Iterative Learning Control
of a Pneumatically Driven Robot Joint

1st Rainer Nitsche
Robotics System Design
Festo SE & Co. KG
Esslingen, Germany
rainer.nitsche@festo.com

2nd Timo Heubach
Robotics System Design
Festo SE & Co. KG
Esslingen, Germany
timo.heubach@festo.com

Abstract—**The unique physical characteristics of pneumatic drives facilitate the creation of robots that are safe lightweight and intuitive to operate. However, pneumatic robots are due to nonlinear behaviour that is hard to control. Especially friction plays a major role and limits the control performance of tracking a desired trajectory. In this contribution, the concept of *Iterative Learning Control (ILC)* is introduced and applied to a pneumatically driven joint build-in a pneumatic cobot. Iterative Learning Control is highly suitable for repeatable control tasks as they appear typically in robot applications [10]. With this approach, it is possible almost perfectly to track a given, repeatable trajectory.**

Index Terms—**model-based control, robot control, flatness, iterative learning control, pneumatic robotics**

I. Introduction

Robot control tasks often follows a repeating trajectory. For this kind of application the concept of learning from the error of the previous control run is obvious. That's why in the field of robot control the concept of ILC is often used [1].

The concept of Iterative Learning Control is based on the idea that repeated runs of a continuous task can be used to gradually improve the performance of the control step by step.

The aim of ILC is to iteratively improve the control quality of such systems through learning.

To achieve this, the error and the control signal of the previous cycle are analyzed and used to adapt the new control signal. Due to the fact that the new control signal is already calculated before each cycle, it is possible to apply non-causal algorithms and filters. This is a big advantage compared to a classical feedback controller. The ILC approach can be considered as a kind of feedforward control, since the control signal only depends of the control error of the previous cycle.

The ILC approach belongs to the class of feedforward controllers and can be extended in the sense of a a two-degree-of-freedom structure by adding a feedback component. A classic PID controller can be used here, for example. Through a combination of feedforward and feedback control, the control control peformance and disturbance rejection can be improved significantly. This is achieved, as the ILC controller reduces the periodic errors and the feedback controller compensates the disturbances [3].

For the pneumatic robot joint under consideration, a feedback controller based on flatness in the sense of [6] is used to have a model-based tracking controller of high performance. In a second step an ILC approach is introduced as an additional controller to almost perfectly track a desired repetitive trajectory. This two-degree-of-freedom concept will be applied to a pneumatic robot joint and tested in experiments. The achieved performance is outstanding.

II. Control Task

For pneumatic robots so far, only fluidic muscles [16], [4] or bellows for soft robots [5] have been used. However, pneumatic robots with rigid links and flexible rotary joints, reducing model complexity and kinematic uncertainty, are less commonly found in the literature. This approach facilitates the development of real-time feasible model-based controllers and has the potential to achieve higher precision. First of all, the robot must be able to track trajectories, which is usually a repetitive task. For pneumatic manipulators with coupled joints, this control task is only reported to be realizable with less accuracy [2] [11] [15].

Iterative learning control approaches for robots are highly suitable [12] but not yet reported for pneumatically driven robots, but for other industrial applications, e. g. [14].

Fig. 1. Pneumatic Cobot from Festo (2020).

Model of the Pneumatic Robot Joint

In this research a single joint (cf. Fig. 2) of a pneumatic robot (cf. Fig. 1) is considered. In this pneumatically driven revolute joint, the two pressure chambers are separated by a swivel

wing of area A_{eff} and an effective radius r_{eff}, which is semi-rotatable to $\pm 135°$. The pressure difference between the two chambers 1 and 2 of the joint generates a driving torque

$$\tau(p_1, p_2) = A_{\text{eff}} \, r_{\text{eff}} \, (p_1 - p_2). \tag{1}$$

The pressure in each chamber is set individually by a massflow $\dot{m}_{1,2}$ into or out of the chamber with a pneumatic valve. A more detailed overview of the pneumatic model can be found in [8].

