2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Trajectory Tracking of a One-Link Flexible Arm via Iterative Learning Control

Michele Pierallini[1,2], Franco Angelini[1,2,3], Riccardo Mengacci[1,2], Alessandro Palleschi[1,2], Antonio Bicchi[1,2,3], and Manolo Garabini[1,2]

***Abstract*— Trajectory tracking of flexible link robots is a classical control problem. Historically, the link elasticity was considered as something to be removed. Hence, the control performance was guaranteed by adopting high-gain feedback loops and, possibly, a dynamic compensation with the result to stiffen up the dynamic behavior of the robot. Nowadays, robots are pushed more and more towards a safe physical interaction with a less and less structured environment. Hence, the design and control of the robots moved to an on-purpose introduction of highly compliant elements in the robot bodies, the so-called soft robotics, and towards control approaches that aim to provide the tracking performance without a substantial change in the robot dynamic behavior. Following this approach, we present an iterative learning control that relies mainly on a feedforward component, hence preserves the robot dynamics, for trajectory tracking of a one-link flexible arm. We provide a condition, based on the system dynamics and similar to the Strong Inertially Coupled property, that ensures the applicability of the proposed control method. Finally, we report simulation and experimental tests to validate the theoretical results.**

I. INTRODUCTION

Recently, inspired by the whole spectrum of living beings, the research community started developing the next generation of robots: the so-called soft robots [1]. Soft robots include systems with lumped elastic elements at the joints [2] and robots characterized by continuum soft bodies [3]. The push that led to the development of soft robotics came from the challenges that robots have to face when they are asked to operate into an unstructured environment. In this case, it is paramount that the robot possesses the ability to effectively and safely manage physical interactions (with objects, environment, human beings, or other robots), and a compliant body is one of the key features to succeed in such scenario [4].

To exploit the full potential of soft robots several planning and control challenges need to be tackled.

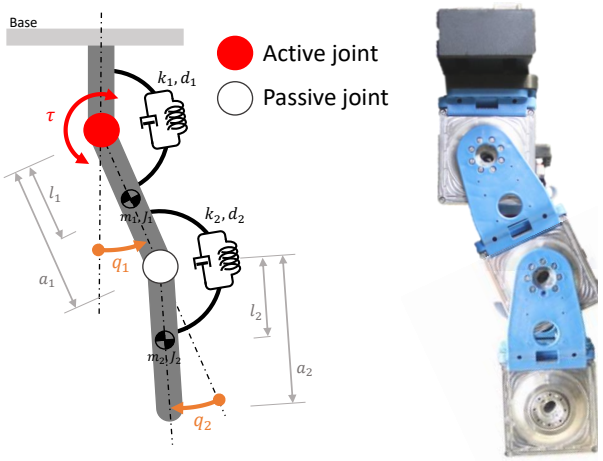
TThis research has received funding in part from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 732737 (ILIAD), No. 780883 (THING), No. 840446 (SoftHandler), No. 871237 (SOPHIA), and No. 779963 (EUROBENCH) as funded project DYSTURBANCE, and in part by the Italian Ministry of Education and Research in the framework of the CrossLab project (Departments of Excellence).

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

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

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

michele.pierallini@gmail.com



(a) Robot scheme. (b) Exp. setup.

Fig. 1. One-link flexible arm. (a) Scheme of the robot model and definition of the dynamic parameters. (b) Experimental testbed.

This work focuses on one of these: the trajectory tracking. This is a well-studied problem in the control and robotic community since the 80s [5]. At that time, the only goal in robot control was to minimize the tracking error, and the elasticity was included in the robot model to take into account dynamic effects that could compromise the performance of the controlled system. Not surprisingly, the classical control approaches for flexible robots [6], [7], [8] were built on two main ideas: the compensation of the robot dynamics and high gain feedback. Typically, these methods had the (often desired) effect of stiffening the robot behavior.

Nowadays, the scenario is different; softness is deliberately included into the robot structure and should be preserved. This idea has been articulated into a model-based paradigm [9] and a learning-based approach [10]. The former, namely elastic structure preserving (ESP), proposes a control strategy able to achieve motion tracking while preserving the elastic structure of the system and damping out the link oscillations. This is done by introducing a new set of coordinates that reflects the desired damped dynamics of the system. Conversely, the latter method does not require any accurate description of the robot model. Indeed, in [10] is proposed an approach based on iterative learning control (ILC) that can lead to good tracking performance without the alteration of the robot mechanical compliance. In fact, as

suggested in [11], the use of control techniques that mainly rely on feedforward actions with low feedback gains results in a minimal change in the robot dynamics (e.g. stiffness).

The aforementioned works are devoted to the control of robots with elasticity lumped at the joints, which are all actuated. In this work, we extend the approaches presented in [10], [12] for a one-link flexible arm that is the simplest flexible-link robot. ILC has been already applied to a flexible link arm in [13]. However, the authors consider a linear closed loop model. We here focus on the control of the absolute angle of the tip of the robot. The main challenge is given by the dependence of the relative degree on the position of the joints and on the robot dynamic parameters. The main idea to tackle this challenge relies on a condition similar to the Strong Inertially Coupled condition introduced in [14]. Conditions on the robot dynamic parameters and trajectories can be imposed to design a controller able to achieve good tracking performance. The controller we design is purely feedforward, thus it does not alter the robot impedance [11]. A further contribution of this work is to prove the convergence of the proposed iterative method, based on [15], also in presence of control gains that weigh the derivatives of the output. We extensively validate the proposed method both in simulation and on a real hardware (Fig. 1).

This paper is organized as follows. In Sec. II, we introduce the robot nonlinear dynamic model under analysis, and we define the control problem. In Sec. III, we derive the control architecture, whose effectiveness is validated in Sec. IV. Finally in Sec. V, we draw the conclusions.

II. Problem Definition

We refer to the model of flexible link robot employed in [16]. Here the system is modeled as a robot with a combination of active and passive elastic joints, i.e.

