2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Combined time-domain optimization design for task-flexible and high performance ILC

Kentaro Tsurumoto[1], Wataru Ohnishi[1], Takafumi Koseki[1], Max van Haren[2], and Tom Oomen[2,3]

***Abstract*— Iterative learning control (ILC) yields substantial performance improvement for repetitive motion tasks. While task-flexibility for non-repetitive motion tasks can be achieved with the use of basis functions, this typically comes with a trade-off in performance or design parameters. This study aims to achieve both task-flexibility and high performance with a single time-domain optimization framework. By defining a criterion combining the cost for performance and task-flexibility, an optimal feedforward with task-flexibility of basis function ILC and high performance surpassing standard norm-optimal ILC is obtained. Numerical validation on a two-mass motion system confirm the capabilities of the developed framework.**

I. INTRODUCTION

Feedforward (FF) control is fundamental for attaining high-speed and superior performance in tracking control systems. While feedback (FB) control is essential for system stabilization and disturbance suppression, it alone cannot concurrently achieve high tracking performance and substantial noise rejection within identical bandwidth. Hence, in precision systems demanding high-speed and high-accurate control, including lithography systems [1], [2] and machine tools [3], the role of precise and highly reliable FF control predicated on an accurate system model becomes pivotal.

Iterative learning control (ILC) [4], [5], [6] is a data-based control method applied to improve the performance of systems with batch-wise repetitive operations. Distinguishing characteristics of ILC involve enhanced tracking-performance as well as analytical measurements for assuring learning convergence. For ILC to be industrially applicable, its framework is required to have:

- R1 Task-flexibly against non-repetitive motion tasks.
- R2 High tracking-performance.
- R3 Few design parameters.

A recognized design constraint of ILC is the trade-off between task-flexibility (R1) and tracking-performance (R2), as illustrated in Fig. 1 [7]. For instance, ILC frameworks such as frequency-domain ILC (FD-ILC) [8], [9] and norm optimal ILC (NO-ILC) [10], [11] can achieve perfect-tracking for systems only containing repetitive exogenous signals, whereas basis function ILC (BF-ILC) [12], [13] can maintain high tracking performance for operations with variations in the motion profile.

*This work was partially supported by JSPS KAKENHI 23H01431 and The Telecommunications Advancement Foundation. Additionally, this is part of the NWO research program VIDI project number 15698 and the ECSEL Joint Undertaking grant agreement 101007311 (IMOCO4.E). Furthermore, this has received funding from the joint JSPS-NWO funding program.

[1]The authors are with the Department of Electrical Engineering and Information Systems, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan k.tsurumoto@ctl.t.u-tokyo.ac.jp

[2]Max van Haren and Tom Oomen are with the Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[3]Tom Oomen is also with the Faculty of Mechanical, Maritime, and Materials Engineering, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands

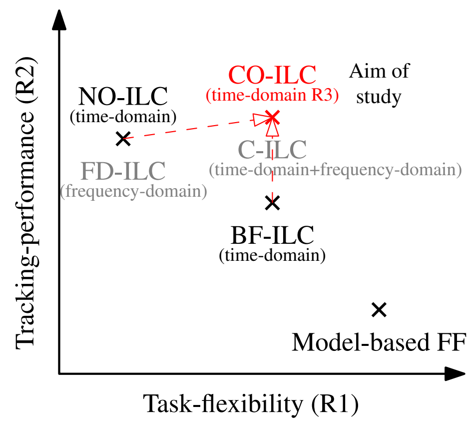

Fig. 1: Trade-off between tracking-performance and task-flexibility for ILC [7]. In this study, time-domain CO-ILC combining standard norm-optimal design (NO-ILC) and basis functions (BF-ILC) is investigated.

To circumvent this hurdle, important developments have been made to combine the task-flexibility of BF-ILC with the performance of FD-ILC. In [14], combined ILC (C-ILC) achieves task-flexibility as high as that of BF-ILC (R1) and superior tracking-performance than that of FD-ILC (R2), by combining basis functions with frequency-domain learning.

Although C-ILC attains both high task-flexibility (R1) and high tracking-performance (R2), it requires the user to design both frequency-domain learning filters and time-domain optimization parameters, not satisfying requirement R3. The aim of this study is to develop a single time-domain optimization framework satisfying requirement R3, while also achieving task-flexibility (R1) and high tracking-performance (R2). This is achieved by combining the criterion based on the optimization for BF-ILC and NO-ILC framework (Fig. 1).

