




Decentralized Trajectory Tracking Control for Soft Robots Interacting With the Environment

Franco Angelini , *Student Member, IEEE*, Cosimo Della Santina , *Student Member, IEEE*, Manolo Garabini, Matteo Bianchi , *Member, IEEE*, Gian Maria Gasparri , Giorgio Grioli , *Member, IEEE*, Manuel Giuseppe Catalano , *Member, IEEE*, and Antonio Bicchi , *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 using 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, iterative learning control (ILC), motion control.

Manuscript received August 31, 2017; accepted February 4, 2018. Date of publication May 31, 2018; date of current version August 15, 2018. This paper was recommended for publication by Associate Editor M. Schwager and Editor T. Murphey upon evaluation of the reviewers' comments. This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement 645599 (Soma), Grant Agreement 688857 (SoftPro), and Grant Agreement 732737 (ILIAD). (*Corresponding author: Franco Angelini.*)

F. Angelini and A. Bicchi are with the Centro di Ricerca "Enrico Piaggio," Università di Pisa, Pisa 56126, Italy, with the Soft Robotics for Human Cooperation and Rehabilitation, Fondazione Istituto Italiano di Tecnologia, Genova 16163, Italy, and also with the Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa 56126, Italy (e-mail: frncangelini@gmail.com; antonio.bicchi@unipi.it).

C. Della Santina and M. Bianchi are with the Centro di Ricerca "Enrico Piaggio," Università di Pisa, Pisa 56126, Italy, and also with the Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa 56126, Italy (e-mail: cosimodellasantina@gmail.com; matteobianchi23@gmail.com).

M. Garabini and G. M. Gasparri are with the Centro di Ricerca "Enrico Piaggio," Università di Pisa, Pisa 56126, Italy (e-mail: manolo.garabini@gmail.com; gasparrigianmaria@gmail.com).

G. Grioli and M. G. Catalano are with the Soft Robotics for Human Cooperation and Rehabilitation, Fondazione Istituto Italiano di Tecnologia, Genova 16163, Italy (e-mail: giorgio.grioli@gmail.com; manuel.catalano@iit.it).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2018.2830351

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 recent 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: 1) robots that take inspiration mostly from invertebrate animals [1] and are accordingly built with continuously deformable elements and 2) robots inspired by the muscle-skeletal system of vertebrates, with compliance concentrated in the robot joints [2], [3]. 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.

In the literature, several trajectory tracking solutions were proposed for soft robots. Feedback linearization was profitably employed in [4] and [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. Second, feedback control laws have some fundamental limitations when they are applied to soft robots. Indeed, Della Santina *et al.* [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 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] and [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, neither systematic design nor analysis tools were provided to actually synthesize an iteratively learned feedforward control with convergence and stability guarantees.

In this paper, 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 the considered scenarios.

This paper is organized as follows. 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 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 an N -joint articulated soft robot with $N_m \geq N$ motors introduced in [11] as

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

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^N$ are the vectors of generalized joint positions, velocities, and accelerations, respectively, whereas $\theta, \dot{\theta}, \ddot{\theta} \in \mathbb{R}^{N_m}$ 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, $J \in \mathbb{R}^{N_m \times N_m}$ is the motor inertia matrix, and $T_{\text{ext}}(q, \dot{q}) \in \mathbb{R}^N$ 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 $F_m \in \mathbb{R}^{N_m}$ are the motor torques.

In this paper, we use a simplified model, introducing the following further assumptions.

- 1) Motor dynamics (1) 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.
- 2) Interactions with the environment can be modeled with a suitable smooth force field [12].
- 3) There exists a change of coordinates between the motor positions θ and two set of variables $r \in \mathbb{R}^N$ and

$d \in \mathbb{R}^{N_m - N}$ such that $\frac{\partial V(q, \theta)^T}{\partial q} = T(q - r, d)$. Here, r can be regarded as a joint reference position, whereas 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 ($N_m = N$), whereas for a variable stiffness actuator (VSA) [15], d indicates the joint cocontraction level ($N_m = 2N$).

Hence, the considered model of an N -joint articulated soft robot is

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

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, whereas 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],

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

where $i = 1, \dots, N$, $[q_i \ \dot{q}_i]^T$ 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 , and 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 $\hat{q} : [0, t_f] \rightarrow \mathbb{R}^N$, with all its time derivatives, and a stiffness adjusting variables d , the control objective is to derive an opportune control action $r : [0, t_f] \rightarrow \mathbb{R}^N$ able to regulate system (3) on \hat{q} in the whole control interval $[0, t_f]$. Other goals that we set out for our control design are as follows.

- 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.,

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

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 $[I_i, \beta_i, \text{ and } \tau_i]$ 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.