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)+D\dot{q}+Kq=F\theta\ , \qquad (1)$$

where $q,\dot{q},\ddot{q}\in\mathbb{R}^n$ are the joint position, velocity and acceleration vectors, respectively. We indicate with n_A the number of active joints, while n_P is the number of the passive ones, such that $n_\mathrm{A}+n_\mathrm{P}=n$. $M(q)\in\mathbb{R}^{n\times n}$ is the inertia matrix of the robot, $C(q,\dot{q})\in\mathbb{R}^{n\times n}$ includes centrifugal and Coriolis terms, $G(q)\in\mathbb{R}^n$ includes the gravity effect, and $D\in\mathbb{R}^{n\times n}$ is the damping matrix. K is the spring matrix, and it is such that $K=\frac{\partial V(q,\theta)}{\partial q}$, where $V(q,\theta):\mathbb{R}^n\times\mathbb{R}^{n_\mathrm{A}}\rightarrow\mathbb{R}$ is the elastic potential of a robot actuated by Series Elastic Actuators (SEAs), and $\theta\in\mathbb{R}^{n_\mathrm{A}}$ is the motor position vector. We here assume that the motor dynamics is negligible, so that θ can be considered as the control input. The matrix $F:\mathbb{R}^n\times\mathbb{R}^{n_\mathrm{A}}\rightarrow\mathbb{R}^n$ maps the active joint control input to the dynamics of all the robot joints. Note that the matrix F is such that the product $F\theta$ is a torque[1].

We follow the classic affine state-space form representation by defining the state vector $x=\left[q^T,\ \dot{q}^T\right]^T\in\mathbb{R}^{2n}$. Thus, the system (1) can be written as

$$\begin{cases}\dot{x}(t)=f(x(t))+g(x(t))u(t) & (2)\\ y(t)=h(x(t))\ , & (3)\end{cases}$$

where t is the time variable, and $f(x)$ and $g(x)$ are the drift vector field and the control vector field, respectively. The output function is $h(x)\in\mathbb{R}^{n_\mathrm{A}}$, while $u\in\mathbb{R}^{n_\mathrm{A}}$ is the control action. Given $x_0\triangleq x(0)$ as initial condition, we also define the map $\psi:\mathbb{R}^{2n}\times\mathbb{R}^{n_\mathrm{A}}\rightarrow\mathbb{R}^{2n}$, solution of the differential equation (2), i.e., $x(t)=\psi(x_0,u,t)$, and the map $\omega:\mathbb{R}^{2n}\times\mathbb{R}^{n_\mathrm{A}}\rightarrow\mathbb{R}^{n_\mathrm{A}}$ such that $y(t)=h(\psi(x_0,u,t))=\omega(x_0,u,t)$.

It is instrumental for the description of the method to define here the Lie Bracket operator, i.e., $[f,g]=\frac{\partial g(x)}{\partial x}f(x)-\frac{\partial f(x)}{\partial x}g(x)=L_fg(x)-L_gf(x)$.

We impose to the system (2)-(3) the assumptions:

A1) the number of joints is equal to $n=2$, and the number of active joints is equal to $n_\mathrm{A}=1$. The latter limits both the dimension of the input vector and of the output vector. For completeness, we report in the Appendix the explicit expression of the dynamic system under study.

A2) The maps ψ and ω are one-to-one.

A3) $f(x),g(x),h(x)$ are Lipshitzs with constants $f_0,\ g_0,\ h_0\in\mathbb{R}$, i.e., $||f(\bar{x}_1)-f(\bar{x}_2)||\leq f_0||\bar{x}_1-\bar{x}_2||,\ \forall\bar{x}_1,\bar{x}_2\in\mathbb{R}^{2n}$.

A4) The system has relative degree r ($r>0$).

Given a desired output trajectory $y_\mathrm{d}:[0,t_\mathrm{f}]\rightarrow\mathbb{R}$ (t_f terminal time), which is feasible, continuous and differentiable for, at least, r times, $\forall t\in[0,t_\mathrm{f}]$, the goal of this work is to design a controller for a system in the form (2)-(3) under assumptions A1-A4, able to track y_d. Additionally, in order to preserve the intrinsic compliance of the system, the control action must be mostly feedforward [11].

III. Problem Solution

A. Control Design

The solution we propose for the tracking problem defined in Sec. II relies on the ILC theory [17]. This framework improves the tracking performance exploiting repetitions of the desired task. The basic idea is to iteratively refine the control input given the tracking results of the previous trials, until a minimization of the error is achieved.

ILC is particularly suitable in case of compliant systems, because it is a (mostly) feedforward control approach, thus, it does not alter the robot dynamic behavior [11].

We here design a control technique based on pure feedforward ILC. Recalling the system (2)-(3) and the assumptions $A1-A4$, we define the control law[2] as

$$u_{j+1}(t)=u_j(t)+\Gamma_j(t)e_j^{(r)}(t)\ , \qquad (4)$$

where the subscript j indicates the iteration number. $\Gamma_j(t)\in\mathbb{R}$ is a learning gain which is both time and iteration varying.

The error signal $e_j^{(r)}(t)$ is defined as

[1]Note that adopting a position control input is instrumental to obtain a straightforward implementation of the method to the experimental testbed. However, analogous results could be obtained employing a torque input. In that case the map F would clearly be a selection matrix.

[2]The control law (4) requires an *initial guess* u_0. This choice is elective, however we employ the constant input required to keep the robot in the starting position of the trajectory, i.e., $y_\mathrm{d}(0)$.

$$
\begin{aligned}
e_j^{(r)}(t) &\triangleq \sum_{i=0}^{r} \gamma_i \left(y_{\mathrm{d}}^{(i)}(t) - y_j^{(i)}(t) \right) \\
&= \underbrace{\sum_{i=0}^{r} \left(L_f^{(i)} h(x_{\mathrm{d}}) - L_f^{(i)} h(x_j) \right) \gamma_i}_{\varphi(x_j, x_{\mathrm{d}})} \\
&\quad + L_g L_f^{(r-1)} h(x_{\mathrm{d}}) u_{\mathrm{d}} - L_g L_f^{(r-1)} h(x_j) u_j \,,
\end{aligned} \tag{5}
$$

where, $y_j^{(i)}(t)$ indicates the $i-$th time derivate of the output y at the $j-$th iteration, while $\gamma_i \in \mathbb{R}, \forall i = 0 \dots r$ are tunable control weights. u_{d} is the desired control input able to bring the robot to the desired state x_{d}, which is a robot state that yields to the desired output y_{d}. Note that both x_{d} and u_{d} are unknown[3].

The proposed controller is similar to those described in [15] and [18]. The main difference between this work and [15] is that [15] does not include any learning weights γ. Conversely, the learning update in [18] is based on the error of the current iteration, thus it is a feedback action, differently from (4).

