State-tracking iterative learning control in frequency domain design for improved intersample behavior

Ohnishi, Wataru; Strijbosch, Nard; Oomen, Tom

DOI
10.1002/rnc.6511

Publication date
2022

Document Version
Final published version

Published in
International Journal of Robust and Nonlinear Control

Citation (APA)
Ohnishi, W., Strijbosch, N., & Oomen, T. (2022). State-tracking iterative learning control in frequency domain design for improved intersample behavior. *International Journal of Robust and Nonlinear Control*, *33*(7), 4009-4027. https://doi.org/10.1002/rnc.6511

Received: 15 December 2021 | Revised: 25 June 2022 | Accepted: 22 August 2022

DOI: 10.1002/rnc.6511

RESEARCH ARTICLE

WILEY

State-tracking iterative learning control in frequency domain design for improved intersample behavior

Wataru Ohnishi[1] | **Nard Strijbosch**[2] | **Tom Oomen**[2,3]

[1]Department of Electrical Engineering and Information Systems, The University of Tokyo, Tokyo, Japan

[2]Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands

[3]Faculty of Mechanical, Maritime, and Materials Engineering, Delft University of Technology, Eindhoven, The Netherlands

Correspondence
Wataru Ohnishi, Department of Electrical Engineering and Information Systems, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.
Email: ohnishi@ieee.org

Funding information
Japan Society for the Promotion of Science, Grant/Award Number: 21K01012; Netherlands Organisation for Scientific Research, Grant/Award Number: 15698

Abstract

Iterative learning control (ILC) yields perfect output-tracking performance at sampling instances for systems that perform repetitive tasks. The aim of this article is to develop a framework for a state-tracking ILC that mitigates oscillatory intersample behavior, which is often encountered in output tracking ILC. As a framework for the analysis, the stability of the iterative domain including the robustness filter and the asymptotic signal is formulated. In addition, as a framework for the design, the design method using frequency response data to reduce the modeling effort, the learning filter design based on inversion, and the specific design procedure of the robustness filter are presented. The designed method is successfully applied to a motion system and it is shown that the presented state-tracking ILC provides better intersample behavior than the standard output-tracking ILC.

KEYWORDS

frequency domain design, intersample behavior, iterative learning control, multirate inversion, system inversion

1 | INTRODUCTION

Iterative learning control (ILC) can result in a significant improvement in the control performance of systems that perform repetitive tasks. The tracking error after each trial can be reduced by updating the control input for the next trial utilizing the error obtained in the previous trial. Consequently, it has been extensively applied to precision machinery, such as semiconductor lithography systems,[1,2] machine tools,[3] industrial printers,[4] mechatronic imaging systems,[5] and industrial robots.[6]

Controllers including ILC are usually implemented as a discrete-time controller with sample and hold device. Therefore, the design of ILC also often focuses on on-sample performance. The theory of on-sample behavior of discrete-time systems, including the monotonic convergence condition for the 2-norm of on-sample tracking performance, has witnessed extensive development. On the other hand, the control performance itself is evaluated in the continuous time since mechatronic systems evolve in continuous time. High tracking performance on the sampling instances does not necessarily mean high-tracking performance between the sampling instances.

Frequency domain design is an intuitive approach to achieve fast convergence and robust convergence.[7] It offers the advantage of using frequency response function (FRF) measurements to determine the monotonic convergence condition, thereby reducing modeling effort required. The analysis of frequency domain design motivates the design of learning filters as exact inverse models of the system to achieve fast convergence and high asymptotic performance. Because this approach usually focuses on tracking the output rather than the state variables of the controlled system, it is referred to as an output-tracking ILC in this article.

The controlled systems with zero-order hold discretization often have unstable zeros or zero near the stability limit,[8,9] which causes problems when attempting to design an inverse system as a learning filter. One design approach is to exploit an approximate inverse of the system such as zero-phase-error tracking control[10] at the cost of tracking performance; while another approach is applying stable inversion with preactuation,[11] which can achieve on-sample perfect tracking at the cost of intersample behavior.

Poor intersample behavior is not desirable because the control performance itself is often evaluated in a continuous time as the functions of mechatronic systems, such as lithography, printing, and milling, evolve continuously. Consequently, to improve intersample behavior multirate ILC[12] based on optimization has been developed; however, it requires a measurement frequency that is faster than the control frequency. In contrast, in this article, the control and measurement frequencies are assumed to be the same.

Although the output-tracking ILC with stable inversion can achieve high tracking performance on-sample, poor intersample behavior is often observed due to cancelation of discretization zero near the stability limit. The aim of this article is to present a concept of *state-tracking* ILC including its stability and performance analysis and design guidelines to improve the intersample behavior. In contrast to the *output-tracking* ILC, the presented state-tracking ILC achieves output tracking as well as state tracking after convergence. Consequently, this state tracking enables improved intersample behavior similar to multirate feedforward control,[13,14] which is a feedforward control strategy aiming for perfect state tracking.

The main contribution of this article is a state tracking ILC framework that improves the intersample behavior. This is achieved through the following sub-contributions.

- C1 The state-tracking ILC framework based on linear periodically time-varying (LPTV) system analysis is presented. The key idea of this approach is to achieve perfect state tracking for every n samples to improve the intersample behavior.
- C2 Analysis tools for both stability and performance of the state tracking ILC have been developed in Section 3. To facilitate the use of FRF measurement data, the LPTV system is transformed into a multi-input multi-output (MIMO) linear time invariant (LTI) system by exploiting lifting techniques. This allows the stability of the ILC to be determined in the frequency domain.
- C3 Design guidelines that exploit FRF measurements for state tracking ILC are derived in Section 4. The presented guidelines include a learning filter that yields fast convergence and a robustness filter that guarantees monotonic convergence using a robustness filter. In addition, model variation can be easily taken into account because robustness filters can be designed while overlaying multiple frequency response data on a plot of stability conditions.
- C4 The effectiveness of the developed method is verified through experiments for both cases involving accurate and inaccurate models, requiring the use of a robustness filter in Section 5.

This article extends the preliminary results reported in Reference 15 coupled with an iteration domain stability analysis that includes a robustness filter, which is required in case the model is not sufficiently accurate. In addition, analogous to the design advantages of output tracking ILC, state-tracking ILC enabled the evaluation of the stability analysis in the iteration domain employing a nonparametric model, that is, FRF data. This advantage reduces the modeling effort of the user because the frequency response data are often inexpensive, accurate, and fast to obtain.

1.1 | Notations

Let $P(z)$ denote a discrete-time, LTI, single-input, single-output (SISO) system, expressed as

$$P(z) = C(zI - A)^{-1}B + D, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times 1}, C \in \mathbb{R}^{1\times n}$, and $D \in \mathbb{R}$.

Let $\mathcal{R}$ denotes the set of rational discrete time transfer matrices. The set of real-rational functions bounded on the unit circle is denoted as $\mathcal{RL}_\infty$. $\mathcal{RH}_\infty$ is the subset of $\mathcal{RL}_\infty$ analytic for $|z| > 1$.

Definition 1 (Discrete-time lifting). Let $u[k] \in \mathbb{R}$ denote a discrete-time signal that corresponds to $u(kT_s)$, where T_s denotes the sampling time and $k \in \mathbb{Z}$. Further, consider a lifted signal over $\tau \in \mathbb{N}$ samples denoted by $\underline{u}[l] = \mathcal{L}u[k]$ with

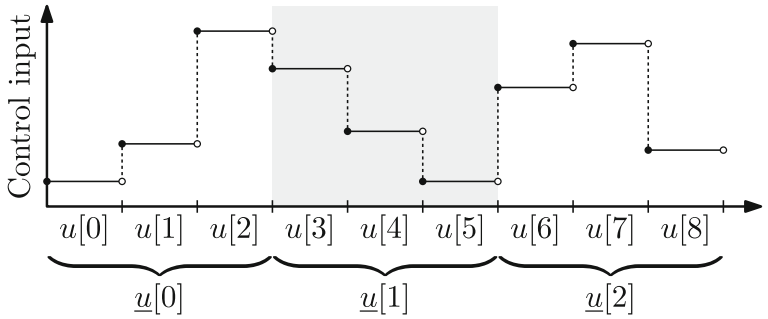

FIGURE 1 Time chart of discrete-time lifting for $\tau = 3$ as an example

$$\underline{u}[l] = \begin{bmatrix} u[l\tau] \\ u[l\tau+1] \\ \vdots \\ u[l\tau+\tau-1] \end{bmatrix} \in \mathbb{R}^{\tau}, \tag{2}$$

where $l \in \mathbb{Z}$ and $\mathcal{L}$ denote the lifting operator, which maps $u \mapsto \underline{u}$. A time chart is shown in Figure 1. An inverse lifting operator is expressed as $u = \mathcal{L}^{-1}\underline{u}$. For more details on the definition, see Reference 16. Note that lifting over τ sample multiplies the sampling period by $1/\tau$, and inverse lifting over τ sample multiplies the sampling period by τ.

