A Provably Stable Iterative Learning Controller for Continuum Soft Robots

Michele Pierallini, Francesco Stella, Franco Angelini, Bastian Deutschmann, Josie Hughes, Antonio Bicchi, Manolo Garabini, and Cosimo Della Santina

***Abstract*—Fully exploiting soft robots' capabilities requires devising strategies that can accurately control their movements with the limited amount of control sources available. This task is challenging for reasons including the hard-to-model dynamics, the system's underactuation, and the need of using a prominent feedforward control action to preserve the soft and safe robot behavior. To tackle this challenge, this letter proposes a purely feedforward iterative learning control algorithm that refines the torque action by leveraging both the knowledge of the model and data obtained from past experience. After presenting a 3D polynomial description of soft robots, we study their intrinsic properties, e.g., input-to-state stability, and we prove the convergence of the controller coping with locally Lipschitz nonlinearities. Finally, we validate the proposed approach through simulations and experiments involving multiple systems, trajectories, and in the case of external disturbances and model mismatches.**

***Index Terms*—Modeling, Control, and Learning for Soft Robots; Underactuated Robots; Motion Control**

I. Introduction

DESPITE considerable work being put into the design and modeling aspect of soft continuum structures [1], [2], the control problem is still open [3]. Existing approaches attack the challenge by means of mostly feedforward learning-based methods [4]–[8] or with feedback model-based approaches [9]–[12]. The former preserves the robot's elasticity without requiring a precise description of the model. Still, it is time-expensive and does not allow drawing any theoretical conclusion on the system's physical properties or behavior.

Manuscript received: 4 29, 2023; Revised: 6 16, 2023; Accepted: 8 7, 2023.
This paper was recommended for publication by Editor J. Kober.
This research has received funding partially from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101016970 (Natural Intelligence), No. 101017274 (DARKO), and No. 101070918 (EMERGE), and in part by Ministry of University and Research (MUR) as part of the PON 2014-2021 "Research and Innovation" resources – Green/Innovation Action - DM MUR 1062/2021, in part by the Italian Ministry of Education and Research in the framework of the "FoReLab" (Future-oriented Research Lab) Project (Departments of Excellence).

Michele Pierallini, Franco Angelini, Antonio Bicchi, and Manolo Garabini are with the Centro di Ricerca "Enrico Piaggio" and the Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Largo Lucio Lazzarino 1, 56126 Pisa, Italy. {michele.pierallini, frncangelini, manolo.garabini}@gmail.com.
Francesco Stella and Josie Hughes are with the CREATE Lab, EPFL, Lausanne, Switzerland. {francesco.stella, josie.hughes}@epfl.ch.
Antonio Bicchi is with the Soft Robotics for Human Cooperation and Rehabilitation, Fondazione Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy (e-mail: antonio.bicchi@unipi.it).
Cosimo Della Santina is with the Department of Cognitive Robotics, Delft University of Technology, 2628 CN Delft, The Netherlands. cosimodellasantina@gmail.com.
Bastian Deutschmann and Cosimo Della Santina are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Oberpfaffenhofen, Germany. bastian.deutschmann@dlr.de.

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

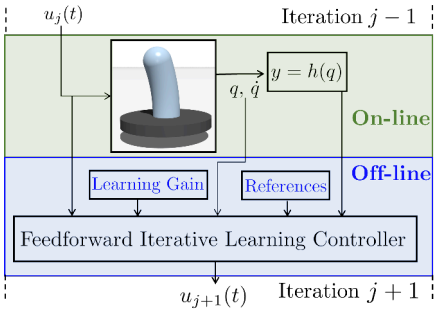


Fig. 1. Proposed iterative learning control scheme for soft continuum robots.

Conversely, the latter leads to high performance while compensating for disturbances at the cost of stiffening up the robot's behavior [13] and relying on a precise system description. Thus, there is no controller available that can preserve the robot's elasticity while achieving good performance [3]. To solve this challenge, we propose a controller that relies upon the intersection of learning-based and model-based methods.

The main contribution of the paper is a pure feedforward torque-based control strategy for soft robots. The algorithm belongs to the class of iterative learning controller (ILC) [14] Fig. 1. The algorithm can cope with model uncertainties and external disturbances while preserving the robot's intrinsic elasticity [13]. It relies on both the Lagrangian robot model [15] and data obtained from previous iterations. ILC algorithms have been applied to soft continuum robots in [16] to reach the optimal planned trajectories and in [17] where a pure feedback controller is implemented without experimental validation. Therefore, a pure feedforward stand-alone ILC framework for soft continuum robots is not at our disposal.

Exploiting the model knowledge speeds up the learning process while coping with the underactuation of the system, disturbances, and model uncertainties. Leveraging the Lyapunov stability of the soft structure, the control action tackles locally Lipschitz nonlinearities in the dynamics. We rigorously prove the controller convergence.

A secondary contribution of this letter is a 3D polynomial curvature description of the robot including the compression variables generalizing [15] and [18].

We validate the learning algorithm by performing simulations and experiments with varying systems, trajectory features, e.g., velocity, and disturbances.

To summarize, this letter's contributions are:

- A polynomial model of continuum robots generalizing [15] into a 3D model and [19] considering a general number of curvature models Sec. II.
- Input-to-state stability proof for soft robots Sec. III.
- A pure torqued-based feedforward iterative learning controller that uses both model-based and data-driven terms,

whose convergence is theoretically proved even in the case of disturbances Sec. IV.
- The validation of the proposed approach in both simulations and real hardware Sec. V.

II. Continuum Arm Model

We introduce a dynamic model for a continuum soft arm.

A. Forward and Differential Kinematics

For the $i-$th soft segment $i=0,\cdots,\infty$ Fig. 2, we generalized the approach in [15] and [18] deriving a 3D model adding also the compression along the central axis, namely $\delta L_i \in \mathscr{X}_q \subset \mathbb{R}$. We model the $i-$th curvature as a polynomial function, i.e.,

$$c_i(s,t) = \sum_{k=0}^{+\infty} \theta_{k_i}(t) s^k, \quad \forall s \in [0,1], \forall t \in [0,t_\mathrm{f}], \tag{1}$$

where θ_{k_i} are the modal weights, t is the time variable and t_f is the terminal time. From (1), the local orientation is computed as $\alpha_i(s,t) = \int_0^s c_i(s',t)\mathrm{d}s'$. Recalling [15], let us assume that $\theta_{0_i}(\cdot) : [0,t_\mathrm{f}] \to \mathscr{X}_q$, is such that $\theta_{0_i}(t) \gg \sum_{k=1}^{+\infty} \theta_{k_i}(t), \forall i, \forall t \in [0,t_\mathrm{f}]$. Furthermore, we introduce the truncation operator $[(\cdot)]_o$ at the $o-$order with $o \in \mathbb{N}$, which, applied to (1), leads to $[c_i(s,t)]_o = \sum_{k=0}^{o} \theta_{k_i}(t) s^k, \forall i, \forall s \in [0,1], \forall t \in [0,t_\mathrm{f}]$.

Let q_i be the configuration of the $i-$th segment, i.e., $q_i = [\phi_i, \theta_{0_i}, \cdots, \theta_{o_i}, \delta L_i]^\top \triangleq \left[\phi_i, \Theta_i^\top, \delta L_i\right]^\top \in \mathscr{X}_q^{o+3} \forall i$. Based on geometric consideration and recalling Fig. 2, we derive the forward kinematics $T_i^s(q_i,s) \in \mathscr{X}_q^{4\times 4}$, which describes each frame pose along the main axis, i.e.,

$$T_i^s(q_i,s) = \begin{bmatrix} R_z(\phi_i)R_y(\alpha_i(s))R_z(-\phi_i) & R_z(\phi_i)R_y(\alpha_i(s))p_{1_i}(q_i) \\ 0_{1\times 3} & 1 \end{bmatrix}, \tag{2}$$

where $p_{1_i}(q_i) = [0, 0, L_i + \delta L_i]^\top \in \mathscr{X}_q^3$ and $R_{(\cdot)}(s) \in SO(3)$. It is worth noting that $\lim_{\Theta_i \to 0_o} T_i^s(q_i) = \begin{bmatrix} R_z(\phi_i) & p_{1_i}(q_i) \\ 0_{1\times 3} & 1 \end{bmatrix}$ which is well-defined even in the origin.

We highlight that, in the case of $o=0$, (2) yields into the Piecewise Constant Curvature (PCC) model [20]. The polynomial description represents a finer representation of the curvature with a higher computational cost and a greater number of state variables that can lead to system underactuation.

