Decentralized Trajectory Tracking Control for Soft Robots Interacting with the Environment

Franco Angelini[1,2,3], *Student Member, IEEE,* Cosimo Della Santina[1,3], *Student Member, IEEE,* Manolo Garabini[1], *Member, IEEE,* Matteo Bianchi[1,3], *Member, IEEE,* Gian Maria Gasparri[1], Giorgio Grioli[2], *Member, IEEE,* Manuel G. Catalano[2], *Member, IEEE,* and Antonio Bicchi[1,2,3], *Fellow, IEEE,*

***Abstract*—Despite the classic nature of the problem, trajectory tracking for soft robots, i.e. robots with compliant elements deliberately introduced in their design, still presents several challenges. One of these is to design controllers which can obtain sufficiently high performance while preserving the physical characteristics intrinsic to soft robots. Indeed, classic control schemes using high gain feedback actions fundamentally alter the natural compliance of soft robots effectively stiffening them, thus *de facto* defeating their main design purpose. As an alternative approach, we consider here to use a low-gain feedback, while exploiting feedforward components. In order to cope with the complexity and uncertainty of the dynamics, we adopt a decentralized, iteratively learned feedforward action, combined with a locally optimal feedback control. The relative authority of the feedback and feedforward control actions adapts with the degree of uncertainty of the learned component. The effectiveness of the method is experimentally verified on several robotic structures and working conditions, including unexpected interactions with the environment, where preservation of softness is critical for safety and robustness.**

***Index Terms*—Articulated Soft Robots, Motion Control, Iterative Learning Control.**

I. Introduction

HUMAN beings are able to effectively and safely perform a large variety of tasks, ranging from grasping to manipulation, from balancing on uneven terrain to running. They are also remarkably resilient to highly dynamic, unexpected events such as impacts with the environment. One of the enabling factors to achieve such performance is the compliant nature of the muscle-skeletal system. In the last decades, biologic actuation inspired the robotic research community, leading to the development of a new generation of robots embedding soft elements within their design, with either fixed or variable mechanical characteristics. Such approaches generated a fast-growing literature on "soft robotics". In the broad family of soft robots, two main subgroups can be distinguished: robots which take inspiration mostly from invertebrate animals [1] and are accordingly built with continuously deformable elements; and robots inspired to the muscle-skeletal system of vertebrates, with compliance concentrated in the robot joints [2], [3]. Work presented in this paper focuses on the control of the latter class of "articulated" soft robots, which are amenable to simpler and more uniform modelization. However, some lessons learned in this context may also prove useful in the control of "continuum" soft robots.

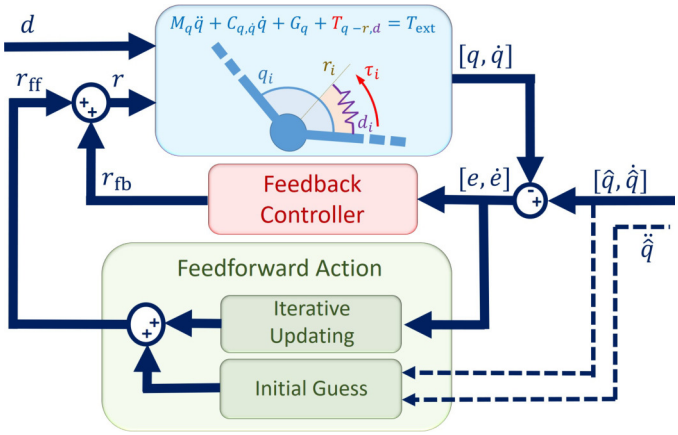


Figure 1: Schematic representation of the control architecture. $[q\ \dot{q}]$ are the Lagrangian variables and $\hat{q}$, $\dot{\hat{q}}$, $\ddot{\hat{q}}$ their desired evolutions. $[e\ \dot{e}]$ is the tracking error. M, C and G are the inertia, centrifugal and potential field terms, T is the spring torques vector, T_{ext} is the environmental external forces vector, d and r are the stiffness and reference inputs. r_{fb} is the feedback action, and r_{ff} is the feedforward action, which is the sum of a precomputed term and an estimated one.

In the literature several trajectory tracking solutions were proposed for soft robots. Feedback linearization was profitably employed in [4], [5] to design feedback control laws. In [6] a backstepping based algorithm was proposed.

However, all these techniques share two common drawbacks. First of all, they need an accurate model of the system. Secondly, feedback control laws have some fundamental limitations when they are applied to soft robots. Indeed, [7] argued that standard control methods fight against, or even completely cancel the physical dynamics of the soft robot to achieve good performances. This typically results in a stiffening of the robot, defeating the original purpose of building robots with physical compliance in their structure. In [7] it is suggested to employ low gain control techniques to have the original softness of the robot minimally perturbed by the control algorithm. This leads to the exploitation of controllers relying mostly on the anticipatory (i.e. feedforward) action in such a way to recover

This research has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 645599 (Soma), No. 688857 (SoftPro), No. 732737 (ILIAD).

[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

frncangelini@gmail.com

from the typically lower performance of a low gain controller. It is also observed that direct use of model-based inverse inputs is rarely applicable to a robotic system, especially if interacting with its environment. Thus, it is considered the use of learning approaches to feedforward control.

Iterative Learning Control (ILC, [8]) has a relatively long history in robotics (see e.g. [9], [10]), where it was applied mostly for rigid robots. In [7] an ILC technique was briefly introduced as a possible approach to learn the necessary anticipatory action in uncertain conditions. However, no systematic design nor analysis tools were provided to actually synthesize an iteratively learned feedforward control with convergence and stability guarantees.

In this work we build upon the intuition provided in [7] a full fledged ILC-based control architecture able to track a desired trajectory with a soft robot with generic, unknown kinematics. The presence of unexpected interactions with an unstructured environment is considered in the analysis, and the convergence is assured. The controller is shown to achieve the desired tracking performance without substantially altering the effective stiffness of the robot.

To validate the ability of the algorithm to robustly work in various experimental conditions, we designed a series of experiments employing soft robots with different serial and parallel kinematic structures and with increasing level of interaction with the external environment. In all experiments the algorithm had only an a-priori knowledge of the number of joints and of the physical characteristics of the elastic robot joints. The algorithm is able to learn the correct control action to precisely track the desired trajectory in all considered scenarios.

The paper is organized as follow: in section II we introduce the control problem and the soft robot dynamical model in use, in section III we derive the control architecture, and we show how all the introduced issues can be addressed. Finally, in section IV the controller effectiveness and robustness is shown.

II. Problem Statement

We refer to the model of a $N-$joint articulated soft robot with $N_\mathrm{m} \geq N$ motors introduced in [11] as

$$\begin{cases} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \dfrac{\partial V(q,\theta)^T}{\partial q} = T_\mathrm{ext}(q,\dot{q}) & (1) \\ J\ddot{\theta} + \dfrac{\partial V(q,\theta)^T}{\partial \theta} = F_\mathrm{m}\,, & (2) \end{cases}$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^N$ are the vectors of generalized joint positions, velocities and accelerations respectively, while $\theta, \dot{\theta}, \ddot{\theta} \in \mathbb{R}^{N_\mathrm{m}}$ are the vectors of motor positions, velocities and accelerations respectively, $M(q) \in \mathbb{R}^{N\times N}$ is the robot inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{N\times N}$ collects the centrifugal, Coriolis and damping terms, $G(q) \in \mathbb{R}^N$ collects gravity effects, $J \in \mathbb{R}^{N_\mathrm{m}\times N_\mathrm{m}}$ is the motor inertia matrix, and $T_\mathrm{ext}(q,\dot{q}) \in \mathbb{R}^N$ collects the interaction forces with the external environment and model uncertainties. $V(q,\theta)$ is the potential of the elastic energy stored in the system, while $F_\mathrm{m} \in \mathbb{R}^{N_\mathrm{m}}$ are the motor torques.

In this work we use a simplified model, introducing the following further assumptions:

- motor dynamics (2) is negligible, or equivalently, it is perfectly compensated by a low level control, so that θ can be considered to be effectively a control input;
- interactions with the environment can be modeled with a suitable smooth force field [12];
- there exists a change of coordinates between the motor positions θ and two set of variables $r \in \mathbb{R}^N$ and $d \in \mathbb{R}^{N_\mathrm{m}-N}$ such that $\frac{\partial V(q,\theta)^T}{\partial q} = T(q-r,d)$. Here, r can be regarded as a joint reference position, while d models parameters used to adjust the stiffness. The elastic torque vector $T(q-r,d) \in \mathbb{R}^N$ models the elastic characteristic of the soft robot. This model depends on the actuator physical implementation and is typically known from the actuator data-sheet [13]. The role of d depends on the considered actuator design. E.g. in the case of series elastic actuators ([14]) d is not present ($N_\mathrm{m} = N$), while for a VSA (Variable Stiffness Actuator [15]), d indicates the joint co-contraction level ($N_\mathrm{m} = 2N$).

Hence the considered model of a $N-$joint articulated soft robot is

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + T(q-r,d) = T_\mathrm{ext}(q,\dot{q}). \quad (3)$$

In this paper, we will consider the design of the control input $r \in \mathbb{R}^N$, i.e. the reference position, so as to achieve prescribed specifications, while the stiffness adjusting variables d are considered as given, possibly time-varying, parameters.

