymbol{q}_d(t)$ and $\boldsymbol{\sigma}_d(t)$ are chosen by the user, and thus all their needed time derivatives are known by design. In the rest of this section, we will omit time dependence and subscript $d$ for notational simplicity.

The first step is to compute the desired $\boldsymbol{\tau}_e$ from (7). Its higher order derivatives are obtained similarly by differentiating (7) once[1]

$$\boldsymbol{M}(\boldsymbol{q})\dddot{\boldsymbol{q}} + \dot{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{n}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{\tau}}_e \quad (12)$$

and twice

$$\boldsymbol{M}(\boldsymbol{q})\ddddot{\boldsymbol{q}} + 2\dot{\boldsymbol{M}}(\boldsymbol{q})\dddot{\boldsymbol{q}} + \ddot{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \ddot{\boldsymbol{n}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \ddot{\boldsymbol{\tau}}_e, \quad (13)$$

and by plugging in the desired values of $\boldsymbol{q}$ and its derivatives.

Next, from the desired values of $\boldsymbol{\tau}_e$ and $\boldsymbol{\sigma}$, we need to compute the associated joint deflections $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$. These are obtained by solving the following system of $2N$ nonlinear equations

$$\begin{cases} \boldsymbol{\tau}_e = \boldsymbol{\tau}_{e1}(\boldsymbol{\phi}_1) + \boldsymbol{\tau}_{e2}(\boldsymbol{\phi}_2) \\ \boldsymbol{\sigma} = \boldsymbol{\sigma}_1(\boldsymbol{\phi}_1) + \boldsymbol{\sigma}_2(\boldsymbol{\phi}_2), \end{cases} \quad (14)$$

where $\boldsymbol{\tau}_{e1}(\cdot)$, $\boldsymbol{\tau}_{e2}(\cdot)$, $\boldsymbol{\sigma}_1(\cdot)$, and $\boldsymbol{\sigma}_2(\cdot)$ are seen as functions expressed in their analytical forms. In general, system (14) needs to be solved numerically, exploiting the joint decoupling property which causes the split into $N$ separate $2 \times 2$ subsystems (still nonlinear, though).

Next, we compute matrices $\boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_2$, $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$, $\boldsymbol{Z}$, $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$. These quantities are all obtained from the underlying spring model, using the solutions of (14). Matrices $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$, $\boldsymbol{Z}$, $\boldsymbol{H}_1$, and $\boldsymbol{H}_2$ are diagonal, with elements defined as:

$$\boldsymbol{Z}_{k,ii} = \frac{\partial \sigma_{ki}}{\partial \boldsymbol{\phi}_{ki}}, \quad k = 1, 2, \qquad \boldsymbol{Z} = \boldsymbol{Z}_1 + \boldsymbol{Z}_2 \quad (15)$$

$$\boldsymbol{H}_{k,ii} = \frac{\partial^2 \sigma_{ki}}{\partial \boldsymbol{\phi}_{ki}^2}, \qquad k = 1, 2. \quad (16)$$

[1]With $\dot{\boldsymbol{M}}(\boldsymbol{q})$, we mean $\frac{\mathrm{d}}{\mathrm{d}t}[\boldsymbol{M}(\boldsymbol{q})]$. Obviously, this quantity depends on $\dot{\boldsymbol{q}}$ as well as on $\boldsymbol{q}$. Similar comments apply to $\dot{\boldsymbol{n}}$ and to higher derivatives.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2016.2526072, IEEE Robotics and Automation Letters



From (6) and (9), we have then (with dependencies omitted)

$$\dot{\boldsymbol{\tau}}_e = \boldsymbol{\Sigma}_1\dot{\boldsymbol{\phi}}_1 + \boldsymbol{\Sigma}_2\dot{\boldsymbol{\phi}}_2. \tag{17}$$

Exploiting the decentralized structure of $\boldsymbol{\tau}_{e1}$ and $\boldsymbol{\tau}_{e2}$ and using (15), we can differentiate (11) and obtain

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{Z}_1\dot{\boldsymbol{\phi}}_1 + \boldsymbol{Z}_2\dot{\boldsymbol{\phi}}_2. \tag{18}$$

Putting (17) and (18) together, we set up the linear system

$$\begin{pmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_2 \\ \boldsymbol{Z}_1 & \boldsymbol{Z}_2 \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{\phi}}_1 \\ \dot{\boldsymbol{\phi}}_2 \end{pmatrix} = \boldsymbol{\mathcal{A}}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \begin{pmatrix} \dot{\boldsymbol{\phi}}_1 \\ \dot{\boldsymbol{\phi}}_2 \end{pmatrix} = \begin{pmatrix} \dot{\boldsymbol{\tau}}_e \\ \dot{\boldsymbol{\sigma}} \end{pmatrix}, \tag{19}$$

where $\boldsymbol{\mathcal{A}}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ is the *decoupling* matrix [13], in which we have emphasized the dependency from $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$. Provided the decoupling matrix is nonsingular, system (19) can be solved for $\dot{\boldsymbol{\phi}}_1$ and $\dot{\boldsymbol{\phi}}_2$ as

$$\begin{pmatrix} \dot{\boldsymbol{\phi}}_1 \\ \dot{\boldsymbol{\phi}}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_2 \\ \boldsymbol{Z}_1 & \boldsymbol{Z}_2 \end{pmatrix}^{-1} \begin{pmatrix} \dot{\boldsymbol{\tau}}_e \\ \dot{\boldsymbol{\sigma}} \end{pmatrix}. \tag{20}$$

Differentiating now (17) and rearranging terms yields

$$\boldsymbol{\Sigma}_1\ddot{\boldsymbol{\theta}}_{m1} + \boldsymbol{\Sigma}_2\ddot{\boldsymbol{\theta}}_{m2} = \ddot{\boldsymbol{\tau}}_e - \boldsymbol{Z}_1\dot{\boldsymbol{\phi}}_1^2 - \boldsymbol{Z}_2\dot{\boldsymbol{\phi}}_2^2 + \boldsymbol{\Sigma}\ddot{\boldsymbol{q}} = \boldsymbol{\beta}_1, \tag{21}$$

where vectors $\dot{\boldsymbol{\phi}}_i^2$ $(i = 1, 2)$ are squared component-wise. Using (16), we can diffentiate also (18) and rearrange it as

$$\boldsymbol{Z}_1\ddot{\boldsymbol{\theta}}_{m1} + \boldsymbol{Z}_2\ddot{\boldsymbol{\theta}}_{m2} = \ddot{\boldsymbol{\sigma}} - \boldsymbol{H}_1\dot{\boldsymbol{\phi}}_1^2 - \boldsymbol{H}_2\dot{\boldsymbol{\phi}}_2^2 + \boldsymbol{Z}\ddot{\boldsymbol{q}} = \boldsymbol{\beta}_2 \tag{22}$$

Equations (21) and (22) provide another linear system,

$$\begin{pmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_2 \\ \boldsymbol{Z}_1 & \boldsymbol{Z}_2 \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{\theta}}_{m1} \\ \ddot{\boldsymbol{\theta}}_{m2} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{pmatrix}, \tag{23}$$

which has the same coefficient matrix $\boldsymbol{\mathcal{A}}$ that appears in eq. (19), making only one inversion necessary to obtain also $\ddot{\boldsymbol{\theta}}_{m1}$ and $\ddot{\boldsymbol{\theta}}_{m2}$. Finally, the desired motor torques $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ are obtained from (5b) and (5c). Also, $\boldsymbol{\tau}_{e1}$ and $\boldsymbol{\tau}_{e2}$ can be evaluated from the spring model using the computed values of $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$, while $\dot{\boldsymbol{\theta}}_{m1} = \dot{\boldsymbol{\phi}}_1 + \dot{\boldsymbol{q}}$ and $\dot{\boldsymbol{\theta}}_{m2} = \dot{\boldsymbol{\phi}}_2 + \dot{\boldsymbol{q}}$.

Some remarks are now in order.

- Computation will be terminated if (14) has no solution. This means that the desired combined motion-stiffness task is not attainable by the given actuator model. Instead, existence of multiple solutions is not a source of problems. In fact, when the system is solved for a (discretized) time sequence of desired data, at each instant the solution is found by a numerical search, starting from the result obtained at the previous step. Since the search is a local process, the new solution will always be close to the older one achieving continuity.
- If $\boldsymbol{\mathcal{A}}$ is not invertible, the system is in a dynamic singularity, restricting the desired motion-stiffness task that can be achieved. Thanks to the joint decoupling property, $\boldsymbol{\mathcal{A}}$ can be rearranged into a block-diagonal matrix with $2 \times 2$ blocks, which can be inverted separately from each other. Thus matrix $\boldsymbol{\mathcal{A}}$ is singular if and only if at least one of these $N$ blocks is singular. For instance, for a joint $i$ with two identical antagonistic springs, this will always happen when $\phi_{1i} = \phi_{2i}$.
- If the initial values of $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ are known, we can avoid the explicit solution of (14) at each time instant. Instead, we can solve (19) and integrate for $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$. Caution must be taken, however, due to error drifts caused by numerical integration.