The governing equations of the rotary joint model are:

$$\ddot{\varphi} = \frac{A_{\text{eff}} r_{\text{eff}} (p_1 - p_2) - \tau_{\text{f}}(\dot{\varphi})}{J}, \tag{2a}$$

$$\dot{p}_1 = \frac{n(RT\dot{m}_1 - p_1 A_{\text{eff}} r_{\text{eff}} \dot{\varphi})}{A_{\text{eff}} r_{\text{eff}} \, (\, |\varphi_{\max} + \varphi|)}, \tag{2b}$$

$$\dot{p}_2 = \frac{n(RT\dot{m}_2 + p_2 A_{\text{eff}} r_{\text{eff}} \dot{\varphi})}{A_{\text{eff}} r_{\text{eff}} \, (\, |\varphi_{\max} - \varphi|)}, \tag{2c}$$

where τ_{f} is the friction torque (cf. Figure 2), n is the polytropic exponent, R the specific gas constant, and T the temperature. The valves are installed next to the drive, so the tubes are short and the mass flow to and from the pressure chamber are considered to be equal to the mass flow set by the valves.
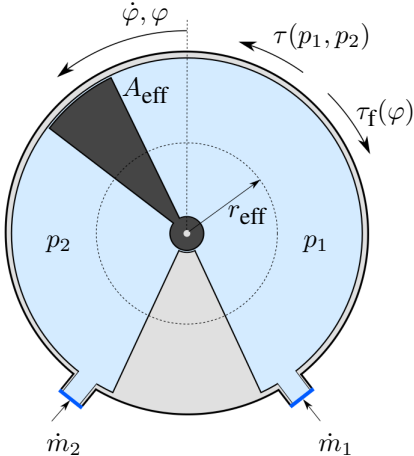

Fig. 2. Schematics of a pneumatic rotary joint.

Remark 2.1: In the following, the friction will be neglected for controller design, since one of the tasks of the ILC is to compensate the friction, which plays a crucial role in controlling the position of the pneumatic joint with high accuracy. ⋄

III. CONTROLLER DESIGN

In a first step of the controller design a subordinary pressure controller based on flatness is introduced. To control the position φ a simple linear PI controller is added.

In a second step, an addidional position controller using the ILC approach is added to have a better or almost perfect tracking behavior for repeating sequences of the desired position φ_{d}.

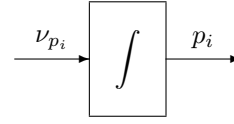The structure of the control loop can be seen in Fig. 5.

Fig. 3. Brunovsky normal form for the pneumatic joint, i. e. (2) with (8).

For the controller design of the position controller of the pneumatic joint (2), the mass flows are the the control inputs

$$\boldsymbol{u} = [\dot{m}_1, \dot{m}_2]^T, \tag{3}$$

and the state vector is defined as

$$\boldsymbol{x} = [\varphi, \dot{\varphi}, p_1, p_2]^T. \tag{4}$$

A. Flatness-based Approach – Inner Loop

According to [13] a flatness-based controller for the pressures in chamber 1 and 2, respectively, are designed. The underlying flatness-based pressure controllers for the chambers can be designed independently, since both chambers are only coupled via the position φ and velocity $\dot{\varphi}$ of the swivel. For the controller design these states can be considered as variable parameters as proposed in [5].

It is easy to see from (2b) that p_1 is a flat output for chamber 1, i. e.:

$$y_1 = p_1 \tag{5a}$$

$$\dot{y}_1 = \dot{p}_1 = \frac{n}{V_1(\varphi)} \left(RT\dot{m}_1 - p_1 \dot{\varphi} V_{\text{spec}} \right) = \theta_1(p_1, \dot{m}_1, \varphi, \dot{\varphi}), \tag{5b}$$

with $V_1(\varphi) = A_{\text{eff}} r_{\text{eff}} \, (\, |\varphi_{\max} + \varphi|)$ and $V_{\text{spec}} = A_{\text{eff}} r_{\text{eff}}$.