The main contributions of this study are summarized as follows:

- C1 Combined optimal ILC (CO-ILC) is developed. The framework consists of the optimization of basis function FF input f^{BF} and residual FF input f^{NO}.
- C2 The presented CO-ILC framework is numerically validated with a two-mass motion system.

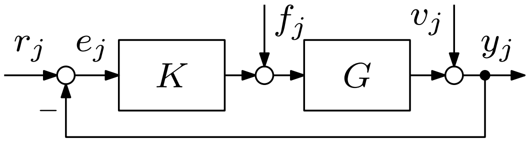

Fig. 2: Block diagram of closed-loop system. j denotes the iteration number of the motion task.

A. Notation

Let $\mathbf{H}(z)$ denote a discrete-time linear time invariant (LTI), single-input single-output (SISO) system. $\widehat{\mathbf{H}}(z)$ is a model of $\mathbf{H}(z)$.

Signal length are assumed to be $N \in \mathbb{N}$. Given input and output vectors $u, y \in \mathbb{R}^{N\times 1}$. Let $h(t)$ be the impulse response vector of $\mathbf{H}(z)$. The finite-time convolution matrix $H \in \mathbb{R}^{N\times N}$ corresponding to $\mathbf{H}(z)$ is expressed as follows:

$$\underbrace{\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} h(0) & h(-1) & \cdots & h(1-N) \\ h(1) & h(0) & \cdots & h(2-N) \\ \vdots & \vdots & \ddots & \vdots \\ h(N-1) & h(N-2) & \cdots & h(0) \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} u[0] \\ u[1] \\ \vdots \\ u[N-1] \end{bmatrix}}_{u}$$

assuming zero initial and final conditions.

Additionally, a weighted two norm for vector x is given by $\|x\|_W^2 = x^\top W x$, a positive definite matrix A is denoted as $A \succ 0$, and I_N is an $N \times N$ identity matrix.

II. PROBLEM FORMULATION

In this section the considered problem is defined by describing the system and introducing norm optimal design, basis function design of ILC, and combined ILC design.

A. System description

The control setup is shown in Fig. 2. Here system G and FB controller K are SISO and trial-invariant. Subscript j denotes the trial number of execution, with r_j denoting the references, f_j the feedforward signal, y_j the measured output, v_j an unknown zero-mean white Gaussian noise, and e_j the measured error given by:

$$e_j = Sr_j - Jf_j - Sv_j, \tag{1}$$

with sensitivity $S = (I + GK)^{-1}$ and process sensitivity $J = SG$. To facilitate the presentation, for section II and III, $r = r_{j+1} = r_j$ and $v_j = 0$ are assumed.

B. ILC frameworks

In this section, the three pre-existing ILC frameworks, 1) norm-optimal ILC, 2) basis function ILC and 3) combined ILC, that are considered in this study, are presented.

1) Norm optimal ILC (NO-ILC): The aim of NO-ILC is to determine the feedforward f_{j+1}, achieving perfect tracking control for trial-invariant reference, i.e., $r = r_{j+1} = r_j$. In NO-ILC, this is achieved by optimizing f_{j+1} based on the following criterion.

Definition 1 (Criterion for NO-ILC)**.** The performance criterion for NO-ILC is given by:

$$V(f_{j+1}) = \|\hat{e}_{j+1}\|_{W_e}^2 + \|f_{j+1}\|_{W_f}^2 + \|f_{j+1} - f_j\|_{W_{\Delta f}}^2, \tag{2}$$

where $W_e \succ 0$, $W_f \succeq 0$, and $W_{\Delta f} \succeq 0$ are user-defined weighting matrices. A common choice for the weighting matrices are $W = w\, I_N$, where w is a scalar.

Corresponding FF input update given by:

$$f_{j+1} = \arg\min_{f_{j+1}} V(f_{j+1}), \tag{3}$$

leads to the following optimal FF update for NO-ILC.

Lemma 1 (Optimal FF update for NO-ILC)**.** *Optimal FF update for NO-ILC is formulated as:*

$$f_{j+1} = Q^{\mathrm{NO}} f_j + L^{\mathrm{NO}} e_j, \tag{4}$$

where

$$\begin{aligned} R &= \widehat{J}^{\top} W_e \widehat{J} + W_f + W_{\Delta f}, \\ L^{\mathrm{NO}} &= R^{-1} \widehat{J}^{\top} W_e && \in \mathbb{R}^{N\times N}, \quad (5a) \\ Q^{\mathrm{NO}} &= R^{-1}(\widehat{J}^{\top} W_e \widehat{J} + W_{\Delta f}) && \in \mathbb{R}^{N\times N}. \quad (5b) \end{aligned}$$

