Frequency-Domain ILC Approach for Repeating and Varying Tasks: With Application to Semiconductor Bonding Equipment

Frank Boeren, Abhishek Bareja, Tom Kok, and Tom Oomen

Abstract—Iterative learning control (ILC) enables high performance for exactly repeating tasks in motion systems. Besides such tasks, many motion systems also exhibit varying tasks. In such cases, ILC algorithms are known to deteriorate performance. An example is given by bonding equipment in semiconductor assembly processes, which contains motion axes with tasks that can vary slightly. The aim of this paper is to develop an ILC approach that obtains high machine performance for possibly varying tasks, while enabling straightforward and effective industrial design rules. In particular, a frequency-domain based design of ILC filters is pursued, which is combined with basis functions to cope with variations in tasks. Application to a high-speed axis of an industrial wire-bonder shows that high servo performance is obtained for both repeating and varying tasks.

I. INTRODUCTION

Feedforward control is widely used in control systems for motion systems, since feedforward can effectively reject disturbances before these affect the system. Indeed, a two-degree-of-freedom configuration, consisting of feedback and feedforward control, is standard in high-precision motion systems [1]. In this control configuration, the main performance improvement is in general realized by exploiting feedforward to minimize the reference-induced servo error.

Iterative Learning Control (ILC) is a feedforward control approach that can significantly enhance the performance of motion systems by learning from previous tasks. To this end, the measured error signal from the previous tasks is exploited to determine a feedforward signal that forces the output to track an a priori known, repeating reference trajectory [2]. Successful application of ILC algorithms in industrial practice are reported for, e.g., linear motors [3], wafer stages [4], [5], [6], micro-robotic deposition manufacturing systems [7], and atomic force microscopes [8], [9].

ILC algorithms are often designed by either using a norm-optimal approach or a frequency-domain loop-shaping approach, see [10] for an overview. In the norm-optimal approach, performance and robustness requirements are prescribed by means of weighting matrices. Addressing the trade-off between performance and robustness with respect to system uncertainty typically requires repeated user interaction to refine the weighting matrices or advanced synthesis tools from robust control [11]. In contrast, performance and robustness goals in frequency-domain ILC are often specified by means of loop-shaping based techniques in the frequency-domain [10]. As such, the trade-off between performance and robustness with respect to system uncertainty can be easily addressed in the design of an ILC algorithm. This low-complexity characteristic of frequency-domain ILC is typically favored in industrial practice. In fact, this ILC design approach closely resembles the manual tuning of PID controllers, which is a dominant control design approach in industrial applications [12].

Independent of the pursued ILC design approach, a key assumption in ILC is that the considered system performs exactly repeating tasks. This assumption is often violated in industrial systems [13], [2], [8]. For instance, several axes in semiconductor bonding equipment have to perform slightly varying tasks. The performance for these axes is significantly deteriorated if the trajectory is changed. As a result, machine performance, which is often dictated by the worst performing axis, is typically poor when using ILC.

In [14], [15], [16], robustness against varying tasks is obtained by extending norm-optimal ILC algorithms with basis functions. However, existing approaches based on basis functions are restricted to norm-optimal ILC, and analogous results for frequency-domain ILC are not yet available. This limits industrial applicability of ILC for systems that comprise both exactly repeating tasks and varying tasks, as is the case in semiconductor bonding equipment.

Although important developments have been made to enhance the performance of industrial systems with repeating tasks using frequency-domain ILC and significant steps are reported to enhance robustness against varying tasks using norm-optimal ILC, at present an approach to design frequency-domain ILC for systems comprising both tasks is lacking. The contribution of this paper is threefold. First, a theoretical framework is developed for frequency-domain ILC with basis functions that is applicable to both repeating and varying tasks. Second, it is shown that the presented ILC approach i) achieves high performance for repeating and varying tasks, and ii) enables the use of standard frequency-domain ILC design tools using loop-shaping for both performance and robustness. Finally, an application to bonding equipment is presented to compare the proposed approach with existing ILC approaches for both repeating and varying tasks. This paper
A. Bonding processes

Discrete semiconductor devices, such as integrated circuits, are manufactured in large batches on a silicon wafer by means of lithography. Upon completion, the processed wafers are diced to divide each wafer into individual semiconductor devices, often referred to as dies. Then, bonding equipment is used to (i) assemble a die onto a frame, and (ii) provide electrical connections to the outside world. Finally, the bonded die and frame are sealed by a molded plastic enclosure.

Die-bonder: The first step in the bonding process is the transfer of individual dies from the wafer and attachment onto a lead-frame. This lead-frame is a metal structure with contact leads that extend outside the package. The process is carried out using a die-bonder, as depicted in Figure 1. The die-bonder can be characterized into four separate units, each having multiple axes. These units are (1) the wafer positioning unit, which positions the wafer relative to the die-transfer unit, (2) the die-transfer unit, which transfers the die from the wafer onto the lead-frame, (3) the work-holder unit, used for accurate positioning of the lead-frame with respect to the die, and (4) the flip unit, used to flip a single die.