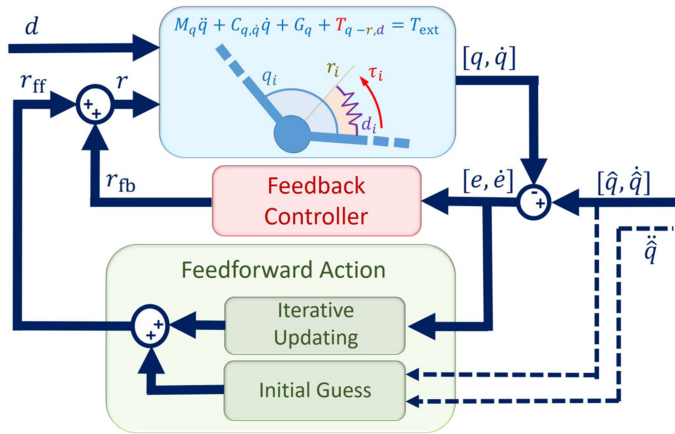


Fig. 1. Schematic representation of the control architecture. $[q \ \dot{q}]$ are the Lagrangian variables and $\hat{q}, \dot{\hat{q}}, \ddot{\hat{q}}$ their desired evolutions. $[e \ \dot{e}]$ is the tracking error. M , C , and G are the inertia, centrifugal, and potential field terms, T is the spring 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.

- 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 form¹ [8]

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

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

¹Note 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 paper, we will use $\alpha = 1$ to match the chosen convergence condition.

defined as

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

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

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

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 “online” and “offline,” respectively. Thus, the “online” term is the one computed during the trial execution (feedback component), whereas the “offline” term is the one computed between 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 (see Section III-A). Goal ii) is achieved considering decentralized gains (see 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

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

then (5) holds.

It is worth noting that feedforward action does not affect this condition, since it does not depend on q . This suggests favoring 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

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

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 the 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

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

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 the presence of unknown 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] and [18], where the sufficient conditions are imposed separately for the online and offline 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 conditions are as follows:

$$\| (I + K_{\text{on}}(t)H(t))^{-1} \| < 1 \quad \forall t \in [0, t_f] \quad (12)$$

$$\| I - K_{\text{off}}(t)H(t) \| < 1 \quad \forall t \in [0, t_f]. \quad (13)$$

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

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

where $e_i = [\hat{q}_i - q_i, \dot{\hat{q}}_i - \dot{q}_i]^T$ 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

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

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

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

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, Ch. 5])

$$\int_0^{t_f} e_i^T Q e_i + R u_i^2 dt \quad (17)$$

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

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

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

$$\dot{S}_i = -S_i A_i - A_i^T S_i + S_i B_i R^{-1} B_i^T S_i - Q \quad (19)$$

with the boundary constraint $S_i(t_f) = \emptyset$. 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 parameters 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

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

Thus, condition (9) can always be fulfilled by choosing $\gamma = \delta \left\| \frac{\partial T(q - \psi, d)}{\partial q} \Big|_{q=q^*} \right\|^{-1}$, 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 decoupled system (14) is

$$|1 - K_{\text{off},i}(t)B_i(t)| < 1 \quad \forall t \in [0, t_f] \quad (21)$$

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 $\forall \epsilon \in [0, 1)$ and $\forall \Gamma_i^T(t) \in \ker\{B_i^T(t)\}$

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

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 $\epsilon = 0.9$ to provide a good tradeoff between ILC convergence rate and stability.

Because of (15) and $\Gamma_i^T(t) \in \ker\{B_i^T(t)\}$, it follows that $\Gamma_i(t) = [K_{\text{poff},i}(t) \ 0]$, 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 $K_{\text{on},i}$

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

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

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

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 $\epsilon = 0.9$. $Q \in \mathbb{R}^{2 \times 2}$ 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 Propositions 3 and 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.

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.

Algorithm 1: Control Procedure Pseudo-Code.

```

1: procedure Initialization
2:   Set( $\hat{q}(t)$ ,  $\dot{\hat{q}}(t)$ ,  $\ddot{\hat{q}}(t)$ )           ▷ Desired trajectory
3:   Set( $d(t)$ )                           ▷ Stiffness input parameter
4:   Set( $Q, R$ )                             ▷ Control weight parameter
5:   Compute( $r^0(t)$ )                         ▷(11)
6:   Evaluate( $K_{\text{on}}(t)$ )                       ▷(18), (19)
7:   Evaluate( $K_{\text{off}}(t)$ )                     ▷(22), (23)
8: procedure Learning
9:    $k \leftarrow 1$ 
10:   $e^{k-1}(t) \leftarrow 0$ 
11:  do
12:    Run_Trial( $r^{k-1}$ ) ▷ Note:  $r^k$  is computed on-line
13:    Store( $e^k, r^k$ )
14:    Update( $r^k$ )           ▷ Off-line update:  $r^k + K_{\text{off}} e^k$ 
15:     $k \leftarrow k + 1$ 
16:  while  $e^{k-1} > \text{threshold}$ 

```

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 nonlinear 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 b. The *qbmove* spring characteristic τ_i in (4) is

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

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), whereas $a = 6.7328 \frac{1}{\text{rad}}$, $k = 0.0222 \text{ N}\cdot\text{m}$, and $m = 0.5 \frac{\text{N}\cdot\text{m}}{\text{rad}}$ 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 dependence (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, whereas 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, whereas Δ_i is estimated as the torque needed to arrange the robot in the initial condition. In all the experiments, Q is set with diagonal elements 1 and 0.01.