Theorem 1 (Sufficient Convergence Condition)**.** *Let us consider a system in the form* (2)-(3)*, and a desired output trajectory* $y_{\mathrm{d}}(t)$*. Under assumptions A1-A4, and assuming* $x_j(0) = x_{\mathrm{d}}(0),\ \forall j$*, a sufficient condition for the convergence of the control law* (4) *is that* $\Gamma_j(t)$ *satisfies*

$$
\left| 1 - \Gamma_j(t) L_g L_f^{(r-1)} h(x_j) \right| \leq \rho < 1 \,, \forall t \in [0, t_{\mathrm{f}}], \forall j \,. \tag{6}
$$

This means that, if (6) *is fulfilled then* $e_j^{(r)}(t) \to 0$ *when* $j \to +\infty$*.*

Proof. The proof is similar to that in [15]. However in [15] there are no control weights γ. In the following we omit the time dependency for the sake of clarity.

Given the control law (4) and the definition of $\varphi(x_j, x_{\mathrm{d}})$ in (5), we have

$$
\begin{aligned}
u_{\mathrm{d}} - u_{j+1} &= \left(1 - \Gamma_j L_g L_f^{(r-1)} h(x_j) \right) (u_{\mathrm{d}} - u_j) - \Gamma_j \varphi(x_j, x_{\mathrm{d}}) \\
&\quad + \Gamma_j \left(L_g L_f^{(r-1)} h(x_j) - L_g L_f^{(r-1)} h(x_{\mathrm{d}}) \right) u_{\mathrm{d}} \,.
\end{aligned} \tag{7}
$$

Given the definitions $\delta u_j \triangleq u_{\mathrm{d}} - u_j$ and $\delta x_j \triangleq x_{\mathrm{d}} - x_j$, we can write the inequality

$$
\begin{aligned}
\left| \delta u_{j+1} \right| &\leq \left| 1 - \Gamma_j L_g L_f^{(r-1)} h(x_j) \right| \left| \delta u_j \right| + \left| \Gamma_j \right| \left| \varphi(x_j, x_{\mathrm{d}}) \right| \\
&\quad + \left| \Gamma_j \right| \left| L_g L_f^{(r-1)} h(x_j) - L_g L_f^{(r-1)} h(x_{\mathrm{d}}) \right| |u_{\mathrm{d}}| \,.
\end{aligned} \tag{8}
$$

Then, recalling (5), we can compute a constant value $\varphi_0 \in \mathbb{R}$ such that

$$
\begin{aligned}
\left| \varphi(x_j, x_{\mathrm{d}}) \right| &= \left| \sum_{i=0}^{r} \left(L_f^{(i)} h(x_{\mathrm{d}}) - L_f^{(i)} h(x_j) \right) \gamma_i \right| \\
&\leq \sum_{i=0}^{r} \left| \left(L_f^{(i)} h(x_{\mathrm{d}}) - L_f^{(i)} h(x_j) \right) \right| |\gamma_i| \\
&\leq \sum_{i=0}^{r} |\gamma_i \phi_i| \left|\left| \delta x_j \right|\right| \leq \sum_{i=0}^{r} \varphi_0 \left|\left| \delta x_j \right|\right| \,,
\end{aligned} \tag{9}
$$

where $\phi_i \in \mathbb{R}$ are constant values such that $\left| L_f^{(i)} h(x_{\mathrm{d}}) - L_f^{(i)} h(x_j) \right| \leq \phi_i \left|\left| \delta x_j \right|\right|, \forall i = 0 \dots r$ (i.e. Lipschitz). Note that we assumed the output y to be differentiable at least r times.

Let be $\eta \in \mathbb{R}$ such that $\left| L_g L_f^{(r-1)} h(x_j) - L_g L_f^{(r-1)} h(x_{\mathrm{d}}) \right| \leq \eta \left|\left| \delta x_j \right|\right|$ (i.e. Lipschitz). Let (6) be true, i.e., $1 > \rho \geq \left| 1 - \Gamma_j L_g L_f^{(r-1)} h(x_j) \right|$. Then, we can derive the following inequality

$$
\left| \delta u_{j+1} \right| \leq \rho \left| \delta u_j \right| + \left| \Gamma_j \right| (\eta |u_{\mathrm{d}}| + \varphi_0) \left|\left| \delta x_j \right|\right| . \tag{10}
$$

Defining $\mu \triangleq \sup_t \left\{ \left| \Gamma_j \right| (\eta |u_{\mathrm{d}}| + \varphi_0) \right\}$, we obtain

$$
\left| \delta u_{j+1} \right| \leq \rho \left| \delta u_j \right| + \mu \left|\left| \delta x_j \right|\right| . \tag{11}
$$

Given assumption A3, we can write the following inequality for the system (2)

$$
\begin{aligned}
\left|\left| \delta x_j \right|\right| &\leq \int_0^t \left(f_0 + g_0 |\delta u_{\mathrm{d}}(\tau)| \right) \left|\left| \delta x_j(\tau) \right|\right| \\
&\quad + \left|\left| g(x_j(\tau)) \right|\right| \left| \delta u_j(\tau) \right| \mathrm{d}\tau \,.
\end{aligned} \tag{12}
$$

Applying the Gronwall's Lemma to (12), leads to

$$
\left|\left| \delta x_j \right|\right| \leq \int_0^t c_1 \left| \delta u_j(\tau) \right| e^{c_2 (t-\tau)} \mathrm{d}\tau \,, \tag{13}
$$

where, $c_1 \triangleq \sup_t \{ f_0 + g_0 |u_{\mathrm{d}}| \}$ and $c_2 \triangleq \sup_t \left\{ \left|\left| g(x_j) \right|\right| \right\}$.

