

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—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 to repeatable control tasks as they appear typically in robot applications [6].

Index Terms—model-based control, robot control

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 lies at hand. 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 on 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 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 performance and disturbance rejection can be improved significantly. This is achieved, as the *ILC* controller reduces the periodic errors and the feedback controller compensates for the disturbances [2].

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

As a drafted version the paper of Steinboeck can be used [8]



Fig. 1. Pneumatic Cobot from Festo (2020)

A. 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 $\pm 135^\circ$. The pressure difference between the two chambers 1 and 2 of the joint generates a driving torque

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

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 [4].

The governing equations of the rotary joint model are:

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

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

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

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.

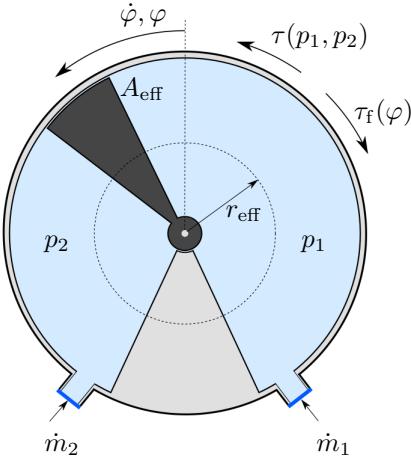


Fig. 2. Schematics of a pneumatic rotary joint.

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

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, a 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 Figure 3.

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

$$\mathbf{u} = [\dot{m}_1, \dot{m}_2]^T, \quad (3)$$

and the state vector is defined as

$$\mathbf{x} = [\varphi, \dot{\varphi}, p_1, p_2]^T. \quad (4)$$

A. Flatness-based Approach – Inner Loop

According to [7] 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 [3].

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

$$y_1 = p_1 \quad (5a)$$

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

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:

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

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

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

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

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

Applying (8) to the model of the rotational joint (2) the systems is in Brunovsky canonical form [5], cf. Figure 4.

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

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

Position Control for the Linearized System

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

$$e = (\varphi_d - \varphi), \quad (10a)$$

$$\tau_{\text{PI}} = k_p e + k_I \int e dt. \quad (10b)$$

Split the torque into chamber pressures

Since the output of the controller (10) is a torque, the torque has to be converted / split into some feasible pressure demands for the underlying pressure controller (9):

$$p_{1,d} = p_m + \frac{\tau_{\text{PI}}}{2 V_{\text{spec}}}, \quad (11a)$$

$$p_{2,d} = p_m - \frac{\tau_{\text{PI}}}{2 V_{\text{spec}}}, \quad (11b)$$

while p_m is the so-called mean pressure, which has to be defined appropriately.

Remark 3.1: The pressures p_i range around this mean pressure. Note, this mean pressure can be changed during time, depending on the state of the robot or joint, i.e. $p_m = p_m(\mathbf{x}, t)$. In this case, this is a nonlinear MIMO controller. \square

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 Iterarive Lerning Control approach is introduced.

Remark 3.2: 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. \square

Interativ Learning Control works as follows: In each iteration ($j = 0, 1, \dots$) a feedforward control value \mathbf{u}_{j+1} is calculated.

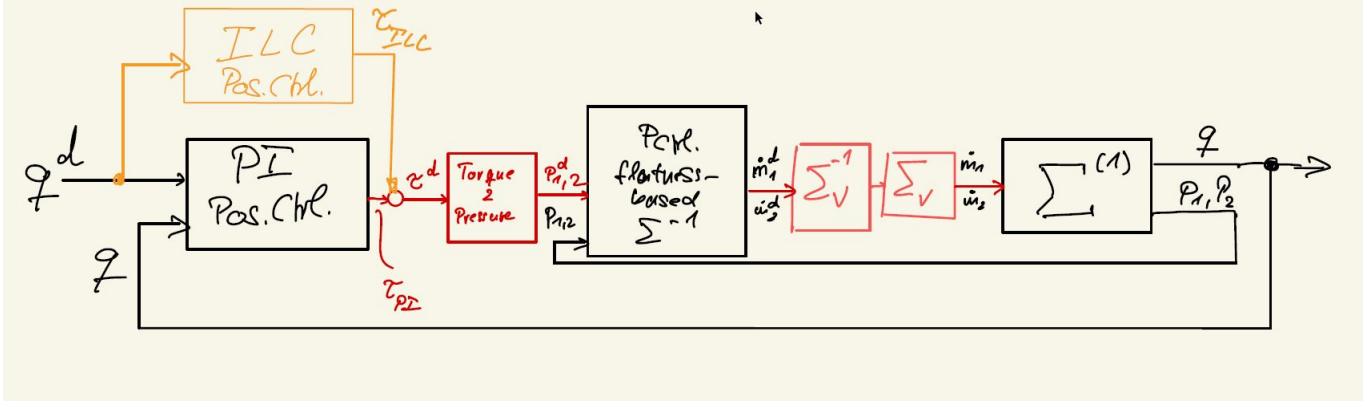


Fig. 3. Control Structure

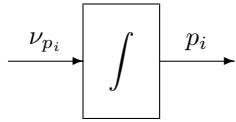


Fig. 4. Brunovský normal form for the pneumatic joint, i.e. (2) with (8).