2) Basis function ILC (BF-ILC): The aim of BF-ILC is to determine the feedforward f_{j+1}, achieving high tracking performance even for operations with varying references, i.e., $r_{j+1} \neq r_j$. While ILC methods like FD-ILC or NO-ILC can achieve perfect tracking for trial-invariant references, tracking performance can severely deteriorate for operations with non-repetitive references [7]. Basis function design overcome this problem by learning the parameters θ_{j+1} for the FF controller $F(\theta_{j+1})$, designed as:

$$f_{j+1} = F(\theta_{j+1}) r_{j+1}, \tag{6}$$

where FF controller $F(\theta)$ is defined as Definition 2 by user-defined basis functions $\Psi \in \mathbb{R}^{N\times N\times n}$, where n denotes the number of basis.

Definition 2 (Parameterized FF controller)**.** Given θ, the parameterized FF input is constructed by:

$$f = F(\theta) r = \Psi_r \theta, \tag{7}$$

where

$$\Psi_r = [\, \Psi[:,:,0]r, \Psi[:,:,1]r, \cdots, \Psi[:,:,n-1]r \,] \in \mathbb{R}^{n\times N}.$$

Similarly to NO-ILC, FF parameters θ_{j+1} are determined by minimizing a criterion defined as follows.

Definition 3 (Criterion for BF-ILC)**.** The performance criterion for BF-ILC is given by:

$$V(\theta_{j+1}) = \|\hat{e}_{j+1}\|_{W_e}^2, \tag{8}$$

where $W_e \succ 0$ is a user-defined weighting matrix.

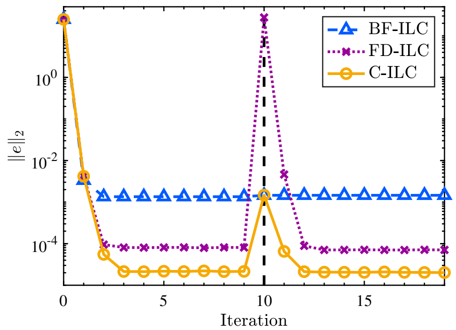

Fig. 3: Error norm per iteration for each ILC method. C-ILC (-o-) achieves performance exceeding FD-ILC (-×-) for repetitive tasks (j = 0-9, 11-19) while achieving same task-flexibility as BF-ILC (-△-) for non-repetitive tasks ($j = 10$).

3) Combined ILC (C-ILC): The aim of C-ILC is to determine the feedforward $f_{j+1} = f_{j+1}^{\mathrm{BF}} + f_{j+1}^{\mathrm{FD}}$, achieving 1) high task-flexibility as that of BF-ILC by f_{j+1}^{BF} and 2) high tracking-performance as that of FD-ILC by f_{j+1}^{FD} [14]. This is achieved by combining the use of basis functions with frequency-domain design of ILC.

To avoid interaction caused by simultaneous learning, performance criterion for basis function FF f_{j+1}^{BF} and residual FF update for f_{j+1}^{FD} are modified as follows.

Definition 4 (Criterion for f_{j+1}^{BF})**.** The performance criterion for C-ILC is given by:

$$V(\theta_{j+1}) = \|\hat{e}_{j+1}^{\theta}\|_{We}^2, \tag{9}$$

where

$$e_j^{\theta} = e_j + J f_j^{\mathrm{FD}}. \tag{10}$$

and $W_e \succ 0$.

Definition 5 (Update of f_{j+1}^{FD})**.** Update of f_{j+1}^{FD} is given by:

$$f_{j+1}^{\mathrm{FD}} = \mathbf{Q}^{\mathrm{FD}}(z)(f_j^{\mathrm{FD}} + \mathbf{L}^{\mathrm{FD}}(z)e_j) + f_j^{\mathrm{BF}} - f_{j+1}^{\mathrm{BF}}, \tag{11}$$

where learning filter $\mathbf{L}^{\mathrm{FD}}(z)$ and robustness filter $\mathbf{Q}^{\mathrm{FD}}(z)$ are pre-designed by the user.

Fig. 3 demonstrates the result of modifications above. C-ILC achieves as high task-flexibility as that of BF-ILC, while potentially exceeding the tracking-performance than that of FD-ILC. See [14] for further information.

C. Problem description

The aforementioned ILC frameworks have several performance and design limitations summarized as follows.