Wire-bonder: In the next step, electrical connections between the die and the lead-frame are made using copper or gold wires, see, e.g., [19] for a detailed description of the wire-bonding process. This process is carried out by a separate machine called a wire-bonder, as shown in Figure 2. The wire-bonder is characterized into three separate units: (1) the $XY$ table, used for horizontal positioning of the wire with respect to the lead-frame, (2) the $Z$-axis, used to position the wire vertically and feed it onto the lead-frame and (3) the work-holder unit, used for indexing of the lead-frame.

The various axes in bonding equipment can be classified into two key categories:

1) Axes with exactly repeating trajectories. This holds for, e.g., all axis in the wire-bonder, and the die-transfer and flip unit of the die-bonder,

2) Axes with varying trajectories. These variations occur due to corrections required in the pick or place position in consecutive tasks. This holds for, e.g., the work-holder unit, and the wafer-positioning unit of the die-bonder.

An overview of the properties of all units in bonding equipment is provided in Table I.

### TABLE I: Units with repeating and slightly varying trajectories in bonding equipment.

<table>
<thead>
<tr>
<th>Bonding process</th>
<th>Unit</th>
<th>Repeating</th>
<th>Slightly varying</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die-bonder</td>
<td>Wafer-positioning unit</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Die-transfer unit</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Flip unit</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Work-holder unit</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Wire-bonder</td>
<td>$XY$ table</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$Z$-axis</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Work-holder unit</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
B. Problem setup

Consider the closed loop system as depicted in Figure 3. Here, the true unknown system $P_0(z)$ is assumed to be discrete-time, single-input single-output, and linear time-invariant, while $C_{fb}(z)$ is a discrete-time stabilizing feedback controller. Furthermore, the sensitivity function is denoted as $S_0(z) = (1 + P_0(z)C_{fb}(z))^{-1}$, while the process sensitivity function is given by $S_0 P_0(z)$.

The closed-loop system in Figure 3 is repeatedly acted upon by a reference $r$ of length $N$. Each repetition of the reference $r$ is called a task, as denoted by subscript $j$. The feedforward signal in the $j$th task is denoted as $f_j$. Using lifted system representations of $S_0(z)$ and $S_0 P_0(z)$, the error signal $e_j$ in task $j$ (resp. $e_{j+1}$ in task $j+1$) is given by

$$e_j = S_0 r_j - S_0 P_0 f_j,$$
$$e_{j+1} = S_0 r_{j+1} - S_0 P_0 f_{j+1}. \quad (2)$$

By assuming that $r_{j+1} = r_j$, the predicted error in task $j+1$ is

$$\hat{e}_{j+1} = e_j - SP(f_{j+1} - f_j), \quad (4)$$

where $SP$ is the lifted system representation of a model $SP(z) = (I + P(z)C_{fb}(z))^{-1} P(z)$ of $S_0 P_0(z)$. In a typical ILC approach, $f_{j+1}$ is determined by using a frequency-domain or a norm-optimal approach. Next, norm-optimal ILC with basis functions is briefly recapitulated.

C. Norm-Optimal ILC with Basis Functions

Norm-optimal ILC is an important class of ILC approaches, in which $f_{j+1}$ is typically determined according to

$$f_{j+1} = \arg \min_{f_{j+1}} J(f_{j+1}), \quad (5)$$

with optimization criterion $J(f_{j+1})$ given by

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

where $W_e$, $W_f$ and $W_{\Delta f}$ are positive-definite weighting matrices, and $\hat{e}_{j+1}$ in (4). Solving (5) yields the following learning update

$$f_{j+1} = Q(f_j + L e_j), \quad (6)$$

with learning filter $L$ and robustness filter $Q$. Detailed expressions for $L$ and $Q$ are presented in, e.g., [10].

A key assumption underlying (6) is that the system performs exactly repeating tasks, i.e., for motion systems it should hold that $r_{j+1} = r_j$ for all $j$. Violating this assumption can significantly reduce the achieved servo performance, as illustrated in the following example.
Example 1. Suppose that i) $f_j = P_0^{-1}r_j$, ii) $Q = I$, and iii) $r_{j+1} \neq r_j$. Note that substituting $f_j = P_0^{-1}r_j$ in (2) gives $e_j = 0$. By using $e_j = 0$ and $Q = I$ in (6), it follows that $f_{j+1} = f_j$, i.e., the ILC algorithm in (6) has converged. By substituting $f_j = f_j = P_0^{-1}r_j$ in (3) and rearranging terms, it follows that

$$e_j = S_0(r_{j+1} - r_j).$$

Expression (7) reveals that the performance in task $j + 1$ can significantly deteriorate if $r_{j+1} \neq r_j$. In fact, the achieved performance can be significantly worse than by using only feedback, for example if $r_{j+1} = -r_j$.

As argued in Section III-A, the assumption that $r_{j+1} = r_j$ for all $j$ is often violated in motion systems. In fact, robustness with respect to varying tasks, i.e., allowing that $r_{j+1} \neq r_j$, is one of the key reasons to introduce basis functions in norm-optimal ILC, see [20], [21], [14], [15]. Basis functions can provide robustness against varying tasks by parametrizing $f_{j+1}$ as a function of $r_{j+1}$. A finite time implementation of such a parametrization is given by

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

where $\mathcal{F}(\theta_{j+1})$ is based on the transfer function $F(z, \theta_{j+1}) = \sum_{i=1}^{n} \psi_i(z)\theta_i$ with basis functions $\psi_i(z), i = 1, 2, ..., n$.