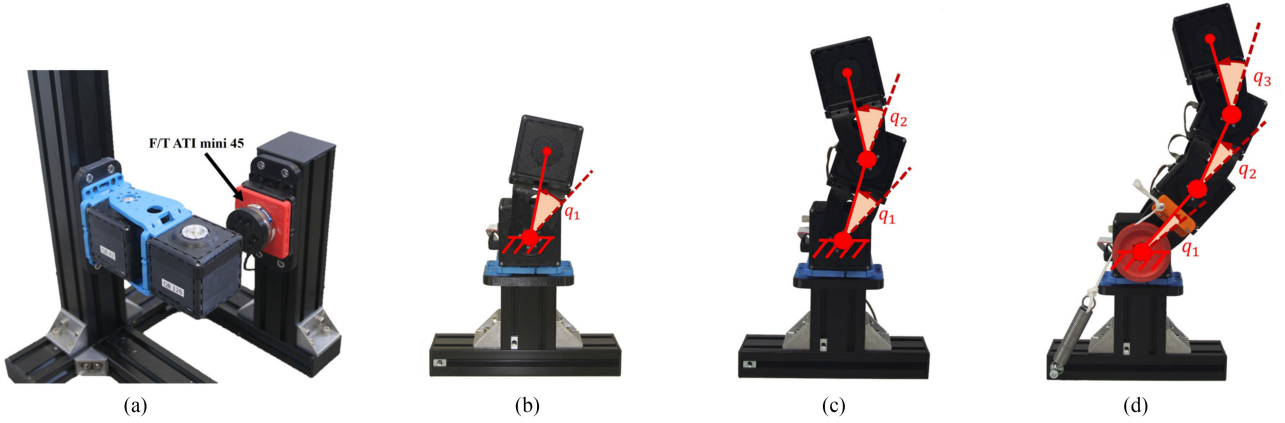


Fig. 2. Experimental setups and reference frames. (a) 1-dof planar configuration (Experiment 1; Section IV-A). (b) After the learning phase, a bar (with a six-axis force/torque ATI mini 45 mounted on it) is placed next to the robot. 1-dof configuration (Experiments 2 and 3; Sections IV-B and IV-C). (c) 2-dof configuration (Experiment 2; Section IV-B). (d) 3-dof configuration (Experiment 2; Section IV-B): A parallel spring is included in this setup to avoid that the torque required for 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.

In the next sections, we will employ

$$E(k) \triangleq \frac{\sum_{i=1}^N \int_0^{t_f} |\hat{q}_i(t) - q_i^k(t)| dt}{N t_f} \quad (26)$$

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), whereas $q_i^k(t)$ is the i th joint position measured by the encoder placed at the i th output shaft at the 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 $(t_a + t_f + t_{off}) \times n_k$, where n_k is the number of performed iterations, t_f is the task terminal time, t_a is the time needed to arrange the robot in the initial condition, and t_{off} is the time needed to compute (offline) 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 $Q = \text{diag}([1, 0.01])$. 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) tend to stiffen the robot (as theoretically described in Section III-A).

The experimental setup is composed of a planar 1-dof (degree of freedom) soft robot and a force sensor (six-axis force/torque ATI mini 45) mounted on a bar fixed to the frame [see Fig. 2(a)]. 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

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

This is a smoothed ramp spanning from 0 to 0.7854 rad 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 $\delta = 0.33 \frac{\text{N}\cdot\text{m}}{\text{rad}}$, to $R = 3$ corresponds $\delta = 0.12 \frac{\text{N}\cdot\text{m}}{\text{rad}}$, and to $R = 5$ corresponds $\delta = 0.08 \frac{\text{N}\cdot\text{m}}{\text{rad}}$. Afterward, 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. 2(a)]. 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 defined as follows:

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

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

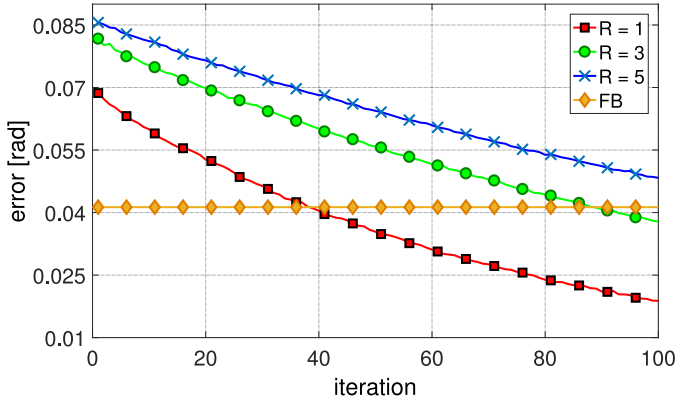


Fig. 3. Experiment 1: evolution of the error over iterations [computed as (26)] for the three R values. Lower R values correspond to lower initial error and faster convergence rate. The orange horizontal line is the error with the PI controller.