Lemma 1 (Lifted system in state space). *The input/output of the lifted system of P over $\tau \in \mathbb{N}$ samples is $\underline{y} = \mathcal{L}y = (\mathcal{L}P\mathcal{L}^{-1})(\mathcal{L}u) = \underline{P}\underline{u}$. $\underline{P}$ can be obtained from a state-space model, and is denoted as*

$$\underline{P} \overset{z}{=} \left[\begin{array}{c|cccc} A^{\tau} & A^{\tau-1}B & A^{\tau-2}B & \cdots & B \\ \hline C & D & 0 & \cdots & 0 \\ CA & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ CA^{\tau-1} & CA^{\tau-2}B & CA^{\tau-3}B & \cdots & D, \end{array}\right] \in \mathcal{R}^{\tau\times\tau}. \tag{3}$$

Proof. This follows from successive substitution of (1).[16(Sect8.2)] ∎

Definition 2 (Downsampling operator). The downsampling operation over τ samples is defined by

$$\mathcal{S}_d : \alpha[k] \mapsto \beta[k], \quad \beta[k] = \alpha[\tau k], \quad k \in \mathbb{Z}. \tag{4}$$

Assumption 1 (Controlled continuous-time system G_c). A system $G_c(s) = C_{G_c}(sI - A_{G_c})^{-1}B_{G_c}$ to be controlled is a continuous time, SISO, strictly proper, LTI system given by minimal realization.

The discrete-time system G with G_c using a zero-order hold with sampling period T_s is denoted as

$$\begin{aligned} x[k+1] &= A_G x[k] + B_G u[k], \\ y[k] &= C_G x[k], \end{aligned} \tag{5}$$

where $x[k] \in \mathbb{R}^n$, $u[k], y[k] \in \mathbb{R}$. $x[k]$, $y[k]$, and $u[k]$ correspond to $x(kT_s)$, $y(kT_s)$, and $u(kT_s)$, respectively.

2 | PROBLEM FORMULATION

This section presents the formulation of the design requirements of the presented state-tracking ILC framework. First, in Section 2.1, the output-tracking ILC is briefly introduced. Subsequently, Section 2.2 introduces the multirate inversion

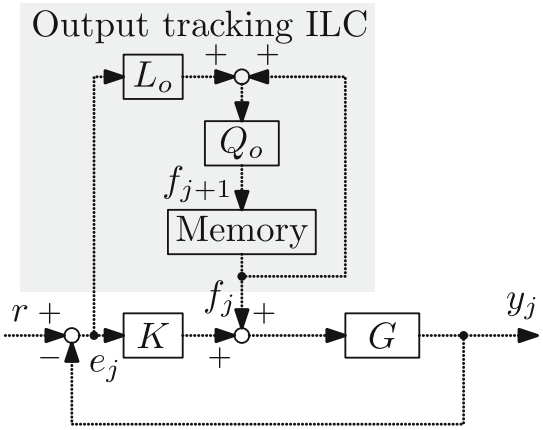

FIGURE 2 Block diagram of output tracking ILC

technique, which is a key to *state-tracking*. Further, Section 2.4 presents the design formulation requirements addressed in this article.

2.1 | Frequency domain design of output tracking ILC

The aim of the general ILC algorithm, including References 7, 17, and 18, is to achieve perfect output tracking at sampling instances for systems that perform repetitive tasks. It updates the feedforward input over iterations by learning from the tracking error obtained in the previous iteration. The design aspect offers the advantage of intuitive loop shaping for a robust design using nonparametric frequency-domain measurements.

The block diagram is shown in Figure 2. The closed-loop system is stabilized by a feedback controller $K \in \mathcal{RL}_\infty$. r denotes the trial-invariant reference, and $j \in \mathbb{Z}_{\geq 0}$ denotes the trial number. Tracking error e_j at jth trial expressed as

$$e_j = Sr - GSf_j, \tag{6}$$

where $S = (I + GK)^{-1}$ denotes the sensitivity function of the closed-loop system. The ILC command during iteration $j + 1$, f_{j+1}, is typically updated as

$$f_{j+1} = Q_o(f_j + L_o e_j), \tag{7}$$

where L_o and Q_o denote the learning filter $L_o \in \mathcal{RL}_\infty$ and the robustness filter $Q_o \in \mathcal{RL}_\infty$. In addition, noncausal operation is allowed for L_o and Q_o because the ILC update is computed offline.

The error propagation is expressed as

$$e_{j+1} = Q_o(1 - GSL_o)e_j + (1 - Q_o)Sr, \tag{8}$$

which converges monotonically in $\|e\|_2$ if

$$|Q_o(e^{i\omega})(1 - G(e^{i\omega})S(e^{i\omega})L_o(e^{i\omega}))| < 1, \quad \forall\omega. \tag{9}$$

An important aspect of the output-tracking ILC in frequency domain design is that the condition (9) can be verified using the measured frequency response data of G and S, which is fast, accurate, and inexpensive to obtain. This results in a reduction of the required design effort.

Using (8) and $e_\infty = \lim_{j\to\infty} e_j$, the asymptotic signals e_∞ are obtained as

$$e_\infty = \frac{1 - Q_o}{1 - Q_o(1 - GSL_o)}Sr. \tag{10}$$

Equations (8) and (10) motivate the design of $L_o = (GS)^{-1}$ to realize fast convergence and $Q_o = 1$ for $e_\infty = 0$, respectively. For an even relative order of a continuous plant model, a discretization zero close to $z = -1$ is generated[8] in G;

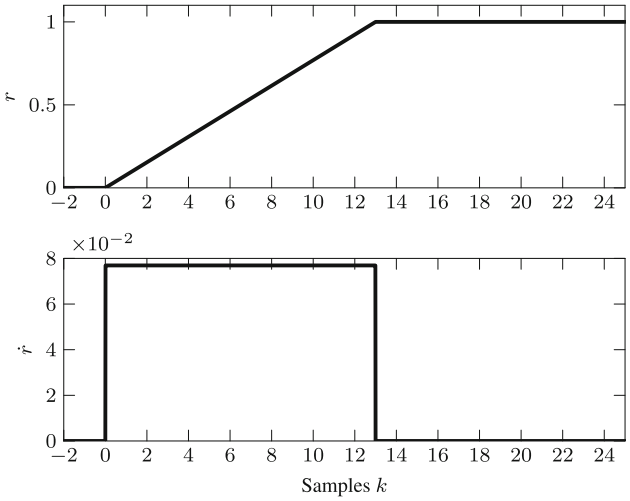

FIGURE 3 Reference r and its derivative signal $\dot{r}$

hence, GS has the same zero. Consequently, the force update (7) creates an oscillating control input with $L_o = (GS)^{-1}$, which results in poor intersample behavior, as demonstrated in Section 5.

To summarize, the output-tracking ILC possesses two primary favorable properties: (i) the availability of a trial domain stability condition (9) and (ii) limited design effort requirement using a nonparametric model. However, the primary disadvantage is that the focus on on-sample output tracking achieved by exploiting the inverse model results in poor intersample behavior. Thus, these characteristics form the motivation for the problem definition of the state-tracking ILC design in Section 2.4.

2.2 | Multirate inversion

Multirate inversion is a system inversion technique that achieves state tracking and possesses the potential to achieve improved intersample tracking performance. This section introduces a multirate feedforward control[13,14] using multirate inversion that achieves perfect *state* tracking for every n time instances in a nominal condition. By achieving both output and state tracking, it is possible to improve the intersample behavior, as demonstrated by the example below.

Example 1 (Single mass motion system). Consider a system $G_c = \frac{1}{s^2}$ with sampling time $T_s = 1$ and a continuous time reference, as shown in Figure 3. The discretized system based on zero-order hold is $G = \frac{0.5(z+1)}{(z-1)^2}$, with a zero at $z = -1$, which renders poor intersampling performance when pole-zero cancelation is performed via a single-rate feedforward controller.

Figure 4 demonstrates tracking error with single-rate and multirate feedforward controllers. To focus on the differences resulting from feedforward controllers, feedback controllers are not used. The state reference, state, and state tracking errors are expressed $r_x = [r, \dot{r}]^\top$, $x := [x_1, x_2]^\top = [y, \dot{y}]^\top$, and $e_x := [e_{x_1}, e_{x_2}]^\top = [r - y, \dot{r} - \dot{y}]^\top$, respectively. The single-rate feedforward control achieves perfect output tracking for every sample, whereas the multirate feedforward control can realize perfect *state* tracking for every $n = 2$ sample. Consequently, multirate feedforward control achieves superior intersample behavior at the cost of on-sample error for odd samples, for example, $k = 1, 13$.

For the above reasons, for single-rate feedforward control, reference trajectories with discontinuous velocities, such as the one shown in Figure 3, have been avoided because of the excitation of the discretization zero. On the other hand, multirate feedforward control can significantly reduce such intersample oscillations by providing higher-order state trajectory in addition to position trajectory.

2.3 | State tracking ILC setup

The key idea of the presented state-tracking ILC is to achieve perfect state tracking every n samples, reminicent to multirate feedforward control, with the aim to improve the intersample behavior, which constitutes Contribution

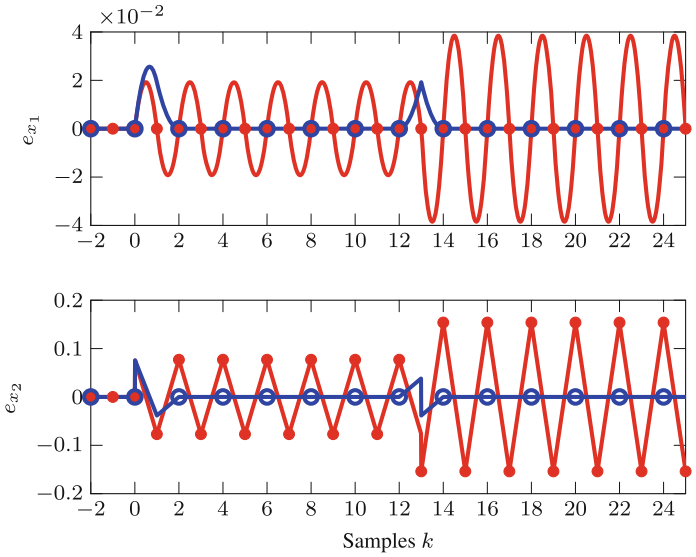

FIGURE 4 Tracking error of single-rate feedforward control (red solid line) and multirate feedforward control (blue solid line). Single-rate feedforward control achieves perfect output tracking for every samples (red filled circle). On the other hand, multirate feedforward control achieves perfect state tracking for every $n = 2$ samples (blue open circle) and it brings better intersampling performance.

