An Input-mapping Initialization Approach to Accelerate Iterative Learning of Similar Tasks

Jeroen Willems*, Erik Hostens*, Bruno Depraetere*,
Daniele Ronzani†, Armin Steinhauser† and Jan Swevers†
*DecisionS lab, Flanders Make
Leuven, Belgium
Email: firstname.lastname@flandersmake.be
†MECO Research Team, KU Leuven and DMMS lab, Flanders Make
Leuven, Belgium
Email: firstname.lastname@kuleuven.be

***Abstract*—Iterative Learning Control (ILC) can yield superior performance for mechatronic systems that execute the same task consecutively. One major limitation of ILC however, is that the ILC algorithm has to relearn the optimal control input signal for every new task, which is time consuming. The convergence speed of the ILC can be improved by hot-starting new tasks, reusing data available from previous tasks. A hot-start is an improved initial input for a new task, which minimizes the need for additional learning. In this paper, a novel transformation-based approach to hot-start tracking or non-tracking tasks for nonlinear systems is derived. The proposed methodology is analysed in simulation examples and its effectiveness is demonstrated experimentally.**

I. Introduction

Iterative Learning Control (ILC) can yield superior performance for mechatronic systems that execute the same task consecutively [1], [2], [3]. The algorithm iteratively learns the optimal control input signal by learning from the error signals of previous iterations of the same task. ILC is therefore able to compensate for repetitive disturbances, and only requires an approximate model of the considered system [3], while still achieving high performance, making this control strategy very appealing for industry.

However, one major limitation is that for every new task, the ILC algorithm has to relearn the optimal control input signal. In an industrial setting, the optimal control signals often need to be learned for a wide range of configurations and tasks that—even though possibly very similar—require multiple time-consuming learning procedures. Examples are motions of weaving looms, where a wide range of patterns needs to be woven, each being almost identical to at least one other. Or looms with different cam profiles, realizing the same type of motion but with small variations. Other examples are positioning stages or pick-and-place robots, where similar tasks emerge from variations of the start and stop point, the allowed speed or acceleration, etc. Besides simply developing a quickly converging ILC, the total learning time for such applications can be drastically reduced by appropriately initializing or hot-starting the learning for new tasks. This hot-start yields an improved initial guess for the new task, reducing the need for additional learning and therefore speeding up convergence.

An algorithm that uses data of previously learned trajectories to initialize a new task is developed in [4]. This is done by deriving a transformation between the desired motion trajectories of the past and the new task. While this approach yields promising results, it assumes linear time-invariant (LTI) system behavior and is only applicable when a reference signal is available.

By employing basis functions, [5] parametrizes the ILC control signal in terms of the task, such that the result can be extrapolated towards different tasks. The downside of this approach, however, is that it is only designed for LTI systems. In [6], a multidimensional interpolation algorithm is employed in order to share data between similar tasks, and the capabilities of this approach are validated on an experimental setup. The downside of this lookup table-based approach is that the tasks need to be ordered on a specific grid, which is not always straightforward.

This paper explores a more generic method by extending the transformation-based approach of [4] towards nonlinear systems. Moreover, the proposed approach is also applicable to tasks that are not described by a motion trajectory. Such tasks can involve complex objectives and constraints, aligning it more with the current demands of industry. The approach is furthermore extended towards the inclusion of multiple past tasks and the developed concepts are validated both in simulation and experimentally on a nonlinear mechatronic system. The developed approach is combined with a generic two-step ILC approach [7]: first an explicit model correction is computed, followed by an optimization of the system inputs using the corrected system dynamics. This ILC is implemented using the learning control toolbox RoFaLT [8].

The remainder of this paper is organized as follows. Section II introduces the notation, the basics of the two-step ILC approach and the considered problem. The approach from [4] is revisited in Section III and is subsequently extended towards nonlinear systems, alternative control objectives and multiple past tasks in Section IV. While these extensions are supported by simulation examples, an experimental validation on a mechatronic slider-crank setup is conducted in Section V and conclusions are drawn in Section VI.

II. Preliminaries

A. Notation

A specific ILC iteration i is indicated by a subscript $(\cdot)_i$ and the collection of N time samples of an arbitrary signal is denoted by $\mathbf{x}_i = [x_i(1) \quad x_i(2) \quad \dots \quad x_i(N)]^T$. Utilizing this notation, a generic nonlinear system $\tilde{\mathbf{y}} = P(\mathbf{u})$ with the observed output $\tilde{\mathbf{y}}$ and applied input $\mathbf{u}$ is modeled as

$$\mathbf{y} = \hat{P}(\mathbf{u}) + \boldsymbol{\alpha} = h(\mathbf{u}, \boldsymbol{\alpha}), \tag{1}$$

where $\mathbf{y}$ is the expected output, $\hat{P}$ are the modeled dynamics and $\boldsymbol{\alpha}$ denote additive, nonparametric correction terms that are further explained below. The operator $\mathcal{T}(\mathbf{x})$ describes the lower triangular Toeplitz matrix of a vector

$$\mathcal{T}(\mathbf{x}) = \begin{bmatrix} x_1 & 0 & \cdots & 0 \\ x_2 & x_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ x_N & x_{N-1} & \dots & x_1 \end{bmatrix}. \tag{2}$$