It is instrumental for the problem definition and for the control derivation to rewrite, without loss of generality, the system (3) in a decoupled form , according to e.g. [16],

$$\begin{bmatrix} \dot{q}_i \\ \ddot{q}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \dfrac{-\beta_i}{I_i} \end{bmatrix} \begin{bmatrix} q_i \\ \dot{q}_i \end{bmatrix} - \begin{bmatrix} 0 \\ \dfrac{1}{I_i} \end{bmatrix} \tau_i(q_i - r_i, d_i) + \begin{bmatrix} 0 \\ D_i(q,\dot{q}) \end{bmatrix}, \quad (4)$$

where $i = 1 \ldots N$, $\begin{bmatrix} q_i & \dot{q}_i \end{bmatrix}^T$ is the state vector composed by the angle and the velocity of a single joint, τ_i is the i-th element of the elastic torque vector T, r_i is the i-th element of the control input r, d_i is the i-th element of d, I_i and β_i are, respectively, the inertia and the damping seen from the i-th joint. $D_i(q,\dot{q})$ collects the terms acting on the i-th joint, i.e. the effects of the dynamic coupling and external forces.

Given a reference trajectory $\hat{q} : [0, t_\mathrm{f}) \to \mathbb{R}^N$, with all its time derivatives, and a stiffness adjusting variables d, the control objective is to derive an opportune control action $r : [0, t_\mathrm{f}) \to \mathbb{R}^N$ able to regulate system (3) on $\hat{q}$ in the whole control interval $[0, t_\mathrm{f})$. Other goals that we set out for our control design are:

i) the controller should not alter the physical mechanical stiffness more than a given amount. Given a $\delta \geq 0$, it has to be assured that the closed loop stiffness of the system remains in a neighborhood of radius δ of the open loop stiffness (as underlined in [7]), i.e.

$$\left\| \left.\frac{\partial T(q-r,d)}{\partial q}\right|_{q\equiv r} - \left.\frac{\partial T(q-\psi(q),d)}{\partial q}\right|_{q\equiv q_*} \right\| \leq \delta, \quad (5)$$

where $\psi(q)$ is a feedback control law, q_* is such that $\psi(q_*) = q_*$, and Euclidean norm is used;

ii) independence from the robot kinematic structure. The controller design can be based only on the knowledge of individual joint dynamic parameters (I_i, β_i and τ_i in (4)), while the terms $D_i(q,\dot{q})$ are completely unknown. In other terms, the controller is completely decentralized at joint level, and can be applied to robots of different kinematic and dynamic structure without modifications;
iii) robustness to environmental uncertainties, i.e. the algorithm convergence has to be assured for every unknown smooth $T_{\text{ext}}(q,\dot{q})$.

Note that requiring (ii) and (iii) implies a robust behavior to system uncertainties too.

III. Control Architecture

In this section we present the general control architecture and its derivation. In particular, we show how the goals defined in section II can be achieved. Note that all the proofs of the propositions and lemmas stated in this section are reported in the appendix.

Fig. 1 shows the general scheme of the proposed control algorithm, merging a low-gain feedback action with an opportune feedforward. The theory of ILC [8] provides a suitable framework to synthesize controllers in which a pure feedback loop and an iterative loop jointly contribute to determine the input evolution. The term "iterative loop" means that the task is repeated, and the knowledge acquired in past trials (i.e. iterations) is exploited to increase the performances of future ones. A generic ILC control law has the form[1] [8]

$$r^k(t) = r^{k-1}(t) + c(e^k, e^{k-1}, t), \tag{6}$$

where k is the iteration index, $r^k : [0, t_\text{f}) \to \mathbb{R}^N$ is the input vector at k-th iteration, hence r^{k-1} is the knowledge acquired from past trials. $r^0(t)$ is the feedforward action at the first iteration. $e^k : [0, t_\text{f}) \to \mathbb{R}^N$ is the error vector at k-th iteration defined as

$$e^k(t) \triangleq \begin{bmatrix} \hat{q}_1(t) - q_1^k(t) \\ \dot{\hat{q}}_1(t) - \dot{q}_1^k(t) \\ \vdots \\ \hat{q}_N(t) - q_N^k(t) \\ \dot{\hat{q}}_N(t) - \dot{q}_N^k(t) \end{bmatrix}, \tag{7}$$

and $c(e^k, e^{k-1}, t)$ is the *updating law* (note that e^0 is assumed null). In this work we consider an iterative update and linear time-variant state feedback

$$c(e^k, e^{k-1}, t) = K_\text{on}(t) e^k(t) + K_\text{off}(t) e^{k-1}(t), \tag{8}$$

where $K_\text{on}(t) \in \mathbb{R}^{N \times 2N}$ and $K_\text{off}(t) \in \mathbb{R}^{N \times 2N}$ collect the control gains. Note that the subscripts "on" and "off" in (8) stand for "on-line" and "off-line" respectively. Thus, the "on-line" term is the one computed during the trial execution (feedback component), while the "off-line" term is the one computed bewtween two consecutive trials (updating component).

The goals listed in section II can be achieved with a proper choice of the control gains $K_\text{on}(t)$ and $K_\text{off}(t)$. In particular, goal (i) will translate into a choice of feedback gains K_on that are sufficiently small (section III-A). Goal (ii) is achieved considering decentralized gains (section III-B), i.e. $K_\text{off}(t) := \text{diag}(K_{\text{off},i}(t))$ and $K_\text{on}(t) := \text{diag}(K_{\text{on},i}(t))$, where $K_{\text{off},i} := [K_{\text{poff},i}\ K_{\text{doff},i}] \in \mathbb{R}^{1\times 2}$ are the feedforward gains, and $K_{\text{on},i} := [K_{\text{pon},i}\ K_{\text{don},i}] \in \mathbb{R}^{1\times 2}$ are the feedback gains proportional to the position and velocity error of the i-th joint. In section III-B it is shown how goal (iii) is achieved with a proper choice of the control gains such that the ILC convergence laws (12) and (13) are satisfied.

In the following we describe the details of the proposed controller components and their derivation. For the sake of readability, we will omit the suffixes k and $k-1$ (indicating the iteration) when they are not necessary.

A. Constraint on Feedback

The goal (i) imposes a restriction in using a high feedback action, as stated by the following proposition (note that this proposition was previously stated without proof in [7])

Proposition 1. *If*

$$\left\| \left. \frac{\partial \psi}{\partial q} \right|_{q \equiv q_*} \right\| \leq \delta \left\| \left. \frac{\partial T(q - \psi, d)}{\partial q} \right|_{q \equiv q_*} \right\|^{-1} \tag{9}$$

then (5) *holds.*

It is worth noting that feedforward action does not affect this condition, since it does not depend on q. This suggests to favour low gain feedback techniques rather than high gain ones when working with soft robots.

In case of decentralized control condition (9) can be simplified as follows.

Lemma 1. *If the control algorithm is decentralized, i.e.* $\frac{\partial \psi}{\partial q}$ *is diagonal, and if*

$$\left\| \left. \frac{\partial \psi_i}{\partial q_i} \right|_{q \equiv q_*} \right\| \leq \delta \left\| \left. \frac{\partial T(q - \psi, d)}{\partial q} \right|_{q \equiv q_*} \right\|^{-1} \quad \forall\, i \tag{10}$$

where $\frac{\partial \psi_i}{\partial q_i}$ *is the* $i-th$ *diagonal element, then* (5) *holds.*

Thus, employing a low-gain controller it is possible to preserve the mechanical behavior of an articulated soft robot. At this point the main issue is to design a low-gain controller able to achieve good tracking performance.

B. Control Design

In this section we describe the derivation of the three components of proposed control algorithm, i.e. blocks *Initial Guess*, *Feedback Controller* and *Iterative Update* in Fig. 1.

The first step is to evaluate the feedforward action at the first iteration $r_i^0(t)$ (initial guess). This is computed solving

$$-\tau_i(\hat{q}_i, r_i^0, d_i) = I_i \ddot{\hat{q}}_i + \beta_i \dot{\hat{q}}_i + \Delta_i, \tag{11}$$

where Δ_i is the torque needed to arrange the robot in the initial condition (known by hypothesis), and $\hat{q}_i(t), \dot{\hat{q}}_i(t), \ddot{\hat{q}}_i(t)$ is the desired trajectory.

