Library-Based Norm-Optimal Iterative Learning Control

James Reed[a], Maxwell Wu[b], Kira Barton[b], Chris Vermillion[a], and Kirti D. Mishra[a]

***Abstract*— This paper presents a new iterative learning control (ILC) methodology, termed *library-based norm-optimal ILC*, which optimally accounts for variations in measurable disturbances and plant parameters from one iteration to the next. In this formulation, previous iteration-varying disturbance and/or plant parameters, along with the corresponding control and error sequences, are intelligently maintained in a dynamically evolving library. The library is then referenced at each iteration, in order to base the new control sequence on the most relevant prior iterations, according to an optimization metric. In contrast with the limited number of library-based ILC methodologies pursued in the literature, the present work (i) selects provably optimal interpolation weights, (ii) presents methods for starting with an empty library and intelligently truncating the library when it becomes too large, and (iii) demonstrates convergence to an optimal performance value. To demonstrate the effectiveness of our new methodology, we simulate our library-based norm-optimal ILC method on a linear time-varying model of a micro-robotic deposition system.**

I. INTRODUCTION

For systems that perform tasks repetitively, iterative learning control (ILC) can serve as an effective mechanism for leveraging past control and corresponding error sequences to inform control decisions at future trials (see [1]).

Traditional ILC is predicated on the plant, external disturbances, initial conditions, trial time, and reference trajectory being invariant from trial to trial. Some of these assumptions have been lifted as in [2], where the fixed trial duration assumption was removed, and in [3], the trial-invariant initial condition assumption was removed. Additionally, in [4] and [5], good learning performance was realized when the reference trajectory was allowed to change after a few trials.

In terms of lifting the assumptions of a trial-invariant plant and disturbance, there are two avenues that have been pursued, separately or in combination: (i) designing the trial-domain closed-loop dynamics to be robust in the presence of variations in the plant or disturbance, and (ii) explicitly accounting for measurable variations in the plant and disturbance within the ILC update law.

The first avenue has been pursued in a variety of robust ILC literature, such as in [6], and is appropriate in scenarios where trial-varying plant parameters and disturbances cannot be measured. However, it can be possible to estimate or directly measure the value of the disturbance or change in plant parameters at the outset of the next trial. For instance, in micro-robotic deposition systems, the structure of the printed colloidal material can be estimated [7]. Additionally, for undersea kite energy systems (see [8]) where kites are controlled to follow figure-eight paths while fulfilling economic objectives, the current speed for the next trial can be estimated given the last trial's current speed. In these scenarios, it stands to reason that the intelligent use of the measurements or estimates of plant parameters or disturbances within the ILC update law can result in improved performance, relative to a robust ILC law that does not make use of this information.

In fact, a relatively small collection of work, including [9], [10], and [11], has fused concepts from higher-order ILC (described in [12]) with the maintenance of a library of past trials' measured disturbances/plant parameters to tailor the ILC update law. Rather than basing each new control sequence entirely on the most recent trial, the new control sequence is updated based on the prior trial(s) whose measured disturbances/plant parameters are deemed most relevant based on the disturbance/plant parameters at the outset of the new trial. Thus, the techniques in [9], [10], and [11] leverage the idea from higher-order ILC of weighting multiple past trials in the learning process, but they select the weights based on the similarity of the trial-varying disturbance/plant parameters at those past trials compared to the current value of the trial-varying disturbance/plant parameters. A work shown in [13] designed a multipoint iterative learning model predictive controller which employed interpolation using a combination of iterative learning control and model predictive control. While this work presents a controller which chooses interpolation weights to interpolate between multiple system models to reduce model uncertainty, we chose to focus on explicitly using past control and error sequences stored as a function of *measured* changing plant/disturbance parameters.

While the aforementioned library-based schemes in [9], [10], and [11] reveal promise, they also leave room for advancement in terms of the optimization of the interpolation process (where parameters of the interpolation process can be selected to minimize/maximize an objective function), the process for dynamically maintaining (and truncating) the library, and the derivation of associated theory. Several recent efforts have begun to address these needs for advancement. Specifically, [9] presents a method of initializing the ILC training process from a library of control signals categorized by a "drift parameter," which varies between trials. While this methodology provides an effective mechanism for interpolating between control sequences indexed by the trial-varying parameter, the paper does not address the optimization of

This research was supported by the NSF award number DGE-1746939 and award number 1727779.

[a]Department of Mechanical and Aerospace Engineering at North Carolina State University, Raleigh, NC 27695, USA.

[b]Department of Mechanical Engineering at the University of Michigan, Ann Arbor, MI 48109, USA.

the interpolation weights, assumes a pre-populated static library, and assumes a slowly varying plant (such that it can be approximated as constant) in its associated theory. The work in [10] performs PD-ILC on a weighted sum of the past trial's control inputs, where the weights are determined not only based on the value of the measured disturbance for each library entry, but also the associated performance. While this method fuses a dynamically evolving library with a weighting scheme that has been shown to be effective on a manufacturing application, the weighting scheme is entirely heuristic, and the methodology is not accompanied with any associated theory. Finally, in [11], a parametric ILC formulation is presented that allows for the tracking of both trial-invariant and trial-varying references in the presence of a trial-varying disturbance. While this paper presents a method for updating the look-up table used for selecting feed-forward control parameters as a function of the external disturbance, it does not store complete control signals or error sequences in a library. It instead stores a set of feed-forward parameters. Additionally, the library size is fixed, causing information from trials with disturbances that do not lie on library indices to be forced onto the surrounding indices sub-optimally.

In the present paper, we introduce a norm-optimal library-based ILC formulation with three unique advancements that differentiate it from prior library-based ILC efforts:

1) Based on the framework of norm-optimal ILC ([14]), and tailored to the case of measurable and trial-varying plant parameters/disturbances, the formulation selects provably optimal interpolation weights, which minimize a quadratic objective function in terms of the error sequence, control sequence, and trial-to-trial deviation in the control sequence.
2) The formulation allows for a dynamically evolving library, which allows the algorithm to start with an empty library and intelligently truncate the library once it has reached a threshold size.
3) The formulation is accompanied by associated theory that demonstrates the convergence of the performance to its optimal value.