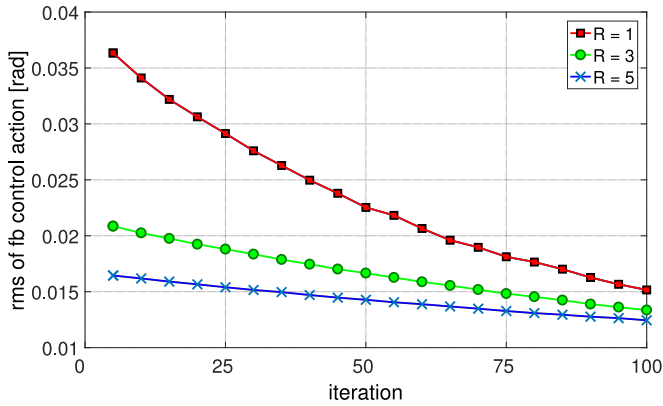


Fig. 4. Experiment 1: root mean square of the feedback action over iterations for the three R values. The feedback contribution decreases over the iterations in each trials. The rms in the purely feedback controller is 0.5372 rad.

rate increases. Indeed, Fig. 3 shows that from iterations 1 to 100, the error decreases are of 73%, 54%, and 44% for $R = 1$, $R = 3$, 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 (whereas 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 1.4 s. 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

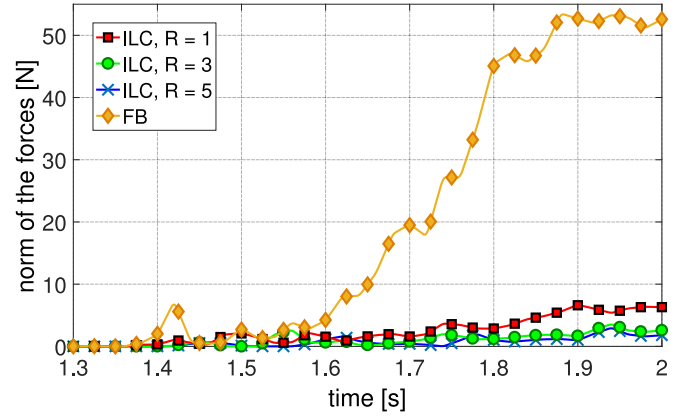


Fig. 5. Experiment 1: evolution of the norm of the forces during and after the impact with the bar. The impact occurs approximately at 1.4 s. The only control action that presents a force peak during the impact is the feedback one. In the ILC cases, lower R values correspond to higher applied forces.

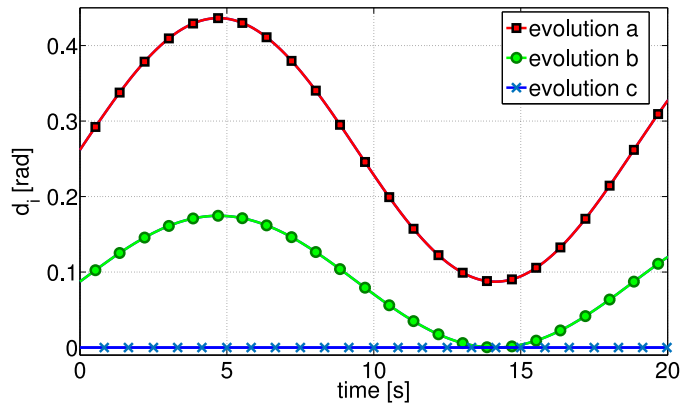


Fig. 6. Experiment 2: stiffness input (d_i) evolution over time for the three different setups. Evolution “a” is the one of the first qbmove for 1-dof case [see Fig. 2(b)], of the second qbmove for 2-dof case [see Fig. 2(c)], and of the third qbmove for 3-dof case [see Fig. 2(d)]. Evolution “b” is the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

controller should be carefully employed when a soft robot is involved, because it hinders the desired soft behavior.

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

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

The stiffness input d for these experiments is time varying and different for each qbmove. 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

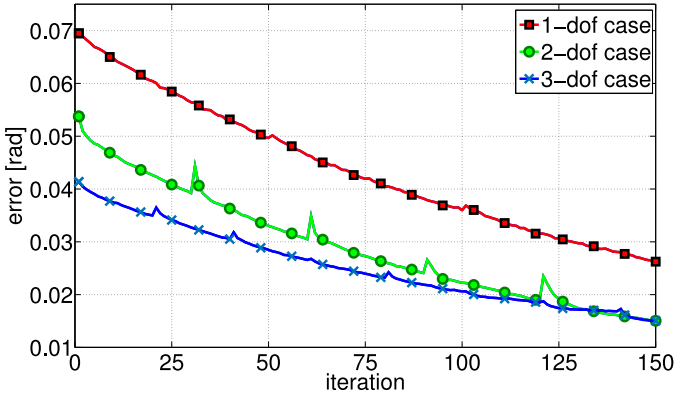


Fig. 7. Experiment 2: evolution of the error over iterations [computed as (26)] for the three setups.

