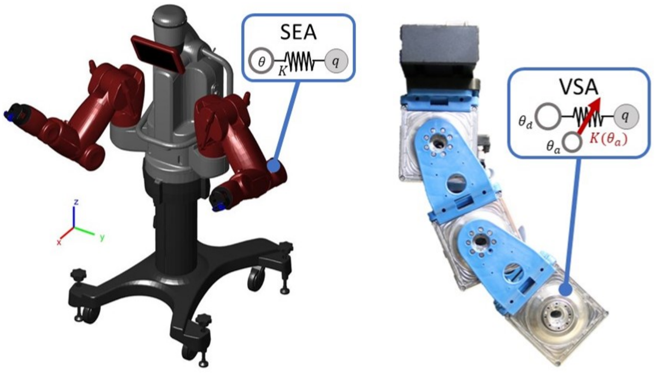
Time Generalization of Trajectories Learned on Articulated Soft Robots

Franco Angelini[1,2,3], Riccardo Mengacci [1,3], Cosimo Della Santina[4],
Manuel G. Catalano[2], Manolo Garabini[1,3], Antonio Bicchi[1,2,3] and Giorgio Grioli[2]

***Abstract*— To avoid feedback-related stiffening of articulated soft robots, a substantive feedforward contribution is crucial. However, obtaining reliable feedforward actions requires very accurate models, which are not always available for soft robots. Learning-based approaches are a promising solution to the problem. They proved to be an effective strategy achieving good tracking performance, while preserving the system intrinsic compliance. Nevertheless, learning methods require rich data sets, and issues of scalability and generalization still remain to be solved. This paper proposes a method to generalize learned control actions to execute a desired trajectory with different velocities - with the ultimate goal of making these learning-based architectures sample efficient. More specifically we prove that the knowledge of how to execute a same trajectory at five different speeds is necessary and sufficient to execute the same trajectory at any velocity - without any knowledge of the model. We also give a simple constructive way to calculate this new feedforward action. The effectiveness of the proposed technique is validated in extensive simulation on a Baxter robot with soft springs playing a drum, and experimentally on a VSA double pendulum performing swinging motions.**



(a) Baxter SEA robot (b) VSA double pendulum

Fig. 1. Examples of articulated soft robots on which the proposed method can be applied, and that we use as validation testbeds. (a) Simulation setup: Baxter robot actuated by SEA. (b) Experimental setup: double pendulum actuated by VSA (qbMoves Advanced). The gravity vector points downward.

I. INTRODUCTION

Animals exploit the compliant characteristics of their body to execute a wide variety of tasks with high performance and precision [1]. Inspired by the effectiveness of the natural example, researchers started including carefully designed elastic components into robots, leading to the so-called soft robotics. In this work, we focus on the vertebrate-inspired branch of this field, namely articulated soft robotics. Mimicking vertebrate animals, elasticity is here lumped at the joint level [2]. Two classic examples of articulated soft actuators are Series Elastic Actuators (SEA) [3] (Fig. 1(a)) and Variable Stiffness Actuators (VSA) [4] (Fig. 1(b)).

The advent of these novel technologies has opened up new exciting research questions concerning their control. For example, it has been recently recognized that classic control algorithms that rely on (high-gain) feedback loops lead to an actual stiffening of the robots structure [5], hindering their deliberately compliant design. Feedforward approaches have been proposed as a valuable solution to this problem, allowing to achieve good tracking performance, while avoiding any alteration of the system mechanical compliance.

In [6] is presented a method that optimizes torque and stiffness profile for explosive movement tasks in robot actuated by VSA. In [7] the authors propose a time-optimal control framework for VSA robots. Both methods mostly rely on feedfoward actions. However, model-based feedforward techniques may be not reliable due to difficulties in obtaining sufficiently reliable models for articulated soft robots.

Conversely, model-free feedfoward methods are able to achieve good tracking performance through learning and repetitions, without requiring any exact system representation. In [8], a decentralized control algorithm based on Iterative Learning Control (ILC) is proposed. This method merges a mostly feedforward control action with a low gain feedback term to obtain satisfactory trajectory execution. A norm optimal ILC is employed in [9], where is proposed a control algorithm for a soft robotic arm during aggressive maneuvers. In [10] the authors exploit Gaussian process regression to obtain a model able to generate feedforward inputs to control a soft robot, limiting the feedback action. In [11] is proposed a solution based on recurrent neural network for dynamic tasks with flexible joint robots. In [12], an ILC algorithm is used for performing complex reaching motions with a continuum soft robot.

Nevertheless, learning based approaches generally present issues in terms of scalability and generalization of the learned control actions. For instance, a learned input able to perfectly execute a specific task, could be completely unsuitable to perform slight variations of that same task, leading to a whole new learning process. Examples of potential task variations are the task end-point (i.e. spatial reference trajectory), task velocity execution and carried payload. On the contrary,

This research has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 732737 (ILIAD), No. 780883 (THING) and No. 840446 (SoftHandler).

[1] Centro di Ricerca "Enrico Piaggio", Università di Pisa, Largo Lucio Lazzarino 1, 56126 Pisa, Italy

[2] Soft Robotics for Human Cooperation and Rehabilitation, Fondazione Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy

[3] Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Largo Lucio Lazzarino 1, 56126 Pisa, Italy

[4] Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar St., Cambridge, MA 02139, USA

frncangelini@gmail.com

observations on human beings show that they excel in the generalization of the acquired motor skills [13]. One example among the others is our ability to progressively refine a motor action repeating a novel task at different speed [14]. Our daily life is full of such examples. Think for instance of a dancer or an athlete who learn a specific motion by first executing it multiple times slowly and then progressively increasing the speed of the motion.

Inspired by these natural insights, this paper proposes a method for generalizing learned control actions to different time scales. The time scaling property of robot trajectories has already been studied, especially in the case of rigid robots. In [15] this property is exploited to determine if a planned path is feasible given the system model and actuation torque constraints. This algorithm was then extended in [16] to take into account the joint elasticity of articulated soft robots. There, an uniform time scaling and robots with fixed constant linear joint stiffness are considered. Both works require the full knowledge of the model. The idea of exploiting the time scale transformations to generalize the control actions w.r.t. time was first introduced for rigid robots in [17].

This paper takes several substantial steps further in the direction of time generalization of learned trajectories with articulated soft robots. First, we prove that five sample inputs are necessary and sufficient to generalize the control action to different velocities - when no knowledge of the model is hypothesized. Then, we present a constructive closed form solution to the problem in the case of time varying linear stiffness, and we discuss the extension of the approach to the case of non linear stiffness. We first show the effectiveness of the method in simulation on a Baxter robot powered by low stiffness Series Elastic Actuators (SEA) [3]. Then, we present an experimental validation of the method on a double pendulum actuated by Variable Stiffness Actuators (VSA) [4]. Two configurations are tested; constant stiffness and variable stiffness.

II. PROBLEM STATEMENT

For the sake of clarity, in the following equations we will use round brackets to refer to functional dependency and curly brackets to indicate term grouping. Square brackets will instead define vectors and matrices.