By introducing a new input ν_{p_1} for $\dot{p}_1$, one gets:

$$\dot{p}_1 = \frac{n}{V_1(\varphi)} \left(RT\dot{m}_1 - p_1 \dot{\varphi} V_{\text{spec}} \right) = \nu_{p_1}. \tag{6}$$

The nonlinearity of chamber 1 can now easily linearised by feedback solving (6) for the system input:

$$\dot{m}_1 = \frac{V_1(\varphi)}{nRT} \left(\nu_{p_1} - \frac{np_1 \dot{V}_1(\dot{\varphi})}{V_1(\varphi)} \right) = \eta_1(\varphi, \dot{\varphi}, p_1, \nu_{p_1}). \tag{7}$$

The same holds for the second chamber, i. e.:

$$\dot{m}_i = \eta_i(\varphi, \dot{\varphi}, p_i, \nu_{p_i}), \quad i \in \{1, 2\}. \tag{8}$$

Applying (8) to the model of the rotational joint (2) the systems is in Brunovsky canonical form [9], cf. Figure 3.

For the model of the rotational joint, linearized by feedback, i. e. (2) with (8), a linear controller for the pressures can easily designed:

$$\nu_{p_i} = \dot{p}_{i,\text{d}} - k_i(p_i - p_{i,\text{d}}), \quad i \in \{1, 2\}. \tag{9}$$

Position Control for the Linearized System

To control the postion of the rotational joint, a simple linear controller is used:

$$e = (\varphi_{\text{d}} - \varphi), \tag{10a}$$

$$\tau_{\text{PI}} = k_{\text{p}} e + k_{\text{I}} \int e \, \mathrm{d}t. \tag{10b}$$
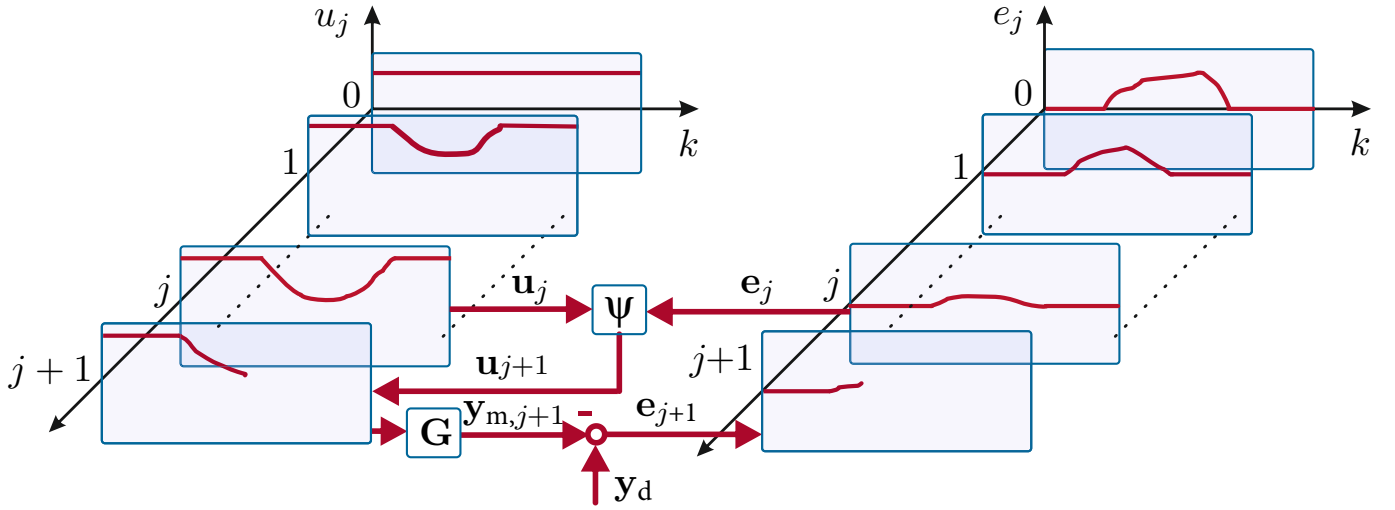
Fig. 4. Illustration of the ILC concept [3].