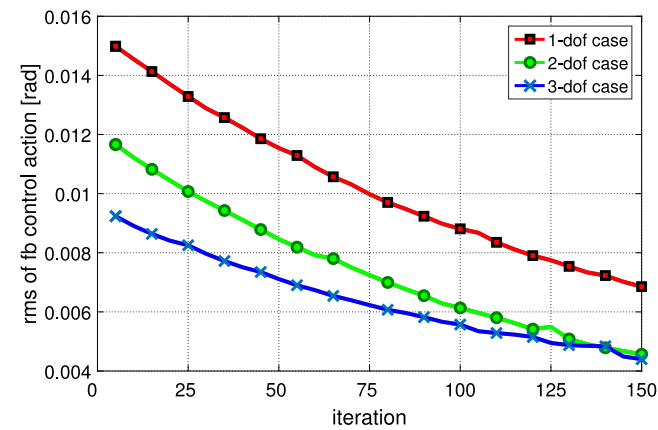


Fig. 8. Experiment 2: root mean square of the feedback action over iterations for the three setups. The feedback contribution decreases over the iterations in each trial.

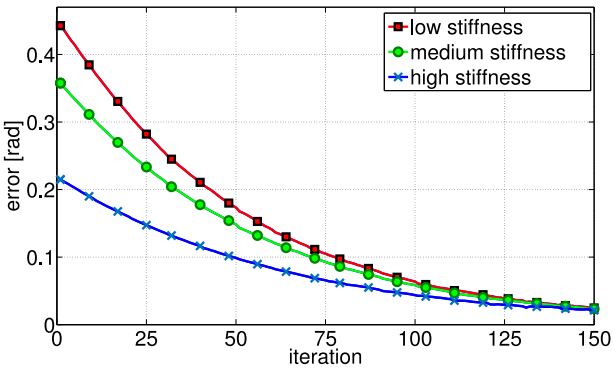


Fig. 9. Experiment 3: evolution of the error over iterations [computed as (26)] for three different constant input stiffness values: $d_1 = 0$ rad, $d_1 = 0.17$ rad, $d_1 = 0.44$ rad, for the low, medium, and high stiffness case, respectively.

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

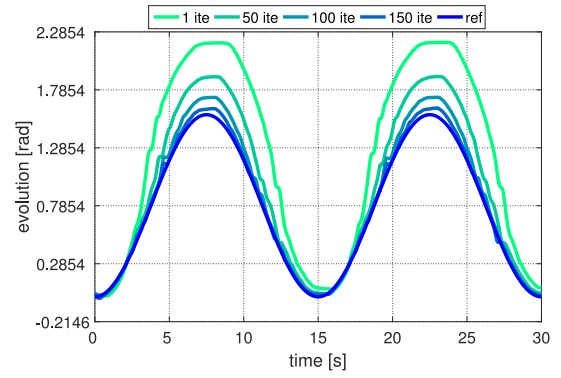


Fig. 10. Experiment 3: trajectory evolution over iterations for the low stiffness case. The algorithm is able to compensate for the strong variation of external torque caused by gravity.

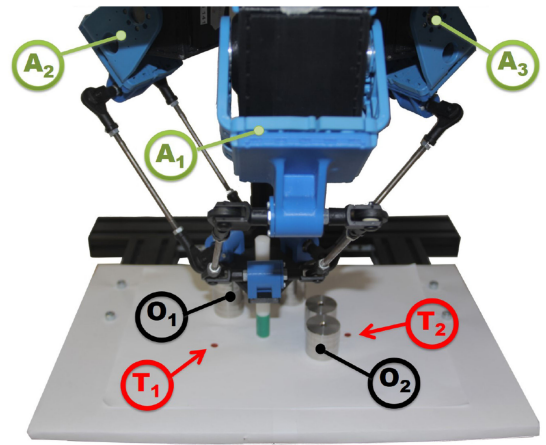


Fig. 11. Experiment 4: Delta robot used for the rest-to-rest experiment from T_1 (Target 1) to T_2 (Target 2). The red dots represent the target spots, whereas the aluminum columns represent the obstacles (O_1 Obstacle 1, O_2 Obstacle 2). The robot has to move its end-effector between the columns of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

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

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

around the dashed line depicted in Fig. 2(b). Note that along that trajectory, the gravity torque changes sign. Three values of