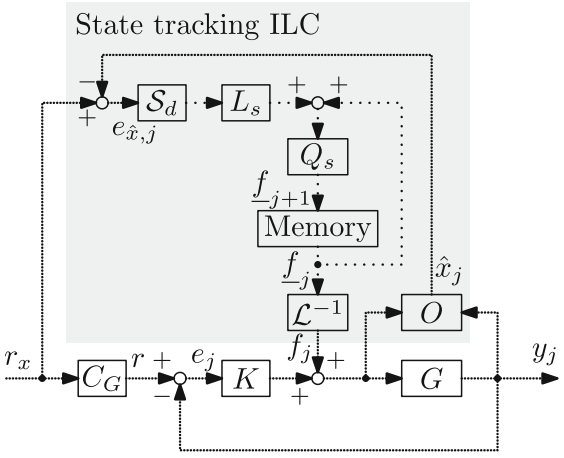

FIGURE 5 Block diagram of state tracking ILC. The high-frequency dots and low-frequency dots denote high-rate signal sampled by T_s and slow-rate signal sampled by nT_s, respectively.

C1. In this article, perfect state-tracking control is defined as an extension of perfect tracking control (PTC)[10,13] to state variables, where the state reference and the estimated state by the observer match at a specified sample period. In this definition, the effect of measurement noise, which is trial-varying disturbances, on the observer is not considered.

To this end, the ILC setup as depicted in Figure 5 is considered. Similar to the standard output-tracking ILC shown in Figure 2, the state-tracking ILC shown in Figure 5 comprises a memory, an update law, and a Q_s filter for robustness. There are two sampling rates in the signal: (i) T_s drawn as high-frequency dots is the high rate, which is the same as the rate of the control input and measurement, and (ii) nT_s drawn as low-frequency dots is the low rate in the lifted domain.

A key difference is that the ILC update law that consists of filter L_s and Q_s is performed in the lifted domain sampled with sampling time nT_s to achieve state-tracking, in contrast to the sampling time T_s in output-tracking ILC. The lines with dense dots and sparse dots denote signals sampled by T_s and nT_s, respectively. Hence, a downsampling operator $\mathcal{S}_d$ and an inverse lifting operator $\mathcal{L}^{-1}$ for n samples are required.

2.4 | Problem definition and contributions

The problem addressed in this article is the development of analysis tools (Contribution C2) and design guidelines (Contribution C3) for the presented state-tracking ILC (Contribution C1). Because of the multirate setting, the overall system is LPTV; therefore, standard frequency-based analysis and design techniques for output tracking ILC cannot be employed. The developed design guidelines exploit frequency response measurement data similar to well-known techniques for output-tracking ILC. In addition, this article addresses the verification of state-tracking ILC in experiments using measured frequency response data (Contribution C4).

3 | APPROACH AND ANALYSIS

In this section, a detailed state-tracking ILC approach is presented with the development of the analysis tools for stability and performance in the iteration domain. This section constitutes Contribution C2.

3.1 | Analysis in a single iteration

This section presents the derivation of the characteristics in a single iteration to analyze the performance and stability in the iterative domain presented in Section 3.2.

The reference r_x in Figure 5 denotes state trajectory to be tracked by the state-tracking ILC with the following assumption.

Assumption 2 (State trajectory r_x). State trajectory $r_x \in \mathbb{R}^n$ that satisfies

$$r[k] = C_G r_x[k], \quad \forall k, \tag{11}$$

is predetermined, where $r[k]$ denotes a reference trajectory for the output $y[k]$.

Remark 1. It is obtained by solving the continuous time state and output equations and results in the inversion of the continuous-time numerator polynomials of $G_c(s)$, see Reference 14.

The state-tracking error estimated by the state observer O is suppressed by the state-tracking ILC through iterations. The state tracking error estimate $e_{\hat{x},j}$ is obtained from the state trajectory r_x and the estimated state $\hat{x}_j$, as defined in next.

Definition 3 (State tracking error estimate). State tracking error estimate $e_{\hat{x},j} \in \mathbb{R}^n$ at jth iteration is obtained by

$$e_{\hat{x},j}[k] = r_x[k] - \hat{x}_j[k], \tag{12}$$

where $\hat{x}_j \in \mathbb{R}^n$ denotes state estimate at jth iteration by state observer $O \in \mathcal{RL}_\infty^{n\times 2}$

$$\hat{x}_j = O \begin{bmatrix} f \\ y_j \end{bmatrix}, \tag{13}$$

where f denotes the input of G, which consists of the ILC and feedback control contributions.

Remark 2. Noncausal operation is allowed for the state observer O, because the state estimation is performed by off-line calculation. In addition, a bounded state estimate is obtained through a back-and-forth filtering operation.[11]

The key idea of state-tracking ILC is to render the transfer characteristic between the lifted control input $\underline{f}_j$ at jth iteration and the estimate of the state-tracking error $e_{\hat{x},j}$ into a square system. It facilitates the design of the inverse system exploited in the learning algorithm in order to achieve perfect state-tracking. Therefore, n-sample lifting of the order of the plant is performed for the closed-loop system.

The lifted version of the force f_{j+1} over n sample as denoted by $\underline{f}_{j+1} \in \mathbb{R}^n$ is updated from jth state tracking error $e_{\hat{x},j}$ as defined in next.

Definition 4 (State-tracking ILC force update).

$$\underline{f}_{j+1} = Q_s \left(\underline{f}_j + L_s e_{\hat{x},j} \right), \tag{14}$$

where $Q_s \in \mathcal{RL}_\infty^{n\times n}$ and $L_s \in \mathcal{RL}_\infty^{n\times n}$ denotes a robustness filter and a learning filter, respectively. Apart from Equation (7), the robustness and learning filters are MIMO systems. With regard to Q_s and L_s, noncausal operations are allowed due to offline computations.

Next, Lemmas 2 and 3, and Definition 5, which are equations for the lifted domain, are prepared to derive the state-tracking error estimates in the lifted domain.

Lemma 2 (Lifted state reference $\underline{r}_x$ and lifted reference $\underline{r}$). *The lifted state reference $\underline{r}_x$ and lifted reference $\underline{r}$ satisfy*

$$\begin{aligned}\underline{r} &= (I_n \otimes C_G)\,\underline{r}_x \\ &= \underline{C_G r_x},\end{aligned} \tag{15}$$

where I_n and $\otimes$ denote the n-by-n identity matrix and the Kronecker tensor product, respectively.

Proof. It is obtained directly from the output equation $r = C_G r_x$ and Definition 1. ∎

Now, the state selection matrix $\mathcal{S}$ is defined to select the first-time instance from the lifted state tracking error estimate.

Definition 5 (State selection matrix $\mathcal{S}$). The first sample elements are selected from the lifted signal.

$$\mathcal{S} = \begin{bmatrix} I_n & O_{n\times n(n-1)} \end{bmatrix} \in \mathbb{R}^{n\times n^2}, \tag{16}$$

where $O_{n\times n(n-1)}$ denotes n-by-$n(n-1)$ matrix of zeros.

Lemma 3 (Relationship between lifting operator and state selection matrix). *Let $\bullet$ denote a signal with $n \times 1$ dimension and sampled by T_s. The following relationship holds.*

$$\mathcal{S}_d \bullet = \mathcal{S}\underline{\bullet} = \mathcal{S}\mathcal{L}\,\bullet\,. \tag{17}$$

Proof. This is obtained from Definitions 1 and 5. $\mathcal{L}\bullet$ is the result of lifting of $\bullet$ and is $n^2 \times 1$ signal sampled by T_s/n. Therefore, since $\mathcal{S}\mathcal{L}\bullet$ is the first n samples of $\mathcal{L}\bullet$ selected, it is an $n \times 1$ signal sampled by T_s/n. This is equivalent to $\mathcal{S}\underline{\bullet}$ and $\mathcal{S}_d\bullet$. ∎

From the aforementioned definitions and lemmas, a fundamental equation for the state-tracking ILC that relates state tracking error estimate, state trajectory, and ILC force in the lifted domain can be derived as follows.

Lemma 4 (State tracking error estimate $e_{\hat{x}}$ in lifted domain).

$$\mathcal{S}_d e_{\hat{x},j} = S_x \underline{r}_x - J_x \underline{f}_j, \tag{18}$$

where

$$S_x := \mathcal{S}\left(I_n - \underline{G_o K S C}\right) \in \mathcal{RL}_\infty^{n\times n^2}, \tag{19}$$

$$J_x := \mathcal{S}\underline{G_o}\mathcal{S} \in \mathcal{RL}_\infty^{n\times n}, \tag{20}$$

and $G_o \in \mathcal{RL}_\infty^{n\times 1}$ denotes a transfer function from the control input f to the state estimate $\hat{x}_j$ as follows:

$$\hat{x}_j = O\begin{bmatrix} f \\ y_j \end{bmatrix} = O\begin{bmatrix} f \\ Gf \end{bmatrix} = O\begin{bmatrix} I_n \\ G \end{bmatrix} f := G_o f. \tag{21}$$

Similar to Remark 2, a noncausal operation is allowed for G_o due to offline calculation.

