Time Generalization of Learned Trajectories on a One-Link Flexible Arm

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

***Abstract*— The elasticity of soft robots is influenced by the employed controller. High-gain feedback techniques can stiffen up the behavior of the robot. On the other hand, learning-based controllers play a crucial role in the trajectory tracking of flexible arm robots. They mainly rely on feedforward components and do not need an accurate description of the model. Hence, they achieve good tracking performances and result in a minimal change of the dynamics of the systems. However, any learning algorithm presents generalization issues: any slight changes in the desired task leads to a completely new learning phase. Motivated by this, we focus on the generalization of a control action to perform trajectories with a different final time. In particular, we introduce a model of flexible arm robot, then we give a necessary and sufficient condition to generalize the control action without any knowledge of the dynamic model. Additionally, we report a learned-based algorithm to computed the control action. Finally, we validate the generalization algorithm simulating a one-link flexible arm, modeled as two degrees of freedom chain, in which the first joint is active, while the second is passive.**

I. Introduction

Soft robots include systems with elastic elements lumped at the joints [1] (i.e. articulated soft robots) and robots characterized by continuously deformable bodies [2] (i.e. continuum soft robots). Controlling soft robots is a challenge yet to be tackled. In particular, three are the main issues [3].

First, flexible robots are characterized by underactuated dynamics [4]. It is not possible to independently associate the acceleration of each joint [5].

Another challenge is that the control action affects both the performance and the flexible arm elastic behavior. Historically, control techniques use compensation of the dynamics and high gain feedback loop, stiffening the robot behavior [6] and [7]. Conversely, feedforward approaches do not change in the dynamic of the robot [8].

The third challenge for flexible arms is obtaining a reliable model of the system [9].

A class of learned-based controllers, namely Iterative Learning Controllers (ILC), tackles these three challenges: the control action is (almost) model-free and feedforward, and dealing with the system underactuation through a dependence on the system relative degree [10].

This 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), and No. 871237 (SOPHIA), 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

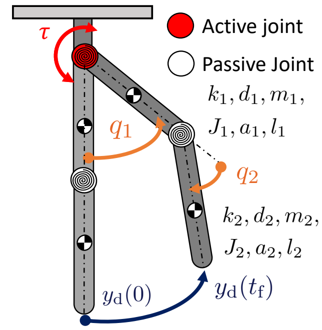


Fig. 1. Model of a one-link flexible arm.

To solve this issue, several generalization algorithms for articulated soft robots have been proposed. In [11] the Authors generalize the control action w.r.t. the velocity execution of the trajectory. In [12], the generalization approach regards the stiffness of the joints. Finally, [13] proposes a method to track a novel trajectory exploiting previous examples and without new learning processes.

In this work, we extend the time generalization approach presented in [11] for one-link flexible arms. In particular, we introduce the model of a flexible arm, then we draw a necessary and sufficient condition on the number of experiments needed to generalized a learned based control action as the one proposed in [14].

II. Problem Definition

We model the one-link flexible arm such according to [4]. Indeed, the continuum arm is discretized as a robot with a combination of active and passive elastic joints, i.e.

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

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 $n_A + n_P = n$. $M(q) \in \mathbb{R}^{n\times n}$ and $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ are the inertia and Coriolis matrix of the robot, respectively. $G(q) \in \mathbb{R}^n$ includes the gravity effect, and $D, K \in \mathbb{R}^{n\times n}$ are the diagonal damping and stiffness matrix. In torque term $F\theta$, θ is the motor position, which is the control input, and $F : \mathbb{R}^n \times \mathbb{R}^{n_A} \to \mathbb{R}^n$ indicates the underactuation matrix.

The output function can be written with a combination of joint positions, i.e.,

$$y = Tq \; , \tag{2}$$

where $T \in \mathbb{R}^{n_A \times n}$ is a linear constant map.

Let us assume what follow.