A. Model

The dynamic model of an articulated soft robot with N joints, with serial or tree topology is

$$M(q(t))\ddot{q}(t) + C(q(t),\dot{q}(t))\dot{q}(t) + F(q(t))\dot{q}(t) + \\ + G(q(t)) + K(t)\left\{q(t) - \theta(t)\right\} = 0, \tag{1}$$

$$B\ddot{\theta}(t) + D\dot{\theta}(t) - K(t)\left\{q(t) - \theta(t)\right\} = \tau(t)\ , \tag{2}$$

where $q \in \mathbb{R}^N$ are the joint angles, with their time derivatives $\dot{q}, \ddot{q}, \dddot{q}, \ddddot{q}$, and $\theta \in \mathbb{R}^N$ are the motor angles, with their time derivatives $\dot{\theta}, \ddot{\theta}$. Dependencies on time t are stressed in (1)-(2). $M(q) \in \mathbb{R}^{N\times N}$ is the inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{N\times N}$ collects Coriolis and centrifugal forces, $F(q) \in \mathbb{R}^{N\times N}$ is a configuration dependent damping matrix. $G(q) \triangleq G_{\mathrm{G}}(q) - J^{\mathrm{T}}(q)F_{\mathrm{ext}}(q) \in \mathbb{R}^N$ is the potential force field, where $G_{\mathrm{G}}(q)$ is the gravity force, $J(q)$ is the robot jacobian matrix, and $F_{\mathrm{ext}}(q) \in \mathbb{R}^6$ is an external force applied at the end-effector. We consider a linear time variant stiffness matrix $K(t) \in \mathbb{R}^{N\times N}$. The case of constant K can be seen as modeling the standard serial elastic actuators (SEA) [3]. The time variance can be regarded as a first order approximation of a variable stiffness characteristics (VSA) [4]. On the motor side, $B \in \mathbb{R}^{N\times N}$ and $D \in \mathbb{R}^{N\times N}$ are the inertia and viscous friction matrices of the motors. Finally, $\tau \in \mathbb{R}^N$ is the actuation torque.

We introduce the following very common hypotheses on the model

H1 K, B, D are here assumed diagonal. These assumptions are made for the sake of space and readability, and they will be relaxed in future extensions of this work.

H2 Model (1) is formulated under the assumption of negligible inertia coupling between link variable q and motor variable θ [18], and non eccentric motor inertia.

B. Time scaling

We describe a change in velocity of the desired joint space trajectory as a linear time scaling. Taken a generic trajectory $q : [0,T] \to \mathbb{R}^N$, an uniform time scale is defined as $q(\alpha t)$, $t \in [0, T/\alpha]$, $\alpha \in \mathbb{R}^+$, where $\alpha \in (0,1)$ leads to slower motion, while $\alpha \in (1,\infty)$ leads to faster motion. Let $\bar{q}$ and $\hat{q}$ be two trajectories defined in $[0,T_1]$ and $[0,T_2]$, respectively. To make the two time intervals coherent will be sufficient to scale the first of $\alpha = \frac{T_2}{T_1}$, or the second of $\frac{T_1}{T_2}$. Therefore the two are one the time scaling of the other if $\bar{q}\left(\frac{T_2}{T_1}t\right) = \hat{q}(t)$, $\forall t$.

C. Goal

Consider a generic reference trajectory in joint space $\hat{q} : [0,T] \to \mathbb{R}^N$, where $T \in \mathbb{R}^+$ is the terminal time. The torque $\tau : [0,T] \to \mathbb{R}^N$ necessary to implement $\hat{q}$ is considered unknown, as well as the knowledge of the model unavailable.

Suppose now that only a set of learned torque-joint evolution pairs $({}^r q, {}^r \tau)$ is available, with

$${}^r q : [0,T_r] \to \mathbb{R}^N, \quad {}^r \tau : [0,T_r] \to \mathbb{R}^N, \tag{3}$$

where the left suffix ${}^r(\cdot)$, $r = 1 \dots w$, refers to the $r-$th pair. $w \in \mathbb{N}$ is the total number of examples. We call ${}^r\beta = \frac{T}{T_r}$ the scaling factor needed to express the $r-$th example in the time scale of q.

We introduce the following working hypotheses

H3 $\hat{q}$ is at least four time differentiable

H4 ${}^r q$ are all time scalings of $\hat{q}$, i.e., ${}^r q({}^r\beta t) = \hat{q}(t)$, $\forall t$

H5 the examples are distinct, i.e., ${}^i\beta \neq {}^j\beta$, $\forall i \neq j$.

The research questions that we aim to solve are

G1 finding which is the minimum number of examples w providing a rich enough information on the task to allow the execution of $\hat{q}$

G2 deriving an algorithm that provides τ given the examples $({}^r q, {}^r \tau)$.

Remark 1. *The examples are going to be the sole and only explicit source of information about the system. We will also know that the system is an articulated soft robot, i.e., it* *is expression of some dynamics in the form* (1)-(2)*, which however we will consider completely unknown.*

Remark 2. *It is easy to see that solving G1 and G2 for $\hat{q}$ implicitly solves the problem for any of its time scalings $\hat{q}(\alpha t)$, $\forall \alpha > 0$. Indeed the only effect of this scaling on the discussed problem is to scale all the ${}^{r}\beta$ of a same factor.*

III. PROPOSED APPROACH

A. Main result

The following theorem answers to G1.

Theorem 1. *Let H1-5 be fulfilled, and the time variance of K be such that*[1]

$$
{}^{r}K({}^{r}\beta t) = K(t), \quad \forall t, r, \tag{4}
$$

where ${}^{r}K$ is the evolution of K within the experiment r.

Then the necessary and sufficient amount w of distinct examples (3) *necessary to analytically express $\tau(t)$ in terms of $({}^{r}\tau(t), {}^{r}q(t))$ without any knowledge of the model is 5.*

Proof. System dynamics (1)-(2) can be rewritten in terms of the Lagrangian link coordinates q and their derivatives $\dot{q}, \ddot{q}, \dddot{q}, \ddddot{q}$. To this end, we first rewrite individually the $2N$ equations of the system as

$$
\begin{aligned}
&\sum_{j=1}^{N} M_{zj}\ddot{q}_j + \sum_{j,l=1}^{N} C_{zjl}\dot{q}_l\dot{q}_j + \sum_{j=1}^{N} F_{zj}\dot{q}_j + \\
&\qquad + G_z + K_z\{q_z - \theta_z\} = 0,
\end{aligned} \tag{5}
$$

$$
B_z\ddot{\theta}_z + D_z\dot{\theta}_z - K_z\{q_z - \theta_z\} = \tau_z \ , \tag{6}
$$

where $z = 1 \dots N$, M_{zj} and F_{zj} are the element (z, j) of the matrix M and F, respectively, C_{zjl} are the Christoffel symbols [19] employed for the evaluation of the matrix C, while G_z is the $z-$th element of G. The matrices K, B and D can be assumed diagonal without loss of generality, and we define their $z-$th diagonal element as K_z, B_z and D_z. Note that we omit the dependency on time and on the Lagrangian coordinates, for the sake of clarity.

Eq. (5) can be solved for θ_z, yielding

$$
\begin{aligned}
\theta_z = q_z + K_z^{-1}\Bigg\{&\sum_{j=1}^{N} M_{zj}\ddot{q}_j + \sum_{j,l=1}^{N} C_{zjl}\dot{q}_l\dot{q}_j + \\
&+ \sum_{j=1}^{N} F_{zj}\dot{q}_j + G_z\Bigg\} \ .
\end{aligned} \tag{7}
$$

Differentiating (5) w.r.t. time and solving for $\dot{\theta}_z$ leads to

$$
\begin{aligned}
\dot{\theta}_z = \dot{q}_z + K_z^{-1}\Bigg\{&\sum_{j=1}^{N} M_{zj}\dddot{q}_j + \sum_{j,h=1}^{N} \frac{\partial M_{zj}}{\partial q_h}\dot{q}_h\ddot{q}_j + \\
&+ 2\sum_{j,l=1}^{N} C_{zjl}\dot{q}_l\ddot{q}_j + \sum_{j,l,h=1}^{N} \frac{\partial C_{zjl}}{\partial q_h}\dot{q}_h\dot{q}_l\dot{q}_j + \\
&+ \sum_{j=1}^{N} F_{zj}\ddot{q}_j + \sum_{j,h=1}^{N} \frac{\partial F_{zj}}{\partial q_h}\dot{q}_h\dot{q}_j + \sum_{h=1}^{N} \frac{\partial G_z}{\partial q_h}\dot{q}_h\Bigg\} + \\
&+ K_z^{-1}\dot{K}_z\{q_z - \theta_z\} \quad ,
\end{aligned} \tag{8}
$$

where θ_z is equal to (7).