Proof. State tracking error is composed of feedback and ILC contribution. According to Lemmas 2 and 3,

$$\mathcal{S}_d e_{\hat{x},j} = \mathcal{S}_d r_x - \mathcal{S}_d \hat{x}_j \tag{22}$$

$$= \mathcal{S}\underline{r}_x - \mathcal{S}\underline{\hat{x}}_j \tag{23}$$

$$= \mathcal{S}\underline{r}_x - \mathcal{S}\left(\underline{G_o S f}_{-j} + \underline{G_o K S C r}_{-x}\right) \tag{24}$$

$$= \mathcal{S}\left(I_n - \underline{G_o K S C}\right)\underline{r}_x - \mathcal{S}\underline{G_o S f}_{-j} \tag{25}$$

$$:= S_x \underline{r}_x - J_x \underline{f}_{-j}. \tag{26}$$

■

Remark 3. Equation (18) simplifies the block diagram in Figure 5 into the block diagram in Figure 6. In addition, (18) has the same structure as (6), where S_x can be interpreted as a sensitivity function from the lifted state reference to the estimated state tracking error, and J_x can be interpreted as a process sensitivity function from the lifted ILC input to the state estimate.

Definition 4 and Lemma 4 enable the stability analysis in the iteration domain.

3.2 | Stability analysis

This section derives the convergence condition of the state-tracking ILC in the iteration domain. To this end, state tracking error and ILC force propagation are formulated.

Theorem 1 (State tracking error and force propagation). *State tracking error propagation and ILC force propagation are formulated as*

$$S_d e_{\hat{x},j+1} = J_x Q_s \left(I_n - L_s J_x\right) z\bar{J}_x^{-1} S_d e_{\hat{x},j} + \left(I_n - J_x Q_s z\bar{J}_x^{-1}\right) S_x \underline{r}_x, \tag{27}$$

$$\underline{f}_{-j+1} = Q_s(I_n - L_s J_x)\underline{f}_{-j} + Q_s L_s S_x \underline{r}_x, \tag{28}$$

where $\bar{J}_x$ is invertible and $\bar{J}_x = zJ_x$.

The proof follows from Definition 4 and Lemma 4. A proof of the invertibility of $\bar{J}_x$ is presented in Section 4.2. Convergence is defined next.

Definition 6 (Convergence). System (28) is convergent if and only if for all $\underline{r}_x, \underline{f}_0 \in \ell_2$, there exists an asymptotic signal $\underline{f}_{-\infty} \in \ell_2$ such that

$$\limsup_{j\to\infty} \left\| \underline{f}_{-\infty} - \underline{f}_{-j} \right\|_{\ell_2} = 0. \tag{29}$$

A convergence condition of (27) and (28) is given next.

Theorem 2 (Convergence condition). *Iterations (27) and (28) converge if and only if*

$$\rho\left(Q_s(e^{i\omega})(I_n - L_s(e^{i\omega})J_x(e^{i\omega}))\right) < 1, \quad \forall \omega \in [0, \pi], \tag{30}$$

where $\rho(\cdot)$ denotes the spectral radius, that is, $\rho(\cdot) = \max_i |\lambda_i(\cdot)|$.

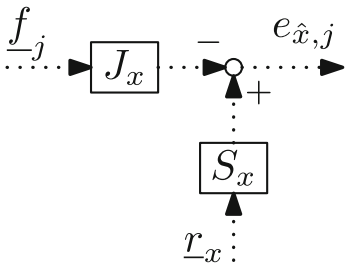

FIGURE 6 Simplified block diagram of state-tracking ILC in lifted domain

Remark 4. In contrast to Equation (9) for the output-tracking ILC that exhibits the SISO system, this equation for the state-tracking ILC exhibits an MIMO system. This indicates that the MIMO filter Q_s has more design flexibility than Q_o, which is a SISO filter. Its design methodology is presented in Section 4.1.

For proof, see Reference 19, Theorem 6. Monotonic convergence is defined as follows:

Definition 7 (Monotonic convergence). System (28) converges monotonically in the ℓ_2 norm from $\underline{f}_{-j}$ to $\underline{f}_{-\infty}$ with $0 \leq \gamma < 1$, if and only if

$$\left\| \underline{f}_{-\infty} - \underline{f}_{-j+1} \right\|_{\ell_2} \leq \gamma \left\| \underline{f}_{-\infty} - \underline{f}_{-j} \right\|_{\ell_2} \quad \forall j. \tag{31}$$

A monotonic convergence condition of (27) and (28) is provided as follows.

Theorem 3 (Monotonic convergence condition). *Iterations (27) and (28) converge monotonically in the ℓ_2 norm if and only if*

$$\gamma := \|Q_s(I - L_s J_x)\|_{\mathcal{L}_\infty} < 1. \tag{32}$$

Note that (32) is equivalent to

$$\overline{\sigma}\left(Q_s\left(e^{j\omega}\right)\left(I - L_s\left(e^{j\omega}\right)J_x\left(e^{j\omega}\right)\right)\right) < 1, \quad \forall \omega \in [0, \pi], \tag{33}$$

where $\overline{\sigma}$ denotes the maximum singular value.

A proof follows from Reference 20, Theorem 2.
From (27) and (28), the asymptotic signals $e_{\hat{x},\infty}$ and $\underline{f}_{-\infty}$ are obtained as

$$\begin{aligned} S_d e_{\hat{x},\infty} &= \lim_{j\to\infty} S_d e_{\hat{x},j} = \left(I - J_x(I - Q_s(I - L_s J_x))^{-1} Q_s L_s\right) S_x \underline{r}_x, \\ \underline{f}_{-\infty} &= \lim_{j\to\infty} \underline{f}_{-j} = (I - Q_s(I - L_s J_x))^{-1} Q_s L_s S_x \underline{r}_x. \end{aligned} \tag{34}$$

The requirement for Q_s to achieve the zero-asymptotic estimated state tracking error is as follows:

Theorem 4 (Requirement in Q_s for $e_{\hat{x},\infty} = 0$). *Assume $L_s(e^{i\omega}), J_x(e^{i\omega}) \neq 0, \forall \omega$. Given that (30) holds, then for all $r_x \in \ell_2, e_{\hat{x},\infty} = 0$, if and only if $Q_s = I_n$.*

For a proof, see Reference 21, Theorem 3.
To conclude, this section presents the state-tracking ILC framework and its convergence condition.

4 | DESIGN OF STATE-TRACKING ILC

This section presents design procedures of the robustness Q_s filter and a learning filter L_s by using nonparametric frequency response data. In addition, a method of using a non-parametric model for analysis is introduced to reduce the user's modeling effort. This section constitutes contribution C3.

4.1 | Robustness Q_s filter design for monotonic convergence

When the monotonic convergence condition shown in (33) is not satisfied by $Q_s = I_n$, it is necessary to design a Q_s filter, which is an MIMO filter. In this section, two designs of Q_s filters that guarantee monotonic convergence are presented: the first method is an intuitive design, but it exhibits conservative performance, whereas the second is a trial-and-error design that may exhibit high performance.

4.1.1 | Intuitive design

An intuitive design of Q_s is a direct design in the n-sample lifted domain, that is, the sampling time nT_s. Let $q_{s1} \in \mathcal{RL}_\infty$ and

$$Q_{s1} = q_{s1} I_n \in \mathcal{RL}_\infty^{n\times n}. \tag{35}$$

The advantage of this design is that the convergence condition (33) is simplified to

$$\overline{\sigma}\left(Q_{s1}\left(e^{j\omega}\right)\left(I_n - L_s\left(e^{j\omega}\right)J_x\left(e^{j\omega}\right)\right)\right) = \overline{\sigma}\left(q_{s1}\left(e^{j\omega}\right)\right)\overline{\sigma}\left(\left(I_n - L_s\left(e^{j\omega}\right)J_x\left(e^{j\omega}\right)\right)\right) < 1, \quad \forall\omega \in [0,\pi]. \tag{36}$$

Consequently, this allows the user to intuitively design Q_{s1} using the Bode plot, as in the case of the SISO output-tracking ILC.[22] However, it fails to exploit the directions in multivariable systems, which results in a conservative design.

4.1.2 | Less conservative design

Another Q_s design method involves designing a SISO filter in the original sampling time, that is, T_s, followed by the application of lifting. Let $q_{s2} \in \mathcal{RL}_\infty$ and

$$Q_{s2} = \mathcal{L}q_{s2}\mathcal{L}^{-1}. \tag{37}$$

In general, the directions can be effectively utilized, and a less conservative design may be created, as shown in (38).

$$\overline{\sigma}\left(Q_{s2}\left(e^{j\omega}\right)\left(I_n - L_s\left(e^{j\omega}\right)J_x\left(e^{j\omega}\right)\right)\right) \le \overline{\sigma}\left(Q_{s2}\left(e^{j\omega}\right)\right)\overline{\sigma}\left(\left(I_n - L_s\left(e^{j\omega}\right)J_x\left(e^{j\omega}\right)\right)\right). \tag{38}$$

Equations (36) and (38) indicate that Q_{s2} can have a higher bandwidth than Q_{s1}, leading to higher state-tracking performance according to (34).

4.2 | Guideline for the learning filter L_s design

This section presents a design guideline for learning filter L_s to realize fast convergence. Equation (27) motivates designing L_s as an inverse system of J_x. However, as J_x has no direct feedthrough according to Assumption 1, the inverse system of J_x is not feasible, and thus, a preview is required to achieve inversion. Theorem 5 proves that one sample preview in the lifted domain is necessary and sufficient to have direct feedthrough.