B. ILC Algorithm

In this paper we make use of the generic two-step ILC approach presented in [7]. Each ILC iteration i consists of the following steps:

1) *Control step*: Given model correction terms $\boldsymbol{\alpha}_i$ (with $\boldsymbol{\alpha}_0 = 0$), calculate the input $\mathbf{u}_i$ that minimizes a given objective function J and constraints g:

$$\begin{aligned} \mathbf{u}_i = \arg\min_{\mathbf{u}} \quad & J(\mathbf{u}, \mathbf{y}_i, \boldsymbol{\alpha}_i) \\ \text{s.t.} \quad & \mathbf{y}_i = h(\mathbf{u}, \boldsymbol{\alpha}_i), \\ & g(\mathbf{u}, \mathbf{y}_i, \boldsymbol{\alpha}_i) \leq 0. \end{aligned} \tag{3}$$

If the objective is to follow a predefined reference $\mathbf{y}_r$, the objective function can be chosen as $J(\mathbf{y}_i) = \|\mathbf{y}_r - \mathbf{y}_i\|_2^2$.

2) *Model correction step*: Calculate $\boldsymbol{\alpha}_{i+1}$ that minimizes the difference between the predicted output $\mathbf{y}_i$ and output $\tilde{\mathbf{y}}_i$ observed after applying input $\mathbf{u}_i$:

$$\boldsymbol{\alpha}_{i+1} = \arg\min_{\boldsymbol{\alpha}} \ \|\tilde{\mathbf{y}}_i - h(\mathbf{u}_i, \boldsymbol{\alpha})\|_2^2. \tag{4}$$

C. Problem formulation

The use case considered is the motion control of a slider-crank mechanism that converts a rotary motion θ, caused by a torque τ, into a linear displacement x_{slider}, as schematically shown in Fig. 1. The goal is to perform a reference tracking task or a point-to-point motion while minimizing a more complex objective, such as the system's energy consumption. Using first principle modeling [6], a nonlinear dynamic model $\hat{P}$ is derived, which can be provided to RoFaLT in the form of ODEs [8]. In the following, a simulation study of the mechanism will be performed, using a perturbed version of the model $\hat{P}$ to introduce model-plant mismatch.

III. Existing approach for LTI systems

In this section, the transformation-based approach from [4] will be discussed. First, an overview of the method will be provided, followed by an application to the considered use case with an analysis of the results.

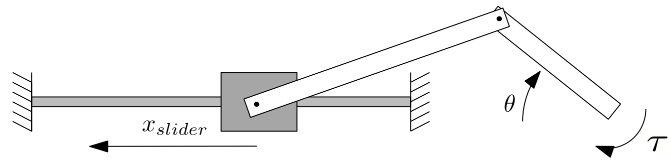

Fig. 1: Schematic representation of the setup.

A. Overview

The following assumptions are made in the context of the existing transformation-based approach:

- P is a discrete-time, linear time-invariant (LTI) system that can be equivalently expressed by $\tilde{\mathbf{y}} = \mathbf{P}\mathbf{u}$, using the lifted system representation [3].
- The system is modeled as $\mathbf{y} = \hat{\mathbf{P}}\mathbf{u}$.
- A reference tracking problem is considered with the objective function being $J(\mathbf{y}) = \|\mathbf{y}_r - \mathbf{y}\|_2^2$.
- $\mathbf{y}_{r,[1]}$ and $\mathbf{y}_{r,[2]}$ are similar and have equal signal length N.

The optimal input signal $\mathbf{u}_{opt,[1]}$ that perfectly tracks $\mathbf{y}_{r,[1]}$ is then given by

$$\mathbf{u}_{opt,[1]} = \mathbf{P}^{-1}\mathbf{y}_{r,[1]}. \tag{5}$$

It is assumed that, instead of using (5), this input signal $\mathbf{u}_{opt,[1]}$ is learned by applying ILC (cf. Section II-B) to track the reference signal $\mathbf{y}_{r,[1]}$, since exact knowledge of the plant is not available. After learning this task, the goal is to track another task with reference signal $\mathbf{y}_{r,[2]}$. Instead of learning this task without any initial knowledge, [4] proposes the following procedure to reduce the number of iterations required to learn $\mathbf{u}_{opt,[2]}$.