Following the standard approach (see e.g., [20]), one can compute the differential kinematics, i.e., $\dot{\xi}_i = J_i(q)\dot{q}_i, \forall i$ where $\dot{\xi}_i \in \mathscr{X}_q^6$ includes the linear and angular velocities of the robot's tip, $\dot{q}_i \in \mathscr{X}_q^{o+3}$ includes the velocities, and the Jacobian $J_i(q) \in \mathscr{X}_q^{6\times n}$ is $J_i(q) = \left[J_{p_i}(q)^\top, J_{o_i}(q)^\top\right]^\top = \partial_q f_{K_i}(q_i,s)^\top$[1], where $f_{K_i}(\cdot,\cdot) : \mathscr{X}_q^{o+3} \times [0,1] \to \mathscr{X}_q^6$ is the forward kinematics map, i.e., (2), and $J_{p_i}(q), J_{o_i}(q) \in \mathscr{X}_q^{3\times n}$ are the position and orientation Jacobian, respectively. For the sake of space, we will remove the time dependence.

B. Dynamics and Output Function

Using the classical Lagrangian approach, one can derive the continuum robot dynamics (refer to [15] for more details), i.e.,

$$M(q)\ddot{q} + n(q,\dot{q}) = A(q)u + J^\top(q)\tau_\mathrm{ext} + \xi\ , \tag{3}$$

where $\ddot{q} \in \mathscr{X}_q^n$ includes the joints acceleration. $M(q) \in \mathscr{X}_q^{n\times n}$ is inertia matrix. $n(q,\dot{q}) \triangleq C(q,\dot{q})\dot{q} + G(q) + D\dot{q} + Kq$ where $C(q,\dot{q}) \in \mathscr{X}_q^{n\times n}$ is the Coriolis matrix, which is computed using the Christoffer coefficients, thus $q^\top\left(\dot{M}(q) - 2C(q,\dot{q})\right)q = 0, \forall q \in \mathscr{X}_q^n$. $K, D \in \mathscr{X}_q^{n\times n}$ such as $K, D \succ 0$ are the stiffness and damping matrices, i.e., $[K]_{11} = \kappa_\phi$, $[D]_{11} = d_\phi$, $[K]_{ij} = \frac{\kappa}{i+j+1}$, $[D]_{ij} = \frac{d}{i+j+1}, \forall i,j = 2,\cdots,n-1$, $[K]_{nn} = \kappa_{\delta L}$, and $[D]_{nn} = d_{\delta L}$, respectively [15]. $\tau_\mathrm{ext} \in \mathscr{X}_q^6$ collects the external forces (if present), i.e., $\sup_{t\in[0,t_\mathrm{f}]}\{\|\tau_\mathrm{ext}\|\} \leq b_{\tau_\mathrm{ext}} < \infty$,and $\xi(\cdot) : [0,t_\mathrm{f}] \to \mathscr{X}_q^n$ such as $\sup_t\{||\xi(t)||\} \leq b_\xi < \infty$ embeds the model artifacts, and it will described in Sec. VI, [14], [21]. $u(\cdot) : [0,t_\mathrm{f}] \to \mathscr{X}_q^m$ is the control input (Sec. VI), and $A(q) \in \mathscr{X}_q^{n\times m}$ is the actuation matrix, which depends on both the geometry and the actuators; hence it will be described in Sec. V. Also, $A(q)$ is such that $\max\{\mathrm{rank}\{A(q)\}\} \leq p$ with $p \leq m \leq n$. Without loss of generality, let us assume that $\mathrm{rank}\{A(q)\} = p, \forall q \in \mathscr{X}_q^n$.

Finally, depending on the curvature truncation order $o \in \mathbb{N}^+$ and the number of actuators, the system (3) may be underactuated [22]. Thus, we can assign only a subset or an (even nonlinear) combination of the state. To this end, we define the output of the system, i.e., $y(\cdot) \in \mathscr{X}_q^p$, and a smooth output function $h(\cdot) : [0,t_\mathrm{f}] \times \mathscr{X}_q^n \to \mathscr{X}_q^p$, i.e.,

$$y = h(q(t))\ , \tag{4}$$

such as $\left|\left|\partial_q h(q)\right|\right| \leq h_0$. Note that, depending on the value of p, the system (3)-(4) may also be non-square.

C. Dynamic Model Properties

Recalling (3), the following property holds for any soft continuum robot [11].

Property 1. *For the system* (3) $\forall q, \dot{q} \in \mathscr{X}_q^n$*, the following positive constants exist (see, e.g., [3], [15]):*

- $b_m \leq ||M(q)|| \leq b_M$*, and* $M(q) = M^\top(q) \succeq 0, \forall q \in \mathscr{X}_q^n$,
- $||C(q,\dot{q})|| \leq b_C(\|\dot{q}\| + \|q\|)$ *and also that* $\left|\left|\partial_q C(q,\dot{q})\right|\right| = \left|\left|\partial_{\dot{q}} C(q,\dot{q})\right|\right| \leq b_{\partial C}$,
- $||G(q)|| \leq b_G\|q\| + b_{g_0}, b_{g_0} \in (0,\infty]$ *and* $\left|\left|\partial_q G(q)\right|\right| \leq b_{\partial G}$,
- $\|J(q)\| \leq b_J \implies 0 < \|A(q)\| \leq b_A$ *for almost all* $q \in \mathscr{X}_q^n$.

Recalling Property 1, we remark two features of soft robots.

Remark 1. *The dynamics of any soft continuum robot* (3) *is locally Lipschitz* $\forall q, \dot{q} \in \mathscr{X}_q^n$.

To solve the control problem, let us assume what follow.

Assumption 1. *The system* (3)-(4) *has relative degree* $r = 2$ *for almost all* $q, \dot{q} \in \mathscr{X}_q^n$*, (see, e.g., [14]).*

Assumption 1 guarantees an inertial coupling between all the variables, which is very common for any Lagrangian system [14], [22]. In practice, it means that the controller inertially acts on the output function (4) and its time derivatives.

Assumption 2. *Let* $y_\mathrm{d}(\cdot) : [0,t_\mathrm{f}] \to \mathscr{X}_q^p$ *be the desired output function. We assume the unique existence of the desired state and bounded control input, i.e.,* $\forall y_\mathrm{d}, \exists! q_\mathrm{d}, \dot{q}_\mathrm{d} \in \mathscr{X}_q^n, u_\mathrm{d}(\cdot) : [0,t_\mathrm{f}] \to \mathscr{X}_q^m$ *with* $\sup_{t\in[0,t_\mathrm{f}]}\|u_\mathrm{d}(t)\| \leq b_{u_\mathrm{d}}$ *such as* $M(q_\mathrm{d})\ddot{q}_\mathrm{d} + n(q_\mathrm{d},\dot{q}_\mathrm{d}) = h(q_\mathrm{d})u_\mathrm{d}$ *and* $y_\mathrm{d} = h(q_\mathrm{d})$.

[1]The symbol $\partial_q(\cdot)$ stands for $\partial_q(\cdot) \triangleq \partial(\cdot)/\partial q$ throughout the paper.

Assumption 2 ensures the trajectory feasibility to provide the existence of the control problem solution [14]. In practice, one can check if the trajectory belongs to the robot's workspace. Recalling Property 1, Remark 1, and Assumption 2, one can state what follows.

Lemma 1. *Let us consider the system* (3)-(4) *with Property 1 and Remark 1. Assumption 2 implies that the distance between the real and desired dynamics is locally Lipschitz, i.e.,*

$$\begin{aligned} \|M(q)-M(q_{\mathrm{d}})\| &\leq b_M \\ \|n(q,\dot{q})-n(q_{\mathrm{d}},\dot{q}_{\mathrm{d}})\| &\leq b_n(\|q\|,\|\dot{q}\|)\|q-q_{\mathrm{d}}\| \end{aligned} \tag{5}$$

$\forall q, q_{\mathrm{d}}, \dot{q}, \dot{q}_{\mathrm{d}} \in \mathscr{X}_q^n$, *with* $b_n(\cdot) \in \mathscr{K}_\infty$ *unknown function.*

Proof. The proof follows by Assumption 2 and (3). □

III. Stability Analysis

We here study the stability of the soft robot in (3). First, we derive a classical result leveraging Lyapunov indirect theorem in Proposition 1. Second, we prove the input-to-state stability for (3). This analysis allows us to derive strong claims while exploring the peculiarities of soft robots. For instance, the majority of ILCs rely on the global Lipschitz assumption [14], and we, leveraging Lemma 2, propose results on the local one.

Proposition 1. *Let us consider one segment of soft continuum robot described in* (3) *with Property 1. Let* $[\overline{q}^\top, 0_{n\times 1}]^\top$, $u(t) \equiv 0$ *be a system equilibrium. If the following inequality holds*

$$\sum_{h=0}^{o} \frac{\kappa}{h+3} > mg \sum_{h=0}^{o} \partial_{q_h}\partial_{\theta_0}\, p_{O_{\mathrm{tip}_z}}(q)\Big|_{q=\overline{q}} \tag{6}$$

with $p_{O_{\mathrm{tip}_z}}$ *is the* $\hat{z}$ *component of the robot tip and* $o \in \mathbb{N}^+$ *is the truncation order; the equilibrium is asymptotically stable.*

Proof. Inequality (6) implies $\partial_q K(q) + \partial_q G(q)\big|_{q=\overline{q}} \succ 0$, which, leveraging the Lyapunov indirect Theorem, guarantees the asymptotic stability of the equilibrium [3]. □

We study the input-to-state stability of the soft robots.