Substituting (13) in (11), leads to

$$
\left| \delta u_{j+1} \right| \leq \rho \left| \delta u_j \right| + \mu c_1 \int_0^t \left| \delta u_j(\tau) \right| e^{c_2 (t-\tau)} \mathrm{d}\tau \,. \tag{14}
$$

We define the $\lambda-$norm such as $|(\cdot)|_\lambda \triangleq \sup_t \left\{ \left| (\cdot) e^{-\lambda t} \right| \right\}$. Then, multiplying both sides of (14) for $e^{-\lambda t},\ \lambda \in \mathbb{R}^+$, we obtain

$$
\left| \delta u_{j+1} \right|_\lambda \leq \rho \left| \delta u_j \right|_\lambda + \sup_t \left\{ \mu c_1 \int_0^t e^{(c_2 - \lambda)(t-\tau)} \mathrm{d}\tau \right\} \left| \delta u_j(\tau) \right|_\lambda \,. \tag{15}
$$

Grouping for $\left| \delta u_j \right|_\lambda$ and solving the integral, leads to

$$
\left| \delta u_{j+1} \right|_\lambda \leq \left(\rho + \frac{\mu c_1 \left(1 - e^{(c_2 - \lambda) t_{\mathrm{f}}} \right)}{\lambda - c_2} \right) \left| \delta u_j \right|_\lambda \,, \tag{16}
$$

which can be rewritten as

$$
\left| \delta u_{j+1} \right|_\lambda \leq \bar{\rho} \left| \delta u_j \right|_\lambda . \tag{17}
$$

For hypothesis $\rho < 1$, then, it is always possible to find a λ such as $\lambda > c_2$, that leads to $\bar{\rho} < 1$. As a result, (16) is a control contraction, i.e., $\left| \delta u_j \right| \to 0$, and from (13) we have $\left|\left| \delta x_j \right|\right| \to 0$ for $j \to +\infty$, which implies $e_j^{(r)} \to 0$. □

[3]Note that x_{d} and u_{d} are required only to prove the convergence of the method. From a practical point of view they are not needed for the implementation of the method, which requires only the measurements of $y_j^{(i)}, \forall i = 0 \dots r, \forall j$.

Theorem 1 assures the convergence of the iterative method in the case that (6) is fulfilled. This can be achieved thanks to the following Corollary.

Corollary 1. *Let consider a system in the form* (2)-(3)*, the controller* (4) *and the same assumptions of Theorem 1. Choosing the learning gain* $\Gamma_j(t)$ *as*

$$\Gamma_j(t) = \frac{1-\varepsilon}{L_g L_f^{(r-1)} h(x_j)},\ \textit{with}\ \varepsilon \in [0,1),\ \forall t \in [0,t_\mathrm{f}]\ , \quad (18)$$

then the convergence condition (6) *holds.*

Proof. Under assumption A4, the proof comes directly by substituting (18) in (6). The complete computation is not reported here for the sake of space. □

Remark 1. *The proposed controller* (4) *is purely feedforward. This solution allows to avoid any alteration of the stiffness of the system. However, a low gain feedback could be used as in [10] or [12] but a convergence condition different from Theorem 1 should be adopted.*

B. Output Function Analysis

The choice of the desired output function $h(x)$ is elective. We here focus on the control of the end-effector angular position, i.e., the summation of the link position variables. However, at the end of the section we report some brief remarks about other output functions. Analogous results could be achieved for other choices of $h(x)$.

Theorem 2. *Let us consider a system in the form* (2)-(3)*, under assumptions A1-A4. Let be* $h(x)$ *such as*

$$y = h(x) = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} x = q_1 + q_2\ . \quad (19)$$

If

$$q_2(t) \neq \frac{\pi}{2} + \pi p, \forall p \in \mathbb{Z}, \forall t \in [0,t_\mathrm{f}], \quad (20)$$

then the result of Corollary 1 holds true.

Proof. The relative degree for the output function (19) is $r = 2$ in most cases. However, it is worth noting that it may change along the desired trajectory y_d. Therefore, it is required to study the behavior of the $L_g L_f h(x)$.

Given the dynamic matrices reported in the Appendix, we have that $\det\{M(q_2)\} = b_1 b_2 - b_3{}^2 \cos(q_2)^2$. It is worth noting that $\det\{M(q_2)\} \neq 0, \forall q$ given the symmetric structure of the matrix M. So, we have

$$y = q_1 + q_2 \implies L_g L_f h(x) = \frac{b_3 k_1 \cos(q_2)}{\det\{M(q_2)\}}\ . \quad (21)$$

This means that (21) has a direct dependency on the trajectory, and it vanishes for $q_2 = \frac{\pi}{2} + \pi p, \forall p \in \mathbb{Z}$. □

Remark 2. *If the output of interest is the active joint position, i.e.,* $y = h(x) = q_1$ *we have*

$$y = q_1 \implies L_g L_f h(x) = \frac{b_2 k_1}{det\{M(q_2)\}}\ . \quad (22)$$

In this case, the relative degree does not vary in all the workspace because $L_g L_f h(x)$ *is always non zero.*

Remark 3. *If the output of interest is the passive joint position, i.e.,* $y = h(x) = q_2$ *we have*

$$y = q_2 \implies L_g L_f h(x) = \frac{b_2 k_1 + b_3 k_1 \cos(q_2)}{det\{M(q_2)\}}\ . \quad (23)$$

Interestingly, this leads exactly to the Strong Inertially Coupled (SIC) condition proposed by Spong in [14]. This condition guarantees an inertial coupling between the active and the passive joints of the robot. In this case, the nullification of $L_g L_f h(x)$ *can be avoided with a specific design of the robot, i.e., the dynamic parameters (see Fig. 1(a) and Appendix) should be such as* $b_2 + b_3 \cos(q_2) \neq 0$*. This holds true if* $m_2 l_2^2 + J_2 > a_1 l_2 m_2$*, which is usually verified.*

IV. Validation

In this section we test the effectiveness of the proposed method on a one-link flexible arm. We validate the controller in six tasks, both in simulation and on real hardware, comparing the results. The dynamic model described in Sec. IV-A is used both for simulating the system and for tuning the gain $\Gamma_j(t)$ of the controller (4). $\Gamma_j(t)$ is chosen such that Corollary 1 holds, then fulfilling the convergence condition in Theorem 1. We set the control parameter ε in (18) as $\varepsilon = 0.9$. The value of the parameters γ is chosen depending on the task. The initial guess u_0 is chosen as the constant input required to maintain the robot in the starting position of the trajectory $y_\mathrm{d}(0)$, i.e., solving $Fu_0 = G(q(0)) + Kq(0)$.

A. Simulation Setup

We simulate a system with the form of Fig. 1(a). The joint springs are assumed linear. We employ the dynamic model reported in the Appendix. The dynamic parameters are reported in Tab. I, where m, J, l, a, k and d are the mass, inertia, length, center of mass distance, spring and damper of each link, respectively. We chose these values to have a dynamic system similar to the experimental platform (Fig. 1(b)). It is worth noting that two values are reported for the mass of the second link, because two different configurations have been tested, one with a heavier payload and one with a lighter one. The reason is that, as described in (21), Γ_j depends on the masses and the inertias.