Procedure 1 Transformation-based approach for LTI systems

1: Construct the lower-triangular Toeplitz matrix $\mathbf{K}_r$ that transforms the desired output of task 1 to the desired output of task 2:

$$\mathbf{K}_r = \mathbf{Y}_{r,[1]}^{-1}\mathbf{Y}_{r,[2]}, \tag{6}$$

where $\mathbf{Y}_{r,[i]} = \mathcal{T}(\mathbf{y}_{r,[i]})$, with $i \in [1, 2]$, according to (2).

2: Compute the hot-start to track $\mathbf{y}_{r,[2]}$ – the estimated input signal $\hat{\mathbf{u}}_{opt,[2]}$ – as follows:

$$\hat{\mathbf{u}}_{opt,[2]} = \mathbf{K}_r\mathbf{u}_{opt,[1]}. \tag{7}$$

The considered approach initializes task 2 using data of the previously learned task 1, linearly approximating a potentially nonlinear input-output relation through (6) and (7).

B. Simulation study for the slider-crank mechanism

In this section, the previously discussed Procedure 1 is applied to the problem defined in Section II-C with the objective of minimizing the tracking error for a given reference $\mathbf{y}_r$. In Fig. 2, the initial input $\mathbf{u}_{init,[1]}$ is shown along with the converged input $\mathbf{u}_{opt,[1]}$ obtained from applying ILC. The

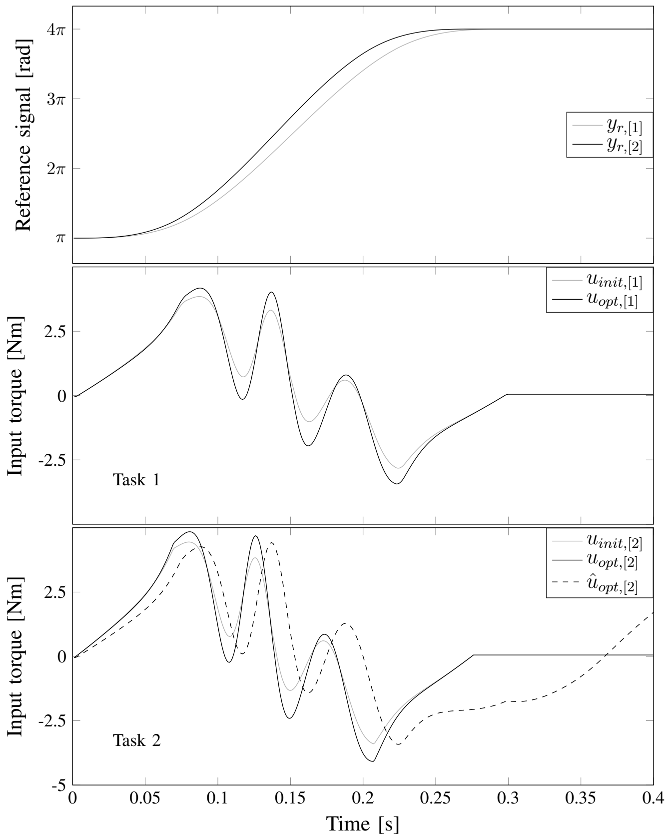

Fig. 2: Reference trajectories (top) and respective initial and optimal input signals for task 1 (middle) and task 2 (bottom), with the hot-start $\hat{\mathbf{u}}_{opt,[2]}$ for task 2 (bottom) derived using Procedure 1.

initial input for task 2, $\mathbf{u}_{init,[2]}$, can be obtained without executing any physical experiments. Finally, the hot-start $\hat{\mathbf{u}}_{opt,[2]}$ is derived using (6) and is also depicted in Fig. 2 along with $\mathbf{u}_{init,[2]}$ and the converged input $\mathbf{u}_{opt,[2]}$. It can be seen that $\hat{\mathbf{u}}_{opt,[2]}$ significantly deviates from $\mathbf{u}_{opt,[2]}$, indicating that Procedure 1 yields a poor hot-start for this problem. Additionally, $\|\mathbf{u}_{opt,[2]} - \hat{\mathbf{u}}_{opt,[2]}\| > \|\mathbf{u}_{opt,[2]} - \mathbf{u}_{init,[2]}\|$, suggesting that the initial input is a better starting point for applying ILC, compared to the computed hot-start. These findings can be attributed to the fact that the linear approximation of the nonlinear input-output relation is not capturing the system dynamics well enough for changing references.

IV. Novel transformation-based approach for nonlinear systems

In this section a novel transformation-based approach is proposed that extends Procedure 1 towards nonlinear systems and allows consideration of non-tracking tasks. First, an overview of the proposed approach will be provided, where we will hot-start the new task with data of one previous task. Second, the approach will be extended such that the hot-start of the new task can be composed of multiple past tasks.

