IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 15, 2021, accepted October 25, 2021, date of publication October 27, 2021,
date of current version November 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123868

Dynamic Output-Only Iterative Learning Control Design

LUKASZ HLADOWSKI [1], KRZYSZTOF GALKOWSKI [2], (Member, IEEE),
AND ERIC ROGERS [3], (Life Member, IEEE)
[1]Institute of Control and Computation Engineering, University of Zielona Góra, 65-246 Zielona Góra, Poland
[2]Institute of Automation, Electronic and Electrical Engineering, University of Zielona Góra, 65-516 Zielona Góra, Poland
[3]Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

Corresponding author: Lukasz Hladowski (L.Hladowski@issi.uz.zgora.pl)

This work was supported in part by the National Science Centre in Poland under Grant 2020/37/B/ST7/03280.

ABSTRACT Iterative learning control applies to systems that repeatedly execute the same finite duration task. The distinguishing feature of this form of control action is that all data generated on a previous execution of the task is available to compute the control action for the subsequent execution. This paper uses the repetitive process stability analysis and optimization techniques to design a dynamic controller that, in contrast to previous designs in the repetitive process/2D systems setting, does not require measurement of the state dynamics or observer-based estimation. Examples to demonstrate the application of the new design are given.

INDEX TERMS Iterative learning control, repetitive processes, output control.

I. INTRODUCTION

The most common form where Iterative Learning Control (ILC) can be employed is illustrated by a robot executing a 'pick and place' operation. In such an application, the sequence of operations is: i) collect an object or payload from a specified location, ii) transfer it over a finite duration, iii) deposit the payload at a fixed location or on a moving conveyor under synchronization, iv) return to the starting location, v) collect the next payload and repeat i)-iv). It can also be applied when the system completes the same finite duration task in sequence with a stoppage-time between the completion of each execution.

In this paper, each execution of the system is termed a trial and trial length is used to denote the duration of a trial. The notation for variables is of the form $\boldsymbol{h}_k(p)$, $0 \leq p \leq \alpha-1$, $k \geq 0$, where $\boldsymbol{h}$ is the vector or scalar variable under consideration, $k \geq 0$ denotes the trial number and for discrete dynamics, α denotes the number of samples along a trial ($\alpha \times$ the sampling period gives the trial length).

Suppose that a reference trajectory (or vector in the multiple-input multiple-output case) is available. The error on each trial can be formed as the difference with the output on this trial. Let $\{\boldsymbol{e}_k\}_k$ be the error sequence generated by an example under consideration. Then the basic premise in ILC is to force convergence in k of this sequence and ensure acceptable transient and steady-state dynamics along the trial. Once a trial is complete, all information generated is available to generate the control input for the next trial, most often as used on the previous trial plus a correction term.

Consider, without loss of generality, the single-input single-output case where $\boldsymbol{u}_k(p)$ is the input applied on trial k. Then one of the possible ILC law has the from $\boldsymbol{u}_{k+1}(p) = \boldsymbol{u}_k(p) + \boldsymbol{k_1}\boldsymbol{e}_k(p+\lambda)$, $\lambda > 0$. This control law is termed phase-lead due to the use of data at sample $p+\lambda$ on trial k at sample p on trial $k+1$, where this data is available once trial k is complete. Moreover, suppose the only previous trial input to the control applied at any sampling instance on the current trial comes from the same sampling instant on the previous trial. In that case, an equivalent feedback control law and ILC has no added benefit.

The earliest research on ILC is widely credited to [1] and focused on robotic applications. Since this early work, ILC has remained a very active research area in terms of theory/algorithm development and applications. The survey papers [2] and [3] are possible starting points for the earlier literature. More recent application areas include additive manufacturing [4], center-articulated industrial vehicles [5] and in healthcare, see, e.g. [6], [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Bing Li.

Consider the single-input single-output discrete dynamics case again. Given that the trial length is finite, one setting for ILC design for discrete systems is first to form so-called supervectors. Then, for example, the outputs along a trial can be assembled into an $N \times 1$ column vector. Completing this operation for the other variables enables the trial-to-trial error updating to be described by a difference equation in the trial number (k). This setting is the basis for many of the currently reported algorithms and applications and is referred to as lifting ILC design.

The trial length in ILC is finite. Hence, trial-to-trial error convergence can occur even if the system state matrix is unstable (as over such a duration, even an unstable system can only produce a bounded output). Also, in robust control, problems arise in the lifting setting due to matrix products formed from state-space matrices of the nominal model and those describing the uncertainty model used. In the lifting approach, the route is first to design a stabilizing feedback control loop and then apply ILC to the resulting dynamics, leading to a two-stage design.

In ILC, information propagation occurs in two directions, i.e., from trial-to-trial (k) and along the trial (p), and hence 2D systems theory can be applied. For discrete linear dynamics, [8] developed results using a state-space model recursive over the complete upper-right quadrant of the 2D plane. Repetitive processes [9] are another class of 2D systems where information propagation in one of the directions of information occurs over a finite duration and, therefore, a closer match to ILC dynamics.

The repetitive process setting for ILC design has seen experimental validation for laws based on linear models, including the robust case, see, e.g., [10], [11], (the last reference is based on a repetitive process setting for analysis). In robustness analysis, the difficulty present in the lifting-based designs does not arise. This setting allows the trial-to-trial error convergence and the along the trial dynamics to be considered in a one-stage design. It extends directly to design for differential dynamics.

A large majority of the ILC laws considered in the 2D systems setting (or others) have been static, e.g., the input for the subsequent trial is an additive linear combination of state feedback on the current trial and feedforward action formed using the error on the previous trial. Such a law assumes access to all entries in the current trial state vector. There have been results on replacing this term by the current trial output with supporting experimental validation, see, e.g. [12]. If a static control law cannot meet the performance specifications, one option is to allow internal dynamics in the control law, termed a dynamic controller.

In previous work [13], a dynamic controller was designed and experimentally verified. This controller requires access to both the state vector on the current trial, storage of the previous trial state vector over the complete trial length. The major novel contributions of this paper relative to previous work is i) a new design that does not use state vector information, and ii) an optimization procedure for tuning the design.

Throughout this paper, the null and identity matrices of compatible dimensions are denoted, respectively, by $\mathbf{0}$ and $\boldsymbol{I}$. Also, a symmetric positive definite (respectively negative definite) matrix, say $\boldsymbol{M}$, is denoted by $\boldsymbol{M} \succ 0$ (respectively $\boldsymbol{M} \prec 0$) and the spectral radius of a matrix is denoted by $\rho(\cdot)$. The next section gives the necessary background on repetitive processes.