A1) The system (1) is SISO (single input single output), i.e., $n_A = 1$.
A2) The system (1) is smooth in the sense of $||C(q_a, \dot{q}_a) - C(q_b, \dot{q}_b)|| \leq C_0 \, ||q_a - q_b|| + C_1 \, ||\dot{q}_a - \dot{q}_b||$, with $C_0, \; C_1 \in \mathbb{R}$.
A3) The system (1) has a relative degree equal to r.
A4) Given a database of learned controllers, we indicate with $\mathrm{r_e}$ a generic experiment and with ω the number of experiments, i.e., $\mathrm{r_e} = 1, \cdots, \omega$.
A5) The desired trajectory to track $y_d : [0, t_f] \to \mathbb{R}$ (t_f terminal time), is feasible, continuous, r times differentiable $\forall t \in [0, t_f]$ and such that $y_d = y(\alpha_T t)$.

Applying the procedure in [11], for a one desired trajectory $y(\alpha_T t)$ scaled from an $\alpha_T \in \mathbb{R}$ factor leads to

$$\begin{cases} \hat{y}(\alpha_T t) = y(\alpha_T t) \; , & (3) \\ \dot{\hat{y}}(\alpha_T t) = \dot{y}(\alpha_T t) = \alpha_T \dfrac{\mathrm{d}y(\alpha_T t)}{\mathrm{d}(\alpha_T t)} \; , & (4) \\ \ddot{\hat{y}}(\alpha_T t) = \ddot{y}(\alpha_T t) = \alpha_T^2 \dfrac{\mathrm{d}^2 y(\alpha_T t)}{\mathrm{d}(\alpha_T t)^2} \; . & (5) \end{cases}$$

Given a ω-dim set of controllers and given the desired trajectory y_d, the goal of this work is to design a model-free and feedforward control action able to track y_d, using the ω-dim database.

III. Problem Solution

In this section, inspired by [11] and [15], we propose a method to generalize the task velocity execution.

Theorem 1. *Let be the system* (1)*, let be* (2) *the output function and let $r_e = 1, \ldots, \omega$ be the $r_e - th$ experiment. In order to time scaling the desired trajectory the necessary and sufficient number of distinct experiments ω to express analytically $F\theta$ without any knowledge of the model is 3.*

Proof. The $i-$th line of the dynamics in (1) can be written as

$$\begin{aligned} &\sum_j M_{ij}(q)\ddot{q}_j + \sum_{jl} \Gamma_{ijl}(q)\dot{q}_j\dot{q}_l \\ &\qquad + G_i(q) + D_i\dot{q}_i + K_i q_i = F_i\theta_i(t) = \tau_i(t) \; , \end{aligned} \tag{6}$$

where Γ_{ijl} are the Christoffel symbols.

Substituting (6) and (3)-(5) in (2) leads to

$$\begin{aligned} &\left(\sum_{ij} T_i M_{ij} \ddot{\hat{q}}_j + \sum_{ijl} T_i \Gamma_{ijl} \dot{\hat{q}}_j \dot{\hat{q}}_l \right) {}^{\mathrm{r_e}}\alpha_T^2 + \left(\sum_i T_i D_i \dot{\hat{q}}_i \right) {}^{\mathrm{r_e}}\alpha_T \\ &\qquad + \sum_i \left(T_i G_i + T_i K_i \hat{q}_i \right) = \tau_{\mathrm{r_e}} \; , \end{aligned} \tag{7}$$

with $i, j, l = 1, \cdots, n$.

Since the system is SISO, (7) is a scalar equation, which can be written as a polynomial in ${}^{\mathrm{r_e}}\alpha_T$

$${}^{\mathrm{r_e}}\alpha_T^2 \xi_2 + {}^{\mathrm{r_e}}\alpha_T \xi_1 + \xi_0 = \tau_{\mathrm{r_e}} \; , \tag{8}$$

where $\xi_0, \xi_1, \xi_2 \in \mathbb{R}$ are functions of time.