To achieve goal (iii) we consider convergence rules assuring convergence of the learning process in presence of unknown

[1]Note that some ILC control laws have the form $r^k = \alpha r^{k-1} + c$, where $\alpha \in (0,1]$ is a *forgetting factor*. In this work we will use $\alpha = 1$ to match the chosen convergence condition.

state dependent force fields. This includes in (3) coupling terms and interactions with the external environment (i.e. D_i in (4)). We use here conditions introduced in [17] [18], where the sufficient conditions are imposed separately for the on-line and off-line terms. Given a system in the form $\dot{x}(t) = f(x(t),t) + H(t)\nu(t) + \mu(t)$, where x, ν and μ are the state, control input and uncertainties vectors, f is the system function and H is the input matrix, the ILC convergence condition are

$$\left\|(I + K_{\text{on}}(t)H(t))^{-1}\right\| < 1\,,\ \forall t \in [0, t_{\text{f}}) \tag{12}$$

$$\|I - K_{\text{off}}(t)H(t)\| < 1\,,\ \forall t \in [0, t_{\text{f}})\,. \tag{13}$$

Thus, we proceed designing the control gains K_{on} and K_{off} such that (12) and (13) are fulfilled. Given the first iteration control action $r_i^0(t)$, computed as in (11), we linearize the dynamics of the decoupled system (4) around the desired trajectory $(\hat{q}_i(t), \dot{\hat{q}}_i(t), r_i^0(t))$, obtaining

$$\dot{e}_i(t) = A_i(t)e_i(t) + B_i(t)u_i(t) + \eta_i(q,\dot{q}) \quad \forall i = 1 \ldots N, \tag{14}$$

where $e_i = \begin{bmatrix}\hat{q}_i - q_i & \dot{\hat{q}}_i - \dot{q}_i\end{bmatrix}^T$ is the vector containing the $2i-1$-th and $2i$-th elements of (7), $u_i(t) = r_i^0(t) - r_i(t)$ is the control variation, $\sigma_i(t) = \frac{\partial \tau_i}{\partial q_i}(t)$ is the stiffness, η_i collects all the uncertainties and

$$A_i(t) = \begin{bmatrix} 0 & 1 \\ \dfrac{-\sigma_i(t)}{I_i} & \dfrac{-\beta_i}{I_i} \end{bmatrix},\ B_i(t) = \begin{bmatrix} 0 \\ \dfrac{\sigma_i(t)}{I_i} \end{bmatrix}. \tag{15}$$

The convergence condition (12) applied to the the decoupled system (14) is rephrased as

$$\left|\frac{1}{1 + K_{\text{on},i}(t)B_i(t)}\right| < 1\,,\ \forall t \in [0, t_{\text{f}})\,, \tag{16}$$

where $K_{\text{on},i}(t)$ are the feedback control gains, and B_i is the input matrix (H in (12)). This inequality is always verified when the term $K_{\text{on},i}(t)B_i(t)$ is positive. Among all the possible local feedback actions, we propose the choice of the feedback control gain $K_{\text{on},i}(t)$ as locally optimal. In particular $K_{\text{on},i}(t)$ is the solution of the time varying linear quadratic optimization problem (see e.g. [19] Chapter 5)

$$\int_0^{t_{\text{f}}} e_i^T Q e_i + R\,u_i^2\,\mathrm{d}t, \tag{17}$$

where $Q \in \mathbb{R}^{2\times 2}$ is a diagonal positive definite matrix and $R \in \mathbb{R}^+$.

The i-th feedback gain vector is given by

$$K_{\text{on},i}(t) = \frac{B_i(t)^T S_i(t)}{R}, \tag{18}$$

where $S_i(t)$ comes from the solution of the time-varying differential matrix Riccati equation

$$\dot{S}_i = -S_i A_i - A_i^T S_i + S_i B_i R^{-1} B_i^T S_i - Q\,, \tag{19}$$

with the boundary constraint $S_i(t_{\text{f}}) = \emptyset$. Hence feedback control gains are automatically tuned by the algorithm, leaving to the user only the choice of Q and R, which do not depend on i and are the only free parameter of the whole algorithm.

The choice of R directly affects the control authority, i.e. by increasing R the use of feedback control is penalized, and the gains $K_{\text{on},i}$ are reduced. This is assured by the following proposition.

Proposition 2. *If $K_{\text{on},i}$ is as in* (18)*, then*

$$\forall \gamma \geq 0 \quad \exists R > 0 \; s.t. \; ||K_{\text{on},i}|| \leq \gamma. \tag{20}$$

Thus, condition (9) can always be fulfilled by choosing $\gamma = \delta \left\| \frac{\partial T(q-\psi,d)}{\partial q}\Big|_{q\equiv q_*} \right\|^{-1}$, achieving goal (i).

Finally, the following proposition assures that the proposed feedback action is compatible with a convergent learning process.

Proposition 3. *The feedback rule in* (18) *fulfills the ILC convergence condition* (16) *for all $R > 0$.*

Condition (13) applied to the the decoupled system (14) is

$$\left|1 - K_{\text{off,i}}(t)B_i(t)\right| < 1\,,\ \forall t \in [0, t_{\text{f}})\,, \tag{21}$$

where $K_{\text{off},i}(t)$ are the iterative control gains.

The following proposition, if fulfilled together with proposition 3, assures the convergence of the learning process.

Proposition 4. *The convergence condition* (21) *is fulfilled by the following decentralized ILC gain $\forall \varepsilon \in [0,1)$ and $\forall\, \Gamma_i^T(t) \in \ker\{B_i^T(t)\}$*

$$K_{\text{off},i}(t) = (1+\varepsilon)B_i(t)^{\dagger} + \Gamma_i(t)\,, \tag{22}$$

where $B_i(t)^{\dagger}$ is the Moore-Penrose pseudoinverse of the matrix $B_i(t)$ in (14)*.*

Increasing the value of the parameter ε makes the convergence rate of the algorithm higher. The reason is that the control gains $K_{\text{off},i}$ are linear w.r.t. ε. Performing some experimental tests (not reported here), we found $\varepsilon = 0.9$ to provide a good trade-off between ILC convergence rate and stability.

Because of (15) and $\Gamma_i^T(t) \in \ker\{B_i^T(t)\}$, it follows that $\Gamma_i(t) = [K_{\text{poff},i}(t) \quad 0]$, where $K_{\text{poff},i}(t) \in \mathbb{R}$. We heuristically choose $\Gamma_i(t)$ to maintain the same balance between proportional and derivative components of the feedback gains $K_{\text{on},i}$

$$K_{\text{poff},i}(t) = \frac{||K_{\text{pon},i}||}{||K_{\text{don},i}||} K_{\text{doff},i}(t)\,. \tag{23}$$

C. Overall Control Action

Combining (6), (8), (18), (22) and (23), the overall control action applied on the $k-$th iteration at the $i-$th joint results

$$\begin{cases} r_i^k(t) = r_i^{k-1}(t) + K_{\text{on},i}(t)\, e_i^k(t) + K_{\text{off},i}(t)\, e_i^{k-1}(t) \\ K_{\text{on},i}(t) = \dfrac{\sigma_i(t)}{I_i R}[S_i^{(2,1)}(t) \quad S_i^{(2,2)}(t)] \\ K_{\text{off},i}(t) = \left(\dfrac{(1+\varepsilon)\, I_i}{\sigma_i(t)}\right) \begin{bmatrix} \dfrac{||S_i^{(2,1)}(t)||}{||S_i^{(2,2)}(t)||} & 1 \end{bmatrix} \end{cases}, \tag{24}$$

where $r_i^k(t)$ is the control input of the i-th joint, $K_{\text{on},i}(t)$ and $K_{\text{off},i}(t)$ are the feedback and iterative control gains of the i-th joint defined in (18), (22) and (23), e_i^k and e_i^{k-1} are the current and previous iteration tracking errors of the i-th joint defined as (7), I_i is the inertia seen by the $i-th$ joint, $-\sigma_i = \frac{\partial \tau_i}{\partial r_i}$, τ_i

is torque model of the i-th joint [13], $S_i^{(2,1)}(t)$ and $S_i^{(2,2)}(t)$ are the elements $2,1$ and $2,2$ of $S_i(t)$, solution of the Riccati equation (19). We impose $\varepsilon = 0.9$. $Q \in \mathbb{R}^{2\times 2}$ and $R \in \mathbb{R}^{+}$ are the weight in the time-variant linear quadratic regulator (17). It is worth noting that this control action can be derived in a completely autonomous manner and that Q and R are the only free parameters left to be tuned by the user.

The control rule (24) achieves all the goals in section II. Goal (i) is achieved by lemma 1 and proposition 2. Goal (ii) is achieved by the decentralized structure of the controller. Finally, goal (iii) is achieved by proposition 3 and proposition 4.

Algorithm 1 briefly summarizes the automatic procedure to learn an appropriate control action to achieve good tracking performance (i.e. low tracking error), given a desired trajectory $\hat{q}(t)$ and a desired stiffness input profile $d(t)$. It is worth noting that changing $\hat{q}(t)$ or $d(t)$ makes worthless for the new task the learned control action $r^k(t)$. This is probably the major limitation of ILC-based control techniques. Future works will address this point.

Algorithm 1 Control procedure pseudo-code

procedure INITIALIZATION
  $Set(\hat{q}(t), \dot{\hat{q}}(t), \ddot{\hat{q}}(t))$ ▷ Desired trajectory
  $Set(d(t))$ ▷ Stiffness input parameter
  $Set(Q, R)$ ▷ Control weight parameter
  $Compute(r^0(t))$ ▷ Eq. (11)
  $Evaluate(K_{\text{on}}(t))$ ▷ Eq. (18), (19)
  $Evaluate(K_{\text{off}}(t))$ ▷ Eq. (22), (23)
procedure LEARNING
  $k \leftarrow 1$
  $e^{k-1}(t) \leftarrow 0$
  do
    $Run\_Trial(r^{k-1})$ ▷ Note: r^k is computed on-line
    $Store(e^k, r^k)$
    $Update(r^k)$ ▷ Off-line update: $r^k + K_{\text{off}} e^k$
    $k \leftarrow k+1$
  while $e^{k-1} > threshold$

Finally, it is worth remarking that through the problem statement and control analysis we made some very basic assumptions. First of all, we assumed that motor dynamics is negligible, and that the VSA low-level controller perfectly tracks the motor position references. Then, we assumed that the desired trajectory $\hat{q}(t), \dot{\hat{q}}(t), \ddot{\hat{q}}(t)$ is feasible, i.e. there are not any hindrances (neither kinematic nor dynamic nor environmental) to the trajectory tracking. Furthermore, a basic assumption in ILC is that the robot is in $\hat{q}(0), \dot{\hat{q}}(0), \ddot{\hat{q}}(0)$ at the beginning of every iteration. Additionally, we assumed that the system state $q(t), \dot{q}(t)$ measurements are accurate. Finally, we hypothesized to have an accurate model of the VSA elastic transmission τ_i and to know the value of I_i, β_i and Δ_i. In section IV we will show through experiments that most of these assumptions can be relaxed without compromising the algorithm convergence and performance.