II. BACKGROUND

Let $\boldsymbol{y_{ref}}(p)$ be a possibly vector valued reference representing the desired output behavior and denote the system output on trial k as $\boldsymbol{y}_k(p) \in \mathbb{R}^m$, $0 \leq p \leq \alpha - 1$, $k \geq 0$. Then the error on trial k is

$$\boldsymbol{e}_k(p) = \boldsymbol{y_{ref}}(p) - \boldsymbol{y}_k(p). \tag{1}$$

The control design problem is to construct an input sequence $\{\boldsymbol{u}_k\}_k$ that when applied forces the error sequence $\{\boldsymbol{e}_k\}$ to converge in k, i.e.,

$$\lim_{k\to\infty} ||\boldsymbol{e}_k|| = 0, \ \lim_{k\to\infty} ||\boldsymbol{u}_k - \boldsymbol{u}_\infty|| = 0, \tag{2}$$

where $||\cdot||$ is a signal norm in a suitably chosen function space with a norm-based topology, also, $\boldsymbol{u}_\infty$ is termed the learned control in some of the literature. It is also necessary to ensure that the dynamics produced along the trials (in p) are also acceptable, where this issue is application-dependent.

A commonly used ILC law has the form

$$\boldsymbol{u}_{k+1}(p) = \boldsymbol{u}_k(p) + \Delta(\boldsymbol{e}_k(p)), \tag{3}$$

where $\Delta(\boldsymbol{e}_k(p))$ is a correction term that, in particular, can use all information contained in the previous trial error $\boldsymbol{e}_k(p)$ given by (1) (and other information from this trial, e.g., the trial state vector). The ILC dynamics operate over $(k,p) \in [0,\infty] \times [0,\alpha-1]$, and has the structure of a 2D system. Consequently, there has been a considerable body of research on ILC law analysis and design using 2D systems theory, starting from [8], which used the Roesser model. Such models operate over $(k,p) \in [0,\infty] \times [0,\infty]$.

Repetitive processes are a distinct class of 2D linear systems that make a series of sweeps, termed trials in the ILC setting (but termed passes in the repetitive process literature), through dynamics defined over a finite duration, which is known as the trial length. The output produced on each trial is termed the trial profile. On completion of the current trial, the process resets to the starting location, and the next trial can begin, either immediately or after some further time has elapsed. The trial profile produced on the previous trial contributes to current trial dynamics, and the dynamics of these processes evolve over $(k,p) \in [0,\infty] \times [0,\alpha-1]$, and hence a closer match to ILC dynamics.

The state-space model of a discrete linear repetitive process over $0 \leq p \leq \alpha - 1$, $k \geq 0$, is

$$\begin{aligned}\boldsymbol{x}_{k+1}(p+1) &= \mathbb{A}\boldsymbol{x}_{k+1}(p) + \mathbb{B}\boldsymbol{u}_{k+1}(p) + \mathbb{B}_0\boldsymbol{y}_k(p),\\ \boldsymbol{y}_{k+1}(p) &= \mathbb{C}\boldsymbol{x}_{k+1}(p) + \mathbb{D}\boldsymbol{u}_{k+1}(p) + \mathbb{D}_0\boldsymbol{y}_k(p),\end{aligned} \tag{4}$$

where on trial k, $\boldsymbol{x}_k(p) \in \mathbb{R}^n$ is the state vector, $\boldsymbol{y}_k(p) \in \mathbb{R}^m$ is the trial profile vector and $\boldsymbol{u}_k(p) \in \mathbb{R}^r$ is the control input vector. The terms $\mathbb{B}_0\boldsymbol{y}_k(p)$ and $\mathbb{D}_0\boldsymbol{y}_k(p)$ describe, respectively, the contributions of the previous trial output to the state and trial profile vectors produced on the next trial. They have a critical role in repetitive process-based ILC analysis and design.

The boundary conditions for processes described by (4) used in this paper have the form

$$\begin{aligned} \boldsymbol{x}_{k+1}(0) &= \boldsymbol{d}_{k+1}, \quad k \geq 0, \\ \boldsymbol{y}_0(p) &= \boldsymbol{f}(p), \quad 0 \leq p \leq \alpha - 1, \end{aligned} \tag{5}$$

where $\boldsymbol{d}_{k+1} \in \mathbb{R}^n$ has known constant entries and $\boldsymbol{f}(p) \in \mathbb{R}^m$ has entries that are known functions of p, and includes the most common case when the state initial conditions for each trial are the same, e.g., in pick and place operations where the sequence of objects to be moved are at the same location. Likewise for the trial initial profile, which is specified by the user or in some cases taken as zero and hence the initial error is the reference trajectory. This paper considers the case when the dynamics of the system can be modeled for analysis and design as linear and time-invariant, which includes the vast majority of the applications that have led to experimental validation and implementation.

Let $\{\boldsymbol{y}_k\}_k$ denote the sequence of trial profiles generated by the considered system. Then this sequence can contain oscillations that increase in amplitude from trial-to-trial (in k). Moreover, standard control action cannot regulate this unwanted behavior, e.g., static state or output feedback control activated by the current trial state or trial profile vector. Instead, this must be augmented by a term activated by previous trial information, i.e., repetitive processes are a class of 2D systems and require a control law with current trial feedback and a feedforward term from the previous trial.

The stability theory for linear repetitive processes [9], in response to the growing oscillations from trial-to-trial (i.e., in k), is of the bounded-input bounded-output form. In particular, a bounded initial trial profile is required to produce a bounded sequence of trial profiles $\{\boldsymbol{y}_k\}$, where boundedness is defined in terms of the norm on the underlying function space. This theory enforces this property either over the finite and fixed trial length or for all possible trials lengths. This latter property can be analyzed mathematically by considering the case when $\alpha \to \infty$.

In many applications, the stronger stability conditions must be imposed, and the following result characterizes stability along the trial of examples described by (4) and (5).

Lemma 1 ([9]): A discrete linear repetitive processes described by (4) and (5) is stable along the trial if and only if (*i*) $\rho(\mathbb{D}_0) < 1$, (*ii*) $\rho(\mathbb{A}) < 1$, and (*iii*) all eigenvalues of $G(z) = \mathbb{C}(z\boldsymbol{I} - \mathbb{A})^{-1}\mathbb{B}_0 + \mathbb{D}_0$ have modulus strictly less than unity for all $|z| = 1$.