Using a simulation study based on an extrusion system originally introduced in [9], we compare our ILC interpolation method against the library-based ILC methods used in [9] and [10]. Finally, we validate our method for continuously updating the error and control libraries in simulation by showing the convergence of the objective function to the objective function value of traditional norm-optimal ILC run for 20 trials with trial-invariant parameters.

II. Problem Formulation

A class of linear parameter varying systems is considered in this study, the discrete-time state-space representation of which is described by:

$$\begin{aligned} x_{k+1} &= A(k,\lambda_k)x_k + B(k,\lambda_k)u_k \\ y_k &= C(k,\lambda_k)x_k \end{aligned} \tag{1}$$

The discrete-time index k ranges from 0 ... $N-1$, where N denotes the trial-invariant trial length. The system state, control input, and system output at k are denoted by x_k, u_k and y_k respectively, and the parameter and time-varying state, input, and output matrices are denoted by $A(k,\lambda_k)$, $B(k,\lambda_k)$, and $C(k,\lambda_k)$ respectively. It is assumed that the initial condition x_0 for (1) is trial-invariant. In addition, a constant external disturbance may be assumed in (1), which is set to zero here without loss of generality. The discrete-time evolution of the parameter $\lambda_k \in \mathbb{R}^{n_\lambda \times 1}$ is assumed here to be parameterized by a constant (in time) trial-varying parameter vector $\lambda'_j \in \mathbb{R}^{n_{\lambda'} \times 1}$, where j denotes the trial index. In one such parameterization method, λ_k evolves according to the nonlinear differential equation in (2), wherein λ'_j serves the role of the trial-varying initial value.

$$\lambda_{k+1} = f_\lambda(\lambda_k), \;\; \lambda_0 = \lambda'_j \tag{2}$$

A special case of this corresponds to $f_\lambda = \lambda'_j$, for which $\lambda_k = \lambda'_j \; \forall k$.

For the combined system defined by (1) and (2), the parameter λ'_j characterizes the system dynamics and, as part of repetitive system operation, is expected to vary from one trial to the next. The following assumptions regarding λ_k and λ'_j are posited:

A1. The trial-varying parameter λ'_j is assumed to be known prior to the start of the j^{th} trial.
A2. The dependence of λ_k on λ'_j is known $\forall\; k, j$. Specifically, with respect to (2), the function f_{λ_j} is known.
A3. The trial-varying parameter λ'_j randomly varies in the closed and bounded interval $[\underline{\lambda}', \overline{\lambda}']$.

Assumptions A1 and A2 ensure that prior to a trial, λ_k is known $\forall\; k$, which allows computation of the matrices in (1) to be used subsequently for designing optimal learning filters, and assumption A3 restricts the problem to physically relevant systems, where randomly varying environmental factors are often at play during iterative learning.

A lifted form representation [1] of the system in (1) in the trial domain is described in (3)-(5), where $Y_j, U_j, D_j(\lambda'_j) \in \mathbb{R}^{N\times 1}$ denote the discrete-time trajectories for the system output, control input, and the contribution of the initial condition to the system output during the j^{th} trial respectively, $G(\lambda'_j) \in \mathbb{R}^{N\times N}$ denotes the lifted system matrix, and $A_k \triangleq A(k,\lambda_k), k = 1 \;\ldots\; N-1$.

$$Y_j = G(\lambda'_j)U_j + D(\lambda'_j) \tag{3}$$

$$D(\lambda'_j) = [A_1x_0 \;\ldots\; \Pi_{i=1}^{k-1}A_ix_0 \;\ldots\; \Pi_{i=1}^{N-1}A_ix_0]^T \tag{4}$$

$$G(\lambda'_j)_{(p,q)} = \begin{cases} C(p,\lambda_p)B(q,\lambda_q) & p=q \\ C(p,\lambda_p)\prod_{i=q}^{p-1}A(i,\lambda_i)B(q-1,\lambda_{q-1}) & p<q \\ 0 & p>q. \end{cases} \tag{5}$$

Assuming that the system in (1) has relative degree equal to 1, which is uniform with respect to $\lambda'_j \; \forall j$, the lifted system matrix is described in (5). If this is not the case, the relative degree of (1) should be known as a function of $\lambda' \in [\underline{\lambda}' \;\; \overline{\lambda}']$. It is also noted here that the contribution $D(\lambda'_j)$ of the initial condition to the system output is trial-varying even if the initial condition is trial-invariant. As noted earlier, in addition to the contribution of the initial condition to the system output Y_j, the term $D(\lambda'_j)$ may also represent the contribution of a constant disturbance to Y_j. In such a case, for many applications, this constant disturbance may be heavily influenced by the parameter λ'_j characterizing the trial-varying system environment. Motivated by this, the following condition on $D(\lambda'_j)$ is assumed:

A4. For any sequence of random ${}^i\lambda'$, $i = 1 \ \dots \ n$ (see A3) with n sufficiently high ($\to \infty$ in the limit), and the sequence belonging to the set $[\underline{\lambda}' \quad \overline{\lambda}']$, $\exists \ k_i \geq 0$ and $\sum_{i=1}^{n} k_i = 1$ such that $\forall \ \lambda' \in [{}^1\lambda' \quad {}^n\lambda'], D(\lambda') = \sum_{i=1}^{n} k_i D({}^i\lambda')$.

Note that assumption A4 is trivially satisfied when $x_0 = 0$, which is applicable for a large class of systems that start the trial *un-energized*. Further, note that for A4 (and A5, to be described shortly), as $n \to \infty$, λ' is *almost* equal to one of the ${}^i\lambda'$, say for $i = i'$, and the assumption is trivially satisfied by setting all $k_i = 0$ except for $k_{i'}$, which is set equal to 1.