IV. EXPERIMENTAL RESULTS

To test the effectiveness of the proposed method in different experimental conditions, we developed an assortment of soft robotic structures, spanning from serial to parallel robots. All these robots are built using the VSA *qbmove maker pro* [3]. This is an actuator implementing the antagonistic principle both to move the output shaft and to vary its stiffness. The antagonistic mechanism is realized via two motors connected to the output shaft through a non-linear elastic transmission. The position of each motor and of the output shaft is measured with a AS5045 magnetic encoder. This sensor has a resolution of 12 bit. The *qbmove* spring characteristic τ_i in (4) is

$$\begin{cases} \tau_i = 2k\cosh(ad_i)\sinh(a(q_i - r_i)) + m(q_i - r_i) \\ \sigma_i = 2ak\cosh(ad_i)\cosh(a(q_i - r_i)) + m, \end{cases} \tag{25}$$

where τ_i, d_i, r_i and q_i are the i-th component of T, d, r and q respectively (defined in section II), while $a = 6.7328\frac{1}{\text{rad}}$, $k = 0.0222\text{Nm}$ and $m = 0.5\frac{\text{Nm}}{\text{rad}}$ are model parameters.

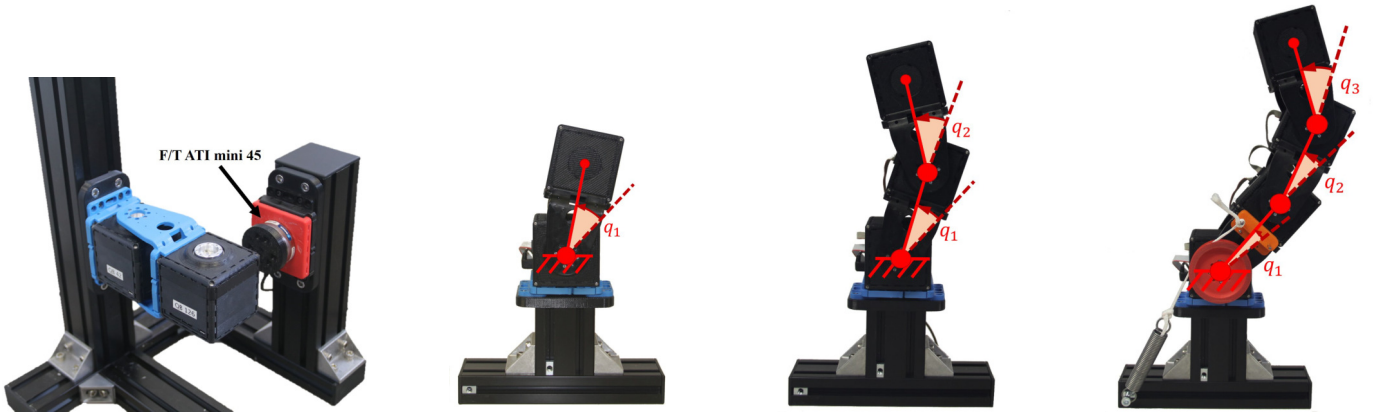
The four experiments are designed to test the algorithm in various working conditions and to show its ability to achieve all the goals in section II. The experiments are presented in increasing order of complexity. Experiment 1 aims to show the dependency (once Q is fixed) of the algorithm on the parameter R and to show the ability of the proposed method to preserve the robot mechanical behavior. In Experiment 2 the algorithm is tested in learning how to invert the system dynamics, with limited external interactions, while in Experiment 3 a change in the sign of the gravity torque is considered. Finally, in Experiment 4 we test the algorithm on a parallel structure and in presence of several abrupt and unexpected contacts with the environment. In order to remain as independent as possible from a given system architecture, the quantities β_i and I_i are estimated through step response in the first phase of each experiment, while Δ_i is estimated as the torque needed to arrange the robot in the initial condition. In all experiments Q is set with diagonal elements 1 and 0.01.

In the next sections we will employ

$$E(k) \triangleq \sum_{i=1}^{N} \frac{\int_0^{t_\text{f}} \left|\hat{q}_i(t) - q_i^k(t)\right| \text{d}t}{N\, t_\text{f}} \tag{26}$$

as definition of the evolution of the tracking error over iterations. Indeed, $\hat{q}_i(t)$ is the $i-$th joint reference trajectory (provided for every experiment), while $q_i^k(t)$ is the $i-$th joint position measured by the encoder placed at the $i-$th output shaft at $k-$th iteration. This error definition is exploited to give a quantitative measure of variation of the tracking performance over iterations. Furthermore, it is worth noting that the error used to refine the control action every iteration is (7). The used actuator does not have any sensor to measure the velocity $\dot{q}_i^k(t)$, so it is estimated through an high-pass filtering of the measured position $q_i^k(t)$. Despite the imprecise velocity measurement, the algorithm is able to converge, proving the robustness of the proposed method.

Finally, the time required for the algorithm convergence strictly depends on the performed experiment. In more detail, the needed time will be $(t_\text{a} + t_\text{f} + t_\text{off}) \times n_\text{k}$ where n_k is the number of performed iterations, t_f is the task terminal time,



(a) 1-dof planar config. (b) 1-dof config. (c) 2-dof config. (d) 3-dof config.

Figure 2: Experimental setups and reference frames. 1-dof planar configuration (Experiment 1, IV-A) (2a). After the learning phase a bar (with a 6−axis force/torque ATI mini 45 mounted on it) is placed next to the robot. 1-dof configuration (Experiment 2, and 3; IV-B, IV-C) (2b). 2-dof configuration (Experiment 2; IV-B) (2c). 3-dof configuration (Experiment 2; IV-B) (2d): a parallel spring is included in this setup to avoid that the torque required to the base actuator exceeds its torque limit. Note that for the success of the experiment the knowledge of the exact elastic constant of the spring is not required.

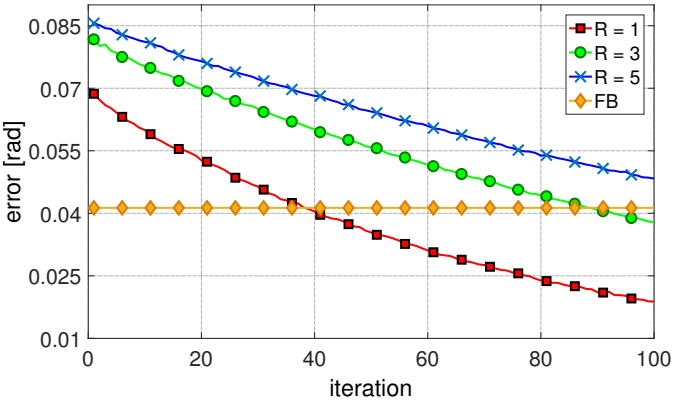


Figure 3: Experiment 1: evolution of the error over iterations (computed as (26)) for the three R values. Lower R values correspond to lower initial error and faster convergence rate. The orange horizontal line is the error with the PII controller.

t_a is the time needed to arrange the robot in the initial condition, and t_{off} is the time needed to compute (off-line) the control action between two trials. Note that the only value that does not depend on the experiment is t_{off}, which is usually negligible.

A. Experiment 1

Experimental Setup: The objective of this experiment is to evaluate the behavior of the system for different values of the parameter R, given $Q = diag([1, 0.01])$. In detail, we analyze the algorithm convergence rate and its softness preservation capability. To lower R values correspond higher feedback and feedforward gains K_{on} and K_{off} (see (24)). This translates into a faster convergence rate for lower R values. On the other hand, higher feedback gains (i.e. lower R values) tends to stiffen the robot (as theoretically described in III-A).

