Iterative Learning Control for Fast and Accurate Position Tracking with an Articulated Soft Robotic Arm

Matthias Hofer, Lukas Spannagl and Raffaello D'Andrea

***Abstract*— This paper presents the application of an iterative learning control scheme to improve the position tracking performance for an articulated soft robotic arm during aggressive maneuvers. Two antagonistically arranged, inflatable bellows actuate the robotic arm and provide high compliance while enabling fast actuation. Switching valves are used for pressure control of the soft actuators. A norm-optimal iterative learning control scheme based on a linear model of the system is presented and applied in parallel with a feedback controller. The learning scheme is experimentally evaluated on an aggressive trajectory involving set point shifts of 60 degrees within 0.2 seconds. The effectiveness of the learning approach is demonstrated by a reduction of the root-mean-square tracking error from 13 degrees to less than 2 degrees after applying the learning scheme for less than 30 iterations.**

I. Introduction

Soft, inflatable robotic manipulators exhibit a number of promising properties. High compliance and low inertia combined with pneumatic actuation enable fast, but nevertheless safe applications ([1], [2], [3] and [4]). However, accurate position control is challenging with soft manipulators, because they typically have a high number of potentially coupled and uncontrollable degrees of freedom. Furthermore, the dynamics of soft materials exhibit viscoelastic material behavior, which is difficult to model from first principles.

One way to improve the tracking performance of soft, inflatable manipulators is to combine soft structures with rigid components. These hybrid designs typically have lower overall compliance and higher inertia compared to their entirely soft counterparts, but also a reduced number of degrees of freedom [5]. Furthermore, the degrees of freedom are limited to specific joints for articulated soft robots. Thereby, the control authority of the remaining degrees of freedom is generally higher, leading to an improved tracking performance. Examples for such designs can be found in [6] and [7].

Alternatively, advanced control approaches can be used to improve the tracking performance of inflatable robotic manipulators. Learning-based open loop control, which purely relies on mechanical feedback, is demonstrated in [8] for a pneumatically-actuated soft manipulator. The authors of [9] achieve a reduction in the tracking error by using model predictive control for a fabric-based soft arm. The control approach is extended in [10] to a nonlinear model predictive controller based on a neural network to describe the dynamics. Reinforcement learning is applied in [11] to simultaneously optimize the accuracy of the controllable stiffness and the position tracking of a manipulator intended for assistive applications. A learned inverse kinematics model is used in [12] to improve the position tracking accuracy with a soft handling assistant. The authors of [13] use iterative learning control (ILC) in combination with low-gain feedback control to improve tracking performance, while preserving the intrinsic compliance of soft robots. A method based on ILC to learn grabbing tasks for a soft fluidic elastomer manipulator is reported in [14].

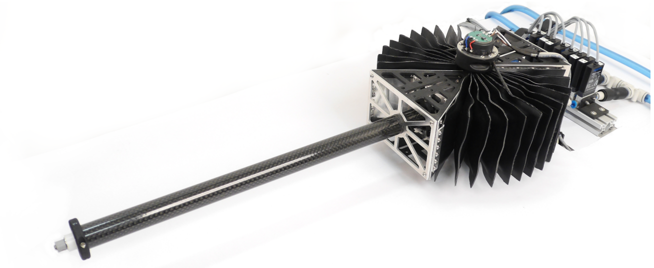

Fig. 1. The articulated soft robotic arm used for the experimental evaluation. It consists of two antagonistically arranged soft bellow actuators and a rigid backbone. The inflatable actuators are made from coated fabric and attached to a lightweight structure built from a combination of 3D printed parts and aluminum plates. The low inertia of the one degree of freedom arm enables fast maneuvers.

In this paper, we propose an ILC scheme to improve the tracking performance for an articulated soft robotic arm (see Fig. 1). Instead of the commonly used proportional valves (e.g. [7], [10]), we deploy switching valves to control the pressure of the actuators (see also [15]). The hardware limitations imposed by this type of valve are addressed by a norm-optimal ILC approach, which is applied in parallel with a feedback controller. The non-causal learning controller is evaluated on repetitive, aggressive angle trajectories as arising in, for example, pick and place applications.

The remainder of this paper is organized as follows: The design of the robotic arm is discussed in Section II along with the derivation of a simple model and a feedback controller. The iterative learning control approach is outlined in Section III and experimental results are presented in Section IV. Finally, a conclusion is drawn in Section V.