To do this, the error e_j and the control u_j must be determined and stored. The output error

$$e = \mathbf{y}_d - \mathbf{y}_m \quad (12)$$

is calculated from the difference between the desired signal \mathbf{y}_d and the measured signal \mathbf{y}_m . In Fig. 5 the sequences of the ILC approach is illustrated [2].

In each iteration, the control u_{j+1} is calculated by the function $\psi(u_j, e_j(u_j))$. The new control therefore only depends on the previous iteration.

The resulting control signal is applied to the system G in order to measure the output $\mathbf{y}_{m,j+1}$. Using the setpoint signal \mathbf{y}_d , the new output error e_{j+1} is determined and stored temporarily. At the same time, the control signal 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 u such that the output error e converges to zero.

For the computation of the control signal u a minimization task is used (cf. [2]). 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

$$r = n - m, \quad (13)$$

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

$$v = r + w \quad (14)$$

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

By introducing

$$\mathbf{u}_j = [u_j[0] \ u_j[1] \ \dots \ u_j[N-1]]^T \in R^{N \times 1} \quad (15)$$

$$\mathbf{y}_{d,j} = [y_{d,j}[v] \ y_{d,j}[v+1] \ \dots \ y_{d,j}[v+N-1]]^T \in R^{N \times 1} \quad (16)$$

$$\mathbf{y}_{m,j} = [y_{m,j}[v] \ y_{m,j}[v+1] \ \dots \ y_{m,j}[v+N-1]]^T \in R^{N \times 1} \quad (17)$$

$$e_j = \mathbf{y}_{d,j} - \mathbf{y}_{m,j} = [e_j[v] \ e_j[v+1] \ \dots \ e_j[v+N-1]]^T \in R^{N \times 1} \quad (18)$$

the ILC control law is given in lifted-shift representation:

$$\mathbf{u}_{j+1} = Q(\mathbf{u}_j + L e_j). \quad (19)$$

The gain matrix L has the task of calculating the influence of the output error e_j in relation to the previous control u_j . 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.

The ILC law (19) can also be specified in the discrete form:

$$u_{j+1}[k] = q[k](u_j[k] + l[k] e_j[k+v]), \quad (20)$$

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

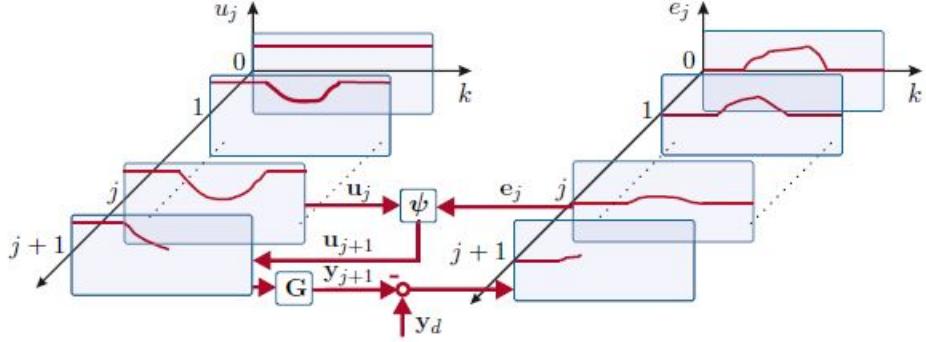


Fig. 5. Illustration of the ILC concept [2]