As part of iterative learning, it is desired here that a multi-objective cost function similar to that used in the literature on norm-optimal ILC is minimized over a small number of trials. One of the objectives traded off in the cost function for norm-optimal ILC is the output tracking error. For the class of systems considered in this study, it is expected that as the parameter λ'_j changes, representing a change in the system environment, the discrete-time desired trajectory $Y_d \in \mathbb{R}^{N \times 1}$ to be tracked may also change. Motivated by this, the desired trajectory is allowed to be a function of the λ'_j, and is denoted by $Y_d^{\lambda'}$. Similar to A4, the following assumption is posited for the desired reference to be tracked:

A5. For any sequence of random ${}^i\lambda'$, $i = 1 \ \dots \ n$ (see A3) with n sufficiently high ($\to \infty$ in the limit), and the sequence belonging to the interval $[\underline{\lambda}' \quad \overline{\lambda}']$, $\exists \ k_i \geq 0$ and $\sum_{i=1}^{n} k_i = 1$ such that $\forall \ \lambda' \in [{}^1\lambda' \quad {}^n\lambda'], Y_d^{\lambda'} = \sum_{i=1}^{n} k_i Y_d^{{}^i\lambda'}$.

Note that A5 is trivially satisfied when the desired reference trajectory is independent of λ', i.e. $Y_d^{\lambda'} = Y_d$.

III. Library-based Norm-optimal ILC

A. Traditional Norm-optimal ILC

In traditional norm-optimal ILC ([14]), it is desired that the objective function in (6) is minimized during every trial.

$$J_{j+1} = E_{j+1}^T Q E_{j+1} + U_{j+1}^T S U_{j+1} + \dots \\ (U_{j+1} - U_j)^T R (U_{j+1} - U_j) \tag{6}$$

In doing so, the tracking error $E_{j+1} = Y_d - Y_{j+1}$, control input effort, and incremental control input effort are optimally traded off, resulting in a small tracking error and control input effort as $j \to \infty$, and good learning transients during iterative learning. More specifically, as $j \to \infty$, the cost function converges to its minimum value $J^\star = E_\infty^T Q E_\infty + U_\infty^T S U_\infty$, wherein E_∞ and U_∞ represent the converged tracking error and control input trajectories. Note that the third term in (6) reduces to zero as $j \to \infty$. The learning control law and the associated learning filters for traditional norm-optimal ILC are computed by setting the derivative of J_{j+1} with respect to U_{j+1} equal to zero, with the resulting law and filters shown in (7)-(9), wherein $G \in \mathbb{R}^{N \times N}$ denotes the lifted system matrix for the underlying linear system.

$$U_{j+1} = L_u U_j + L_e E_j \tag{7}$$

$$L_u = (G^T Q G + S + R)^{-1} (G^T Q G + R) \tag{8}$$

$$L_e = (G^T Q G + S + R)^{-1} (G^T Q) \tag{9}$$

Owing to assumption A1, the idea of norm-optimal ILC summarized in the preceding paragraph can be straightforwardly extended to the class of systems considered in this study if (7)-(9) can be implemented for every $\lambda'_j \in [\underline{\lambda}' \quad \overline{\lambda}']$, i.e. prior to a trial, using $\lambda'_j = \lambda$, compute $G(\lambda)$ and, using which, L_u and L_e, per (8) and (9) respectively, and implement (7) using the trajectories U_j^λ and E_j^λ that correspond to the trials with $\lambda'_j = \lambda$. Clearly, as λ'_j varies continuously in $[\underline{\lambda}' \quad \overline{\lambda}']$, an infinite number of U_j^λ and E_j^λ trajectories must be stored and used in (7) for parametric iterative learning [11], which is not possible in practice. However, a sufficiently granular library that stores and updates a finite number of control input and tracking error trajectories can be effective in parameterizing norm-optimal ILC to *approximately* account for this full range of λ'_j.

B. Library-based Norm-optimal ILC

Our *library based norm-optimal* ILC method can be seen depicted in Figure 1, where a library is updated in the iteration domain as a function of the control and error sequences as well as the trial and time-varying disturbances or plant parameters, and used to create interpolation filters which are then multiplied by norm-optimal learning filters to create the next control input.

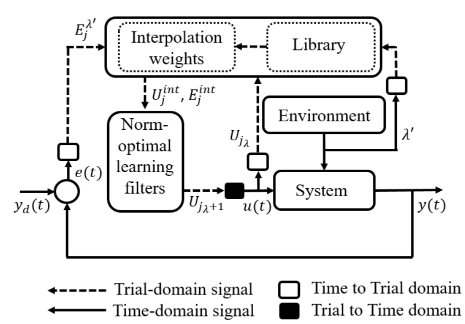

Fig. 1: Diagram of proposed *library-based norm-optimal ILC* method.

With the underlying motivation of formulating an objective function that is equivalent to the objective function used in traditional norm-optimal ILC and described in (6), the central ideas of a *library* and *interpolation* are introduced next. For simplicity, it will be assumed that $\lambda'_j \in \mathbb{R}$; however, the arguments that follow can be extended to the vector case. A library of control input and tracking error trajectories is denoted by the pair of matrices $\mathcal{L}^u_{\lambda'}, \mathcal{L}^e_{\lambda'} \in \mathbb{R}^{N \times N_j^{\mathcal{L}}}$, which are defined as $\mathcal{L}^u_{\lambda'} = [U_j^1 \ \dots \ U_j^{N_j^{\mathcal{L}}}]$ and $\mathcal{L}^e_{\lambda'} = [E_j^1 \ \dots \ E_j^{N_j^{\mathcal{L}}}]$ respectively, the corresponding λ' values ${}^i\lambda', i = 1 \ \dots \ N_j^{\mathcal{L}}$ are ordered from the minimum ${}^1\lambda'$ to maximum ${}^{N_j^{\mathcal{L}}}\lambda'$. It is noted here that for library-based norm-optimal ILC, these library matrices are updated iteratively, and consequently, the number of columns as well as the columns themselves are trial-varying.