II. Articulated Soft Robotic Arm

The articulated soft robotic arm used for the experimental evaluation is presented in the first part of this section. It serves as a testbed for the study of control algorithms for this type of system. While the number of degrees of freedom is limited to one, it nevertheless exhibits the complex dynamics inherent in this type of system, e.g. the nonlinear pressure

The authors are members of the Institute for Dynamic Systems and Control, ETH Zürich, Switzerland. Email correspondence to Matthias Hofer hofermat@ethz.ch.

dynamics and viscoelastic material behavior. A simple model of the soft actuator and the robotic arm is derived in the second part and a cascaded control architecture is discussed in the last part of this section.

A. Design

The hybrid robotic arm consists of two soft bellow actuators and a rigid joint structure. The soft actuator is discussed in the first part and the complete robotic arm in the second part of this section. The design of the actuators and the robotic arm is inspired by the one presented in [15]. However, several properties of the actuators and the robotic arm are optimized.

A soft bellow actuator (Fig. 2) is made from thermoplastic polyurethane coated nylon (Rivertex® Riverseal® 842). It consists of twelve single cushions, which are connected by an internal seam. Placing the inner seam off-center leads to an expansion in the angular direction upon pressurization. The fabric pieces are prepared with a cutting plotter and processed with high frequency welding. Thereby, the fabric sheets are clamped between two electrodes and a high frequency alternating electromagnetic field is applied. The material coating is heated over its full seam thickness, eventually causing the fabric sheets to bond together (see [16] for more details). Three high frequency weldable polyvinyl chloride tubing flanges are welded to each bellow. Tubing is glued to the flanges, which connect two tubes with the pressure regulating valves and the third one with the pressure sensor. Instead of the commonly employed proportional solenoid valves, we use switching solenoid valves. While proportional valves can continuously adjust the nozzle, switching valves only have a binary state, namely fully open or fully closed. Continuous adjustment of the mass flow (and consequently of the pressure) is achieved by pulse width modulation (PWM). Four 2/2-way switching valves (Festo MHJ10) are used per actuator to control the pressure. Two valves are used as inlet valves, placed between the source pressure and the actuator, and two as outlet valves, placed between the actuator and the environment. The pressure is measured in each actuator and at the source using Bürkert 8230 pressure transducers. The temperature is measured at the pressure supply using a temperature sensor (Texas Instruments LM35).

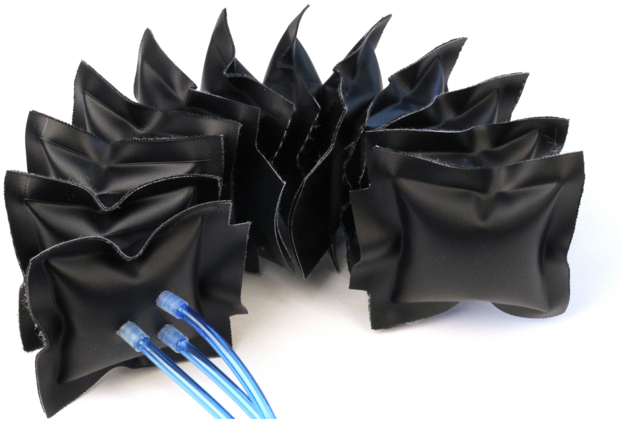

Fig. 2. The bellow-type actuator made from polyurethane coated nylon in fully inflated state. The actuator consists of twelve single cushions and the front side measures $100 \times 100\,\mathrm{mm}$ in a deflated state. When fully inflated, the maximum actuation range, is approximately $190\,^\circ$. Two tubes connect the actuator to the control valves, while the third one is used to measure the pressure in the actuator.

Note that the four deployed switching valves cost 2.5 times less than a proportional valve from the same manufacturer (Festo MPYE), which is commonly used in soft robotics applications (e.g. [7], [17]). However, the advantage in price comes at the cost of a reduced flow capacity, which can be limiting for fast maneuvers. A control approach to address this limitation is presented in Section III.

With respect to the prototype presented in [15], the actuation range is increased by using twelve instead of six single cushions. Furthermore, the pressure is directly measured in the actuator and not in the connecting tubes. Thereby, the actuator volume filters the effect of the pressure ripples caused by the switching valves and hence reduces valve jitter. Note that doubling the volume of the actuator requires the use of twice as many valves and connecting tubes in order to retain a comparable time constant of the pressure dynamics with respect to the smaller actuator.