Instead of determining a signal $f_j$ according to (5), ILC with basis functions aims to determine the parameters $\theta_{j+1}$. Typically, the optimization problem is given by

$$\theta_{j+1} = \arg \min_{\theta_{j+1}} J_\theta(\theta_{j+1}),$$

with optimization criterion

$$J_\theta(\theta_{j+1}) = ||\hat{\epsilon}_{j+1}||^2_{W_\epsilon} + ||\theta_{j+1}||^2_{W_\theta} + ||\theta_{j+1} - \theta_j||^2_{W_{\Delta\theta}},$$

where $W_\epsilon$, $W_\theta$ and $W_{\Delta\theta}$ are positive-definite weighting matrices, and $\hat{\epsilon}_{j+1}$ as in (4). Based on a similar derivation as used for (6), the following learning update can be derived

$$\theta_{j+1} = Q_\theta(\theta_j + \mathcal{L}_\theta e_j),$$

where $Q_\theta$ and $\mathcal{L}_\theta$ are a robustness and learning filter, respectively. The following example shows that robustness against varying tasks can be obtained with basis functions in ILC.

Example 2. Suppose that i) $\mathcal{F}(\theta_j) = P_0^{-1}$, ii) $Q_\theta = I$, and iii) $r_{j+1} \neq r_j$. Then, $f_j$ parametrized as in (8) is equal to $f_j = P_0^{-1}r_j$. Furthermore, by substituting $f_j = P_0^{-1}r_j$ in (2) it follows that $e_j = 0$. By using $e_j = 0$ and $Q_\theta = I$ in (10), it follows that $\theta_{j+1} = \theta_j$, i.e., (10) has converged. As a result, $\mathcal{F}(\theta_{j+1}) = \mathcal{F}(\theta_j)$, which implies that $\mathcal{F}(\theta_{j+1}) = P_0^{-1}$ and consequently from (8) that

$$f_{j+1} = P_0^{-1}r_{j+1}.$$
IV. PROJECTION-BASED ILC

In this section, an ILC approach is presented that achieves high performance for repeating and varying tasks, while relying on easy-to-use design rules. An overview of the proposed approach is depicted in Fig. 4, and is briefly introduced next.

Proposed projection-based ILC approach:
Design \( L(z) \) and \( Q(z) \) in frequency-domain using loop-shaping methods, and determine \( f_{j+1} \) according to an ILC learning update. Then, depending on the type of task:

a. Axes with exactly repeating tasks:
Apply \( f_{j+1} \) in the next task.

b. Axes with varying tasks:
Determine \( \theta_{j+1} \) such that \( f_{j+1}^{\text{proj}} = \mathcal{F}(\theta_{j+1})r_{j+1} \) approximates \( f_{j+1} \). Then, apply \( f_{j+1}^{\text{proj}} \) in the next task.

In the proposed approach, a feedforward signal \( f_{j+1}^{\text{proj}} \) is determined for each axis, irrespective of the typical tasks performed by an axis. Only for axes subject to varying tasks, an add-on is proposed to determine parameters \( \theta_{j+1} \) of a feedforward controller based on the signal \( f_{j+1} \). Similar to norm-optimal ILC with basis functions, this projection step provides robustness against varying tasks, as in Example 2.

A. Frequency-domain ILC

A frequency-domain ILC algorithm to determine \( f_{j+1} \) based on \( f_j \) and \( e_j \) is given by

\[
f_{j+1} = Q(z)(f_j + L(z)e_j),
\]

where the learning filter \( L(z) \) and robustness filter \( Q(z) \) are discrete-time transfer functions, see, e.g., [10]. A condition for monotonic convergence in the 2-norm of the input signal \( f_j \) over an infinite time interval is given by

\[
\sup_{\omega \in [0,2\pi]} |Q(e^{j\omega})(1 - L(e^{j\omega})S_0(e^{j\omega})P_0(e^{j\omega}))| < 1.
\]

A proof of (13) and monotonic convergence conditions for frequency-domain ILC are proposed in [23].

Design considerations for \( L(z) \) and \( Q(z) \) are well-known for frequency-domain ILC algorithms, see, e.g., [18] and [10]. In a closed-loop setting with a parallel controller structure, \( L \) should approximate the inverse of \( S_0P_0 \). The design of \( Q \) is a trade-off between convergence and the achievable asymptotic error. Ideally, \( Q \) has a unit gain for all frequencies.

B. Projection step for Frequency-domain ILC

For axes subject to varying tasks, a projection step is proposed to approximate \( f_{j+1} \) in (12) with \( f_{j+1}^{\text{proj}} = \mathcal{F}(\theta_{j+1})r_{j+1} \). Since \( f_{j+1}^{\text{proj}} \) is a function of \( r_{j+1} \), robustness against varying tasks can be achieved, see Example 2. First, an equivalent parametrization for \( f_{j+1}^{\text{proj}} \) is presented in Definition 1 to simplify notation. Second, basis functions are proposed for high-precision motion systems. Finally, an optimization problem is posed to determine \( \theta_{j+1} \).