Interpolation of the columns of $\mathcal{L}^u_{\lambda'}, \mathcal{L}^e_{\lambda'}$ is referred to as the method for computing control input and tracking error trajectories corresponding to a $\lambda' \in [{}^1\lambda' \quad {}^{N_j^{\mathcal{L}}}\lambda']$ and $\neq {}^1\lambda' \ \dots \ {}^{N_j^{\mathcal{L}}}\lambda'$. Interpolated control input and tracking error trajectories U_j^{int} and E_j^{int} are described in (10), wherein k_j^u and k_j^e denote the interpolation weights:

$$U_j^{int} = \sum_{i=1}^{N_j^{\mathcal{L}}} {}^ik_j^u G^{-1}(\lambda'_j)G({}^i\lambda'_j)U_j^i, \;\; E_j^{int} = \sum_{i=1}^{N_j^{\mathcal{L}}} {}^ik_j^e E_j^i. \tag{10}$$

To gain physical insight into interpolation, consider a representative interpolation method that uses the trajectories corresponding only to the neighboring points ${}^i\lambda'$ and ${}^{i+1}\lambda'$, with ${}^i\lambda' < \lambda'_j <^{i+1} \lambda'$ and the distance-based interpolation weights ${}^ik_j^u = {}^ik_j^e = \frac{{}^{i+1}\lambda' - \lambda'_j}{{}^{i+1}\lambda' - {}^i\lambda'}$ and ${}^{i+1}k_j^u = {}^{i+1}k_j^e = \frac{\lambda'_j - {}^i\lambda'}{{}^{i+1}\lambda' - {}^i\lambda'}$. Such distance-based interpolation weights were used in [11] for parametric ILC.

It is noteworthy that if $\lambda'_j = \lambda' \; \forall j$, the optimum value $J^\star$ of (6) resulting from the use of (7)-(9) will be a function of λ', and is more appropriately represented as $J^\star_{\lambda'}$, which is equal to $E_\infty^{\lambda'^T} Q E_\infty^{\lambda'} + U_\infty^{\lambda'^T} S U_\infty^{\lambda'}$, wherein $E_\infty^{\lambda'}$ and $U_\infty^{\lambda'}$ represent the converged tracking error and control input trajectories corresponding to $\lambda'_j = \lambda' \; \forall j$. As λ'_j varies randomly in $[\underline{\lambda'} \quad \overline{\lambda'}]$, per assumption A3, it is desired that as $j \rightarrow \infty$, the new objective function to be formulated should converge to $J^\star_{\lambda'} \; \forall \lambda' \in [\underline{\lambda'} \quad \overline{\lambda'}]$. As will be discussed shortly, it is useful to transform this specification into an equivalent form. To this end, let $j_{\lambda'}$ denote the trial index corresponding to λ', i.e. $j_{\lambda'}$ counts the number of trials for which $\lambda'_j = \lambda'$. It is not difficult to see that, owing to A3, as $j \rightarrow \infty$, $j_{\lambda'} \rightarrow \infty \; \forall \; \lambda' \in [\underline{\lambda'} \quad \overline{\lambda'}]$. The equivalent specification requires that as $j_{\lambda'} \rightarrow \infty$, the objective function to be formulated converges to $J^\star_{\lambda'} \; \forall \; \lambda'$.

Using the ideas of library and interpolation, and motivated by the specification described in the preceding paragraph, the objective function for library-based norm-optimal ILC is formulated in (11), wherein $E_{j_{\lambda'}+1} = Y_d^{\lambda'} - Y_{j_{\lambda'}+1}$, and U_j^{int} is defined in (10):

$$\begin{aligned} J_{j_{\lambda'}+1} = E^T_{j_{\lambda'}+1} Q E_{j_{\lambda'}+1} + U^T_{j_{\lambda'}+1} S U_{j_{\lambda'}+1} + \dots \\ (U_{j_{\lambda'}+1} - U_j^{int})^T R (U_{j_{\lambda'}+1} - U_j^{int}) \end{aligned} \tag{11}$$

The major difference between the traditional and library-based norm-optimal objective functions lies in the third term that penalizes the incremental control input effort. While in traditional norm-optimal ILC, it is reasonable to desire that a large correction to the control input never results during iterative learning, for parametric or library-based norm-optimal ILC, the system dynamics and the desired reference trajectory change every trial, thereby potentially necessitating a larger change in the control input across any two consecutive trials. In library-based norm-optimal ILC, it makes more sense to penalize $(U_{j_{\lambda'}+1} - U_j^{int})$, where U_j^{int} serves as a proxy for $U_{j_{\lambda'}}$.

There are two questions that remain to be answered: (i) akin to (7) for traditional norm-optimal ILC, what is the optimal structure of $U_{j_{\lambda'}+1}$ in terms of U_j^{int} and E_j^{int}, and (ii) what are the optimal interpolation weights for computing the interpolated vectors U_j^{int} and E_j^{int}? The first of these is addressed in Theorem 1, and the second of these is addressed after Theorem 1 in Proposition 1. In the following theorem, $\forall j$ such that $\lambda'_j = \lambda'$, we define $G_{\lambda'} \triangleq G(\lambda')$ and $D_{\lambda'} \triangleq D(\lambda')$.

Theorem 1. *For an arbitrarily fixed sufficiently high j, the learning control law of (12)-(14), with U_j^{int} and E_j^{int} defined in (10), minimizes the objective function in (11) for the combined system (1) and (2) satisfying* A1-A5 $\forall \lambda'_j \in [\underline{\lambda'} \quad \overline{\lambda'}]$ *and for a sufficiently high $N_j^{\mathcal{L}}$ ($\rightarrow \infty$ in the limit) if ${}^ik_j^u = {}^ik_j^e \geq 0 \; \forall \; i = 1 \; \dots \; N_j^{\mathcal{L}}$ and $\sum_{i=1}^{N_j^{\mathcal{L}}} {}^ik_j^u = 1$.*

$$U_{j_{\lambda'}+1} = L^u_{\lambda'} U_j^{int} + L^e_{\lambda'} E_j^{int} \tag{12}$$

$$L^u_{\lambda'} = (G_{\lambda'}{}^T Q G_{\lambda'} + S + R)^{-1}(G^T_{\lambda'} Q G_{\lambda'} + R) \tag{13}$$

$$L^e_{\lambda'} = (G^T_{\lambda'} Q G_{\lambda'} + S + R)^{-1}(G^T_{\lambda'} Q) \tag{14}$$

Proof: When the cost function in (11) is differentiated with respect to $U_{j_{\lambda'}+1}$, and set equal to the zero vector, the operation results in (15), wherein $\hat{Y}_d^{\lambda'} \triangleq Y_d^{\lambda'} - D_{\lambda'}$:

$$U_{j_{\lambda'}+1} = (G^T_{\lambda'} Q G_{\lambda'} + S + R)^{-1}(G^T_{\lambda'} Q \hat{Y}_d^{\lambda'} + R U_j^{int}) \tag{15}$$