Theorem 5 (Direct feedthrough of zJ_x). *$\bar{J}_x$ is given as*

$$\bar{J}_x := zJ_x = z\mathcal{S}\underline{G_o}S, \tag{39}$$

where

$$J_x \overset{z}{:=} \left[\begin{array}{c|c} A_{J_x} & B_{J_x} \\ \hline C_{J_x} & O_{n\times n}\end{array}\right], \quad \text{hence} \quad zJ_x \overset{z}{=} \left[\begin{array}{c|c} A_{J_x} & B_{J_x} \\ \hline C_{J_x}A_{J_x} & C_{J_x}B_{J_x}\end{array}\right], \tag{40}$$

with full rank $C_{J_x}B_{J_x}$.

Proof. $C_G = I_n$ is assumed by an ideal observer without loss of generality. $\underline{G}$ is obtained by n sample lifting of G and formulated as

$$\underline{G} \overset{z}{=} \left[\begin{array}{c|cccc} A_G^n & A_G^{n-1}B_G & A_G^{n-2}B_G & \cdots & B_G \\ \hline I & 0 & 0 & \cdots & 0 \\ A_G & B_G & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ A_G^{n-1} & A_G^{n-2}B_G & A_G^{n-3}B_G & \cdots & 0 \end{array}\right]. \tag{41}$$

Next, the state selection matrix $\mathcal{S}$ selects the first-time instance, and thus $\mathcal{S}\underline{G}$ is obtained as

$$\mathcal{S}\underline{G} \overset{z}{=} \left[\begin{array}{c|cccc} A_G^n & A_G^{n-1}B_G & A_G^{n-2}B_G & \cdots & B_G \\ \hline I & 0 & 0 & \cdots & 0 \end{array}\right]. \tag{42}$$

Hence, with 1 sample preview, $z\mathcal{S}\underline{G}$ can be obtained as

$$z\mathcal{S}\underline{G} \overset{z}{=} \left[\begin{array}{c|cccc} A_G^n & A_G^{n-1}B_G & A_G^{n-2}B_G & \cdots & B_G \\ \hline A_G^n & A_G^{n-1}B_G & A_G^{n-2}B_G & \cdots & B_G \end{array}\right], \tag{43}$$

where $[A_G^n \quad A_G^{n-1}B_G \quad A_G^{n-2}B_G \quad \cdots \quad B_G]$ is full rank based on the controllable assumption in Assumption 1. Further, the sensitivity function S and its lifted system are defined as follows:

$$S := \overset{z}{=} \left[\begin{array}{c|c} A_S & B_S \\ \hline C_S & D_S \end{array}\right] \quad \text{and} \quad \underline{S} := \overset{z}{=} \left[\begin{array}{c|c} \underline{A}_S & \underline{B}_S \\ \hline \underline{C}_S & \underline{D}_S \end{array}\right], \tag{44}$$

$$\text{where} \quad \underline{D}_S = \begin{bmatrix} D_S & 0 & \cdots & 0 \\ C_S B_S & D_S & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ C_S A_S^{n-2} B_S & C_S A_S^{n-3} B_S & \cdots & D_S \end{bmatrix}, \tag{45}$$

where $\underline{D}_S$ is full rank as $D_S \neq 0$ because S represents a bi-proper system.

Moreover, considering the Sylvester's rank inequality, $C_{J_x}B_{J_x} = [A_G^n \quad A_G^{n-1}B_G \quad A_G^{n-2}B_G \quad \cdots \quad B_G]\underline{D}_S$ is determined to be full rank. ∎

After completing the design zJ_x with direct feedthrough, the learning filter L_s is designed as $L_s = z(zJ_x)^{-1}$. In the case zJ_x is a nonminimum phase system, stable inversion[11] can be exploited to obtain bounded signals.

4.3 | Limited design effort for user

State tracking ILC analyzes the LPTV system as explained in Section 3; however, the procedure presented in this section allows the measured frequency response data to be employed in the procedure, thereby reducing the effort required to design on part of the user. This is a key step that constitutes Contribution C3. Lifting the measured frequency response data, when combined with the analysis shown in Contribution C2, enables the design of robustness filters to be as intuitive as output-tracking ILC in frequency domain design.

The lifted frequency response data are obtained using Lemma 5.

Lemma 5 (Lifted system in FRF). *The lifted FRF matrix over n sample $\underline{P}(z) \in \mathbb{C}^{n\times n}$ is obtained as*

$$\underline{P}(z) = \begin{bmatrix} P^{(0)}(z) & z^{-1}P^{(n-1)}(z) & z^{-1}P^{(n-2)}(z) & \cdots & z^{-1}P^{(1)}(z) \\ P^{(1)}(z) & P^{(0)}(z) & z^{-1}P^{(n-1)}(z) & \cdots & z^{-1}P^{(2)}(z) \\ P^{(2)}(z) & P^{(1)}(z) & P^{(0)}(z) & \cdots & z^{-1}P^{(3)}(z) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P^{(n-1)}(z) & P^{(n-2)}(z) & P^{(n-3)}(z) & \cdots & P^{(0)}(z) \end{bmatrix}, \tag{46}$$

where

$$P^{(\sigma)}\left(z^n\right) = \frac{z^\sigma}{T}\sum_{k=0}^{T-1} P\left(z\phi^k\right)\phi^{k\sigma}, \quad \phi = e^{\frac{2\pi j}{n}}, \tag{47}$$

for $0 \leq \sigma \leq n-1$ and $\sigma \in \mathbb{Z}_{\geq 0}$.

For a proof, see Reference 23, Section 6.2.1.

In contrast to the lifting for a parametric model formulated in (3), Lemma 5 enables to compute a frequency response matrix of J_x defined by (20) using the measured frequency response data. Thus, the convergence condition (30) can be determined by the nonparametric frequency response data, with J_x calculated by (20) and the learning filter L_s calculated by (3).

The design procedure is summarized in Algorithm 1.

Algorithm 1. State-tracking ILC design

(a) Identify the frequency response data of G_{FRF} and obtain a parametric model G via system identification.
(b) Design a stabilizing feedback controller K and state observer O using G.
(c) Compute the parametric model G_oS and lift G_oS using (2).
(d) Compute the frequency response data $(G_oS)_{\mathrm{FRF}}$ using G_{FRF} and apply lifting by (46).
(e) Compute J_x using (20) and design the learning filter by $L_s = z(zJ_x)^{-1}$.
(f) Check the convergence condition using (30) with $Q_s = I_n$. If the convergence condition is not satisfied, satisfy (30) according to Section 4.1 by updating the model G or designing the robustness Q_s filter.

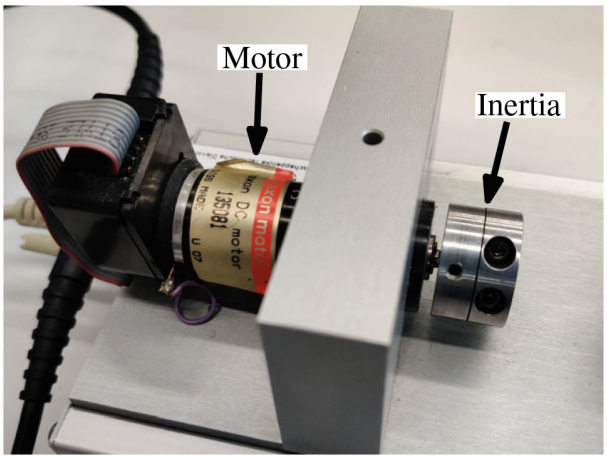

FIGURE 7 Motion setup: A single inertia system

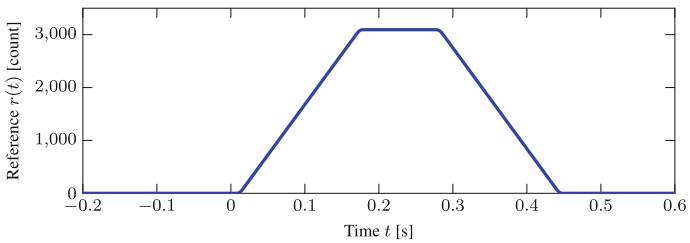

FIGURE 8 Motion reference $r(t)$ for the single iteration with a length of 800 ms

5 | EXPERIMENTAL VALIDATION

ILC experiments are conducted using the experimental setup shown in Figure 7 for the angle reference shown in Figure 8. The measured frequency response data and a model from input current reference to measured angle are shown in Figure 9. In this article, the third order is chosen, whereas the choice of the order of the plant model is a design parameter for the user. Higher-order models with higher accuracy reduce the gain of the convergence condition in (33), allowing for a higher frequency bandwidth for the robustness filter. This achieves state tracking for higher-order states at the expense of lengthening the state tracking interval. This section constitutes Contribution C4.