Definition 1. The feedforward signal \( f_{j+1}^{\text{proj}} \) is parameterized as

\[
f_{j+1}^{\text{proj}}(\theta_j) = \Psi_{r_j} \theta_j,
\]

with parameters \( \theta_j \in \mathbb{R}^{n_{\theta}} \) and basis functions

\[
\Psi_{r_j} = [\psi_1 r_j \ \psi_2 r_j \ \ldots \ \psi_{n_{\theta}} r_j] \in \mathbb{R}^{N \times n_{\theta}}.
\]

The design of the basis functions \( \Psi_{r_j} \) depends on the properties of \( P_0 \). For high-precision motion systems, a good description of the low-frequency behavior of \( P_0 \) can be obtained by using basis functions that are derivatives of \( r_j \) [2]. Typically, \( \Psi_{r_j} \) consist of position, velocity, acceleration, jerk and snap setpoints, which correspond to the 0\(^{\text{th}} \), 1\(^{\text{st}} \), 2\(^{\text{nd}} \), 3\(^{\text{rd}} \) and 4\(^{\text{th}} \) order derivative of \( r_j \), respectively. For example, for \( r_j \) as shown in Fig. 5, \( \Psi_{r_j} \) is depicted in Fig. 6. Further motivation for this basis is given in [24] and [2].

Next, an optimization problem is presented to determine \( \theta_{j+1} \). Let \( f_{j+1} \) be determined by (12), and let \( e_{j+1} \) in task \( j+1 \) be given by (4). Then, \( \theta_{j+1} \) is determined according to

\[
\hat{\theta}_{j+1} = \arg \min_{\theta_{j+1}} J(\theta_{j+1}),
\]

with optimization criterion

\[
J(\theta_{j+1}) = \|e_{j+1} - f_{j+1}^{\text{proj}}(\theta_{j+1})\|^2,
\]

and predicted error \( e_{j+1}^{\text{proj}} \), assuming that \( r_{j+1} = r_j \), given by

\[
e_{j+1}^{\text{proj}}(\theta_{j+1}) = e_j - \mathcal{SP}(f_{j+1}^{\text{proj}}(\theta_{j+1}) - f_j),
\]

where \( f_{j+1}^{\text{proj}}(\theta_{j+1}) \) is defined according to Definition 1.

By substituting (4) and (16) in (15), and rearranging terms it follows that (15) is equivalent to

\[
J(\theta_{j+1}) = \|\mathcal{SP} f_{j+1} - \mathcal{SP} \Psi_{r_j} \theta_{j+1}\|^2.
\]

Clearly, minimizing with respect to \( \theta_{j+1} \) in (14) with \( J(\theta_{j+1}) \) in (17) is equivalent to the linear least squares solution to

\[
\Psi_{r_j} \theta_{j+1} = \mathcal{SP} f_{j+1},
\]
with \( SPf_{j+1} \in \mathbb{R}^{N \times 1} \), and \( \Psi_{SPr_j} \) defined as
\[
\Psi_{SPr_j} = [SP\psi_1, SP\psi_2, ..., SP\psi_n] \in \mathbb{R}^{N \times n_o}. \tag{19}
\]
By assuming that \( \Psi_{SPr_j}^T \Psi_{SPr_j} \) is nonsingular, the unique solution to (18) is given by
\[
\hat{\theta}_{j+1} = (\Psi_{SPr_j}^T \Psi_{SPr_j})^{-1} \Psi_{SPr_j}^T SPf_{j+1}. \tag{20}
\]

**Remark 3.** The assumption that \( \Psi_{SPr_j}^T \Psi_{SPr_j} \) is nonsingular imposes a persistence of excitation condition on \( r_j \). If \( \Psi_{SPr_j}^T \Psi_{SPr_j} \) is singular, the parameters \( \hat{\theta}_{j+1} \) are non-unique and multiple solutions for \( \theta_{j+1} \) exist. A particular solution for (20) can be found using the pseudo-inverse of \( \Psi_{SPr_j}^T \Psi_{SPr_j} \).

Weighting both \( f_{j+1} \) and \( f_{j+1}^{proj} \) with \( SP \) in (17) is essential to achieve high performance. To see this, note that the weighting with \( SP \) in (17) implies that \( f_{j+1}^{proj}(\theta_{j+1}) \) is determined such that \( \hat{e}_{j+1}^{proj} \) approximates \( \hat{e}_{j+1} \). Hence, \( \theta_{j+1} \) in (14) is computed with respect to the performance measure \( \hat{e}_{j+1} \).

**Remark 4.** In the proposed method, \( f_{j+1} \) is determined by means of frequency-domain ILC. Clearly, it is also possible to determine \( f_{j+1} \) based on (5), i.e., norm-optimal ILC, and then determine \( \theta_{j+1} \) as in (14). However, this approach still requires cumbersome tuning of weighting functions to address the trade-off between performance and robustness in ILC.

**C. Projection-based ILC Procedure**

In this section, two procedures are presented for the proposed projection-based ILC approach. First, Procedure 1 is proposed for axes subject to exactly repeating tasks, and therefore implements standard frequency-domain ILC (SILC).