Asymptotic stability for the processes considered holds if and only if the first condition of this result holds. Hence the stability properties for examples defined by (4) and (5) are completely characterized by the state-space quadruple $\{\mathbb{A}, \mathbb{B}_0, \mathbb{C}, \mathbb{D}_0\}$.

The application of repetitive process stability theory to the design of ILC laws is based on the fact that the controlled dynamics can be written in the form (4) where the error sequence $\{\boldsymbol{e}_k\}_k$ is taken as the trial profile sequence. Use of this setting and applying Lemma 1 results in monotonic trial-to-trial error convergence with the regulation of the along the trial dynamics. If only (i) is enforced then, trial-to-trial error convergence can occur due to the finite trial length, even if the state matrix does not satisfy condition (ii). The following section proceeds to ILC design, where condition (iii) is replaced by an LMI (Linear Matrix Inequality) condition.

III. DESIGN OF STATIC ILC LAWS

In this paper, analysis and design is based on a discrete linear time-invariant model of the dynamics, written in the ILC setting as

$$\begin{aligned} \boldsymbol{x}_k(p+1) &= \boldsymbol{A}\boldsymbol{x}_k(p) + \boldsymbol{B}\boldsymbol{u}_k(p), \quad 0 \leq p \leq \alpha - 1 \\ \boldsymbol{y}_k(p) &= \boldsymbol{C}\boldsymbol{x}_k(p), \end{aligned} \tag{6}$$

where the notation used is the same as for (4), and the ILC law has the structure of (3). Introduce the following vector in terms of (6)

$$\boldsymbol{\eta}_{k+1}(p) = \boldsymbol{x}_{k+1}(p) - \boldsymbol{x}_k(p) \tag{7}$$

and consider the choice of

$$\Delta \boldsymbol{e}_k(p+1) = \boldsymbol{K_1}\boldsymbol{\eta}_{k\mathcal{C}1}(\boldsymbol{p}) + \boldsymbol{K_2}\boldsymbol{e}_k(p), \tag{8}$$

where $\boldsymbol{K_1}$ and $\boldsymbol{K_2}$ are compatibly dimensioned matrices to be found. Then in [14] a static ILC law was developed, starting from the controlled dynamics state-space model of the form (4) with the control input terms deleted, state vector $\boldsymbol{\eta}_{k+1}(p)$, the trial error as the output profile vector and

$$\begin{aligned} \mathbb{A} &= \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K_1}, \quad \mathbb{B}_0 = \boldsymbol{B}\boldsymbol{K_2}, \\ \mathbb{C} &= -\boldsymbol{C}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K_1}), \quad \mathbb{D}_0 = \boldsymbol{I} - \boldsymbol{C}\boldsymbol{B}\boldsymbol{K_2}. \end{aligned} \tag{9}$$

For implementation, the resulting control law is

$$\begin{aligned} \boldsymbol{u}_{k+1}(p) = \boldsymbol{u}_k(p) &+ \boldsymbol{K_1}(\boldsymbol{x}_{k+1}(p) - \boldsymbol{x}_k(p)), \\ &+ \boldsymbol{K_2}(\boldsymbol{y_{ref}}(p+1) - \boldsymbol{y}_k(p+1)), \end{aligned} \tag{10}$$

where the last term on the right-hand side is phase-lead ILC. Moreover, this term is implementable as the signal $\boldsymbol{y_{ref}}$ is known, and the second term in $p+1$ is available as it is generated on the previous trial. This term is the unique feature of ILC, and if such a term is not present, then it can be easily shown that the ILC law can be replaced by an equivalent feedback control loop.

Two applications relevant issues arise with this last law for general application. Firstly, it requires the availability of the state vector for measurement. Secondly, the complete previous trial state vector must be stored to calculate the control input for the subsequent trial. In the first case, an observer can be designed, and for ii) one alternative is to design a static control law based on trial output information only. This approach was considered in [12], starting from

$$\Delta(\boldsymbol{e}_k(p))=\boldsymbol{K_1}\boldsymbol{\mu}_{k+1}(p+1)+\boldsymbol{K_2}\boldsymbol{\mu}_{k+1}(p)+\boldsymbol{K_3}\boldsymbol{e}_k(p+1), \tag{11}$$

where $\boldsymbol{K_1}$, $\boldsymbol{K_2}$ and $\boldsymbol{K_3}$ are control law matrices to be designed and

$$\boldsymbol{\mu}_k(p)=\boldsymbol{y}_k(p-1)-\boldsymbol{y}_{k-1}(p-1)=\boldsymbol{C}\boldsymbol{\eta}_k(p) \tag{12}$$

and the extra term in this control law relative to the previous case has been added as a means, if necessary, of compensating for the effects of not assuming that the state vector is available for use.

Following the analysis in [12], the resulting control law for application is

$$\begin{aligned}\boldsymbol{u}_k(p) &= \boldsymbol{u}_{k-1}(p)+\boldsymbol{K_1}(\boldsymbol{y}_k(p)-\boldsymbol{y}_{k-1}(p))\\ &\quad+\boldsymbol{K_2}(\boldsymbol{y}_k(p-1)-\boldsymbol{y}_{k-1}(p-1))\\ &\quad+\boldsymbol{K_3}(\boldsymbol{y}_{ref}(p+1)-\boldsymbol{y}_{k-1}(p+1)).\end{aligned} \tag{13}$$

The last term in (13) is phase lead on the previous trial error and the second and third terms are proportional action acting on the error, respectively, between the current and previous trials at p and $p-1$. While the use of current trial data has appeared in many ILC designs to manipulate the plant dynamics along the trial, see e.g. [15] and has been found to increase initial tracking and disturbance rejection [16], the coupling of previous and current trial data is an addition to this class of updates.