The sampling time for the controller is set as $T_s = 6$ ms, and intersample data are obtained by every 0.5 ms. This indicates that the output-tracking ILC aims to track outputs every 6 ms, and the state-tracking ILC aims to track states every 18 ms. Learning gain α is exploited to mitigate the amplification of the trial-varying disturbance[20] by replacing (7) and (14) as

$$f_{j+1} = Q_o(f_j + \alpha L_o e_j) \quad \text{and} \tag{48}$$

$$\underline{f}_{j+1} = Q_s(\underline{f}_j + \alpha L_s e_{\hat{x},j}), \tag{49}$$

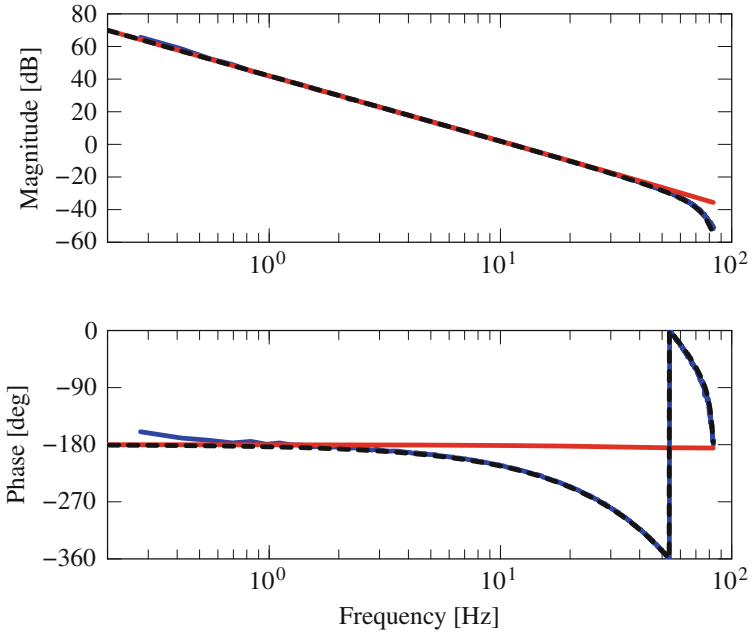

FIGURE 9 Frequency response data (blue solid line), well-identified continuous time model (red solid line), and discrete time model (black dashed line). Due to the accurate model, the monotonic convergence condition is satisfied without the design of the Q-filter, as shown in Figure 10.

respectively. The learning gain α is set to 0.3 in the following experiment. This value of 0.3 is set to accommodate quantization noise, which is trial-varying disturbances, in order to achieve tracking performance close to that of encoder quantization.

The experiments are conducted using two examples: one wherein the system is accurately identified such that monotonic convergence can be guaranteed with $Q_s = I_n$, and another wherein the Q_s filter has to be designed. A feedback controller that guarantees the stability of the closed-loop system is designed. In the following experiments, a proportional-differential (PD) controller with the gain crossover frequency 5.6 Hz, 30.1° phase margin, and 8.69 dB gain margin is used. The observer is designed using the third-order plant model with a Butterworth-type pole placement with 40 Hz bandwidth. The bandwidth of the observer is a design parameter determined by the accuracy of the model and the amount of measurement noise.

The following experimental results conclude that the presented state-tracking ILC outperforms the standard output-tracking ILC.

5.1 | Case 1: Performance with accurately identified model

This section compares the experimental results of the output and state tracking ILCs using the accurately identified model shown in Figure 9. Further, the differences in the characteristics of output and state tracking are clarified. Figure 10 shows that the monotonic convergence condition is satisfied for both the output-tracking ILC and the state-tracking ILC without designing the robustness filter, that is, $Q_o = 1, Q_s = I_n$. It follows that, by (10) and Theorem 4, perfect tracking and perfect state tracking can be achieved under the condition of no trial-varying disturbances, respectively.

Figure 11 shows the norm comparison of the two methods and further demonstrates that the presented state-tracking ILC exhibits better performance compared to the output-tracking ILC. Figure 12 shows the tracking error comparison at the best iterations. Although the output-tracking ILC shown in Figure 12A yields a small on-sample error, a large intersample tracking error is observed. The intersample behavior is excited at the timing of the change in acceleration of the reference trajectory shown in Figure 8. Since $L_o = (SG)^{-1}$ has an unstable pole due to discretization, stable inversion is applied, and thus the waveform of the intersample error propagates backward in time. In contrast, the state-tracking ILC shown in Figure 12B yields a small intersample tracking error. Although intersample error occurs at the timing of acceleration changes, unlike the output-tracking ILC, the errors are not propagated. In particular, a small tracking error at every $n = 3$ sample is observed because the state-tracking ILC aims to realize perfect state tracking at every n sampling instance.

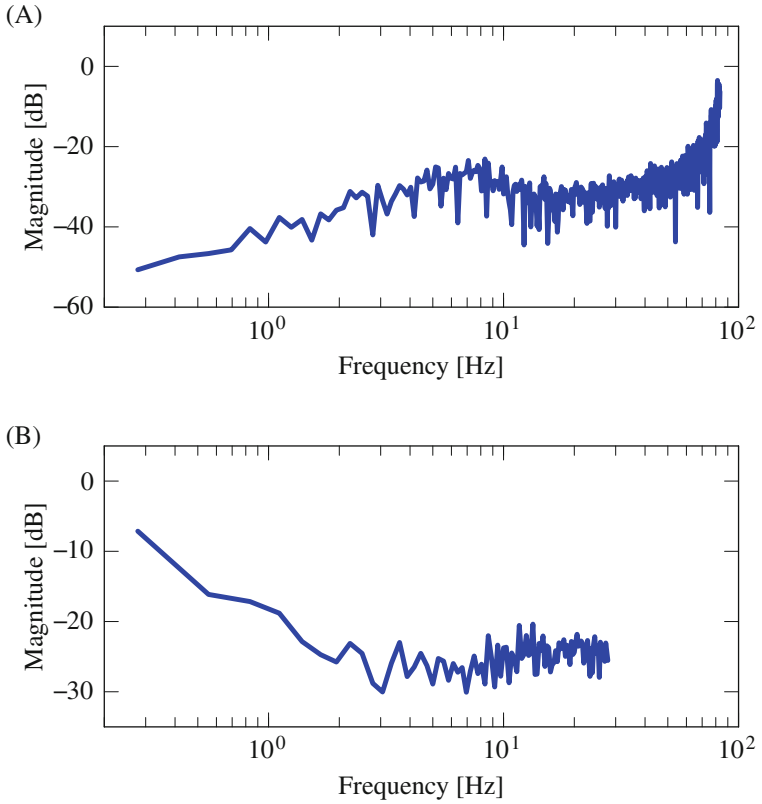

FIGURE 10 Convergence conditions calculated by (9) and (33), using the accurate model shown in Figure 9. Since the gain is less than 0 dB in all frequencies, both satisfy the monotonic convergence condition. (A) Output tracking ILC. (B) State tracking ILC

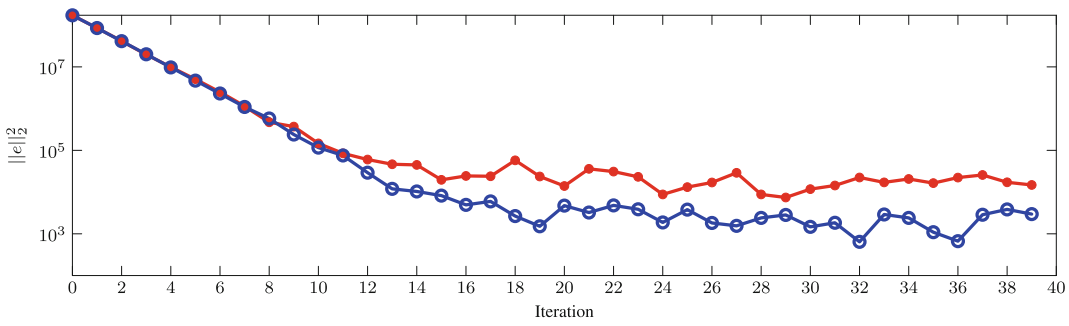

FIGURE 11 Experimental results with well-identified model. State tracking ILC (blue solid line with open circle) outperforms output tracking ILC (red solid line with filled circle). Comparing the best trials, the state-tracking ILC had 12 times better tracking performance.

5.2 | Case 2: Performance with inaccurately identified model

This section presents an experimental comparison of output and state tracking ILCs using the inaccurately identified model shown in Figure 13. Further, the difference in results owing to the robustness filter design method has been clarified.

Figure 14 indicates that in the absence of robustness filters the convergence condition is not satisfied. Therefore, to satisfy the monotonic convergence condition, robustness filters with the maximum bandwidth are designed for the three conditions of output and state tracking ILCs. Compared to the intuitive design Q_{s1}, the less conservative design Q_{s2} effectively utilizes the directions in multivariable system and thus satisfies the convergence condition with higher bandwidth robustness filter.

Figure 15 shows the norm comparison of the three methods, demonstrating that the presented state tracking ILC with less-conservative robustness filter design exhibits better performance compared to the output-tracking ILC. However,

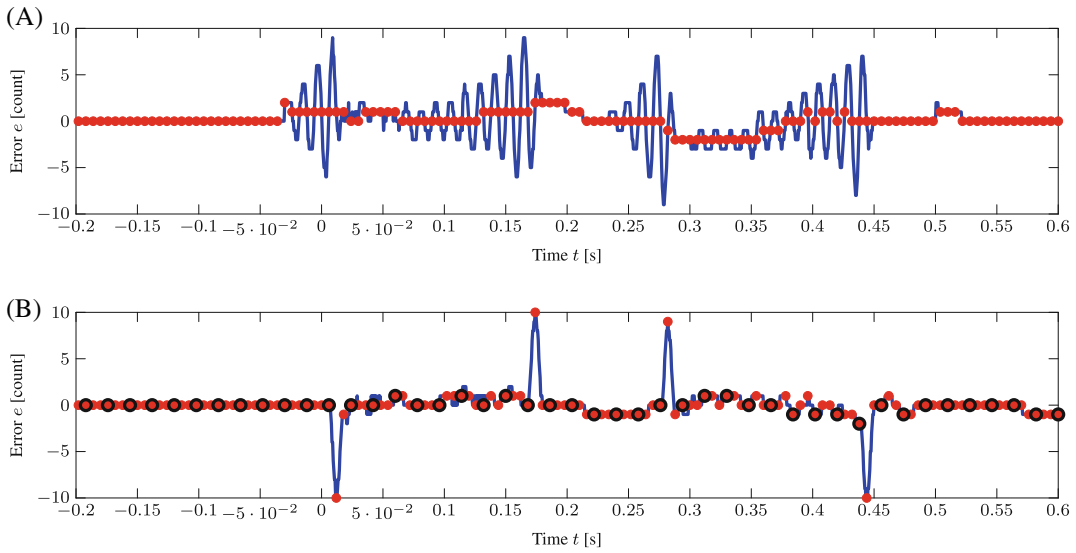

FIGURE 12 Tracking error comparison of the best iteration. The presented state-tracking ILC outperforms the output-tracking ILC in intersample behavior (blue solid line). The output-tracking ILC achieves a small tracking error at every T_s (red filled circle) at the cost of oscillating intersample behavior. The state-tracking ILC achieves a small tracking error at every nT_s (black open circle), and the oscillating intersample behavior is not observed. It is achieved by the concept of the state-tracking of the presented ILC. (A) Output tracking ILC at iteration 29. (B) State tracking ILC at iteration 32