**Procedure 1. (SILC) Axes with Exactly Repeating Tasks**

(A) Initialization Procedure

1. Determine a parametric model \( SP(z) \).
2. Design \( L(z) \) and \( Q(z) \) using loop-shaping in the frequency-domain.
3. Set \( j = 0 \).

(B) Given \( f_j \) and measured \( e_j \) from the \( j^{th} \) task, determine \( f_{j+1} \) according to the learning update in (12).

(C) Terminate: set \( j \rightarrow j+1 \) and repeat (B) until convergence.

Procedure 2 is proposed for axes subject to varying tasks, and implements projection-based ILC (PILC).

**Procedure 2. (PILC) Axes with Varying Tasks**

(A) Initialization Procedure

1. Determine a parametric model \( SP(z) \).
2. Design \( L(z) \) and \( Q(z) \) using loop-shaping in the frequency-domain.
3. Select \( \Psi_{r_0} \) and \( \theta_0 \), and set \( f_0^{proj} = \Psi_{r_0} \theta_0 \).
4. Set \( j = 0 \).

(B) Given \( f_j^{proj} \), measured \( e_j \) in the \( j^{th} \) task, \( r_j \) and \( r_{j+1} \), perform the following steps to determine \( f_{j+1}^{proj} \)

1. Construct \( f_{j+1} = Q(f_j^{proj} + Le_j) \).
2. Solve \( \hat{\theta}_{j+1} = (\Psi_{SPr_j}^T \Psi_{SPr_j})^{-1} \Psi_{SPr_j}^T SPf_{j+1} \).
3. Construct \( f_{j+1}^{proj} = \Psi_{r_{j+1}} \hat{\theta}_{j+1} \).

(C) Terminate: set \( j \rightarrow j+1 \) and repeat (B) until convergence.
for standard ILC in Procedure 1, monotonic convergence of \( f_{j+1} \) can be verified by the well-known condition (13). Redesigning \( L(z) \) and \( Q(z) \) is required if (13) is not satisfied. For Procedure 2, sufficient conditions for monotonic convergence of \( f^\text{proj}_{j+1} \) in the 2-norm are derived in the next section.

**Remark 5.** Convergence of Procedure 2 is insensitive to the initial parameters \( \theta_0 \) since (18) has an analytic solution.

The projection-based ILC approach in Procedure 2 is related to feedforward approaches based on model-inversion techniques, see, e.g., [25], [26]. Therefore, it is expected that similar performance can be obtained with model-based feedforward for axes with varying tasks. However, for axes with exactly repeating tasks, ILC can significantly improve performance compared to model-based feedforward [10]. Thus, model-based feedforward (resp. ILC) results in mediocre performance compared to model-based feedforward [10]. Therefore, it is expected that similar performance can be obtained with model-based feedforward techniques, see, e.g., [25], [26].

**Lemma 1.** Let \( L \) and \( Q \) denote the convolution matrices corresponding to \( L(z) \) and \( Q(z) \), respectively. Then, the frequency-domain condition for monotonic convergence in the 2-norm as given in (13) implies that
\[
\bar{\sigma}(\Psi(I - \mathcal{L}_0 P_0)) < 1,
\]
if \( L(z) \) and \( Q(z) \) are stable and causal.

A proof of Lemma 1 follows directly from [23, Theorem 8]. Note that (21) implies that (12) is monotonically convergent in the 2-norm for a finite time interval.

**Remark 6.** Lemma 1 does not necessarily hold if \( L(z), Q(z) \in \mathbb{R} \mathcal{L}_\infty \).

To proceed, a condition for monotonic convergence of \( f^\text{proj}_{j+1} \) is given in the following lemma.

**Lemma 2.** Let \( L \) and \( Q \) denote the convolution matrices corresponding to \( L(z) \) and \( Q(z) \), respectively. The input signal \( f^\text{proj}_{j+1} \) is monotonic convergent in the 2-norm if and only if
\[
\bar{\sigma}\left(\Psi_{r_{j+1}}(\Psi^T_{SP_{r_j}} \Psi_{SP_{r_j}})^{-1} \Psi^T_{SP_{r_j}} SP Q(I - \mathcal{L}_0 P_0)\right) < 1.
\]

**Proof.** Substituting (20) in \( f^\text{proj}_{j+1} = \Psi_{r_{j+1}} \theta_{j+1} \) gives
\[
f^\text{proj}_{j+1} = \Psi_{r_{j+1}} \left(\Psi^T_{SP_{r_j}} \Psi_{SP_{r_j}}\right)^{-1} \Psi^T_{SP_{r_j}} SP f_{j+1},
\]
From Step B1 in Procedure 2 it follows that \( f_{j+1} = Q(f^\text{proj}_j + \mathcal{L}e_j) \) in PILC. In finite time, this becomes
\[
f_{j+1} = Q(f^\text{proj}_j + \mathcal{L}e_j).
\]
By using (24) in (23) it follows that
\[
f^\text{proj}_{j+1} = \Psi_{r_{j+1}} \left(\Psi^T_{SP_{r_j}} \Psi_{SP_{r_j}}\right)^{-1} \Psi^T_{SP_{r_j}} SP Q(f^\text{proj}_j + \mathcal{L}e_j)\]
Substituting (2) into (25) yields
\[
f^\text{proj}_{j+1} = \Psi_{r_{j+1}} \left(\Psi^T_{SP_{r_j}} \Psi_{SP_{r_j}}\right)^{-1} \Psi^T_{SP_{r_j}} SP Q(I - \mathcal{L}_0 P_0) f^\text{proj}_{j+1} + \mathcal{L}_0 r_{j+1}.
\]
Then, condition (22) follows from [23, Theorem 2].