Proposition 2. *Any soft continuum robot in* (3) *with Property 1 is Input-to-State Stable* $\forall q, \dot{q} \in \mathscr{X}_q^n, \forall t \in [0, t_{\mathrm{f}}]$.

Proof. Recalling (3) assuming $\xi(t) \equiv 0$, let $V(\cdot) : \mathscr{X}_q^n \times \mathscr{X}_q^n \to \mathbb{R}^+$ be a Lyapunov function, i.e., $V(q,\dot{q}) = \frac{b_1}{2} q^\top K q + \frac{b_2}{2}\dot{q}^\top M(q)\dot{q}$ where $b_1, b_2 \in \mathscr{X}_q^+ \backslash \{0, +\infty\}$ with $b_2 > b_1$. Computing the time derivative of $V(q,\dot{q})$ and noting that $q^\top\left(\dot{M}(q,\dot{q}) - 2C(q,\dot{q})\right)q = 0$, $\forall q, \dot{q} \in \mathscr{X}_q^n$, lead to $\dot{V}(q,\dot{q}) = (b_2-b_1)q^\top K\dot{q} - \frac{b_2}{2}\dot{q}^\top D\dot{q} - \frac{b_2}{2}\dot{q}^\top\left(G(q) - A(q)u - J^\top(q)\tau_{\mathrm{ext}}\right)$. Since $K \succ 0$, one can write the following inequality

$$\begin{aligned} \dot{V}(q,\dot{q}) \leq & \frac{b_2-b_1}{2}\left(q^\top K q + \dot{q}^\top K\dot{q}\right) - \frac{b_2}{2}\dot{q}^\top D\dot{q} \\ & - \frac{b_2}{2}\dot{q}^\top\left(G(q) - A(q)u - J^\top(q)\tau_{\mathrm{ext}}\right). \end{aligned} \tag{7}$$

From (7), computing the square of the gravity and control term and defining $Q \triangleq (D + (b_2-b_1)K) \succ 0$ lead to

$$\begin{aligned} \dot{V}(q,\dot{q}) \leq & -(b_2-b_1)q^\top K q - \frac{b_2}{2}\dot{q}^\top Q\dot{q} \\ & - \left(\dot{q} + b_2 Q^{-1}(G(q) + A(q)u)\right)^\top Q^{-1}\left(\dot{q} + b_2 Q^{-1}(G(q)A(q)u)\right) \\ & - \left(\dot{q} - b_2 Q^{-1}J^\top(q)\tau_{\mathrm{ext}}\right)^\top Q^{-1}\left(\dot{q} - b_2 Q^{-1}J^\top(q)\tau_{\mathrm{ext}}\right) \\ & + b_2^2 u^\top A^\top(q) Q A(q) u + b_2^2 G^\top(q) Q^{-1} G(q) \\ \leq & -(b_2-b_1)q^\top K q - \frac{b_2}{2}\dot{q}^\top Q\dot{q} + b_2^2 u^\top A^\top(q) Q A(q) u \\ & + b_2^2 G^\top(q) Q G(q) + b_2^2 \tau_{\mathrm{ext}}^\top J(q) Q J^\top(q)\tau_{\mathrm{ext}}. \end{aligned} \tag{8}$$

Recalling Property 1, one has that $b_2^2 u^\top A^\top(q) Q A(q) u \leq b_2^2 b_A^2 \|Q\|\|u\| \triangleq \eta(\|u\|)$, with $\eta(\cdot) \in \mathscr{K}$, $b_2^2 G^\top(q) Q^{-1} G(q) \leq b_2^2 b_G^2 q^\top \|Q\| q$, and $b_2^2 \tau_{\mathrm{ext}}^\top(q) J(q) Q^{-1} J^\top(q)\tau_{\mathrm{ext}} \leq b_2^2 b_{\tau_{\mathrm{ext}}} q^\top \|Q\| q$, Thus, (8) becomes

$$\begin{aligned} \dot{V}(q,\dot{q}) \leq & -(b_2-b_1)q^\top K q - \frac{b_2}{2}\dot{q}^\top Q\dot{q} + \eta(||u||) \\ & - \frac{b_2}{2} q^\top\left((1 - b_2(b_G^2 + b_{\tau_{\mathrm{ext}}})(b_2-b_1)K - b_2(b_G^2 + b_{\tau_{\mathrm{ext}}})D\right)q. \end{aligned} \tag{9}$$

Note that $\forall b_G, b_{\tau_{\mathrm{ext}}}, \exists b_2 > b_1$ such that $(1 - b_2(b_G^2 + b_{\tau_{\mathrm{ext}}})(b_2-b_1)K - b_2(b_G^2 + b_{\tau_{\mathrm{ext}}})D \succ 0$. Thus (9) can be rewritten as $\dot{V}(q,\dot{q}) \leq -\chi(||q||, ||\dot{q}||) + \eta(||u||)$ with $\chi(\cdot) \in \mathscr{K}\mathscr{L}$. This achieves the proof. □

Proposition 2 analyzes the stability in a local and nonlinear fashion considering a control action. Conversely, Proposition 1 claims linear results, which hold only in an equilibrium point neighborhood, and it is a special case of Proposition 2. Both Propositions 1 and 2 claim that if the robot stiffness overcomes the gravity term, then the system is Lyapunov stable. Finally, the scope of Proposition 2 is twofold. First, it describes an intrinsic property derived from the soft continuum body. Second, it is instrumental in the derivation of the controller.

IV. Iterative Learning Control Design

In this section, relying on Sec. II-III, we introduce the iterative learning controller proving its convergence even in the case of locally Lipschitz nonlinearities, external disturbances, and model artifacts. Then, we propose a learning gain choice, which ensures convergence.

Managing locally Lipschitz nonlinearities is still an open problem in the ILC framework. Hence, most of the algorithms assume the system to be globally Lipschitz [13], [14]. However, invoking Property 1, this assumption is too strict. To solve this challenge, in [23], a saturated action is employed. Conversely, we draw on the input-to-state property of the continuous robots to deal with these nonlinearities.

Lemma 2. *Let us consider* (3) *with Assumptions 1-2, and let us recall Proposition 2. Let* $f(\cdot) : \mathscr{X}_q^{2n} \to \mathscr{X}_q^{2n}$ *be one of the entities in* (3)-(4)*, then the next inequality holds true* $\forall f(\cdot)$*:* $\|f(x_1) - f(x_2)\| \leq f_0(\|x\|)\|x_1 - x_2\| \leq b_f\|x_1 - x_2\|$*, with* $f_0(\|x\|) \in \mathscr{K}_\infty$ *and* $b_f \in \mathscr{X}^+$ *finite.*

Proof. Recalling Proposition 2 and (9), one has that

$$\|x(t)\| \leq \beta(\|x_0\|, t) + \gamma(\sup_t\{\|u(t)\|\}), \tag{10}$$

$\forall t \in [0, t_{\mathrm{f}}]$ with $\beta(\cdot) \in \mathscr{K}\mathscr{L}$ and $\gamma(\cdot) \in \mathscr{K}$. Furthermore, Assumption 2 guarantees the existence of a bounded control

input. Hence, (10) implies a bounded state and thus, that $\exists b_f \in \mathscr{X}_q^+$ such that $\sup_{t\in[0,t_\mathrm{f}]} f_0(\|x(t)\|) \le b_f < +\infty$. □

Lemma 2 holds for all soft robots such as (3) [2], [3] and it implies $\|n(q,\dot{q})\| \le b_n(\|q\| + \|\dot{q}\|)$, with $b_n \in \mathscr{X}_q^+$ finite.

Leveraging Lemma 2, the following Lemma provides Gronwall-like inequalities for the soft robot dynamics in (3).