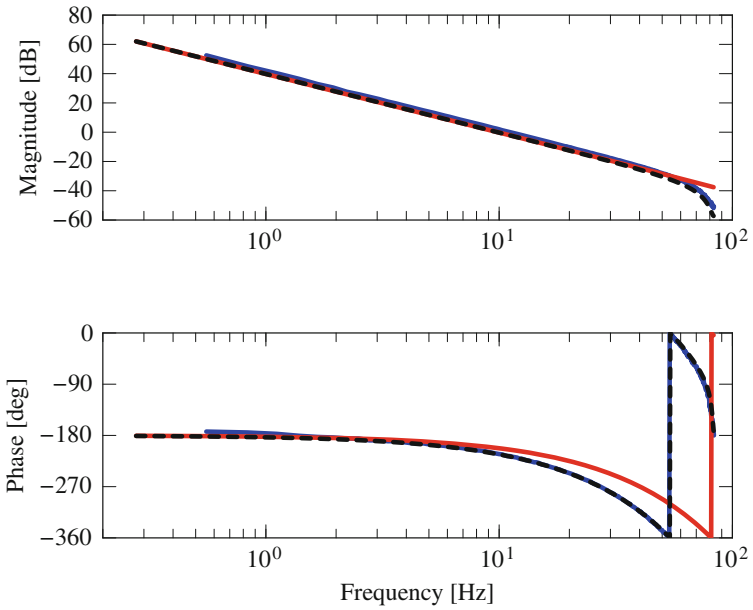

FIGURE 13 Frequency response data (blue solid line), inaccurately identified continuous time model (red solid line), and discrete time model (black dashed line). The offset in gain brings about the need for the Q_s filter shown in Figure 14.

owing to the robustness filter with lower bandwidths, the intuitive design method resulted in larger tracking errors than the others. This is because the robustness filter in intuitive design is designed directly in the lifted domain with a low sampling period.

Figure 16 shows the tracking error comparison at the best iterations. The output-tracking ILC in Figure 16A demonstrates a behavior similar to the $Q_o = 1$ condition shown in Section 5.1. Although the error on-sample is small, a large intersample tracking error is observed. Further, owing to the robustness filter with lower bandwidths, the intuitive design shown in Figure 16B yields a large tracking error. In addition, the less conservative robustness filter design

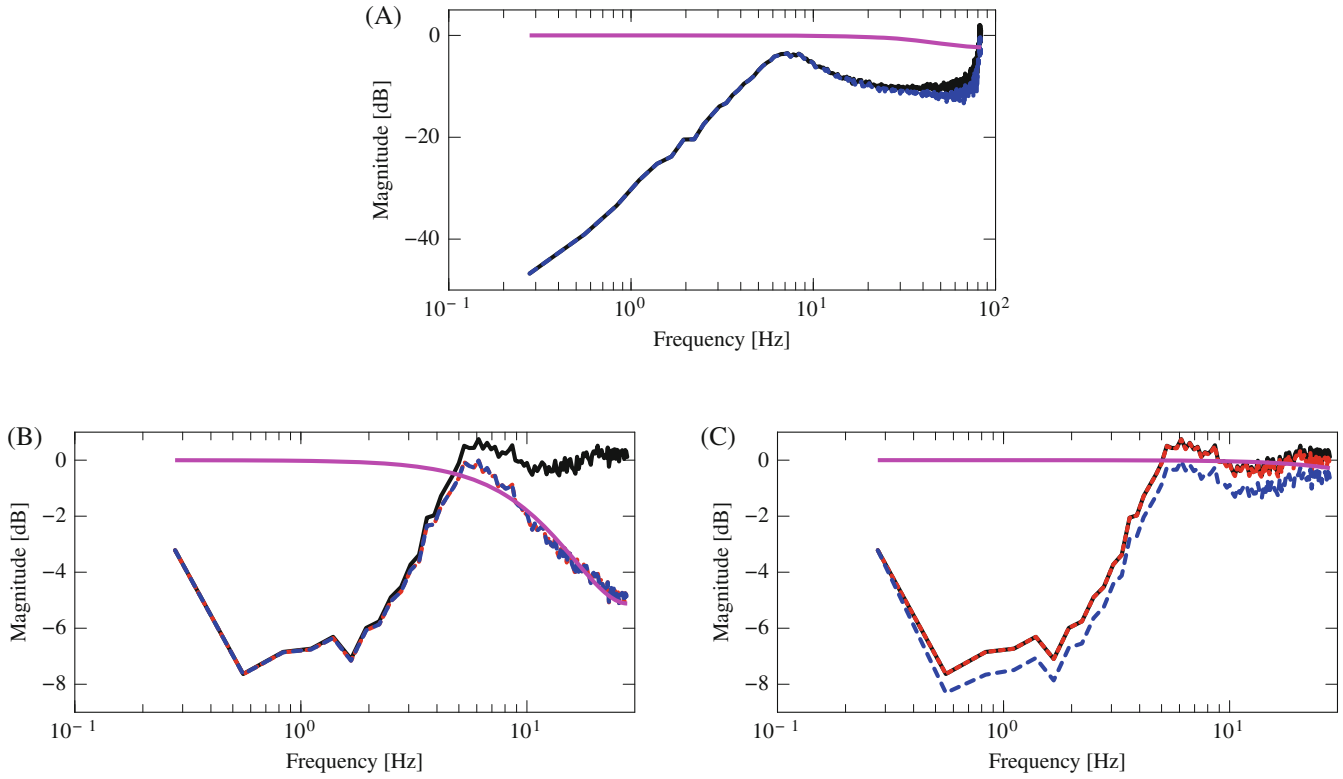

FIGURE 14 Convergence conditions with inaccurately identified model calculated by (9) and (33). The comparison between (B) and (C) shows that there is a trade-off between intuitive design and high bandwidth in state-tracking ILC. (A) Output tracking ILC. Whereas $1 - GSL_o$ (black solid line) does not satisfy the monotonic convergence condition, $Q_o(1 - GSL_o)$ (blue dashed line) does with the inclusion of Q_o (pink solid line). (B) State tracking ILC, intuitive design (Q_{s1}). Whereas $\overline{\sigma}(I - L_sJ_x)$ (black solid line) does not satisfy the monotonic convergence condition, $\overline{\sigma}(Q_{s1}(I - L_sJ_x))$ (blue dashed line) does with the inclusion of $\overline{\sigma}(Q_{s1})$ (pink solid line). Since $\overline{\sigma}(Q_{s1}(I - L_sJ_x))$ is equal to $\overline{\sigma}(Q_{s1})\overline{\sigma}(I - L_sJ_x)$ (red dashed line), it allows for intuitive Q-filter design. (C) State tracking ILC, less conservative design (Q_{s2}). Whereas $\overline{\sigma}(I - L_sJ_x)$ (black solid line) does not satisfy the monotonic convergence condition, (blue dashed line) does with the inclusion of Q_{s2} (pink solid line). By using directions in multivariable system, Q_{s2} can have a higher bandwidth than Q_{s1}, which leads to a higher tracking performance.

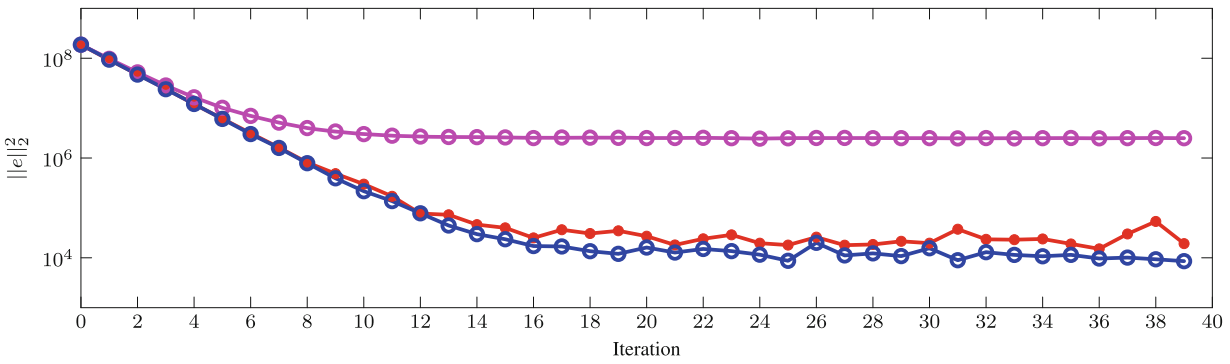

FIGURE 15 Experimental results with inaccurately identified model. State tracking ILC with less conservative Q_{s2} (blue solid line with open circle) outperforms output tracking ILC (red solid line with filled circle) and state tracking ILC with intuitive design Q_{s1} (pink solid line with open circle). In the comparison of the best trials, the state-tracking ILC with less conservative Q_{s2} has 1.8 times better tracking performance than the output tracking ILC.