A. Overview

Unlike the transformation-based approach presented in [4], the approach proposed in Procedure 2 represents a linear transformation exclusively in the input-space of the system. When applied to nonlinear systems and under the assumption of task similarity, nonlinearities of the input-output relation are captured by the ILC algorithm and the proposed transformation yields a hot-start superior to the result obtained by Procedure 1. Since the proposed approach is not considering the references but derives a transformation exclusively based on inputs, another advantage is that it is not limited to tracking tasks.

Procedure 2 Transformation-based approach for nonlinear systems

1: Obtain $\mathbf{u}_{init,[1]}$ for task 1 by solving (3) with $\boldsymbol{\alpha}_i = 0$.
2: Learn $\mathbf{u}_{opt,[1]}$ by applying ILC (cf. Section II-B).
3: Obtain $\mathbf{u}_{init,[2]}$ for task 2 by solving (3) with $\boldsymbol{\alpha}_i = 0$. Note that this step does not require any physical experiments.
4: Construct $\mathbf{K}_{[i]}$ that transforms $\mathbf{u}_{init,[i]}$ to $\mathbf{u}_{opt,[i]}$ as

$$\mathbf{K}_{[i]} = \mathbf{U}_{init,[i]}^{-1}\mathbf{U}_{opt,[i]}, \tag{8}$$

where $\mathbf{U}_{init,[i]} = \mathcal{T}(\mathbf{u}_{init,[i]})$ and $\mathbf{U}_{opt,[i]} = \mathcal{T}(\mathbf{u}_{opt,[i]})$.
5: Compute the hot-start for task 2 with $i = 1$ as follows:

$$\hat{\mathbf{u}}_{opt,[2]} = \mathbf{K}_{[1]}\mathbf{u}_{init,[2]}. \tag{9}$$

Proof that (9) is equal to (6) for LTI systems. Consider an LTI system modeled with the non-singular, lower-triangular Toeplitz matrix $\hat{\mathbf{P}}$ such that

$$\mathbf{u}_{init,[i]} = \hat{\mathbf{P}}^{-1}\mathbf{y}_{r,[i]}. \tag{10}$$

Since it holds for Toeplitz matrices that

$$\mathcal{T}(\mathbf{u})\mathbf{v} = \mathcal{T}(\mathbf{v})\mathbf{u} \quad \forall\ \mathbf{u}, \mathbf{v} \in \mathbb{R}^N, \tag{11}$$

one can reformulate (9) into

$$\begin{aligned}
\hat{\mathbf{u}}_{opt,[2]} &= \mathbf{U}_{init,[1]}^{-1}\mathbf{U}_{opt,[1]}\mathbf{u}_{init,[2]} && (12)\\
&= \mathbf{U}_{init,[1]}^{-1}\mathbf{U}_{init,[2]}\mathbf{u}_{opt,[1]} && (13)\\
&= (\hat{\mathbf{P}}^{-1}\mathbf{Y}_{r,[1]})^{-1}\hat{\mathbf{P}}^{-1}\mathbf{Y}_{r,[2]}\mathbf{u}_{opt,[1]} && (14)\\
&= \mathbf{Y}_{r,[1]}^{-1}\mathbf{Y}_{r,[2]}\mathbf{u}_{opt,[1]} = \mathbf{K}_r\mathbf{u}_{opt,[1]}. && (15)
\end{aligned}$$

Hence, for LTI systems, Procedure 2 and Procedure 1 are equivalent. □

B. Reference tracking problem

In this section we revisit the tracking problem of Section III-B, but using Procedure 2 to derive the hot-start $\hat{\mathbf{u}}_{opt,[2]}$. The resulting hot-start is shown in Fig. 3, along with the previously obtained input for the purpose of comparison. It is obvious that the hot-start derived using Procedure 2 is significantly closer to $\mathbf{u}_{opt,[2]}$ than the hot-start obtained from Procedure 1. A comparison of the tracking error resulting from the obtained input signals is also given in Fig. 3, clearly showing that the

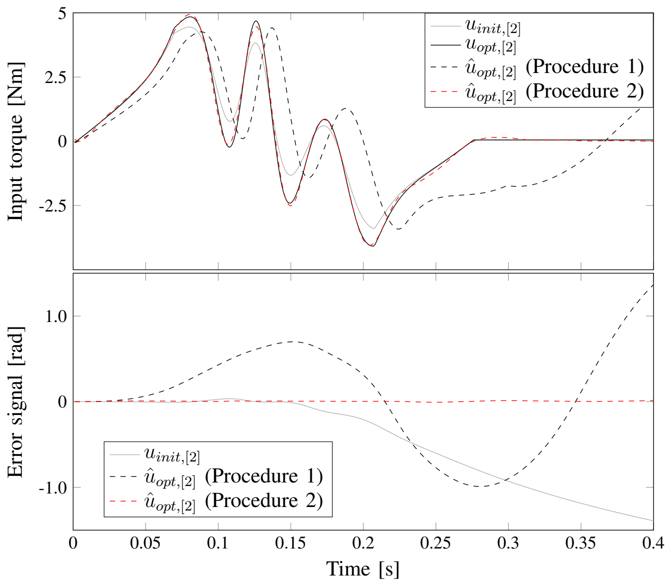

Fig. 3: Initial, optimal and hot-started (from Procedure 1 and 2) inputs (top) and their resulting tracking error (bottom).

hot-start using Procedure 2 yields the best performance. The approach proposed in Procedure 2 thus provides a hot-start that outperforms Procedure 1, resulting in a superior starting point for the ILC algorithm and therefore requiring less iterations to converge.