TABLE I
DYNAMIC MODEL PARAMETERS.

	m[kg]	l[m]	a[m]	J[kgm^2]	$k[\frac{\mathrm{Nm}}{\mathrm{rad}}]$	$d[\frac{\mathrm{Nms}}{\mathrm{rad}}]$
Link 1	0.45	0.06	0.12	0.010	3	0.05
Link 2	0.45,0.15	0.06	0.12	0.005	3	0.05

B. Experimental Setup

Fig. 1(b) depicts the experimental setup. As elastic actuator we employ a *qbmove Advanced* [19]. This is a variable stiffness actuator, whose elastic transmission is realized through an antagonistic mechanism that connects two motors to the output shaft. To change the stiffness both motors have to move in opposite directions, while a movement in the same direction changes the equilibrium position. The motors and the link of the actuator are equipped with a AS5045 12

bit magnetic encoder. The elastic torque τ and the nonlinear stiffness function σ of the actuator are

$$\tau = 2\beta \cosh(\alpha\theta_s)\sinh(\alpha(q_1-\theta_e)) \quad (24)$$

$$\sigma = 2\alpha\beta \cosh(\alpha\theta_s)\cosh(\alpha(q_1-\theta_e)) , \quad (25)$$

where $\alpha = 6.7328\text{rad}^{-1}$, $\beta = 0.0222\text{Nm}$, and q_1 is the Lagrangian variable of the first link. This actuator can be controlled through θ_s and θ_e. θ_s is a parameter which tunes the desired stiffness profile and will be set constant during the experiments, while θ_e is the motor equilibrium position, and can be assumed as control input, i.e., θ in (1). It is worth noting that the resulting stiffness profile will be nonlinear.

In order to implement the passive joint, we employ a second *qbmove Advanced* actuator, where θ_s is set constant, while θ_e is set null. This pragmatic solution allows us to have a passive joint equipped with a torsional spring and a position encoder sensor.

As for the simulations, also in this case we test two configurations: heavy (450g) and light (150g) payload.

Note that the controller (4) requires up to the $r-$order differentiation of the output signal. These derivatives are numerically estimated in the experimental phase. Thus, the main differences between simulations and experiments are related to the inaccuracy of the dynamic model, inaccuracy of the estimate of the output derivatives and to the nonlinearity of the joint stiffness.

C. Tested Trajectories

Both in simulation and on the real hardware, we perform three tests in two different configurations. In particular, we use three different trajectories and two different payloads. The payload is 450g in the first case and 150g in the second case. As output function $h(x)$ we employ the absolute angle of the tip of the robot, i.e., (19).

As desired trajectories we use:

1) a sinusoidal signal lasting for $t_f = 6\text{s}$, i.e.,
$$y_d(t) = \frac{\pi}{8}\cos(t+\pi)+\frac{\pi}{8} ; \quad (26)$$
2) a minimum jerk signal that starts from the initial position $y_0 = 0$ and reaches the final position $y_f = \frac{\pi}{4}$ in $t_f = 10\text{s}$, i.e.,
$$y_d(t) = y_f\left(10\left(\frac{t}{t_f}\right)^3 - 15\left(\frac{t}{t_f}\right)^4 + 6\left(\frac{t}{t_f}\right)^5\right) ; \quad (27)$$
3) a minimum jerk signal analogous to (27) but with $t_f = 1\text{s}$. This leads to a more challenging task.

The tested trajectories are such that Theorem 2 is valid, thus the relative degree of the system is always $r = 2$ along the trajectories. In each trial the starting configuration is $x(0) = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}^T$.

Tab. II lists the control parameters $[\gamma_0, \gamma_1, \gamma_2]$ employed in each simulative and experimental trial.

In order to quantify the tracking performance of the controller, we use as metric the root mean square (RMS) error.

TABLE II
SIMULATIONS AND EXPERIMENTS CONTROL WEIGHTS ($[\gamma_0,\ \gamma_1,\ \gamma_2]$).

Simulations	Sinusoid	Slow min. jerk	Fast min. jerk
Heavy Payload	[250, 10, 1]	[250, 10, 1]	[250, 10, 1]
Light Payload	[250, 10, 1]	[250, 10, 1]	[250, 10, 1]
Experiments	Sinusoid	Slow min. jerk	Fast min. jerk
Heavy Payload	[200, 0.5, 0.5]	[250, 0.5, 0.5]	[80, 1, 0.1]
Light Payload	[150, 5, 0.3]	[250, 5, 0.3]	[200, 5, 0.3]

D. Simulation and Experimental Results

1) Heavy Payload: In the experiments the stiffness parameter is set to $\theta_s = 0.524\text{rad}$ for both joints, that leads to a joint stiffness equal to 5.04Nm/rad in case of zero deflection.

In the case of the sinusoidal reference (26), the iterative method is executed for 15 iterations in the simulation and for 30 in the experiment. Fig. 2 compares the simulation and experimental results. Fig. 2(a) shows error evolution over iterations, while Fig. 2(b) reports the tracking performance at the last iteration. Figures 2(c-d) compares the joint evolution obtained in simulation and in the experiment.

In the case of the slow minimum jerk reference (27), the iterative method is executed for 15 iterations in the simulation and for 30 in the experiment. Fig. 3 compares the simulation and experimental results. Fig. 3(a) shows error evolution over iterations, while Fig. 3(b) reports the tracking performance at the last iteration. Figures 3(c-d) compares the joint evolution obtained in simulation and in the experiment. Fig. 4 shows a photo-sequence of the trajectory execution at the last iteration.

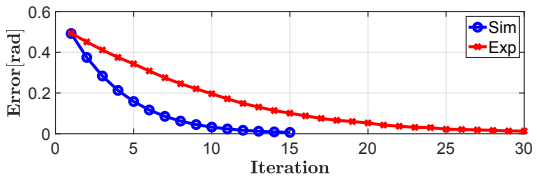
In the case of the fast minimum jerk reference (27), the iterative method is executed for 15 iterations in the simulation. For the experimental trial only 20 iterations were performed since the method already converged. Fig. 5 compares the simulation and experimental results. Fig. 5(a) shows error evolution over iterations, while Fig. 5(b) reports the tracking performance at the last iteration. Figures 5(c-d) compares the joint evolution obtained in simulation and in the experiment.