The use of only static output information in an ILC law can be very restrictive in terms of the performance level. To illustrate this point, consider the case of a system described by the transfer-function

$$G(s)=\frac{4}{s^2+0.8s+4}. \tag{14}$$

Using a sampling period of $T_s=0.01$ secs gives the model of (6) with

$$\begin{aligned}\boldsymbol{A} &= \begin{bmatrix}0.9918 & -0.01992\\ 0.01992 & 0.9998\end{bmatrix},\\ \boldsymbol{B} &= \begin{bmatrix}0.01992\\ 0.0001995\end{bmatrix},\\ \boldsymbol{C} &= \begin{bmatrix}0 & 1\end{bmatrix}.\end{aligned} \tag{15}$$

Consider the application of the static control law (13) in the case when the design results in

$$\begin{aligned}\boldsymbol{K_1} &= 0.005018,\\ \boldsymbol{K_2} &= -23.56,\\ \boldsymbol{K_3} &= -0.00409.\end{aligned} \tag{16}$$

Figure 1 shows the reference signal for this case and Figures 2, 3 and 4 confirm the relatively poor performance

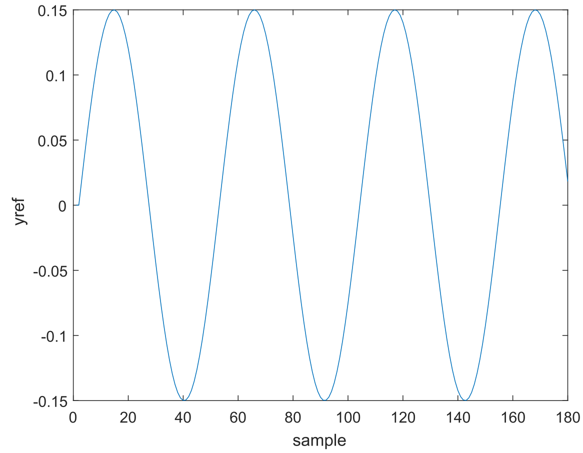


FIGURE 1. **The reference signal.**

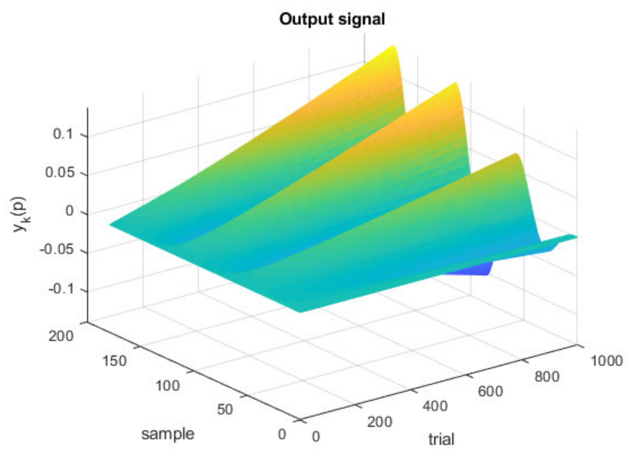


FIGURE 2. **Plot of the output against trial and sample numbers produced for the static control law defined by (13) and (16) applied to (14).**

of this design, with, in particular, very slow trial-to-trial error convergence. In the case of Figure 4

$$||\boldsymbol{e}_k|| = \sqrt{\frac{1}{\alpha}\sum_{p=0}^{\alpha-1}|\boldsymbol{e}_k|^2} \tag{17}$$

which is a common measure used in ILC to measure trial-to-trial error convergence.

In this paper, the controller matrices $\boldsymbol{K_1}$, $\boldsymbol{K_2}$, $\boldsymbol{K_3}$ are obtained by solving appropriately designed LMIs, such that the mean square error of (17) is minimized to achieve an ILC convergence, see, e.g. [14]. The main parameters related to the ILC practical implementation that should be taken into account are the convergence speed; the maximum allowed state, output, and input signals maximum values; the absence of oscillations (along the trial and from trial to trial), and also robustness against the system parameter changes. If a static control law fails to satisfy the design constraints then the premise of this paper is that a dynamic controller is one option to investigate.

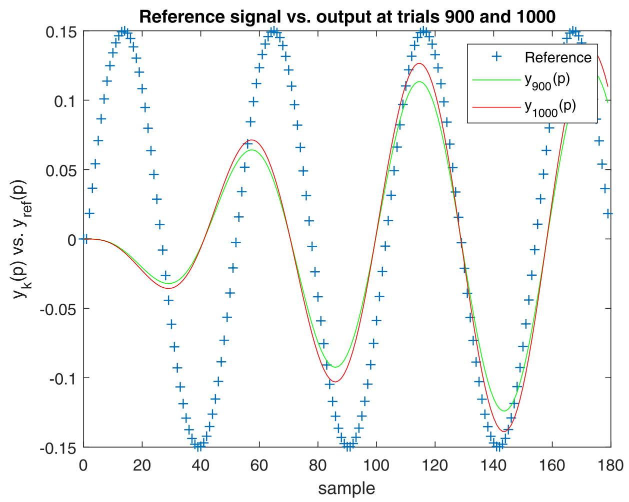


FIGURE 3. Outputs and the reference signal for the static control law (13) and (16) applied to (14).

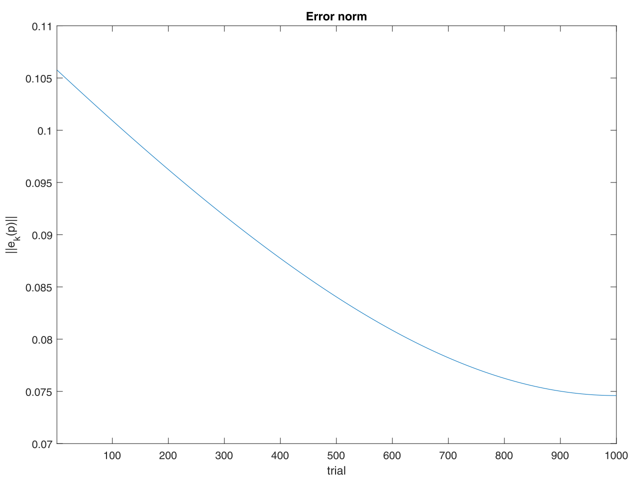


FIGURE 4. Norm of the error signal for the static control law (13) and (16) applied to (14).

IV. DESIGN OF DYNAMIC ILC LAWS

One option in cases such as the example in the last section is to consider the use dynamic controllers, i.e., they have their own internal (state) dynamics. Early papers on the use of such control laws in the ILC setting include [17]. Also there has been previous research reported on repetitive process based design of dynamic controllers [13], [18].