Now, consider the interpolated error E_j^{int} of (10), which is transformed to (16) using (3):

$$E_j^{int} = \sum_{i=1}^{N_j^{\mathcal{L}}} {}^ik_j^e (Y_d^{{}^i\lambda'_j} - G_{{}^i\lambda'_j} U_j^i - D_{{}^i\lambda'_j}) \tag{16}$$

Using ${}^ik_j^e = {}^ik_j^u \; \forall \; i = 1 \; \dots \; N_j^{\mathcal{L}}$ in (16), (17) is derived:

$$E_j^{int} = \sum_{i=1}^{N_j^{\mathcal{L}}} {}^ik_j^u Y_d^{{}^i\lambda'_j} - \sum_{i=1}^{N_j^{\mathcal{L}}} {}^ik_j^u G_{{}^i\lambda'_j} U_j^i - \sum_{i=1}^{N_j^{\mathcal{L}}} {}^ik_j^u D_{{}^i\lambda'_j} \tag{17}$$

By pre-multiplying the second term in (17) by $G_{\lambda'} G^{-1}_{\lambda'}$, and using the fact that $\sum_{i=1}^{N_j^{\mathcal{L}}} {}^ik_j^u = 1$, along with assumption A4, assumption A5, and the definition (10) of U_j^{int}, the equality in (18) is derived:

$$E_j^{int} = Y_d^{\lambda'_j} - G_{\lambda'} U_j^{int} - D_{\lambda'_j} \tag{18}$$

Rearranging (18), and substituting $Y_d^{\lambda'_j}$ from (15), the equality in (19) is derived.

$$\begin{aligned} U_{j_{\lambda'}+1} = (G^T_{\lambda'} Q G_{\lambda'} + S + R)^{-1} \dots \\ (G^T_{\lambda'} Q (E_j^{int} + G_{\lambda'} U_j^{int}) + R U_j^{int}) \end{aligned} \tag{19}$$

On rearranging (19), the control law in (12)-(14) results. □

Remark 1. *It is recalled here that the assumptions A4 and A5 trivially hold* $\forall\ j$ *if the initial condition* $x_0 = 0$ *(a zero vector) and* $Y_d^{\lambda'} = Y_d$ *respectively, implying that optimality of the library-based norm-optimal ILC law in* (12)-(14) *is optimal* $\forall\ j$. *For systems with* $x_0 \neq 0, Y_d^{\lambda'} \neq Y_d$, *the assumptions A4 and A5 hold for a sufficiently large library, i.e. for a sufficiently large* $N_j^{\mathcal{L}}$, *thereby implying that* (12)-(14) *may be sub-optimal for the first few trials.*

Let ${}^i k_j^u = {}^i k_j^e = {}^i k_j\ \forall\ j$ and $i = 1\ \dots\ N_j^{\mathcal{L}}$, and $U_j^{int} \triangleq \bar{A}V, E_j^{int} \triangleq \bar{H}V$, wherein $V = [{}^1 k_j\ \dots\ {}^{N_j^{\mathcal{L}}} k_j]^T \in \mathbb{R}^{N_j^{\mathcal{L}} \times 1}$, $\bar{A}$ and $\bar{H}$ are easily computed using (10). An optimal method for computation of the interpolation weights ${}^i k_j$ is presented in Proposition 1, wherein the second of the two questions posed earlier is investigated.

Proposition 1. *The solution that minimizes the optimization problem:*

$$\min_{[{}^1 k_j\ \dots\ {}^{N_j^{\mathcal{L}}} k_j]^T} \quad J_{j_{\lambda'}+1} \text{ of (11)} \tag{20}$$

$$\text{s.t.} \sum_{i=1}^{N_j^{\mathcal{L}}} {}^i k_j = 1,\ 0 \le {}^i k_j \le 1,\ \forall\ i = 1 \dots\ N_j^{\mathcal{L}} \tag{21}$$

$$\text{and system dynamics (3)} - (5) \tag{22}$$

represents a unique global optimum within the feasible solution set described in (21).
Proof: The proof follows from noting that the cost function in (11) is quadratic in $V = [{}^1 k_j\ \dots\ {}^{N_j^{\mathcal{L}}} k_j]^T$ for the (linear) lifted form system dynamics of (3) – (5), the quadratic form being shown in (23).

$$J_{j_{\lambda'}+1} = \bar{V}^T \bar{W}_1 \bar{V} + \bar{W}_2 \bar{V} + \bar{W}_3, \tag{23}$$

$$\begin{aligned}\bar{W}_1 &= F^T QF + P^T QP + T^T ST + \bar{A}^T R\bar{A} + T^T RT + \\ &\quad Z^T SZ + Z^T RZ + F^T QP + T^T RZ + T^T SZ \\ &\quad - T^T R\bar{A} + P^T QF + Z^T RT + Z^T ST - \bar{A}^T RT \\ \bar{W}_2 &= -2M^T QF - 2M^T QP,\ \bar{W}_3 = M^T QM.\end{aligned} \tag{24}$$

where $M = (Y_d^{\lambda'} - D_{\lambda'}), F = G_{\lambda'} L_{\lambda'}^u \bar{A},\ P = G_{\lambda'} L_{\lambda'}^e \bar{H}$, $T = L_{\lambda'}^u \bar{A}$, and $Z = L_{\lambda'}^e \bar{H}$. The constraints appearing in the optimization problem can be written as shown in (25), wherein $A_{eq} \in \mathbb{R}^{N_j^{\mathcal{L}} \times 1}$, $A_{ineq} \in \mathbb{R}^{2N_j^{\mathcal{L}} \times N_j^{\mathcal{L}}}$, and $B_{ineq} \in \mathbb{R}^{2N_j^{\mathcal{L}} \times 1}$ are defined in (26). In equation (26), $\bar{0}_{N_j^{\mathcal{L}} \times 1}$ represents a column of zeros and $\bar{1}_{N_j^{\mathcal{L}} \times 1}$ represents a column of ones.