Differentiating (5) twice w.r.t. time and solving for $\ddot{\theta}_z$ leads to (9) (see next page), where θ_z and $\dot{\theta}_z$ are equal to (7) and (8).

Substituting (7), (8) and (9) into (6) for each $z = 1 \dots N$, we obtain the whole system dynamics written only in terms of the Lagrangian coordinates q and their derivatives. Let us define ${}^{r}\alpha \triangleq \frac{1}{{}^{r}\beta}$. We can now specify the influence of a time scaling ${}^{r}\beta$ on the system dynamics. In particular we start noticing that

$$
\hat{q}(t) = {}^{r}q({}^{r}\beta t), \tag{10}
$$

$$
\dot{\hat{q}}(t) = {}^{r}\dot{q}({}^{r}\beta t) = {}^{r}\beta\frac{\partial\, {}^{r}q({}^{r}\beta t)}{\partial\, {}^{r}\beta t}, \tag{11}
$$

$$
\ddot{\hat{q}}(t) = {}^{r}\ddot{q}({}^{r}\beta t) = {}^{r}\beta^2\frac{\partial^2\, {}^{r}q({}^{r}\beta t)}{\partial\, ({}^{r}\beta t)^2}, \tag{12}
$$

$$
\dddot{\hat{q}}(t) = {}^{r}\dddot{q}({}^{r}\beta t) = {}^{r}\beta^3\frac{\partial^3\, {}^{r}q({}^{r}\beta t)}{\partial\, ({}^{r}\beta t)^3}, \tag{13}
$$

$$
\ddddot{\hat{q}}(t) = {}^{r}\ddddot{q}({}^{r}\beta t) = {}^{r}\beta^4\frac{\partial^4\, {}^{r}q({}^{r}\beta t)}{\partial\, ({}^{r}\beta t)^4} \ . \tag{14}
$$

Using (4) yields

$$
\hat{K}(t) = {}^{r}K({}^{r}\beta t) \ , \tag{15}
$$

$$
\dot{\hat{K}}(t) = {}^{r}\beta\frac{\partial\, {}^{r}K({}^{r}\beta t)}{\partial\, {}^{r}\beta t} \ , \tag{16}
$$

$$
\ddot{\hat{K}}(t) = {}^{r}\beta^2\frac{\partial^2\, {}^{r}K({}^{r}\beta t)}{\partial\, ({}^{r}\beta t)^2} \ . \tag{17}
$$

We can now write the dynamics of each of the given ${}^{r}q({}^{r}\beta t)$ in terms of the reference trajectory $\hat{q}(t)$, obtaining (18) (reported as floating object for the sake of readability).

Each different learned trajectory ${}^{r}q(t)$, once scaled, presents a dynamic model in the form (18). It is possible to note that each term of the dynamics multiplies a power of the parameter ${}^{r}\alpha$, ranging from ${}^{r}\alpha^0$ to ${}^{r}\alpha^4$. Collecting these terms into four variables $\xi_z^0, \dots \xi_z^4 \in \mathbb{R}$, we can rewrite (18) as a fourth order polynomial of ${}^{r}\alpha$

$$
{}^{r}\tau_z({}^{r}\beta t) = {}^{r}\alpha^4\xi_z^4(t) + {}^{r}\alpha^3\xi_z^3(t) + {}^{r}\alpha^2\xi_z^2(t) + {}^{r}\alpha\xi_z^1(t) + \xi_z^0(t) \ . \tag{19}
$$

Defining $\xi^0 \triangleq [\xi_1^0, \dots \xi_N^0]^{\mathrm{T}}, \dots \xi^4 \triangleq [\xi_1^4, \dots \xi_N^4]^{\mathrm{T}}$, we can write the time scaled dynamics of the whole system in a matrix form

$$
{}^{r}\tau({}^{r}\beta t) = {}^{r}\alpha^4\xi^4(t) + {}^{r}\alpha^3\xi^3(t) + {}^{r}\alpha^2\xi^2(t) + {}^{r}\alpha\xi^1(t) + \xi^0(t) \ . \tag{20}
$$

[1] This condition means that $K(t)$ is time scaled as q.

$$\begin{aligned}\ddot{\theta}_z = \ddot{q}_z + K_z^{-1} \Bigg\{ & \sum_{j=1}^{N} M_{zj}\ddddot{q}_j + \sum_{j,h=1}^{N} \frac{\partial M_{zj}}{\partial q_h}\left\{\ddot{q}_h\ddot{q}_j + 2\dot{q}_h\dddot{q}_j\right\} + \sum_{j,h,p=1}^{N} \frac{\partial^2 M_{zj}}{\partial q_h q_p}\dot{q}_p\dot{q}_h\ddot{q}_j + 2\sum_{j,l=1}^{N} C_{zjl}\left\{\ddot{q}_l\ddot{q}_j + \dot{q}_l\dddot{q}_j\right\} + \\ & + \sum_{j,l,h=1}^{N} \frac{\partial C_{zjl}}{\partial q_h}\left\{\ddot{q}_h\dot{q}_l\dot{q}_j + 4\dot{q}_h\dot{q}_l\ddot{q}_j\right\} + \sum_{j,l,h,p=1}^{N} \frac{\partial^2 C_{zjl}}{\partial q_h q_p}\dot{q}_p\dot{q}_h\dot{q}_l\dot{q}_j + \sum_{j=1}^{N} F_{zj}\dddot{q}_j + \sum_{j,h=1}^{N} \frac{\partial F_{zj}}{\partial q_h}\left\{\ddot{q}_h\dot{q}_j + 2\dot{q}_h\ddot{q}_j\right\} + \\ & + \sum_{j,h,p=1}^{N} \frac{\partial^2 F_{zj}}{\partial q_h q_p}\dot{q}_p\dot{q}_h\dot{q}_j + \sum_{h=1}^{N} \frac{\partial G_z}{\partial q_h}\ddot{q}_h + \sum_{h,p=1}^{N} \frac{\partial^2 G_z}{\partial q_h q_p}\dot{q}_p\dot{q}_h \Bigg\} + 2K_z^{-1}\dot{K}_z\left\{\dot{q}_z - \dot{\theta}_z\right\} + K_z^{-1}\ddot{K}_z\left\{q_z - \theta_z\right\} \quad .\end{aligned} \tag{9}$$