B. Iterative Learning Control – Outer Loop

In the controller desing so far, no friction model (cf. Remark 2.1) or other model uncertainties has been considered. To address this issues an Iterative Lerning Control approach is introduced and applied to control the position of the pneumatic robot joint.

Remark 3.1: One condition to successfully apply an ILC concept to an application is, that the trajectories are continuously repeating. Another condition is, that the system to be controlled is stable. That is why an underlying controller, e. g. the flatness-based pressure and position controller is applied to stabilize the system.   ◇

Iterative Learning Control works as follows:
In each iteration $(j = 0, 1, \ldots)$ a feedforward control value $\boldsymbol{u}_{j+1}$ is computed. In the feedforward control value $\boldsymbol{u}$ all the the sample values of a cycle j are collected upto the number of samples N during one cycle/iteration, as defined in (14).

To do this, the error $\boldsymbol{e}_j$ and the control $\boldsymbol{u}_j$ must be determined and stored. The output error

$$\boldsymbol{e} = \boldsymbol{y}_\mathrm{d} - \boldsymbol{y}_\mathrm{m} \equiv \boldsymbol{\varphi}_\mathrm{d} - \boldsymbol{\varphi} \tag{11}$$

is computed from the difference between the desired signal $\boldsymbol{y}_\mathrm{d}$ and the measured signal $\boldsymbol{y}_\mathrm{m}$. In Fig. 4 the sequences of the ILC approach is illustrated [3].

In each iteration, the control $\boldsymbol{u}_{j+1}$ is computed by the function $\boldsymbol{\psi}(\boldsymbol{u}_j, \boldsymbol{e}_j(\boldsymbol{u}_j))$. The new control therefore only depends on the previous iteration.

The resulting control signal is applied to a system G in order to measure the output $\boldsymbol{y}_{\mathrm{m},j+1}$. Using the setpoint signal $\boldsymbol{y}_\mathrm{d}$, the new output error $\boldsymbol{e}_{j+1}$ is determined and stored temporarily. At the same time, the control signal $\boldsymbol{u}_{j+1}$ is also saved.

This process can be repeated as often as required, since the goal of this procedure is to adjust the feedforward control $\boldsymbol{u}$ such that the output error $\boldsymbol{e}$ converges to zero.

For the computation of the control signal $\boldsymbol{u}$ a minimization task is used (cf. [3]). To do this, a gain matrix $L \in R^{N \times N}$ and

a filter matrix $Q \in R^{N \times N}$ is defined, where N is the number of samplings during one iteration.

The ILC law depends on the relative degree r of the system. For linear systems, the relative degree can be calculated by

$$r = n - m, \tag{12}$$

where the highest exponent of the numerator polynomial of the transfer function is given by m and the highest exponent of the denominator polynomial is given by n.

For discrete-time systems, $r = 1$ always applies. It can therefore also be interpreted as a delay.

In practical applications, in addition to the delay due to the relative degree r of the system, delays can also occur due to digital-to-analog or analog-to-digital conversions. This delay is given by the number of sampling steps w required for the conversion. This means that a total delay

$$v = r + w \tag{13}$$

can be introduced and taken into account in the ILC law.

By introducing

$$\boldsymbol{u}_j = \begin{bmatrix} u_j[0] & u_j[1] & \ldots & u_j[N-1] \end{bmatrix}^T \in R^{N \times 1} \tag{14}$$

$$\boldsymbol{y}_{\mathrm{d},j} = \begin{bmatrix} y_{\mathrm{d},j}[v] & y_{\mathrm{d},j}[v+1] & \ldots & y_{\mathrm{d},j}[v+N-1] \end{bmatrix}^T \in R^{N \times 1} \tag{15}$$

$$\boldsymbol{y}_{\mathrm{m},j} = \begin{bmatrix} y_{\mathrm{m},j}[v] & y_{\mathrm{m},j}[v+1] & \ldots & y_{\mathrm{m},j}[v+N-1] \end{bmatrix}^T \in R^{N \times 1} \tag{16}$$

