Sparse Iterative Learning Control with Application to a Wafer Stage: Achieving Performance, Resource Efficiency, and Task Flexibility

Tom Oomen[a], Cristian R. Rojas[b]

[a]Eindhoven University of Technology, Faculty of Mechanical Engineering, Control Systems Technology group, PO Box 513, 5600MB Eindhoven, The Netherlands.
[b]Automatic Control Lab, Electrical Engineering, KTH – Royal Institute of Technology, S-100 44 Stockholm, Sweden.

Abstract

Trial-varying disturbances are a key concern in Iterative Learning Control (ILC) and may lead to inefficient and expensive implementations and severe performance deterioration. The aim of this paper is to develop a general framework for optimization-based ILC that allows for enforcing additional structure, including sparsity. The proposed method enforces sparsity in a generalized setting through convex relaxations using ℓ_1 norms. The proposed ILC framework is applied to the optimization of sampling sequences for resource efficient implementation, trial-varying disturbance attenuation, and basis function selection. The framework has a large potential in control applications such as mechatronics, as is confirmed through an application on a wafer stage.

1. Introduction

Iterative Learning Control (ILC) enables significant performance improvements for batch-to-batch control applications, by generating a command signal that compensates for repetitive disturbances through learning from previous iterations, also called batches or trials. Theoretical and implementation aspects, including convergence, causality, and robustness, have been addressed in, e.g., [12], [1], [45], [41], [44]. Furthermore, successful applications have been reported in, e.g., robotics [53], mechatronics [9], manufacturing [27], building control [43], nuclear fusion [19], and rehabilitation [20]. However, several disadvantages of present ILC frameworks that limit further applications include i) high implementation cost due to highly unstructured command signals, which are expensive to implement; ii) amplification of trial-varying disturbances, including measurement noise; iii) inflexibility to changing reference trajectories. The aim of the present paper is to develop an ILC design framework that enforces sparsity, which enables addressing these aspects i)-iii).

Regarding i) ILC typically generates signals that require a large number of command signal updates thus leading to an expensive implementation. ILC directly learns from measured signals that are contaminated by trial-varying disturbances such as measurement noise. These trial-varying disturbances are often modeled as a realization of a stochastic process [32]. As a result, the ILC command signals have infinite support. In sharp contrast, command signals that are obtained through traditional feedforward designs, including [31], have finite support and are highly sparse. Command signals with a high number of non-zero elements, or another appropriate structural constraint, may lead to a prohibitively expensive implementation, e.g., in wireless sensor networks, wireless control applications, or embedded platforms with shared resources [22]. Note that this is a different aspect than the actual computation of the command signal itself, which can be done in between subsequent tasks, see [60] for results in this direction.

Regarding ii), ILC typically amplifies trial-varying disturbances. In fact, typical ILC approaches amplify these disturbances by a factor of two, as is shown in the present paper. Approaches to attenuate trial-varying disturbances include norm-optimal ILC with appropriate input weighting [12], higher-order ILC for addressing disturbances with trial-domain dynamics [24], and stochastic approximation-based ILC [14]. Also, a wavelet filtering-based approach is presented in [33], where a certain noise attenuation is achieved by setting certain wavelet coefficients to zero. In the present paper, a different approach is pursued to attenuate disturbances, where also wavelets immediately fit into the formulation, yet the sparsity can be enforced in an optimal way.

Regarding iii), changing reference signals typically lead to performance degradation of ILC algorithms [5], since these essentially constitute trial-varying disturbances. This is in sharp contrast to traditional feedforward designs [31] and is widely recognized in ILC designs. A basis task approach is proposed in [26], where the command input is segmented. A basis function framework is developed and applied in [54], [34], [8] using polynomial basis functions, which is further extended to rational basis functions in [61]. These basis functions are typically selected based on prior information, e.g., based on the approach in [31], and trial-and-error.

In model estimation and signal processing, the use of measured signals has comparable consequences, which has led to new regularization-based approaches that enforce sparsity. Early approaches include the non-negative garrote [10] and Least Absolute Shrinkage and Selection Operator (LASSO) [50]. These are further generalized in [52], [25], [13], [3]. Related applications in system identification include [46], [38].

Although important developments have been made in ILC and several successful applications have been reported, present approaches do not yet exploit the potential of enforcing addi-

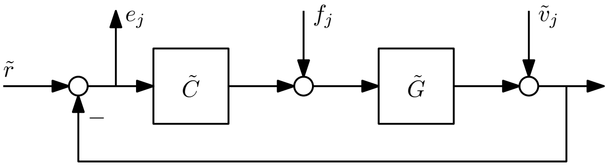

Figure 1: Parallel ILC structure (3) as an example of (1).

tional structure and sparsity. The aim of the present paper is to develop a unified optimization-based approach to ILC that allows for explicitly enforcing structure and sparsity, enabling improved resource efficiency, disturbance attenuation, and flexibility to varying reference signals. The approach employs convex relaxations, enabling the use of standard optimization routines.

The main contribution of the present paper is a unified framework to sparse ILC. As subcontributions, trial-varying disturbances are analyzed in detail for explicit ILC algorithms (Section 3). Subsequently, a general optimization-based framework to sparse ILC is developed (Section 4), including many specific cases that are relevant to ILC applications. The results are confirmed through an application to a wafer stage system (Section 5). Related developments to the results in the present paper include the use of sparsity in control, where the main results have been related to Model Predictive Control (MPC), see [2], [29], [21].

Notation: Throughout, $\|x\|_{\ell_p}$ denotes the usual ℓ_p norm, $p \in \mathbb{Z}_{>0}$. Also, $\|x\|_0 = \sum_i \mathbf{1}(x_i \neq 0)$, i.e., the cardinality of x. Note that $\|x\|_0$ is not a norm, since it does not satisfy the homogeneity property. It relates to the general ℓ_p norm by considering the limit $p \to 0$ of $\|x\|_p$. A signal $x \in \mathbb{R}^N$ is called sparse if many of its components are zero, in which case $\|x\|_0 \ll N$. In addition, $\|\tilde{X}\|_{\mathcal{L}_\infty}$ and $\|\tilde{X}\|_{\mathcal{H}_\infty}$ denote the usual $\mathcal{L}_\infty$ and $\mathcal{H}_\infty$ norms of discrete time systems, respectively. Throughout, J denotes a system that maps an input space to an output space, operating either over finite or infinite time, which follows from the context. In certain cases, the system is assumed linear, time invariant, and scalar, with transfer function representation $\tilde{J}$. The power spectrum of a signal x is denoted ϕ_x, and is defined as in [32, Section 2.3].

2. Problem formulation

Consider the ILC system

$$e_j = r - Jf_j - v_j \tag{1}$$

be given, where $e_j \in \ell_2$ denotes the error signal to be minimized, $r \in \ell_2$ is the reference signal, $f_j \in \ell_2$ denotes the command signal, and $v_j \in \ell_2$ represents trial-varying disturbances, including measurement noise. Here and in the sequel, all signals are tacitly assumed to have appropriate dimensions. Furthermore, J represents the true system, either open-loop or closed-loop, with causal and stable transfer function $\tilde{J} \in \mathcal{RH}_\infty$. The index $j \in \mathbb{Z}_{\geq 0}$ refers to the trial number. Throughout, the command signal f_{j+1} is generated by an ILC algorithm

$$f_{j+1} = F(f_j, e_j), \tag{2}$$

where the ILC update F is defined in more detail later on. The general setup (1) encompasses the parallel ILC setup in Figure 1, where

$$e_j = S\tilde{r} - SGf_j - S\tilde{v}_j \tag{3}$$

where S follows from its transfer function $\tilde{S} = \frac{1}{1+\tilde{G}\tilde{C}}$, $r = S\tilde{r}$, $J = SG$, $v_j = S\tilde{v}_j$, and $\tilde{C}$, $\tilde{G}$ are assumed to be linear.

From (2) and (1), it is immediate that the trial-varying disturbance v_j directly affects the ILC command signal. In view of this observation, the problem investigated in this paper is to develop an ILC algorithm (2) that satisfies the following requirements:

R1) the iteration (1)-(2) is convergent over j;

R2) the iteration (1)-(2) leads to a small error e_j in the presence of trial-invariant disturbances r and trial-variant disturbances v_j;

R3) the resulting command signal f_j has a certain structure, including
  (a) a small $\|f_j\|_0$, and/or,
  (b) a piecewise constant f_j with a small number of jumps.

Here, R1 is a basic requirement for any ILC algorithm and ensures stability in the trial domain, in addition to the assumed stability in the time domain that is guaranteed by stability of J in (1), see also [45] for the stability of such two-dimensional systems. Requirement R2 essentially states that the ILC algorithm should effectively compensate for r, while avoiding amplification of trial-varying disturbances v_j. Requirement R3 is imposed to enable resource-efficient implementations in terms of sampling or communication requirements, depending on the particular application requirements.