The hybrid robotic arm consists of two bellow-type actuators, which are arranged antagonistically between a rigid support structure consisting of two prisms (see Fig. 3). A prism consists of two 3D printed triangles (polyamide PA12), which are connected by aluminum plates. The first and last cushion of each actuator are attached to the aluminum plates of either the static or the moving prism. The two prisms are connected by revolute joints, which are integrated into the triangles. The angle of the arm is measured by a rotary encoder (Novotechnik P4500) providing angle data with an accuracy to one-tenth of a degree. Finally, a carbon fiber rod is attached to the moving prism to form the arm.

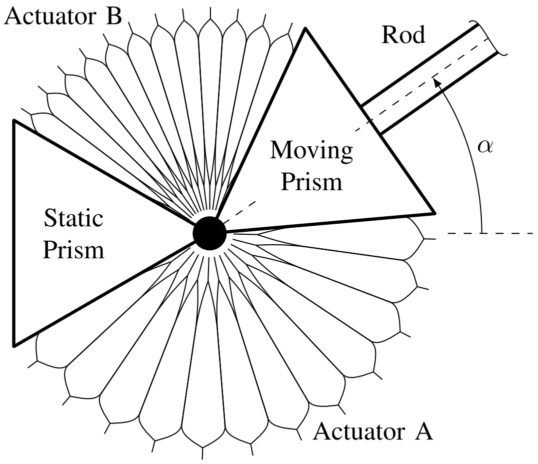

Fig. 3. Schematic drawing of the articulated soft robotic arm including two actuators, namely A and B, the static and moving prisms, the rod and the revolute joint indicated by the black circle. In the configuration shown, the pressure in actuator A is higher than in actuator B, leading to a deflection in the positive α-direction.

A pressure difference between actuators A and B leads to a different expansion of the two actuators and consequently to a torque on the moving prism. In order to maximize the expansion of each actuator in the angular direction and hence minimize the expansion in the radial direction, a strap is attached around both actuators.

As a consequence of the increased angle range compared to the design presented in [15], the spring-like retraction forces of the deflated actuators are reduced. Therefore, the required actuation pressure for a given angle is reduced as well. The combination of 3D printed plastic parts with aluminum plates considerably reduces the inertia of the robotic arm compared to the previous design. The hybrid design presented combines the benefits of soft actuators, namely high compliance and fast actuation, with the advantages of classic robotic arms, namely that the expansion of the soft actuators is concentrated in a single degree of freedom.

B. Modeling

The modeling of the robotic arm is described in this section. It is important to note that, for all models discussed here, the focus lies on simplicity rather than accuracy. The motivation is to capture the principle dynamics and compensate for inaccuracies and model uncertainties by a learning-based control approach.

The model of the robotic arm can be divided into two subsystems. The pneumatic system including the soft actuator and the valves is described first and the rigid body dynamics of the robotic arm are discussed in a second part.

The pressure dynamics of the actuator can be derived from first principles. Taking the time derivative of the ideal gas law relates the change in pressure to the mass flow in or out of the actuator and the change in volume and temperature. We assume isothermal conditions and neglect changes in the volume, which leads to the following expression for the pressure dynamics of an actuator,

$$\dot{p} = \dot{m}\frac{RT}{V} + \dot{T}\frac{mR}{V} - \dot{V}\frac{p}{V} \approx \dot{m}\frac{RT}{V}, \tag{1}$$

where R denotes the ideal gas constant, m the air mass in the actuator, $\dot{m}$ the mass flow in or out of the actuator, T, $\dot{T}$ the temperature and its derivative and V the volume and $\dot{V}$ its derivative. Given the isochoric assumption, the volume of the actuator is approximated as constant, hence independent of the angle of the robotic arm. The average volume is measured to be $0.45\,\mathrm{L}$. Note that the model presented here neglects interactions with the antagonistic actuator and the elasticity of the fabric material.

More sophisticated models of the pressure dynamics including an identification of the angle dependent volume relation were also investigated. However, it only slightly improved the pressure control performance, but required additional identification experiments. Therefore, we only use a coarse approximation of the pressure dynamics in this work and rely on the learning approach to compensate for the approximation.

A static model of the valves employed is derived in [15], which is also used in this work. The experimentally identified model relates the mass flow of one actuator to the up and downstream pressures of its valves and to the applied duty cycles,

$$\dot{m} = f_{\text{valves}}(p_{\text{u}}, p_{\text{d}}, dc_{\text{in}}, dc_{\text{out}}), \tag{2}$$

where p_{u} and p_{d} denote up and downstream pressure and dc_{in}, dc_{out} the duty cycles of the inlet and outlet valves. For the inlet valves, the upstream pressure corresponds to source pressure and the downstream pressure to the controlled pressure in the actuator. For the outlet valves, upstream pressure is related to the pressure in the actuator and downstream pressure to ambient pressure.