$$\begin{aligned} {}^r\tau_z = {}^r\alpha^2 B_z\ddot{\hat{q}}_z + B_z\hat{K}_z^{-1} \Bigg\{ & {}^r\alpha^4 \sum_{j=1}^{N} M_{zj}\ddddot{\hat{q}}_j + {}^r\alpha^4 \sum_{j,h=1}^{N} \frac{\partial M_{zj}}{\partial \hat{q}_h}\left\{\ddot{\hat{q}}_h\ddot{\hat{q}}_j + 2\dot{\hat{q}}_h\dddot{\hat{q}}_j\right\} + {}^r\alpha^4 \sum_{j,h,p=1}^{N} \frac{\partial^2 M_{zj}}{\partial \hat{q}_h\hat{q}_p}\dot{\hat{q}}_p\dot{\hat{q}}_h\ddot{\hat{q}}_j + \\ & + 2\,{}^r\alpha^4 \sum_{j,l=1}^{N} C_{zjl}\left\{\ddot{\hat{q}}_l\ddot{\hat{q}}_j + \dot{\hat{q}}_l\dddot{\hat{q}}_j\right\} + {}^r\alpha^4 \sum_{j,l,h=1}^{N} \frac{\partial C_{zjl}}{\partial \hat{q}_h}\left\{\ddot{\hat{q}}_h\dot{\hat{q}}_l\dot{\hat{q}}_j + 4\dot{\hat{q}}_h\dot{\hat{q}}_l\ddot{\hat{q}}_j\right\} + {}^r\alpha^4 \sum_{j,l,h,p=1}^{N} \frac{\partial^2 C_{zjl}}{\partial \hat{q}_h\hat{q}_p}\dot{\hat{q}}_p\dot{\hat{q}}_h\dot{\hat{q}}_l\dot{\hat{q}}_j + {}^r\alpha^3 \sum_{j=1}^{N} F_{zj}\dddot{\hat{q}}_j + \\ & + {}^r\alpha^3 \sum_{j,h=1}^{N} \frac{\partial F_{zj}}{\partial \hat{q}_h}\left\{\ddot{\hat{q}}_h\dot{\hat{q}}_j + 2\dot{\hat{q}}_h\ddot{\hat{q}}_j\right\} + {}^r\alpha^3 \sum_{j,h,p=1}^{N} \frac{\partial^2 F_{zj}}{\partial \hat{q}_h\hat{q}_p}\dot{\hat{q}}_p\dot{\hat{q}}_h\dot{\hat{q}}_j + {}^r\alpha^2 \sum_{h=1}^{N} \frac{\partial G_z}{\partial \hat{q}_h}\ddot{\hat{q}}_h + {}^r\alpha^2 \sum_{h,p=1}^{N} \frac{\partial^2 G_z}{\partial \hat{q}_h\hat{q}_p}\dot{\hat{q}}_p\dot{\hat{q}}_h \Bigg\} + \\ & - 2\,{}^r\alpha B_z\hat{K}_z^{-1}\dot{\hat{K}}_z\hat{K}_z^{-1} \Bigg\{ {}^r\alpha^3 \sum_{j=1}^{N} M_{zj}\dddot{\hat{q}}_j + {}^r\alpha^3 \sum_{j,h=1}^{N} \frac{\partial M_{zj}}{\partial \hat{q}_h}\dot{\hat{q}}_h\ddot{\hat{q}}_j + 2\,{}^r\alpha^3 \sum_{j,l=1}^{N} C_{zjl}\dot{\hat{q}}_l\ddot{\hat{q}}_j + {}^r\alpha^3 \sum_{j,l,h=1}^{N} \frac{\partial C_{zjl}}{\partial \hat{q}_h}\dot{\hat{q}}_h\dot{\hat{q}}_l\dot{\hat{q}}_j + \\ & + {}^r\alpha^2 \sum_{j=1}^{N} F_{zj}\ddot{\hat{q}}_j + {}^r\alpha^2 \sum_{j,h=1}^{N} \frac{\partial F_{zj}}{\partial \hat{q}_h}\dot{\hat{q}}_h\dot{\hat{q}}_j + {}^r\alpha \sum_{h=1}^{N} \frac{\partial G_z}{\partial \hat{q}_h}\dot{\hat{q}}_h \Bigg\} - {}^r\alpha^2 B_z\hat{K}_z^{-1}\ddot{\hat{K}}_z\hat{K}_z^{-1} \Bigg\{ {}^r\alpha^2 \sum_{j=1}^{N} M_{zj}\ddot{\hat{q}}_j + {}^r\alpha^2 \sum_{j,l=1}^{N} C_{zjl}\dot{\hat{q}}_l\dot{\hat{q}}_j + \\ & + {}^r\alpha \sum_{j=1}^{N} F_{zj}\dot{\hat{q}}_j + G_z \Bigg\} + 2\,{}^r\alpha^2 B_z\hat{K}_z^{-1}\dot{\hat{K}}_z\hat{K}_z^{-1}\dot{\hat{K}}_z\hat{K}_z^{-1} \Bigg\{ {}^r\alpha^2 \sum_{j=1}^{N} M_{zj}\ddot{\hat{q}}_j + {}^r\alpha^2 \sum_{j,l=1}^{N} C_{zjl}\dot{\hat{q}}_l\dot{\hat{q}}_j + {}^r\alpha \sum_{j=1}^{N} F_{zj}\dot{\hat{q}}_j + G_z \Bigg\} + \\ & + {}^r\alpha D_z\dot{\hat{q}}_z + D_z\hat{K}_z^{-1} \Bigg\{ {}^r\alpha^3 \sum_{j=1}^{N} M_{zj}\dddot{\hat{q}}_j + {}^r\alpha^3 \sum_{j,h=1}^{N} \frac{\partial M_{zj}}{\partial \hat{q}_h}\dot{\hat{q}}_h\ddot{\hat{q}}_j + 2\,{}^r\alpha^3 \sum_{j,l=1}^{N} C_{zjl}\dot{\hat{q}}_l\ddot{\hat{q}}_j + {}^r\alpha^3 \sum_{j,l,h=1}^{N} \frac{\partial C_{zjl}}{\partial \hat{q}_h}\dot{\hat{q}}_h\dot{\hat{q}}_l\dot{\hat{q}}_j + \\ & + {}^r\alpha^2 \sum_{j=1}^{N} F_{zj}\ddot{\hat{q}}_j + {}^r\alpha^2 \sum_{j,h=1}^{N} \frac{\partial F_{zj}}{\partial \hat{q}_h}\dot{\hat{q}}_h\dot{\hat{q}}_j + {}^r\alpha \sum_{h=1}^{N} \frac{\partial G_z}{\partial \hat{q}_h}\dot{\hat{q}}_h \Bigg\} - {}^r\alpha D_z\hat{K}_z^{-1}\dot{\hat{K}}_z\hat{K}_z^{-1} \Bigg\{ {}^r\alpha^2 \sum_{j=1}^{N} M_{zj}\ddot{\hat{q}}_j + {}^r\alpha^2 \sum_{j,l=1}^{N} C_{zjl}\dot{\hat{q}}_l\dot{\hat{q}}_j + \\ & + {}^r\alpha \sum_{j=1}^{N} F_{zj}\dot{\hat{q}}_j + G_z \Bigg\} + {}^r\alpha^2 \sum_{j=1}^{N} M_{zj}\ddot{\hat{q}}_j + {}^r\alpha^2 \sum_{j,l=1}^{N} C_{zjl}\dot{\hat{q}}_l\dot{\hat{q}}_j + {}^r\alpha \sum_{j=1}^{N} F_{zj}\dot{\hat{q}}_j + G_z \quad ,\forall z = 1 \dots N \,. \end{aligned} \tag{18}$$

Note that (20) is a system of N equations, while (19) is a single equation. Note also that the superscript of ξ is not an exponentiation but an index. Defining $\xi \triangleq [\xi^{4^{\mathrm{T}}}, \xi^{3^{\mathrm{T}}}, \xi^{2^{\mathrm{T}}}, \xi^{1^{\mathrm{T}}}, \xi^{0^{\mathrm{T}}}]^{\mathrm{T}} \in \mathbb{R}^{5N}$, (20) can be written also in a matrix product form

$$\begin{bmatrix} {}^r\alpha^4 I & {}^r\alpha^3 I & {}^r\alpha^2 I & {}^r\alpha I & I \end{bmatrix} \xi(t) = {}^r\tau({}^r\beta t) \;, \tag{21}$$

where $I \in \mathbb{R}^{N\times N}$ is the identity matrix. Each different torque input ${}^r\tau(t)$ linked to a trajectory ${}^rq(t)$ adds a row to (21). Stacking w control actions yields the linear application

$$P_w\xi(t) = U_w(t) \;, \tag{22}$$

where

$$P_w = \begin{bmatrix} {}^1\alpha^4 I & {}^1\alpha^3 I & {}^1\alpha^2 I & {}^1\alpha I & I \\ {}^2\alpha^4 I & {}^2\alpha^3 I & {}^2\alpha^2 I & {}^2\alpha I & I \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ {}^w\alpha^4 I & {}^w\alpha^3 I & {}^w\alpha^2 I & {}^w\alpha I & I \end{bmatrix}, \tag{23}$$

$$U_w(t) = \begin{bmatrix} {}^1\tau^{\mathrm{T}}({}^1\beta t) & {}^2\tau^{\mathrm{T}}({}^2\beta t) & \dots & {}^w\tau^{\mathrm{T}}({}^w\beta t) \end{bmatrix}^{\mathrm{T}} \,.$$