3. Analysis of Trial-Varying Disturbances in Explicit ILC

The main contribution of this paper is a design framework for ILC that allows for enforcing sparsity and structure in the command signals, which in turn allows for explicitly addressing aspects i)-iii) as mentioned in Section 1. The main mechanism behind these aspects i)-iii) are trial-varying disturbances. First, ILC uses measured data, where trial-varying disturbances lead to highly unstructured command signals, which in turn are expensive to implement. Second, these trial-varying disturbances are amplified by typical ILC algorithms. Third, typical ILC algorithms are inflexible to changing reference signals, which can in fact be interpreted as trial-varying disturbances. Since trial-varying disturbances have a central role in all these aspects i)-iii), these trial-varying references are analyzed in typical ILC algorithms in this section.

In particular, explicit linear ILC algorithms of the general form

$$f_{j+1} = Q(f_j + Le_j) \tag{4}$$

are considered. The infinite time scalar case is considered, where $Q : \ell_2 \mapsto \ell_2$ and $L : \ell_2 \mapsto \ell_2$. Here, Q and L have associated transfer functions $\tilde{Q} \in \mathcal{RL}_\infty$ and $\tilde{L} \in \mathcal{RL}_\infty$. Note that $\tilde{J} \in \mathcal{RH}_\infty$ reflects causality and stability of the system. The fact that $\tilde{Q} \in \mathcal{RL}_\infty$ and $\tilde{L} \in \mathcal{RL}_\infty$ reflects that typical ILC algorithms are typically non-causal, and are usually implemented such that bounded solutions are obtained through finite-time preview or via stable inversion through a bilateral Z-transform [60].

The trial-varying disturbance v_j in (1) will propagate throughout the iterations through the iteration-domain update (4). The following assumption is widely adopted [32].

Assumption 1. *Let $v_j = Hn_j$, where n_j is i.i.d. zero-mean white noise with variance λ_e, $\tilde{H}$ monic and bistable.*

Clearly, v_j typically does not have compact support. As a result, f_{j+1} will not have compact support in general due to ILC algorithm (4).

To enable a more detailed analysis, the following auxiliary result provides a suitable condition to guarantee that the iteration defined by (1) and (4) converges.

Theorem 2. *The iteration defined by* (1) - (4) *converges monotonically in the ℓ_2 norm to a fixed point f_∞ and resulting e_∞ iff*

$$\|\tilde{Q}(1 - \tilde{L}\tilde{J})\|_{\mathcal{L}_\infty} < 1.$$

Proof. Substituting (1) into (4) leads to $f_{j+1} = Q(I - LJ)f_j + QLr - QLv_j$. Using transfer function representations and subsequent application of the Banach fixed-point theorem in conjunction with [57, Theorem 4.4] yields the desired result. □

Note that Theorem 2 allows for non-causal ILC algorithms, i.e., $Q, L \in \mathcal{RL}_\infty$. This is more general compared to related analyses, including [35, Chapter 3], which only allow for causal ILC algorithms by restricting to the $\mathcal{H}_\infty$ norm.

The following result is the main result of this section and reveals the propagation of noise in the iteration defined by (1) and (4).

Theorem 3. *Given the system* (1) *and ILC update* (4) *with $f_0 = 0$, Assumption 1, and that the iteration is stable in the sense of Theorem 2, then the spectrum of e_∞ is given by*

$$\phi_{e_\infty} = \left|\frac{1-\tilde{Q}}{1-\tilde{Q}(1-\tilde{L}\tilde{J})}\right|^2 \phi_r + \left(1 + \frac{|\tilde{J}\tilde{Q}\tilde{L}|^2}{1-|\tilde{Q}(1-\tilde{L}\tilde{J})|^2}\right)\phi_v. \tag{5}$$

Theorem 3 provides a detailed analysis of the propagation of noise for the general ILC algorithm (4). In special cases, the result can be further simplified. For instance, in inverse-model ILC, $\tilde{Q} = 1$ and $\tilde{L} = \tilde{J}^{-1} \in \mathcal{RH}_\infty$, in which case Theorem 3 reveals that

$$\phi_{e_\infty} = 2\phi_v. \tag{6}$$

The result (6) reveals that the limit error spectrum involves an amplification of the noise spectrum by a factor of two.

Inclusion of a learning gain $\alpha \in (0, 1]$ in inverse-model ILC, i.e., replacing (4) by $f_{j+1} = Q(f_j + \alpha L e_j)$, mitigates the amplification of trial-varying disturbances, i.e.,

$$\phi_{e_\infty} = \left(1 + \frac{\alpha^2}{2\alpha - \alpha^2}\right)\phi_v.$$

By taking $\alpha \to 0$, a first-order Taylor series approximation yields

$$\phi_{e_\infty} \approx \left(1 + \frac{1}{2}\alpha\right)\phi_v. \tag{7}$$

Hence, by choosing α nonzero and small leads to a limit error $\phi_{e_\infty} = \phi_v$, which intuitively corresponds to the optimal result, since the iteration-domain feedback (4) cannot attenuate v_j in iteration j. Note that $\alpha = 0$ turns of learning, in which case the trial-invariant error due to r is not compensated for. An alternative to attenuate ϕ_v is to re-design the controller C in (3), which should from a disturbance attenuation perspective be designed such that $\tilde{S} \approx \tilde{H}^{-1}$, as is advocated in [6]. Note that this affects J in (1).

Remark 4. *The results in this section rely on infinite time signals and LTI systems. Alternative ILC designs based on finite-time optimization [12], see also the forthcoming section, explicitly address the boundary effects, typically leading to an LTV ILC update* (2)*, even if J is LTI. In [60], it is shown that these optimization-based designs are equivalent to a certain linear-quadratic-tracking problem. As a result, the solution reaches a certain stationary value for sufficiently long task lengths, in which case an LTI L and Q can be derived for which the results of Theorem 3 apply. This also implies that the design of weighting filters for such optimization-based design can be further investigated using the results presented in this section, as is briefly summarized in the next section.*

4. Sparse ILC

In this section, the general optimization-based ILC framework is presented that allows for enforcing additional structure compared to alternative ILC structure. In fact, traditional norm-optimal ILC algorithms [12] are recovered as a special case. In the next subsection, the general framework is presented and motivated, followed by specific design choices in the subsequent sections.

4.1. General approach

Throughout, the criterion

$$\begin{aligned}\mathcal{J}(f_{j+1}) =& \frac{1}{2}\|W_e e_{j+1}\|_2^2 + \frac{1}{2}\|W_f f_{j+1}\|_2^2 \\ &+ \frac{1}{2}\|W_{\Delta f}\left(f_{j+1} - f_j\right)\|_2^2 + \lambda\|D f_{j+1}\|_1\end{aligned} \tag{8}$$

is considered. Here, finite time signals of length N are considered to obtain an optimization problem with a finite number of decision variables, i.e., $e_j, f_j \in \mathbb{R}^N$. The matrices are defined in the sequel and are assumed to have compatible dimensions. In addition, existence of a unique solution is typically assumed, which can be directly enforced by assuming appropriate positive (semi-) definiteness assumptions on the design variables W_e, W_f, $W_{\Delta f}$, D, and λ. Also, e_{j+1} in (8) is considered to be the noise-free prediction $e_{j+1} = r - Jf_{j+1}$. Since also r is unknown, the main idea in ILC is to use this approximation also for e_j, leading to

$$e_{j+1} = e_j - J(f_{j+1} - f_j), \tag{9}$$

where e_j is the measured error signal during trial j. Thus, substituting (9) into (8) renders the optimization problem as a function of the known variables e_j, f_j, user-defined variables, and the decision variable f_{j+1}.

The motivation for considering (8) is as follows. First, if $\lambda = 0$, then standard norm-optimal ILC is recovered, e.g., as in [23]. In this case, an analytic solution of the form (4) is directly obtained with

$$L = (J^T \bar{W}_e J + W_{\Delta f})^{-1} J^T \bar{W}_e \tag{10}$$

$$Q = (J^T \bar{W}_e J + \bar{W}_f + \bar{W}_{\Delta f})^{-1} (J^T \bar{W}_e J + \bar{W}_{\Delta f}), \tag{11}$$

where $\bar{W}_e = W_e^T W_e$, $\bar{W}_f = W_f^T W_f$, and $\bar{W}_{\Delta f} = W_{\Delta f}^T W_{\Delta f}$.

The second motivation stems from the observation that the terms $\frac{1}{2}\|W_f f_{j+1}\|_2^2$ and $\frac{1}{2}\|W_{\Delta f}\left(f_{j+1} - f_j\right)\|_2^2$ essentially involve a ridge regression or Tikhonov regularization. If $f_j = 0$, then the two terms coincide. If $f_j \neq 0$, i.e., during the ILC iterations, then W_f typically leads to $Q \neq I$ in (4), providing robustness with respect to modeling errors [11]. Increasing $W_{\Delta f}$ attenuates trial-varying disturbances, which is similar to reducing α in (7). Note that W_f penalizes the command input and hence to a certain extent mitigates the attenuation of trial-varying disturbances. However, it is highly recommended to use $W_{\Delta f}$ to decrease trial-varying disturbances, since the use of W_f leads to a non-zero limit error e_∞ in the presence of trial-invariant disturbances, which coincides with a $\tilde{Q} \neq 1$ in Theorem 3.