The rigid body dynamics of the robotic arm are assumed to be driven by the pressure difference between the two actuators, defined as $\Delta p = p_{\text{A}} - p_{\text{B}}$. A positive pressure difference, Δp, accelerates the arm in the positive α-direction (see Fig. 3). The dynamics of the robotic arm with Δp as input and the arm angle α as output, are identified using system identification. The same identification procedure as in [15] is applied. A continuous-time second-order model is assumed for the dynamics of the robotic arm. More complex models including the actuator pressures as states and the set point pressure difference as an input are not considered for the sake of simplicity and will be compensated by the learning approach. The resulting transfer function is,

$$G(s) = \frac{\alpha(s)}{\Delta p(s)} = \kappa\frac{\omega_0^2}{\omega_0^2 + 2\delta\omega_0 s + s^2}. \tag{3}$$

The complex variable is denoted by s and the numeric values of the parameters are $\kappa = 7.91\,\mathrm{rad/bar}$, $\omega_0 = 14.14\,1/\mathrm{s}$ and $\delta = 0.31$. The design improvements discussed in the previous subsection are reflected in the parameter values identified. Compared to the model presented in [15], the gain of the transfer function, κ, is more than four times larger due to the reduced spring-like retraction forces. The continuous time model is discretized with a sampling time of $1/50\,\mathrm{s}$ leading to the following linear-time-invariant system,

$$\begin{aligned}\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix} &= \underbrace{\begin{bmatrix}0.96 & 0.18\\-0.36 & 0.80\end{bmatrix}}_{:=A}\begin{bmatrix}x_1(k{-}1)\\x_2(k{-}1)\end{bmatrix} + \underbrace{\begin{bmatrix}0.09\\0.91\end{bmatrix}}_{:=B}u(k{-}1)\\ y(k) &= \underbrace{\begin{bmatrix}1 & 0\end{bmatrix}}_{:=C}\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix},\end{aligned} \tag{4}$$

where k denotes the time index, (x_1, x_2) corresponds to the state $(\alpha, \dot{\alpha})$ normalized by $(\pi, 10\pi\,1/\mathrm{s})$ and u to the control input Δp (in Pa) normalized by $1\mathrm{e}5\,\mathrm{Pa}$. The arm deflection α is directly measured by the rotary encoder. This model will be used for the iterative learning control discussed in Sec. III.

C. Control

A cascaded control architecture similar to [15] is applied based on time scale separation of the faster pressure dynamics in an inner loop and the slower arm dynamics in an outer loop (see Fig. 4). A proportional-integral-derivative (PID) controller is used in the outer loop to compute a required pressure difference, u_{PID}, to reach a desired arm angle. The PID controller includes a feed forward term to improve tracking performance and has the following form,

$$u_{\text{PID}} = k_{\text{ff}} y_{\text{D}} + k_{\text{p}}(y_{\text{D}} - y^j) + k_{\text{i}} \int (y_{\text{D}} - y^j)dt + k_{\text{d}}\dot{y}^j, \quad (5)$$

where the time index is omitted for the ease of notation. The desired angle is denoted by y_{D} and the measured angle by y^j. The set point for the pressure difference is then translated to the individual pressure set points for the inner control loops. The underlying assumption is that set point changes of the pressure can be tracked instantaneously by the inner control loop. From the control input determined by (5), the set point pressures are computed as,

$$\begin{aligned} p_{\text{A,des}} &= \min(p_{\text{max}}, \max(p_0, p_0 + u_{\text{PID}})) \\ p_{\text{B,des}} &= \min(p_{\text{max}}, \max(p_0, p_0 - u_{\text{PID}})), \end{aligned} \quad (6)$$

where p_0 is the ambient air pressure and p_{max} the maximum allowed pressure.

Next, we present the pressure controller based on the model presented in the previous subsection. We describe the control algorithm for actuator A, but it is analogously implemented for actuator B. In order to smooth the effects of the PWM, a moving average filter is used to filter the measured pressures and hence reduce valve jitter. We assume a first order model for the pressure dynamics,

$$\dot{p}_{\text{A}} = \frac{1}{\tau_{\text{p}}}(p_{\text{A,des}} - p_{\text{A}}), \quad (7)$$

where τ_{p} is the time constant of the desired pressure dynamics and is used as a tuning parameter for the controller. We combine this expression with (1) to obtain the desired mass flow $\dot{m}$ as a function of the current pressure deviation,

$$\dot{m}_{\text{A}} = \frac{V}{RT\tau_{\text{p}}}(p_{\text{A,des}} - p_{\text{A}}). \quad (8)$$

The valve model (2) is used to compute the required duty cycles for both the in and outlet valves from the desired mass flow and the known up and downstream pressures of the valves. A detailed explanation of this procedure can be found in [15]. Note that the number of valves is doubled compared to [15], but the same duty cycles are applied to both in and both outlet valves of actuator A and B.