In order to be able to generalize the control actions w.r.t. velocity, we need to solve (22) for $\xi(t)$. Therefore P_w must be invertible. Since P_w is a Vandermonde matrix [20], it is always not-singular for ${}^{r_1}\alpha \neq {}^{r_2}\alpha, \forall r_1, r_2 = 1 \dots w, r_1 \neq r_2$, which is always true given H5.

Thus 5 examples not only are sufficient to generalize to each time scaling, but they are also the minimum amount for achieving this goal. Indeed, if $w < 5$ the matrix will not be invertible. If $w > 5$, $w - 5$ columns will be linearly dependent from the others. □

Remark 3. *The case of constant linear stiffness is a special case of the one treated in Theorem 1. Thus, $\dot{K} = \ddot{K} = 0$ leads to the same result, i.e., a minimum number w equal to 5.*

Corollary 1. *In the case of absence of any potential term (except for the elastic coupling between link and motors) in the robot dynamics, the number w of input signals required to track the desired trajectory $\hat{q}(t)$ with different velocities* *is reduced from 5 to 4.*

Proof. This result comes directly from inspecting (18). Indeed, the only term of the dynamics that is multiplied by ${}^{r}\alpha^{0}$ is $G(q)$, thus $\xi^{0} = G(q)$. Therefore, the absence of $G(q)$ removes the identity column from P_w. □

B. Control evaluation

The following corollary answers to G2.

Corollary 2. *Each time scaling of the reference trajectory* ${}^{*}q(t) = \hat{q}(\alpha t)$ *is realized as open loop evolution of the system* (1)-(2) *when* $w = 5$, *and the following feedforward control action is used*

$$\begin{aligned} {}^{*}\tau\left(\frac{t}{\alpha}\right) &= \begin{bmatrix}\alpha^{4}I & \alpha^{3}I & \alpha^{2}I & \alpha I & I\end{bmatrix}\xi(t), \\ \xi(t) &= P_5^{-1}U_5(t), \end{aligned} \tag{24}$$

where P_5 *and* U_5 *are defined as in* (23).

Proof. Since we proved P_5 to be full rank, it can be inverted. This allows to solve (22) as $\xi(t) = P_5^{-1}U_5$. The thesis follows from the direct application of (21). □

Note that this corollary implies that $\xi(t)$ is the control action needed for implementing $\hat{q}(t)$ itself.

IV. VARIABLE STIFFNESS ACTUATORS

In the more general case of VSAs, the elastic torque is given by a non linear function $\frac{\partial V^{\mathrm{T}}(q,\theta)}{\partial q}$, where $V(\cdot)$ is the elastic potential and $\theta \in \mathbb{R}^{N_{\mathrm{m}}}$, $N_{\mathrm{m}} \geq N$. Given the non linearity of $\frac{\partial V^{\mathrm{T}}(q,\theta)}{\partial q}$, finding an analytical closed form solution is non trivial. For a solution to exist in closed form, $\frac{\partial V^{\mathrm{T}}(q,\theta)}{\partial q}$ should be at least invertible, twice differentiable, homogeneous and with homogeneous inverse. Since it is unlikely for $\frac{\partial V^{\mathrm{T}}(q,\theta)}{\partial q}$ to possess all these qualities, we will look instead at an approximate solution. As a trade-off between generality and simplicity, we will consider only VSA that present an elastic torque function such that

$$\frac{\partial V^{\mathrm{T}}(q,\theta)}{\partial q} = g(\theta_{\mathrm{a}}(t))f(q(t) - \theta_{\mathrm{d}}(t)) \; . \tag{25}$$

$\theta_{\mathrm{d}} \in \mathbb{R}^{N}$ are (a combination of) the positions of the motors moving the robot's reference configuration, while $\theta_{\mathrm{a}} \in \mathbb{R}^{N_{\mathrm{m}}-N}$ are (a combination of) the positions of the motors varying the stiffness of the system. Fig. 1(b) graphically shows θ_{d} and θ_{a}. In [21], we discuss how to achieve such a separation of variables for a wide class of VSAs.

Assuming that the stiffness profile is tunable by the user is equivalent to set $\theta_{\mathrm{a}}(t)$. This means that the term $g(\theta_{\mathrm{a}}(t)) = K'(t)$ is a given non linear function of t. Assuming, without loss of generality, that the elastic torque contribution is null when the deflection $\phi \triangleq q - \theta_{\mathrm{d}}$ is null, then linearizing (25) w.r.t. the deflection ϕ, leads to

$$\begin{aligned} \frac{\partial V^{\mathrm{T}}(q,\theta)}{\partial q} &= g(\theta_{\mathrm{a}})f(\phi) \\ &= K'(t)\left\{ \left.\frac{\partial f(\phi)}{\partial \phi}\right|_{\phi=0} \phi + O'(\phi^{2}) \right\} \\ &= K(t)\left\{q - \theta_{\mathrm{d}} + O(\phi)\right\} , \end{aligned} \tag{26}$$

where $O(\cdot)$ is the function collecting the higher order terms that become negligible when ϕ is small.

Lemma 1. *Applying the method* (24) *to a VSA robot with elastic torque of the type* (25)*, leads to an error on the identified dynamic terms* ξ *that is proportional to the residual* $O(\phi)$*. In particular the error in the estimation of the control action is equal to*

$$E^{\mathrm{VSA}} \triangleq P_5^{-1}\left\{O(\phi) + D\dot{O}(\phi,\dot{\phi}) + B\ddot{O}(\phi,\dot{\phi},\ddot{\phi})\right\} \; . \tag{27}$$

Proof. The thesis comes directly by substitution. First we split the motor dynamics into the dynamics related to θ_{d} and the one related to θ_{a}. As mentioned, we can consider only the N equations related to θ_{d} . Let us define

$${}^{r}\tau^{\mathrm{VSA}} = {}^{r}\tau + {}^{r}R(\phi^{2}) \; , \tag{28}$$

where ${}^{r}\tau = [\cdots, {}^{r}\tau_{z}, \cdots]^{\mathrm{T}}$, and ${}^{r}\tau_{z}$ is defined as (19). Then, substituting (26) into (5)-(6), and following the same procedure described in the proof of Theorem 1 leads to

$$\begin{aligned} {}^{r}\tau^{\mathrm{VSA}} &= {}^{r}\tau + O(\phi) + D\dot{O}(\phi,\dot{\phi}) + B\ddot{O}(\phi,\dot{\phi},\ddot{\phi}) \\ &= {}^{r}\tau + O(\phi) + 2D\frac{\partial O}{\partial(\phi^{2})}\phi\dot{\phi} + 2B\left\{\frac{\partial O}{\partial(\phi^{2})}\phi\ddot{\phi} + \right. \\ &\qquad \left. +2\frac{\partial^{2} O}{\partial(\phi^{2})^{2}}\phi^{2}\dot{\phi}^{2} + \frac{\partial O}{\partial(\phi^{2})}\dot{\phi}^{2}\right\} \\ &\triangleq {}^{r}\tau + {}^{r}R(\phi^{2}) \; . \end{aligned} \tag{29}$$