The third and main motivation for considering the extended criterion (8) is the additional term $\lambda\|Df_j\|_1$ that is used to enforce sparsity and structure. Note that sparsity is measured directly through the ℓ_0 norm. However, inclusion of an ℓ_0 penalty in the criterion (8) leads to a non-convex optimization problem, which in fact is NP-hard, see [37]. The ℓ_1 norm is a convex relaxation of the ℓ_0 norm. To see this, note that (8) is essentially in Lagrangian form. For the purpose of explanation, consider the simplified form by selecting $W_e = I$, $j = 1$, $f_0 = 0$, $W_f = 0$, $W_{\Delta f} = 0$, and $D = I$. Using (9)

$$\mathcal{J}(f_1) = \frac{1}{2}\|e_0 - Jf_1\|_2^2 + \lambda\|f_1\|_1, \tag{12}$$

which is equivalent to the primal optimization problem

$$\begin{aligned} \min_{f_1} \quad & \frac{1}{2}\|e_0 - Jf_1\|_2^2 \\ \text{subject to} \quad & \|f_1\|_1 \leq t. \end{aligned} \tag{13}$$

for the range of t where the constraint in (13) is active. This implies that for a given value of λ, there exists a value of t for which (12) and (13) have identical minima. In this simplified case, the interpretation in [50] applies to the ILC problem. In particular, the constraint in (13) is plotted in Figure 2 in addition to several elliptical contour lines of the objective function in (13). The solution to (13) corresponds to the smallest ellipsoid that touches the rhombus of the constraint. If this happens at the corner, as is the case in Figure 2, then one of the coefficients is zero and a sparse solution is obtained.

In contrast, traditional norm-optimal ILC, i.e., corresponding to the solution (10) - (11), typically does not lead to a sparse solution with zero entries in f_1. To see this, consider a similar simplified case as in (12)

$$\mathcal{J}(f_1) = \frac{1}{2}\|e_0 - Jf_1\|_2^2 + \tau\|f_1\|_2, \tag{14}$$

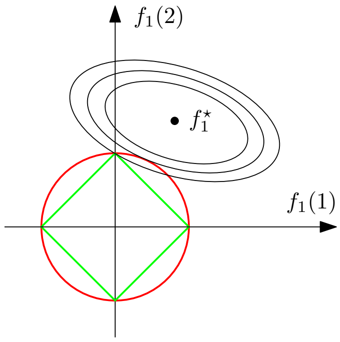

Figure 2: Enforcing sparsity in ILC. Assuming $N = 2$, f_1 contains two elements. The constraint set, i.e., the ℓ_1 ball is plotted in green. In addition, ellipsoidal contour lines corresponding to the objective in (13) are plotted. The optimal solution is found at the point where the contour line first touches the constraint set, which in this case implies $f_1(1) = 0$, hence f_1 is sparse. In contrast, in the ridge regression case of (15) (whose constraint is shown in red), the solution is not sparse. In particular, this solution is obtained when the contour lines of the objective function in (15) first touches the constraint set corresponding to the ℓ_2 ball. In addition, $f_1^\star$ denotes the unconstrained solution to the objective function in (13) and (15).

which is again in Lagrangian form. Here, τ directly relates to the weights in (8) if W_f and $W_{\Delta f}$ are selected as the common diagonal case with initialization $f_0 = 0$. The primal optimization problem corresponding to (14) is given by

$$\begin{aligned} \min_{f_1} \quad & \frac{1}{2}\|e_0 - Jf_1\|_2^2 \\ \text{subject to} \quad & \|f_1\|_2 \leq t. \end{aligned} \tag{15}$$

In Figure 2, the constraint is again shown together with the contour lines of the objective function. Due to the lack of corners of the constraint in this two-dimensional example, zeros only occur when the principal axes of the elliptical contour lines of the objective functions coincide with the axes of the command input. Hence, the ℓ_1 norm promotes sparse solutions, whereas the ℓ_2 norm in general does not.

The results in Figure 2 involve a two-dimensional example. From this example, it can also be observed why the ℓ_1 norm is chosen as a surrogate of the ℓ_0 norm, because it is the convex envelope of the ℓ_0 norm in the ℓ_2 ball, i.e., it is the largest convex function whose graph lies below that of the ℓ_0 norm in the set $\|f_1\|_2 \leq 1$.

A similar mechanism takes place in higher dimensions. Indeed, the ℓ_1 ball, in any finite number of dimensions, can be shown to be a polytope, i.e., described by a set of linear inequalities. In this case, the corners in Figure 2 correspond to extreme points, which is a 0-dimensional face. Here, a face is subset of F of a convex set such that if f_1 is in F and $f_1 = ay + (1 - a)z$, with y and z in the convex set and $a \in [0, 1]$, then a should be equal to 0 or 1. In the ℓ_1 ball, the faces correspond to sparse sets, i.e., to points most of whose entries are 0, so when the ellipsoid intercepts one of these faces, the solution is typically highly sparse. The higher dimensional case is further investigated in [50], [17] and [13].

Finally, it is remarked that if $\lambda > 0$, then the solution to (8) typically cannot be obtained in closed-form as in (10)-(11).

Interestingly, a unique solution to (8) exists due to convexity. The optimization problem (8) can be readily solved using general convex optimizers. In addition, several efficient algorithms have been developed, see, e.g., [25, Chapter 5] for an overview. Several of such algorithms provide the entire solution path as a function of λ. The particular algorithm depends on the choice of D, but several relevant choices are outlined below.

4.2. Sparse command signals via lasso

In view of requirement R3a in Section 2, in certain applications it is required to have a sparse command signal f_j. To this end, D in (8) can be selected as $D = I$. As a result, the value of $\lambda > 0$ will dictate the sparsity of the solution. In addition, in this classical lasso approach, W_f and $W_{\Delta f}$ may be selected as $W_f = 0$ and $W_{\Delta f} = 0$, i.e., traditional design guidelines for norm-optimal ILC regarding positive definiteness of these matrices, as in [23], need not be considered, even for the situation where J is singular. The resulting criterion becomes

$$\mathcal{J}(f_{j+1}) = \frac{1}{2}\|W_e(e_j - Jf_{j+1})\|_2^2 + \lambda\|f_{j+1}\|_1, \tag{16}$$

which closely reflects the original lasso approach in [50].

4.3. Elastic net lasso

In the lasso ILC approach in Section 4.2, the commonly used weighting matrices in W_f and $W_{\Delta f}$ are set to zero. Interestingly, by selecting either W_f or $W_{\Delta f}$ unequal to zero, an ILC algorithm that relates to the elastic net is obtained, see [59], which combines lasso and ridge regression. An important advantage is that the elastic net improves group sparsity, where several components become zero simultaneously. Notice that a drawback of the so-called naive elastic net, which coincides with $W_f \neq 0$, $W_{\Delta f} = 0$, is that it leads to a double shrinkage, and it benefits from a correction step [59]. In contrast, in ILC the alternative choice $W_f = 0$, $W_{\Delta f} \neq 0$ can be made, which enforces sparsity in addition to attenuating trial-varying disturbances, see Section 4.1.

4.4. Sparse updates via fused lasso

In view of Requirement R3b, it may be required that the signal f_j is not necessarily sparse but piecewise constant, i.e., its value is only changed occasionally in time. This requires a certain structure of signal, which is different than sparsity $\|f_j\|_0$. The main idea is to select D as

$$D_f = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}, \tag{17}$$

a choice which is also known as the fused lasso, e.g., [51] and leads to the criterion

$$\mathcal{J}(f_{j+1}) = \frac{1}{2}\|W_e(e_j - Jf_{j+1})\|_2^2 + \lambda\|D_f f_{j+1}\|_1. \tag{18}$$

Interestingly, the fused lasso (18) can be recast as a traditional lasso of the form (16), yet with an increment-input-output system description. To establish the connection, let $\tilde{J}^i = \tilde{J}(1 - z^{-1})$ be the increment-input-output system. Also, expand D_f in (17) as

$$D_f^i = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}.$$

Then, a change of variables

$$f_{j+1}^i = D_f^i f_{j+1},$$

where f_{j+1}^i denotes the incremental input, leads to

$$\mathcal{J}(f_{j+1}) = \frac{1}{2}\|W_e(e_j - J^i f_{j+1}^i)\|_2^2 + \lambda\|f_{j+1}^i\|_1,$$

with $J^i = J(D_f^i)^{-1}$ corresponding to $\tilde{J}^i$.

4.5. Sparse fused lasso

Up to this point, Requirement R3a and Requirement R3b have been addressed separately in Section 4.2 and Section 4.4, respectively. In certain applications, it may be desired to impose both Requirement R3a and Requirement R3b.

Interestingly, Requirement R3a and Requirement R3b can be enforced both by selecting

$$D = \begin{bmatrix} \alpha D_f & I \end{bmatrix}, \tag{19}$$

in (8). Here, the parameter λ can still be chosen to enforce sparsity, i.e., Requirement R3a, whereas the additional tuning parameter $\alpha \in \mathbb{R}_{\geq 0}$ enforces Requirement R3b. This leads to the so-called sparse fused lasso [52]. Note that additional requirements can easily be incorporated using a similar construction as (19).