We provide now some further insight on the solution of system (14) in the case of identical springs with *cubic* flexible torque on both sides of each joint

$$\boldsymbol{\tau}_{e1}(\boldsymbol{\phi}_1) = \boldsymbol{K}\boldsymbol{\phi}_1 + \boldsymbol{K}_c\boldsymbol{\phi}_1^3 \tag{24a}$$

$$\boldsymbol{\tau}_{e2}(\boldsymbol{\phi}_2) = \boldsymbol{K}\boldsymbol{\phi}_2 + \boldsymbol{K}_c\boldsymbol{\phi}_2^3, \tag{24b}$$

with constant, positive $\boldsymbol{K} = \mathrm{diag}\{K_i\}$ and $\boldsymbol{K}_c = \mathrm{diag}\{K_{ci}\}$. Therefore, for each $i = 1, \ldots, N$ and $k = 1, 2$, we have

$$\sigma_{ki} = K_i + 3K_{ci}\,\phi_{ki}^2 \tag{25}$$

$$\zeta_{ki} = 6K_{ci}\,\phi_{ki} \tag{26}$$

$$\eta_{ki} = 6K_{ci} \tag{27}$$

The minimum possible stiffness for joint $i$ is thus $\sigma_i = 2K_i$. In order to avoid singularity problems, this lower bound should never be reached. We solve now eq. (14) as follows. For each joint $i$, consider the system

$$\tau_{ei} = K_i(\phi_{1i} + \phi_{2i}) + K_{ci}(\phi_{1i}^3 + \phi_{2i}^3) \tag{28}$$

$$\sigma_i = 2K_i + 3K_{ci}(\phi_{1i}^2 + \phi_{2i}^2). \tag{29}$$

Defining from (29)

$$R_i^2 = \phi_{1i}^2 + \phi_{2i}^2 = \frac{\sigma_i - 2K_i}{3K_{ci}}, \tag{30}$$

it can be seen that all solutions are parametrized by a scalar $\xi_i \in [0, 2\pi)$ such that $\phi_{1i} = R_i\cos\xi_i$ and $\phi_{2i} = R_i\sin\xi_i$. Replacing in (28) we get

$$(\cos\xi_i + \sin\xi_i) + \frac{\sigma_i - 2K_i}{3K_i}(\cos^3\xi_i + \sin^3\xi_i) = \frac{\tau_{ei}}{K_i R_i}, \tag{31}$$

which is a single trigonometric equation in $\xi_i$. This equation is sufficiently smooth, and thus easily solvable by a numerical root finder. Note that at least one solution always exists if $|\tau_{e,i}| \leq \sqrt{2}K_iR_i[1 + 0.5(\sigma_i - 2K_i)/(3K_i)]$. Furthermore, also slightly larger values of $|\tau_{e,i}|$ are admissible when $\sigma_i > 4K_i$. However, in the case of the cubic model (and also more in general), it is recommended to set an upper bound to $\sigma_i$, in order to keep spring deformations reasonably limited.

### C. Feedback linearization control

The same procedure outlined in Sec. III-B can be used, with minor modifications, to implement a feedback linearization law for VSA robots. With the desired trajectories $\boldsymbol{q}_d(t)$ and $\boldsymbol{\sigma}_d(t)$, we compute the linear control signals $\boldsymbol{v}_{\boldsymbol{q}}$ and $\boldsymbol{v}_{\boldsymbol{\sigma}}$ from the *actual* values of $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, $\dddot{\boldsymbol{q}}$, $\boldsymbol{\sigma}$, and $\dot{\boldsymbol{\sigma}}$

$$\begin{aligned} \boldsymbol{v}_{\boldsymbol{q}} = \dddot{\boldsymbol{q}}_{\mathrm{d}} &+ \boldsymbol{K}_3(\ddot{\boldsymbol{q}}_d - \ddot{\boldsymbol{q}}) + \boldsymbol{K}_2(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) \\ &+ \boldsymbol{K}_1(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{K}_0(\boldsymbol{q}_d - \boldsymbol{q}) \end{aligned} \tag{32a}$$

$$\boldsymbol{v}_{\boldsymbol{\sigma}} = \ddot{\boldsymbol{\sigma}}_d + \boldsymbol{K}_{\boldsymbol{\sigma}_1}(\dot{\boldsymbol{\sigma}}_d - \dot{\boldsymbol{\sigma}}) + \boldsymbol{K}_{\boldsymbol{\sigma}_0}(\boldsymbol{\sigma}_d - \boldsymbol{\sigma}), \tag{32b}$$

where all six gain matrices are diagonal with their diagonal elements being such that the two characteristic polynomials

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2016.2526072, IEEE Robotics and Automation Letters



$$p_{q,i}(s) = s^4 + K_{3i}s^3 + K_{2i}s^2 + K_{1i}s + K_{0i} \tag{33a}$$

$$p_{\sigma,i}(s) = s^2 + K_{\sigma_1,i}s + K_{\sigma_0,i} \tag{33b}$$

are Hurwitz. Then, the motor torques achieving feedback linearization control are obtained using the previous inverse dynamics computations, by only replacing $\dddot{\boldsymbol{q}}_d$ with $\boldsymbol{v_q}$ and $\ddot{\boldsymbol{\sigma}}_d$ with $\boldsymbol{v_\sigma}$. Use of high gains and possible addition of an integral action to $\boldsymbol{v_q}$ and $\boldsymbol{v_\sigma}$ (with similar requisites on the gain coefficients) can contribute to the controller accuracy and robustness to dynamic modeling errors.

However, the computations cannot be performed as such since they require, instead of desired values, measures of $\ddot{\boldsymbol{q}}$, $\dddot{\boldsymbol{q}}$, $\boldsymbol{\sigma}$, and $\dot{\boldsymbol{\sigma}}$ that are difficult or impossible to obtain. In the following, we assume that *full state* measurements are available, with the values of $\boldsymbol{q}$, $\boldsymbol{\phi}_1$, and $\boldsymbol{\phi}_2$ given, and those of $\dot{\boldsymbol{q}}$, $\dot{\boldsymbol{\phi}}_1$, and $\dot{\boldsymbol{\phi}}_2$ obtained by numerical differentiation.

First, the values of $\boldsymbol{\tau}_{e1}$, $\boldsymbol{\tau}_{e2}$, $\boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_2$, $\boldsymbol{Z}_1$, and $\boldsymbol{Z}_2$ (as well as of $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$) can be immediately computed from $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$. Next, $\boldsymbol{\tau}_e$, $\boldsymbol{\sigma}$, $\dot{\boldsymbol{\tau}}_e$ and $\dot{\boldsymbol{\sigma}}$ are obtained from (6), (11), (17), and (18), respectively. We invert then (7) and (12) to find the unknown values of $\ddot{\boldsymbol{q}}$ and $\dddot{\boldsymbol{q}}$ as

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{q})\big[\boldsymbol{\tau}_e - \boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}})\big] \tag{34}$$

$$\dddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{q})\left[\dot{\boldsymbol{\tau}}_e - \Big(\dot{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{n}}(\boldsymbol{q},\dot{\boldsymbol{q}})\Big)\right], \tag{35}$$

where $\ddot{\boldsymbol{q}}$ in (35) is the one just computed in (34). The solution of (14) is no longer necessary. This means that no numerical root finding is required for feedback linearization, which also eliminates the possibility of failure at this step. Finally, equations (7), (12), and (19) are also skipped as such.

## IV. The VSA-NEA Algorithm

Starting from the previous dynamic analysis in symbolic form, we present now a *numerical* algorithm for computing the desired motor torques $\boldsymbol{\tau}_{d1}$ and $\boldsymbol{\tau}_{d2}$ in the case of VSA-based robots, starting from sufficiently smooth $\boldsymbol{q}_d(t)$ and $\boldsymbol{\sigma}_d(t)$ and without the need to have available a symbolic dynamic model in closed form. The algorithm can be obtained as an extension of the standard NEA. Indeed, $\boldsymbol{\tau}_e$ itself can be computed by the standard NEA, the only difference being that the input parameters do not include information about motor inertias. The main difficulty lies in computing $\dddot{\boldsymbol{\tau}}_e$, and for this higher-order dynamic equations need to be considered —see (12) and (13). In particular, the recursive algorithm VSA-NEA will go two differentiation levels further than the NEA, computing higher derivatives of the motion variables.

As usual, all quantities will be conveniently expressed in the moving reference frame of the considered link, since all dynamic parameters of the robot will be constant in these frames. Before proceeding, we recall the notations used in the presence of multiple reference frames. The symbol ${}^i\boldsymbol{x}_j$ denotes a quantity $\boldsymbol{x}$ (e.g., a velocity) of link $j$ expressed in frame $i$, no superscript denoting by default $i = 0$. Thus, for instance, we have ${}^i\boldsymbol{\gamma}_i = {}^i\boldsymbol{R}_0\,{}^0\boldsymbol{\gamma}_i = \boldsymbol{R}_i^T\,\dot{\boldsymbol{\omega}}_i$. The orientation of frame $i$ with respect to $i-1$ and the relative position of the two origins, expressed in frame $i$, are described, respectively, by ${}^{i-1}\boldsymbol{R}_i$ and ${}^i\boldsymbol{p}_{i,i-1}$, using the the standard Denavit-Hartenberg convention. With $\hat{\boldsymbol{z}}_i$, we mean the $z$-axis unit vector of frame $i$. It can be easily seen that $\hat{\boldsymbol{z}}_0 = (0\ 0\ 1)^T$, ${}^i\hat{\boldsymbol{z}}_i = \hat{\boldsymbol{z}}_0$, $\hat{\boldsymbol{z}}_i = {}^0\boldsymbol{R}_i\hat{\boldsymbol{z}}_0$, and ${}^i\hat{\boldsymbol{z}}_{i-1} = {}^i\boldsymbol{R}_{i-1}\,{}^{i-1}\hat{\boldsymbol{z}}_{i-1} = {}^i\boldsymbol{R}_{i-1}\hat{\boldsymbol{z}}_0$.

The final forward and backward recursion are outlined next, while proofs are reported in the Appendix. In the following, we give the complete inverse dynamics algorithm, enabling to compute $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ from given values of $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, $\dddot{\boldsymbol{q}}$, $\ddddot{\boldsymbol{q}}$, $\boldsymbol{\sigma}$ and $\dot{\boldsymbol{\sigma}}$ (all quantities are desired ones). At the end, we explain what to change when computing the feedback linearization control law.