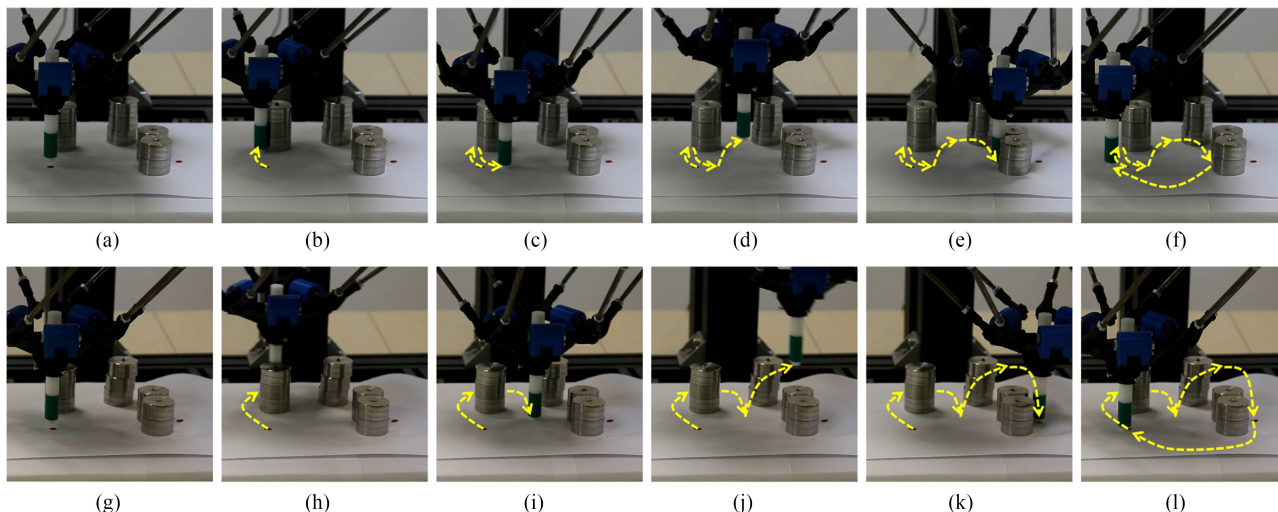


Fig. 12. First and last iteration photo sequence of the Delta experiment. (a), (g): The robot should stand over the red dot (Target 1). (b), (h): The robot has to pass through the two columns of Obstacle 1. In the first iteration, it cannot do it and it collides with one of the columns. (c), (i): The 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. (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.

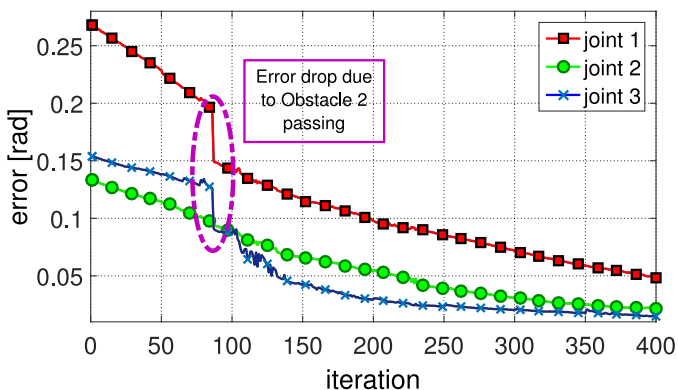


Fig. 13. Experiment 4: evolution of the error over iterations for the three joints (note that it is not the error mean value between the joints) of the Delta robot. At iterations 87 and 106, there are two drops of the error since the robot learned how to pass Obstacles 2 and 1, respectively.

constant stiffness input are also considered here, i.e., $d_1 = 0$ rad, $d_1 = 0.17$ rad, and $d_1 = 0.44$ rad. In this experiment, we impose $R = 1$, corresponding to a maximum stiffness variation δ in (9) of $0.33 \frac{\text{N}\cdot\text{m}}{\text{rad}}$ in the low stiffness case, $0.43 \frac{\text{N}\cdot\text{m}}{\text{rad}}$ in the medium stiffness case, and $1.46 \frac{\text{N}\cdot\text{m}}{\text{rad}}$ in the high stiffness case. The time required to converge was approximately 1.4 h 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 Experiments 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 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. 12, 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 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 $0.55 \frac{\text{N}\cdot\text{m}}{\text{rad}}$. The time required to converge was approximately 2.1 h.

Result: Fig. 12 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. 13 shows the error evolution over iterations. It is worth noting that at the 87th 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. CONCLUSION AND FUTURE WORKS

In this paper, 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, and different number of joints.

One of the goals of soft robotics is to design robots 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 exploiting 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 paper 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 this paper 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 proposed algorithm.

APPENDIX PROOFS OF PROPOSITIONS

In this section, we prove all the propositions stated in Section III.

Proposition 1: If

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

then (5) holds.

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

$$\begin{aligned} & \left\| \frac{dT(q - r, d)}{dq} \Big|_{q=r} - \frac{dT(q - \psi, d)}{dq} \Big|_{q=q_*} \right\| = \\ & \left\| \frac{\partial T(q - r, d)}{\partial q} \Big|_{q=r} - \frac{\partial T(q - \psi, d)}{\partial q} \Big|_{q=q_*} \left(1 - \frac{\partial \psi}{\partial q} \Big|_{q=q_*} \right) \right\|. \end{aligned} \quad (32)$$

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

$$\frac{\partial T(q - r, d)}{\partial q} \Big|_{q=r} = \frac{\partial T(q - \psi, d)}{\partial q} \Big|_{q=q_*} \quad (33)$$