- While NO-ILC achieves high tracking-performance for trial-invariant reference (R2) and only require few design parameters (R3), it is highly susceptible to non-repetitive variances between tasks.
- BF-ILC achieves task-flexibility against non-repetitive references (R1) with few design parameters (R3), but typically performs poor compared to NO-ILC or FD-ILC due to parameterization of $F(\theta)$ not being rich enough to represent the inverse system G^{-1}.

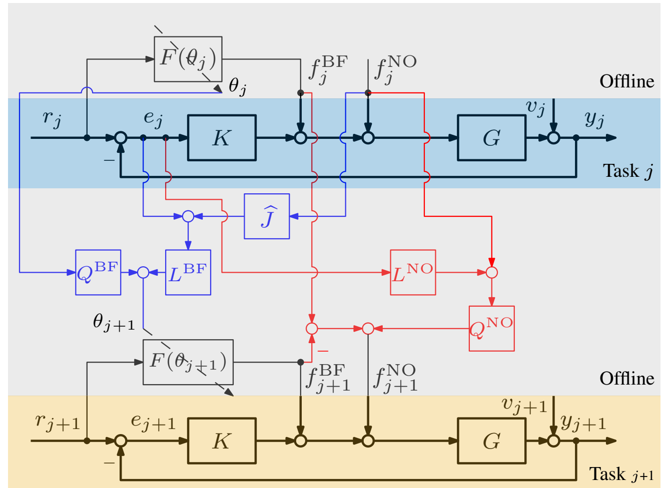

Fig. 4: Updating procedure of CO-ILC.

- C-ILC achieves both task-flexibility (R1) and high tracking-performance (R2), but requires the user to design frequency-domain filters $\mathbf{L}^{\mathrm{FD}}(z)$ and $\mathbf{Q}^{\mathrm{FD}}(z)$ in addition to standard BF-ILC design.

The problem addressed in this study is to develop an ILC framework that satisfies all of the requirements R1-R3 for industrial applicability of ILC, combining the advantages of pre-existing frameworks.

III. COMBINED OPTIMAL DESIGN

In this section, the proposed combined optimal ILC (CO-ILC) framework ($f = f^{\mathrm{NO}} + f^{\mathrm{BF}}$) is presented in twofold. First, the optimization criterion for deriving the residual FF input f^{NO} for R2 is discussed. Then, the optimization criterion is combined with the criterion of basis functions f^{BF} for R1, resulting to a single time-domain optimization problem (R3). The entire scheme is presented in Procedure 1 and illustrated as Fig. 4.

A. Norm optimal representation of residual FF update

In this study, optimizing the following criterion is proposed for the residual FF input f^{NO}.

Definition 6 (Criterion for f_{j+1}^{NO})**.** The performance criterion for f_{j+1}^{NO} is given as:

$$V(f_{j+1}^{\mathrm{NO}}) = \|\hat{e}_{j+1}\|_{W_e^{\mathrm{NO}}}^2 + \|f_{j+1}^{\mathrm{NO}} + f_{j+1}^{\mathrm{BF}} - f_j^{\mathrm{BF}}\|_{W_f^{\mathrm{NO}}}^2, \tag{12}$$

where $W_e^{\mathrm{NO}} \succ 0$, $W_f^{\mathrm{NO}} \succeq 0$ are user-defined weighting matrices.

Remark 1. As W_f is sufficient for assuring monotonic convergence, $W_{\Delta f}$ is omitted from the criterion to facilitate the presentation.

Corresponding FF input update given by:

$$f_{j+1}^{\mathrm{NO}} = \arg\min_{f_{j+1}^{\mathrm{NO}}} V(f_{j+1}^{\mathrm{NO}}), \tag{13}$$

leads to the following optimal FF update.

Procedure 1 (Update of proposed CO-ILC).

Design

- *Convolution matrix $\widehat{J}$ of process sensitivity model $\widehat{\mathbf{J}}(z)$*
- *Basis function Ψ_r*
- *Initial f_0^{BF} based on initial FF parameter θ_0*
- *Initial $f_0^{\mathrm{NO}} = 0$*
- *Weight W_e^{NO}, W_f^{NO}, W_e^{BF}*

and start with $j = 0$.