C. Minimizing energy consumption

In this section, the problem introduced in Section II-C is considered with the objective to minimize energy consumption of the system, denoted as

$$\min_{\boldsymbol{\tau},\boldsymbol{\omega}} \quad \Delta t\, \boldsymbol{\omega}^T \boldsymbol{\tau} + \Delta t\, \frac{R_m}{K_m^2}\, \boldsymbol{\tau}^T \boldsymbol{\tau}, \tag{16}$$

where Δt is the sampling time, $\boldsymbol{\tau}$ represents the torque, $\boldsymbol{\omega}$ is the angular velocity and R_m and K_m denote the rotary motor resistance and motor constant, respectively. Additionally, the following motion constraints are taken into account:

- The key constraint, graphically represented in Fig. 4, involves the timing of the displacement of the slider:

$$x_{\text{slider}}(k) \geq x_{\min} \quad \forall k \in \{k_{\text{left}}, \ldots, N - k_{\text{right}}\}. \tag{17}$$

- The motor velocity must be positive at all times $\frac{d\theta}{dt} = \omega \geq 0$.
- The initial condition is set to $\theta_0 = \pi$, $\omega_0 = 0$.
- The final condition is set to $\theta_N = 3\pi$, $\omega_N = 0$.

The two considered tasks have $N = 150$ samples with $k_{\text{left}} = k_{\text{right}} = 35$ for task 1 and $k_{\text{left}} = k_{\text{right}} = 44$ for task 2. The initial input $\mathbf{u}_{init,[2]}$ and the hot-start $\hat{\mathbf{u}}_{opt,[2]}$ (derived using Procedure 2) are shown in Fig. 5, along with the converged input signal $\mathbf{u}_{opt,[2]}$ if we would have applied ILC directly. It can be seen that the resulting hot-start $\hat{\mathbf{u}}_{opt,[2]}$ resembles $\mathbf{u}_{opt,[2]}$, which is also indicated by comparing the normed difference $\|\mathbf{u}_{opt,[2]} - \hat{\mathbf{u}}_{opt,[2]}\| < \|\mathbf{u}_{opt,[2]} - \mathbf{u}_{init,[2]}\|$.

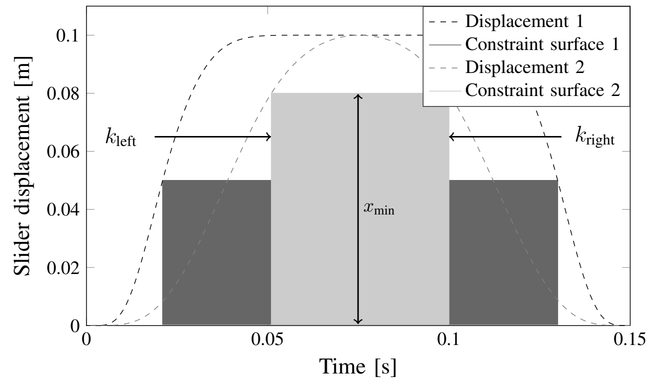

Fig. 4: Two different constraint surfaces with corresponding feasible displacement.

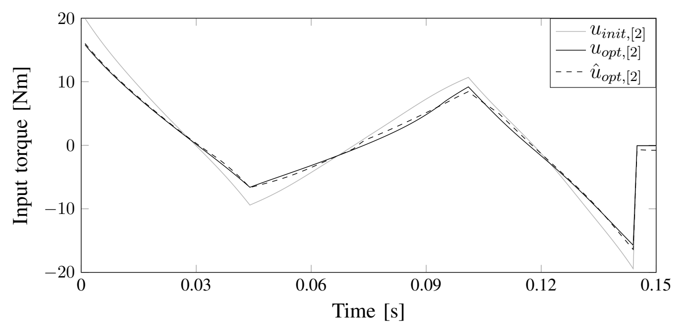

Fig. 5: Initial input signal $\mathbf{u}_{init,[2]}$, the hot-start $\hat{\mathbf{u}}_{opt,[2]}$ derived using Procedure 2, and the optimal input signal $\mathbf{u}_{opt,[2]}$ for task 2.

D. Inclusion of multiple previous tasks

As shown in Procedure 3, the proposed Procedure 2 can be extended to take multiple past tasks into account, which is supposed to increase the quality of the hot-start.