Then, substituting (29) into (22), and inverting P_5 leads to the thesis

$$\xi^{\mathrm{VSA}}(t) = P_5^{-1}U_5^{\mathrm{VSA}}(t) = P_5^{-1}U_5(t) + P_5^{-1}R(\phi^{2}) \; , \tag{30}$$

where $R(\phi^{2}) = [{}^{1}R^{\mathrm{T}}, \ldots {}^{5}R^{\mathrm{T}}]^{\mathrm{T}}$. □

V. VALIDATION

In this section we present simulation and experimental results to validate the proposed method. We define the performance metric of the tracking error at the joints as

$$e \triangleq \frac{1}{NT}\sum_{z=1}^{N}\int_{0}^{T}\left|\hat{q}_{z}(t) - q_{z}(t)\right|\mathrm{d}t \; , \tag{31}$$

where $\hat{q}_z$ and q_z are the reference and measured trajectory of the $z-$th joint, respectively.

A. Simulation results

We test the method on a Baxter robot[2] simulated with MATLAB Robotic Toolbox[3] (Fig.1(a)). The robot dynamics is augmented with the series elastic actuator (SEA) dynamics, whose parameters are $J_z = 0.001$Nms^2/rad, $D_z = 0.01$Nms/rad, $K_z = 50$Nm/rad and joint friction $F_z = 0.8$Nms/rad. We simulate the robot in a drum playing task. The reference trajectory is reported in the video attachment and in Fig. 2: both 7DoF arms of the robot perform a vertical movement in the task space to hit the instrument. During the trajectory the robot end-effectors hit the drum, modeled as a spring $K_{\mathrm{E}} = 600$Nm/rad. Thus, the environmental interaction causes a contact force. The time scaling parameters

[2] https://www.rethinkrobotics.com/

[3] https://it.mathworks.com/help/robotics/

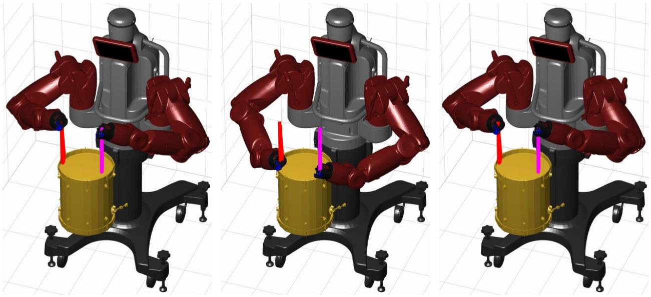


Fig. 2. Simulation. Photo-sequence of Baxter tracking the reference trajectory time scaled with $\alpha = 3.3333$, i.e. $\beta = 0.3$. The control input has been obtained with the proposed generalization method (24) given a training set of 5 examples.

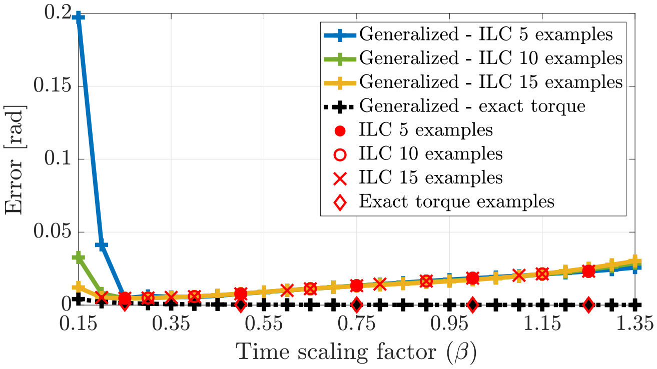


Fig. 3. Simulation. The abscissa axis represents different values of the time scaling parameter $\beta = 1/\alpha$. The ordinate axis represents the tracking error metric (31). The red markers indicate the tracking performance of the input signals used as training set for the generalization method (24).

employed are chosen such that the terminal time is equal to $T \in \{0.25, 0.5, 0.75, 1, 1.25\}$s. We considered as reference time for the generalization $T = 1$s, leading to the time scaling factors $\beta = 1/\alpha \in \mathbb{A}_5 = \{0.25, 0.5, 0.75, 1, 1.25\}$. Once acquired the five input torques, we generalize w.r.t. to a new time scaling parameter γ applying (24). We test the predicted control action for all terminal times within the set $\{0.15, 0.20, \ldots, 1.30, 1.35\}$s, obtained with the time scaling factors $\beta = 1/\alpha \in \{0.15, 0.2, \ldots 1.3, 1.35\}$, testing a total of 25 predictions.

Fig. 3 summarizes the results. The abscissa axis represents different values of the time scaling parameter β. The ordinate axis represents the tracking error metric (31). The error obtained in absence of any controller is $e = 1.0648$rad. First, we show, as a benchmark, the result (black dashed line in Fig. 3) obtained given a training set of five exact torque examples, i.e., control inputs that lead to a null tracking error. The tracking performance of these control inputs is depicted in the figure as red diamond markers. In this case the control actions do not introduce any noise in the identification process, leading to a perfect tracking for each tested generalization, i.e., for each tested value of $\alpha = 1/\beta$.

Then, we test the method in presence of a non null tracking error. We employ the controller presented in [21], i.e., a decentralized model-free method based on ILC. This algorithm learns a feedforward torque control action necessary to track a desired joint trajectory by iteratively attempting to perform the tracking task. A brief description of the controller is reported in Appendix. The performance of the five input examples (obtained with 200 iterations) is reported in Fig. 3 as red circle markers, while the solid blue line shows the error metric (31) evaluated for each simulated execution with a different scaling factor $\beta = 1/\alpha$. Results show that the tracking performance achieved by the generalized control inputs are similar to the one obtained through learning. Indeed, the solid blue line approximately lies over the red circle markers. The only exception is for the fastest motions, i.e., smaller $\beta = 1/\alpha$, where the imperfections of the input examples do not allow to achieve good tracking performance. Fig. 4 shows the tracking performance obtained with the proposed method when the desired trajectory is time scaled by $\alpha = 3.3333$, i.e., $T = 0.3$s, while Fig. 2 shows a photo-sequence of the same trial.

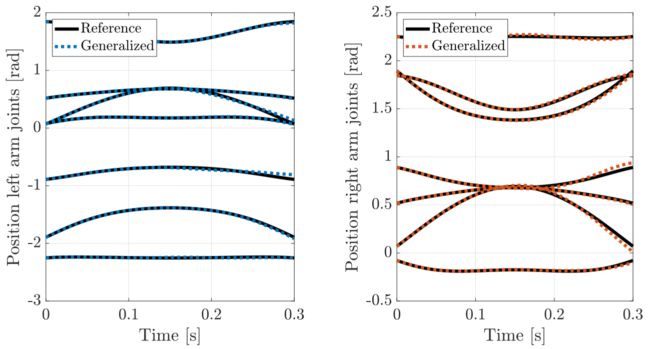


Fig. 4. Simulation. Tracking performance of the predicted control action (dashed line) given a training set of 5 examples. The time scale factor is $\beta = 0.3$. Left hand side figure refers to the left arm of the Baxter, while right hand side figure refers to the right arm.