In [13], [18] the controller for implementation has the structure

$$\begin{aligned} \boldsymbol{u}_{k+1}(p) = \boldsymbol{u}_k(p) &+ \boldsymbol{C_c}\boldsymbol{\eta^c}_{k+1}(p+1) \\ &+ \boldsymbol{E_c}(\boldsymbol{x}_{k+1}(p) - \boldsymbol{x}_k(p)) \\ &+ \boldsymbol{D_c}(\boldsymbol{y_{ref}}(p+1) - \boldsymbol{y}_k(p+1)) \end{aligned} \quad (18)$$

where

$$\boldsymbol{\eta^c}_{k+1}(p) = \boldsymbol{x^c}_{k+1}(p-1) - \boldsymbol{x^c}_k(p-1)$$

and the superscript c denotes the controller dynamics. Also $\boldsymbol{x^c}_{k+1}(p-1)$ and $\boldsymbol{x^c}_k(p-1)$ denote, respectively, the controller state vector on trials $k+1$ and k. Implementation of this controller therefore requires access to the system state vector ($\boldsymbol{x}$) on both the current and previous trials, whereas the controller state dynamics ($\boldsymbol{x^c}$) are directly available. If this is not possible and state control action is still preferred, an observer will also be required. Moreover, the state vector on every two successive trials has to be stored to implement the controller.

The remainder of this section develops a new dynamic controller that requires only access to the trial output, where the controller state dynamics are described by

$$\begin{aligned} \boldsymbol{\eta^c}_{k+1}(p+1) = &\hat{\boldsymbol{A}}_{\boldsymbol{0,c}}\boldsymbol{C}\boldsymbol{\eta}_{k+1}(p) \\ &+ \boldsymbol{A_c}\boldsymbol{\eta^c}_{k+1}(p) + \boldsymbol{B_{0,c}}\boldsymbol{e}_k(p) \end{aligned} \quad (19)$$

and for the analysis purposes only, rewrite the state equation of (6) in the form

$$\boldsymbol{x}_k(p) = \boldsymbol{A}\boldsymbol{x}_k(p-1) + \boldsymbol{B}\boldsymbol{u}_k(p-1). \quad (20)$$

Using (7), (3) and (12), it follows immediately that $\boldsymbol{\mu}_k(p) = \boldsymbol{C}\boldsymbol{\eta}_k(p)$.

Suppose that the control signal is formed as a combination of feedback action on the current trial plus a component from the previous trial. One way to realize this is to define the ILC input increment as

$$\begin{aligned} \Delta\boldsymbol{u_{kC1}}(\boldsymbol{p}) = &\boldsymbol{C_c}\boldsymbol{\eta^c}_{k+1}(p+1) + \hat{\boldsymbol{E}}_{\boldsymbol{c}}\boldsymbol{\mu}_{k+1}(p+1) \\ &+ \boldsymbol{D_c}\boldsymbol{e_k}(p+1), \end{aligned} \quad (21)$$

where, in comparison to (18), the output increment $\boldsymbol{\mu}_{k+1}(p+1)$ is used and hence the replacement of $\boldsymbol{E_c}$ by $\hat{\boldsymbol{E}}_{\boldsymbol{c}}$. Using (6), (7) and (3) gives

$$\begin{aligned} \boldsymbol{u}_{k+1}(p) = \boldsymbol{u}_k(p) &+ \boldsymbol{C_c}\left(\boldsymbol{x^c}_{k+1}(p) + \boldsymbol{x^c}_k(p)\right) \\ &+ \hat{\boldsymbol{E}}_{\boldsymbol{c}}\left(\boldsymbol{y}_{k+1}(p) - \boldsymbol{y}_k(p)\right) \\ &+ \boldsymbol{D_c}\left(\boldsymbol{y_{ref}}(p+1) - \boldsymbol{y}_k(p+1)\right), \end{aligned} \quad (22)$$

The last term on the right-hand side is phase-lead ILC, i.e., the added benefit of this form of control is present. Moreover, the state vector of the dynamic controller $\boldsymbol{x^c}_k(p)$ is directly available and does not need to be estimated. Also (22) does not require controlled system state information; only knowledge of the measured input and output signals of the controlled system is used.

The controlled dynamics can be written as

$$\begin{aligned} \boldsymbol{\eta}_{k+1}(p+1) &= (\boldsymbol{A} + \boldsymbol{B}\hat{\boldsymbol{E}}_{\boldsymbol{c}}\boldsymbol{C})\boldsymbol{\eta}_{k+1}(p) \\ &\quad + \boldsymbol{B}\boldsymbol{C_c}\boldsymbol{\eta^c}_{k+1}(p) + \boldsymbol{B}\boldsymbol{D_c}\boldsymbol{e}_k(p), \\ \boldsymbol{\eta^c}_{k+1}(p+1) &= \hat{\boldsymbol{A}}_{\boldsymbol{0,c}}\boldsymbol{C}\boldsymbol{\eta}_{k+1}(p) + \boldsymbol{A_c}\boldsymbol{\eta^c}_{k+1}(p) \\ &\quad + \boldsymbol{B_{0,c}}\boldsymbol{e}_k(p), \\ \boldsymbol{e}_{k+1}(p) &= (-\boldsymbol{C}\boldsymbol{A} - \boldsymbol{C}\boldsymbol{B}\hat{\boldsymbol{E}}_{\boldsymbol{c}}\boldsymbol{C})\boldsymbol{\eta}_{k+1}(p) \\ &\quad - \boldsymbol{C}\boldsymbol{B}\boldsymbol{C_c}\boldsymbol{\eta^c}_{k+1}(p) \\ &\quad + (\boldsymbol{I} - \boldsymbol{C}\boldsymbol{B}\boldsymbol{D_c})\boldsymbol{e}_k(p), \end{aligned} \quad (23)$$

or, on introducing,

$$\mathbb{x}_k(p) = \begin{bmatrix} \eta_k(p) \\ {\eta^c}_k(p) \end{bmatrix}, \tag{24}$$

$$\begin{aligned} \begin{bmatrix} \eta_{k+1}(p+1) \\ {\eta^c}_{k+1}(p+1) \end{bmatrix} &= H_1 \begin{bmatrix} \eta_{k+1}(p) \\ {\eta^c}_{k+1}(p) \end{bmatrix} \\ &\quad + \begin{bmatrix} BD_c \\ B_{0,c} \end{bmatrix} e_k(p), \\ e_{k+1}(p) &= H_2 \begin{bmatrix} \eta_{k+1}(p) \\ \eta^c_{k+1}(p) \end{bmatrix} \\ &\quad + (I - CBD_c) e_k(p), \end{aligned} \tag{25}$$