2) Light Payload: In the experiments the stiffness parameter is set lower, i.e., to $\theta_s = 0.35\text{rad}$ for both joints. This leads to a joint stiffness equal to 1.57Nm/rad in case of zero deflection.

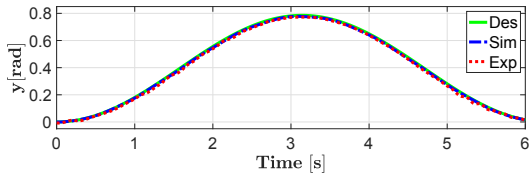
In the case of the sinusoidal reference (26), the iterative method is executed for 15 iterations in the simulation and for 30 in the experiment. Fig. 6 compares the simulation and experimental results. Fig. 6(a) shows error evolution over iterations, while Fig. 6(b) reports the tracking performance at the last iteration. Figures 6(c-d) compares the joint evolution obtained in simulation and in the experiment.

In the case of the slow minimum jerk reference (27), the iterative method is executed for 15 iterations in the simulation and for 30 in the experiment. Fig. 7 compares the simulation and experimental results. Fig. 7(a) shows error evolution over iterations, while Fig. 7(b) reports the tracking performance at the last iteration. Figures 7(c-d) compares the joint evolution obtained in simulation and in the experiment.

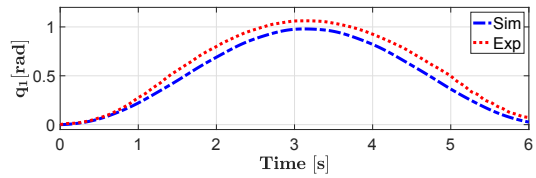
In the case of the fast minimum jerk reference (27),



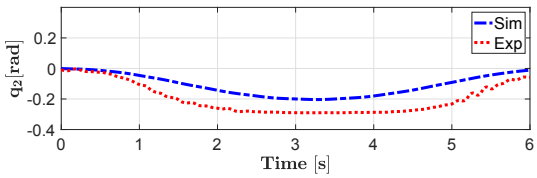
(a) Error evolution over iteration.



(b) Output tracking perforamnce.

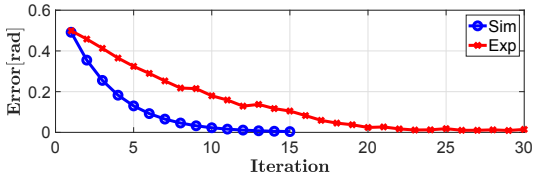


(c) Joint 1 position evolution.

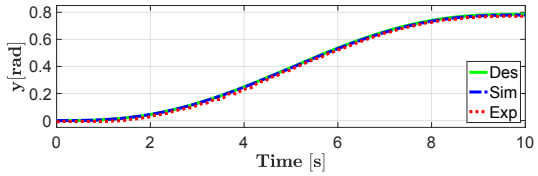


(d) Joint 2 position evolution.

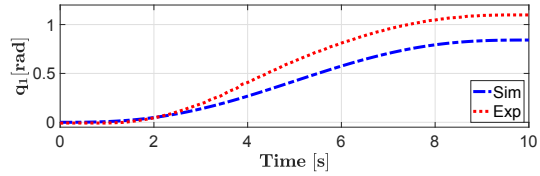
Fig. 2. Simulation and experimental results for the sinusoidal trajectory in the heavy payload scenario.



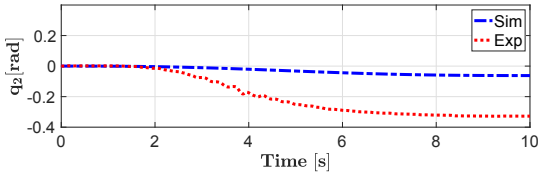
(a) Error evolution over iteration.



(b) Output tracking perforamnce.



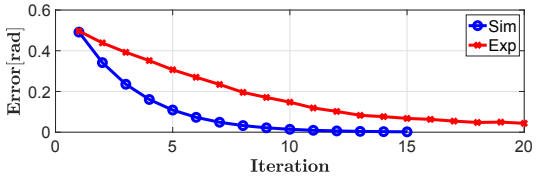
(c) Joint 1 position evolution.



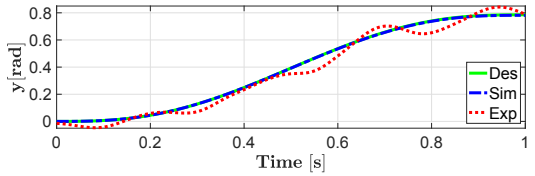
(d) Joint 2 position evolution.

Fig. 3. Simulation and experimental results for the slow minimum jerk trajectory in the heavy payload scenario.

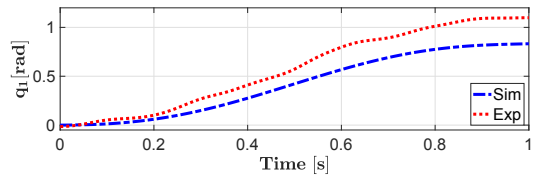
Fig. 4. Photo-sequence of the the last iteration of the slow minimum jerk trajectory in the heavy payload scenario.



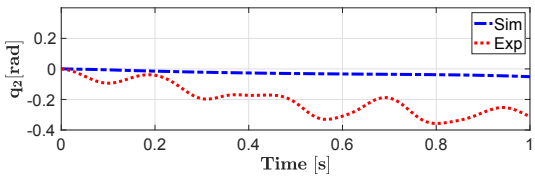
(a) Error evolution over iteration.



(b) Output tracking perforamnce.

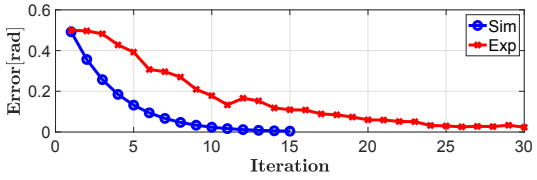


(c) Joint 1 position evolution.

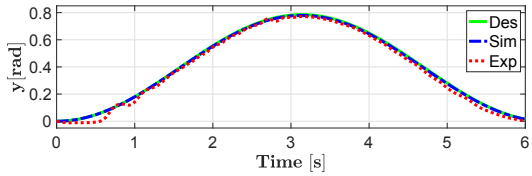


(d) Joint 2 position evolution.