### A. Forward recursion

The following equations need to be propagated from the robot base to the tip, for $i = 1, \ldots, N$:

$${}^i\boldsymbol{\omega}_i = {}^i\boldsymbol{R}_{i-1}({}^{i-1}\boldsymbol{\omega}_{i-1} + \dot{\theta}_i\hat{\boldsymbol{z}}_0) \tag{36}$$

$${}^i\boldsymbol{\gamma}_i = {}^i\boldsymbol{R}_{i-1}({}^{i-1}\boldsymbol{\gamma}_{i-1} + \ddot{\theta}_i\hat{\boldsymbol{z}}_0 + {}^{i-1}\boldsymbol{\omega}_{i-1}\times\dot{\theta}_i\hat{\boldsymbol{z}}_0) \tag{37}$$

$$\begin{aligned}{}^i\boldsymbol{\iota}_i = {}^i\boldsymbol{R}_{i-1}[\,&{}^{i-1}\boldsymbol{\iota}_{i-1} + \dddot{\theta}_i\hat{\boldsymbol{z}}_0 + {}^{i-1}\boldsymbol{\gamma}_{i-1}\times\dot{\theta}_i\hat{\boldsymbol{z}}_0\\ &+ {}^{i-1}\boldsymbol{\omega}_{i-1}\times(2\ddot{\theta}_i\hat{\boldsymbol{z}}_0 + {}^{i-1}\boldsymbol{\omega}_{i-1}\times\dot{\theta}_i\hat{\boldsymbol{z}}_0)\,]\end{aligned} \tag{38}$$

$$\begin{aligned}{}^i\boldsymbol{\varsigma}_i = {}^i\boldsymbol{R}_{i-1}\{\,&{}^{i-1}\boldsymbol{\varsigma}_{i-1} + \ddddot{\theta}_i\hat{\boldsymbol{z}}_0 + 3\,{}^{i-1}\boldsymbol{\gamma}_{i-1}\times\ddot{\theta}_i\hat{\boldsymbol{z}}_0\\ &+ 3\,{}^{i-1}\boldsymbol{\omega}_{i-1}\times(\dddot{\theta}_i\hat{\boldsymbol{z}}_0 + {}^{i-1}\boldsymbol{\omega}_{i-1}\times\ddot{\theta}_i\hat{\boldsymbol{z}}_0)\\ &+ 2\,{}^{i-1}\boldsymbol{\gamma}_{i-1}\times({}^{i-1}\boldsymbol{\omega}_{i-1}\times\dot{\theta}_i\hat{\boldsymbol{z}}_0)\\ &+ {}^{i-1}\boldsymbol{\omega}_{i-1}\times[{}^{i-1}\boldsymbol{\omega}_{i-1}\times({}^{i-1}\boldsymbol{\omega}_{i-1}\times\dot{\theta}_i\hat{\boldsymbol{z}}_0)\\ &+ {}^{i-1}\boldsymbol{\gamma}_{i-1}\times\dot{\theta}_i\hat{\boldsymbol{z}}_0] + {}^{i-1}\boldsymbol{\iota}_{i-1}\times\dot{\theta}_i\hat{\boldsymbol{z}}_0\}\end{aligned} \tag{39}$$

$$\begin{aligned}{}^i\boldsymbol{a}_i = {}^i\boldsymbol{R}_{i-1}\,&{}^{i-1}\boldsymbol{a}_{i-1} + {}^i\boldsymbol{\omega}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1})\\ &+ {}^i\boldsymbol{\gamma}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1} + \ddot{d}_i\,{}^i\hat{\boldsymbol{z}}_{i-1} + 2\dot{d}_i({}^i\boldsymbol{\omega}_i\times{}^i\hat{\boldsymbol{z}}_{i-1})\end{aligned} \tag{40}$$

$${}^i\boldsymbol{a}_{c_i} = {}^i\boldsymbol{a}_i + {}^i\boldsymbol{\omega}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{c_i,i}) + {}^i\boldsymbol{\gamma}_i\times{}^i\boldsymbol{p}_{c_i,i} \tag{41}$$

$$\begin{aligned}{}^i\boldsymbol{j}_i = {}^i\boldsymbol{R}_{i-1}\,&{}^{i-1}\boldsymbol{j}_{i-1} + 2\,{}^i\boldsymbol{\gamma}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1})\\ &+ {}^i\boldsymbol{\omega}_i\times[{}^i\boldsymbol{\gamma}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1} + {}^i\boldsymbol{\omega}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1})]\\ &+ {}^i\boldsymbol{\iota}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1} + \dddot{d}_i\,{}^i\hat{\boldsymbol{z}}_{i-1} + 3\ddot{d}_i\,{}^i\boldsymbol{\omega}_i\times{}^i\hat{\boldsymbol{z}}_{i-1}\\ &+ 3\dot{d}_i[{}^i\boldsymbol{\gamma}_i\times{}^i\hat{\boldsymbol{z}}_{i-1} + {}^i\boldsymbol{\omega}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\hat{\boldsymbol{z}}_{i-1})]\end{aligned} \tag{42}$$

$$\begin{aligned}{}^i\boldsymbol{j}_{c_i} = {}^i\boldsymbol{j}_i + {}^i\boldsymbol{\iota}_i\times{}^i\boldsymbol{p}_{c_i,i} &+ 2\,{}^i\boldsymbol{\gamma}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{c_i,i})\\ &+ {}^i\boldsymbol{\omega}_i\times[{}^i\boldsymbol{\gamma}_i\times{}^i\boldsymbol{p}_{c_i,i} + {}^i\boldsymbol{\omega}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{c_i,i})]\end{aligned} \tag{43}$$

$$\begin{aligned}{}^i\boldsymbol{s}_i = {}^i\boldsymbol{R}_{i-1}\,&{}^{i-1}\boldsymbol{s}_{i-1} + {}^i\boldsymbol{\varsigma}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1} + 3\,{}^i\boldsymbol{\iota}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1})\\ &+ 3\,{}^i\boldsymbol{\gamma}_i\times[{}^i\boldsymbol{\gamma}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1} + {}^i\boldsymbol{\omega}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1})]\\ &+ {}^i\boldsymbol{\omega}_i\times[{}^i\boldsymbol{\iota}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1} + 2\,{}^i\boldsymbol{\gamma}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1})]\\ &+ {}^i\boldsymbol{\omega}_i\times\{{}^i\boldsymbol{\omega}_i\times[{}^i\boldsymbol{\gamma}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1} + {}^i\boldsymbol{\omega}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{i,i\text{-}1})]\}\\ &+ \ddddot{d}_i\,{}^i\hat{\boldsymbol{z}}_{i-1} + 4\dddot{d}_i\,{}^i\boldsymbol{\iota}_i\times{}^i\hat{\boldsymbol{z}}_{i-1} + 8\ddot{d}_i\,{}^i\boldsymbol{\gamma}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\hat{\boldsymbol{z}}_{i-1})\\ &+ 4\dot{d}_i\,{}^i\boldsymbol{\omega}_i\times[{}^i\boldsymbol{\gamma}_i\times{}^i\hat{\boldsymbol{z}}_{i-1} + {}^i\boldsymbol{\omega}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\hat{\boldsymbol{z}}_{i-1})]\\ &+ 6\ddot{d}_i[{}^i\boldsymbol{\gamma}_i\times{}^i\hat{\boldsymbol{z}}_{i-1} + {}^i\boldsymbol{\omega}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\hat{\boldsymbol{z}}_{i-1})]\\ &+ 4\dddot{d}_i\,{}^i\boldsymbol{\omega}_i\times{}^i\hat{\boldsymbol{z}}_{i-1}\end{aligned} \tag{44}$$

$$\begin{aligned}{}^i\boldsymbol{s}_{c_i} = {}^i\boldsymbol{s}_i + {}^i\boldsymbol{\varsigma}_i\times{}^i\boldsymbol{p}_{c_i,i} &+ 3\,{}^i\boldsymbol{\iota}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{c_i,i})\\ &+ 3\,{}^i\boldsymbol{\gamma}_i\times[{}^i\boldsymbol{\gamma}_i\times{}^i\boldsymbol{p}_{c_i,i} + {}^i\boldsymbol{\omega}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{c_i,i})]\\ &+ {}^i\boldsymbol{\omega}_i\times[{}^i\boldsymbol{\iota}_i\times{}^i\boldsymbol{p}_{c_i,i} + 2\,{}^i\boldsymbol{\gamma}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{c_i,i})]\\ &+ {}^i\boldsymbol{\omega}_i\times\{{}^i\boldsymbol{\omega}_i\times[{}^i\boldsymbol{\gamma}_i\times{}^i\boldsymbol{p}_{c_i,i} + {}^i\boldsymbol{\omega}_i\times({}^i\boldsymbol{\omega}_i\times{}^i\boldsymbol{p}_{c_i,i})]\}.\end{aligned} \tag{45}$$

The following symbols have been used:

$\boldsymbol{\omega}_i$ angular velocity of frame $i$;

$\boldsymbol{\gamma}_i$ angular acceleration of frame $i$;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2016.2526072, IEEE Robotics and Automation Letters



$\boldsymbol{\iota}_i$ angular jerk of frame $i$;
$\boldsymbol{\varsigma}_i$ angular snap of frame $i$;
$\boldsymbol{a}_i$ acceleration of the origin of frame $i$;
$\boldsymbol{a}_{c_i}$ acceleration of the center of mass (CoM) of link $i$;
$\boldsymbol{j}_i$ jerk of the origin of frame $i$;
$\boldsymbol{j}_{c_i}$ jerk of the CoM of frame $i$;
$\boldsymbol{s}_i$ snap of the origin of frame $i$;
$\boldsymbol{s}_{c_i}$ snap of the CoM of frame $i$;
$\boldsymbol{p}_{c_i,i}$ position of the CoM of *augmented* link $i$ (i.e., of the link plus the motor mounted on it) w.r.t. the origin of frame $i$.