III. Iterative Learning Control

In this section, we present an ILC scheme to improve the angle tracking performance with the robotic arm. ILC is a method to improve the tracking performance for repetitive tasks ([18]). The tracking error from the previous iteration is used to compensate for the disturbances in the current iteration.

We apply a norm-optimal iterative learning control (NOILC) scheme (see [19]), which is based on the linear model derived in Section II and a quadratic cost function. Since the NOILC approach can only account for repetitive disturbances, it is applied in parallel with the previously presented feedback controller, which compensates for non-repetitive disturbances (as discussed in [18]). The resulting control architecture is depicted in Fig. 4, where the superscript index j is used to denote an iteration of the learning scheme. The tracking error in iteration j is used to compute a correction signal, which is added to the feedback controller in the next iteration. The lifted-system framework (see [18]) is used to derive the NOILC algorithm. The following variables are introduced,

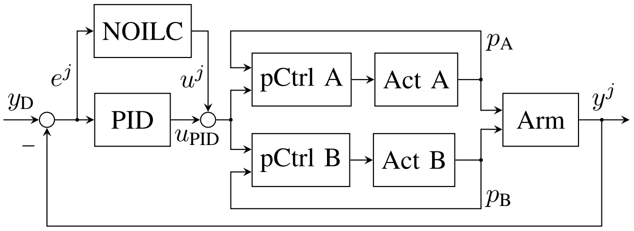

Fig. 4. Control architecture with the NOILC scheme in parallel configuration with the PID feedback controller in the outer control loop. The feed forward signal of the PID controller is not depicted for the sake of clarity. The input to the NOILC is the normalized error between the desired, y_{D}, and measured angle, y^j. The current value of the correction signal u^j is added to the control input of the feedback controller, u_{PID}, and forms the input to the inner control loops. The pressure controllers (pCtrl A, pCtrl B) adjust the pressures in each actuator (Act A, Act B). These pressures are the inputs to the system (Arm).

$$\begin{aligned} y_{\text{D}} &:= \big(y_{\text{D}}(1), \dots, y_{\text{D}}(N)\big) & d &:= \big(d(1), \dots, d(N)\big) \\ y^j &:= \big(y^j(1), \dots, y^j(N)\big) & u^j &:= \big(u^j(0), \dots, u^j(N{-}1)\big) \\ e^j &:= \big(e^j(1), \dots, e^j(N)\big) & &\text{all} \in \mathbb{R}^N, \end{aligned} \quad (9)$$

where y_{D} is the desired output and y^j the measured output in iteration j. The normalized error in iteration j is denoted by e^j, the repetitive disturbance by d and the correction signal applied in iteration j by u^j. The desired output y_{D} is the same for all iterations and the disturbance d is assumed to be repetitive between successive iterations. Note that four variables in (9) are shifted by one time step to account for the one step time delay induced by the plant. The dimension of the lifted-system, that is the number of time steps, N, is defined by the time duration of the desired trajectory and the sampling time of the NOILC approach, T_{ILC}.

Defining the lifted state matrix,

$$P = \begin{bmatrix} CB & 0 & 0 & 0 \\ \vdots & & CB & 0 \\ CA^{N-1}B & \cdots & CAB & CB \end{bmatrix} \in \mathbb{R}^{N \times N}, \quad (10)$$

allows us to express the dynamics in the lifted system framework, namely,

$$\begin{aligned} y^j &= Pu^j + d \\ e^j &= y_{\text{D}} - y^j = y_{\text{D}} - Pu^j - d \\ e^{j+1} &= y_{\text{D}} - Pu^{j+1} - d \\ &= e^j - P(u^{j+1} - u^j), \end{aligned} \quad (11)$$

where we assume a zero initial condition for the state. The principle idea in NOILC is to obtain the correction input by minimizing a quadratic cost function of the predicted tracking error of the next iteration. The linear model in (11) is used to predict the tracking error one iteration ahead. Additional terms can be added to the cost function to improve the transient learning behavior. The following objective function is used in this work,