1) *Perform the jth experiment and measure e_j.*
2) *Determine θ_{j+1} based on (20).*
3) *Construct f_{j+1}^{BF} based on (7).*
4) *Construct f_{j+1}^{NO} based on (21).*
   a) *Reset $f_{j+1}^{\mathrm{NO}} = 0$ when $r_{j+1} \neq r_j$.*
5) *Increment $j \to j+1$ and return to 1).*

Theorem 1 (Optimal FF update for (12)). *Optimal FF update for (12) is formulated as:*

$$f_{j+1}^{\mathrm{NO}} = Q^{\mathrm{NO}} f_j^{\mathrm{NO}} + L^{\mathrm{NO}} e_j + f_j^{\mathrm{BF}} - f_{j+1}^{\mathrm{BF}}, \tag{14}$$

where

$$L^{\mathrm{NO}} = \left(\widehat{J}^{\top} W_e^{\mathrm{NO}} \widehat{J} + W_f^{\mathrm{NO}}\right)^{-1} \widehat{J}^{\top} W_e^{\mathrm{NO}} \in \mathbb{R}^{N\times N}, \tag{15a}$$

$$Q^{\mathrm{NO}} = L^{\mathrm{NO}} \widehat{J} \qquad \in \mathbb{R}^{N\times N}. \tag{15b}$$

Proof. The proof follows from [11], based on the optimal condition $\frac{\partial V(f_{j+1}^{\mathrm{NO}})}{\partial f_{j+1}^{\mathrm{NO}}} = 0$. □

Comparing (11) and (14), optimizing (12) leads to an identical residual FF update. Additionally, assuming $\widehat{J}$ invertible, (14) can be rearranged as:

$$f_{j+1}^{\mathrm{NO}} = Q^{\mathrm{NO}}(f_j^{\mathrm{NO}} + \widehat{J}^{-1} e_j) + f_j^{\mathrm{BF}} - f_{j+1}^{\mathrm{BF}}. \tag{16}$$

Learning filter $\mathbf{L}^{\mathrm{FD}}(z)$ is typically designed as an inverse of $\mathbf{J}(z)$ [15]. Therefore, optimizing (12) meets the design incentive of (11).

B. Combined optimal for task-flexibility and performance

For the developed CO-ILC framework, optimizing the following criterion combining (9) and (12) is proposed.

Definition 7 (Combined criterion for CO-ILC). Combined performance criterion for CO-ILC is given as:

$$V(f_{j+1}) = \|\hat{e}_{j+1}\|^2_{W_e^{\mathrm{NO}}} + \|f_{j+1}^{\mathrm{NO}} + f_{j+1}^{\mathrm{BF}} - f_j^{\mathrm{BF}}\|^2_{W_f^{\mathrm{NO}}} + \|\hat{e}_{j+1}^{\theta}\|^2_{W_e^{\mathrm{BF}}}, \tag{17}$$

where $W_e^{\mathrm{NO}} \succ 0$, $W_f^{\mathrm{NO}} \succeq 0$, $W_e^{\mathrm{BF}} \succ 0$ are user-defined weighting matrices.

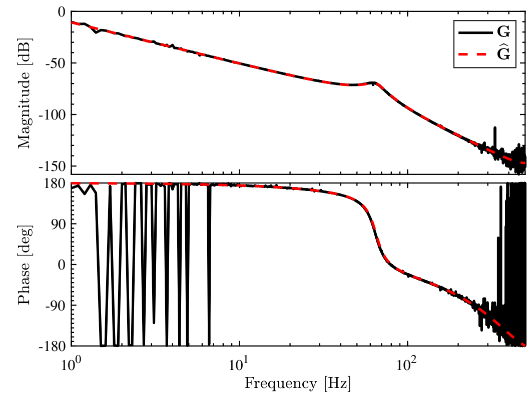

Fig. 5: Frequency response data $\mathbf{G}$ (—) and model $\widehat{\mathbf{G}}$ (- -) of the two-mass motion system for numerical validation.

Corresponding residual FF f_{j+1}^{NO} and basis function FF parameter θ_{j+1} update given by:

$$\theta_{j+1} = \arg\min_{\theta_{j+1}} V(f_{j+1}), \tag{18}$$

$$f_{j+1}^{\mathrm{NO}} = \arg\min_{f_{j+1}^{\mathrm{NO}}} V(f_{j+1}), \tag{19}$$

lead to the following optimal updates.