The initialization of the forward recursion is zero for all quantities, with the exception of ${}^0\boldsymbol{a}_0$, which is set to $-\vec{\boldsymbol{g}}$ to account for gravitational effects, $\vec{\boldsymbol{g}}$ being the gravity acceleration vector in frame 0. The above equations are valid both for revolute and prismatic joints, with $\theta_i = q_i$ in the former case and $d_i = q_i$ in the latter. Moreover, if joint $i$ is prismatic, $\dot{\theta}_i = \ddot{\theta}_i = \dddot{\theta}_i = \ddddot{\theta}_i = 0$; if it is revolute, $\dot{d}_i = \ddot{d}_i = \dddot{d}_i = \ddddot{d}_i = 0$, with considerable simplifications.

### B. Backward recursion

The following equations need to be propagated from the robot tip to the base, for $i = N, \ldots, 1$:

$$ {}^i\boldsymbol{F}_i = m_i\, {}^i\boldsymbol{a}_{c_i} \tag{46} $$
$$ {}^i\dot{\boldsymbol{F}}_i = m_i\, {}^i\boldsymbol{j}_{c_i} \tag{47} $$
$$ {}^i\ddot{\boldsymbol{F}}_i = m_i\, {}^i\boldsymbol{s}_{c_i} \tag{48} $$
$$ {}^i\boldsymbol{N}_i = {}^i\boldsymbol{I}_i\, {}^i\boldsymbol{\gamma}_i + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{I}_i\, {}^i\boldsymbol{\omega}_i) \tag{49} $$
$$ \begin{aligned} {}^i\dot{\boldsymbol{N}}_i = {}& {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{I}_i\, {}^i\boldsymbol{\gamma}_i + {}^i\boldsymbol{N}_i) + {}^i\boldsymbol{I}_i({}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{\omega}_i + {}^i\boldsymbol{\iota}_i) \\ & + {}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{I}_i\, {}^i\boldsymbol{\omega}_i \end{aligned} \tag{50} $$
$$ \begin{aligned} {}^i\ddot{\boldsymbol{N}}_i = {}& {}^i\boldsymbol{\gamma}_i \times ({}^i\boldsymbol{I}_i\, {}^i\boldsymbol{\gamma}_i + 2\, {}^i\boldsymbol{N}_i) + {}^i\boldsymbol{\iota}_i \times {}^i\boldsymbol{I}_i\, {}^i\boldsymbol{\omega}_i \\ & + {}^i\boldsymbol{I}_i[2({}^i\boldsymbol{\iota}_i \times {}^i\boldsymbol{\omega}_i) + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{\gamma}_i) + {}^i\boldsymbol{\varsigma}_i] \\ & + {}^i\boldsymbol{\omega}_i \times [{}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{I}_i\, {}^i\boldsymbol{\gamma}_i + 2\, {}^i\boldsymbol{I}_i({}^i\boldsymbol{\iota}_i + {}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{\omega}_i) + {}^i\dot{\boldsymbol{N}}_i] \end{aligned} \tag{51} $$
$$ {}^i\boldsymbol{f}_i = {}^i\boldsymbol{R}_{i+1}\, {}^{i+1}\boldsymbol{f}_{i+1} + {}^i\boldsymbol{F}_i \tag{52} $$
$$ {}^i\dot{\boldsymbol{f}}_i = {}^i\boldsymbol{R}_{i+1}\, {}^{i+1}\dot{\boldsymbol{f}}_{i+1} + {}^i\dot{\boldsymbol{F}}_i \tag{53} $$
$$ {}^i\ddot{\boldsymbol{f}}_i = {}^i\boldsymbol{R}_{i+1}\, {}^{i+1}\ddot{\boldsymbol{f}}_{i+1} + {}^i\ddot{\boldsymbol{F}}_i \tag{54} $$
$$ \begin{aligned} {}^i\boldsymbol{n}_i = {}& {}^i\boldsymbol{R}_{i+1}\, {}^{i+1}\boldsymbol{n}_{i+1} + {}^i\boldsymbol{p}_{c_i,i} \times {}^i\boldsymbol{F}_i + {}^i\boldsymbol{p}_{i,i-1} \times {}^i\boldsymbol{f}_i \\ & + {}^i\boldsymbol{N}_i \end{aligned} \tag{55} $$
$$ \begin{aligned} {}^i\dot{\boldsymbol{n}}_i = {}& {}^i\boldsymbol{R}_{i+1}\, {}^{i+1}\dot{\boldsymbol{n}}_{i+1} + ({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{p}_{c_i,i}) \times {}^i\boldsymbol{F}_i \\ & + {}^i\boldsymbol{p}_{c_i,i} \times {}^i\dot{\boldsymbol{F}}_i + ({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{p}_{i,i-1} + \dot{d}_i\, {}^i\hat{\boldsymbol{z}}_{i-1}) \times {}^i\boldsymbol{f}_i \\ & + {}^i\boldsymbol{p}_{i,i-1} \times {}^i\dot{\boldsymbol{f}}_i + {}^i\dot{\boldsymbol{N}}_i \end{aligned} \tag{56} $$
$$ \begin{aligned} {}^i\ddot{\boldsymbol{n}}_i = {}& {}^i\boldsymbol{R}_{i+1}\, {}^{i+1}\ddot{\boldsymbol{n}}_{i+1} + 2\,({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{p}_{c_i,i}) \times {}^i\dot{\boldsymbol{F}}_i \\ & + [{}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{p}_{c_i,i} + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{p}_{c_i,i})] \times {}^i\boldsymbol{F}_i \\ & + [{}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{p}_{i,i-1} + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{p}_{i,i-1}) + \ddot{d}_i\, {}^i\hat{\boldsymbol{z}}_{i-1} \\ & + 2\dot{d}_i\, {}^i\boldsymbol{\omega}_i \times {}^i\hat{\boldsymbol{z}}_{i-1}] \times {}^i\boldsymbol{f}_i + {}^i\boldsymbol{p}_{c_i,i} \times {}^i\ddot{\boldsymbol{F}}_i + {}^i\ddot{\boldsymbol{N}}_i \\ & + 2\,({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{p}_{i,i-1} + \dot{d}_i\, {}^i\hat{\boldsymbol{z}}_{i-1}) \times {}^i\dot{\boldsymbol{f}}_i + {}^i\boldsymbol{p}_{i,i-1} \times {}^i\ddot{\boldsymbol{f}}_i \end{aligned} \tag{57} $$
$$ \tau_{ei} = \begin{cases} {}^i\boldsymbol{n}_i^T\, {}^i\hat{\boldsymbol{z}}_{i-1} + D_i\dot{q}_i & \text{if joint } i \text{ is revolute} \\ {}^i\boldsymbol{f}_i^T\, {}^i\hat{\boldsymbol{z}}_{i-1} + D_i\dot{q}_i & \text{if joint } i \text{ is prismatic} \end{cases} \tag{58} $$
$$ \dot{\tau}_{ei} = \begin{cases} ({}^i\dot{\boldsymbol{n}}_i + {}^i\boldsymbol{n}_i \times {}^i\boldsymbol{\omega}_i)^T\, {}^i\hat{\boldsymbol{z}}_{i-1} + D_i\ddot{q}_i \\ ({}^i\dot{\boldsymbol{f}}_i + {}^i\boldsymbol{f}_i \times {}^i\boldsymbol{\omega}_i)^T\, {}^i\hat{\boldsymbol{z}}_{i-1} + D_i\ddot{q}_i \end{cases} \tag{59} $$
$$ \ddot{\tau}_{ei} = \begin{cases} [{}^i\ddot{\boldsymbol{n}}_i + 2({}^i\dot{\boldsymbol{n}}_i \times {}^i\boldsymbol{\omega}_i) + {}^i\boldsymbol{n}_i \times {}^i\boldsymbol{\gamma}_i \\ \quad + ({}^i\boldsymbol{n}_i \times {}^i\boldsymbol{\omega}_i) \times {}^i\boldsymbol{\omega}_i]^T\, {}^i\hat{\boldsymbol{z}}_{i-1} + D_i\dddot{q}_i \\ [{}^i\ddot{\boldsymbol{f}}_i + 2({}^i\dot{\boldsymbol{f}}_i \times {}^i\boldsymbol{\omega}_i) + {}^i\boldsymbol{f}_i \times {}^i\boldsymbol{\gamma}_i \\ \quad + ({}^i\boldsymbol{f}_i \times {}^i\boldsymbol{\omega}_i) \times {}^i\boldsymbol{\omega}_i]^T\, {}^i\hat{\boldsymbol{z}}_{i-1} + D_i\dddot{q}_i \end{cases} \tag{60} $$
$$ \langle \tau_{ei}\,, \sigma_i \rangle \;\mapsto\; \langle \phi_{1i}\,, \phi_{2i} \rangle \tag{61} $$
$$ \tau_{e1,i} = \tau_{e1,i}(\phi_{1i}) \qquad \tau_{e2,i} = \tau_{e2,i}(\phi_{2i}) \tag{62} $$
$$ \sigma_{1i} = \sigma_{1i}(\phi_{1i}) \qquad \sigma_{2i} = \sigma_{2i}(\phi_{2i}) \tag{63} $$
$$ \zeta_{1i} = \frac{\partial \sigma_{1i}}{\partial \phi_{1i}} \qquad \zeta_{2i} = \frac{\partial \sigma_{2i}}{\partial \phi_{2i}} \tag{64} $$
$$ \zeta_i = \zeta_{1i} + \zeta_{2i} \tag{65} $$
$$ \eta_{1i} = \frac{\partial^2 \sigma_{1i}}{\partial \phi_{1i}^2} \qquad \eta_{2i} = \frac{\partial^2 \sigma_{2i}}{\partial \phi_{2i}^2} \tag{66} $$
$$ A_i^{-1} = \frac{1}{\sigma_{1i}\zeta_{2i} - \sigma_{2i}\zeta_{1i}} \begin{pmatrix} \zeta_{2i} & -\sigma_{2i} \\ -\zeta_{1i} & \sigma_{1i} \end{pmatrix} \tag{67} $$
$$ \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix} = A_i^{-1} \begin{pmatrix} \dot{\tau}_{ei} \\ \dot{\sigma}_i \end{pmatrix} \tag{68} $$
$$ \dot{\theta}_{m1,i} = \dot{\phi}_{1i} + \dot{q}_i \qquad \dot{\theta}_{m2,i} = \dot{\phi}_{2i} + \dot{q}_i \tag{69} $$
$$ \beta_{1i} = \ddot{\tau}_{ei} - \zeta_{1i}\dot{\phi}_{1i}^2 - \zeta_{2i}\dot{\phi}_{2i}^2 + \sigma_i\ddot{q}_i \tag{70} $$
$$ \beta_{2i} = \ddot{\sigma}_i - \eta_{1i}\dot{\phi}_{1i}^2 - \eta_{2i}\dot{\phi}_{2i}^2 + \zeta_i\ddot{q}_i \tag{71} $$
$$ \begin{pmatrix} \ddot{\theta}_{m1,i} \\ \ddot{\theta}_{m2,i} \end{pmatrix} = A_i^{-1} \begin{pmatrix} \beta_{1i} \\ \beta_{2i} \end{pmatrix} \tag{72} $$
$$ \tau_{1i} = B_{1i}\,\ddot{\theta}_{m1,i} + D_{m1,i}\,\dot{\theta}_{m1,i} + \tau_{e1,i} \tag{73} $$
$$ \tau_{2i} = B_{2i}\,\ddot{\theta}_{m2,i} + D_{m2,i}\,\dot{\theta}_{m2,i} + \tau_{e2,i}, \tag{74} $$