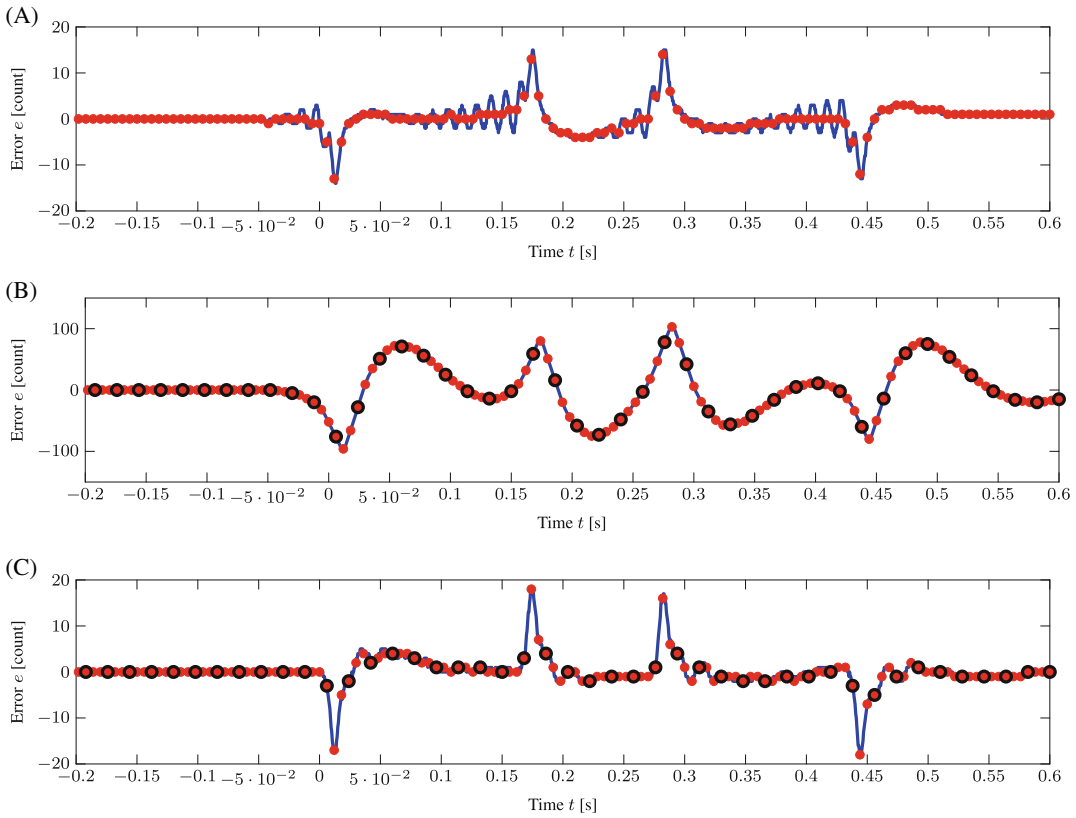

FIGURE 16 Tracking error comparison of the best iteration. The presented state-tracking ILC with less conservative Q_{s2} filter design outperforms the output-tracking ILC in intersample behavior (blue solid line). The tracking error for each sampling period T_s is shown by (red filled circle), and the tracking error for each $3T_s$ is shown by (black open circle). The intuitive Q_{s1} design resulted in low tracking performance due to the low bandwidth of the robustness filter. The trend in (A) and (C) is the same as in Figure 12: Oscillatory intersample behavior is observed for the output-tracking ILC, while the state-tracking ILC with less-conservative design shows an improved tracking performance. (A) Output tracking ILC at iteration 36 with Q_o. (B) State tracking ILC at iteration 24 with intuitively designed Q_{s1}. (C) State tracking ILC at iteration 39 with less conservative Q_{s2}

in state-tracking ILC, shown in Figure 16C, exhibits better performance than other design methods, including both on-sample and intersample tracking errors.

6 | CONCLUSION

The developed ILC framework enables the improvement in intersample behavior by incorporating the idea of state-tracking in design. Based on an LPTV stability analysis, convergence conditions are derived that guarantee convergence and monotonic convergence. The accurate and inexpensive nonparametric frequency response data can be exploited to verify the convergence conditions. Additionally, the design of a robustness filter that guarantees monotonic convergence conditions even with inaccurate models is presented, demonstrating the effectiveness of the proposed method in practical conditions. Application of the design framework to a motion system and the experimental validation demonstrate the superior tracking performance and improved intersample behavior over the traditional output-tracking ILC.

Ongoing research focuses on an integrated optimization of observer design and robustness filters. Future research focuses on applications to higher-order systems, including MIMO systems.

ACKNOWLEDGMENTS

This work is part of the research programme VIDI with project number 15698, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO), and is supported by JSPS KAKENHI Grant Number 21K01012.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest for this article.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Wataru Ohnishi https://orcid.org/0000-0002-4221-9305

REFERENCES

1. van der Meulen SH, Tousain RL, Bosgra OH. Fixed structure feedforward controller design exploiting iterative trials: application to a wafer stage and a desktop printer. *J Dyn Syst Measur Control Trans ASME*. 2008;130(5):051006. doi:10.1115/1.2957626
2. Mishra S, Coaplen J, Tomizuka M. Precision positioning of wafer scanners segmented iterative learning control for nonrepetitive disturbances. *IEEE Control Syst Mag*. 2007;27(4):20-25.
3. Hayashi T, Fujimoto H, Isaoka Y, Terada Y. Projection-based iterative learning control for ball-screw-driven stage with consideration of rolling friction compensation. *IEEJ J Ind Appl*. 2020;9(2):132-139. doi:10.1541/ieejjia.9.132
4. Oomen T. Advanced motion control for precision mechatronics: control, identification, and learning of complex systems. *IEEJ J Ind Appl*. 2018;7(2):127-140.
5. Csencsics E, Ito S, Schlarp J, Schitter G. System integration and control for 3D scanning laser metrology. *IEEJ J Ind Appl*. 2019;8(2):207-217. doi:10.1541/ieejjia.8.207
6. Wallén J, Norrlöf M, Gunnarsson S. A framework for analysis of observer-based ILC. *Asian J Control*. 2011;13(1):3-14. doi:10.1002/asjc.261
7. Oomen T. Learning for advanced motion control. Proceedings of the International Workshop on Advanced Motion Control; 2020; Kristiansand, Norway.
8. Åström K, Hagander P, Sternby J. Zeros of sampled systems. *Automatica*. 1984;20(1):31-38.
9. Hagiwara T, Yuasa T, Araki M. Stability of the limiting zeros of sampled-data systems with zero-and first-order holds. *Int J Control*. 1993;58(6):1325-1346.
10. Tomizuka M. Zero phase error tracking algorithm for digital control. *J Dyn Syst Meas Control*. 1987;109:65-68.
11. van Zundert J, Oomen T. On inversion-based approaches for feedforward and ILC. *Mechatronics*. 2018;50(September):282-291. doi:10.1016/j.mechatronics.2017.09.010
12. Oomen T, van de Wijdeven J, Bosgra O. Suppressing intersample behavior in iterative learning control. *Automatica*. 2009;45(4):981-988. doi:10.1016/J.AUTOMATICA.2008.10.022
13. Fujimoto H, Hori Y, Kawamura A. Perfect tracking control based on multirate feedforward control with generalized sampling periods. *IEEE Trans Ind Electron*. 2001;48(3):636-644.
14. Ohnishi W, Beauduin T, Fujimoto H. Preactuated multirate feedforward control for independent stable inversion of unstable intrinsic and discretization zeros. *IEEE/ASME Trans Mechatron*. 2019;24(2):863-871. doi:10.1109/TMECH.2019.2896237
15. Ohnishi W, Strijbosch N, Oomen T. Multirate state tracking for improving intersample behavior in iterative learning control. Proceedings of the IEEE International Conference on Mechatronics; 2021; Chiba, Japan.
16. Chen T, Francis BA. *Optimal Sampled-Data Control Systems*. Springer; 1995.
17. Bristow DA, Tharayil M, Alleyne AG. A survey of iterative learning control. *IEEE Control Syst Mag*. 2006;26(3):96-114. doi:10.1109/MCS.2006.1636313
18. Moore KL. Iterative learning control for deterministic systems. In: Grimble MJ, Johnson MA, eds. *Advances in Industrial Control*. Springer; 1993:23-36.
19. Norrlöf M, Gunnarsson S. Time and frequency domain convergence properties in iterative learning control. *Int J Control*. 2002;75(14):1114-1126. doi:10.1080/00207170210159122
20. Oomen T, Rojas CR. Sparse iterative learning control with application to a wafer stage: achieving performance, resource efficiency, and task flexibility. *Mechatronics*. 2017;47:134-147. doi:10.1016/j.mechatronics.2017.09.004
21. Blanken L, Oomen T. Multivariable iterative learning control design procedures: from decentralized to centralized, illustrated on an industrial printer. *IEEE Trans Control Syst Technol*. 2020;28(4):1534-1541. doi:10.1109/TCST.2019.2903021
22. Strijbosch N, Oomen T, Blanken L. Frequency domain design of iterative learning control and repetitive control for complex motion systems. Proceedings of the IEEJ International Workshop on Sensing, Actuation, Motion Control, and Optimization; 2018; Tokyo, Japan.
23. Bittanti S, Colaneri P. *Periodic Systems: Filtering and Control*. Springer; 2009.

How to cite this article: Ohnishi W, Strijbosch N, Oomen T. State-tracking iterative learning control in frequency domain design for improved intersample behavior. *Int J Robust Nonlinear Control*. 2023;33(7):4009-4027. doi: 10.1002/rnc.6511