Theorem 2 (Optimal update for CO-ILC). *Optimal update for (17) is formulated as:*

$$\theta_{j+1} = Q^{\mathrm{BF}} \theta_j + L^{\mathrm{BF}}(e_j + \widehat{J} f_j^{\mathrm{NO}}), \tag{20}$$

$$f_{j+1}^{\mathrm{NO}} = Q^{\mathrm{NO}} f_j^{\mathrm{NO}} + L^{\mathrm{NO}} e_j + f_j^{\mathrm{BF}} - f_{j+1}^{\mathrm{BF}}, \tag{21}$$

where

$$L^{\mathrm{BF}} = \left(\Psi_r^{\top} \widehat{J}^{\top} W_e^{\mathrm{BF}} \widehat{J} \Psi_r\right)^{-1} \Psi_r^{\top} \widehat{J}^{\top} W_e^{\mathrm{BF}} \in \mathbb{R}^{n\times N}, \tag{22a}$$

$$Q^{\mathrm{BF}} = L^{\mathrm{BF}} \widehat{J} \Psi_r \qquad \in \mathbb{R}^{n\times n}. \tag{22b}$$

and Q^{NO}, L^{NO} same as (15).

Proof. The proof follows from [11], based on the optimal condition $\frac{\partial V(f_{j+1})}{\partial \theta_{j+1}} = 0$ and $\frac{\partial V(f_{j+1})}{\partial f_{j+1}^{\mathrm{NO}}} = 0$. □

Remark 2. Theorem 2 is implemented in a two step manner. By first updating θ_{j+1} based on (20) and constructing f_{j+1}^{BF}, update (21) for f_{j+1}^{NO} can be constructed.

The key contribution of CO-ILC is that the design parameters solely required for the update of residual FF update are only weight W_e^{NO} and W_f^{NO}. Typically, these weights are selected as $W_e^{\mathrm{NO}} = I_N$ and $W_f^{\mathrm{NO}} = \epsilon I_N$ with ϵ being a sufficiently tiny scalar value. Contrary to frequency-domain design (11) requiring the preparation of filters $\mathbf{L}^{\mathrm{FD}}(z)$ and $\mathbf{Q}^{\mathrm{FD}}(z)$, this leads to less design effort for the user, satisfying requirement R3.

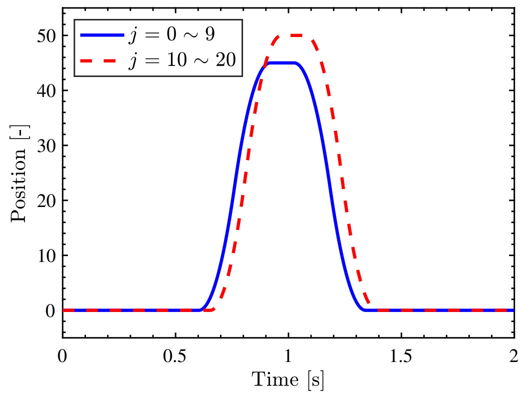

Fig. 6: Trajectory r_j used for validation. To validate both, high tracking-performance against repetitive tasks and high task-flexibility against non-repetitive tasks, (—) and (- -) are performed for the first and last 10 trials, respectively.

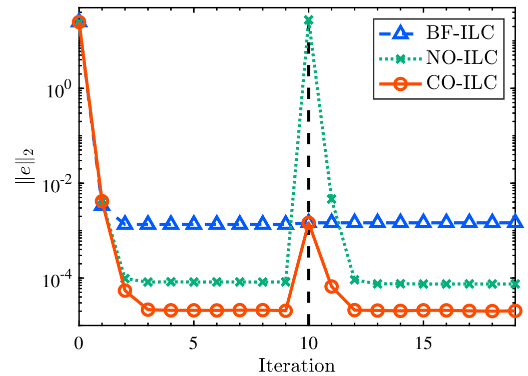

Fig. 7: Error norm per iteration for each learning method. The proposed CO-ILC (-⊖-) achieves as high task-flexibility as BF-ILC (-△-), while exceeding the performance of NO-ILC (-✱-) against repetitive tasks.

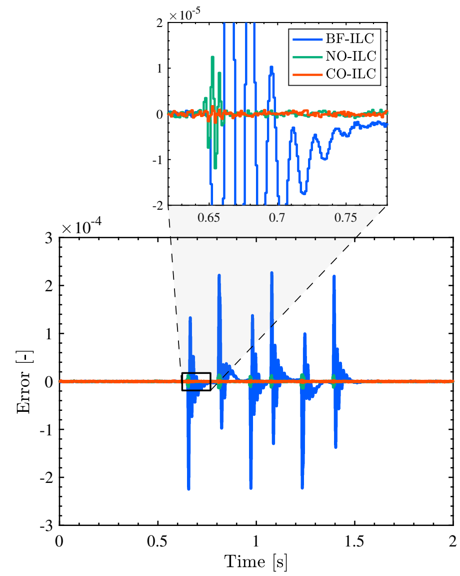

Fig. 8: Comparison of the tracking error of the final iteration. While CO-ILC (—) approaches near noise-level tracking error, BF-ILC (—) and NO-ILC (—) respectively suffer from near resonance and high frequency error.