Procedure 3 Transformation-based approach for nonlinear systems with inclusion of multiple past tasks

1: Given tuples $\left\{\mathbf{u}_{init,[i]}, \mathbf{u}_{opt,[i]}\right\} \forall\, i \in I = \{1, \ldots, N_{\text{tasks}}\}$.
2: Compute $\mathbf{K}_{[i]}$ $\forall\ i \in I$ according to (8).
3: For a new task $j \notin I$ obtain $\mathbf{u}_{init,[j]}$ and compute separate hot-starts according to (9) for every past task as

$$\hat{\mathbf{u}}_{[j],[i]} = \mathbf{K}_{[i]} \mathbf{u}_{init,[j]}. \tag{18}$$

4: Combine the generated hot-starts to form the actual hot-start $\hat{\mathbf{u}}_{opt,[j]}$. A possible combination can be achieved by using inverse distance weighting [9],

$$\hat{\mathbf{u}}_{opt,[j]} = \frac{\sum_{i=1}^{N_{\text{tasks}}} w_i\, \hat{\mathbf{u}}_{[j],[i]}}{\sum_{i=1}^{N_{\text{tasks}}} w_i}, \tag{19}$$

where $w_i = (d_i^p)^{-1}$ is a weight with d_i a scalar distance measure from the data of tasks i and j and p is the power of the weight.

Remark. *An exemplary choice for d_i in (19) is to consider the norm difference in nominal model inputs $d_i = \|\mathbf{u}_{init,[j]} - \mathbf{u}_{init,[i]}\|$. Alternatively, one can e.g. weigh on the basis of time since last execution of the task, a certain machine parameter, or even among different machines of the same type.*

To show its effectiveness, Procedure 3 is applied to the problem stated in Section IV-C, but with four similar past tasks $k_{\text{left}} = k_{\text{right}} \in \{36, 50, 40, 39\}$ and $k_{\text{left}} = k_{\text{right}} = 44$ for the new task. In Fig. 6, the optimal inputs $\mathbf{u}_{opt,[i]}$ of the past tasks are shown along with the hot-starts from each past task to the new task (18), where a substantial similarity of the separate hot-starts can be seen, indicating that the nonlinear dynamics of the slider-crank system are excited in approximately the same way, despite the variation in tasks.

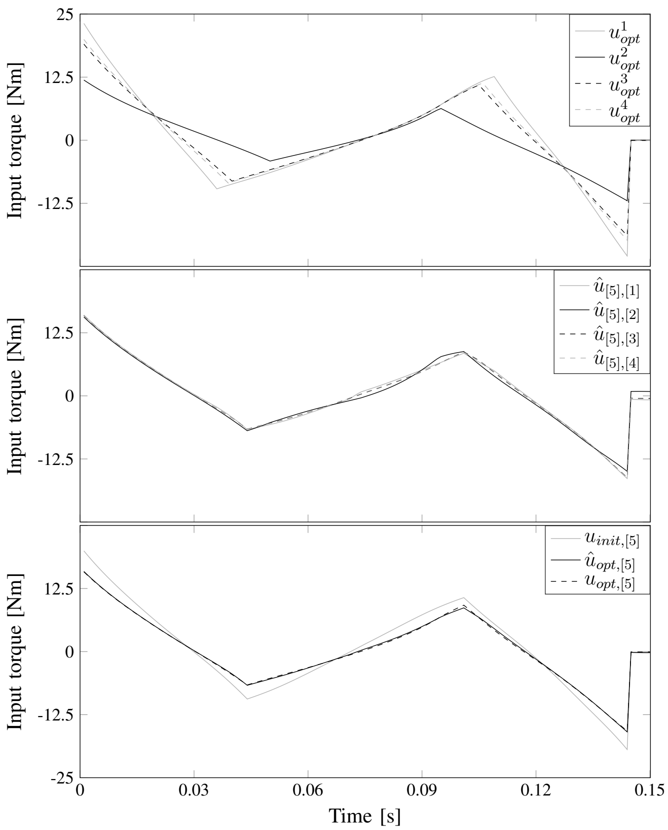

Fig. 6: The optimal inputs (top) and the separate hot-starts (middle) from four past tasks; the initial and optimal input in comparison with the hot-start derived using Procedure 3 (bottom).

Employing (19) with $w_i = \|\mathbf{u}_{init,[5]} - \mathbf{u}_{init,[i]}\|^{-1}$, the hot-start $\hat{\mathbf{u}}_{opt,[5]}$ is obtained as shown on the bottom of Fig. 6. Due to the inclusion of multiple tasks, the quality of the hot-start is increased compared to the results shown in Fig. 5.