**Theorem 1.** Suppose that \( L(z), Q(z), SP(z) \) and \( \psi_i(z), i = 1, 2, ..., n_0 \), are stable and causal, and \( r_{j+1} = r_j \). Let \( L, Q \) and \( SP \) denote the convolution matrices corresponding to \( L(z), Q(z) \) and \( SP(z) \), respectively. Then, \( f^\text{proj}_{j+1} \) is monotonic convergent in the 2-norm if
\[
\sup_{\omega \in [0, 2\pi]} |Q(e^{j\omega})(1 - L(e^{j\omega})S_0(e^{j\omega})P_0(e^{j\omega}))| < 1.
\]

**Proof.** Condition (22) in Lemma 2 is upper bounded by
\[
\bar{\sigma}\left(\Psi_{r_{j+1}}(\Psi^T_{SP_{r_j}} \Psi_{SP_{r_j}})^{-1} \Psi^T_{SP_{r_j}} SP Q(I - \mathcal{L}_0 P_0)\right) \leq \bar{\sigma}(\Gamma) \bar{\sigma}(Q(I - \mathcal{L}_0 P_0))
\]
where
\[
\Gamma = \Psi_{r_{j+1}}(\Psi^T_{SP_{r_j}} \Psi_{SP_{r_j}})^{-1} \Psi^T_{SP_{r_j}} SP.
\]
If \( \bar{\sigma}(\Gamma) = 1 \), it follows from (27) that (22) is equal to (21). Then, Lemma 1 implies that (26) is a condition for monotonic convergence of \( f^\text{proj}_{j+1} \).

To show that \( \bar{\sigma}(\Gamma) = 1 \) for the considered case, note that causality of \( SP(z) \) and \( \psi_i(z), i = 1, 2, ..., n_0 \) imply that \( SP \) and \( \Psi_{r_{j+1}}(\Psi^T_{SP_{r_j}} \Psi_{SP_{r_j}})^{-1}\Psi^T_{SP_{r_j}} \) commute in (28). As a result, (28) is equivalently given by
\[
\Gamma = SP \Psi_{r_{j+1}}(\Psi^T_{SP_{r_j}} \Psi_{SP_{r_j}})^{-1} \Psi^T_{SP_{r_j}} SP.
\]
Furthermore, the assumption that \( r_{j+1} = r_j \) implies that an equivalent expression for \( \Gamma \) is given by
\[
\Gamma = \Psi_{SP_{r_j}}(\Psi^T_{SP_{r_j}} \Psi_{SP_{r_j}})^{-1} \Psi^T_{SP_{r_j}}.
\]
Expression (30) is a projection matrix, which has the property that \( \bar{\sigma}(\Gamma) = 1 \). Then, from (27) it follows that
\[
\bar{\sigma}\left(\Psi_{r_{j+1}}(\Psi^T_{SP_{r_j}} \Psi_{SP_{r_j}})^{-1} \Psi^T_{SP_{r_j}} SP Q(I - \mathcal{L}_0 P_0)\right) = \bar{\sigma}(Q(I - \mathcal{L}_0 P_0),
\]
and Lemma 1 implies that (26) is a condition for monotonic convergence of \( f^\text{proj}_{j+1} \).

Concluding, it is shown that monotonic convergence of PILC in Procedure 2 can be guaranteed by designing \( L(z) \).
and $Q(z)$ such that the frequency-domain condition in (13) holds. That is, monotonic convergence of $f_{j+1}$ directly implies monotonic convergence of $f_{j+1}^{(m)}$ in PILC. Essentially, the projection step adds additional robustness to the ILC scheme, as precisely characterized by (27). Similar to Procedure 1, redesigning $L(z)$ and $Q(z)$ is required if (13) is not satisfied.

VI. APPLICATION TO WIRE-BONDING

A. Case study

In this section, a case study is presented based on the Z-axis of a wire-bonder. This case study aims to:

i) Confirm that SILC (Procedure 1) and PILC (Procedure 2) are determined using identical design methods;

ii) Present simulation results for SILC and PILC with respect to random varying reference trajectories from task to task;

iii) Present experimental and simulation results for SILC and PILC with respect to:
   a. exactly repeating tasks;
   b. varying tasks.

B. Experimental setup

A schematic representation of the mechanics of the Z-axis is shown in Figure 8. This axis is used to position an ultrasonic transducer carrying the wire at its tip, as in [27], and is actuated by means of a voice coil motor, while a linear encoder with a resolution of 5.5 nm is used as position sensor. The structure is connected with hinges to the XY table. All experiments are performed with a sampling frequency of 8 kHz.