where:

$m_i$ mass of *augmented* link $i$;
${}^i\boldsymbol{I}_i$ central inertia tensor of link $i$, in frame $i$;
$\boldsymbol{F}_i$ total force acting on the CoM of link $i$;
$\boldsymbol{N}_i$ total torque acting on link $i$;
$\boldsymbol{f}_i$ total force exerted on link $i$ by link $i-1$;
$\boldsymbol{n}_i$ total torque exerted on link $i$ by link $i-1$;
$\sigma_{ki}$ is the $i$-th element of $\boldsymbol{\sigma}_k$;
$\zeta_{ki}$ is the $i$-th diagonal element of $\boldsymbol{Z}_k$;
$\eta_{ki}$ is the $i$-th diagonal element of $\boldsymbol{H}_k$.

In (61), we denoted by the symbol $\mapsto$ the resolution of the $i$-th subsystem of (14). For the initialization of the backward recursion, $\boldsymbol{f}_{N+1}$ and $\boldsymbol{n}_{N+1}$ are, respectively, the forces and torques exerted by the end-effector on the environment, i.e., the *opposite* of the external forces and torques acting on the end-effector. If present, these are passed to the algorithm as an additional input, otherwise they are set identically to $\boldsymbol{0}$. If the external forces/torques are already expressed in frame $N$, then ${}^N\boldsymbol{R}_{N+1}$ will be the $3 \times 3$ identity matrix.

While $m_i$ represents the mass of the *whole* augmented link, ${}^i\boldsymbol{I}_i$ accounts for all of the inertial properties of link $i$, *except* for the drive inertia moments.

### C. Variant for feedback linearization

A variant of the VSA-NEA algorithm can be used to compute a feedback linearization control, as explained in Sec. III-C. Equations (62)–(66) are evaluated first in a separate initial loop; next, (6), (11), (17), (18), (34) and (35) are evaluated. Indications on how to compute efficiently (34) and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2016.2526072, IEEE Robotics and Automation Letters



(35) can be found in [15]. During the execution, $\dddot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{\sigma}}$ are replaced by $\boldsymbol{v_q}$ and $\boldsymbol{v_\sigma}$, respectively. Equations (58)–(59), (61)–(66) and (68) are skipped.

## V. Numerical Results

The algorithms have been validated by several numerical tests. For illustration, we show here some meaningful results of the inverse dynamics computation. We consider a spatial 3-dof robot in the presence of gravity (acting on links 2 and 3), with the three rotational joints driven by antagonistic VSA with cubic profiles of flexibility torques. Its parameters are given in Tab. I.

TABLE I
KINEMATIC AND DYNAMIC PARAMETERS

| | | 1 | 2 | 3 |
|---|---|---|---|---|
| $\alpha_i$ | rad | $\pi/2$ | 0 | 0 |
| $a_i$ | cm | 0 | 30 | 30 |
| $d_i$ | cm | 0 | 0 | 0 |
| $\theta_i$ | rad | $q_1$ | $q_2$ | $q_3$ |

| | | | | |
|---|---|---|---|---|
| $m_i$ | kg | 9.0478 | 6.7858 | 5.0894 |
| ${}^i\boldsymbol{p}_{cx_i,i}$ | cm | 0 | -15 | -15 |
| ${}^i\boldsymbol{p}_{cy_i,i}$ | cm | 0 | 0 | 0 |
| ${}^i\boldsymbol{p}_{cz_i,i}$ | cm | 0 | 0 | 0 |
| ${}^i\boldsymbol{I}_{xx_i}$ | kg m$^2$ | 0.1299 | 0.0139 | 0.0104 |
| ${}^i\boldsymbol{I}_{yy_i}$ | kg m$^2$ | 0.0185 | 0.0578 | 0.0434 |
| ${}^i\boldsymbol{I}_{zz_i}$ | kg m$^2$ | 0.1299 | 0.0578 | 0.0434 |
| $D_i$ | kg m$^2/s$ | $10^{-5}$ | $10^{-5}$ | $10^{-5}$ |
| $B_{k,i}$ | kg m$^2$ | 3.20 | 3.05 | 1.98 |
| $D_{mk,\,i}$ | kg m$^2/s$ | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ |
| $K_i$ | Nm/rad | 400 | 400 | 400 |
| $K_{c,\,i}$ | Nm/rad$^3$ | 2000 | 2000 | 2000 |

The link motion $\boldsymbol{q}_d$ was specified by a smooth rest-to-rest trajectory (polynomials of degree 7, with zero initial and final boundary conditions on velocity, acceleration, and jerk) lasting $T = 4$ s. In a first test, $\boldsymbol{\sigma}_d$ was also specified by a rest-to-rest polynomial trajectory of degree 3, identical for all joints, with zero boundary conditions on $\dot{\boldsymbol{\sigma}}$ at $t = 0$ and $t = T$. The stiffness of each joint goes from $\sigma_{min} = 850$ Nm/rad, which is close to the physical minimum possible stiffness, to $\sigma_{max} = 1275$ Nm/rad, achieving a 50% increase. Figure 1 shows the reference trajectories, while the obtained results are reported in Figs. 2–3.

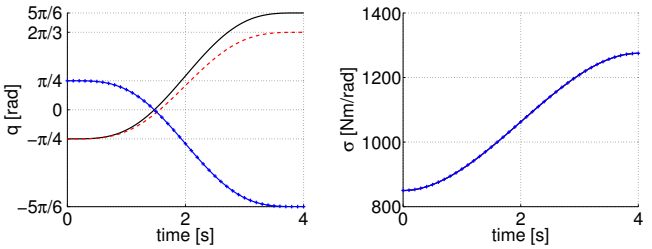


Fig. 1. [Left] Joint position $\boldsymbol{q}_d$. Color codes for Figs. 1–3: solid black = joint 1, dashed red = 2, blue with "+" markers = 3. [Right] Joint stiffness $\boldsymbol{\sigma}_d$ (same profile for all joints).

We performed two other tests with the same joint trajectory $\boldsymbol{q}_d$ shown in Fig. 1, but one with $\sigma_i$ identically equal to $\sigma_{min}$ for all $i$, and a second with $\sigma_i$ identically equal to $\sigma_{max}$. Since the joint movement is the same, the elastic torque $\boldsymbol{\tau}_e$ does not change with respect to Fig. 3. however, all the motor torques $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ have much larger absolute values in the second case. This shows how the motors spend considerable energy in order to keep a higher level of stiffness. In Fig. 4, we compare the desired nominal torques just for joint 1. It can be seen how the initial values of $\tau_{d1,1}$ and $\tau_{d2,1}$ for $\sigma_{di} = \sigma_{min}$ $(i = 1, \ldots, 7)$ coincide with the initial values in Fig. 2, while their final values for $\sigma_{di} = \sigma_{max}$ coincide the final values in the same figures.

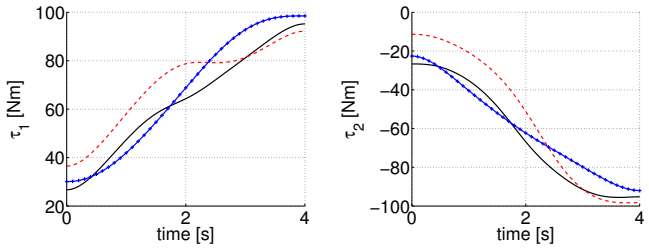


Fig. 2. [Left] Motor torque $\boldsymbol{\tau}_{d1}$. [Right] Motor torque $\boldsymbol{\tau}_{d2}$.