In a typical application setting where we have to learn consecutive tasks and in the case without hot-starts, as shown in Fig. 7, the ILC has to relearn upon switching between tasks, yielding chattering behavior in terms of energy consumption and constraint violations on the slider displacement

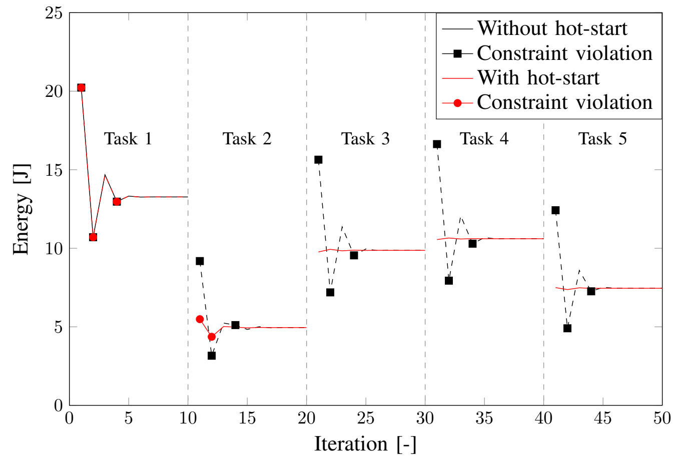

Fig. 7: Energy consumption for the considered tasks over iterations.

in numerous iterations. With the application of the proposed hot-starts, the chattering effect is drastically reduced, faster convergence is obtained and the number of constraint violations is lower. Finally, Fig. 8 shows a detailed view of the slider displacement in the first iteration of task 5. It is obvious that the displacement obtained without a hot-start yields a constraint violation, whereas the application of Procedure 3 reduces constraint violations significantly.

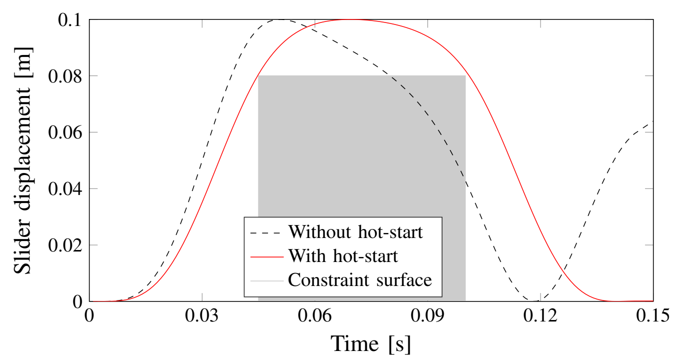

Fig. 8: Comparison of the slider displacement in the first iteration of task 5.

V. Experimental Validation

In this section, the proposed approach of Procedure 3 will be experimentally validated on the setup shown in Fig. 9. The mechanism is driven by a motor that converts a rotary motion into a linear displacement—a motion conversion that often emerges in industrial applications. The challenging nonlinearities of this setup stem from kinematic and dynamic effects like dead points and discontinuities (e.g. due to Coulomb friction). As a basis, we consider the problem of Section IV-C, but with a task length of $N = 600$ samples, three past tasks $k_{\text{left}} = k_{\text{right}} \in \{210, 160, 190\}$ and the new task $k_{\text{left}} = k_{\text{right}} = 175$.

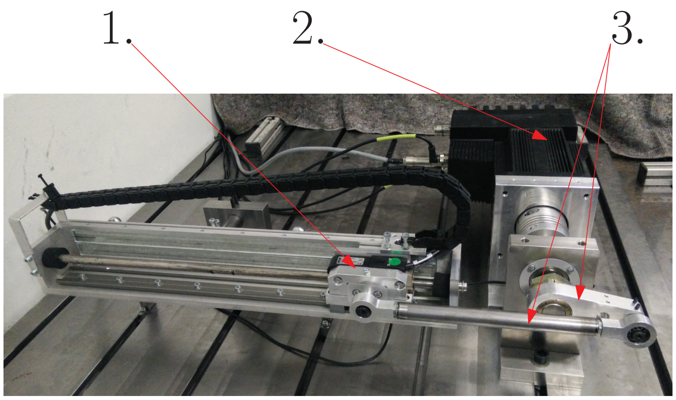

Fig. 9: The mechatronic setup with its components: Linear slider (1), rotary motor (2) and the crank (3).

In Fig. 10, the three optimal inputs $\mathbf{u}_{opt,[i]}$ for the past tasks are shown along with the hot-starts from each past task to the new task.