$$A_{eq} \bar{V} = 1,\ A_{ineq} \bar{V} \le \bar{B}_{ineq} \tag{25}$$

$$A_{eq} = [1, \dots, 1],\ A_{neq} = \begin{bmatrix} -\mathbb{I}_{N_j^{\mathcal{L}} \times N_j^{\mathcal{L}}} \\ \mathbb{I}_{N_j^{\mathcal{L}} \times N_j^{\mathcal{L}}} \end{bmatrix},\ B_{neq} = \begin{bmatrix} \bar{0}_{N_j^{\mathcal{L}} \times 1} \\ \bar{1}_{N_j^{\mathcal{L}} \times 1} \end{bmatrix}. \tag{26}$$

The objective function in equation (23) and the constraints shown in equations (25) are now posed in the form of a quadratic program that when solved gives the globally and uniquely optimal solution (where the solution is a vector of interpolation weights) to the optimization problem. □

As was alluded to earlier, if $\lambda_j' = \lambda'\ \forall\ j$, the optimal cost is given by $J_{\lambda'}^{\star} = {E_{\infty}^{\lambda'}}^T Q E_{\infty}^{\lambda'} + {U_{\infty}^{\lambda'}}^T S U_{\infty}^{\lambda'}$. The formulation of the library-based objective function in (11) was motivated by the fact that when λ_j' varies according to A3, the associated norm-optimal ILC method should result in convergence of the cost function $J_{j_{\lambda'}}$ to $J_{\lambda'}^{\star}\ \forall \lambda' \in [\underline{\lambda}'\ \ \overline{\lambda}']$ as $j_{\lambda'} \to \infty$ or $j \to \infty$. With this underlying motivation, the convergence of $J_{j_{\lambda'}}$ as $j_{\lambda'} \to \infty$ is studied next for the *optimal learning control law* of (12)-(14) and the *optimal interpolation method* (20)-(22). In particular, the case of $S = 0$ is considered, for which the tracking error converges to zero as $j \to \infty$, i.e. $E_{\infty}^{\lambda'} = \underline{0}$, implying that the equality in (27) holds.

$$Y_d^{\lambda'} = G_{\lambda'} U_{\infty}^{\lambda'} + D_{\lambda'}\ \forall \lambda' \in [\underline{\lambda}'\ \ \overline{\lambda}'] \tag{27}$$

We begin with the following lemma:

Lemma 1. *For the library-based norm-optimal control law* (12)-(14) *with* $S = \underline{0}$, $E_{j_{\lambda'}+1}^T Q E_{j_{\lambda'}+1} \le E_j^{int^T} Q E_j^{int}$, *and* $|U_{\infty}^{\lambda'} - U_{j_{\lambda'}+1}| \le |U_{\infty}^{\lambda'} - U_j^{int}|$ *for a sufficiently large* j.
Proof: For a sub-optimal choice of $U_{j_{\lambda'}+1} = U_j^{int}$, $J_{j_{\lambda'}+1} = (\hat{Y}_d^{\lambda'} - G_{\lambda'} U_j^{int})^T Q (\hat{Y}_d^{\lambda'} - G_{\lambda'} U_j^{int})$. Using (27), $J_{j_{\lambda'}+1} = (G_{\lambda'} \delta U_j^{int})^T Q (G_{\lambda'} \delta U_j^{int})$, with $\delta U_j^{int} = U_{\infty}^{\lambda'} - U_j^{int}$. For the optimal choice of $U_{j_{\lambda'}+1}$ (12)-(14), the cost $\delta U_j^{int^T} G_{\lambda'}^T Q G_{\lambda'} \delta U_j^{int} \ge \delta U_{j_{\lambda'}+1}^T G_{\lambda'}^T Q G_{\lambda'} \delta U_{j_{\lambda'}+1} + (U_{j_{\lambda'}+1} - U_j^{int})^T R (U_{j_{\lambda'}+1} - U_j^{int})$, where $\delta U_{j_{\lambda'}+1} = U_{\infty}^{\lambda'} - U_{j_{\lambda'}+1}$, and as $(U_{j_{\lambda'}+1} - U_j^{int})^T R (U_{j_{\lambda'}+1} - U_j^{int}) > 0$, $\delta U_j^{int^T} G_{\lambda'}^T Q G_{\lambda'} \delta U_j^{int} \ge \delta U_{j_{\lambda'}+1}^T G_{\lambda'}^T Q G_{\lambda'} \delta U_{j_{\lambda'}+1}$. Noting that $Q > 0$ in norm-optimal ILC and $G_{\lambda'}$ is full rank $\forall\ \lambda'$, implying that $G_{\lambda'}^T Q G_{\lambda'} > 0$, the inequality implies $|U_{\infty}^{\lambda'} - U_{j_{\lambda'}+1}| \le |U_{\infty}^{\lambda'} - U_j^{int}|$. □

Lemma 1 suggests that the library must be updated after every trial using the corresponding control input and error trajectories such that the interpolated trajectories E_j^{int} and δU_j^{int} converge to 0 $\forall\ \lambda'$, which in turn will imply that $E_{j_{\lambda'}+1}, \delta U_{j_{\lambda'}+1} \to 0\ \forall\ \lambda'$. Towards this end, the *library update law* of adding $U_{j_{\lambda'}+1}$ computed for the next trial and the resulting $E_{j_{\lambda'}+1}$ to $\mathcal{L}_{\lambda'}^u$ and $\mathcal{L}_{\lambda'}^e$ respectively is introduced here. Per this law, the library-based norm-optimal learning law of (12)-(14) with the interpolated U_j^{int} and E_j^{int} computed using (20)-(22) is used. In addition, it will be required that some existing entries of the library are dropped in order to keep the size of the library manageable – see Algorithm 1 presented shortly, which is based on this library update law. In Theorem 2, the convergence of $E_{j_{\lambda'}+1}, \delta U_{j_{\lambda'}+1} \to 0\ \forall\ \lambda'$ as $j \to 0$ under assumption A3, according to which λ' varies randomly in $[\underline{\lambda}'\ \ \overline{\lambda}']$, is shown.