4.6. Basis function ILC

In recent extensions to ILC, several basis functions are employed. On the one hand, wavelet basis functions are used in, e.g., [33]. These immediately fit in the formulation (8), see also [52, Section 2.1.3], enabling a systematic way for thresholding while explicitly addressing the performance criterion.

On the other hand, flexibility to varying reference signals is achieved by employing basis functions that depend on the reference. In particular, the command signal is parameterized as $f_{j+1} = \Psi(r)\theta_{j+1}$, see, e.g., [54], [34], [8], [61]. The proposed framework can be employed to minimize the number of required basis functions. For instance, a large set can be postulated, e.g., following the guidelines in [31]. Next, an alternative formulation of (8) can be considered, e.g.,

$$\begin{aligned} \min_{\theta_{j+1}} \quad & \|\theta_{j+1}\|_1 \\ \text{subject to} \quad & \frac{1}{2}\|W_e e_{j+1}\|_2^2 + \frac{1}{2}\|W_f \Psi(r)\theta_{j+1}\|_2^2 \\ & + \frac{1}{2}\|W_{\Delta f}\Psi(r)\left(\theta_{j+1} - \theta_j\right)\|_2^2 \leq t, \end{aligned}$$

where a suitable value of t can be obtained by solving the standard norm-optimal ILC in (10)-(11).

4.7. Extensions, analysis, and discussion

A general framework for enforcing sparsity and structure in iterative learning control has been proposed, and several specific choices have been outlined. The main idea is that the proposed framework, involving criterion (8), leads to a command signal f_j that is either sparse, i.e., $\|f_j\|_0$ is small, or has structure, i.e., a piecewise constant f_j with a small number of jumps. This is achieved by the lasso and fused lasso, respectively. This structure has several important benefits. First, this leads to a reduced implementation cost in terms of sampling or communication requirements due to highly structured command signals. Second, it substantially reduces the amplification of trial-varying disturbances. From the perspective of ILC, the proposed sparse ILC framework essentially generates basis functions [54], which becomes explicit in Section 4.7.1. In turn, these weighting filters enable which leads to a reduction of trial-varying disturbances [8]. Also, from [8], it is clear that the basis has to be sufficiently rich, which is investigated in more detail in Section 4.7.2. Third, this idea of automated basis function selection can also be combined with earlier ideas on using basis functions, i.e., to select them from a pre-specified library $\Psi(r)$, see Section 4.6.

From another perspective, the use of sparsity can also be seen as a time-domain filtering operation. Indeed, in typical ILC designs, frequency-domain specifications are used, e.g., to account for trial-varying disturbances, see [48] and [42]. This also relates to the idea of the static learning gain α in (7). The proposed framework essentially provides a learning gain α that turns off learning in regions where it does not contribute significantly to attenuating trial-invariant disturbances, in view of not amplifying trial-varying disturbances.

Further extensions that are beyond the scope of the present paper but can be directly incorporated include group lasso [56], adaptive lasso [58], reweighted ℓ_1 [16], and the use of nonconvex penalties [4].

4.7.1. Re-estimation for debiasing

Note that the lasso shrinks the estimate compared to the least-squares terms in (8). Through a re-estimation step of the nonzero coefficients, debiasing is obtained. Note that in certain cases, the bias helps to obtain a smaller overall error, i.e., including both bias and variance aspects, which closely relates to the well-known Stein estimator [49]. However, for ILC such a bias is undesired, since it is automatically eliminated by performing iterations, see Theorem 2. Thus, it is expected that as the ILC iterations increase, the advantages of re-estimating for debiasing become more important. Similar re-estimation steps are proposed in [46], [47], [36, Page 439], [30, Section 7.1]. Interestingly, in the context of ILC, the idea of enforcing sparsity followed by a re-estimation step essentially has the same role as a Q-filter in traditional ILC, see [5] for details. Finally, it is pointed out that the re-estimation essentially involves a standard norm-optimal ILC design, i.e., (8) with $\lambda = 0$, in conjunction with a command input f_{j+1} with reduced dimension. These can in fact be solved efficiently, i.e., linearly in the dimension of the reduced command input, see [60] for details.

4.7.2. Sparse signal recovery

The main motivation for using the ℓ_1 norm in (8) essentially is to provide a convex relaxation of the ℓ_0 norm. In case the optimal command input, i.e., for $j \to \infty$ and $v_j = 0$, the signal f_∞ that minimizes $\mathcal{J}(f_\infty)$, is sparse, a relevant question is whether this optimal sparse vector can be recovered using the formulation (8). The answer depends on the sparsity of the underlying optimal command input f_j, as well as on the matrix J. In [17], a sufficient condition that relies on the restricted isometry property is provided. However, these conditions are violated for many practical cases. Nonetheless, the formulation (8) provides an effective way to enforce sparsity.

4.7.3. Monotonic convergence

Monotonic convergence is a commonly used requirement for practical applications. Indeed, it is well-known that poorly designed ILC algorithms can lead to a significant learning transient. It is well-known that traditional norm-optimal ILC, i.e., setting $\lambda = 0$ in (8), is monotonically convergent in f_j, see, e.g., [11], where the usual assumption $v_j = 0$ is tacitly assumed to analyze monotonic convergence. However, if $\lambda > 0$, the criterion (8) involves multiple norms, i.e., both the ℓ_1 and the ℓ_2 norm. As a result, monotonic convergence requires a more detailed analyis.

To proceed, consider for instance the elastic net lasso of Section 4.3 with $D = I$, $W_f = 0$, $W_{\Delta f} > 0$. In this case, monotonic convergence of the ILC cannot be guaranteed in general if $\lambda = 0$. Interestingly, in this case the criterion (8) can be recast as

$$\begin{aligned}\mathcal{J}(f_{j+1}) =& \frac{1}{2}\left\| \left(\begin{bmatrix} W_e e_j \\ 0 \end{bmatrix} + \begin{bmatrix} W_e \\ W_{\Delta f} \end{bmatrix} f_j \right) - \begin{bmatrix} W_e \\ W_{\Delta f} \end{bmatrix} f_{j+1} \right\|_2^2 \\ &+ \lambda \|f_{j+1}\|_1 .\end{aligned} \tag{20}$$

Next, there exists a value of τ such that the optimization problem

$$\begin{aligned}\min_{f_{j+1}} \quad & \|f_{j+1}\|_1 \\ \text{subject to} \quad & \frac{1}{2}\left\| \begin{bmatrix} W_e e_j \\ 0 \end{bmatrix} - \begin{bmatrix} W_e \\ W_{\Delta f} \end{bmatrix} (f_{j+1} - f_j) \right\|_2^2 \leq \tau.\end{aligned} \tag{21}$$

has an identical solution as (20) at a certain iteration j. By fixing τ during the iterations, the criterion (22) can be directly used to enforce monotonic convergence of f_j in the ℓ_1 norm. Alternatively, the formulation

$$\begin{aligned}\min_{f_{j+1}} \quad & \frac{1}{2}\left\| \begin{bmatrix} W_e e_j \\ 0 \end{bmatrix} - \begin{bmatrix} W_e \\ W_{\Delta f} \end{bmatrix} (f_{j+1} - f_j) \right\|_2^2 \\ \text{subject to} \quad & \|f_{j+1}\|_1 \leq \beta\end{aligned} \tag{22}$$

can be considered for a fixed β, which enforces monotonic convergence in the ℓ_2 norm.

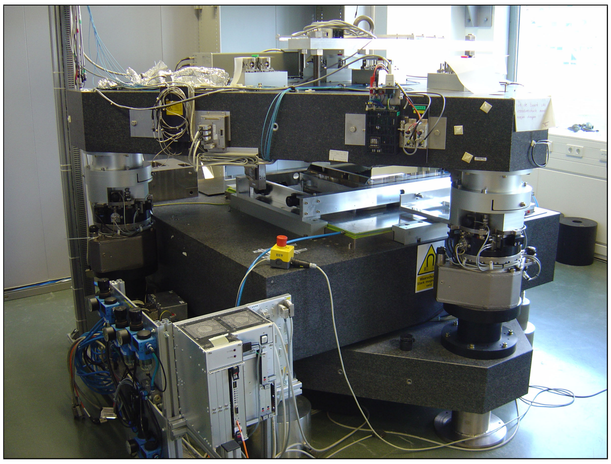

Figure 3: Considered wafer stage application.

5. Application to a Wafer Stage

5.1. Setup

The considered system is a wafer stage, see Figure 3. Wafer stages are positioning systems that are used in the production of integrated circuits (ICs) through a photolithographic process. The considered wafer stage is controlled in all six motion degrees-of-freedom, i.e., three translations and three rotations. The system is a dual-stage system, where the long stroke enables a stroke of 1 m in the horizontal plane, whereas the short stroke enables a positioning accuracy of 1 nm. Further details on the system and the considered actuation and sensor system are provided in [40]. Throughout, a sampling frequency of 1 kHz is adopted, as in [39].