where

$$H_1 = \begin{bmatrix} (A + B\hat{E}_{cC}) & BC_c \\ \hat{A}_{0,cC} & A_c \end{bmatrix},$$

$$H_2 = \begin{bmatrix} -CA - CB\hat{E}_{cC} & -CBC_c \end{bmatrix}.$$

Therefore, the controlled dynamics are described by the discrete linear repetitive process state-space model

$$\begin{aligned} \mathbb{x}_{k+1}(p+1) &= \mathbb{A}\mathbb{x}_{k+1}(p) + \mathbb{B}_0 e_k(p), \\ e_{k+1}(p) &= \mathbb{C}\mathbb{x}_{k+1}(p) + \mathbb{D}_0 e_k(p), \end{aligned} \tag{26}$$

where

$$\begin{aligned} \mathbb{A} &= \begin{bmatrix} A + B\hat{E}_{cC} & BC_c \\ \hat{A}_{0,cC} & A_c \end{bmatrix}, \\ \mathbb{B}_0 &= \begin{bmatrix} BD_c \\ B_{0,c} \end{bmatrix}, \\ \mathbb{C} &= \begin{bmatrix} -C(A + B\hat{E}_{cC}) & -CBC_c \end{bmatrix}, \\ \mathbb{D}_0 &= I - CBD_c. \end{aligned} \tag{27}$$

The following result is used in the proof of the theorem below.

Lemma 2 ([9]): A discrete linear repetitive process described by (26) and (27) is stable along the trial if there exist

$$\mathbb{P}_1 = \begin{bmatrix} \hat{P}_{11} & 0 \\ 0 & \hat{P}_{22} \end{bmatrix} \succ 0, \tag{28}$$

where $\hat{P}_{11}$ and $\hat{P}_{22}$ are, respectively, of the same dimensions as $\mathbb{A}$ and $\mathbb{D}_0$ and are such that

$$\begin{bmatrix} \mathbb{P}_1 - W - W^T & W\Phi^T \\ \Phi W & -\mathbb{P}_1 \end{bmatrix} \prec 0 \tag{29}$$

where

$$\Phi = \begin{bmatrix} \mathbb{A} & \mathbb{B}_0 \\ \mathbb{C} & \mathbb{D}_0 \end{bmatrix} \tag{30}$$

and W is a compatibly dimensioned matrix.

The main result of this paper is now established.

Theorem 1: The stability along the trial property holds for the ILC controlled dynamics described by (26) and (27) if there exist compatibly dimensioned matrices $\mathbb{P}_1 \succ 0$ and W satisfying Lemma 2 and compatibly dimensioned matrices N and X, with X invertible, such that the following LMI is feasible

$$\begin{aligned} &\begin{bmatrix} \mathbb{P}_1 - W - W^T & (\hat{A}W + \hat{B}N(\hat{C} + L))^T \\ \hat{A}W + \hat{B}N(\hat{C} + L) & -\mathbb{P}_1 \end{bmatrix} \prec 0, \\ &\hat{C}W = X(\hat{C} + L), \end{aligned} \tag{31}$$

where

$$\hat{A} = \begin{bmatrix} A & 0 & 0 \\ 0 & 0 & 0 \\ -CA & 0 & I \end{bmatrix}, \tag{32}$$

$$\hat{B} = \begin{bmatrix} B & 0 \\ 0 & I \\ -CB & 0 \end{bmatrix}, \tag{33}$$

$$\hat{C} = \begin{bmatrix} C & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}. \tag{34}$$

and L is a known matrix of compatible dimensions. If this LMI is feasible, the following matrix defines the controller of (22)

$$K = \begin{bmatrix} \hat{E}_c & C_c & D_c \\ \hat{A}_{0,c} & A_c & B_{0,c} \end{bmatrix}, \tag{35}$$

where $K = NX^{-1}$.

Proof: Rewrite (30) using (27) as

$$\begin{aligned} &\Phi \\ &= \begin{bmatrix} A + B\hat{E}_{cC} & BC_c & BD_c \\ \hat{A}_{0,cC} & A_c & B_{0,c} \\ -CA - CB\hat{E}_{cC} & -CBC_c & I - CBD_c \end{bmatrix}, \end{aligned} \tag{36}$$

or, equivalently,

$$\begin{aligned} &\Phi \\ &= \begin{bmatrix} A & 0 & 0 \\ 0 & 0 & 0 \\ -CA & 0 & I \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & I \\ -CB & 0 \end{bmatrix} \begin{bmatrix} \hat{E}_{cC} & C_c & D_c \\ \hat{A}_{0,cC} & A_c & B_{0,c} \end{bmatrix} \end{aligned} \tag{37}$$

Hence

$$\Phi W = \hat{A}\hat{W} + \hat{B}K\hat{C}W, \tag{38}$$

and $\hat{A}$, $\hat{B}$, $\hat{C}$ and K are given, respectively, by (32), (33), (34) and (35).

The constraint

$$\hat{C}W = X(\hat{C} + L), \tag{39}$$

enables (38) to be written as

$$\Phi W = \hat{A}W + \hat{B}KX(\hat{C} + L) \tag{40}$$

Finally, setting

$$N = KX \tag{41}$$

yields

$$\Phi W = A\hat{W} + \hat{B}N(\hat{C} + L) \tag{42}$$

Applying (42) to (29) and using (41) (with $\boldsymbol{K} = \boldsymbol{N}\boldsymbol{X}^{-1}$) complete the proof.

The partitioned form of $\boldsymbol{K}$ in this last result gives the matrices required to implement the control law (22). Note that the matrix $\boldsymbol{L}$ of (31) has known constant entries selected before solving the LMI of (31). Its introduction provides extra freedom in obtaining the controller matrices of (35). The first attempt can be made using $\boldsymbol{L} = \boldsymbol{0}$. If the solution does not yield satisfactory convergence speed, the optimization algorithm developed next can be used.