Theorem 2. *For the combined system dynamics of* (1) *and* (2) *that satisfy A1 through A5, the library-based norm-optimal ILC, i.e. the optimal learning control law of* (12)-(14), *the optimal interpolation method of* (20)-(22), *and the library update law described earlier, result in the convergence of* $E_{j_{\lambda'}+1}, \delta U_{j_{\lambda'}+1} \to 0\ \forall\ \lambda' \in [\underline{\lambda}'\ \ \overline{\lambda}']$ *as* $j_{\lambda'} \to \infty$.
Proof: Lemma 1 suggests that it is sufficient to show that

$E_j^{int}, \delta U_j^{int} \to 0 \ \forall\ \lambda' \in [\underline{\lambda'} \ \ \overline{\lambda'}]$ as $j \to \infty$. Moreover, due to (27), only δU_j^{int} will be considered. The proof follows from (i) $\forall\ k = 0 \ \dots \ N-1$ and $\forall\ \lambda' \in [\underline{\lambda'} \ \ \overline{\lambda'}]$, the distance between the k^{th} elements of the desired control input $U_\infty^{\lambda'}(k)$ and the interpolated control input $U_j^{int}(k)$ computed using the optimal interpolation method of (20)-(22) is less than the distance $|U_\infty^{\lambda'}(k) - p|$, where p lies in the convex hull formed by the k^{th} elements of the control input trajectories in the library at the j^{th} trial, i.e. $p \in \texttt{conv}(U_j^1(k) \ \dots \ U_j^{N_j^{\mathcal{L}}}(k))$ (in particular, p represents the library entries $U_j^i(k)$ at the j^{th} trial), (ii) assumption A3, which ensures that (i) holds $\forall\ \lambda' \in [\underline{\lambda'} \ \ \overline{\lambda'}]$, (iii) Lemma 1, which ensures that $|U_\infty^{\lambda'} - U_{j_{\lambda'}+1}| \leq |U_\infty^{\lambda'} - U_j^{int}|$, and (iv) the library update described earlier, which ensures that $U_{j_{\lambda'}+1}$ computed using (12)-(14) and (20)-(22) is stored in the library, implying that the convex hull formed by the library entries at the next trial $j+1$ is closer to $U_\infty^{\lambda'}$.

In order to see the argument in (i), consider the following sub-optimal solution of (20)-(22) for the interpolation weights. Set all but any two arbitrarily chosen interpolation weights equal to zero in (21). Using an argument similar to that presented in Lemma 1, it can be seen that for any such sub-optimal solution $\hat{U}_j^{int}$ (i.e. for any combination of these two non-zero weights), $|U_\infty^{\lambda'} - U_j^{int}| < |U_\infty^{\lambda'} - \hat{U}_j^{int}|$. Extending this argument to *any* pair of the interpolation weights, the argument in (i) is obtained. □

To update the libraries $\mathcal{L}_{\lambda'}^u$ and $\mathcal{L}_{\lambda'}^e$, control and error trajectories are added into the library, indexed by λ' at each trial. To prevent the libraries from getting too large as many trials occur, the columns of the libraries are sorted into bins with N_e number of edges, indexed by their associated ${}^i\lambda'$ at the end of each trial. From these bins, the N_k most recent U_{j+1} and E_{j+1} vectors are kept. The total number of elements in each bin is $N_j^{\mathcal{L}} = N_e N_k - 1$. This causes the libraries $\mathcal{L}_{\lambda'}^u$ and $\mathcal{L}_{\lambda'}^e$ to be $\in \mathbb{R}^{N \times N_j^{\mathcal{L}}}$. For an initial parameter λ' that is randomly trial varying, starting from an empty library, the process can be seen in Algorithm 1.

IV. Results

In this section, we perform an assessment of our norm-optimal library-based ILC approach, using a micro-robotic extrusion system as a case study. The results demonstrate (i) the effectiveness of our interpolation method as compared to the methods shown in [9] and [10], along with (ii) convergence of the objective function to the norm-optimal value when the initial drift parameter λ' is constant from trial to trial.

A. Simulation Model

To test our new library-based ILC interpolation method, we consider an additive manufacturing system: a micro-robotic deposition model that extrudes thixotropic ink. This is the same system model that was used in [9], which enables a fair comparison to the algorithm used in [9].

Algorithm 1 Library-based norm-optimal ILC

Input: Plant model as a function of λ', reference trajectory Y_d, minimum and maximum values λ' can take (λ'_- , λ'_+), λ dynamics (equation (2)), a finite sequence of λ'_j where $j = 1, \ \ \dots, \text{j}_{end}$, the number of bin edges N_e, total number of elements to keep per bin, N_k.

Initialization : Select any control input that stabilizes the plant and perform one trial. Store the control input and corresponding error sequence into the libraries $\mathcal{L}_{\lambda'}^u$ and $\mathcal{L}_{\lambda'}^e$ indexed by the value of λ' at the first trial. Bin the space between $[\underline{\lambda'} \ \ \overline{\lambda'}]$ by using N_e edges.

while $j \leq \text{j}_{end}$ **do**
  $\lambda' = \lambda'_{j+1}$
  Compute learning filters $L_{\lambda'}^u$ and $L_{\lambda'}^e$ using (13) and (14)
  Compute U_j^{int} and E_j^{int} using (20)-(22)
  Create $U_{j_{\lambda'}+1} = U_{j+1}$ using (12)
  Perform the $j+1^{st}$ trial using the computed U_{j+1}.
  Store the control input and error trajectories U_{j+1} and E_{j+1} into the libraries $\mathcal{L}_{\lambda'}^u$ and $\mathcal{L}_{\lambda'}^e$ indexed by λ'.
  Perform the library truncation process by keeping only the U_{j+1} and E_{j+1} vectors from the N_k most recent trials per bin and per library.
  $j = j+1$
end while

B. Interpolation Results

In this section, the new interpolation method shown in Section III is compared to the Gaussian interpolation filter method used in [9] and the heuristic weighting method used in [10]. The methodology for testing the interpolation methods was to create partially converged libraries ($\mathcal{L}_{\lambda'}^u, \mathcal{L}_{\lambda'}^e$) corresponding to $\lambda' = [0.2, 0.6, 1.0]$ by running traditional norm-optimal ILC for three trials. Then, using each method, the interpolation weights were computed with a corresponding control sequence for a set of test λ'. The objective function value was then calculated for each test case, J_{intrp}, resulting from the control sequence. Lastly, it was compared to the objective function value achieved by traditional norm-optimal ILC run for 20 trials where λ' is equal to the test λ' and is trial-invariant. These results can be seen in Figure 2. In this figure, it is clear that the new interpolation method shown in Section III yields a better interpolated control input because it has the lowest value of $J_{intrp} - J_{ILC20}$ for each parameter λ'.