IV. NUMERICAL VALIDATION

A. System Description

In this study, the developed method is validated with a two-mass motion system. The discrete-time system $\mathbf{G}$ is controlled at a sampling time of $T_s = 1\,\mathrm{ms}$. Transfer functions of the true system, system model, FB controller are given by:

$$\mathbf{G} = \frac{2.4244 \times 10^{-7}(z + 5.095)(z + 0.4373)(z - 0.1571)}{z(z-1)^2(z^2 - 1.761z + 0.9157)},$$
$$\widehat{\mathbf{G}} = \frac{2.4725 \times 10^{-7}(z + 5.093)(z + 0.4372)(z - 0.157)}{z(z-1)^2(z^2 - 1.757z + 0.9141)},$$
$$\mathbf{K} = \frac{108.61(z - 0.9606)(z + 1)}{(z^2 - 1.65z + 0.7035)},$$

with the bode plot of $\mathbf{G}$ and $\widehat{\mathbf{G}}$ illustrated as Fig. 5.

To simulate noise, white noise with a variance of 10^{-13} is injected as v_j.

B. Learning Setup

1) Design parameters: As stated in Procedure 1, design parameters required for learning are: process sensitivity model $\widehat{J}$, basis function Ψ_r, weight W_e^{NO}, W_f^{NO}, W_e^{BF}, and initial FF parameter θ_0. In this study, these parameters are accordingly defined as:

$$\widehat{J} = (I + \widehat{G}K)^{-1}\widehat{G}, \quad \Psi_r = [\ddot{r}, \dddot{r}, \ddddot{r}],$$
$$W_e^{\mathrm{NO}} = W_e^{\mathrm{BF}} = I_N, \quad W_f^{\mathrm{NO}} = 4 \times 10^{-13} I_N, \quad \theta_0 = 0.$$

are

2) Learning Task: The objective of this validation is to test both, tracking-performance against repetitive tasks and task-flexibility against non-repetitive tasks. To achieve this two reference trajectories shown in Fig. 6 are utilized. The solid line is used as the reference for the first 10 trials, while the dashed line is used as the reference for the last 10 trials.

C. Simulation Results

Fig. 7 depicts the norm comparison of the NO-ILC, BF-ILC, and CO-ILC. This result demonstrates that the presented CO-ILC offers high tracking-performance, surpassing that of NO-ILC, while retaining task-flexibility as that of BF-ILC.

Fig. 8 shows the tracking error comparison of the final iterations. Similarly to C-ILC [14] in Fig. 3, results show that CO-ILC exceeds the performance of NO-ILC owing to the cooperative learning facilitated by basis functions.

Fig. 9 presents the FF parameter learning comparison of BF-ILC and CO-ILC. Owing to the contribution of the regularization of virtual error e_j^θ in (17), CO-ILC is able to perform FF parameter learning identical to BF-ILC. This leads to a similar task flexibility capability, as illustrated in the 10^{th} iteration shown in Fig. 7.

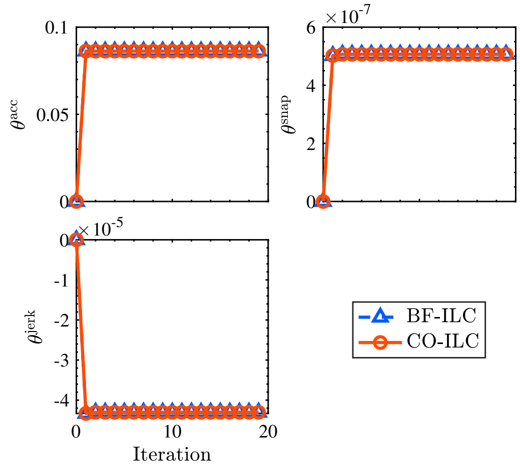

Fig. 9: FF parameter θ_j learned by the basis functions. Learning of CO-ILC (-⊖-) performs identical parameter learning overlapping that of the BF-ILC (-△-) framework.

Owing to CO-ILC learning identical FF parameter to that of BF-ILC, basis function FF input f^{BF} consists the main component of C-ILC input, as shown in Fig. 10. However, these selected basis functions are unable to learn system zeros. This leads to an error signal containing the resonance frequency displayed in Fig. 8. By the residual FF input f^{NO} compensating the residual dynamics uncaptured by the basis functions, CO-ILC performs high tracking-performance far exceeding that of BF-ILC, potentially higher than that of NO-ILC.