Finally, as exploratory test to evaluate the possibility of exploiting more than the minimum amount of data for the generalization, we increase the number of examples used to generalize the control action. Therefore a non-square matrix (23) results, that we pseudo-invert - thus solving a minimum root mean square error approximation problem. To this end we randomly pick 10 more time scaling factors in the available set, and we learn the associated torque. We test two cases, the first one with 10 examples, the second one with 15 examples. These randomly picked examples are $\beta = 1/\alpha \in \mathbb{A}_{10} = \mathbb{A}_5 \cup \{0.3, 0.4, 0.65, 0.9, 1.15\}$ and $\beta = 1/\alpha \in \mathbb{A}_{15} = \mathbb{A}_{10} \cup \{0.2, 0.35, 0.6, 0.8, 1.1\}$. Then, we use these examples to obtain an approximation of the necessary control action. The green and orange solid lines in Fig. 3 show the results. In both cases, the increased number of examples does not lead to a performance improvement inside the bounds given by the slowest and fastest example. Outside of these bounds we can notice how increasing the number of examples reduces the error introduced by the noise in the input example (fastest motion). However, there is a slight performance degradation for the slowest motions, probably due to overfitting. Future work will better analyze this behavior.

B. *Experimental results*

We test the proposed method on a double pendulum actuated by VSA (Fig. 1(b)). In particular we employ two *qbmove* Advanced [22] that are agonistic-antagonistic actuators with variable stiffness, thus they satisfy the conditions discussed

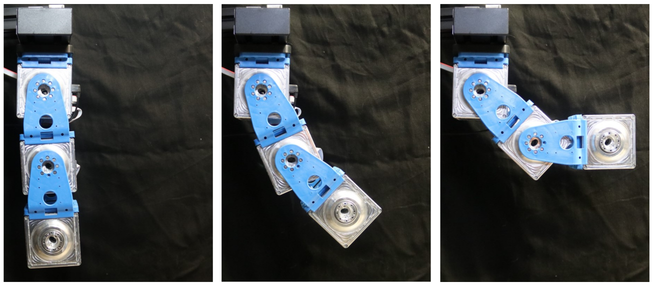


Fig. 5. Experiment: variable stiffness. Photo-sequence of the VSA double pendulum tracking the reference trajectory time scaled with $\beta = 0.9167$. The control input has been obtained with the proposed generalization method (24).

in Sec IV. The employed controller is the one described in the Appendix [21]. As reference trajectory we choose a five order minimum jerk trajectory for each joint, that goes from $q_z(0) = 0$rad to $q_z(T) = \pi/4$rad in T seconds, i.e.,

$$q_z(t) = \frac{\pi}{4}\left\{10\left\{\frac{t}{T}\right\}^3 - 15\left\{\frac{t}{T}\right\}^4 + 6\left\{\frac{t}{T}\right\}^5\right\}. \qquad (32)$$

This choice allows to have a non linear reference that is four time differentiable. Fig. 5 shows a photo-sequence of the trajectory.

The terminal time of the five examples is set equal to $T \in \{4, 6, 8, 10, 12\}$s. Choosing as reference time of the generalization $T = 6$s leads to the time scaling factors $\beta = 1/\alpha \in \{2/3, 1, 4/3, 5/3, 2\}$. Once acquired the five input torque examples, it is possible to generalize w.r.t. to a new time scaling parameter γ applying (24). We proceed testing the predicted control action for each terminal time ranging in the interval $\{3, 3.5, \dots, 12.5, 13\}$s, related to the time scaling parameters $\beta = 1/\alpha \in \{0.5 : 0.5/6 : 13/6\}$, testing a total of 21 predictions.

We investigate two cases; constant stiffness $K_z = 5$Nm/rad, $z = 1, 2$ and variable stiffness. In the latter case the stiffness profile is changed with a minimum jerk trajectory behavior (similar to (32)) from $K_z(0) = 5$Nm/rad to $K_z(T) = 20$Nm/rad, $z = 1, 2$.

Fixed stiffness: The learning process ended after 25 iterations. The results for this case is reported in Fig. 6. Red circle markers in the figure represent the tracking performance of the five input control actions learned through iterations. The blue solid line and the plus sign markers represent different tested values. Similarly to the simulation case, the tracking error obtained with the predicted control action of the generalized method is comparable to the error obtained with the learning process. When generalizing w.r.t. to time scale parameters outside of the learned range, the performance degrades. This is due to the noise introduced by the feedforward actions. In particular, we have that for slower trajectories, i.e., $\beta > 1$, the tracking performance becomes influenced by static friction and by its combination with the inertial and elastic effects. This results can be noticed in Fig. 7. This figure shows the tracking performance of the control input generalized with $\beta = 1.1667$ (dashed line). After testing the proposed method, we performed an additional learning process for the scaled trajectory, and we compared the tracking performance after 25 iterations. Both cases present a similar behavior validating our method. The influence of the static friction can be noticed at the end of the tracking in joint 1.

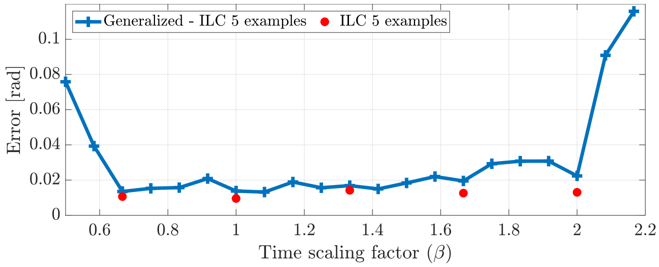


Fig. 6. Experiment: constant stiffness. The abscissa axis represents different values of the time scaling parameter $\beta = 1/\alpha$. The ordinate axis represents the tracking error metric (31). The red circles indicate the error obtained by the five input examples at the end of the learning process. Generalizing these control inputs using our method yields the error indicated by the blue plus sign markers.

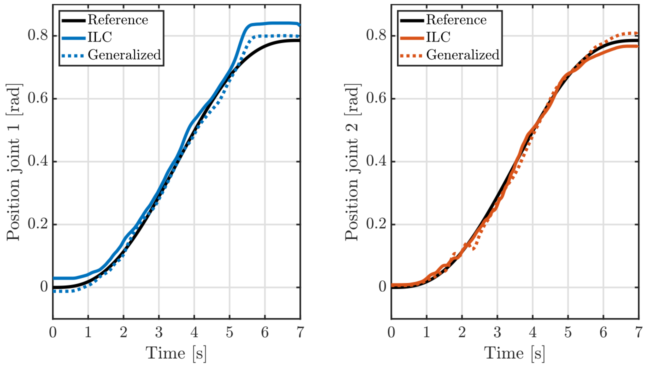


Fig. 7. Experiment: constant stiffness. Comparison between tracking performance for the predicted control action (dashed line) and for the control action after a new learning process (solid line). The time scale factor is $\beta = 1.1667$. Left hand side figure refers to Joint 1, while right hand side figure refers to Joint 2.

Variable stiffness: The learning process ended after 25 iterations. The results for this case is reported in Fig. 8. Red circle markers in the figure represent the tracking performance of the five input control actions. The blue solid line and the plus sign markers represent different tested values. Also in this case we obtain similar results despite the error committed due to linearization. As for the constant stiffness case, the static friction influences the results for slow trajectories. Fig. 9 reports an example of predicted torque (dashed line). The parameter β in this experiment is set to $\beta = 1.5$. After the prediction we learned (25 iterations) a new control action through ILC for the same time scale parameter. The obtained control actions are very similar (cf. solid and dashed lines in Fig. 9). Finally, Fig. 5 shows the robot tracking the desired trajectory time scaled with $\beta = 0.9167$.