Lemma 3. *Let us consider the soft robot dynamics* (3)*, with Property 1 under Assumption 1. Let the thesis of Lemma 2 be true. Let* $q_0, \dot{q}_0 \in \mathscr{X}_q^n$ *be the initial condition such as* $||q_\mathrm{d}(0) - q_0|| \le b_0$ *finite, and* $\dot{q}_0 = \dot{q}_\mathrm{d}(0)$*. Then, there exist some positive constants* $b_{\delta q} < 1, b_{\delta\dot{q}} < 1$*, and* $\lambda > 0$ *such that*

$$||\delta q(t)||_\lambda \le b_{\delta q}||\delta u(t)||_\lambda + b_0 \tag{11}$$
$$||\delta\dot{q}(t)||_\lambda \le b_{\delta\dot{q}}||\delta u(t)||_\lambda + b_\xi \tag{12}$$

where $\delta u_j \triangleq u_\mathrm{d} - u_j$*,* $\delta q_j \triangleq q_\mathrm{d} - q_j$*,* $\delta\dot{q}_j \triangleq \dot{q}_\mathrm{d} - \dot{q}_j$*, and* $\|(\cdot)\|_\lambda$ *is the* $\lambda-$*norm of* $(\cdot)$*, i.e.,* $\|(\cdot)\|_\lambda \triangleq \sup_t\{\|(\cdot)\|\mathrm{e}^{-\lambda t}\}$*,* $\lambda > 0$*.*

Proof. Evaluating (3) in the desired and generic state leads to

$$\begin{aligned} &M(q_\mathrm{d})\ddot{q}_\mathrm{d} - M(q)\ddot{q} + n(q_\mathrm{d},\dot{q}_\mathrm{d}) - n(q,\dot{q}) = \\ &A(q_\mathrm{d})u_\mathrm{d} - A(q)u + \xi(t) + J^\top(q)\tau_\mathrm{ext}\,. \end{aligned} \tag{13}$$

Recalling Property 1 and Lemma 2, one can write

$$||\delta\ddot{q}|| \le \frac{b_n + b_A b_{u_\mathrm{d}} + b_{\tau_\mathrm{ext}}}{b_M}||\delta q|| + \frac{b_n}{b_M}||\delta\dot{q}|| + \frac{b_A}{b_M}||\delta u|| \tag{14}$$

where $\delta\ddot{q} \triangleq \ddot{q}_\mathrm{d} - \ddot{q}$. Defining the state $z \triangleq [||\delta q||, ||\delta\dot{q}||]^\top \in \mathscr{X}_q^2$, (14) can be rewritten as $\dot{z} = Fz + B\|\delta u\|$, i.e.,

$$\dot{z} \le \begin{bmatrix} 0 & 1 \\ \frac{b_n + b_A b_{u_\mathrm{d}} + b_{\tau_\mathrm{ext}}}{b_M} & \frac{b_n}{b_M} \end{bmatrix} z + \begin{bmatrix} 0 \\ \frac{b_A}{b_M} \end{bmatrix}||\delta u|| + \begin{bmatrix} 0 \\ \frac{b_\xi}{b_M} \end{bmatrix}, \tag{15}$$

which is a linear time invariant system where $F \in \mathscr{X}_q^{2\times 2}$, $B \in \mathscr{X}_q^2$. Applying the Gronwall lemma to (15) leads to

$$z \le \begin{bmatrix} b_0 \\ \frac{b_\xi t_\mathrm{f}}{b_M} \end{bmatrix} + \int_0^{t_\mathrm{f}} \mathrm{e}^{F(t-\tau)}FB\mathrm{d}\tau + B\int_0^{t_\mathrm{f}}||\delta u(\tau)||\,\mathrm{d}\tau\,. \tag{16}$$

Defining $p \triangleq 2\sqrt{{b_n}^2 + 4b_M b_n + 4b_A b_M b_{u_\mathrm{d}} + 4b_M b_{\tau_\mathrm{ext}}}$, one has $\mathrm{e}^{F(t-\tau)} \triangleq \mathrm{e}^F$, whose elements are $[\mathrm{e}^F]_{ij}$ for $i,j = 1,2$, i.e.,

$$[\mathrm{e}^F]_{11} = \left(\frac{p + b_M}{2p}\mathrm{e}^{\frac{b_n+p}{2b_M}} + \frac{p - b_M}{2p}\mathrm{e}^{\frac{b_n-p}{2b_M}}\right)\mathrm{e}^{t-\tau}, \tag{17}$$
$$[\mathrm{e}^F]_{12} = \frac{b_M}{p}\left(\mathrm{e}^{\frac{b_n+p}{2b_M}} - \mathrm{e}^{\frac{b_n-p}{2b_M}}\right)\mathrm{e}^{t-\tau}, \tag{18}$$
$$[\mathrm{e}^F]_{21} = -\frac{(b_n + b_{\tau_\mathrm{ext}} + b_A b_{u_\mathrm{d}})}{p}\left(\mathrm{e}^{\frac{b_n-p}{2b_M}} - \mathrm{e}^{\frac{b_n+p}{2b_M}}\right)\mathrm{e}^{t-\tau}, \tag{19}$$
$$[\mathrm{e}^F]_{22} = \frac{b_n - p}{2p}\left(\mathrm{e}^{\frac{b_n+p}{2b_M}} - \mathrm{e}^{\frac{b_n-p}{2b_M}}\right)\mathrm{e}^{t-\tau}. \tag{20}$$

Leveraging (16), one has

$$z \le \begin{bmatrix} b_0 \\ \frac{b_\xi t_\mathrm{f}}{b_M} \end{bmatrix} + \int_0^{t_\mathrm{f}} \frac{b_A b_n}{b_M}\begin{bmatrix} [\mathrm{e}^F]_{11} + [\mathrm{e}^F]_{12} b_n \\ [\mathrm{e}^F]_{21} + [\mathrm{e}^F]_{22} b_n + \frac{1}{b_n} \end{bmatrix}||\delta u(\tau)||\,\mathrm{d}\tau\,. \tag{21}$$

Computing the $\lambda-$norm and the integral yield to

$$\begin{aligned} ||\delta q||_\lambda \le{}& \frac{b_A b_n}{b_M^2}\left(\frac{b_n(b_n + b_A b_{u_\mathrm{d}} + b_{\tau_\mathrm{ext}})}{p(\lambda-1)}\left(\mathrm{e}^{\frac{b_n-p}{b_M}} - \mathrm{e}^{\frac{b_n}{b_M}}\right)\right. \\ &\left. -\left(p - b_n + b_n\mathrm{e}^{\frac{b_n}{b_M}}b_M + p\mathrm{e}^{\frac{b_n}{b_M}}\right)\Big/\left(2b_M(\lambda-1)\right)\right) \\ &\times \mathrm{e}^{-\frac{b_n-p}{2b_M}}\left(\mathrm{e}^{t_\mathrm{f}(1-\lambda)} - 1\right)||\delta u||_\lambda + b_0 \le b_{\delta q}||\delta u||_\lambda + b_0\,, \end{aligned} \tag{22}$$

$$\begin{aligned} ||\delta\dot{q}||_\lambda \le{}& \frac{b_A b_n\left(2{b_M}^2 + {b_n}^2 p - b_n p^2\right)}{b_M^2 2b_M p(\lambda-1)}\left(\mathrm{e}^{\frac{b_n-p}{2b_M}} - \mathrm{e}^{\frac{b_n+p}{2b_M}}\right) \\ &\times\left(\mathrm{e}^{(1-\lambda)t_\mathrm{f}} - 1\right)||\delta u||_\lambda + b_\xi \le b_{\delta\dot{q}}||\delta u||_\lambda + b_\xi\,, \end{aligned} \tag{23}$$

in which for any $b_A, b_\xi, b_M, b_n, b_{u_\mathrm{d}}, b_{\tau_\mathrm{ext}} > 0$ finite $\exists\lambda > 0$ such that $b_{\delta q} < 1$ and $b_{\delta\dot{q}} < 1$, i.e., (11)-(12) hold true. □

Remark 2. *The initial shift* b_0 *enables the framework to deal also with regulation tasks. However, to prove the stability of the final equilibrium one must rely on the stiffness-gravity relation, e.g.,* (6)*, or use a feedback term in the controller.*

A. Proposed Algorithm

Running into time and iteration domains, the ILC algorithm reshapes the pure feedforward control law using model information and data until it is able to track the desired trajectory.

Let us define the following pure feedforward learning rule, which combines model-based and data-driven terms, i.e.,

$$\begin{aligned} u_{j+1}(t) = {}& \underbrace{\left(I_m - \Upsilon_j(t)\partial_q h(q_j)M^{-1}(q_j)A(q_j)\right)u_j(t)}_{\text{Model-based term}} \\ &+ \underbrace{\partial_q h(q_j)M^{-1}(q_j)n(q_j,\dot{q}_j)}_{\text{Model-based term}} + \underbrace{K_\mathrm{P}\left(y_\mathrm{d} - y_j\right) + K_\mathrm{V}\left(\dot{y}_\mathrm{d} - \dot{y}_j\right) + \ddot{y}_\mathrm{d}}_{\text{Data-driven term}}, \end{aligned} \tag{24}$$

where $j = 0, 1, \cdots$ is the iteration index, $K_\mathrm{P}, K_\mathrm{V} \succ 0$ are the control gains, and $\Upsilon_j(t) \in \mathscr{X}_q^{m\times p}$ is the learning gain, which plays a fundamental role in the convergence analysis. Since (24) uses data from previous iterations is feedforward, and it preserves the robot stiffness [13] leading to safe behavior.