The experimental setup is composed of a planar 1−dof (degree of freedom) soft robot and a force sensor (6−axis force/torque ATI mini 45) mounted on a bar fixed to the frame (Fig. 2a). The experiment is divided in two steps. First of all, we apply the algorithm to the robot (in this phase the bar with the sensor is absent) using as reference trajectory

$$\hat{q}_1(t) = 0.074t^5 - 0.393t^4 + 0.589t^3, \quad t \in [0, 2). \tag{27}$$

This is a smoothed ramp spanning from 0 to 0.7854rad in $t_f = 2$s. This step is repeated three times, each one testing the algorithm with a different value of the parameter R: $R = 1$, $R = 3$ and $R = 5$. The maximum stiffness variation δ in (9) increases lowering R. In detail, to $R = 1$ corresponds $\delta = 0.33 \frac{\text{N m}}{\text{rad}}$, to $R = 3$ corresponds $\delta = 0.12 \frac{\text{N m}}{\text{rad}}$, and to $R = 5$ corresponds $\delta = 0.08 \frac{\text{N m}}{\text{rad}}$. Afterwards, in the second step, we place the bar with the force sensor next to the robot, in such a way that an impact will occur during the trajectory tracking (see Fig. 2a). We measure the force applied by the robot using the three different control action obtained at the end of the learning phase of the previous step. Furthermore, a simple purely feedback controller is also considered in such a way to evaluate the ability of the proposed method to preserve the robot soft behavior w.r.t. a different control law. The employed feedback controller is:

$$r(t) = \int_0^{t_f} \left(\int_0^{t_f} 0.3\xi(t)\,dt + 2\xi(t) \right) dt + 2\xi(t), \; t \in [0, 2) \tag{28}$$

where $\xi(t) = \hat{q}(t) - q(t)$. To achieve performance comparable to the proposed algorithm the PII is heuristically tuned, resulting in high gains. Indeed, the maximum stiffness variation δ in (9) is $1.61 \frac{\text{N m}}{\text{rad}}$, that is much bigger w.r.t. the ILC ones. The stiffness input d_1 is equal to 0rad (minimum stiffness) for all the cases. The time required to converge was approximately 0.2h for each R value.

Results: The results of the first step are reported in Fig. 3. This shows the evolution of the error over iterations (computed as (26)) for the three R values. Lowering R the convergence rate increases. Indeed, Fig. 3 shows that from iteration 1 to iteration 100 the error decreases of 73%, 54% and 44% for

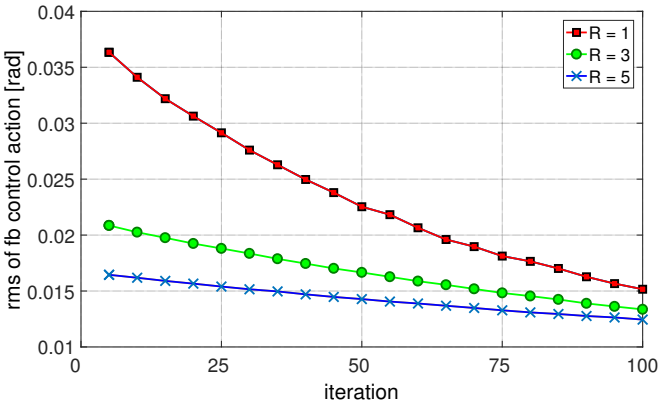


Figure 4: Experiment 1: root mean square of the feedback action over iterations for the three R values. The feedback contribution decreases over the iterations in each trials. The rms in the purely feedback controller is 0.5372rad.

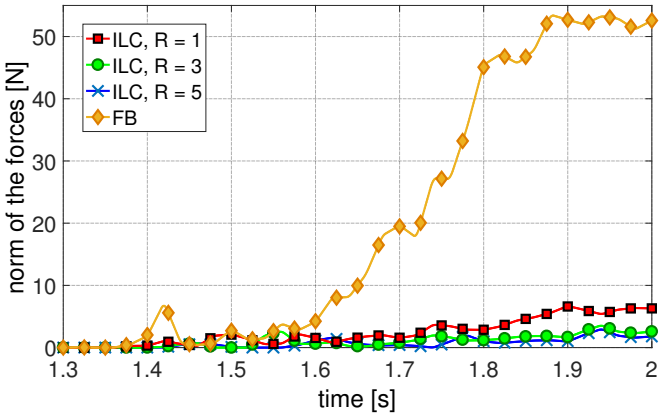


Figure 5: Experiment 1: evolution of the norm of the forces during and after the impact with the bar. The impact occurs approximately at 1.4s. The only control action that presents a force peak during the impact is the feedback one. In the ILC cases lower R values correspond to higher applied forces.

$R = 1$, $R = 3$ and $R = 5$ respectively. Furthermore, it is worth noting that lower R values correspond to lower error values at the first iteration, even though $r^0(t)$ is equal for the three R values. This is caused by the higher feedback gains K_{on}. On the other hand, Fig. 4 shows the root mean square of the feedback action exerted by the proposed controller at each iteration for the three R values. As expected, in the case $R = 1$ the feedback contribution is bigger w.r.t. the other two cases. Fig. 4 shows that the norm of the feedback control action decreases over the iterations (while the feedforward contribution increases). The results of the second step of the experiment are reported in Fig. 5. This shows the evolution of the norm of the force measured by the sensor during and after the impact. Note that the impact occurs approximately at 1.4s. As expected, the applied forces are lower when the feedback gains are lower (i.e. higher R). In particular, the purely feedback controller presents the higher applied forces, and it is the only controller presenting a force peak during the impact. This means that a high feedback controller should be carefully employed when a soft robot is involved, because it hinders the desired soft behavior.

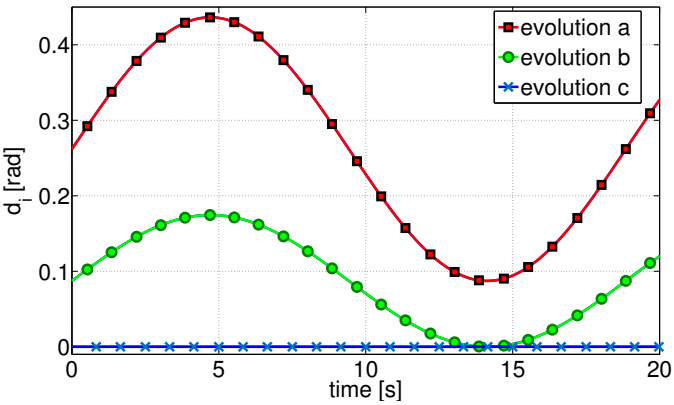


Figure 6: Experiment 2: stiffness input (d_i) evolution over time for the three different setups. Evolution 'a' is the one of the first qbmove for 1-dof case (see Fig. 2b), of the second qbmove for 2-dof case (see Fig. 2c) and of the third qbmove for 3-dof case (see Fig. 2d). Evolution 'b' is the one of the first qbmove for the 2-dof case and of the second qbmove for 3-dof case. Evolution 'c' is the one of the first qbmove for the 3-dof case.

B. Experiment 2

Experimental Setup: Three different setups are considered, consisting of serial chains of one, two and three qbmoves as shown in Fig. 2. In the 3-dof case a spring is added in parallel to cope with torque limitation issues. The spring is not included in the model, and thus it takes the role of an external disturbance for the algorithm. The reference trajectory for each joint is

$$\hat{q}_i(t) = (-1)^i \frac{\pi}{12} \cos(2t), \quad t \in [0, 20). \tag{29}$$

The stiffness input d for these experiments is time-varying and different for each qbmoves. This is done to show the ability of the algorithm to cope with time-varying inputs d. Fig. 6 shows the stiffness input d_i for each joint for the three setups. The maximum stiffness variation δ in (9) is imposed here as

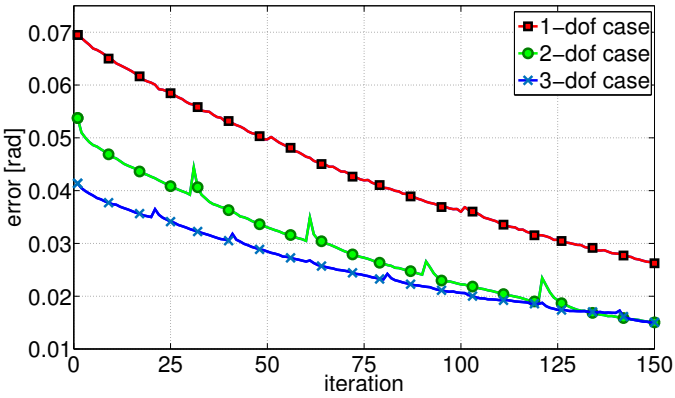


Figure 7: Experiment 2: evolution of the error over iterations (computed as (26)) for the three setups.

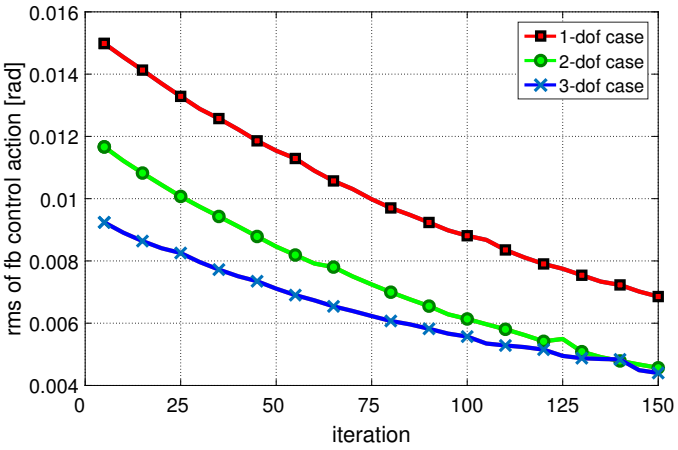


Figure 8: Experiment 2: root mean square of the feedback action over iterations for the three setups. The feedback contribution decreases over the iterations in each trials.