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

$$\begin{aligned} & \left\| \frac{\partial T(q - r, d)}{\partial q} \Big|_{q=r} - \frac{\partial T(q - \psi, d)}{\partial q} \Big|_{q=q_*} \left(1 - \frac{\partial \psi}{\partial q} \Big|_{q=q_*} \right) \right\| \\ &= \left\| \frac{\partial T(q - \psi, d)}{\partial q} \Big|_{q=q_*} \frac{\partial \psi}{\partial q} \Big|_{q=q_*} \right\| \\ &\leq \left\| \frac{\partial T(q - \psi, d)}{\partial q} \Big|_{q=q_*} \right\| \left\| \frac{\partial \psi}{\partial q} \Big|_{q=q_*} \right\| \end{aligned} \quad (34)$$

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

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

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])

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

Combining (34) and (36) yields

$$\begin{aligned} & \left\| \frac{dT(q - r, d)}{dq} \Big|_{q=r} - \frac{dT(q - \psi, d)}{dq} \Big|_{q=q_*} \right\| \\ &\leq \left\| \frac{\partial T(q - \psi, d)}{\partial q} \Big|_{q=q_*} \right\| \max_i \left\| \frac{\partial \psi_i}{\partial q_i} \Big|_{q=q_*} \right\| \end{aligned} \quad (37)$$

which implies the thesis. ■

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

$$\forall \gamma \geq 0 \quad \exists R > 0 \text{ s.t. } \|K_{\text{on},i}\| \leq \gamma. \quad (38)$$

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 $S(t_f) = \emptyset$, then $S(t)$ is bounded in norm $\forall t \in [0, t_f]$, $\forall R > 0$. This derives from many classic results in optimal control theory (see, e.g., [21] and [22]). Thus, it is always possible to bound $\|S\|$ with a sufficiently large constant

$\Sigma > 0$. Hence, from (18)

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

where, from (15), $B_{\text{max}} = \max_{t \in [0, t_f]} \frac{\sigma(t)}{I}$ 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 , which implies the thesis by choosing $R = \frac{B_{\text{max}} \Sigma}{\gamma}$. ■

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

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

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 $\forall \epsilon \in [0, 1)$ and $\forall \Gamma_i^T(t) \in \ker\{B_i^T(t)\}$:

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

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

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

Substituting (42) in (21) yields to $|\epsilon| < 1$, 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 Biotechnol.*, vol. 31, no. 5, pp. 287–294, 2013.
- [2] A. Albu-Schäffer *et al.*, “Soft robotics,” *IEEE Robot. Autom. Mag.*, vol. 15, no. 3, pp. 20–30, Sep. 2008.
- [3] C. Della Santina *et al.*, “The quest for natural machine motion: An open platform to fast-prototyping articulated soft robots,” *IEEE Robot. Autom. Mag.*, vol. 24, no. 1, pp. 48–56, Mar. 2017.
- [4] G. Buondonno and A. De Luca, “Efficient computation of inverse dynamics and feedback linearization for VSA-based robots,” *IEEE Robotics Autom. Lett.*, vol. 1, no. 2, pp. 908–915, 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 5487–5494.
- [6] F. Petit, A. Daasch, and A. Albu-Schaffer, “Backstepping control of variable stiffness robots,” *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 6, pp. 2195–2202, Nov. 2015.
- [7] C. Della Santina *et al.*, “Controlling soft robots: Balancing feedback and feedforward elements,” *IEEE Robot. Autom. Mag.*, vol. 24, no. 3, pp. 75–83, Sep. 2017.
- [8] D. Bristow, M. Tharayil, and A. G. Alleyne, “A survey of iterative learning control,” *IEEE Control Syst.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.
- [9] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operation of robots by learning,” *J. Robot. Syst.*, 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*. New York, NY, USA: Springer, 2016, ch. 21.
- [12] G. Adams and M. Nosonovsky, “Contact modeling—Forces,” *Tribol. Int.*, vol. 33, no. 5, pp. 431–442, 2000.
- [13] G. Grioli *et al.*, “Variable stiffness actuators: The user’s point of view,” *Int. J. Robot. Res.*, vol. 34, no. 6, pp. 727–743, 2015.
- [14] G. Pratt and M. M. Williamson, “Series elastic actuators,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Human Robot Interaction Cooperative Robots*, 1995, vol. 1, pp. 399–406.
- [15] B. Vanderborght *et al.*, “Variable impedance actuators: A review,” *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1601–1614, 2013.
- [16] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. New York, NY, USA: Springer, 2009, ch. 8.
- [17] P. Ouyang, B. Petz, and F. Xi, “Iterative learning control with switching gain feedback for nonlinear systems,” *J. Comput. Nonlinear Dyn.*, vol. 6, no. 1, 2011, Art. no. 011020.
- [18] P.-i. Pipatpaibul and P. Ouyang, “Application of online iterative learning tracking control for quadrotor UAVs,” *ISRN Robot.*, vol. 2013, 2013, Art. no. 476153.
- [19] A. E. Bryson, *Applied Optimal Control: Optimization, Estimation and Control*. Boca Raton, FL, USA: CRC Press, 1975.
- [20] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, vol. 7. Kongens Lyngby, Denmark: Tech. Univ. Denmark, 2008, p. 15.
- [21] R. E. Kalman *et al.*, “Contributions to the theory of optimal control,” *Boletín de la Sociedad Matemática 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,” *J. Differential Equations*, vol. 8, no. 2, pp. 258–263, 1970.
- [23] B. D. Anderson and J. B. Moore, *Linear Optimal Control*, vol. 197. Englewood Cliffs, NJ, USA: Prentice-Hall, 1971.