Eq. (24) needs an initial guess, which can be chosen as

$$u_0(t) = A^+(q_\mathrm{d})M(q_\mathrm{d})\ddot{y}_\mathrm{d} + A^+(q_\mathrm{d})n(q_\mathrm{d},\dot{q}_\mathrm{d})\,, \tag{25}$$

or an equilibrium torque, e.g., $u_0(t) \equiv 0, \forall t \in [0,t_\mathrm{f}]$. Note that (24) is feedforward, it preserves the robot's stiffness [13] leading to safe behavior. The method is described in Alg. 1.

Algorithm 1 Proposed algorithm.

Input: max_iter, toll, $y_\mathrm{d}, \dot{y}_\mathrm{d}, \ddot{y}_\mathrm{d}$
Output: $u(\cdot) : [0,t_\mathrm{f}] \to \mathscr{U}^m$

procedure COMPUTE CONTROL ACTION
  $u_0(t) \leftarrow$ `initialGuess(`$y_\mathrm{d}, \dot{y}_\mathrm{d}, \ddot{y}_\mathrm{d}$`)` ▷ Eq. (25) or $0_{m\times 1}$
  $j \leftarrow 0$, $e_0(t) \leftarrow 0_{m\times 1}$
  while $\left\{\left|\left|e_j(t)\right|\right| >$ toll or $j \le$ max_iter$\right\}$ **do**
    while $t \le t_\mathrm{f}$ **do** ▷ On-Line
      $u_{j+1}(t) \leftarrow$ `sendingControl()`
      `data` $\leftarrow$ `recodingIMU()`
    end while
    $q_j(t), y_j(t) \leftarrow$ `getState(data)` ▷ Off-Line
    $u_{j+1}(t) \leftarrow$ `getControl(`$q_j(t), y_j(t)$`)` ▷ Eq. (24)
    ${}^2e_j(t) \leftarrow {}^2e_{j+1}(t)$, $u_j(t) \leftarrow u_{j+1}(t)$, $j \leftarrow j+1$
  end while
  return $u_{j+1}(t)$ ▷ Control action
end procedure

We define the tracking error $e_j(t) \in \mathscr{X}_q^p$ as follows

$$e_j(t) = y_\mathrm{d}(t) - y_j(t)\,. \tag{26}$$

In Theorem 1, we prove the controller (24) convergence even in the case of external disturbances and artifacts, i.e.,

state and time non-repetitive disturbances w.r.t. the iteration domain, respectively.

Theorem 1. *Let us consider the system* (3)-(4) *with Property 1 under Assumptions 1-2. Let* $q_{0_j}, \dot{q}_{0_j} \in \mathscr{X}_q^n$ *be the initial condition such as* $||q_{0_j} - q_\mathrm{d}(0)|| \leq b_{q_0}, \forall j$ *with* $b_{q_0} \in [0,\infty)$ *and* $\dot{q}_{0_j} = \dot{q}_\mathrm{d}(0), \forall j$. *Let* $\xi_j(\cdot) : \mathbb{N} \times [0,t_\mathrm{f}] \to \mathscr{X}_q^n$ *be such as* $\sup_t\{\max_j\{||\xi_j(t)\}\} \leq b_\xi, \forall j$. *Let* (24) *be the control law and* (26) *the error. If the learning gain* $\Upsilon_j(t) \in \mathscr{X}_q^{m\times p}$ *satisfies*

$$\left|\left|I_m - \Upsilon_j(t)\partial_q h(q_j)M^{-1}(q_j)A(q_j)\right|\right| \leq \overline{\sigma} < 1\,, \tag{27}$$

$\forall t \in [0,t_\mathrm{f}], \forall j = 0,1,\ldots,$ *then*

$$\lim_{j\to+\infty} \left\|e_j(t)\right\|_\lambda = b_\mathrm{e} \in [0,\infty)\,. \tag{28}$$

Proof. Recalling (24) and expanding $\ddot{y}_\mathrm{d}$, one has

$$\begin{aligned} u_{j+1} = &\left(I_m - \Upsilon_j\partial_q h(q_j)M^{-1}(q_j)A(q_j)\right)u_j + K_\mathrm{V}\dot{e}_j + K_\mathrm{P}e_j \\ &+ \partial_q h(q_j)M^{-1}(q_j)n(q_j,\dot{q}_j) - \partial_q h(q_\mathrm{d})M^{-1}(q_\mathrm{d})n(q_\mathrm{d},\dot{q}_\mathrm{d}) \\ &+ \partial_q h(q_\mathrm{d})M^{-1}(q_\mathrm{d})A(q_\mathrm{d})u_\mathrm{d} + J^\top(q)\tau_\mathrm{ext} + \xi_j\,. \end{aligned} \tag{29}$$

From (29), one can write the following inequality

$$\begin{aligned} ||\delta u_{j+1}|| \leq &\left|\left|I_m - \Upsilon_j\partial_q h(q_j)M^{-1}(q_j)A(q_j)\right|\right| ||\delta u_j|| \\ &+ b_K\left(||\delta q_j|| + ||\delta\dot{q}_j||\right) + \|J^\top(q)\|\|\tau_\mathrm{ext}\| + b_\xi \\ &+ \left|\left|\partial_q h(q_\mathrm{d})M^{-1}(q_\mathrm{d})A(q_\mathrm{d}) - \Upsilon_j\partial_q h(q_j)M^{-1}(q_j)A(q_j)\right|\right| ||u_\mathrm{d}|| \\ &+ \left|\left|\partial_q h(q_j)M^{-1}(q_j)n(q_j,\dot{q}_j) - \partial_q h(q_\mathrm{d})M^{-1}(q_\mathrm{d})n(q_\mathrm{d},\dot{q}_\mathrm{d})\right|\right|\,, \end{aligned} \tag{30}$$

where $\delta u_{j+1} \triangleq u_j - u_\mathrm{d}$ and $b_K = \max\{\|K_\mathrm{V}\|, \|K_\mathrm{P}\|\}$. Recalling (27), Lemma 3, Property 1, and Assumption 2, (30) becomes

$$\begin{aligned} ||\delta u_{j+1}|| \leq &\sigma\,||\delta u_j|| + (b_\Upsilon b_{u_\mathrm{d}} + b_{\tau_\mathrm{ext}})\,||q_j|| + b_\xi \\ &+ \left(\frac{b_h b_n}{b_M} + b_K\right)\left(||\delta q_j|| + ||\delta\dot{q}_j||\right)\,, \end{aligned} \tag{31}$$

where $b_\Upsilon > 0$ finite is the Lipschitz-like constant of $\left|\left|\partial_q h(q_\mathrm{d})M^{-1}(q_\mathrm{d})A(q_\mathrm{d}) - \Upsilon_j\partial_q h(q_j)M^{-1}(q_j)A(q_j)\right|\right|$, whose existence is guaranteed since it is a linear combination of Lipschitz functions. Leveraging Lemma 3 leads to

$$\begin{aligned} ||\delta u_{j+1}||_\lambda \leq &\left(\sigma + b_\Upsilon b_{u_\mathrm{d}} b_{\delta q} + \left(\frac{b_h b_n}{b_M} + b_K\right)\right. \\ &\left.\times\left(b_{\delta q} + b_{\delta\dot{q}}\right)\right)||\delta u_j||_\lambda + \frac{b_0 + b_\xi t_\mathrm{f}}{\lambda} \triangleq \overline{\sigma}\,||\delta u_j||_\lambda + \overline{\xi}\,. \end{aligned} \tag{32}$$

For hypotheses $\sigma < 1$, as Lemma 3 states $\exists\lambda$ large enough such that $\overline{\sigma} < 1$ and $\frac{b_0 + b_\xi t_\mathrm{f}}{\lambda} \triangleq \overline{\xi} < 1$. Hence, considering the previous j executions when j approaches infinity yields

$$\lim_{j\to\infty} ||\delta u_{j+1}||_\lambda \leq \lim_{j\to\infty} \overline{\sigma}^j\,||\delta u_0||_\lambda + \overline{\sigma}\frac{1-\overline{\xi}^j}{1-\overline{\xi}} = \frac{\overline{\sigma}}{1-\overline{\xi}} \triangleq b_\mathrm{e}\,. \tag{33}$$

This achieves the proof. □

Remark 3. *External disturbances* $J^\top(q)\tau_\mathrm{ext}$ *(state non-repetitive disturbances) and unmodeled model mismatches (state repetitive disturbances) can be learned by the iterative law* (24) *leading to* $b_\mathrm{e} = 0$ *in* (28)*, [21]. Conversely, noise or actuator artifacts* $\xi_j(\cdot)$ *(time non-repetitive disturbances) imply a bounded convergence* (28) *[21]. If* $\xi_j(\cdot) \equiv 0, \forall t \in [0,t_\mathrm{f}], \forall j$*, Theorem 1 guarantees* $b_\mathrm{e} = 0$*, i.e., perfect tracking.*

Recalling the soft robot dynamics (3), let us propose a choice to select the learning gain $\Upsilon_j(t)$ ensuring (27).