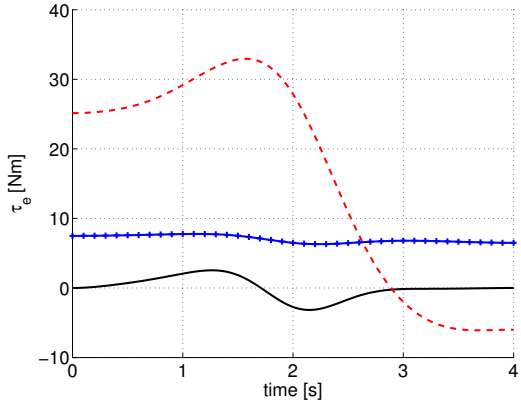


Fig. 3. Total elastic torque $\boldsymbol{\tau}_e = \boldsymbol{\tau}_{e1} + \boldsymbol{\tau}_{e2}$.

The algorithm was implemented in Matlab, automatically converted to C code, and then run as a `mex` function on a standard personal computer. The execution times were always in the order of $10^{-5} \div 10^{-4}$ s per iteration. Very similar times have been obtained also for more complex robots such as a 7-dof manipulator with the kinematics of a KUKA LWR arm. All source codes are available in the multimedia material accompanying this paper.

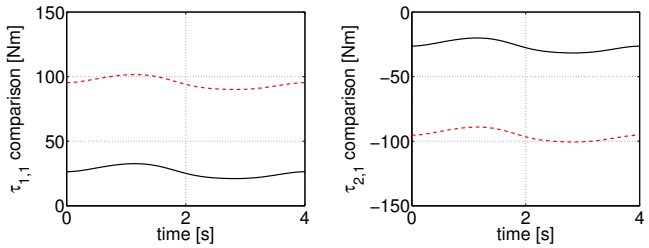


Fig. 4. Comparison between the two motor torques $\tau_{d1,1}$ and $\tau_{d2,1}$ for joint 1 with $\sigma_{d1} = 850$ Nm/rad (solid black lines) and $\sigma_{d1} = 1275$ Nm/rad (dashed red lines).

## VI. Conclusions

An efficient numerical algorithm has been presented for solving the inverse dynamics problem in a class of robots with Variable Stiffness Actuation, as a generalization of the one recently proposed by us for robots with elastic joints.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2016.2526072, IEEE Robotics and Automation Letters



The general procedure is conceptually divided in two distinct parts. The first one, which is the same as in robots with elastic joints, is an extension of the recursive Newton-Euler algorithm for rigid robots. Recursive equations are defined up to the fourth differential order for link motion variables, and up to the second time derivative for forces/torques. The core of the second part of the algorithm requires solving for each joint a two-by-two nonlinear system in terms of the desired stiffness and computed total transmission torque. This system depends on the specific technology of the VSA device and its solution, which is in general obtained numerically by a root finding method, has been illustrated here for antagonistic transmissions with cubic torque profiles. The overall complexity of the algorithm grows linearly with the number of VSA joints.

A simple variant of the algorithm allows to compute also the feedback linearization control law for the considered VSA-based robots. Using full state measurements, this variant does not require a numerical root finding method and can be easily implemented as well in real time.

Just as their symbolic counterparts, the proposed numerical algorithms are based on the knowledge of the robot dynamic model. Thus, the quality of the results will depend on the accuracy of the kinematic and dynamic parameters, in particular for what concerns the nonlinear spring model. Future work will focus on sensitivity analysis to perturbed dynamic parameters and on the extension to Variable Impedance Actuators (VIA), including variable damping.

## APPENDIX: DERIVATION OF THE ALGORITHM

Equations (36), (37), (40), (41), (46), (49), (52), (55) and (58) are simply the standard NEA, applied to elastic robots. Equations (38), (42), (43), (47), (50), (53), (56) and (59) first appeared in [17]; derivation of these formulas can be found there. Thus, the only proofs shown here will be those of (39), (44), (45), (48), (51), (54), (57), and (60)–(74), along with some little intermediate results from [17].

### A. Angular snap

Derivation of (39) can start from the expression of (38), referred to base coordinates:

$$\begin{aligned}\boldsymbol{\iota}_i =& \boldsymbol{\iota}_{i-1} + \dddot{\theta}_i\hat{\boldsymbol{z}}_{i-1} + 2\ddot{\theta}_i(\boldsymbol{\omega}_{i-1}\times\hat{\boldsymbol{z}}_{i-1})\\ &+ \dot{\theta}_i\boldsymbol{\gamma}_{i-1}\times\hat{\boldsymbol{z}}_{i-1} + \dot{\theta}_i\boldsymbol{\omega}_{i-1}\times(\boldsymbol{\omega}_{i-1}\times\hat{\boldsymbol{z}}_{i-1}).\end{aligned} \tag{75}$$

Differentiating it with respect to time, we obtain:

$$\begin{aligned}\boldsymbol{\varsigma}_i =& \boldsymbol{\varsigma}_{i-1} + \ddddot{\theta}_i\hat{\boldsymbol{z}}_{i-1} + \dddot{\theta}_i\dot{\hat{\boldsymbol{z}}}_{i-1} + 2\dddot{\theta}_i(\boldsymbol{\omega}_{i-1}\times\hat{\boldsymbol{z}}_{i-1})\\ &+ 2\ddot{\theta}_i(\boldsymbol{\gamma}_{i-1}\times\hat{\boldsymbol{z}}_{i-1}) + 2\ddot{\theta}_i(\boldsymbol{\omega}_{i-1}\times\dot{\hat{\boldsymbol{z}}}_{i-1})\\ &+ \ddot{\theta}_i(\boldsymbol{\gamma}_{i-1}\times\hat{\boldsymbol{z}}_{i-1}) + \dot{\theta}_i(\boldsymbol{\iota}_{i-1}\times\hat{\boldsymbol{z}}_{i-1}) + \dot{\theta}_i(\boldsymbol{\gamma}_{i-1}\times\dot{\hat{\boldsymbol{z}}}_{i-1})\\ &+ \ddot{\theta}_i\boldsymbol{\omega}_{i-1}\times(\boldsymbol{\omega}_{i-1}\times\hat{\boldsymbol{z}}_{i-1}) + \dot{\theta}_i\boldsymbol{\gamma}_{i-1}\times(\boldsymbol{\omega}_{i-1}\times\hat{\boldsymbol{z}}_{i-1})\\ &+ \dot{\theta}_i\boldsymbol{\omega}_{i-1}\times(\boldsymbol{\gamma}_{i-1}\times\hat{\boldsymbol{z}}_{i-1} + \boldsymbol{\omega}_{i-1}\times\dot{\hat{\boldsymbol{z}}}_{i-1}).\end{aligned} \tag{76}$$

Considering that

$$\begin{aligned}\dot{\hat{\boldsymbol{z}}}_{i-1} =& {}^0\dot{\boldsymbol{R}}_{i-1}\hat{\boldsymbol{z}}_0 = \boldsymbol{S}(\boldsymbol{\omega}_{i-1})\,{}^0\boldsymbol{R}_{i-1}\hat{\boldsymbol{z}}_0 = \boldsymbol{S}(\boldsymbol{\omega}_{i-1})\hat{\boldsymbol{z}}_{i-1}\\ =& \boldsymbol{\omega}_{i-1}\times\hat{\boldsymbol{z}}_{i-1},\end{aligned} \tag{77}$$

we can rewrite the previous equation in the following form

$$\begin{aligned}\boldsymbol{\varsigma}_i =& \boldsymbol{\varsigma}_{i-1} + \dddot{\theta}_i\hat{\boldsymbol{z}}_{i-1} + 3\ddot{\theta}_i\boldsymbol{\omega}_{i-1}\times\hat{\boldsymbol{z}}_{i-1} + 3\ddot{\theta}_i\boldsymbol{\gamma}_{i-1}\times\hat{\boldsymbol{z}}_{i-1}\\ &+ 3\ddot{\theta}_i\boldsymbol{\omega}_{i-1}\times(\boldsymbol{\omega}_{i-1}\times\hat{\boldsymbol{z}}_{i-1}) + \dot{\theta}_i(\boldsymbol{\iota}_{i-1}\times\hat{\boldsymbol{z}}_{i-1})\\ &+ \dot{\theta}_i\boldsymbol{\omega}_{i-1}\times[\boldsymbol{\gamma}_{i-1}\times\hat{\boldsymbol{z}}_{i-1} + \boldsymbol{\omega}_{i-1}\times(\boldsymbol{\omega}_{i-1}\times\hat{\boldsymbol{z}}_{i-1})]\\ &+ 2\dot{\theta}_i\boldsymbol{\gamma}_{i-1}\times(\boldsymbol{\omega}_{i-1}\times\hat{\boldsymbol{z}}_{i-1}).\end{aligned} \tag{78}$$

Describing the preceding equation with respect to the frame $i$ and rearranging terms, we obtain (39).