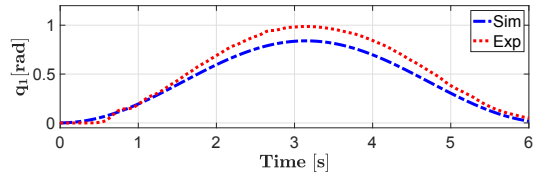
Fig. 5. Simulation and experimental results for the fast minimum jerk trajectory in the heavy payload scenario.



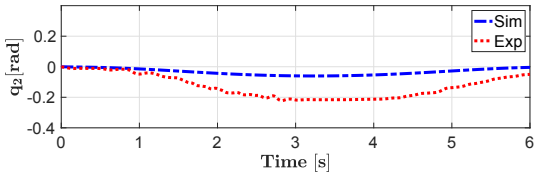
(a) Error evolution over iteration.



(b) Output tracking perforamnce.

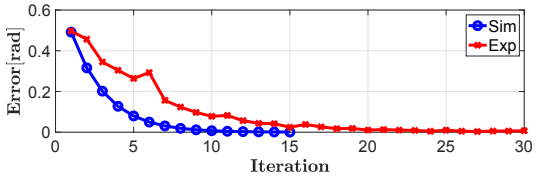


(c) Joint 1 position evolution.

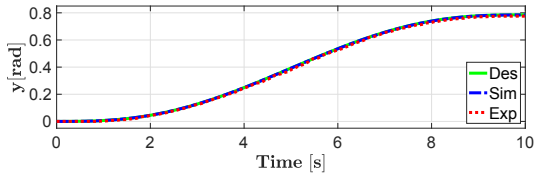


(d) Joint 2 position evolution.

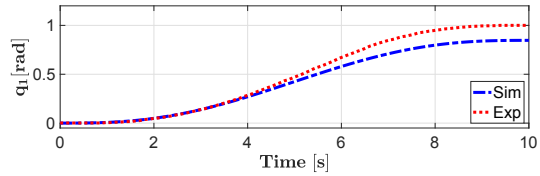
Fig. 6. Simulation and experimental results for the sinusoidal trajectory in the light payload scenario.



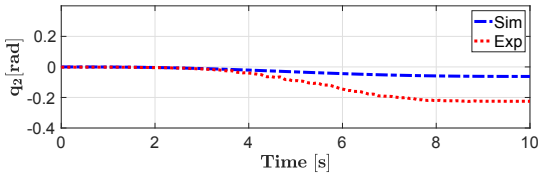
(a) Error evolution over iteration.



(b) Output tracking perforamnce.



(c) Joint 1 position evolution.



(d) Joint 2 position evolution.

Fig. 7. Simulation and experimental results for the slow minimum jerk trajectory in the light payload scenario.

the iterative method is executed for 15 iterations in the simulation. For the experimental trial only 20 iterations were performed since the method already converged. Fig. 8 compares the simulation and experimental results. Fig. 8(a) shows error evolution over iterations, while Fig. 8(b) reports the tracking performance at the last iteration. Figures 8(c-d) compares the joint evolution obtained in simulation and in the experiment.

E. Discussion

Several choices for the initial guess u_0 could be made to speed up the learning process. However, we decided to test the method in the worst case scenario, i.e., a constant initial guess equal to the equilibrium torque. This is the reason why the error at the first iteration is equal for each pair of simulation and experiment.

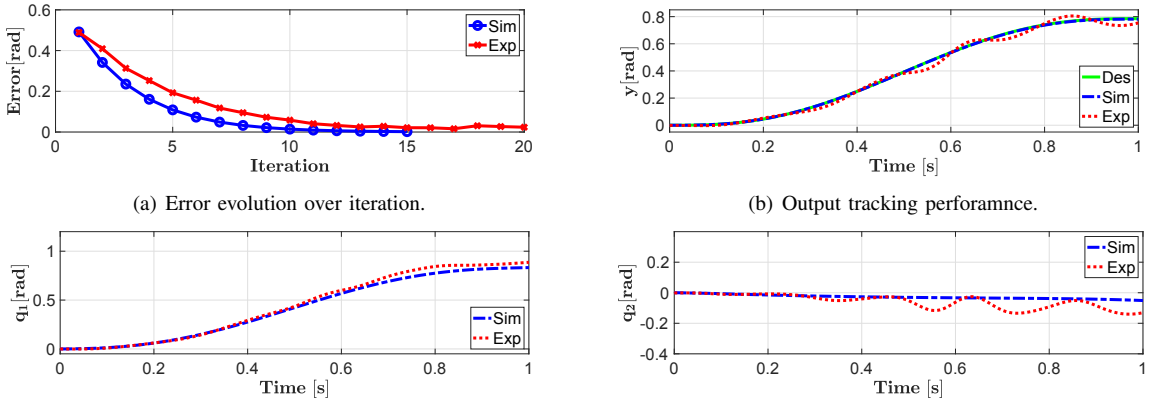
Results show that the proposed method is able to improve the tracking error, achieving satisfying tracking performance in all the trials. This means that, even though the second joint is passive, the controller is able to execute the task. In the simulation case, the learning convergence is smoother, thanks to the perfect match between the model and the knowledge of the controller, and thanks to the good estimation of the derivatives. However, also in the experiments the method converges to comparably good tracking results. The RMS error obtained at the last iteration averaged between all trials is 0.003rad for the simulations and 0.018rad for the experiments.

Note that $q_2 \neq \pi/2$ holds for all the performed tasks, thus Theorem 2 is valid, and the relative degree is $r = 2$.

The most challenging task performed is the fast minimum jerk (Fig. 5 and Fig. 8). In the simulations, this is perfectly executed, while in the experiments there are slight oscillations in both configurations. This result can be linked to the velocity and acceleration of the reference signal. Indeed, the numerical estimate of these signals could reduce the tracking performance.

V. CONCLUSIONS

This work deals with trajectory tracking of a one-link flexible arm. The system is modeled with two elastic joints, the first one is active and the second one is passive. The goal is to obtain a controller able to achieve good tracking performance while preserving the natural compliance of the system. The proposed solution is a feedforward control approach based on iterative learning control. We guarantee the convergence of the method and the feasibility of the



(a) Error evolution over iteration.

(b) Output tracking perforamnce.

(c) Joint 1 position evolution.