Proposition 3. *Under the same Assumptions of Theorem 1, if one chooses the learning gain* $\Upsilon_j(t) \in \mathscr{X}_q^{m\times m}$ *such as*

$$\Upsilon_j(t) = \zeta\left(\frac{\partial_q h(q_j)A(q_j)}{b_M}\right)^+, \quad \zeta \in (0,1)\,, \tag{34}$$

$\forall t \in [0,t_\mathrm{f}], \forall j$*; then* (27) *is fulfilled and Theorem 1 holds true.*

Proof. The proof is achieved by substituting (34) in (27). □

Note that (34) does not need a precise model of the inertia matrix but only a scalar estimation of $M(q)$. However, the control law (24) exploits the knowledge of the model and it does not require output high-order derivatives [14]. Finally, the well-definiteness of (34) is assured from Assumption 1.

V. Validation

In this section, we vastly test the efficacy of the controller (24) executing regulation and trajectory tracking tasks in simulations and experiments. We compare our method with state-of-the-art ones. All unities are expressed in the MSK.

We perform three sets of tests. (i) We simulate a soft continuum inverted pendulum modeled with two PCC segments where only the second is actuated, namely 2PCC. (ii) We simulate a soft segment whose curvature is polynomial with $o = 2$ (1), namely Poly. (iii) We perform experiments using [24], which is modeled via PCC and ΔPCC [25].

We use the controller (24), $\Upsilon_j(t)$ is (34), and the control gains are listed in Tab. I. We compare the results with a PD output feedback control, i.e., $u = {}^{PD}K_P e(t) + {}^{PD}K_V\dot{e}(t)$ (Tab. I). The error is (26) and its RMS evaluates the performance.

Since for all cases it holds that $\mathrm{rank}\{A(q)\} = 3, \forall q \in \mathscr{X}_q^n$, i.e., $p = 3$, tests are performed using as output (4) $y = C_\mathrm{o}q = \sum_{j=0}^N[\phi_j, \sum_{i=0}^o \theta_{j_i}, \delta L_j]$, where N is the number of segments.

a) Actuation Matrix: Recalling (3), we describe $A(q) \in \mathscr{X}_q^{n\times m}$, which maps the control action to the continuum structure [20]. Let us consider non-extensible and frictionless pulley tendons Fig. 2. The matrix $A(q)$ can be written as $A(q) = [A_1(q),\cdots,A_m(q)]$ where $A_j(q) \in \mathscr{X}_q^n$ for $j = 1,\cdots,m$. Recalling Fig. 2, the $j-$th tendon applies in $p_{T_j} \in \mathscr{X}_q^3$ a pure force, which can be written in the local frame as $f_j = [0,0,-u_j]^\top \in \mathscr{X}_q^3$. Thus, for each j, one has $A_j(q) = \partial_{u_j}J_{p_{T_j}}^\top(q)f_j$, $J_{p_{T_j}}^\top(q) = J_p^\top(q) + \left(\partial_q R_1(q)p_{T_j}\right)^\top$, with $J_{p_{T_j}}^\top(q) \in \mathscr{X}_q^{n\times 3}$. Referring to Fig. 2, the four tendons

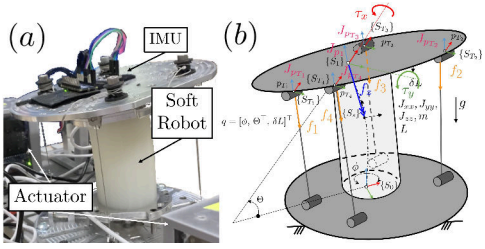


Fig. 2. (a) Experimental Platform, (b) soft robot model and tendon applied to the robot tip. The four tendons apply f_j for $j = 1,\cdots,4$ to the robot structure. They can be reorganized in 3 actions: $[\tau_x, \tau_y, f_z]^\top \in \mathscr{X}_q^3$.

$(m = 4)$ lead to three control actions, i.e., $v = [\tau_x, \tau_y, f_z]^\top \in$

$\mathscr{X}_q^3$, thus only three outputs can be assigned, i.e., $p=3$. Recalling that $\text{rank}\{A(q)\}=p$ leads to two considerations. First, depending on the polynomial order, the system (3)-(4) may be underactuated. Second, it would be useful to write (3) considering $v \in \mathscr{X}_q^p$ as input. This can be done via geometric consideration [20], i.e., $u=\bar{v}+Pv$, with $\bar{v}, v \in \mathscr{X}_q^p$ and $P \in \mathscr{X}_q^{m\times p}$; where $\bar{v}$ is a pre-compression action which is often present in these systems, and P maps the actions v into the forces u. For instance, recalling Fig. 2, P is selected as $P=-\left[-1,1,1;1,-1,1;1_{1\times 3};1_{1\times 3}\right]$.

TABLE I
CONTROL GAINS

	2PCC (HG)	Poly	*Reg.*	*Track*	*Dist.*	*In. G.*	*B.&F.*
K_P	50(200)	80	10	25	40	40	80
K_V	1(20)	1	1	1	1	3	5
${}^{PD}K_P$	1e4	1e4	–	–	–	10	–
${}^{PD}K_V$	10	10	–	–	–	1	–

A. Simulation Results: 2PCC

We here simulate two PCC segments, i.e., $o=0$, $n=6$, $N=2$, and $m=p=3$. The dynamic parameters are: $L_i=0.3$, $I_{xx_i}=I_{yy_i}=5\text{e}-3$, $I_{zz_i}=0$, $m_i=0.25$, $\kappa=10$, and $d=5$ for $i=1,2$. The selected output function is $y=\sum_{i=1}^{2}[\phi_i,\theta_i,\delta L_i]$, while $A=[I_3,I_3]^\top \in \mathbb{R}^{6\times 3}$. Note that the system is underactuated and we aim to track a minimum (min.) jerk trajectory with initial point $y_0=[0,0.02,0]$ and final $y_\text{f}=[\pi/3,\pi,-0.05]$ in $t_\text{f}=5$sec. We compare the results between (24) with $u_0(t)\equiv 0$, PD controller, (24) with (25) and low and high gains (HG case) Tab. I. Fig.3 shows the results. Fig. 3(a) reports the final output, while Fig.3(b) depicts the RMS evolution. Finally, Fig. 3(c) shows a photo sequence at the final iteration.

B. Simulation Results: Poly

We simulate the system (3) with $o=2$, i.e., $\Theta=[\theta_0,\theta_1,\theta_2]^\top$ thus $n=5$ and $N=1$. The dynamic parameters are: $L=0.3$, $I_{xx}=I_{yy}=3\text{e}-4$, $I_{zz}=0$, $m=1$, $\kappa=10$, $d=5$, and $A(q)$ is such as in Sec. V-0a Fig. 2, i.e., $m=4$ and $p=3$. The output function is $y=\left[\phi,\sum_{i=0}^{2}\theta_i,\delta L\right]^\top$. The task is the trajectory tracking of a min. jerk with $y_0=[0,0.01,0]$ and $y_\text{f}=[\pi/4,\pi/2,-3\text{e}-3]$ in $t_\text{f}=10$sec. We compare the results between (24) with $u_0(t)\equiv 0$, PD controller, and (24) with (25). Fig.4 shows the results. Fig. 4(a) reports the final output, while Fig.4(b) depicts the RMS evolution. Finally, Fig. 4(c) shows a photo sequence at the final iteration.

C. Experiment Results

The experimental platform Fig. 2 is composed of a silicon soft continuum robot [24], four Dynamixel XD430-T350-R motors, and the IMU BNO055. The motors act on the structure by means of four stainless steel tendons AISI 316, which reach the robot tip via pulleys [24].

Data acquired from the IMU sensor are converted in the PCC model via (2) [26], and used to compute the next torque control action via (24). The output function is $y=q$, and the parameters are: $L=0.098$, $I_{xx}=I_{yy}=3\text{e}-4$, $I_{zz}=0$, $m=1$, $\kappa=10$, and $d=5$ [24] and $A(q)$ is such as in Sec. V-0a.

The control effectiveness is tested via nine experiments with varying trajectories, external disturbances, unmodeled loads, and initial guesses. We also perform one test using a PD controller for comparison.

We present a regulation task, namely *Regulation*, then four trajectory tracking tasks. We perform trajectory tracking tasks with and without an unmodeled load (200g) on the robot tip, i.e., *Track*. We run tests in the presence of external disturbances, namely *Disturbance*. To exploit the benefit of using a model-based term, we compare two learnings via (25) and a null one, i.e., *Initial Guess*. Additionally, in the *Initial Guess* tasks, we implement a PD control. Finally, we execute a complex trajectory, i.e., *Back&ForthX2*, which is the most challenging task with and without an unmodeled load (200g).