$$J(u^{j+1}) = \frac{1}{2}[e^{j+1^{\mathsf{T}}} M e^{j+1} + (u^{j+1}-u^{j})^{\mathsf{T}} S (u^{j+1}-u^{j}) + u^{j+1^{\mathsf{T}}} D^{\mathsf{T}} W D u^{j+1}], \qquad (12)$$

where M, S, W are positive semi-definite cost matrices of appropriate dimensions and

$$D = \frac{1}{T_{\text{ILC}}} \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{N \times N} \qquad (13)$$

approximates the derivative of the correction input, by the first order Euler forward numerical differentiation. The first term in (12) penalizes the predicted tracking error in the next iteration and can be replaced by the last expression in (11). The second term penalizes a change in the correction input between successive iterations and the last term the derivative of the correction input. These last two terms are included to ensure a stable learning behavior by avoiding an overcompensation of the disturbance. Including a cost on the absolute change in correction input ensures that the NOILC approach does not excessively compensate for the previous tracking error before the computed correction input can show effect. Penalizing the derivative of the correction input limits the high frequency content of the correction input, which is handled in traditional ILC approaches by the so-called Q filters (see [18]). Constraints on the correction input are neglected in the minimization of (12), such that the optimal solution can be computed in closed form. The feasibility of the solution is imposed by limiting the resulting pressure set points in (6) to the feasible interval $[p_0, p_{\text{max}}]$ at a later stage. The optimal solution is therefore given by,

$$\begin{aligned} u^{j+1^{\star}} &= \underset{u^{j+1}}{\operatorname{argmin}}\, J(u^{j+1}) \\ &= (P^{\mathsf{T}} M P + S + D^{\mathsf{T}} W D)^{-1} (P^{\mathsf{T}} M P + S) u^{j} \\ &\quad + (P^{\mathsf{T}} M P + S + D^{\mathsf{T}} W D)^{-1} P^{\mathsf{T}} M e^{j}. \end{aligned} \qquad (14)$$

Note that the inverse in (14) exists if the cost matrix S has full rank. The correction input of the NOILC approach is added to u_{PID} as given in (5) for each time step,

$$u^{j}_{\text{tot}}(k) = u_{\text{PID}}(k) + u^{j}(k), \quad k = 1, \ldots, N, \qquad (15)$$

and consequently u^{j}_{tot} is used in (6) instead of u_{PID}.

Simpler ILC approaches (e.g. model-free), such as PD-type ILC (see [18]), were also investigated. However, contrary to the discussed NOILC approach they converged for very aggressive trajectories (compare Sec. IV) only if the high-frequency content of the correction signal was suppressed significantly by tuning a Q filter accordingly. However, the Q filter clearly limited the possible improvement of the tracking performance.

The previously introduced approach requires to store a correction signal and we finish this section by investigating how much that costs. Assuming a price of 0.03 USD per gigabyte of permanent hard disk drive memory as of 2018 [20], a precision of two bytes to store the final correction signal, and a sampling time T_{ILC} as stated in the next section, it costs 1 USD to store 10.5 years of correction signal.

IV. Experimental Results

In this section, we present the results from experimental evaluations of the proposed NOILC scheme on the articulated soft robotic arm.

The control algorithms are executed on a laptop computer (Intel Core i7 CPU, 2.8 GHz) and the valves and sensors are interfaced over a Labjack T7 Pro device. The pressure controllers are executed at 200 Hz, the PID feedback controller runs at 50 Hz and the NOILC scheme has a sampling time of $T_{\text{ILC}} = 1/50\,\text{s}$. The source pressure is set to 3 bar, the maximum pressure constraint to $p_{\text{max}} = 1.4\,\text{bar}$, the time constant of the pressure controllers to $\tau_{\text{p}} = 1/50\,\text{s}$ and the PWM frequency is set to 200 Hz. The cost matrices of the NOILC scheme are $M = I_N$, $S = 0.1 \cdot I_N$ and $W = 2\text{e}{-5} \cdot I_N$ with $I_N \in \mathbb{R}^{N \times N}$ being the identity matrix.

The reference trajectory to evaluate the learning scheme has a duration of 8 s (resulting in $N = 400$) and includes several set point jumps of 60 ° within 0.2 s and peak angular accelerations of $12000\,°/\text{s}^2$. It is computed from trapezoidal angular velocity profiles, which limit the maximum angular acceleration and correspondingly the excited frequency band. The reader is referred to the video attachment to gain an impression of the experiments conducted.