$$\begin{aligned} \boldsymbol{e}_j &= \boldsymbol{y}_{\mathrm{d},j} - \boldsymbol{y}_{\mathrm{m},j} \\ &= \begin{bmatrix} e_j[v] & e_j[v+1] & \ldots & e_j[v+N-1] \end{bmatrix}^T \in R^{N \times 1}, \end{aligned} \tag{17}$$

the ILC control law is given in lifted representation:

$$\boldsymbol{u}_{j+1} = Q\left(\boldsymbol{u}_j + L\boldsymbol{e}_j\right). \tag{18}$$

The gain matrix L has the task of compute the influence of the output error $\boldsymbol{e}_j$ in relation to the previous control $\boldsymbol{u}_j$.
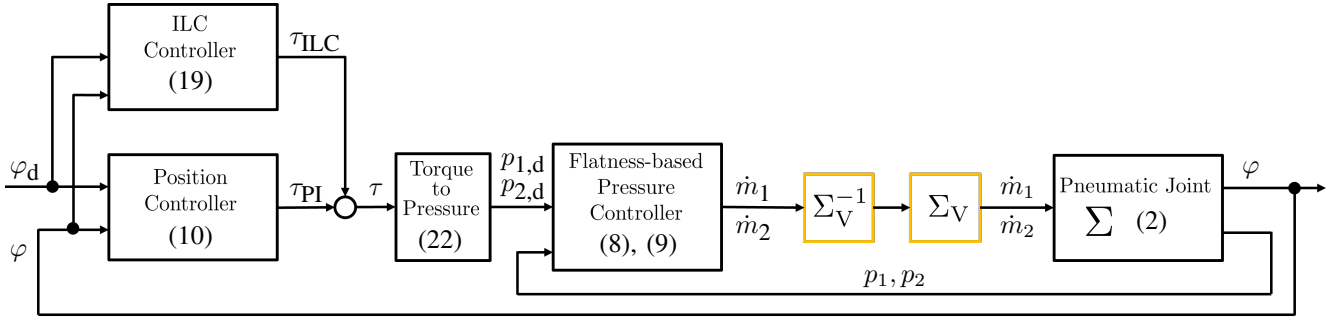
Fig. 5. Overall control structure. The compensation of the valve behavior, i.e. $\Sigma_V \circ \Sigma_V^{-1}$, is not considered here.

The choice of matrix elements determine the influence of the output error on the new feedforward-control. The filter matrix Q, on the other hand, is used to suppress measurement noise and repetitive interference. It acts like a low-pass filter that only allows low frequencies to pass through and filters out high frequencies. Furthermore acausale filters can be used, e. g. a Gaussian filter as proposed in [3].

The ILC law (18) can also be specified in the discrete form:

$$u_{j+1}[k] = q[k]\big(u_j[k] + l[k]\,e_j[k+v]\big), \tag{19a}$$
$$\tau_{\text{ILC}} = u_{j+1}[k] \tag{19b}$$

where $k = 0, 1, \ldots, N-1$ is the sampling index.

Remark 3.2: Stability and convergence can be guaranteed for the ILC control law (18), by ensuring

$$\rho(Q(I - IG)) < 1, \tag{20a}$$
$$\rho(GQ(I - IG)G^{-1}) < 1, \tag{20b}$$

where I is the identity matrix and ρ is the spectral radius as defined in [7]. ◇

C. Splitting the Torque into Chamber Pressures

The output of the controllers (10) and (19b) are torques, which have to be added and then converted or split respectively into some feasible pressure demands for the underlying pressure controller (9), i.e.:

$$\tau = \tau_{\text{PI}} + \tau_{\text{ILC}}, \tag{21}$$
$$p_{1,\text{d}} = p_{\text{m}} + \frac{\tau}{2\,V_{\text{spec}}}, \tag{22a}$$
$$p_{2,\text{d}} = p_{\text{m}} - \frac{\tau}{2\,V_{\text{spec}}}, \tag{22b}$$

while p_{m} is the so-called mean pressure, which has to be defined appropriately.