To enable a detailed comparison between the various approaches in Section 4, the identified model in [39] is considered as true system, i.e., the result as described in [39] is denoted G_o. In addition, the feedback controller designed in [39] is adopted to stabilize the system. In Figure 4, the open-loop G_o and closed-loop S_oG_o are depicted. In addition, a closed-loop model is made, where a model error is introduced by selecting $J = 0.7S_oG_o$. This model error is introduced to investigate robust convergence properties of ILC. The resulting model J is also depicted in Figure 4.

The additive noise $\tilde{v}_j$ is zero mean white noise with a normal distribution and variance $\lambda_e = 1.5 \cdot 10^{-7}$. As a result, H in Assumption 1 has transfer function $\tilde{H} = \frac{1}{1+\tilde{G}\tilde{C}}$.

The task r is shown in Figure 5, which is a position signal. In addition, the corresponding scaled acceleration profile is depicted, which is expected to constitute the main contribution of f_j [31], [34]. For the considered wafer stage application in Figure 3, the constant velocity phase is most important for performance, see [15, Figure 16 and Figure 20], which takes place between 0.03 s and 0.24 s.

The goal of this section is to illustrate and compare the proposed approaches in Section 4. The reference situation, i.e., feedback only with $f_0 = 0$ in Figure 1 is shown in Figure 6 (solid blue), Figure 7, and Figure 8. In particular, the approaches in Section 4 are applied in this section.

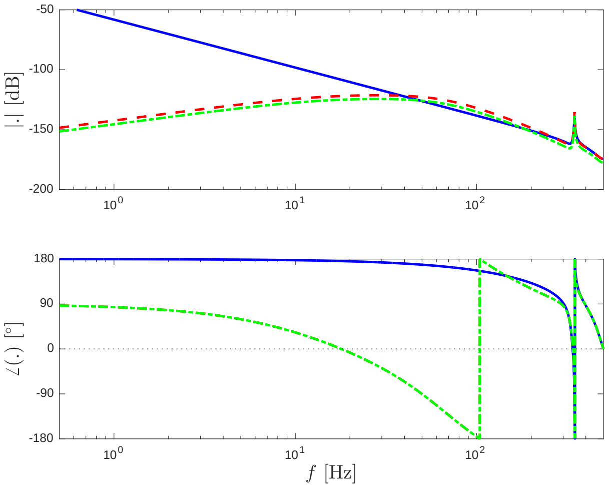

Figure 4: Open-loop true system G_o in (solid blue), closed-loop true system S_oG_o (dashed red), closed-loop model J (dash-dotted green).

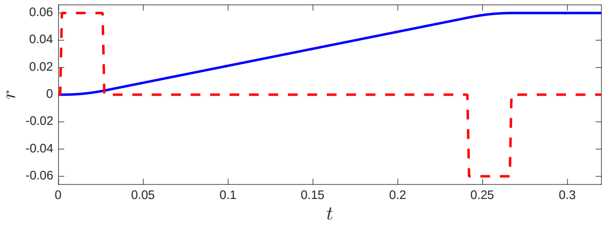

Figure 5: Reference r in (1) (solid blue), scaled acceleration profile (dashed red).

5.2. Traditional Norm-Optimal ILC

First, the traditional norm-optimal ILC solution is implemented with $\lambda = 0$ in (8) with the analytic solution (10)-(11). Here, $W_e = I$, $W_f = 0$, and $W_{\Delta f} = 10^{-10}$. Notice that $W_{\Delta f}$ is relatively small but nonzero, since a nonzero $W_{\Delta f}$ or W_f is required to enforce a unique optimal solution.

The results after 40 iterations are depicted in Figure 6. Clearly, the error is reduced to a very small value. As is expected, the feedforward is nonzero at every time instant and very noisy.

To further analyze these results, the 2 norm of the stochastic, i.e., trial-varying, part of the error is computed as $\sqrt{\sum_{t=1}^{N}(e_j(t) - \hat{e}_\infty(t))^2}$, see Figure 7. Here, $\hat{e}_\infty$ is computed as follows. After a sufficient number of iterations n_{conv}, the ILC algorithm is assumed to have converged, after which an additional iterations n_{iter} is used to compute $\hat{e}_\infty = \frac{1}{n_{\text{iter}}} \sum_{j=n_{\text{conv}}}^{n_{\text{conv}}+n_{\text{iter}}-1} e_j$. Clearly, Figure 7 reveals that the trial-varying part of the error is amplified by a factor 2, which corroborates the result of Theorem 3, where $Q \approx 1$ due to the specific selection of weighting filters.

To further investigate the amplification of trial-varying disturbances, the spectrum of the trial-varying part of the errors in Figure 7 is estimated, see Figure 8. In addition, the spectrum $\phi_v = \left|\frac{1}{1+\tilde{G}\tilde{C}}\right|^2 \lambda_e$ is computed, as well as $2\phi_v$. Again, this clearly confirms the result of Theorem 3. In particular, the presented ILC approach with $\lambda = 0$ and small W_f and $W_{\Delta f}$ leads to a

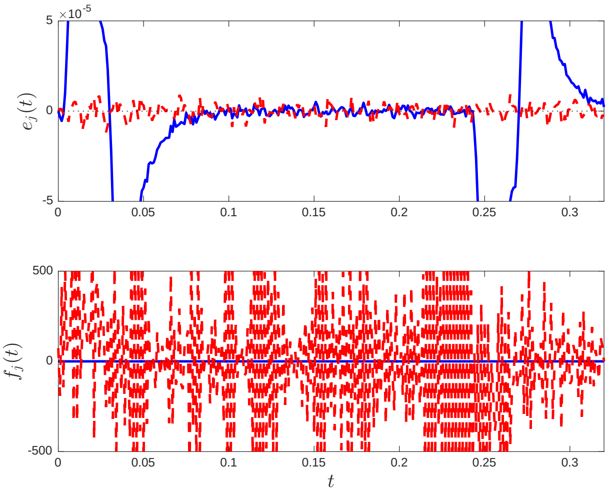

Figure 6: Top: error e_j. Bottom: command signal f_j. Shown are iteration $j = 0$ (solid blue), iteration $j = 40$ for traditional norm-optimal ILC of Section 5.2 (dashed red).

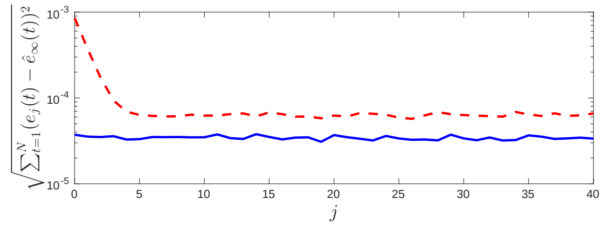

Figure 7: Trial-varying part of the error signal $\sqrt{\sum_{t=1}^{N}(e_j(t) - \hat{e}_\infty(t))^2}$ for feedback only, i.e., $f_j = 0, j = 0, 1, 2, \ldots$ (solid blue) and traditional norm-optimal ILC of Section 5.2 (dashed red).

perfect attenuation of trial-invariant disturbances. However, it amplifies trial-varying disturbances by a factor two, and leads to an f_j with large $\|f_j\|_0$, violating Requirement R3a, as well as R3b.

Summarizing, the results in Figure 6, Figure 7, and Figure 8 confirm that norm-optimal ILC amplifies trial-varying disturbances, and leads to a non-sparse solution in view of Requirement R3a and Requirement R3b.

5.3. Lasso ILC

To address Requirement R3a, the approach in Section 4.2 is applied. In particular, $W_e = I$, $W_f = 0$, and $W_{\Delta f} = 0$, $D = I$, and $\lambda = 5 \cdot 10^{-9}$. Next, the ILC iteration is started, and after 40 iterations it leads to e_{40} and f_{40} in Figure 11. Interestingly, $\|f_{40}\|_0$ is much smaller for the lasso ILC approach compared to the results of Section 5.2, as is confirmed in Figure 9, thereby addressing Requirement R3a.

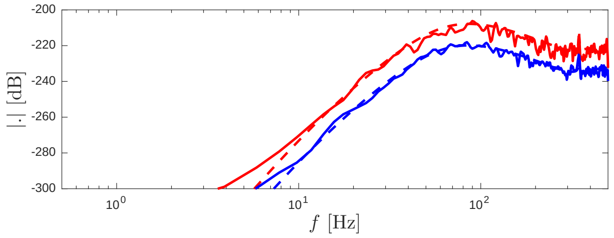

Figure 8: Estimated spectrum of trial-varying part of the error without ILC (solid blue) and for traditional norm-optimal ILC of Section 5.2 (solid red). Also shown are the spectra $2\phi_v$ (dashed blue) and ϕ_v (dashed red).