The mechanics of the Z-axis are modeled by two masses connected by springs and dampers, as depicted in Figure 9. Mass $m_1$ is the mass of the main structure carrying the actuator and sensor. The hinges that connect this structure to the fixed world are modeled with stiffness $k_1$ and damping $d_1$, while $z_1$ represents the encoder position measurement. Mass $m_2$ represents the mass of the transducer. The clamping of the transducer with the main structure induces resonant dynamics, and are denoted by $k_2, d_2$.

The feedback controller for the Z-axis is a discrete-time PD controller in series with two notch filters at 1000 Hz and 2046 Hz, respectively. A state-space representation of the discrete-time feedback controller $C_{fb}(z)$ is given by

$A_c = \begin{bmatrix} 1.126 & -0.739 & 0.502 & -0.297 \ 2 & 0 & 0 & 0 \ 0 & 1 & 0 & 0 \ 0 & 0 & 1 & 0 \end{bmatrix}$, \quad $B_c = \begin{bmatrix} 0.063 \ 0 \ 0 \ 0 \end{bmatrix}$,

$C_c = \begin{bmatrix} -0.018 & 0.012 & -0.016 & -0.012 & -0.007 \ 9.317 \times 10^{-4} \end{bmatrix}$.

Next, Procedure 2 is invoked and explained step-by-step.
C. Step A1 - Determine a parametric model $SP(z)$

The identified frequency response data of the system $P_0$ is used to estimate a discrete time parametric model $P(z)$. Figure 10 shows the Bode plot of this model $P(z)$ and the frequency response of $P_0$. Visual inspection reveals that $P(z)$ is an accurate model of the frequency response in the frequency range up to approximately 1500 Hz. A state-space model of $P(z)$ is given by

$$A_p = \begin{bmatrix} 3.407 & -1.195 & 0.835 & -0.482 \\ 4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \end{bmatrix}, \quad B_p = \begin{bmatrix} 32 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$C_p = \begin{bmatrix} 2.43 & 4.293 & -7.63 & 10 \end{bmatrix}, \quad D_p = 30.76.$$

Based on $C_{fb}(z)$ and $P(z)$, a minimal state-space realization for $SP(z)$ is determined with McMillan degree 9.

D. Step A2 - ILC filters $L(z)$ and $Q(z)$

In the proposed PILC approach, $L(z)$ and $Q(z)$ are designed using standard ILC design rules, as elaborated upon in Procedure 2. As such, identical design methods are used as for SILC in Procedure 1. The key difference between both methods lies in the projection step outlined in Section IV-B.

Here, the learning filter $L(z)$ is determined using the ZPETC algorithm [29]. The robustness filter $Q(z)$ is designed as a 4th order low-pass filter with a cut-off at 750 Hz to satisfy the condition for monotonic convergence in Theorem 1, as shown in Figure 11.

E. Step A3 - Feedforward parameterization and $\theta_0$

1) $\Psi$: For the considered system, the selected basis functions are position, velocity, acceleration, jerk, and snap of the reference trajectory. The position and velocity basis functions can compensate for the first flexible mode at 23 Hz, while acceleration feedforward is used to compensate for the rigid body dynamics. The jerk and snap terms can compensate for the low-frequency contributions of residual (higher-order) modes of the system [24]. All derivatives of the reference signal are computed using the forward Euler difference method.

2) $\theta_0$: The initial parameters are set to zero. This enables a comparison between the performance of the proposed algorithm and the performance without feedforward control.

F. Steps B1–B3 - Simulation Results

In this section, two simulation case studies are provided to demonstrate the proposed PILC approach. First, the performance of the proposed PILC approach is investigated with respect to random varying reference trajectories from task to task. Second, a case study is presented to show the limitations of the allowable reference variations. The true system is supposed to be equal to the nominal model $P$, i.e., $P_0 = P$.

Case Study I: Random varying 4th-order reference trajectory

The simulation results in this case study are obtained by performing 25 tasks with random varying trajectories from task to task. The nominal reference trajectory used in this section is depicted in Figure 5, and is designed according to the procedure in [24] with design parameters $r_{\max} = 3 \times 10^{-3}$ [m], $v_{\max} = 3$ [ms$^{-1}$], $a_{\max} = 300$ [ms$^{-2}$], $j_{\max} = 3 \times 10^5$ [ms$^{-3}$]. Based on the nominal design of $r$, 25 reference trajectories are generated by means of random selection from the following ranges of the design parameters: $r_{\max} = 3 \times 10^{-3} \pm 5 \times 10^{-4}$ [m], $v_{\max} = 3 \pm 1$ [ms$^{-1}$], $a_{\max} = 300 \pm 50$ [ms$^{-2}$], $j_{\max} = 3 \times 10^5 \pm 5 \times 10^4$ [ms$^{-3}$]. The reference trajectory is varied after each task. Figure 12 shows that $\|e_j\|_2$ for the PILC approach is invariant with respect to random variations in a 4th-order reference trajectory, while the performance obtained with SILC significantly deteriorates due to varying reference trajectories. This confirms that the proposed PILC
The number of tasks is given by

\[ j \]

and confirms that PILC can handle the change from reference trajectory \( r_1 \) to \( r_2 \) in task \( j = 10 \), as reflected in a comparable \( \| e_{10} \|_2 \) and \( \| e_9 \|_2 \). However, the performance deteriorates drastically after the change from \( r_2 \) to \( r_3 \) at task \( j = 20 \). This illustrates that the variation in the trajectory from task to task cannot be arbitrarily large. Essentially, the performance with the proposed PILC approach is invariant to changes in the reference if the considered reference trajectories have a similar frequency content. This is especially relevant if i) the basis is of limited complexity, and ii) the dominant frequency content is not filtered off by the Q-filter design. Note that the references \( r_1 \) and \( r_2 \) in Figure 13 have a similar frequency content, while \( r_3 \) has a significant high-frequency contribution.