External disturbances are state nonrepetitive while loads are state repetitive. Noise or artifacts are time nonrepetitive [21].

To describe the segment, we use a PCC model, i.e., $o=0$ in (3). However, to avoid numeric instability, we use its improved parametrization (see e.g., [25]), which leads to well-defined dynamics $\forall q,\dot{q}\in\mathscr{X}_q^n$. For the $i-$th segment, one has $\alpha_i=[\Delta_{x_i},\Delta_{y_i},\delta L_i]^\top$ defined as $\Delta_{x_i}=\theta_i d_{\Delta_i}\cos(\phi_i)$ and $\Delta_{y_i}=\theta_i d_{\Delta_i}\sin(\phi_i)$, where $d_{\Delta_i}=1$ [25]. The learning stops if RMS<0.1, Alg.1, and the control gains are listed in Tab. I

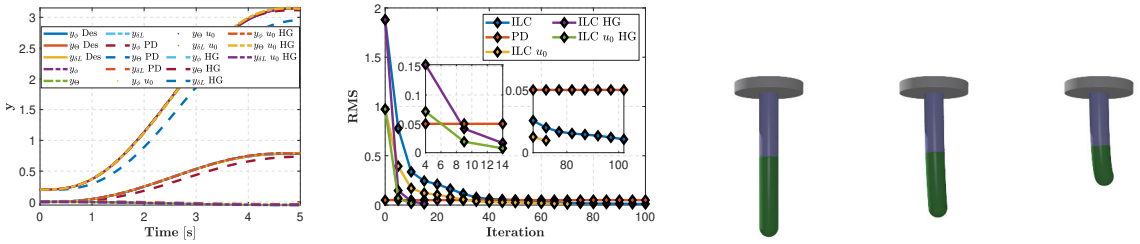
Regulation: Using a PCC model, the arm starts from $y_0=[-0.7,0.05,0]$ up to $y_\text{f}=-[\pi/2,-0.7,3\text{e}-3]$ following a constant reference and $u_0(t)\equiv 0$. The learning stops if $\|e_j(t_\text{f})\|<0.03$. Fig.5 shows the results. Fig. 5(a) reports the final output, and Fig. 5(b) depicts the RMS error. Finally, Fig. 6(a) depicts a photo sequence of the last iteration.

Track: The robot's model is ΔPCC, and it starts from $y_0=-[0.05,0.01,0]$ reaching $y_\text{f}=[0.15,0.2,-5\text{e}-3]$ via a min. jerk trajectory with $t_\text{f}=5$sec and $u_0(t)\equiv 0$. We tested the controller with and without an unmodeled load of 200g attached to the robot tip, i.e., the M test. The load represents a state-repetitive disturbance in the ILC framework [21]. Fig.5 shows the results. Fig. 5(c) reports the final output tracking, and Fig. 5(d) depicts the RMS error. Finally, Fig. 6(b) depicts a photo sequence in the M test case.

Disturbance: The robot's model is ΔPCC, and it starts from $y_0=-[0.06,0.04,0]$ up to $y_\text{f}=[0.25,0.05,-5\text{e}-3]$ via a min. jerk trajectory with $t_\text{f}=3$sec. The initial guess is model-based, i.e., (25). The operator hits/stops the robot during iterations $j=2$ and $j=4$ creating an external disturbance. This experiment validates the learning capabilities in the case of state non-repetitive disturbances [21]. Fig. 7 shows the results. Fig. 7(a) reports the final output tracking, Fig. 7(b) depicts the RMS error, and Fig. 7(c) depicts a photo sequence highlighting the disturbances at $j=4$ and the final iteration.

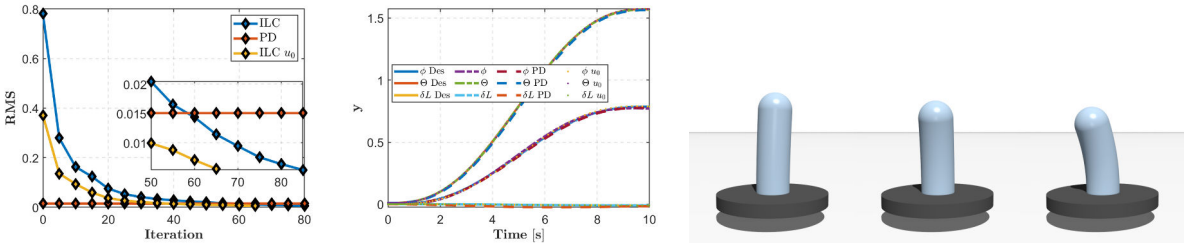
Initial Guess: The robot's model is ΔPCC, and it starts from $y_0=[0.1,-0.05,0]$ reaching $y_\text{f}=-[0.25,-0.25,5\text{e}-3]$ via a min. jerk trajectory with $t_\text{f}=5$sec. These tests are meant to highlight the advantages of using the model-based initial guess (25), i.e., the MB test, w.r.t. a null one, i.e., $u_0(t)\equiv 0$. Additionally, we implement a PD controller to compare its performance with the ILC one, Tab. I. Fig.8 shows the results. Fig. 8(a) reports the final output tracking, and Fig.s 8(b) depicts the RMS error. In Fig.s 8(a)-8(b), we report the results with (25), $u_0(t)\equiv 0$, and the PD controller.

Back&ForthX2: The robot's model is ΔPCC, and it starts from $y_0=[0.01,-0.01,0]$ swinging between the following points $y_1=-[0.25,-0.15,3\text{e}-3]$, y_0, $y_3=-[0.25,0.2,5\text{e}-3]$, and again y_0, via a composition min. jerk trajectories with $t_\text{f}=30$sec. We also perform this test with a 200g unmodeled load attached to the robot tip, namely the



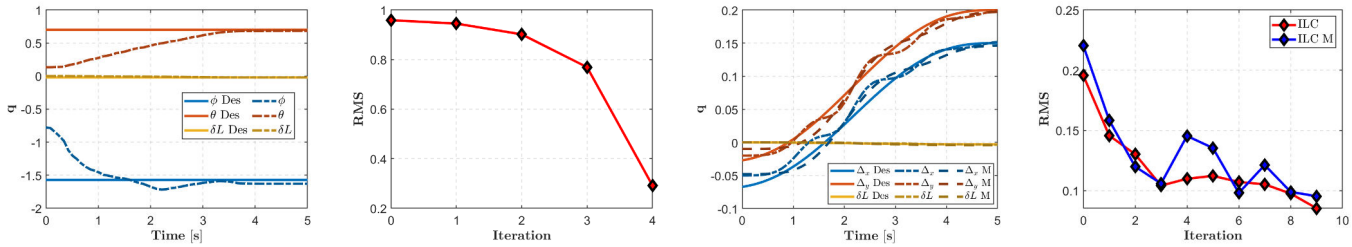
(a) Error evolution over iteration. (b) Output tracking performance. (c) Photo sequence of the last iteration.

Fig. 3. Simulation results for the soft inverted pendulum modeled with two PCC segments (purple and green) performing a trajectory tracking task.



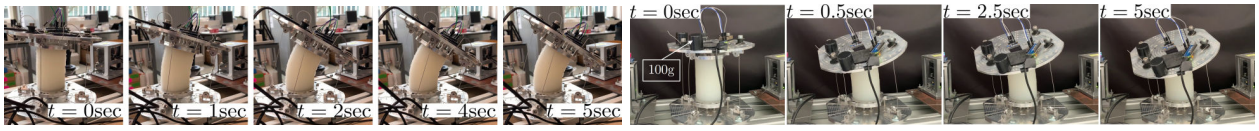
(a) Error evolution over iteration. (b) Output tracking performance. (c) Photo sequence of the last iteration.

Fig. 4. Simulation results for the polynomial curvature, i.e., $o = 2$, performing a trajectory tracking task.



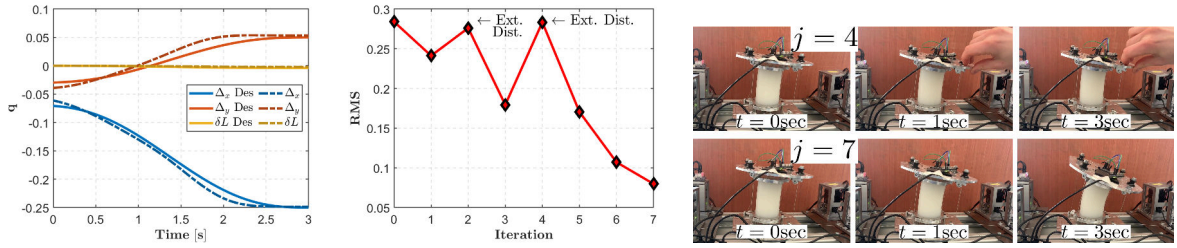
(a) Final output tracking. (b) RMS over iteration. (c) Final output tracking. (d) RMS over iteration.