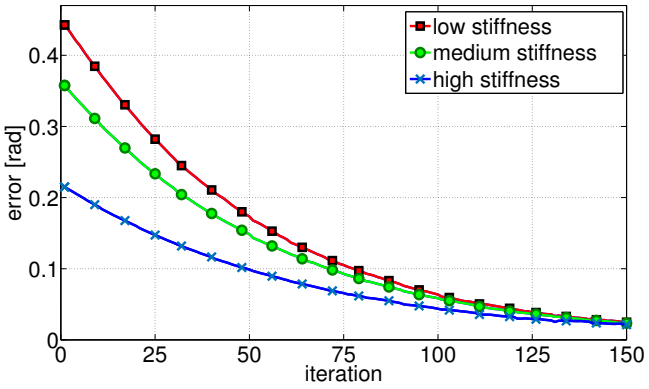


Figure 9: Experiment 3: evolution of the error over iterations (computed as (26)) for three different constant input stiffness values: $d_1 = 0$ rad, $d_1 = 0.17$ rad, $d_1 = 0.44$ rad, for the low, medium, and high stiffness case respectively.

$0.6\frac{\text{N m}}{\text{rad}}$, resulting in $R = 3$. The time required to converge was approximately 1h for each setup.

Results: Fig. 7 shows the evolution of error over iterations (computed as (26)) for the three setups. The proposed choice of $r^0(t)$ allows to achieve a rather small error already at the first execution. The learning process refines the control action further reducing error of more than 60% for all the considered setups. The minimum error can be observed for the 3-dof case, since unmodeled effects as static friction and hysteresis become negligible for higher deflections of the spring. Fig. 8 shows the root mean square of the feedback action exerted by the proposed controller at each iteration for the three setups. The feedback contribution decreases over the iterations while the feedforward contribution remains approximately constant.

C. Experiment 3

Experimental Setup: The term $\eta_i(q,\dot{q})$ in (14) collects system uncertainties not taken into account in the initial control action. This experiment aims to test the effectiveness of the ILC algorithm also in case of a major change in $\eta_i(q,\dot{q})$, caused by a relevant variation in the gravity torque. To test this

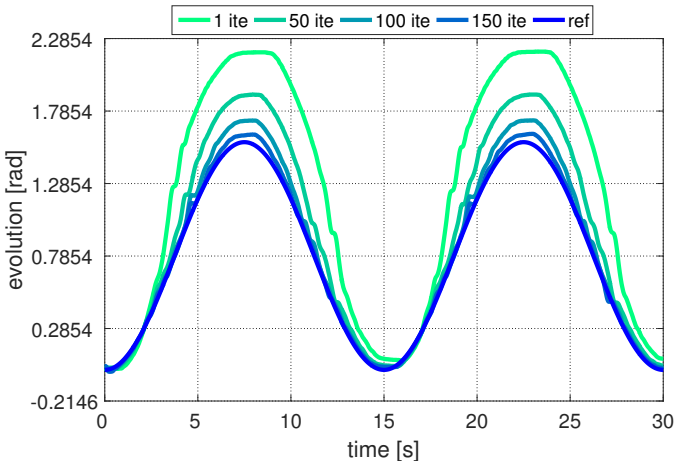


Figure 10: Experiment 3: Trajectory evolution over iterations for the low stiffness case. The algorithm is able to compensate for the strong variation of external torque caused by gravity.

condition, we impose the following reference trajectory for the robot depicted in Fig. 2b

$$\hat{q}_1(t) = -\frac{\pi}{4}\cos\left(\frac{4\pi}{30}t\right) + \frac{\pi}{4}, \quad t \in [0, 30) \tag{30}$$

around the dashed line depicted in Fig. 2b. Note that along that trajectory the gravity torque changes sign. Three value of constant stiffness input are also considered here, i.e. $d_1 =$ 0rad, $d_1 = 0.17$rad, $d_1 = 0.44$rad. In this experiment we impose $R = 1$, corresponding to a maximum stiffness variation δ in (9) of $0.33\frac{\text{N m}}{\text{rad}}$ in the low stiffness case, $0.43\frac{\text{N m}}{\text{rad}}$ in the medium stiffness case and $1.46\frac{\text{N m}}{\text{rad}}$ in the high stiffness case. The time required to converge was approximately 1.4h for each stiffness case.

Results: Fig. 9 shows the evolution of the error over iterations (computed as (26)) with low, medium and high constant stiffness input d. It is worth noting that the error at first iteration in this experiment is considerably bigger w.r.t. to the error at first iteration in Experiment 1 and 2. This is due to the fact that in Experiment 3 the gravity torque has a considerable change during the robot motion. Fig. 10 shows the time evolution of the link trajectory in four meaningful iterations for the low stiffness case, which exhibits the largest initial error. Results show that in 150 iterations the desired trajectory is tracked with an error reduction greater than 90% w.r.t. the initial error for all the cases.

D. Experiment 4

Experimental Setup: The goal of this experiment is twofold. First of all, we evaluate the ability of the algorithm to cope with a parallel structure where coupling terms are typically stronger w.r.t. a serial one: the robot is a 3-dof Delta (see Fig. 11) composed of three actuators connected to the end-effector through a parallel structure. Furthermore, we test the ability of the algorithm to converge in presence of impacts with the environment during the learning phase. We consider here a trajectory at the level of the end-effector (demonstrated to the robot by manually moving the end-effector along the

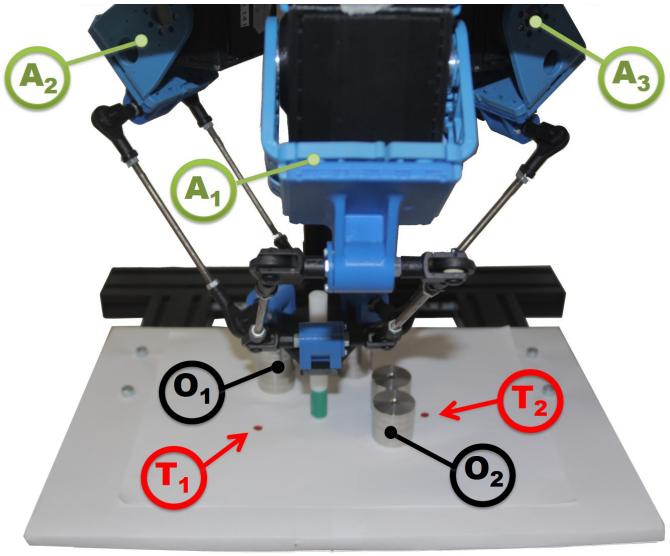


Figure 11: Experiment 4: delta robot used for the rest-to-rest experiment from T_1 (Target 1) to T_2 (Target 2). The red dots represent the target spots while the aluminum columns represent the obstacles (O_1 Obstacles 1, O_2 Obstacles 2). The robot has to move its end-effector between the columns of the Obstacle 1 and has to jump over Obstacle 2. A_1, A_2 and A_3 are the three actuators.

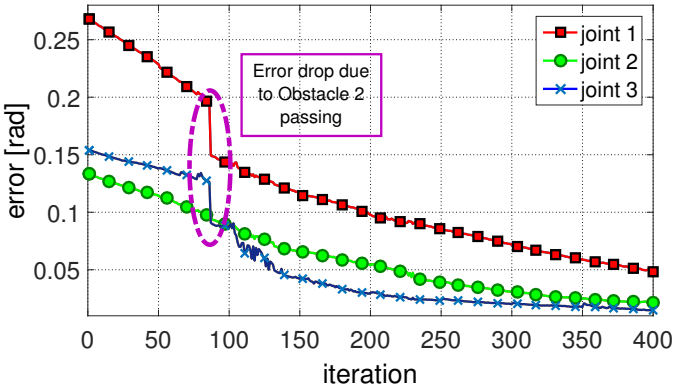


Figure 12: Experiment 4: evolution of the error over iterations for the three joints (note that it is not the error mean value between the joints) of the Delta robot. At iteration 87 and 106 there are two drops of the error since the robot learned how to pass Obstacle 2 and Obstacle 1, respectively.

desired trajectory): a rest-to-rest task through two obstacles, each consisting of two aluminum columns (O_1 and O_2 in Fig. 11). The demonstrated end-effector trajectory is to pass through Obstacle 1 and to jump over Obstacle 2 (as shown in Fig. 13, and in the attached video footage). In the replay phase, a standard (rigid) robot would follow the recorded path accurately under suitably high gain, but if the environment includes a human, or is changing, or anyhow we have a soft robot, high gain cannot be used. Thus, we set the input stiffness profile time-varying: the robot is stiff during the positioning over the target points (T_1 and T_2, marked as red dots in Fig. 11), so that the precision is improved, and it is soft during the obstacles passing phases to be adaptable to the external environment. In this experiment we use $R = 3$, corresponding to a maximum stiffness variation δ in (9) of $0.55\frac{\text{N m}}{\text{rad}}$. The time required to converge was approximately 2.1h.

Result: Fig. 13 shows the trajectory tracking improvement between the first and the last iteration. Initially the robot can neither pass through the columns nor jump over the barricade, failing to fulfill the task. At the end of the learning process the robot is able to successfully accomplish the task. Fig. 12 shows the error evolution over iterations. It is worth noting that at 87-th iteration the error drops significantly. This is due to the fact that the algorithm refines the control action on a level that allows the robot to pass through Obstacle 2, significantly improving the trajectory tracking performance.

V. Conclusions and Future Works

In this work, we presented a trajectory tracking controller for articulated soft robots that combines a low-gain feedback component, a rough initial estimation of the feedforward action and a learned refinement of that action. The proposed algorithm is designed to be independent from the kinematic structure of the robot, to maintain the robot soft behavior, and to be robust to external uncertainties as unexpected interactions with the environment. Various experimental setups were built to test the effectiveness of the controller in many working conditions, i.e. serial and parallel structure, different degrees of interaction with the external environment, different number of joints.