**G. Steps B1–B3 - Experimental Results**

As introduced in Sect. VI-A, the aim of the experimental case study is to present experimental results for SILC and PILC with respect to i) exactly repeating tasks and ii) varying tasks. First, experimental results are presented for SILC and PILC with respect to exactly repeating tasks.

The experimental results in this case study are obtained by performing 20 tasks on the Z-axis of a wire-bonder. The reference trajectories in these tasks correspond to common pick and place tasks for the Z-axis during production. The reference trajectories \( r_j \) used in task \( j = 0, 1, \ldots, 9 \), and in task \( j = 10, 11, \ldots, 19 \) are depicted in Figure 15. Note that \( r_{10} \) in task \( j = 10 \) is varied with respect to \( r_9 \) in task \( j = 9 \). As such, experimental results for exactly repeating tasks are presented in task \( j = 0, 1, \ldots, 9 \) and \( j = 11, \ldots, 19 \), while experimental results for varying tasks are given in task \( j = 10 \).

**Case Study I: Exactly repeating tasks**

The reference trajectory \( r_j \) is exactly repeating in task \( j = 0, 1, \ldots, 9 \), and in task \( j = 11, \ldots, 19 \). Figure 16 shows that starting from \( \| e_0 \|_2 = 0.8 \text{ mm} \) in task \( j = 0 \), the 2-norm of \( e_0 \) in task \( j = 9 \) has converged to 3.7 \( \mu \text{m} \) for SILC and 7.4 \( \mu \text{m} \) for PILC. For exactly repeating tasks, the performance with PILC is slightly less than with SILC due to the limited basis of \( f_{j}^{\text{proj}} \) as defined in Def. 1. This observation is confirmed by the time domain error signals in task \( j = 9 \) as depicted in Figure 17. Similar results are obtained for tasks \( j = 11, \ldots, 19 \).

Next, consider SILC and PILC for varying tasks.

**Case Study II: Varying tasks**

Reference \( r_{10} \) in task \( j = 10 \) is varied with respect to \( r_9 \) in task \( j = 9 \), as depicted in Figure 15. Therefore, the key experimental result motivating the proposed PILC approach is obtained in task \( j = 10 \). Figure 16 shows that \( \| e_{10} \|_2 \) for SILC is equal to 67 \( \mu \text{m} \) (from 3.7 \( \mu \text{m} \) in task \( j = 9 \)). This confirms that the servo performance for SILC severely deteriorates for varying reference trajectories. In contrast, the
increase in 2-norm for PILC is restricted to 9.1 \( \mu m \) (from 7.7 \( \mu m \) in task \( j = 9 \)). This result illustrates that the servo performance obtained with PILC is invariant to changes in the reference between tasks. The corresponding time domain error signals for SILC and PILC in task \( j = 10 \) as depicted in Figure 17 confirm these observations.

Based on the experimental results for SILC and PILC with respect to exactly repeating tasks and varying tasks, the following guidelines are proposed to achieve high performance:

i) **Exactly repeating tasks**: implement Procedure 1;

ii) **Varying tasks**: implement Procedure 2.

As such, a framework is obtained that is applicable to systems subject to both repeating as varying tasks.

VII. CONCLUSIONS AND ONGOING RESEARCH

In this paper, an ILC approach is proposed that is applicable to both repeating as varying tasks, while relying on loop-shaping based design rules in frequency-domain. For axes subject to exactly repeating tasks, a frequency-domain ILC approach is used to achieve high performance. For axes subject to varying tasks, a projection step is proposed to obtain robustness with respect to varying task. Experimental results on a high-speed axis of a wire-bonder confirm that high performance is obtained for repeating and varying trajectories.

The proposed ILC approach can be straightforwardly applied to other axes in die- and wire-bonders. For axes with varying tasks, a thorough analysis should be performed to quantify the allowable variations in the reference trajectory. For axes with dominant non-linear dynamics, such as position-dependent friction, cogging and non-linearities in the motor, the proposed approach requires extension of the used set of basis functions. The proposed ILC approach can be extended to compensate for other known repeating disturbances, and is applicable to systems with non-minimum phase zeros. Ongoing research focuses on extensions to a rational basis [16], [30], input shaping [31], random iteration-to-iteration variations [9], inferential control [32], and multivariable systems. Finally, connections should be investigated between the closely related design approach in [8] and the proposed approach in the present paper.

VIII. ACKNOWLEDGEMENTS

This research is partially supported by NXP Semiconductors, Philips Innovation Services and the Innovational Research Incentives Scheme under the VENI grant Precision Motion: Beyond the Nanometer (no. 13073) awarded by NWO (The Netherlands Organisation for Scientific Research) and STW (Dutch Science Foundation).

REFERENCES