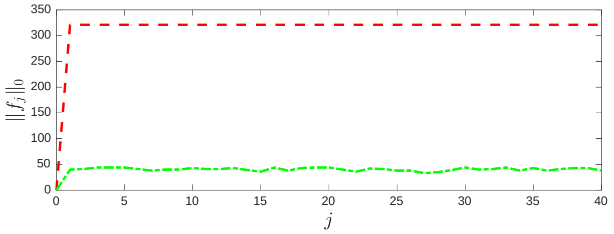

Figure 9: ℓ_0 norm of the command input f_j for norm-optimal ILC of Section 5.2 (dashed red) and lasso ILC of Section 5.3 (dash-dotted green), which leads to a reduced error signal.

Also, the 2 norm of the error signal is computed, see Figure 10. Clearly, the error reduces significantly over the iterations. Finally, also the re-estimated lasso, as is explained in Section 4.7, is implemented. The results are also depicted in Figure 10. Interestingly, it can be observed that re-estimating leads to a smaller limit error, as is expected. However, note that during the iterations, the approach of Section 4.2 leads to a smaller error compared to the re-estimated version in several of the initial iterations. An explanation for this aspect is that the biased estimate leads to a smaller overall error, which is a similar effect as in the Stein estimator. Hence, it is concluded that for non-iterative approaches, the biased estimate can be useful in terms of a bias/variance trade-off, but in the iterative schemes the benefit of re-estimation is clearly confirmed in Figure 11.

5.4. Elastic net lasso ILC

In this section, the approach of Section 4.3 is pursued, where the lasso regularisation is extended with a ridge regression term. In particular, $W_f = 0$, while $W_{\Delta f} = 1 \cdot 10^{-6} I$. The resulting error e_{40} and command input f_{40} are depicted in Figure 12. The error is of comparable magnitude as the lasso ILC in Section 11, while the command input is substantially smoother. The error in fact has slightly reduced compared to lasso ILC, as is shown in Figure 10, which comes at the price of a slower convergence rate due to an increased $W_{\Delta f}$. Notice that the elastic net lasso can also be improved by re-estimation, which is not done here to facilitate the presentation.

5.5. Fused lasso ILC

The results in the previous sections have addressed Requirement R3a. In certain situations, e.g., wireless sensors or embedded implementations, it may be required to minimize the number of times the command input is updated, i.e., Requirement R3b. This is a different form of structure compared to sparsity. To address this, the fused lasso of Section 4.4 is employed.

In particular, in the general criterion (8) is considered, where the weighting filters are selected as $W_f = W_{\Delta f} = 0$, $D = D_f$ in (17), and $\lambda = 3 \cdot 10^{-12}$.

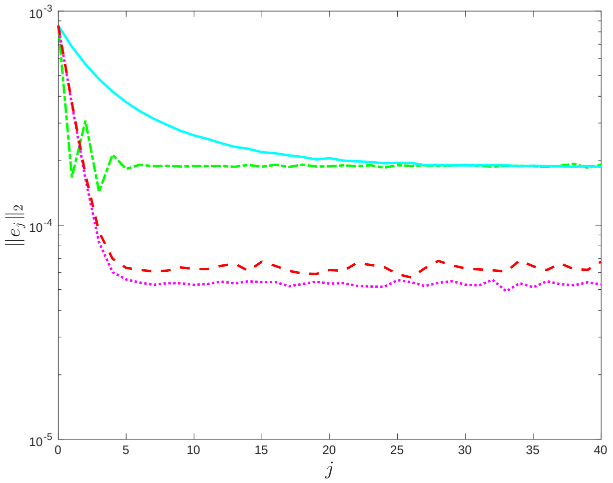

Figure 10: Computed ℓ_2 norm of the error for various ILC algorithms. Traditional norm-optimal ILC in Section 5.2 (dashed red) leads to a significant error reduction in the initial iterations, and then remains at a certain level due to the amplification of trial-varying disturbances. The lasso approach of Section 5.3 with re-estimation (dotted magenta) leads to a significant reduction in the initial iterations, in addition to a reduced limit error, since it reduces amplification of trial-varying disturbances. Also note that the lasso approach without re-estimation (dash-dotted green) leads to an improved estimate in the first iteration, yet remains at a large error after convergence, which is due to the bias error in the solution. Finally, the elastic-net lasso approach of Section 5.4 is shown (solid cyan), which leads to a comparable converged performance as the lasso ILC (dash-dotted green), since both do not include re-estimation in this case.

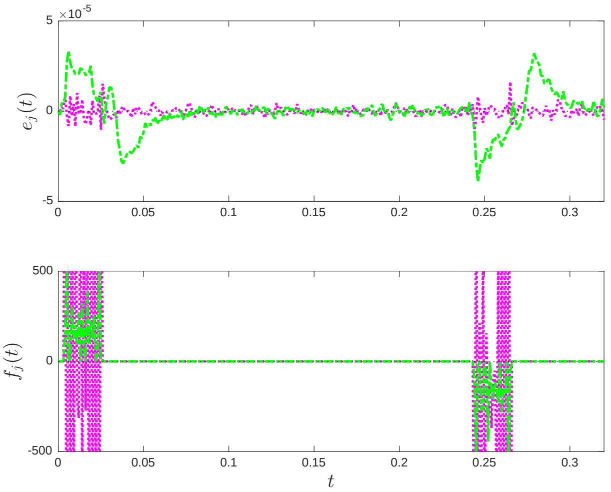

Figure 11: Top: error e_{40} at iteration $j = 40$. Bottom: command signal f_{40} at iteration $j = 40$. Shown are lasso ILC of Section 5.3 (dash-dotted green) and re-estimated lasso ILC of Section 5.3 (dotted magenta).

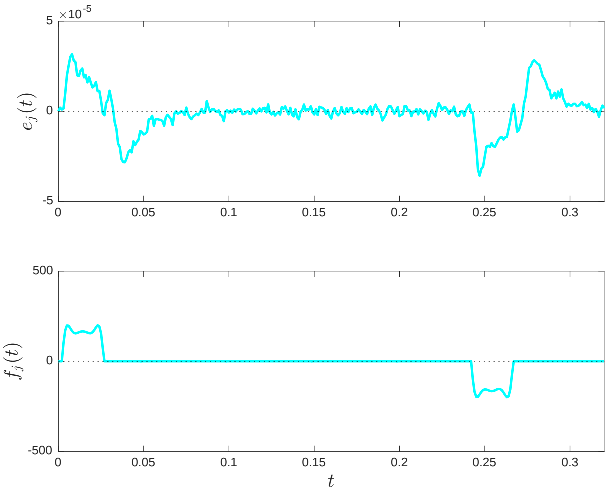

Figure 12: Top: error e_{40} at iteration $j = 40$. Bottom: command signal f_{40} at iteration $j = 40$. Shown is the elastic net lasso ILC of Section 5.4 (solid cyan), leading to a smooth command input f_j.

Next, the ILC iteration is invoked. The results are shown in Figure 13. Compared to the results of Figure 11 in Section 5.3, the error has reduced significantly. However, this comes at the price of sparsity. Indeed, only the first samples are zero, since the algorithm is initialized with $f_1(0) = 0$. Interestingly, only a limited number of command signal updates are required to achieve a small error signal. This will also attenuate the effect of trial-varying disturbances. Note that the error can be further reduced by including a re-estimation step, which is not shown here to facilitate the presentation.

5.6. *Sparse fused lasso ILC*

In the previous sections, Requirement R3a and Requirement R3b are achieved separately in Section 5.3 and Section 5.5, respectively. To address both requirements simultaneously, the sparse fused lasso approach of Section 4.5 is adopted. The regularization penalties in (19) are selected such that these essentially combine the two penalties in Section 5.3 and Section 5.5.

The results are depicted in Figure 14. It can directly be observed that it combines the sparsity of Section 5.3 while at the same time reducing the number of command signal updates as in Section 5.5. As such, it is concluded that the sparse fused lasso addresses Requirement R3a and Requirement R3b simultaneously. The relative penalties can be further tuned to balance the importance of both penalties, as well as the resulting error signal. In addition, the resulting error signal can be further enhanced through a re-estimation step.

6. Conclusion

A general framework is presented that extends optimization-based iterative learning control to include additional structure, including sparsity. The approach is shown on a mechatronic system, where it is shown to have significant benefits, including i) resource-efficiency in terms of sparse command signals, e.g.,

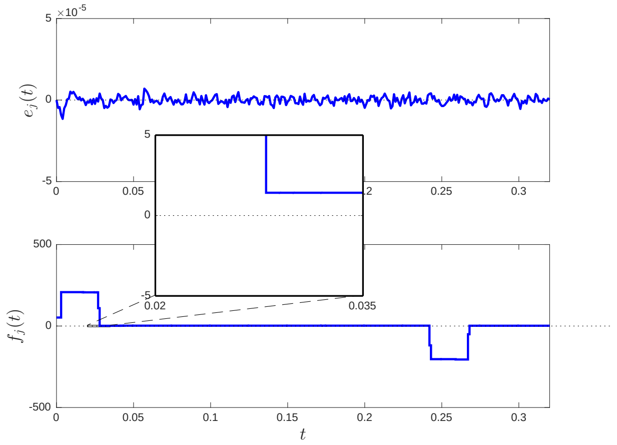

Figure 13: Top: error e_{40} at iteration $j = 40$. Bottom: command signal f_{40} at iteration $j = 40$. Shown is the fused lasso ILC of Section 5.5 (solid cyan), leading to a command input f_j that addresses Requirement R3b. In particular, the command signal f_{40} aims to minimize the error signal in addition to the updates, i.e., instants where f_{40} changes as a function of time. This does not explicitly address sparsity of f_{40} itself, as can be clearly observed from the zoom plot.