Franco Angelini (S’18) received the B.S. degree in computer engineering in 2013 and M.S. degree (*cum laude*) in automation and robotics engineering in 2016 from the University of Pisa, Pisa, Italy, where he is currently working toward the Ph.D. degree in robotics at the Research Center “Enrico Piaggio.”

He is a Fellow of the Italian Institute of Technology. His current research focuses on control of soft robotic systems.



Cosimo Della Santina (S’16) received the B.S. degree (*cum laude*) in computer engineering in 2011 and M.S. degree (*cum laude*) in automation and robotics engineering in 2014 from the University of Pisa, Pisa, Italy, where he is currently working toward the Ph.D. degree in robotics at the Research Center “Enrico Piaggio.”

His research interests include model-based design, sensing, and control of soft robots and soft hands.



Manolo Garabini received the B.S. and M.S. degrees in mechanical engineering and Ph.D. degree in robotics from the Università di Pisa, Pisa, Italy, in 2007, 2010, and 2014, respectively.

His research interests include control systems and variable impedance actuation.



Matteo Bianchi (M'12) received the B.S. and M.S. degrees (*cum laude*) in biomedical engineering and Ph.D. degree in haptics, robotics, bio-engineering, control engineering from University of Pisa, Pisa, Italy, in 2004, 2007, and 2012, respectively.

He is currently an Assistant Professor with the Centro di Ricerca "Enrico Piaggio," University of Pisa, Pisa, Italy, and a Research Affiliate with the Mayo Clinic, Rochester, MN, USA. He is the local Principal Investigator with the University of Pisa for Project SOFTPRO. He is an author of contributions

to international conferences and journals. He is a reviewer and member of the editorial board and organizing committee of international journals and conferences. He is a Coeditor of the book *Human and Robot Hands* (Springer, 2016). His research interests include haptic interface design and control; medical and assistive robotics; advanced human-robot interaction; human and robotic hands: sensing and control; and human-inspired control for soft robots.

Prof. Bianchi was a recipient of several international awards including the Best Paper Award at the IEEE Robotics and Automation Society (RAS) Haptics Symposium 2016. He is a Co-Chair of the IEEE RAS Technical Committee on Robotic Hands, Grasping and Manipulation.



Gian Maria Gasparri received the B.S. degree in computer engineering, M.S. degree (*cum laude*) in automation and robotics engineering, and Ph.D. degree in robotics from the University of Pisa, Pisa, Italy, in 2008, 2012, and 2016, respectively.

His research interests include soft robots and robotic locomotion.



Giorgio Grioli (M'14) received the Ph.D. degree in robotics, automation and engineering from the University of Pisa, Pisa, Italy, in 2011.

He is a Researcher with the Italian Institute of Technology, Genoa, Italy, where he investigates design, modeling, and control of soft robotic systems applied to augmentation of, rehabilitation of, and interaction with the human. He is the author of more than 60 articles (both journal papers and conference proceedings) in the fields of soft robotic actuation, robot hand design, and haptics.

Dr. Grioli is an Associate Editor for International Conference on Robotics and Automation and International Conference on Rehabilitation Robotics and is currently co-editing a special issue of the *Actuators* journal on "Variable Stiffness and Variable Impedance Actuators."



Manuel Giuseppe Catalano (M'13) received the B.S. and M.S. degrees in mechanical engineering and Ph.D. degree in robotics from the University of Pisa, Pisa, Italy, in 2006, 2008, and 2013, respectively.

He is currently a Researcher with the Italian Institute of Technology, Genoa, Italy, and a collaborator of Centro di Ricerca "Enrico Piaggio," University of Pisa. His research interests include the design of soft robotic systems, human-robot interaction, and prosthetics.

Dr. Catalano was the recipient of the Georges Giralt Ph.D. Award in 2014, the prestigious annual European award given for the best Ph.D. thesis by euRobotics AISBL.



Antonio Bicchi (F'05) received the Laurea degree (*cum laude*) in mechanical engineering from University of Pisa, Italy, in 1984, and Ph.D. degree in mechanical engineering from University of Bologna, Bologna, Italy, in 1989.

He was a Visiting Scholar with Artificial Intelligence Laboratory, Massachusetts Institute of Technology in 1989. He is a Professor of automatic control with the Department of Information Engineering and Centro Enrico Piaggio, University of Pisa, Pisa, Italy, where he has been leading the robotics research

group since 1990. Since 2009, he has been a Senior Researcher with the Italian Institute of Technology, Genoa, Italy, where he leads the soft robotics for human cooperation and rehabilitation research line. His research interests include robotics, haptics, and automatic control. He has authored or coauthored more than 400 papers on international journals, books, and refereed conferences. He was granted an ERC Advanced Grant in 2011.