Remark 3.3: The pressures p_i range around the mean pressure p_{m}. Note, this mean pressure can be changed during time, depending on the state of the robot or joint, i.e. $p_{\text{m}} = p_{\text{m}}(\boldsymbol{x}, t)$. In this case, this is a nonlinear MIMO controller. ◇

The overall control structure can be seen in Fig.5. The compensation of the nonlinear valve behavior, i.e. $\Sigma_V \circ \Sigma_V^{-1}$, is not considered here. A detailed description of those mass flow controllers acting on the valve Σ_V can be found in [8].

IV. EXPERIMENTAL SETUP AND RESULTS

The proposed concept has been tested in simulation and in an experimental setup. However, in this paper only the experimental results are shown and discussed. For the experiments a simple P-type ILC has been used, i.e. the gain in (19a) is defined as $l[k] = k_{\text{p,ILC}}$, $\forall k$.

A. Experimental Setup

The introduced control concept was tested on an experimental setup as seen in Fig. 6. This test bench consists of a single pneumatic joint, which is controlled within a rapid control prototyping system from dSpace®. The testbench consists of pressure control valve for the supply pressure (1), buffer volume (2), flow sensors (3), pressure sensor for the supply pressure (4), pneumatic valve with valve controller (5), pressure sensor for the chambers (6,7), pneumatic joint (8), and an interface PCB to the dSpace®-system.

B. Experimental Results

To test the proposed concept a typical test-trajectory is applied to one of the swivel drives (Fig. 2) of the pneumatic robot (Fig. 1). The test trajectory consists of dynamic and stationary parts as can be seen in Fig. 7a. The trajectory tracking performance after several interations (repetitions) can be seen in Fig. 7a and the tracking error in Fig. 7b.
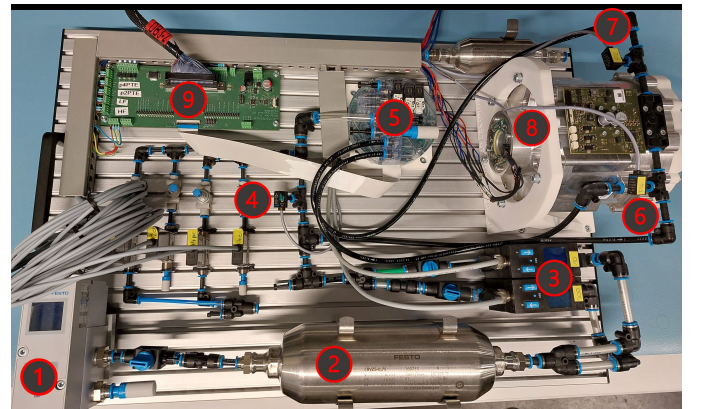