A. INCREASING THE CONVERGENCE SPEED

If the choice of $\boldsymbol{L} = \boldsymbol{0}$ is not successful, minimization of the following objective function should be undertaken, starting with $\boldsymbol{L} = \boldsymbol{0}$

$$F(\boldsymbol{E}, \boldsymbol{U}, \boldsymbol{\Delta}) = \sum_{p=0}^{\alpha-1}\Bigg(\boldsymbol{\Delta}(p)\boldsymbol{\Delta}^T(p) + \sum_{k=1}^{\beta-1}\left(\boldsymbol{e}_k(p)\boldsymbol{e}_k^T(p)\cdot\delta + \boldsymbol{u}_k(p)\boldsymbol{u}_k^T(p)\cdot\gamma\right)\Bigg), \tag{43}$$

where

$$\begin{aligned}\boldsymbol{E} &= \left[\boldsymbol{e}_1^T, \boldsymbol{e}_2^T, \ldots, \boldsymbol{e}_{\beta-1}^T\right]^T,\\ \boldsymbol{U} &= \left[\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \ldots, \boldsymbol{u}_{\beta-1}^T\right]^T,\\ \boldsymbol{\Delta} &= \left[\Delta\boldsymbol{u}_\beta^T(0), \Delta\boldsymbol{u}_\beta^T(1), \ldots, \Delta\boldsymbol{u}_\beta^T(\alpha-1)\right]^T,\end{aligned} \tag{44}$$

where β denotes the number of trials considered and $\Delta\boldsymbol{u}_k(p)$ is defined in (21). The sum over this number of trials aims to provide better noise rejection, and the parameter δ provides a balance between both components, which is application-dependent. The rationale behind the inclusion of the term in $\boldsymbol{\Delta}$ of (44) in this cost function is that it penalizes solutions that have oscillations on the last considered trial (i.e., the changes in $\boldsymbol{u}$ for the next trial are not trivially small). It is assumed that the number of simulated trials β is large enough to obtain a reasonably small error, and below $\beta = 50$ is used and also $\delta = 100$ and $\gamma = 10^{-3}$.

Many numerical algorithms can be used to minimize this objective function. In this paper, a Nelder-Mead (NM) algorithm is used, see, e.g., [19] for the relevant background.

An algorithm for implementing the design is formed by the following steps.

1) Set $\boldsymbol{L} = \boldsymbol{0}$, solve (31) and calculate $\boldsymbol{K}$ using (35).
2) Simulate the controlled system formed by applying the control law (22).
3) Evaluate the results and if they are acceptable — END. If not, implement the refinement phase given by the following steps.
4) Calculate the initial value of the performance index (43), taking into account (44), using the data from the last step.
5) Modify the value of $\boldsymbol{L}$ using an optimization algorithm rule (e.g., apply the Nelder-Mead algorithm)
6) For the new value of $\boldsymbol{L}$, solve (31) and calculate $\boldsymbol{K}$ of (35). (Note that in this step $\boldsymbol{L}$ does not change during the solving of the LMI of (31)).
7) Simulate the controlled system formed by applying the control law (22).
8) Calculate the performance index of (43) taking into account (44).
9) If the value of (43) calculated in the previous step is small enough — END, otherwise go to Step 5.

B. CASE STUDIES

The poor performance of the static control law of (13) for the model of (14) shows the need for an alternative control law. One solution is to use the dynamic control law of (22). For a fair comparison, it is assumed that $\boldsymbol{L} = \boldsymbol{0}$. Completing the design of Theorem 1 gives the following dynamic controller state-space model matrices

$$\begin{aligned}\boldsymbol{A_c} &= \begin{bmatrix}-0.1189 & 0.4409\\ -0.2733 & 1.096\end{bmatrix},\\ \hat{\boldsymbol{A}}_{\boldsymbol{0},\boldsymbol{c}} &= \begin{bmatrix}-85.15 & 7880\\ -389.9 & 1734\end{bmatrix},\\ \boldsymbol{B_{0,c}} &= \begin{bmatrix}532.7\\ 113.8\end{bmatrix},\\ \boldsymbol{C_c} &= \begin{bmatrix}-6.384 & -0.1585\end{bmatrix},\\ \boldsymbol{D_c} &= 3644,\\ \hat{\boldsymbol{E}}_{\boldsymbol{c}} &= [-50.092 - 5266.5254].\end{aligned} \tag{45}$$

Figures 5–8 give performance results for the controlled dynamics and confirm that in comparison to the static controller, better performance is achieved. This design delivers much improved performance with relatively fast trial-to-trial error convergence (As in the static controller case, the simulation was performed for 2000 trials. To better visualize the convergence, only the first 10 trials are shown (the remaining results show almost no changes in the signals)).

The considered system in the last example is a second order lag. To examine the new design for a more complex case and the role of the matrix $\boldsymbol{L}$, consider the example where

$$\boldsymbol{A} = \begin{bmatrix} 0.3879 & 1 & 0 & 0 & 0 & 0 & 0\\ -0.3898 & 0.3879 & -0.03809 & -0.5504 & -0.2367 & 0.4823 & 0.8585\\ 0 & 0 & -0.1575 & 1 & 0 & 0 & 0\\ 0 & 0 & -0.07758 & -0.1575 & -0.1772 & 0.361 & 0.6427\\ 0 & 0 & 0 & 0 & 0.03529 & 1 & 0\\ 0 & 0 & 0 & 0 & -0.008214 & 0.03529 & 0.6258\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

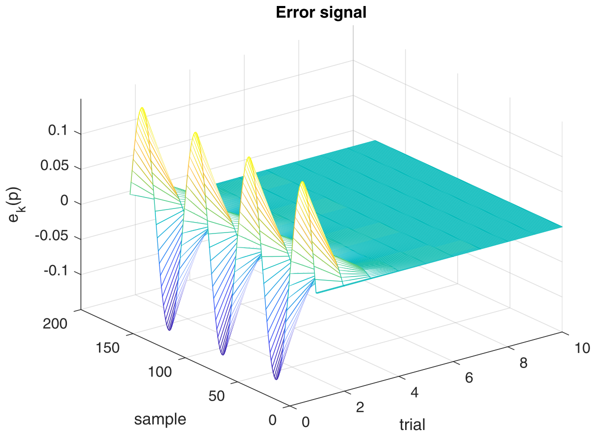


FIGURE 5. Trial errors plotted against sample and trial number for the dynamic controller (22) and (45) applied to (14).