Defining $\xi = [\xi_0, \xi_1, \xi_2]^\top \in \mathbb{R}^3$, (8) represents the $\mathrm{r_e}-$th line of a linear system. Indeed, Performing $\mathrm{r_e} = 1, \; 2, \; 3$ experiments, leads to the linear system

$$P\xi(t) = \tau(t) \; . \tag{9}$$

We are interested to determinate the $\xi(t)$ vector, so (9) has to be invertible. The matrix $P \in \mathbb{R}^{3\times 3}$ is a Vandermonde matrix which is alway invertible if and only if ${}^{\mathrm{r_e}}\alpha \neq {}^{r_i}\alpha, \forall \mathrm{r_e} \neq r_i$. □

In order to compute, the control τ_d such as $y_d = h_d(t) = h(t\alpha_T)$ can be found as

$$\begin{cases} \xi = P^{-1}\tau \\ \tau_d(t) = \left[\alpha_T^2, \;\; \alpha_T \;\; 1 \right] \xi \end{cases} . \tag{10}$$

Remark 1. *Eq.* (10) *leads to a feedforward action. The method only depends on the database, i.e. τ, which collects the employed controller.*

In [14], we proposed a pure feedforward ILC-based control law and we proved its convergence via inertial conditions. We here summarize the result of [14] in the following Theorem, which is used to compute $\tau_{\mathrm{r_e}}$ $\forall \mathrm{r_e}$.

Theorem 2. *Let be system* (1)*, and a desired output $y_d(t)$. Assuming $q_j(0) = q_d(0)$, $\forall j$ (=iteration index). The control law*

$$\theta_{j+1}(t) = \theta_j(t) + \frac{\varepsilon}{TM(q)^{-1}F} \sum_{i=0}^{r} \gamma_i \left(y_d^{(i)}(t) - y_j^{(i)}(t) \right) \; , \tag{11}$$

with $\varepsilon \in [0, 1)$, $\gamma_i \in \mathbb{R}, i = 1, \cdots, r$ control gains and $y_j^{(i)}$ the $i-$th derivative of the output, is convergent.

Proof. See [14]. □

IV. Validation

In this section, we test the effectiveness of the generalization method. We simulate a system as depicted in Fig. 1, where the first joint is active and the second is passive. The dynamic model is reported in [14].

The dynamic parameters $m = 0.45$kg, $J = 0.01$kgm^2, $l = 0.06$m, $a = 0.12$m, $k = 3$N/rad and $d = 0.05$Ns/rad are the mass, inertia, length, center of mass distance, spring and damper of each link, respectively.

The value of the parameters γ in (11) are chosen such as $\gamma = [\gamma_0, \; \gamma_1, \; \gamma_2] = [100, \; 10, \; 1]$. Since the starting position of the robot is $x(0) = 0_{4\times 1}$, the initial guess θ_0 in (11) is null.

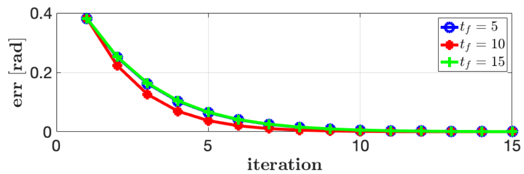
In order to quantify the tracking performance, the root mean square (RMS) is employed.

We here focus on the angular position of the robot, i.e., $T = \begin{bmatrix} 1 & 1 \end{bmatrix}$, and $q = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^\top$ in (2). Hence, the relative degree r of the system is such as $r = 2$.

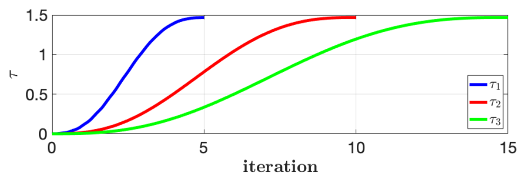
As desired trajectory, we use a minimum jerk signal that starts from the initial position $y_0 = 0$ and reaches the final position $y_f = \frac{\pi}{4}$.

The ILC database is obtained performing $\omega = 3$ minimum jerk trajectories with different final time, in particular $t_f = \{5, 10, 15\}$s so $\alpha_T = \left\{ \frac{1}{2}, \; 1, \; \frac{3}{2} \right\}$ w.t.r. of the 10s trajectory.

Then, we tested the generalizating approach applying (10) for 25 scaled factors $\alpha_T = \left\{\frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \cdots, \frac{5}{2}\right\}$ with the respect of the 10s minimum jerk.



(a) Error evolution over iteration.



(b) ILC Database.