One of the goals of soft robotics is to design robot that are resilient, energy efficient and safe when interacting with the environment or any human beings. The proposed control technique, thanks to all its described features, allows to exploit the compliant behavior of any articulated soft robot, achieving simultaneously good performance. Unfortunately, any learned control action will be suited only for the given desired trajectory $\hat{q}(t)$ and stiffness parameter profile $d(t)$. A variation of any of these two will lower the tracking performance. Therefore, a new learning phase will be needed for every new task. This issue will be addressed in future works.

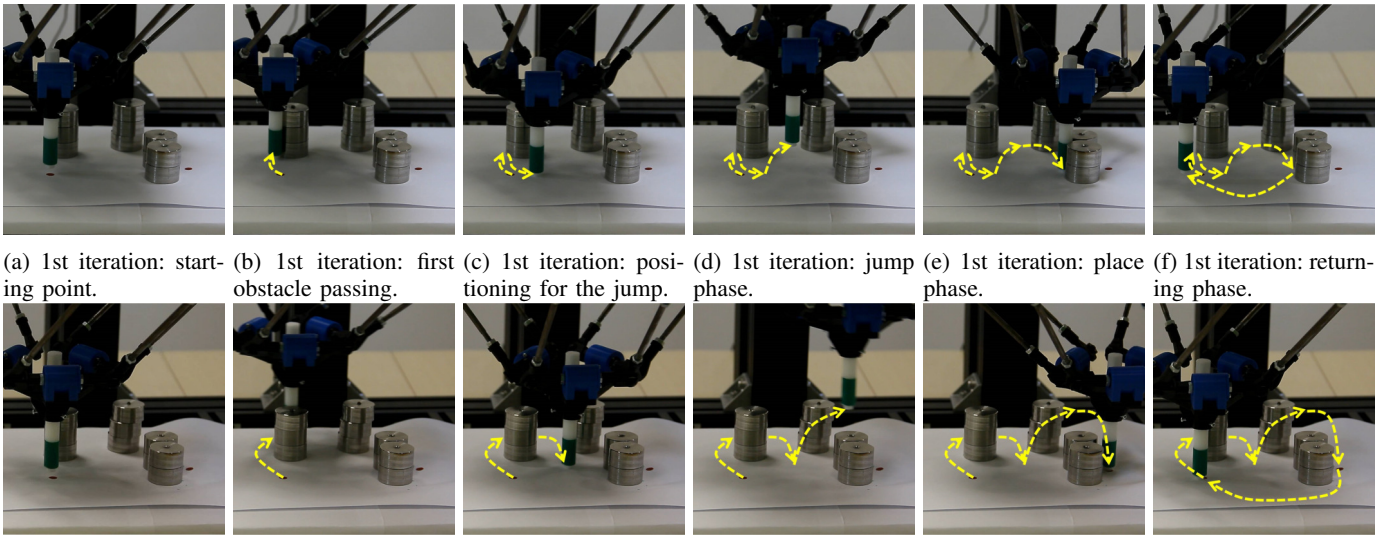
This work focused on articulated soft robots, where the system compliance is embedded in the robot joints. However, we believe that the issues discussed and faced in the present work could be useful also for continuously deformable soft robots. In first approximation, the presented results could be applied to a finite element approximation of the continuously deformable soft robots. However, some limitations have to be considered, thus future works will be devoted to expanding our analysis to such class of robots, testing and potentially extending the here proposed algorithm.

Appendix
Proofs of Propositions

In this section we proof all the propositions stated in section III.

Proposition 1. *If*

$$\left\| \left. \frac{\partial \psi}{\partial q} \right|_{q \equiv q_*} \right\| \leq \delta \left\| \left. \frac{\partial T(q-\psi, d)}{\partial q} \right|_{q \equiv q_*} \right\|^{-1} \tag{31}$$



(a) 1st iteration: starting point. (b) 1st iteration: first obstacle passing. (c) 1st iteration: positioning for the jump. (d) 1st iteration: jump phase. (e) 1st iteration: place phase. (f) 1st iteration: returning phase.

(g) 400th iteration: starting point. (h) 400th iteration: first obstacle passing. (i) 400th iteration: positioning for the jump. (j) 400th iteration: jump phase. (k) 400th iteration: place phase. (l) 400th iteration: returning phase.

Figure 13: First and last iteration photo sequence of the Delta experiment. (a, g): The robot should stand over the red dot (Target 1). (b, h): The robot has to pass through the two columns of Obstacle 1. In the first iteration it can not do it and it collides with one of the columns. (c, i): The robot prepares itself to jump over Obstacle 2. (d, j): the robot has to jump over the obstacle. In the first iteration the jump is too small and the robot fails. (e, k): The robot should position itself over the second target. In the first iteration, since it failed the jump, the robot stops against the obstacle. (f, l): The robot returns to the starting position.

then (5) *holds.*

Proof. By using the chain rule, it is possible to rewrite the first term of (5) as

$$\left\| \left.\frac{\mathrm{d}T(q-r,d)}{\mathrm{d}q}\right|_{q\equiv r} - \left.\frac{\mathrm{d}T(q-\psi,d)}{\mathrm{d}q}\right|_{q\equiv q_*} \right\| = \left\| \left.\frac{\partial T(q-r,d)}{\partial q}\right|_{q\equiv r} - \left.\frac{\partial T(q-\psi,d)}{\partial q}\right|_{q\equiv q_*} \left(1 - \left.\frac{\partial \psi}{\partial q}\right|_{q\equiv q_*}\right) \right\|. \tag{32}$$

Note that from the definition of q_* and r, the following equation holds

$$\left.\frac{\partial T(q-r,d)}{\partial q}\right|_{q\equiv r} = \left.\frac{\partial T(q-\psi,d)}{\partial q}\right|_{q\equiv q_*}, \tag{33}$$

that, together with the Cauchy Schwarz matrix axiom, yields to

$$\left\| \left.\frac{\partial T(q-r,d)}{\partial q}\right|_{q\equiv r} - \left.\frac{\partial T(q-\psi,d)}{\partial q}\right|_{q\equiv q_*} \left(1 - \left.\frac{\partial \psi}{\partial q}\right|_{q\equiv q_*}\right) \right\| = \left\| \left.\frac{\partial T(q-\psi,d)}{\partial q}\right|_{q\equiv q_*} \left.\frac{\partial \psi}{\partial q}\right|_{q\equiv q_*} \right\| \leq \left\| \left.\frac{\partial T(q-\psi,d)}{\partial q}\right|_{q\equiv q_*} \right\| \left\| \left.\frac{\partial \psi}{\partial q}\right|_{q\equiv q_*} \right\| \tag{34}$$

that brings to the thesis by directly applying the hypothesis. □

Lemma 1. *If the control algorithm is decentralized, i.e.* $\frac{\partial \psi}{\partial q}$ *is diagonal, and if*

$$\left\| \left.\frac{\partial \psi_i}{\partial q_i}\right|_{q\equiv q_*} \right\| \leq \delta \left\| \left.\frac{\partial T(q-\psi,d)}{\partial q}\right|_{q\equiv q_*} \right\|^{-1} \quad \forall\, i \tag{35}$$

where $\frac{\partial \psi_i}{\partial q_i}$ *is the* $i-th$ *diagonal element, then* (5) *holds.*

Proof. By hypothesis $\frac{\partial \psi}{\partial q}$ is diagonal, thus (see e.g. [20])

$$\left\| \frac{\partial \psi}{\partial q} \right\| = \max_i \left| \frac{\partial \psi_i}{\partial q_i} \right|. \tag{36}$$

Combining (34) and (36) yields

$$\left\| \left.\frac{\mathrm{d}T(q-r,d)}{\mathrm{d}q}\right|_{q\equiv r} - \left.\frac{\mathrm{d}T(q-\psi,d)}{\mathrm{d}q}\right|_{q\equiv q_*} \right\| \leq \left\| \left.\frac{\partial T(q-\psi,d)}{\partial q}\right|_{q\equiv q_*} \right\| \max_i \left\| \left.\frac{\partial \psi_i}{\partial q_i}\right|_{q\equiv q_*} \right\| \tag{37}$$

which implies the thesis. □

Proposition 2. *If* $K_{\mathrm{on},i}$ *is as in* (18)*, then*

$$\forall \gamma \geq 0 \quad \exists R > 0 \; s.t. \; ||K_{\mathrm{on},i}|| \leq \gamma. \tag{38}$$

Proof. For the sake of readability in this proof we omit the index i. We start noting that if $S(t)$ is solution of (19), with the boundary constraint $S(t_\mathrm{f}) = \emptyset$, then $S(t)$ is bounded in norm $\forall t \in [0, t_\mathrm{f})$, $\forall R > 0$. This derives from many classic results in optimal control theory (see e.g. [21], [22]). Thus, it is always possible to bound $||S||$ with a sufficiently large constant $\Sigma > 0$. Hence, from (18)

$$||K_{\mathrm{on}}|| = \left\| \frac{B(t)^T S(t)}{R} \right\| \leq \frac{1}{R} ||B(t)^T|| \cdot ||S(t)|| \leq \frac{1}{R} (B_{\max} \Sigma), \tag{39}$$

where, from (15), $B_{\max} = \max_{t\in[0,t_f)} \frac{\sigma(t)}{I}$ for the 2-norm. Note that Σ is known by the evaluation of S in (19). Thus, $||K_{\text{on}}||$ is always upper bounded by an hyperbolic function of R, from which the thesis by choosing $R = \frac{B_{\max}\Sigma}{\gamma}$. □

Proposition 3. *The feedback rule in* (18) *fulfills the ILC convergence condition* (16) *for all* $R > 0$.