C. Continuous Interpolation Starting from an Empty Library

In this section, the results are generated via Algorithm 1. The results of this can be seen in Figure 3. To visualize the convergence of the objective function value per trial to the converged value of the objective function achieved by traditional ILC run for 20 trials where λ' is trial-invariant, the value of J_{ILC20} was computed for a discretized range of λ' between 0 and 1. The range was then divided into 6 "optimal cost sections". The optimal cost values corresponding to the maximum and minimum value of λ' per section were used to

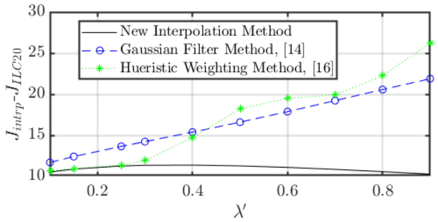

Fig. 2: Objective function value after interpolation for each of the three methods, starting from a partially converged library, minus the objective function value of ILC run for 20 trials with a trial-invariant λ'. A lower value of $J_{intrp} - J_{ILC20}$ is better.

create the shaded rectangles seen in Figure 3. A simulation was then run for 100 trials, and the objective function values were binned by the corresponding λ' value at each trial and given a similar color to the sections in which the λ' values per trial corresponded to. From this figure, it can be seen that the objective function values in each bin converge to the section showing the converged value of the objective function. Additionally, in Figure 4, we show time traces of the output being tracked. From this figure, the error between the desired output and output per trial can be seen to decrease as the trial number increases, due to the trials plotted being chosen for the proximity of their λ' values to each other.

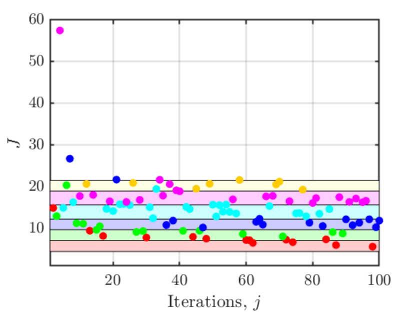

Fig. 3: Convergence of the library-based norm-optimal ILC method starting from a empty library to the *optimal cost sections*, when the value of the initial parameter $\lambda'_j \in [\underline{\lambda'} \;\; \overline{\lambda'}]$ varies randomly at each trial.

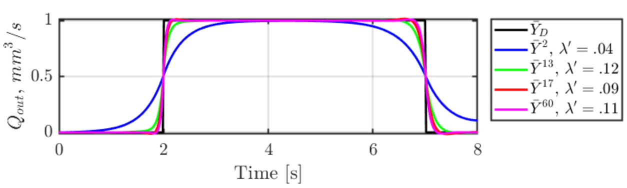

Fig. 4: System outputs over multiple trials with similar λ' values compared to the reference trajectory.

V. Conclusions and Future Work

In this work, we have presented a new methodology, termed *library-based norm-optimal ILC*, for systems where a trial-varying disturbance or change in plant parameters can be estimated or measured at the outset of each trial. Additionally, we presented a library generation strategy, and a library interpolation strategy for generating new control signals once a new initial parameter is encountered, with a proof showing that the interpolation weights calculated by this method globally and uniquely minimize the objective function at the next trial within a set of constraints. Additionally, we presented associated theory for the convergence of the performance to its optimal value. Finally, we showed the efficacy of our methodology on a micro-robotic deposition system. In these results, we demonstrated that our interpolation method outperforms prior library-based ILC techniques and showed the effectiveness of our library-based norm-optimal ILC method in continuous use, starting from an empty library. In the future, investigations into optimal methods for selecting the bin parameters used in Algorithm 1 will be performed. Additionally, methods for selecting the optimal number of past control and error trials to keep in each bin will be investigated. Finally, theoretical insights into the robustness of this control algorithm as well as the convergence rate will be explored.

References

[1] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE control systems magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[2] M. Cobb, M. Wu, K. Barton, and C. Vermillion, "Flexible-time economic iterative learning control: A case study in airborne wind energy," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 5580–5586.

[3] M. Cobb, J. Reed, M. Wu, K. Mishra, K. Barton, and C. Vermillion, "Flexible-time receding horizon iterative learning control with application to marine hydrokinetic energy systems," March 2021, submitted to Transaction on Control Systems Technology.

[4] 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*, vol. 24, no. 8, pp. 944–953, 2014.

[5] J. Bolder and T. Oomen, "Rational basis functions in iterative learning control—with experimental verification on a motion system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 722–729, 2014.

[6] J.-X. Xu and Z. Qu, "Robust iterative learning control for a class of nonlinear systems," *Automatica*, vol. 34, no. 8, pp. 983–988, 1998.

[7] C. Zhu and J. E. Smay, "Thixotropic rheology of concentrated alumina colloidal gels for solid freeform fabrication," *Journal of Rheology*, vol. 55, no. 3, pp. 655–672, 2011.

[8] A. Ghasemi, D. J. Olinger, and G. Tryggvason, "A nonlinear computational model of tethered underwater kites for power generation," *Journal of Fluids Engineering*, vol. 138, no. 12, 2016.

[9] D. J. Hoelzle and K. Barton, "Flexible iterative learning control using a library based interpolation scheme," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 3978–3984.

[10] C. DiMarco, C. Vermillion, and J. C. Ziegert, "Disturbance and performance-weighted iterative learning control with application to modulated tool path-based manufacturing," in *ASME 2016 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers Digital Collection, 2016.

[11] K. D. Mishra and K. Srinivasan, "Parametric iterative learning control: A method for adaptation of feedforward control, and application to gearshift control," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 218–223.

[12] Z. Bien and K. M. Huh, "Higher-order iterative learning control algorithm," in *IEE Proceedings D (Control Theory and Applications)*, vol. 136, no. 3. IET, 1989, pp. 105–112.

[13] J. Lu, Z. Cao, and F. Gao, "Multipoint iterative learning model predictive control," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 8, pp. 6230–6240, 2018.

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