### B. Linear snap

In the same way as done for $\boldsymbol{\varsigma}$, derivation of (44) can start from the base frame form of $\boldsymbol{\jmath}$:

$$\begin{aligned}\boldsymbol{\jmath}_i =& \boldsymbol{\jmath}_{i-1} + \boldsymbol{\iota}_i\times\boldsymbol{p}_{i,i\text{-}1} + 2\boldsymbol{\gamma}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1})\\ &+ \boldsymbol{\omega}_i\times[\boldsymbol{\gamma}_i\times\boldsymbol{p}_{i,i\text{-}1} + \boldsymbol{\omega}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1})] + \dddot{d}_i\hat{\boldsymbol{z}}_{i-1}\\ &+ 3\ddot{d}_i\boldsymbol{\omega}_i\times\hat{\boldsymbol{z}}_{i-1} + 3\dot{d}_i\boldsymbol{\gamma}_i\times\hat{\boldsymbol{z}}_{i-1} + 3\dot{d}_i\boldsymbol{\omega}_i\times(\boldsymbol{\omega}_i\times\hat{\boldsymbol{z}}_{i-1}).\end{aligned} \tag{79}$$

Differentiating, we have

$$\begin{aligned}\boldsymbol{s}_i =& \boldsymbol{s}_{i-1} + \boldsymbol{\varsigma}_i\times\boldsymbol{p}_{i,i\text{-}1} + \boldsymbol{\iota}_i\times\dot{\boldsymbol{p}}_{i,i\text{-}1} + 2\boldsymbol{\iota}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1})\\ &+ 2\boldsymbol{\gamma}_i\times(\boldsymbol{\gamma}_i\times\boldsymbol{p}_{i,i\text{-}1}) + 2\boldsymbol{\gamma}_i\times(\boldsymbol{\omega}_i\times\dot{\boldsymbol{p}}_{i,i\text{-}1})\\ &+ \boldsymbol{\gamma}_i\times[\boldsymbol{\gamma}_i\times\boldsymbol{p}_{i,i\text{-}1} + \boldsymbol{\omega}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1})]\\ &+ \boldsymbol{\omega}_i\times[\boldsymbol{\iota}_i\times\boldsymbol{p}_{i,i\text{-}1} + \boldsymbol{\gamma}_i\times\dot{\boldsymbol{p}}_{i,i\text{-}1} + \boldsymbol{\gamma}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1})\\ &+ \boldsymbol{\omega}_i\times(\boldsymbol{\gamma}_i\times\boldsymbol{p}_{i,i\text{-}1} + \boldsymbol{\omega}_i\times\dot{\boldsymbol{p}}_{i,i\text{-}1})] + \ddddot{d}_i\hat{\boldsymbol{z}}_{i-1}\\ &+ \dddot{d}_i\dot{\hat{\boldsymbol{z}}}_{i-1} + 3\dddot{d}_i\boldsymbol{\omega}_i\times\hat{\boldsymbol{z}}_{i-1} + 6\ddot{d}_i\boldsymbol{\gamma}_i\times\hat{\boldsymbol{z}}_{i-1}\\ &+ 3\ddot{d}_i\boldsymbol{\omega}_i\times\dot{\hat{\boldsymbol{z}}}_{i-1} + 3\dot{d}_i\boldsymbol{\iota}_i\times\hat{\boldsymbol{z}}_{i-1} + 3\dot{d}_i\boldsymbol{\gamma}_i\times\dot{\hat{\boldsymbol{z}}}_{i-1}\\ &+ 3\ddot{d}_i\boldsymbol{\omega}_i\times(\boldsymbol{\omega}_i\times\hat{\boldsymbol{z}}_{i-1}) + 3\dot{d}_i\boldsymbol{\gamma}_i\times(\boldsymbol{\omega}_i\times\hat{\boldsymbol{z}}_{i-1})\\ &+ 3\dot{d}_i\boldsymbol{\omega}_i\times(\boldsymbol{\gamma}_i\times\hat{\boldsymbol{z}}_{i-1} + \boldsymbol{\omega}_i\times\dot{\hat{\boldsymbol{z}}}_{i-1}).\end{aligned} \tag{80}$$

From (77) and (36), it is possible to rewrite $\dot{\hat{\boldsymbol{z}}}_{i-1}$ as

$$\dot{\hat{\boldsymbol{z}}}_{i-1} = \boldsymbol{\omega}_{i-1}\times\hat{\boldsymbol{z}}_{i-1} = (\boldsymbol{\omega}_i - \dot{\theta}_i\hat{\boldsymbol{z}}_{i-1})\times\hat{\boldsymbol{z}}_{i-1} = \boldsymbol{\omega}_i\times\hat{\boldsymbol{z}}_{i-1}. \tag{81}$$

Moreover, from geometrical considerations,

$$\dot{\boldsymbol{p}}_{i,i\text{-}1} = \boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1} + \dot{d}_i\hat{\boldsymbol{z}}_{i-1}. \tag{82}$$

Substituting these and after some manipulations, we get

$$\begin{aligned}\boldsymbol{s}_i =& \boldsymbol{s}_{i-1} + \boldsymbol{\varsigma}_i\times\boldsymbol{p}_{i,i\text{-}1} + \boldsymbol{\iota}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1} + \dot{d}_i\hat{\boldsymbol{z}}_{i-1})\\ &+ 2\boldsymbol{\iota}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1}) + 3\boldsymbol{\gamma}_i\times(\boldsymbol{\gamma}_i\times\boldsymbol{p}_{i,i\text{-}1})\\ &+ 2\boldsymbol{\gamma}_i\times[\boldsymbol{\omega}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1} + \dot{d}_i\hat{\boldsymbol{z}}_{i-1})]\\ &+ \boldsymbol{\gamma}_i\times[\boldsymbol{\omega}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1})] + \boldsymbol{\omega}_i\times(\boldsymbol{\iota}_i\times\boldsymbol{p}_{i,i\text{-}1})\\ &+ \boldsymbol{\omega}_i\times[\boldsymbol{\gamma}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1} + \dot{d}_i\hat{\boldsymbol{z}}_{i-1})]\\ &+ \boldsymbol{\omega}_i\times[\boldsymbol{\gamma}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1})] + \boldsymbol{\omega}_i\times\{\boldsymbol{\omega}_i\times[\boldsymbol{\gamma}_i\times\boldsymbol{p}_{i,i\text{-}1}\\ &+ \boldsymbol{\omega}_i\times(\boldsymbol{\omega}_i\times\boldsymbol{p}_{i,i\text{-}1} + \dot{d}_i\hat{\boldsymbol{z}}_{i-1})]\} + \dddot{d}_i\hat{\boldsymbol{z}}_{i-1}\\ &+ 4\ddot{d}_i\boldsymbol{\omega}_i\times\hat{\boldsymbol{z}}_{i-1} + 6\ddot{d}_i[\boldsymbol{\gamma}_i\times\hat{\boldsymbol{z}}_{i-1} + \boldsymbol{\omega}_i\times(\boldsymbol{\omega}_i\times\hat{\boldsymbol{z}}_{i-1})]\\ &+ 3\dot{d}_i\boldsymbol{\iota}_i\times\hat{\boldsymbol{z}}_{i-1} + 6\dot{d}_i\boldsymbol{\gamma}_i\times(\boldsymbol{\omega}_i\times\hat{\boldsymbol{z}}_{i-1})\\ &+ 3\dot{d}_i\boldsymbol{\omega}_i\times(\boldsymbol{\gamma}_i\times\hat{\boldsymbol{z}}_{i-1}) + 3\dot{d}_i\boldsymbol{\omega}_i\times[\boldsymbol{\omega}_i\times(\boldsymbol{\omega}_i\times\hat{\boldsymbol{z}}_{i-1})].\end{aligned} \tag{83}$$

Rearranging the terms, collecting similar expressions, and referring $\boldsymbol{s}$ to the base frame, we obtain (44).

The linear snap of the CoM, i.e., $\boldsymbol{s}_{c_i}$, can be computed from $\boldsymbol{s}_i$. Equation (45) can be obtained from (44), by simply letting $i$ instead of $i-1$, $c_i$ instead of $i$, and considering that $\dot{d}_{c_i}$, $\ddot{d}_{c_i}$, $\dddot{d}_{c_i}$, and $\ddddot{d}_{c_i}$ are all equal to zero.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2016.2526072, IEEE Robotics and Automation Letters



### C. Force and torque derivatives on the centers of mass

Equation (48) is trivial, while (51) is obtained from

$$\begin{aligned}\dot{\boldsymbol{N}}_i &= \boldsymbol{\omega}_i \times \boldsymbol{I}_i \boldsymbol{\gamma}_i + \boldsymbol{\omega}_i \times \boldsymbol{N}_i + \boldsymbol{I}_i[\boldsymbol{\gamma}_i \times \boldsymbol{\omega}_i] \\ &\quad + \boldsymbol{I}_i \boldsymbol{\iota}_i + \boldsymbol{\gamma}_i \times \boldsymbol{I}_i \boldsymbol{\omega}_i \end{aligned} \tag{84}$$

$$\boldsymbol{N}_i = \frac{d}{dt}(\boldsymbol{I}_i \boldsymbol{\omega}_i) \tag{85}$$

$$\dot{\boldsymbol{I}}_i = \boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{I}_i - \boldsymbol{I}_i \boldsymbol{S}(\boldsymbol{\omega}_i), \tag{86}$$

where $\boldsymbol{S}(\boldsymbol{\omega}_i)$ is a skew-symmetric matrix computed from $\boldsymbol{\omega}_i$ such that $\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{v} = \boldsymbol{\omega}_i \times \boldsymbol{v}$ for every vector $\boldsymbol{v}$. Equation (85) is Euler's law of motion, while (86) is obtained from $\boldsymbol{I}_i = {}^0\boldsymbol{R}_i \, {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{R}_0$, using the result ${}^0\dot{\boldsymbol{R}}_i = \boldsymbol{S}(\boldsymbol{\omega}_i){}^0\boldsymbol{R}_i$. Then:

$$\begin{aligned}{}^i\ddot{\boldsymbol{N}}_i &= {}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\gamma}_i + {}^i\boldsymbol{\omega}_i \times [\boldsymbol{S}({}^i\boldsymbol{\omega}_i) \, {}^i\boldsymbol{I}_i - {}^i\boldsymbol{I}_i \boldsymbol{S}({}^i\boldsymbol{\omega}_i)] \, {}^i\boldsymbol{\gamma}_i \\ &\quad + {}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\iota}_i + [\boldsymbol{S}({}^i\boldsymbol{\omega}_i) \, {}^i\boldsymbol{I}_i]({}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{\omega}_i) \\ &\quad - [{}^i\boldsymbol{I}_i \boldsymbol{S}({}^i\boldsymbol{\omega}_i)]({}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{\omega}_i) + {}^i\boldsymbol{I}_i({}^i\boldsymbol{\iota}_i \times {}^i\boldsymbol{\omega}_i) \\ &\quad + [\boldsymbol{S}({}^i\boldsymbol{\omega}_i) \, {}^i\boldsymbol{I}_i - {}^i\boldsymbol{I}_i \boldsymbol{S}({}^i\boldsymbol{\omega}_i)] \, {}^i\boldsymbol{\iota}_i + {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\varsigma}_i \\ &\quad + {}^i\boldsymbol{\iota}_i \times {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\omega}_i + 2({}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{N}_i) + {}^i\boldsymbol{\omega}_i \times {}^i\dot{\boldsymbol{N}}_i \\ &= {}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\gamma}_i + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\gamma}_i) \\ &\quad - {}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{I}_i({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{\gamma}_i) + {}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\iota}_i \\ &\quad + {}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{I}_i({}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{\omega}_i) - {}^i\boldsymbol{I}_i[{}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{\omega}_i)] \\ &\quad + {}^i\boldsymbol{I}_i({}^i\boldsymbol{\iota}_i \times {}^i\boldsymbol{\omega}_i) + {}^i\boldsymbol{\omega} \times {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\iota}_i - {}^i\boldsymbol{I}_i({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{\iota}_i) \\ &\quad + {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\varsigma}_i + {}^i\boldsymbol{\iota}_i \times {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\omega}_i + 2({}^i\boldsymbol{\gamma}_i \times {}^i\boldsymbol{N}_i) + {}^i\boldsymbol{\omega}_i \times {}^i\dot{\boldsymbol{N}}_i. \end{aligned} \tag{87}$$

The final expression is obtained by collecting similar terms.

### D. Generalized torques

Again, (54) is trivial, while (57) is computed from

$$\dot{\boldsymbol{p}}_{c_i,i} = {}^0\dot{\boldsymbol{R}}_i \, {}^i\boldsymbol{p}_{c_i,i} = \boldsymbol{S}(\boldsymbol{\omega}_i){}^0\boldsymbol{R}_i \, {}^i\boldsymbol{p}_{c_i,i} = \boldsymbol{\omega}_i \times \boldsymbol{p}_{c_i,i} \tag{88}$$

$$\begin{aligned}\dot{\boldsymbol{n}}_i &= \dot{\boldsymbol{n}}_{i+1} + (\boldsymbol{\omega}_i \times \boldsymbol{p}_{c_i,i}) \times \boldsymbol{F}_i + \boldsymbol{p}_{c_i,i} \times \dot{\boldsymbol{F}}_i \\ &\quad + (\boldsymbol{\omega}_i \times \boldsymbol{p}_{i,i\text{-}1} + \dot{d}_i \hat{\boldsymbol{z}}_{i-1}) \times \boldsymbol{f}_i + \boldsymbol{p}_{i,i\text{-}1} \times \dot{\boldsymbol{f}}_i + \dot{\boldsymbol{N}}_i. \end{aligned} \tag{89}$$

Differentiating (89), with the help of (82) and (88), yields

$$\begin{aligned}\ddot{\boldsymbol{n}}_i &= \ddot{\boldsymbol{n}}_{i+1} + [\boldsymbol{\gamma}_i \times \boldsymbol{p}_{c_i,i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{p}_{c_i,i})] \times \boldsymbol{F}_i \\ &\quad + (\boldsymbol{\omega}_i \times \boldsymbol{p}_{c_i,i}) \times \dot{\boldsymbol{F}}_i + (\boldsymbol{\omega}_i \times \boldsymbol{p}_{c_i,i}) \times \dot{\boldsymbol{F}}_i + \boldsymbol{p}_{c_i,i} \times \ddot{\boldsymbol{F}}_i \\ &\quad + [\boldsymbol{\gamma}_i \times \boldsymbol{p}_{i,i\text{-}1} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{p}_{i,i\text{-}1} + \dot{d}_i \hat{\boldsymbol{z}}_{i-1}) + \ddot{d}_i \hat{\boldsymbol{z}}_{i-1} \\ &\quad + \dot{d}_i \boldsymbol{\omega}_i \times \hat{\boldsymbol{z}}_{i-1}] \times \boldsymbol{f}_i + (\boldsymbol{\omega}_i \times \boldsymbol{p}_{i,i\text{-}1} + \dot{d}_i \hat{\boldsymbol{z}}_{i-1}) \times \dot{\boldsymbol{f}}_i \\ &\quad + (\boldsymbol{\omega}_i \times \boldsymbol{p}_{i,i\text{-}1} + \dot{d}_i \hat{\boldsymbol{z}}_{i-1}) \times \dot{\boldsymbol{f}}_i + \boldsymbol{p}_{i,i\text{-}1} \times \ddot{\boldsymbol{f}}_i + \ddot{\boldsymbol{N}}_i, \end{aligned} \tag{90}$$

from which it is relatively easy to obtain (57). Finally, from the base-frame form of (59) and using (81)

$$\begin{aligned}\ddot{\tau}_{e,\,i} &= (\ddot{\boldsymbol{n}}_i + \dot{\boldsymbol{n}}_i \times \boldsymbol{\omega}_i + \boldsymbol{n}_i \times \boldsymbol{\gamma}_i)^T \, \hat{\boldsymbol{z}}_{i-1} \\ &\quad + (\dot{\boldsymbol{n}}_i + \boldsymbol{n}_i \times \boldsymbol{\omega}_i)^T (\boldsymbol{\omega}_i \times \hat{\boldsymbol{z}}_{i-1}) + D_i \ddot{\boldsymbol{q}}_i \\ &= (\ddot{\boldsymbol{n}}_i + \dot{\boldsymbol{n}}_i \times \boldsymbol{\omega}_i + \boldsymbol{n}_i \times \boldsymbol{\gamma}_i)^T \, \hat{\boldsymbol{z}}_{i-1} \\ &\quad + [\dot{\boldsymbol{n}}_i \times \boldsymbol{\omega}_i + (\boldsymbol{n}_i \times \boldsymbol{\omega}_i) \times \boldsymbol{\omega}_i]^T \, \hat{\boldsymbol{z}}_{i-1} + D_i \ddot{\boldsymbol{q}}_i \end{aligned} \tag{91}$$

where the well-known property of the scalar triple product $\boldsymbol{a}^T(\boldsymbol{b} \times \boldsymbol{c}) = (\boldsymbol{a} \times \boldsymbol{b})^T \boldsymbol{c}$ has been used. Collecting for $\hat{\boldsymbol{z}}_{i-1}$ and expressing all quantities into frame $i$ will return (60). This holds for revolute joints; for prismatic joints, the proof is almost identical and is omitted.

Equations (62)–(63) are simply the model-based computations of the spring torques and stiffnesses for joint $i$, while (64)–(66) are the joint-decoupled versions of equations (15) and (16). Equation (67) gives the closed-form expression of the $i$-th diagonal block of the rearranged matrix $\boldsymbol{\mathcal{A}}$, defined in (23). Equations (68) solves (19) for joint $i$. Equations (70)–(74) are the joint-decoupled versions of (21), (22), (20), (5b) and (5c), in this order.

## REFERENCES


[1] S. Haddadin, "Optimal exploitation of soft-robot dynamics," in *Soft Robotics*, A. Verl, A. Albu-Schäffer, O. Brock, and A. Raatz, Eds. Springer, 2015, pp. 92–99.

[2] S. Haddadin and E. Croft, "Physical human-robot interaction," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2016 (in press).

[3] A. De Luca and W. Book, "Robots with flexible elements," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 287–319.

[4] B. Vanderborght *et al.*, "Variable impedance actuators: A review," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1601–1614, 2013.

[5] R. Schiavi, G. Grioli, S. Sen, and A. Bicchi, "VSA-II: A novel prototype of variable stiffness actuator for safe and performing robots interacting with humans," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 2171–2176.

[6] R. Van Ham, B. Vanderborght, M. Van Damme, B. V. B, and D. Lefeber, "MACCEPA, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot dynamic computer simulation of robotic mechanisms," *Robotics and Autonomous Systems*, vol. 55, no. 10, pp. 761–768, 2007.

[7] S. Wolf, O. Eiberger, and G. Hirzinger, "The DLR FSJ: Energy based design of a variable stiffness joint," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 5082–5089.

[8] A. Jafari, N. Tsagarakis, B. Vanderborght, and D. Caldwell, "AwAS-II: A new actuator with adjustable stiffness based on the novel principle of adaptable pivot point and variable lever ratio," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 4638–4643.

[9] Markus Grebenstein *et al*, "The DLR Hand Arm System," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 3175–3182.

[10] N. Kashiri, M. Laffranchi, N. Tsagarakis, I. Sardellitti, and D. Caldwell, "Dynamic modeling and adaptable control of the CompAct arm," in *Proc. IEEE Int. Conf. on Mechatronics*, 2013, pp. 477–482.

[11] [Online]. Available: http://www.naturalmotioninitiative.com

[12] G. Palli, C. Melchiorri, and A. De Luca, "On the feedback linearization of robots with variable joint stiffness," in *Proc. IEEE Int. Conference on Robotics and Automation*, 2008, pp. 1753–1759.

[13] A. De Luca, F. Flacco, A. Bicchi, and R. Schiavi, "Nonlinear decoupled motion-stiffness control and collision detection/reaction for the VSA-II variable stiffness device," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 5487–5494.

[14] M. Spong, "Modeling and control of elastic joint robots," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 109, no. 4, pp. 310–319, 1987.

[15] G. Buondonno and A. De Luca, "A recursive Newton-Euler algorithm for robots with elastic joints and its application to control," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 5526–5532.

[16] J. Y. Luh, M. W. Walker, and R. P. Paul, "On-line computational scheme for mechanical manipulators," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 102, no. 2, pp. 69–76, 1980.

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