The results of the angle tracking experiment are shown in Fig. 5. The tracking performance is improved significantly by applying the NOILC scheme, with a reduced root-mean-square (RMS) error from 13 ° to less than 2 ° after 30 iterations as can be seen in Fig. 6. The non-causality of the NOILC scheme allows the elimination of the delay between the angle response and the set point, which occurs when no learning control is used.

When the feedback controller is used without any learning, the robotic arm bounces back slightly after an initial steep response when commanding an angle jump of 60 °. The reason for this is that it compresses the deflating actuator, causing the pressure in this actuator to temporarily increase and hence the arm to bounce back. Note that if proportional valves were deployed, the temporary pressure increase in the compressed actuator would be less pronounced because of their higher flow capacity. Considering the full pressure dynamics on the other hand (i.e. not simplifying (1)) would not avoid the pressure build-up, because the valves of the deflating actuator are already fully open. The NOILC scheme compensates for this effect by increasing the pressure difference for a short period, which can be seen by the peaks in the total pressure difference applied in iteration 30 (red curve) occurring at $t = \{1.6, 2.6, 3.6, 4.6\,\text{s}\}$.

Note that if the same control input is applied, the disturbances are repeatable over different iterations of the learning scheme since they arise from unmodeled dynamics. Over a single iteration, the behavior is also repeatable, as can be seen for example by the similar angular response when commanding a set point jump from $-30\,°$ to $30\,°$ occurring at $t = \{2.6, 4.6\,\text{s}\}$.

A comparison between the commanded pressure differences reveals that the feedback controller without learning has to rely on integral action to reach the set point, while

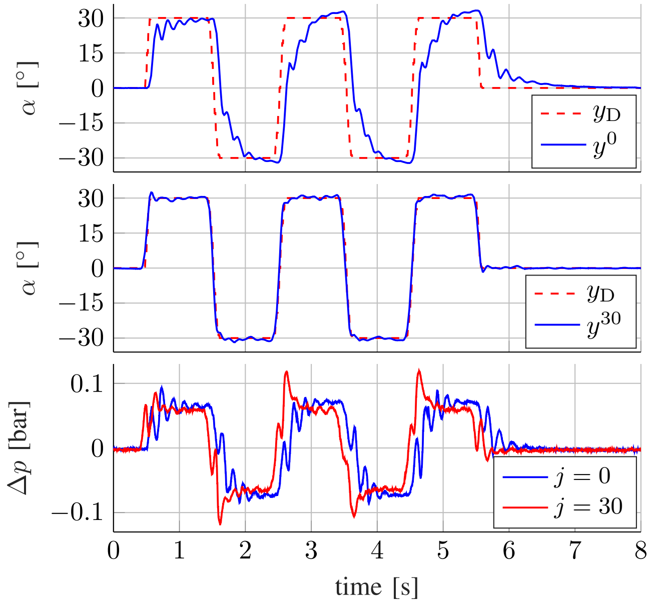

Fig. 5. Experimental results of the robotic arm tracking an angular set point trajectory. The top plot shows the tracking performance when the feedback controller is used only (no learning, iteration 0). In this case, the angle initially follows the set point with a steep response, then slightly bounces back and finally reaches the set point. The middle plot shows the improved tracking performance after applying the NOILC scheme for 30 iterations. The bottom plot depicts the total pressure difference applied in iteration 0 (blue) and iteration 30 (red) and reveals the importance of the non-casual nature of the NOILC scheme, resulting in a shifted pressure difference input anticipating the repetitive disturbances. For the entire trajectory, the total pressure difference is never close to its maximum pressure constraint.

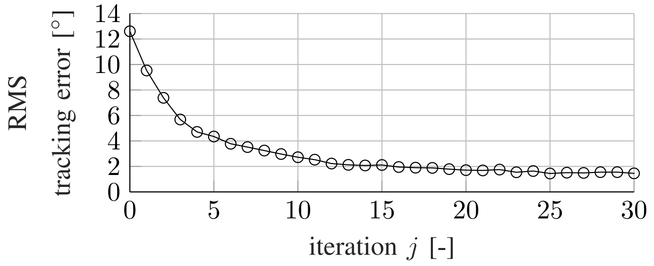

Fig. 6. The RMS tracking error plotted over 30 iterations of applying the NOILC scheme. The error decreases from 13 ° to less than 2 ° following a monotonic trend and remains constant thereafter. I.e. no early stopping is required. Deploying the learning approach for only 10 iterations already accounts for 90 % of the improvement achieved after 30 iterations.

the NOILC directly adjusts the pressure difference required for a certain angular set point.