Proof. Rewriting (16) for the considered feedback control yields

$$\left| \frac{R}{R + B_i(t)^T S_i(t) B_i(t)} \right| < 1\,,\ \forall t \in [0, t_f)\,,\ \forall i\,. \tag{40}$$

which is always true if $B_i(t)^T S_i(t) B_i(t) \in \mathbb{R}$ is positive. This is true if $S_i(t)$ is positive definite, which is the case since Q is positive definite in $t \in [0, t_f)$ [23]. □

Proposition 4. *The convergence condition* (21) *is fulfilled by the following decentralized ILC gain* $\forall \varepsilon \in [0,1)$ *and* $\forall\ \ \Gamma_i^T(t) \in \ker\{B_i^T(t)\}$

$$K_{\text{off},i}(t) = (1+\varepsilon) B_i(t)^\dagger + \Gamma_i(t)\,, \tag{41}$$

where $B_i(t)^\dagger$ *is the Moore-Penrose pseudoinverse of the matrix* $B_i(t)$ *in* (14).

Proof. The thesis follows directly by substitution

$$K_{\text{off,i}}(t) B_i(t) = ((1+\varepsilon) B_i(t)^\dagger + \Gamma_i(t)) B_i(t) = 1 + \varepsilon. \tag{42}$$

Substituting (42) in (21) yields to $|\varepsilon| < 1$, which is always true as assumed by hypothesis. □

REFERENCES

[1] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics," *Trends in biotechnology*, vol. 31, no. 5, pp. 287–294, 2013.

[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*, vol. 15, no. 3, pp. 20–30, 2008.

[3] 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*, vol. 24, no. 1, pp. 48–56, 2017.

[4] G. Buondonno and A. De Luca, "Efficient computation of inverse dynamics and feedback linearization for vsa-based robots," in *IEEE Robotics and Automation Letters (RA-L) paper presented at the 2016 IEEE International Conference on Robotics and Automation (ICRA) Stockholm, Sweden, May 16-21, 2016*, 2016.

[5] 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 *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 5487–5494, IEEE, 2009.

[6] F. Petit, A. Daasch, and A. Albu-Schaffer, "Backstepping control of variable stiffness robots," *Control Systems Technology, IEEE Transactions on*, vol. 23, no. 6, pp. 2195–2202, 2015.

[7] C. Della Santina, M. Bianchi, G. Grioli, F. Angelini, M. Catalano, M. Garabini, and A. Bicchi, "Controlling soft robots: balancing feedback and feedforward elements," *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 75–83, 2017.

[8] D. Bristow, M. Tharayil, A. G. Alleyne, *et al.*, "A survey of iterative learning control," *Control Systems, IEEE*, vol. 26, no. 3, pp. 96–114, 2006.

[9] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic systems*, vol. 1, no. 2, pp. 123–140, 1984.

[10] A. Tayebi, "Adaptive iterative learning control for robot manipulators," *Automatica*, vol. 40, no. 7, pp. 1195–1203, 2004.

[11] B. Siciliano and O. Khatib, "Springer handbook of robotics," ch. 21, Springer, 2016.

[12] G. Adams and M. Nosonovsky, "Contact modelingforces," *Tribology International*, vol. 33, no. 5, pp. 431–442, 2000.

[13] G. Grioli, S. Wolf, M. Garabini, M. Catalano, E. Burdet, D. Caldwell, R. Carloni, W. Friedl, M. Grebenstein, M. Laffranchi, *et al.*, "Variable stiffness actuators: The user?s point of view," *The International Journal of Robotics Research*, vol. 34, no. 6, pp. 727–743, 2015.

[14] G. Pratt, M. M. Williamson, *et al.*, "Series elastic actuators," in *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1, pp. 399–406, IEEE, 1995.

[15] B. Vanderborght, A. Albu-Schäffer, A. Bicchi, E. Burdet, D. G. Caldwell, R. Carloni, M. Catalano, O. Eiberger, W. Friedl, G. Ganesh, *et al.*, "Variable impedance actuators: A review," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1601–1614, 2013.

[16] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, "Robotics: modelling, planning and control," ch. 8, Springer Science & Business Media, 2009.

[17] P. Ouyang, B. Petz, and F. Xi, "Iterative learning control with switching gain feedback for nonlinear systems," *Journal of Computational and Nonlinear Dynamics*, vol. 6, no. 1, p. 011020, 2011.

[18] P.-i. Pipatpaibul and P. Ouyang, "Application of online iterative learning tracking control for quadrotor uavs," *ISRN robotics*, vol. 2013, 2013.

[19] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[20] K. B. Petersen, M. S. Pedersen, *et al.*, "The matrix cookbook," *Technical University of Denmark*, vol. 7, p. 15, 2008.

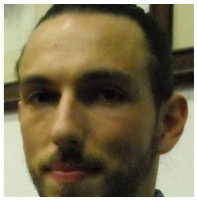
[21] R. E. Kalman *et al.*, "Contributions to the theory of optimal control," *Bol. Soc. Mat. Mexicana*, vol. 5, no. 2, pp. 102–119, 1960.

[22] D. Jacobson, "New conditions for boundedness of the solution of a matrix riccati differential equation," *Journal of Differential Equations*, vol. 8, no. 2, pp. 258–263, 1970.

[23] B. D. Anderson and J. B. Moore, *Linear optimal control*, vol. 197. Prentice-hall Englewood Cliffs, 1971.

Franco Angelini received the B.S degree in computer engineering and the M.S. cum laude in automation and robotics engineering from University of Pisa. He is a fellow of the Italian Institute of Technology and he is currently working toward the Ph.D. degree in robotics at Research Center E.Piaggio of the University of Pisa. His main research interest is control of soft robotic systems.

Cosimo Della Santina received the B.S degree cum laude in computer engineering and the M.S. cum laude in automation and robotics engineering from University of Pisa. He is is currently working towards the Ph.D. degree in robotics at Research Center E.Piaggio of the University of Pisa. His main research interests include model based design, sensing, and control of soft robots and soft hands.



Manolo Garabini received the Laurea degree in mechanical engineering and the doctoral degree in robotics from the E. Piaggio Research Center at the University of Pisa. His main research interests are control systems and variable impedance actuation. He is a member of the IEEE.

Matteo Bianchi is currently working as Assistant Professor at the University of Pisa - Centro di Ricerca E. Piaggio and Research Affiliate at Mayo Clinic (Rochester, MN - US). He is the local principal investigator of the University of Pisa for the EU funded project SOFTPRO and serves as co-chair of the IEEE Robotics and Automation Society (RAS) Technical Committee on Robotic Hands, Grasping and Manipulation. His research interests include haptic interface design and control; medical and assistive robotics; advanced human-robot interaction; human and robotic hands: sensing and control; human-inspired control for soft robots. He is an author of contributions to international conferences and journals. He serves as reviewer and member of the editorial board and organizing committee of international journals and conferences. He is co-editor of the book Human and Robot Hands, Springer International Publishing. He is recipient of several international awards including the Best Paper Award at the IEEE-RAS Haptics Symposium 2016.

Gian Maria Gasparri received the Laurea degree cum laude in automation and robotics engineering from University of Pisa. He got his Ph.D. in robotics automation and Bioengineering from University of Pisa in 2016. His main research topics are soft robots and robotic locomotion.

Giorgio Grioli is a Researcher in the Italian Institute of Technology where he investigates design, modelling and control of soft robotic systems applied to augmentation of, rehabilitation of and interaction with the human. He got his PhD in Robotics, Automation and Engineering from University of Pisa in 2011. He is author of more than 60 articles (both journal papers and conference proceedings) in the fields of soft robotic actuation, robot hand design and haptics. He serves as Associated Editor for ICRA and ICORR and is currently co-editing a special issue of the Actuators journal on "Variable Stiffness and Variable Impedance Actuators".

Manuel G. Catalano received the Laurea degree in mechanical engineering and the doctoral degree in robotics from the E. Piaggio Research Center at the University of Pisa. He is currently a Researcher at the Italian Institute of Technology and a collaborator of Centro di Ricerca E. Piaggio of the University of Pisa. His main research interests are in the design of Soft Robotic systems, Human Robot Interaction and Prosthetics. In 2014, he was the winner of the Georges Giralt PhD Award, the prestigious annual European award given for the best PhD thesis by euRobotics AISBL.

Antonio Bicchi is a Professor of Automatic Control in the Department of Information Engineering (DII) and Centro E. Piaggio of the University of Pisa, where he leads the robotics research group since 1990. Since 2009 he is a Senior Researcher at the Italian Institute of Technology, Genoa, where he leads the Soft Robotics for Human Cooperation and Rehabilitation Research Line. His main research interests are in the field of robotics, haptics, and automatic control. He has published more than 400 papers on international journals, books, and refereed conferences. He has been granted an ERC Advanced Grant in 2011, and is an IEEE Fellow since 2005.