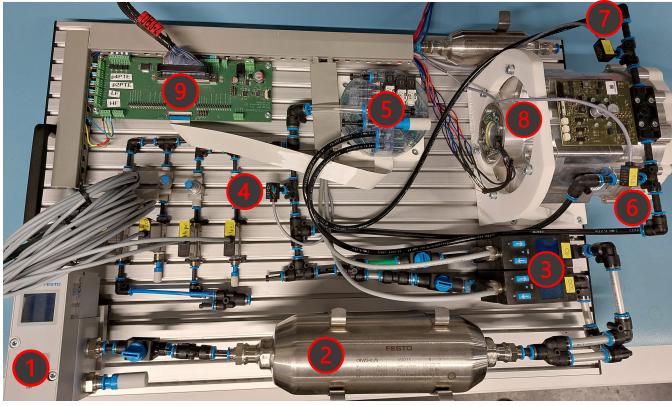


Fig. 6. Experimental Setup

IV. EXPERIMENTAL RESULTS

V. SUMMARY

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] 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.
- [3] Valentin Falkenhahn. Modellierung und Modellbasierte Regelung von Kontinuum-Manipulatoren, 2017.
- [4] K. Hoffmann, D. Müller, R. Simon, and O. Savodny. On trajectory tracking control of fluid-driven actuators. *at – Automatisierungstechnik*, 69(11):970–980, 2021.
- [5] Alberto Isidori. *Nonlinear control systems*. Springer Science & Business Media, third edition, 1995.
- [6] Richard W. Longman. Iterative Learning Control and Repetitive Control for Engineering Practice. *International journal of control*, 73(10):930–954, 2000.
- [7] Ralf Rothfuß. Anwendung der flachheitsbasierten Analyse und Regelung nichtlinearer Mehrgrößensysteme, 1997.
- [8] 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.

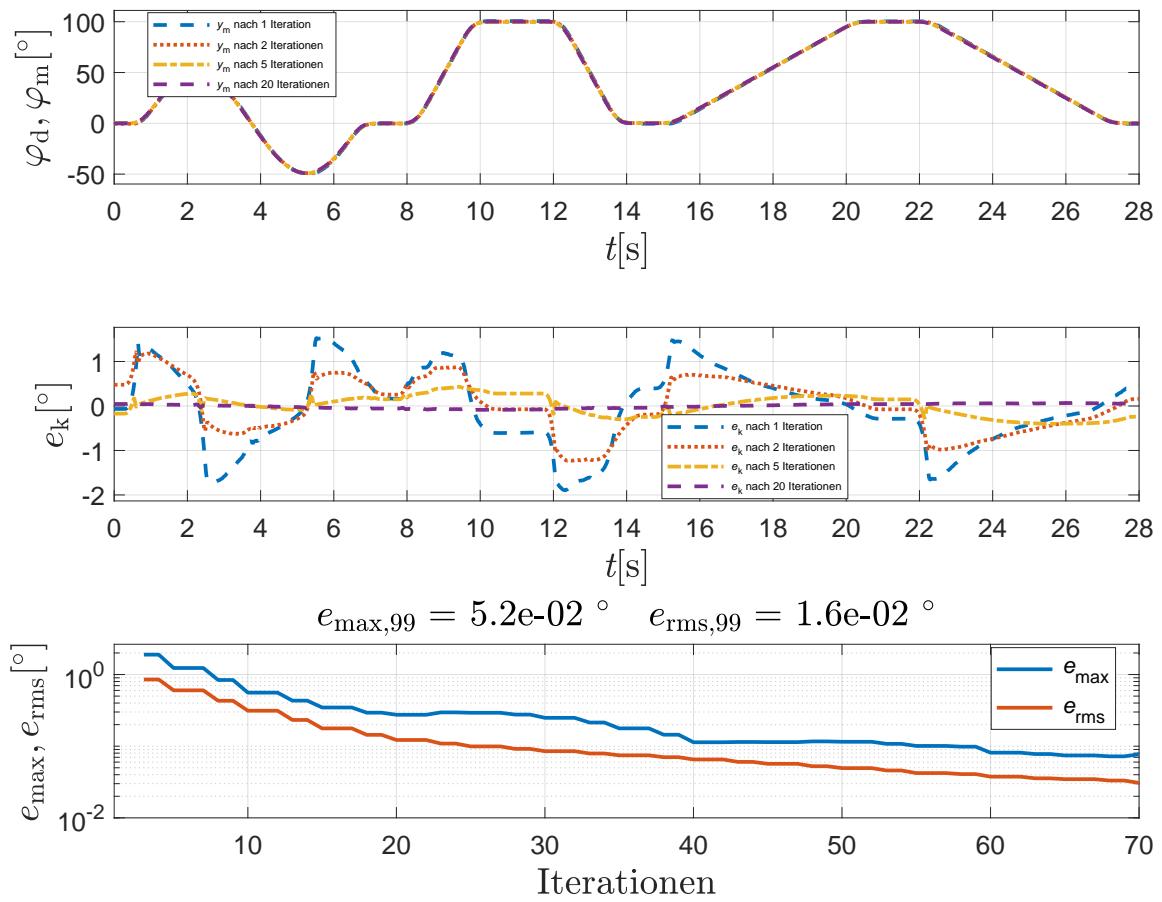


Fig. 7. Measurement results for a robot joint