VI. CONCLUSIONS

In this paper, we proposed an in depth analysis about the generalization of acquired control inputs to track a desired trajectory with different velocities. This ability can be beneficial in several applications as walking or pick-and-place tasks. We developed a method for articulated soft robots with time-varying linear stiffness. The derived analytic

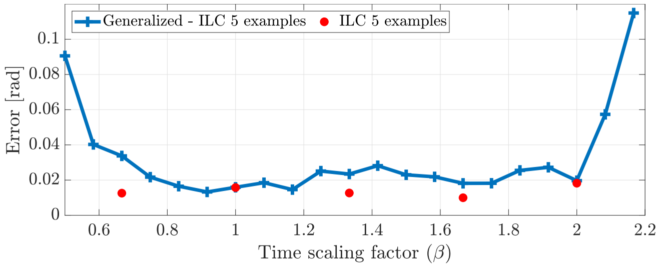


Fig. 8. Experiment: variable stiffness. The abscissa axis represents different values of the time scaling parameter $\beta = 1/\alpha$. The ordinate axis represents the tracking error metric (31). The red circles indicate the error obtained by the five input examples at the end of the learning process. Generalizing these control inputs using our method yields the error indicated by the blue plus sign markers.

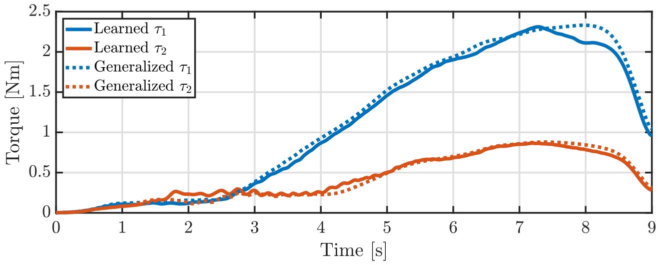


Fig. 9. Experiment: variable stiffness. Comparison between the predicted control action (dashed line) and the learned control action after a new learning process (solid line). The time scale parameter is $\beta = 1.5$.

solution is then extended to the case of variable stiffness actuators. Results prove that five control inputs are necessary and sufficient to generalize w.r.t. velocity variation. To test the validity of the method we performed simulations on a SEA Baxter robot and experiments on a VSA double pendulum, with fixed and varying stiffness.

The proposed method is agnostic to the procedure employed to acquire input examples τ. However, its benefits are maximized in case of model-free control laws, as ILC. Indeed, in case of model-based methods, the proposed technique could be employed to avoid recomputing the robot dynamics, that could be a time-consuming operation.

APPENDIX

The controller employed in Sec. V is a decentralized model-free method based on Iterative Learning Control (ILC). The control law for the $z-$th joint at the $i-$th iteration is

$$\tau_{z,i} = \tau_{z,i-1} + L_{\mathrm{UP},z}\tilde{x}_{z,i-1} + L_{\mathrm{FB},z}\tilde{x}_{z,i} \;, \qquad (33)$$

where $L_{\mathrm{UP},z}$ and $L_{\mathrm{FB},z}$ are the update and feedback gains for the $z-$th joint, and $\tilde{x}_{z,i} \triangleq \left[\hat{q}_z - q_{z,i}, \dot{\hat{q}}_z - \dot{q}_{z,i}\right]^{\mathrm{T}}$ is the position and velocity tracking error. Note that the subscript i refers to the current iteration, while the subscript $i-1$ refers to the previous iteration. The method requires an initial guess $\tau_{z,0}$ that can be put equal to zero or estimated on a rough approximation of the robot dynamic model. $L_{\mathrm{FB},z}$ is set as a linear quadratic regulator. Both $L_{\mathrm{UP},z}$ and $L_{\mathrm{FB},z}$ are tuned to guarantee the convergence of the iterative algorithm. In case of VSA robots, a feedback controller is added to independently regulate the desired stiffness profile commanded through θ_{a}. For the sake of space we cannot report here the full algorithm. The interested reader can refer to [21] for its detailed description.

REFERENCES

[1] T.J. Roberts and E. Azizi. Flexible mechanisms: the diverse roles of biological springs in vertebrate movement. *Journal of Experimental Biology*, 214(3):353–361, 2011.

[2] A. Albu-Schäffer, O. Eiberger, M. Grebenstein, S. Haddadin, C. Ott, T. Wimböck, S. Wolf, and G. Hirzinger. Soft robotics. *Robotics & Automation Magazine, IEEE*, 15(3):20–30, 2008.

[3] G. Pratt et al. Series elastic actuators. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 1, pages 399–406. IEEE, 1995.

[4] B. Vanderborght et al. Variable impedance actuators: A review. *Robotics and autonomous systems*, 61(12):1601–1614, 2013.

[5] C. Della Santina, M. Bianchi, G. Grioli, F. Angelini, M.G. Catalano, M. Garabini, and A. Bicchi. Controlling soft robots: balancing feedback and feedforward elements. *IEEE Robotics & Automation Magazine*, 24(3):75–83, 2017.

[6] D. Braun, M. Howard, and S. Vijayakumar. Optimal variable stiffness control: formulation and application to explosive movement tasks. *Autonomous Robots*, 33(3):237–253, 2012.

[7] A. Zhakatayev, M. Rubagotti, and H.A. Varol. Time-optimal control of variable-stiffness-actuated systems. *IEEE/ASME Transactions on Mechatronics*, 22(3):1247–1258, 2017.

[8] F. Angelini, C. Della Santina, M. Garabini, M. Bianchi, G.M. Gasparri, G. Grioli, M.G. Catalano, and A. Bicchi. Decentralized trajectory tracking control for soft robots interacting with the environment. *IEEE Transactions on Robotics*, 34(4):924–935, 2018.

[9] M. Hofer, L. Spannagl, and R. D'Andrea. Iterative learning control for fast and accurate position tracking with a soft robotic arm. *arXiv preprint arXiv:1901.10187*, 2019.

[10] T. Beckers and S. Hirche. Keep soft robots soft–a data-driven based trade-off between feed-forward and feedback control. *arXiv preprint arXiv:1906.10489*, 2019.

[11] K. Takahashi, T. Ogata, J. Nakanishi, G. Cheng, and S. Sugano. Dynamic motion learning for multi-dof flexible-joint robots using active–passive motor babbling through deep learning. *Advanced Robotics*, 31(18):1002–1015, 2017.

[12] A.D. Marchese, R. Tedrake, and D. Rus. Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator. *The International Journal of Robotics Research*, 35(8):1000–1019, 2016.

[13] J.A. Adams. Historical review and appraisal of research on the learning, retention, and transfer of human motor skills. *Psychological bulletin*, 101(1):41, 1987.

[14] M.C. Carter and D.C. Shapiro. Control of sequential movements: Evidence for generalized motor programs. *Journal of neurophysiology*, 52(5):787–796, 1984.

[15] J.M. Hollerbach. Dynamic scaling of manipulator trajectories. *Journal of Dynamic Systems, Measurement, and Control*, 106(1):102–106, 1984.

[16] A. De Luca and R. Farina. Dynamic scaling of trajectories for robots with elastic joints. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 3, pages 2436–2442. IEEE, 2002.

[17] S. Kawamura and N. Sakagami. Planning and control of robot motion based on time-scale transformation. In *Advances in Robot Control*, pages 157–178. Springer, 2006.

[18] M.W. Spong. Modeling and control of elastic joint robots. *Journal of dynamic systems, measurement, and control*, 109(4):310–318, 1987.

[19] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[20] K.B. Petersen, M.S. Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[21] R. Mengacci, F. Angelini, M.G. Catalano, G. Grioli, A. Bicchi, and M. Garabini. On the motion/stiffness decoupling property of articulated soft robots with application to model-free torque iterative learning control. *The International Journal of Robotics Research*.

[22] C. Della Santina, C. Piazza, G.M. Gasparri, M. Bonilla, M.G. Catalano, G. Grioli, M. Garabini, and A. Bicchi. The quest for natural machine motion: An open platform to fast-prototyping articulated soft robots. *IEEE Robotics & Automation Magazine*, 24(1):48–56, 2017.