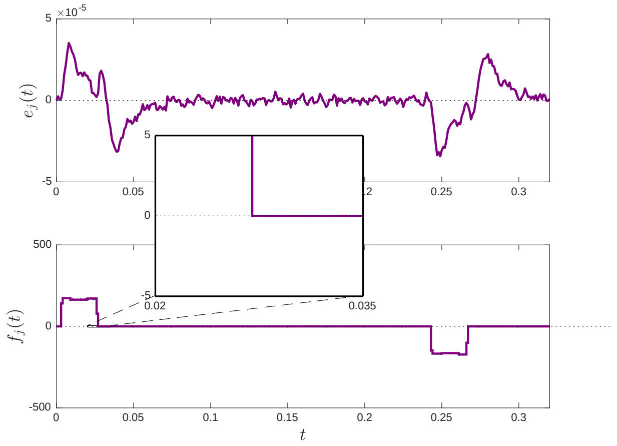

Figure 14: Top: error e_{40} at iteration $j = 40$. Bottom: command signal f_{40} at iteration $j = 40$. Shown is the sparse fused lasso ILC of Section 5.6 (solid cyan), leading to a command input f_j that addresses Requirement R3b. In comparison to the fused lasso approach in Section 5.5, here additional regularization parameters enforce a zero input signal, which can clearly be seen in the zoom plot.

facilitating embedded controller implementations; ii) resource-efficiency in terms of limiting the number of changes in the command signal, e.g., facilitating implementation in limited-capacity communication networks; iii) automated basis function selection in flexible iterative learning control employing basis functions; and iv) attenuation of trial-varying disturbances, which for the considered wafer scanner example leads to significant performance increase. Regarding the latter, a detailed analysis of trial-varying disturbances in ILC reveals that such trial-varying exogenous signals are often amplified by typical ILC algorithms. The proposed framework enables a significant reduction of this amplification, typically up to a factor of two.

The proposed framework enables many user-specific choices, and can be easily extended. For instance, for third-order or higher-order setpoints, it may be useful to impose regularization parameters of equally high polynomial orders, known as polynomial trend filtering [52, Section 2.1.2], which is a special case of the general criterion (8).

Ongoing research focusses on specialized algorithms for the considered scenarios, enabling faster computation. In addition, the correlation between variables is subject of further investigation. Finally, various aspects of monotonic convergence, which has here been analyzed in terms of the ℓ_1 norm, are being investigated, including robust monotonic convergence conditions [55], [18] and data-driven ILC frameworks [28], [7]. Finally, connections to recent frequency-domain designs, including [48] and [42], where convex optimization is used to design L and Q filters in certain frequency ranges, e.g., to attenuate trial-varying disturbances, could enable a further improvement in the design of ILC controllers.

Appendix

In this section, a proof of Theorem 3 is provided. Several auxiliary results are presented. In particular, note that at iteration j, the error is a function of all previous signals affecting the loop due to the iteration-domain integrator in (4). In the following lemma, the summation of j terms of the trial-invariant disturbance r in (1) is eliminated.

Lemma 5. *Consider the system* (1) *and ILC update* (4) *with* $f_0 = 0$ *and assume that iteration is stable in the sense of Theorem 2. Then,*

$$\begin{aligned} e_j = &\left(1 - \tilde{J}\frac{1-(\tilde{Q}(1-\tilde{L}\tilde{J}))^j}{1-\tilde{Q}(1-\tilde{L}\tilde{J})}\tilde{Q}\tilde{L}\right) r \\ &- v_j - \tilde{J}\sum_{n=0}^{j-1}(\tilde{Q}(1-\tilde{L}\tilde{J})^n\tilde{Q}\tilde{L}v_{j-n-1}. \end{aligned} \tag{23}$$

Proof. Substituting (1) into (4) yields

$$f_{j+1} = \tilde{Q}((1-\tilde{L}\tilde{J})f_j + \tilde{L}r - \tilde{L}v_j).$$

Given $f_0 = 0$ and subsequent successive substitution yields $f_1 = \tilde{Q}(\tilde{L}r - \tilde{L}v_0)$, $f_2 = \tilde{Q}((1-\tilde{L}\tilde{J})+1)\tilde{Q}\tilde{L}r - \tilde{Q}(1-\tilde{L}\tilde{J})\tilde{Q}\tilde{L}v_0 - \tilde{Q}\tilde{L}v_1$, and hence

$$f_j = \sum_{i=0}^{j-1} (\tilde{Q}(1 - \tilde{L}\tilde{J}))^i \tilde{Q}\tilde{L}r - \sum_{n=0}^{j-1} (\tilde{Q}(1 - \tilde{L}\tilde{J}))^n \tilde{Q}\tilde{L}v_{j-1-n}.$$

Next, using the geometric series

$$\sum_{l=0}^{j-1} r^l = \frac{1 - r^j}{1 - r}, \tag{24}$$

this leads to

$$f_j = \frac{1 - (\tilde{Q}(1 - \tilde{L}\tilde{J}))^j}{1 - \tilde{Q}(1 - \tilde{L}\tilde{J})} \tilde{Q}\tilde{L}r - \sum_{n=0}^{j-1} (\tilde{Q}(1 - \tilde{L}\tilde{J}))^n \tilde{Q}\tilde{L}v_{j-1-n}. \tag{25}$$

Finally, substitution of (25) into (1) yields the desired result (23). □

The result (23) reveals that the error contains a summation over j trial-varying disturbance terms v_j, whereas the influence of the trial-invariant disturbances is captured in a single term through the use of a geometric series. Although the trial-varying disturbance varies on each experiment, a closed-form expression can be obtained by exploiting Assumption 1.

Lemma 6. *Let Assumption 1 hold. Then, under the assumptions of Lemma 5,*

$$\begin{aligned}\phi_{e_j} &= \left| 1 - \tilde{J} \frac{1 - (\tilde{Q}(1 - \tilde{L}\tilde{J}))^j}{1 - \tilde{Q}(1 - \tilde{L}\tilde{J})} \tilde{Q}\tilde{L} \right|^2 \phi_r \\ &+ \left(1 + \left| \tilde{J}\tilde{Q}\tilde{L} \right|^2 \frac{1 - |\tilde{Q}(1 - \tilde{L}\tilde{J})|^{2j}}{1 - |\tilde{Q}(1 - \tilde{L}\tilde{J})|^2} \right) \phi_v\end{aligned} \tag{26}$$

Proof. Taking spectra yields

$$\begin{aligned}\phi_{e_j} &= \left| 1 - \tilde{J} \frac{1 - (\tilde{Q}(1 - \tilde{L}\tilde{J}))^j}{1 - \tilde{Q}(1 - \tilde{L}\tilde{J})} \tilde{Q}\tilde{L} \right|^2 \phi_r \\ &+ \left(1 + \left| \tilde{J}\tilde{Q}\tilde{L} \right|^2 \sum_{n=0}^{j-1} \left| (\tilde{Q}(1 - \tilde{L}\tilde{J}))^n \right|^2 \right) \phi_v.\end{aligned}$$

Next, using (24) yields the desired result (26). □

The closed-form solution (26) enables a direct proof of Theorem 3.

Proof. (of Theorem 3) Taking the limit $j \to \infty$ implies that $|(1 - \tilde{L}\tilde{J}))^j| \to 0$, directly leading to the desired result (5). □

Acknowledgements

This paper is the result of several research visits of both authors, which is supported in part of the research programme VENI with project number 13073, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO). In addition, the authors gratefully acknowledge the fruitful discussions with Jurgen van Zundert, Maurice Heemels, Dip Goswami, and Martijn Koedam for resource-efficient control, as part of the Robust Cyber-Physical Systems (RCPS) project (no. 12694).

[1] H.-S. Ahn, K. L. Moore, and Y. Chen. *Iterative Learning Control: Robustness and Monotonic Convergence for Interval Systems*. Communications and Control Engineering Series. Springer-Verlag, London, UK, 2007.

[2] M. Annergren, A. Hansson, and B. Wahlberg. An ADMM algorithm for solving ℓ_1 regularized MPC. In *Proc. 2012 Conf. Dec. Contr.*, pages 4486–4491, Maui, HI, USA, 2012.

[3] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. *Optimization with Sparsity-Inducing Penalties*, volume 4 of *Foundations and Trends in Machine Learning*. 2011.

[4] D. Bertsimas, A. King, and R. Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2), 2016.

[5] F. Boeren, A. Bareja, T. Kok, and T. Oomen. Frequency-domain ILC approach for repeating and varying tasks: With application to semiconductor bonding equipment. *IEEE Trans. Mech.*, 21(6):2716–2727, 2016.