Fig. 2. $R\overline{R}$: simulation results for the minimum jerk trajectory.

Fig. 2 reports the simulative results of the iterative learning process. Fig. 2(a) shows error evolution over iterations, while Fig. 2(b) reports the torques computed at the last iteration, which are collected in $\tau = [\tau_1,\ \tau_2,\ \tau_3]^\mathsf{T}$ of (10).

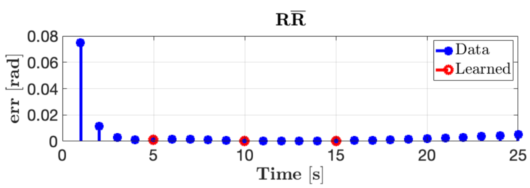


Fig. 3. RMS over time execution.

Fig. 3 depicts the RMS for all the tested trajectories testing the 25 scaled trajectories.

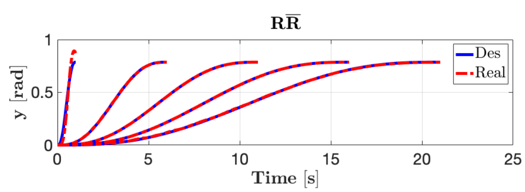


Fig. 4. Output tracking performance.

Finally, Fig. 4 reports a subset, i.e. $\alpha_T = \left\{\frac{1}{10}, \frac{3}{5}, \ \frac{11}{10}, \ \frac{21}{10}\right\}$, of the tracking performance at the last iteration.

Results show that the proposed method is able to achieve good tracking performances, without any description of the model, while preserving the elastic behavior and compensating the presence of passive elements in the chain.

Not surprisingly, the best tracking performances are in the middle of the already learned torques: the error grows linearly from $t_f = 15$sec.

The RMS error averaged in all the 25 scale simulation trials is about 0.0032rad.

V. Conclusion

This article proposes an algorithm to generalize acquired control inputs w.r.t. the velocity of the desired trajectory for a one-link flexible arm. Results show that the minimum number of experiments required to achieve velocities generalization is three. The proposed method is feedforward and model-free. Simulation results show that it archives good tracking performances.

Future work will study the generalization of learned trajectories w.r.t. stiffness behavior and space trajectory.

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

[3] Alessandro De Luca and Wayne J Book. Robots with flexible elements. In *Springer Handbook of Robotics*, pages 243–282. Springer, 2016.

[4] 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.

[5] Russ Tedrake. Underactuated robotics: Learning, planning, and control for efficient and agile machines course notes for mit 6.832. *Working draft edition*, 3, 2009.

[6] 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.

[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] 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.

[9] J Lin and FL Lewis. Improved measurement/estimation technique for flexible link robot arm control. In *Proceedings of 32nd IEEE Conference on Decision and Control*, pages 627–632. IEEE, 1993.

[10] Alberto Isidori. *Nonlinear Control Systems Design 1989: Selected Papers from the IFAC Symposium, Capri, Italy, 14-16 June 1989*. Elsevier, 2014.

[11] Franco Angelini, Riccardo Mengacci, Cosimo Della Santina, Manuel G Catalano, Manolo Garabini, Antonio Bicchi, and Giorgio Grioli. Time generalization of trajectories learned on articulated soft robots. *IEEE Robotics and Automation Letters*, 5(2):3493–3500, 2020.

[12] Riccardo Mengacci, Franco Angelini, Manuel G Catalano, Giorgio Grioli, Antonio Bicchi, and Manolo 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*, page 0278364920943275, 2020.

[13] Franco Angelini, Cosimo Della Santina, Manolo Garabini, Matteo Bianchi, and Antonio Bicchi. Control architecture for human-like motion with applications to articulated soft robots.

[14] Michele Pierallini, Franco Angelini, Riccardo Mengacci, Alessandro Palleschi, Antonio Bicchi, and Manolo Garabini. Trajectory tracking of a one-link flexible arm via iterative learning control.

[15] Sadao Kawamura, Norihisa Fukao, and Hiroaki Ichii. Planning and control of robot motion based on time-scale transformation and iterative learning control. In *Robotics Research*, pages 213–220. Springer, 2000.