Fig. 6. Experimental Setup: the pneumatic joint (8) is controlled via a proportional Piezo valve (5) and a rapid-control-prototyping-system from dSpace®.
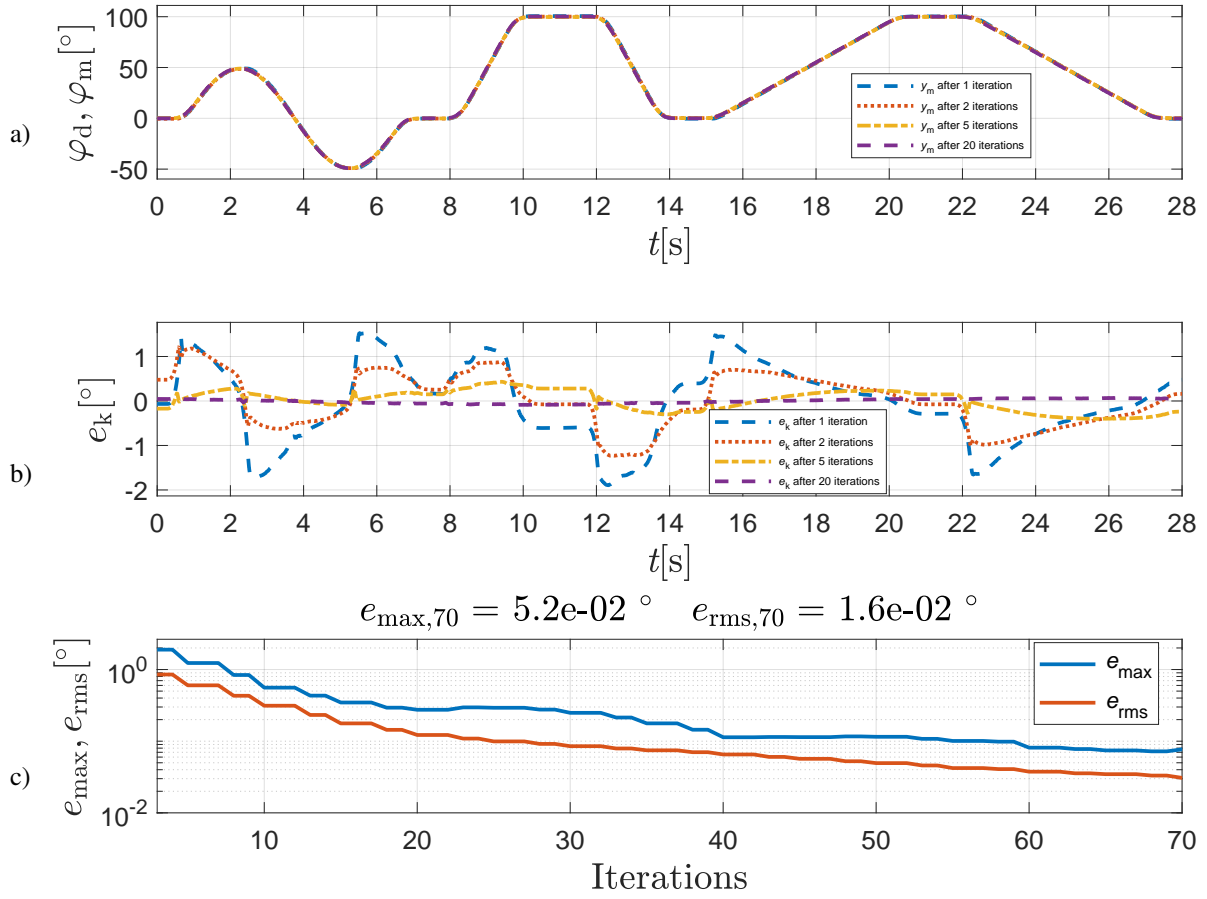
$$e_{\max,70} = 5.2\text{e-}02~^\circ \qquad e_{\mathrm{rms},70} = 1.6\text{e-}02~^\circ$$

Fig. 7. Measurement resutls for a robot joint.

Remember, the control value τ_{ILC} depends on the previous iteration. Since for the first iteration there is no previous one, the performance after the first iteration corresponds to the tracking performance of the flatness-based controller only. A maximal tracking error of $e_{\max} \approx 1.8°$ can be seen after the first run, i.e. for the controller without the ILC contribution. As can be seen clearly after several runs, the tracking error decreases significantly. It can be seen in Fig. 7b, that after 20 runs the tracking error is almost zero.

As mentioned in Remark 2.1, no friction compensation is implemented explicitly in the proposed control concept, but the friction effects are compensated perfectly by the ILC approach.

The evolution of the maximal control error and the root-mean-square-error (rms) over several runs is almost neglectable, as can be easily seen in Fig. 7c. Note, the ordinate in Fig. 7c is scaled logarithmically.