V. CONCLUSION

In this study, a time-domain CO-ILC framework achieving task-flexibly with high performance is introduced. By combining criterions based on BF-ILC; to ensure task-flexibility, and NO-ILC; to achieve high performance, the framework is truncated to a single time-domain optimization problem. This leads to the framework only requiring limited design parameters. Numerical validation with a two-mass motion system verify the proposed CO-ILC framework inherits improved task-flexibility as that of BF-ILC, while achieving higher tracking-performance than that of NO-ILC against repetitive tasks, with less design parameters than that of C-ILC.

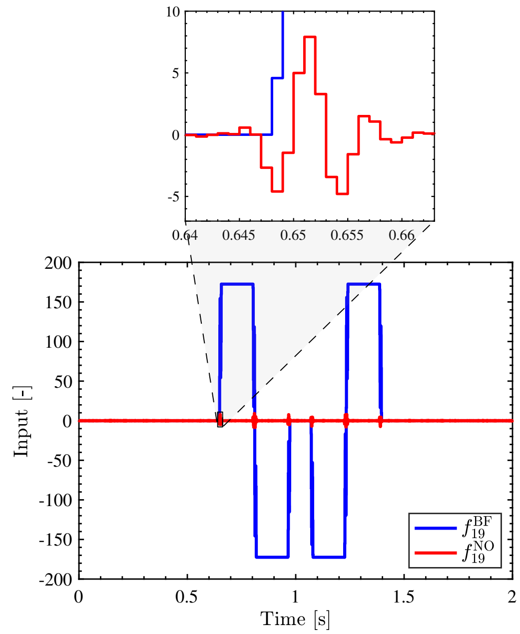

Fig. 10: Comparison of each learned FF input for the proposed CO-ILC framework. The input is mainly composed of the basis function FF input f_{19}^{BF} (—), while the residual FF input f_{19}^{NO} (—) compensates residual dynamics uncaptured by the basis functions.

REFERENCES

[1] S. Mishra, J. Coaplen, and M. Tomizuka, "Precision Positioning of Wafer Scanners Segmented Iterative Learning Control for Nonrepetitive Disturbances," *IEEE Control Systems*, vol. 27, no. 4, pp. 20–25, 2007.

[2] S. H. Van Der Meulen, R. L. Tousain, and O. H. Bosgra, "Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 130, no. 5, pp. 1–16, sep 2008.

[3] T. Hayashi, H. Fujimoto, Y. Isaoka, and Y. Terada, "Projection-based iterative learning control for ball-screw-driven stage with consideration of rolling friction compensation," *IEEJ Journal of Industry Applications*, vol. 9, no. 2, pp. 132–139, 2020.

[4] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems," in *The 23rd IEEE Conference on Decision and Control*, 1984, pp. 1064–1069.

[5] A. Bristow, Douglas, M. Tharayil, and A. G. Andrew, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[6] K. L. Moore, *Iterative learning control for deterministic systems*. Springer London, 2012.

[7] L. Blanken, F. Boeren, D. Bruijnen, and T. Oomen, "Batch-To-Batch Rational Feedforward Control: From Iterative Learning to Identification Approaches, with Application to a Wafer Stage," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 826–837, apr 2017.

[8] M. Norrlöf and S. Gunnarsson, "Time and frequency domain convergence properties in iterative learning control," *International Journal of Control*, vol. 75, no. 14, pp. 1114–1126, 2002.

[9] T. Oomen, "Learning for Advanced Motion Control," in *International Workshop on Advanced Motion Control*. Institute of Electrical and Electronics Engineers (IEEE), nov 2020, pp. 65–72.

[10] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control using optimal feedback and feedforward actions," *International Journal of Control*, vol. 65, no. 2, pp. 277–293, 1996.

[11] S. Gunnarsson and M. Norrlöf, "On the design of ILC algorithms using optimization," *Automatica*, vol. 37, pp. 2011–2016, 2001.

[12] 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.

[13] J. Bolder and T. Oomen, "Rational basis functions in iterative learning control - With experimental verification on a motion system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 722–729, mar 2015.

[14] K. Tsurumoto, W. Ohnishi, and T. Koseki, "Task flexible and high performance ILC: Preliminary analysis of combining a basis function and frequency-domain approach," in *IFAC 22nd Triennial World Congress*, 2023.

[15] J. Van Zundert and T. Oomen, "On inversion-based approaches for feedforward and ILC," *Mechatronics*, vol. 50, no. October 2017, pp. 282–291, 2018.