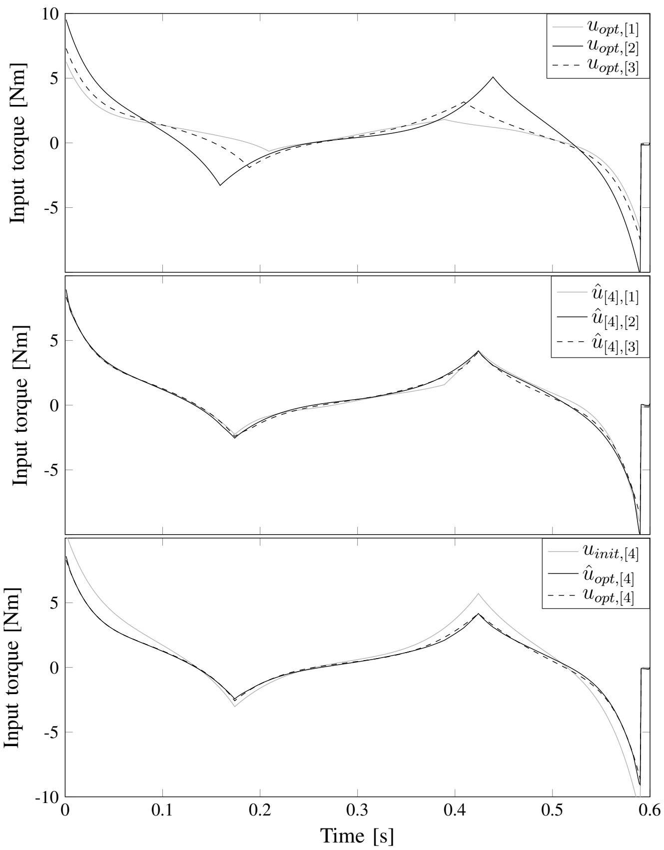

Fig. 10: The optimal inputs (top) and the separate hot-starts (middle) from three past tasks; the initial and optimal input in comparison with the hot-start derived using Procedure 3 (bottom).

Comparable conclusions to Section IV-D can be drawn, namely that the nonlinear dynamics of the slider-crank system are excited in a comparable way, despite the variation in tasks. The three separate hot-starts are combined to form the hot-start $\hat{\mathbf{u}}_{opt,[4]}$ using $w_i = \|\mathbf{u}_{init,[4]} - \mathbf{u}_{init,[i]}\|^{-1}$ in (19). The result is shown on the bottom of Fig. 10, where it can be seen that $\hat{\mathbf{u}}_{opt,[4]}$ is very close to the optimal result $\mathbf{u}_{opt,[4]}$, confirming that the proposed approach is also applicable on the experimental setup.

VI. Conclusions

In this paper, we have developed a transformation-based approach to initialize or hot-start similar tasks in ILC, on the basis of data obtained during previous tasks.

The goal is to minimize the total calibration time of an industrial machine, if multiple tasks have to be learned. This is done by extending the approach of [4] towards nonlinear systems and tasks that are not described by a predefined output trajectory, but involve more complex objectives and constraints, aligning it more with current industrial demands. Additionally, the approach is extended to include multiple past tasks that are weighted based on the similarity of the tasks. To show the effectiveness of the developed concepts they are validated on a nonlinear mechatronic slider-crank setup, both in simulation and experimentally.

Future research will be focussed on expanding the proposed procedure to tasks with an unequal number of samples and further analysis of its applicability and limitations, including an analysis on the required level of similarity between tasks.

Acknowledgment

This work has been carried out within the framework of Flanders Make's ICON project 'RoFaLC' (Robust and Fast Learning Control, IWT.150531) and SBO project 'MultiSysLeCo' (Multi-System Learning Control) funded by the agency Flanders Innovation & Entrepreneurship (VLAIO) and Flanders Make. Flanders Make is the Flemish strategic research centre for the manufacturing industry.

References

[1] K. L. Moore, *Iterative learning control for deterministic systems*. Springer Science & Business Media, 2012.
[2] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems," in *23rd IEEE Conference on Decision and Control*. IEEE, 1984, pp. 1064–1069.
[3] 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.
[4] P. Janssens, G. Pipeleers, and J. Swevers, "Initialization of ilc based on a previously learned trajectory," in *American Control Conference (ACC)*. IEEE, 2012, pp. 610–614.
[5] J. van de Wijdeven and O. H. Bosgra, "Using basis functions in iterative learning control: analysis and design theory," *International Journal of Control*, vol. 83, no. 4, pp. 661–675, 2010.
[6] J. Willems, E. Hostens, B. Depraetere, A. Steinhauser, and J. Swevers, "Learning control in practice: Novel paradigms for industrial applications," in *IEEE Conference on Control Technology and Applications (CCTA)*, 2018.
[7] M. Volckaert, J. Swevers, and M. Diehl, "A two step optimization based iterative learning control algorithm," in *ASME 2010 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2010, pp. 579–581.
[8] A. Steinhauser, T. D. Son, E. Hostens, and J. Swevers, "Rofalt: An optimization-based learning control tool for nonlinear systems," in *Advanced Motion Control (AMC), 2018 IEEE 15th International Workshop on*. IEEE, 2018, pp. 198–203.
[9] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*. ACM, 1968, pp. 517–524.