Remark 4.1: Controlling the smallest drive in the robot, i.e. the drive for the hand-axis, is really a challenging task. For this axis the pneumatic actuator is rather tiny and therefore the relation between the driving torque created by the pneumatics and the friction torque is extremely unfavorable. With the proposed ILC concept it was possible to achieve almost perfect tracking performance, even for the pneumatic drive of the hand-axis. This has so far not been possible with any other control concept. ◇

V. SUMMARY

This paper successfully demonstrated that the use of Iterative Learning Control Learning Control (ILC) improves the control performance of repetitive motion significantly over time, even for complex nonlinear systems. By learning from previous cycles, the control signal can be adapted for the next cycle, resulting in improved control quality. Especially for pneumatic joints built in a collaborative robot, this concept is highly suitable, since friction and model inaccuracies can be compensated almost perfectly.

A major advantage of the ILC algorithm is that it can be used in parallel with any existing controller. One prerequisite for this is, that the desired trajectory must be identical for each cycle, which is why the ILC approach is mostly used in a controller cascade in parallel with the superimposed controller.

The acausal filtering in the ILC algorithm enables the control signal to be filtered without phase shift. This is particularly important when there is measurement noise in the controlled variable and when used simultaneously with an existing controller.

It is successfully demonstrated, that a two-degree-of-freedom approach (flatness-based pressure controller and ILC position controller) leads to an almost perfect tracking performance for repetitive trajectories for pneumatic robot joints.

REFERENCES

[1] Hyo-Sung Ahn, YangQuan Chen, and Kevin L. Moore. Iterative Learning Control: Brief Survey and Categorization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1099–1121, 2007.

[2] J.E. Bobrow and B.W. McDonell. Modeling, identification, and control of a pneumatically actuated, force controllable robot. *IEEE Transactions on Robotics and Automation*, 14(5):732–742, 1998.

[3] M. Böck, T. Glück, A. Kugi, and A. Steinböck. *Fortgeschrittene Methoden der nichtlinearen Regelung*. TU Wien - Institut für Automatisierungs- und Regelungstechnik Gruppe für komplexe dynamische Systeme, 2015.

[4] David Bou Saba, Paolo Massioni, Eric Bideaux, and Xavier Brun. Flatness-based control of a two degrees-of-freedom platform with pneumatic artificial muscles. *Journal of Dynamic Systems, Measurement, and Control*, 141(2), October 2018.

[5] Valentin Falkenhahn. Modellierung und Modellbasierte Regelung von Kontinuum-Manipulatoren, 2017.

[6] M. Flies, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples, 1995.

[7] Gene H. Golub and Charles F. Van Loan. *Matrix Computations - 4th Edition*. Johns Hopkins University Press, Philadelphia, PA, 2013.

[8] K. Hoffmann, D. Müller, R. Simon, and O. Savodny. On trajectory tracking control of fluid-driven actuators. *at – Automatisierungstechnik*, 69(11):970–980, 2021.

[9] Alberto Isidori. *Nonlinear control systems*. Springer Science & Business Media, third edition, 1995.

[10] Richard W. Longman. Iterative Learning Control and Repetitive Control for Engineering Practice. *International journal of control*, 73(10):930–954, 2000.

[11] G. Mattiazzo, S. Mauro, T. Raparelli, and M. Velardocchia. Control of a six-axis pneumatic robot. *Journal of Robotic Systems*, 19(8):363–378, July 2002.

[12] David H. Owens. *Iterative Learning Control - An Optimization Paradigm*. Advances in Industrial Control. Springer-Verlag GmbH, 2016.

[13] Ralf Rothfuß. Anwendung der flachheitsbasierten Analyse und Regelung nichtlinearer Mehrgrößensysteme, 1997.

[14] G. Stadler, A. Steinboeck, L. Marko, A. Deutschmann-Olek, and A. Kugi. Iterative learning and feedback control for the curvature and contact force of a metal strip on a roll. *Control Engineering Practice*, 121:105071, 2022.

[15] Javad Taghia, André Wilkening, and Oleg Ivlev. Position control of soft-robots with rotary-type pneumatic actuators. In *German Conference on Robotics*, 2012.

[16] B. Tondu, S. Ippolito, J. Guiochet, and A. Daidie. A seven-degrees-of-freedom robot-arm driven by pneumatic artificial muscles for humanoid robots. *The International Journal of Robotics Research*, 24(4):257–274, April 2005.