[6] F. Boeren, D. Bruijnen, and T. Oomen. Enhancing feedforward controller tuning via instrumental variables: With application to nanopositioning. *Int. J. Contr.*, 90(4):746–764, 2017.

[7] J. Bolder, S. Kleinendorst, and T. Oomen. Data-driven multivariable ILC: Enhanced performance by eliminating L and Q filters. *Int. J. Rob. Nonlin. Contr.*, To appear.

[8] J. Bolder, T. Oomen, S. Koekebakker, and M. Steinbuch. Using iterative learning control with basis functions to compensate medium deformation in a wide-format inkjet printer. *Mechatronics*, 24(8):944–953, 2014.

[9] J. Bolder, J. van Zundert, S. Koekebakker, and T. Oomen. Enhancing flatbed printer accuracy and throughput: Optimal rational feedforward controller tuning via iterative learning control. *IEEE Trans. Ind. Electr.*, 64(5):4207–4216, 2017.

[10] L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.

[11] D. A. Bristow. Weighting matrix design for robust monotonic convergence in norm optimal iterative learning control. In *Proc. 2008 Americ. Contr. Conf.*, pages 4554–4560, Seattle, WA, USA, 2008.

[12] D. A. Bristow, M. Tharayil, and A. G. Alleyne. A survey of iterative learning control: A learning-based method for high-performance tracking control. *IEEE Contr. Syst. Mag.*, 26(3):96–114, 2006.

[13] P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data*. Springer Series in Statistics. Springer, Heidelberg, Germany, 2011.

[14] M. Butcher and A. Karimi. *Advances in the Theory of Control, Signals and Systems with Physical Modeling*, chapter Iterative Learning Control Using Stochastic Approximation Theory with Application to a Mechatronic System, pages 49–64. Number 407 in LNCIS. Springer, 2011.

[15] H. Butler. Position control in lithographic equipment an enabler for current-day chip manufacturing. *IEEE Contr. Syst. Mag.*, 31(5):28–47, 2011.

[16] E. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.*, 14:877–905, 2008.

[17] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[18] T. Duy Son, G. Pipeleers, and J. Swevers. Robust monotonic convergent iterative learning control. *IEEE Trans. Automat. Contr.*, 61(4):1063–1068, 2016.

[19] F. Felici and T. Oomen. Enhancing current density profile control in tokamak experiments using iterative learning control. In *Proc. 54th Conf. Dec. Contr.*, pages 5370–5377, Osaka, Japan, 2015.

[20] C. Freeman, A.-M. Hughes, J. Burridge, P. Chappell, P. Lewin, and E. Rogers. Iterative learning control of FES applied to the upper extremity for rehabilitation. *Contr. Eng. Prac.*, 17(3):368–381, 2009.

[21] M. Gallieri. *ℓ_{asso}-MPC - Predictive Control with ℓ_1-Regularised Least Squares*. Springer Theses. Springer, Switzerland, 2015.

[22] K. Goossens, A. Azevedo, K. Chandrasekar, M. Dev Gomony, S. Goossens, M. Koedam, Y. Li, D. Mirzoyan, A. Molnos, A. Beyranvand Nejad, A. Nelson, and S. Sinha. Virtual execution platforms for mixed-time-criticality systems: The CompSOC architecture and design flow. *SIGBED Review*, 10(3):23–34, 2014.

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

[24] S. Gunnarsson and M. Norrlöf. On the disturbance properties of high order iterative learning control algorithms. *Automatica*, 42:2031–2034, 2006.

[25] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with*

Sparsity: The Lasso and Generalizations. CRC Press, 2015.

[26] D. J. Hoelzle, A. G. Alleyne, and A. J. Wagoner Johnson. Basis task approach to iterative learning control with applications to micro-robotic deposition. *IEEE Trans. Contr. Syst. Techn.*, 19(5):1138–1148, 2011.

[27] D. J. Hoelzle and K. L. Barton. On spatial iterative learning control via 2-D convolution: Stability analysis and computational efficiency. *IEEE Trans. Contr. Syst. Techn.*, 24(4):1504–1512, 2016.

[28] P. Janssens, G. Pipeleers, and J. Swevers. A data-driven constrained norm-optimal iterative learning control framework for LTI systems. *IEEE Trans. Contr. Syst. Techn.*, 21(2):546–551, 2013.

[29] S. Khoshfetrat Pakazad, H. Ohlsson, and L. Ljung. Sparse control using sum-of-norms regularized model predictive control. In *Proc. 2013 Conf. Dec. Contr.*, pages 5758–5763, Firenze, Italy, 2013.

[30] S.-J. Kim, W. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 trend filtering. *SIAM Review*, 51(2):339–360, 2009.

[31] P. Lambrechts, M. Boerlage, and M. Steinbuch. Trajectory planning and feedforward design for electromechanical motion systems. *Contr. Eng. Prac.*, 13:145–157, 2005.

[32] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, NJ, USA, second edition, 1999.

[33] R. Merry, R. van de Molengraft, and M. Steinbuch. Iterative learning control with wavelet filtering. *Int. J. Rob. Nonlin. Contr.*, 18(10):1052–1071, 2008.

[34] S. van der Meulen, R. L. Tousain, and O. H. Bosgra. Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer. *J. Dyn. Syst., Meas., and Contr.*, 130:051006–1, 2008.

[35] K. L. Moore. *Iterative Learning Control for Deterministic Systems*. Springer-Verlag, London, UK, 1993.

[36] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[37] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, 1995.

[38] H. Ohlsson, L. Ljung, and S. Boyd. Segmentation of ARX-models using sum-of-norms regularization. *Automatica*, 46(6):1107–1111, 2010.

[39] T. Oomen. Controlling aliased dynamics in motion systems? An identification for sampled-data control approach. *Int. J. Contr.*, 87(7):1406–1422, 2014.

[40] T. Oomen, R. van Herpen, S. Quist, M. van de Wal, O. Bosgra, and M. Steinbuch. Connecting system identification and robust control for next-generation motion control of a wafer stage. *IEEE Trans. Contr. Syst. Techn.*, 22(1):102–118, 2014.

[41] D. H. Owens. *Iterative Learning Control: An Optimization Paradigm*. Advances in Industrial Control. Springer, 2016.

[42] W. Paszke, E. Rogers, and K. Gałkowski. Experimentally verified generalized KYP lemma based iterative learning control design. *Contr. Eng. Prac.*, 53:57–67, 2016.

[43] C. Peng, L. Sun, W. Zhang, and M. Tomizuka. Optimization-based constrained iterative learning control with application to building temperature control systems. In *Proc. 2016 Conf. AIM*, pages 709–715, Banff, Alberta, Canada, 2016.

[44] G. Pipeleers and K. L. Moore. Unified analysis of iterative learning and repetitive controllers in trial domain. *IEEE Trans. Automat. Contr.*, 59(4):953–965, 2014.

[45] E. Rogers, K. Galkowski, and D. H. Owens. *Control Systems Theory and Applications for Linear Repetitive Processes*. Number 349 in LNCIS. Springer, Berlin, Germany, 2007.

[46] C. R. Rojas and H. Hjalmarsson. Sparse estimation based on a validation criterion. In *Proc. 2011 Conf. Dec. Contr.*, pages 2825–2830, Orlando, FL, USA, 2011.

[47] C. R. Rojas, R. Tóth, and H. Hjalmarsson. Sparse estimation of polynomial and rational dynamical models. *IEEE Trans. Automat. Contr.*, 59(11):2962–2977, 2014.

[48] T. D. Son, G. Pipeleers, and J. Swevers. Multi-objective iterative learning control using convex optimization. *Eur. J. Contr.*, 33:35–42, 2017.

[49] C. Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 197–206. Univ. of Calif. Pres, 1956.

[50] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

[51] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. R. Statist. Soc. B*, 67(1):91–108, 2005.

[52] R. J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *The Annals of Statistics*, 39(3):1335–1371, 2011.

[53] J. Wallén Axehill, I. Dressler, S. Gunnarsson, and A. Robertsson. Estimation-based ILC applied to a parallel kinematic robot. *Contr. Eng. Prac.*, 33:1–9, 2014.

[54] J. van de Wijdeven and O. Bosgra. Using basis functions in iterative learning control: Analysis and design theory. *Int. J. Contr.*, 83(4):661–675, 2010.

[55] J. van de Wijdeven, T. Donkers, and O. Bosgra. Iterative learning control for uncertain systems: Robust monotonic convergence analysis. *Automatica*, 45:2383–2391, 2009.

[56] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

[57] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[58] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[59] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. R. Statist. Soc. B*, 67(2):301–320, 2005.

[60] J. van Zundert, J. Bolder, S. Koekebakker, and T. Oomen. Resource-efficient ILC for LTI/LTV systems through LQ tracking and stable inversion: Enabling large tasks on a position-dependent industrial printer. *Mechatronics*, 38:76–90, 2016.

[61] J. van Zundert, J. Bolder, and T. Oomen. Optimality and flexibility in iterative learning control for varying tasks. *Automatica*, 67:295–302, 2016.