(d) Joint 2 position evolution.

Fig. 8. Simulation and experimental results for the fast minimum jerk trajectory in the light payload scenario.

approach via a condition on the dynamic parameters of the system and the trajectory.

Future extensions of this work will analyze the case of multiple passive joints, will investigate also the applicability of model-free and neural network approaches [20].

APPENDIX

We report here the full description of the considered one-link flexible arm dynamics (Fig.1(a)). The robot elastic potential can be written as

$$V(q,\theta) = 0.5\ k_2\ q_2^2\ +\ 0.5\ k_1\ (q_1-\theta)^2\ .$$

The dynamic matrices are

$$D = \mathrm{diag}([d_1,d_2]),\ K = \mathrm{diag}([k_1,k_2]),\ F = [k_1,\ 0]^T\ ,$$

$$M(q) = \begin{bmatrix} b_1+b_2+2b_3\cos(q_2) & b_2+b_3\cos(q_2) \\ b_2+b_3\cos(q_2) & b_2 \end{bmatrix},$$

$$C(q,\dot{q}) = \begin{bmatrix} -b_3\dot{q}_2\sin(q_2) & -b_3\sin(q_2)(\dot{q}_1+\dot{q}_2) \\ b_3\dot{q}_1\sin(q_2) & 0 \end{bmatrix},$$

$$G(q) = \begin{bmatrix} g_0\left(m_2\left(l_2\sin(q_1+q_2)+a_1\sin(q_1)\right)+l_1m_1\sin(q_1)\right) \\ g_0 l_2 m_2\sin(q_1+q_2) \end{bmatrix},$$

where $b_1 = m_2{a_1}^2 + m_1{l_1}^2 + J_1$, $b_2 = m_2{l_2}^2 + J_2$ and $b_3 = a_1 l_2 m_2$.

REFERENCES

[1] Alin Albu-Schaffer, Oliver Eiberger, Markus Grebenstein, Sami Haddadin, Christian Ott, Thomas Wimbock, Sebastian Wolf, and Gerd Hirzinger. Soft robotics. *IEEE Robotics & Automation Magazine*, 15(3):20–30, 2008.

[2] Bram Vanderborght, Alin Albu-Schäffer, Antonio Bicchi, Etienne Burdet, Darwin G Caldwell, Raffaella Carloni, MG Catalano, Oliver Eiberger, Werner Friedl, Ganesh Ganesh, et al. Variable impedance actuators: A review. *Robotics and autonomous systems*, 61(12):1601–1614, 2013.

[3] Daniela Rus and Michael T Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467–475, 2015.

[4] Rolf Pfeifer and Gabriel Gómez. Morphological computation–connecting brain, body, and environment. In *Creating brain-like intelligence*, pages 66–83. Springer, 2009.

[5] Steven D Eppinger, Warren P Seering, et al. Three dynamic problems in robot force control. *IEEE Transactions on Robotics and Automation*, 8(6):751–758, 1992.

[6] Patrizio Tomei. A simple pd controller for robots with elastic joints. *IEEE Transactions on automatic control*, 36(10):1208–1213, 1991.

[7] Alessandro De Luca and Pasquale Lucibello. A general algorithm for dynamic feedback linearization of robots with elastic joints. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 1, pages 504–510. IEEE, 1998.

[8] Jong H Oh and Jin S Lee. Control of flexible joint robot system by backstepping design approach. *Intelligent Automation & Soft Computing*, 5(4):267–278, 1999.

[9] Manuel Keppler, Dominic Lakatos, Christian Ott, and Alin Albu-Schäffer. Elastic structure preserving (esp) control for compliantly actuated robots. *IEEE Transactions on Robotics*, 34(2):317–335, 2018.

[10] Franco Angelini, Cosimo Della Santina, Manolo Garabini, Matteo Bianchi, Gian Maria Gasparri, Giorgio Grioli, Manuel Giuseppe Catalano, and Antonio Bicchi. Decentralized trajectory tracking control for soft robots interacting with the environment. *IEEE Transactions on Robotics*, 34(4):924–935, 2018.

[11] Cosimo Della Santina, Matteo Bianchi, Giorgio Grioli, Franco Angelini, Manuel Catalano, Manolo Garabini, and Antonio Bicchi. Controlling soft robots: balancing feedback and feedforward elements. *IEEE Robotics & Automation Magazine*, 24(3):75–83, 2017.

[12] R. Mengacci, F. Angelini, M.G. Catalano, G. Grioli, A. Bicchi, and M. Garabini. On the motion/stiffness decoupling property of articulated soft robots with application to model-free torque iterative learning control. *The International Journal of Robotics Research, (cond. accepted https://tinyurl.com/uky647a)*.

[13] M Poloni and G Ulivi. Iterative learning control of a one-link flexible manipulator. In *Robot Control 1991*, pages 393–398. Elsevier, 1992.

[14] Mark W Spong. Partial feedback linearization of underactuated mechanical systems. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, volume 1, pages 314–321. IEEE, 1994.

[15] Hyun-Sik Ahn, Chong-Ho Choi, and Kwang-bae Kim. Iterative learning control for a class of nonlinear systems. *Automatica*, 29(6):1575–1578, 1993.

[16] A De Luca, L Lanari, and G Ulivi. Output regulation of a flexible robot arm. In *Analysis and Optimization of Systes*, pages 833–842. Springer, 1990.

[17] Douglas A Bristow, Marina Tharayil, and Andrew G Alleyne. A survey of iterative learning control. *IEEE control systems magazine*, 26(3):96–114, 2006.

[18] Mingxuan Sun and Danwei Wang. Closed-loop iterative learning control for non-linear systems with initial shifts. *International Journal of Adaptive Control and Signal Processing*, 16(7):515–538, 2002.

[19] Cosimo Della Santina, Cristina Piazza, Gian Maria Gasparri, Manuel Bonilla, Manuel Giuseppe Catalano, Giorgio Grioli, Manolo Garabini, and Antonio Bicchi. The quest for natural machine motion: An open platform to fast-prototyping articulated soft robots. *IEEE Robotics & Automation Magazine*, 24(1):48–56, 2017.

[20] Tong Yang, Ning Sun, He Chen, and Yongchun Fang. Neural network-based adaptive antisiwing control of an underactuated ship-mounted crane with roll motions and input dead zones. *IEEE Transactions on Neural Networks and Learning Systems*, 31(3):901–914, 2019.