Fig. 5. Experimental results for the *Regulation* Fig.s 5(a)-5(b) and *Track* tasks Fig.s 5(c)-5(d). In the *Reg.* case, the model is PCC, while in *Track* is ΔPCC.



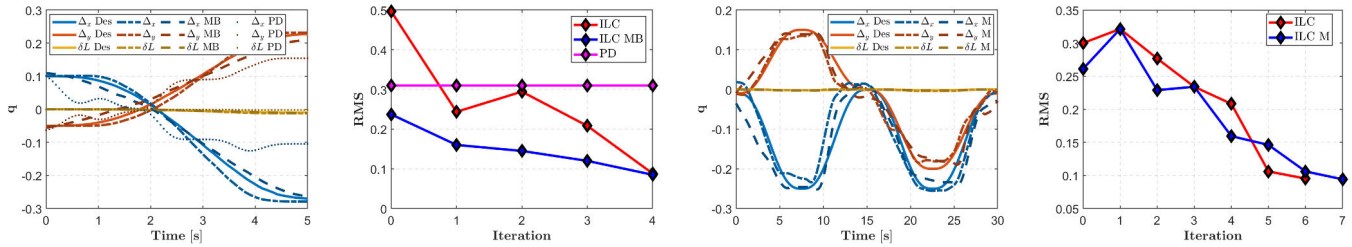
(a) Photo sequence at the final iteration *Regulation* task. (b) Photo sequence at the final iteration *Track* task.

Fig. 6. Last iteration photo sequence of the *Regulation* Fig. 6(a) and *Track* Fig. 6(b) task. The latter displays the loads (200 g) case.



(a) Final output tracking. (b) RMS over iteration. (c) Photo-sequence at iteration $j = 4$ and final *Dist.* task.

Fig. 7. Experimental results for the *Disturbance* task. The operator hits the robot at iterations $j = 2$ and $j = 4$. The model is ΔPCC.



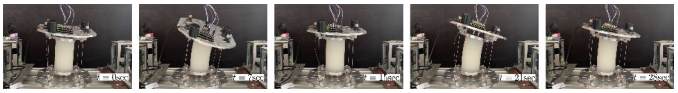
(a) Final output tracking. (b) RMS over iteration. (c) Final output tracking. (d) RMS over iteration.

Fig. 8. Experimental results for the *Initial Guess* Fig.s 8(a)-8(b) and *Back&ForthX2* Fig.s 8(a)-8(b) task. The model is ΔPCC.

M test. The initial guess is (25). Fig.8 shows the results. Fig. 8(c) reports the final output tracking, and Fig. 8(d) depicts the RMS error. Finally, Fig. 9 depicts two photo sequences of the M and regular tests Fig. 9(b) and Fig. 9(a), respectively.

Please refer to the Video attachment for further details.

(a) Photo sequence of the *Back&ForthX2* task.



(b) Photo sequence of the *Back&ForthX2* task with the load.

Fig. 9. Photo sequence of the *Back&ForthX2* task at the final iteration in the case of load (Fig. 9(b)) and without (Fig. 9(a)).

D. Discussion

Results show that the controller can reach the desired position Fig.5(a), and track the trajectory Fig.s 3(b), 4(b), 5(c), 7(a), 8(a), and 8(c) with good performances and interesting deformations Fig.s 3(a), 4(a), 5(d), 7(b), 8(b), and 8(d).

The controller (24) deals with different continuum robots even in the underactuated case Fig.s 3(a) and 4(a). Additionally, the algorithm copes with various task velocities, model mismatches, and disturbances Fig.s 5, 7, and 8.

Non-monotonic error convergence, e.g., Fig.s 7(b), is due to artifacts in the actuators, noise, and initial shift as Theorem 1 and Remark 3 claim. Thus, we define a convergence threshold, i.e., `toll`, which is always reached. For instance, in the load test, the convergence is less smooth Fig.s 5(d) and 8(d).

In the case of a null initial guess, the system does not move at the first iteration Fig.s 5(b) and 8(b). Conversely, exploiting the model-based one, i.e., (25), the RMS for $j = 0$ is reduced Fig.s 3(a), 4(a), 5(d) 7(b), 8(b), and 8(d). The ILC convergence velocity depends on the control gains Fig. 3(b) Tab. I, [14].

Finally, the feedforward nature of the controller (24) does not alter the intrinsic system elasticity preserving its safe behavior. ILC outperforms the PD's tracking performance, especially when the task velocity increases Fig.s 3, 4, and 8(b). Further, the PD controller is a feedback method, which stiffens up the robot's behavior [13] jeopardizing the task execution.

VI. Conclusion

In this work, we presented a control algorithm for soft continuum robots. We formulated a 3D polynomial curvature model. We also prove the input-to-state stability property managing locally Lipschitz nonlinearities. The main contribution of the letter is a pure feedforward torque iterative learning controller, whose convergence is established via a theoretical analysis. The effectiveness of the approach is verified via simulations and experiments. Future research will focus on enhancing the iterative learning controller by incorporating machine learning techniques further to improve the generalizability of the controller and the robot model estimation.

References

[1] C. Laschi *et al.*, "Soft robotics: Technologies and systems pushing the boundaries of robot abilities," *Science robotics*, vol. 1, no. 1, p. eaah3690, 2016.

[2] C. Armanini *et al.*, "Soft robots modeling: A structured overview," *IEEE Transactions on Robotics*, 2023.

[3] C. Della Santina *et al.*, "Model based control of soft robots: A survey of the state of the art and open challenges," *arXiv preprint arXiv:2110.01358*, 2021.

[4] S. Satheeshbabu *et al.*, "Open loop position control of soft continuum arm using deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5133–5139.

[5] D. Kim *et al.*, "Review of machine learning methods in soft robotics," *Plos one*, vol. 16, no. 2, p. e0246102, 2021.

[6] R. Morimoto *et al.*, "Model-free reinforcement learning with ensemble for a soft continuum robot arm," in *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)*. IEEE, 2021, pp. 141–148.

[7] T. G. Thuruthel *et al.*, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 124–134, 2018.

[8] J. M. Bern *et al.*, "Soft robot control with a learned differentiable model," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2020, pp. 417–423.

[9] E. Franco *et al.*, "Energy-shaping control of soft continuum manipulators with in-plane disturbances," *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 236–255, 2021.

[10] H.-S. Chang *et al.*, "Energy-shaping control of a muscular octopus arm moving in three dimensions," *Proceedings of the Royal Society A*, vol. 479, no. 2270, p. 20220593, 2023.

[11] P. Pustina *et al.*, "Feedback regulation of elastically decoupled underactuated soft robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4512–4519, 2022.

[12] O. Fischer *et al.*, "Dynamic task space control enables soft manipulators to perform real-world tasks," *Advanced Intelligent Systems*, vol. 5, no. 1, p. 2200024, 2023.

[13] F. Angelini *et al.*, "Decentralized trajectory tracking control for soft robots interacting with the environment," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 924–935, 2018.

[14] M. Pierallini *et al.*, "Iterative learning control for compliant underactuated arms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.

[15] C. Della Santina *et al.*, "Control oriented modeling of soft robots: the polynomial curvature case," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 290–298, 2019.

[16] A. D. Marchese *et al.*, "Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 1000–1019, 2016.

[17] Y. Xu *et al.*, "Impedance iterative learning backstepping control for output-constrained multisection continuum arms based on pma," *Micromachines*, vol. 13, no. 9, p. 1532, 2022.

[18] F. Stella *et al.*, "Piecewise affine curvature model: a reduced-order model for soft robot-environment interaction beyond pcc," *arXiv preprint arXiv:2211.10188*, 2022.

[19] ——, "An experimental validation of the polynomial curvature model: identification and optimal control of a soft underwater tentacle," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 410–11 417, 2022.

[20] C. Della Santina *et al.*, "Dynamic control of soft robots with internal constraints in the presence of obstacles," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6622–6629.

[21] M. Pierallini *et al.*, "A robust iterative learning control for continuous-time nonlinear systems with disturbances," *IEEE Access*, vol. 9, pp. 147 471–147 480, 2021.

[22] M. W. Spong, "Partial feedback linearization of underactuated mechanical systems," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 1. IEEE, 1994, pp. 314–321.

[23] J. Zhang *et al.*, "Convergence analysis of saturated iterative learning control systems with locally lipschitz nonlinearities," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 10, pp. 4025–4035, 2019.

[24] B. Deutschmann *et al.*, "Open source tendon-driven continuum mechanism: A platform for research in soft robotics," in *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*. IEEE, 2022, pp. 54–61.

[25] C. Della Santina *et al.*, "On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1001–1008, 2020.

[26] J. Hughes *et al.*, "Sensing soft robot shape using imus: An experimental investigation," in *Experimental Robotics: The 17th International Symposium*. Springer, 2021, pp. 543–552.