V. Conclusion

A norm-optimal ILC scheme has been presented to improve the position tracking performance with an articulated soft robotic arm. The non-causal learning scheme is based on a simple model of the dynamics, which guides the learning and brings advantages compared to simpler ILC approaches, e.g. PD-type ILC. As opposed to a model-based feedback controller, the tracking performance of the learning scheme is not limited by the model accuracy. Results have shown that performance can be improved significantly for an aggressive trajectory. Learning a correction signal for different reference trajectories is a practical alternative to both more expensive valves and more advanced causal control approaches. Future work includes the design of a fully inflatable system and will investigate how learning control generalizes to systems with multiple degrees of freedom.

Acknowledgment

The authors would like to thank Michael Egli, Daniel Wagner and Marc-Andrè Corzillius for their contribution to the development of the prototype.

References

[1] B. Gorissen, D. Reynaerts, S. Konishi, K. Yoshida, J. Kim, and M. De Volder, “Elastic inflatable actuators for soft robotic applications,” *Advanced Materials*, vol. 29, no. 43, 2017.

[2] C. M. Best, J. P. Wilson, and M. D. Killpack, “Control of a pneumatically actuated, fully inflatable, fabric-based, humanoid robot,” in *IEEE International Conference on Humanoid Robots (Humanoids)*, 2015.

[3] S. Sanan, M. H. Ornstein, and C. G. Atkeson, “Physical human interaction for an inflatable manipulator,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2011.

[4] B. Mosadegh, P. Polygerinos, C. Keplinger, S. Wennstedt, R. Shepherd, U. Gupta, J. Shim, K. Bertoldi, C. Walsh, and G. Whitesides, “Pneumatic networks for soft robotics that actuate rapidly,” *Advanced Functional Materials*, vol. 24, no. 15, pp. 2163–2170, 2014.

[5] R. F. Natividad, M. R. D. Rosario, P. C. Y. Chen, and C. H. Yeow, “A hybrid plastic-fabric soft bending actuator with reconfigurable bending profiles,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[6] H. Kim, A. Kawamura, Y. Nishioka, and S. Kawamura, “Mechanical design and control of inflatable robotic arms for high positioning accuracy,” *Advanced Robotics*, vol. 32, no. 2, pp. 89–104, 2018.

[7] M. Jordan, D. Pietrusky, M. Mihajlov, and O. Ivlev, “Precise position and trajectory control of pneumatic soft-actuators for assistance robots and motion therapy devices,” in *IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2009.

[8] T. G. Thuruthel, E. Falotico, M. Manti, and C. Laschi, “Stable open loop control of soft robotic manipulators,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1292–1298, 2018.

[9] M. T. Gillespie, C. M. Best, and M. D. Killpack, “Simultaneous position and stiffness control for an inflatable soft robot,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[10] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, “Learning nonlinear dynamic models of soft robots for model predictive control with neural networks,” in *IEEE International Conference on Soft Robotics (RoboSoft)*, 2018.

[11] Y. Ansari, M. Manti, E. Falotico, M. Cianchetti, and C. Laschi, “Multiobjective optimization for stiffness and position control in a soft robot arm module,” *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 108–115, 2018.

[12] R. F. Reinhart and J. J. Steil, “Hybrid mechanical and data-driven modeling improves inverse kinematic control of a soft robot,” *Procedia Technology*, vol. 26, pp. 12 – 19, 2016.

[13] F. Angelini, C. D. Santina, M. Garabini, M. Bianchi, G. M. Gasparri, G. Grioli, M. G. Catalano, and A. Bicchi, “Decentralized trajectory tracking control for soft robots interacting with the environment,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 924–935, 2018.

[14] A. D. Marchese, R. Tedrake, and D. Rus, “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.

[15] M. Hofer and R. D’Andrea, “Design, modeling and control of a soft robotic arm,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[16] M. J. Troughton, *Handbook of Plastics Joining, Second Edition: A Practical Guide (Plastics Design Library)*. William Andrew, 2008.

[17] D. Büchler, H. Ott, and J. Peters, “A lightweight robotic arm with pneumatic muscles for robot learning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[18] D. A. Bristow, M. Tharayil, and A. G. Alleyne, “A survey of iterative learning control,” *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, 2006.

[19] D. Owens, *Iterative Learning Control : An Optimization Paradigm*. Springer Publishing Company, 2016.

[20] hblok. (2019) Historical cost of computer memory and storage. [Online]. Available: https://hblok.net/blog/storage/