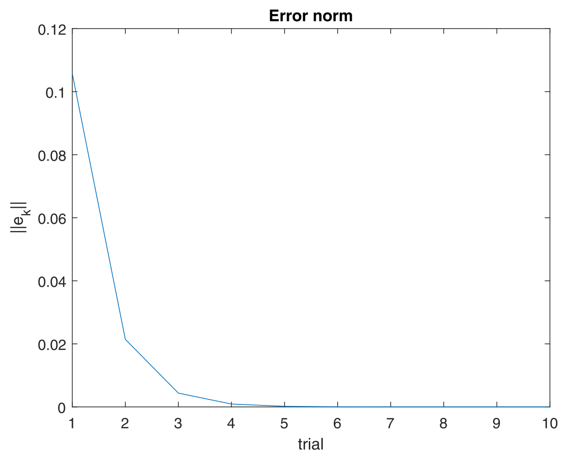


FIGURE 6. Trial-to-trial error norm plotted against the trial number for the dynamic controller (22) and (45) applied to (14).

$$\boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.03125 \end{bmatrix},$$

$$\boldsymbol{C}^T = \begin{bmatrix} 0.008985 \\ 0.01101 \\ -0.0007348 \\ -0.01062 \\ -0.004565 \\ 0.009302 \\ 0.01656 \end{bmatrix}. \tag{46}$$

In this case solving the LMI of (31) with $\boldsymbol{L} = \boldsymbol{0}$ results in the controller state-space model matrices:

$$\boldsymbol{A_c} = \begin{bmatrix} 1.784 \cdot 10^{-6} & -2.491 \cdot 10^{-7} \\ 1.074 \cdot 10^{-6} & -8.448 \cdot 10^{-7} \end{bmatrix},$$

$$\hat{\boldsymbol{A}}_{\boldsymbol{0,c}} = \begin{bmatrix} 0.0007592 \\ -0.0002012 \end{bmatrix},$$

$$\boldsymbol{B_{0,c}} = \begin{bmatrix} -1.096 \cdot 10^{-6} \\ 6.252 \cdot 10^{-6} \end{bmatrix},$$

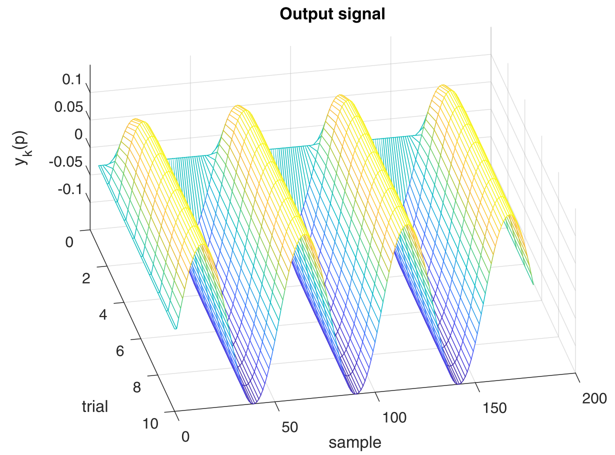


FIGURE 7. Output plotted against the sample and trial numbers for the dynamic controller (22) and (45) applied to (14).

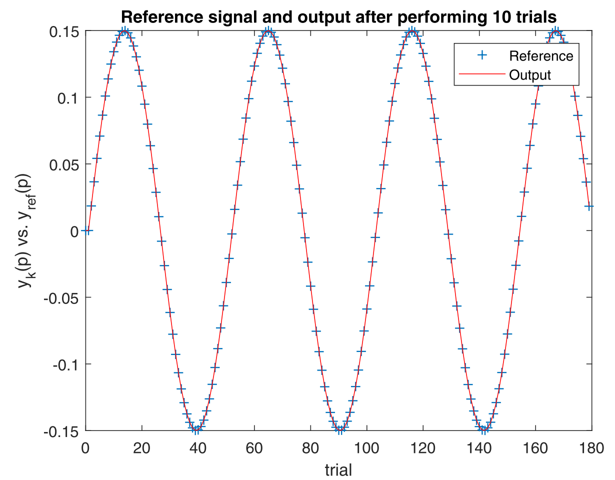


FIGURE 8. Comparison of the output versus the reference signal for the dynamic controller (22) and (45) applied to (14).

$$\begin{aligned} \boldsymbol{C_c} &= \begin{bmatrix} 0.000502 & -0.0003922 \end{bmatrix}, \\ \boldsymbol{D_c} &= 1.496, \\ \hat{\boldsymbol{E}}_{\boldsymbol{c}} &= -260.8. \end{aligned} \tag{47}$$

Some of the entries in matrices of (47) are very small in magnitude (especially $\boldsymbol{A_c}$ and $\hat{\boldsymbol{A}}_{\boldsymbol{0,c}}$), which manifests itself in a very slow trial-to-trial error convergence speed — see Figs. 9, 10 and 11. In effect, the controller has negligible dynamics and the final result is relatively poor.

Completing the optimization algorithm of the previous section using the Nelder-Mead [19] technique results in

$$\boldsymbol{L} = \begin{bmatrix} 0.1168 & 0.8661 & 0.216 & 0.7638 & 0.4054 & 0.7959 & 0.5272 & 0.02071 & 0.05958 & 0.8079 \\ 0.9728 & 0.8352 & 0.8106 & 0.1526 & 0.7493 & 1.018 & 0.7757 & 0.6388 & 0.7943 & 0.853 \\ 0.08891 & 0.8638 & 0.9609 & 0.7989 & 0.06456 & 0.01209 & 0.7165 & 0.6928 & 0.438 & 0.5252 \\ 0.9997 & 0.226 & 0.6708 & 0.6893 & 0.6377 & 0.4754 & 0.6461 & 0.1653 & 0.9838 & 0.4747 \end{bmatrix}. \tag{48}$$

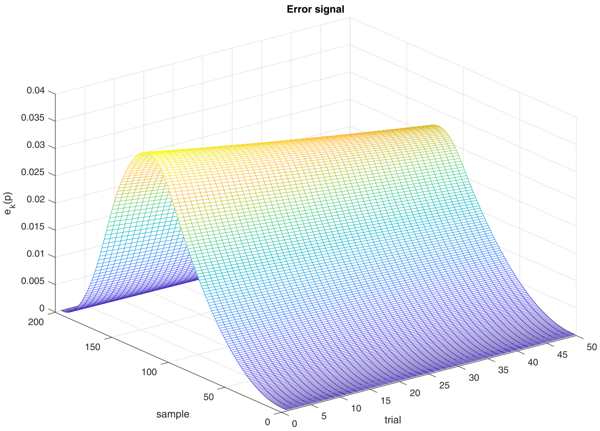


FIGURE 9. Error signal over 50 trials for $L = 0$.

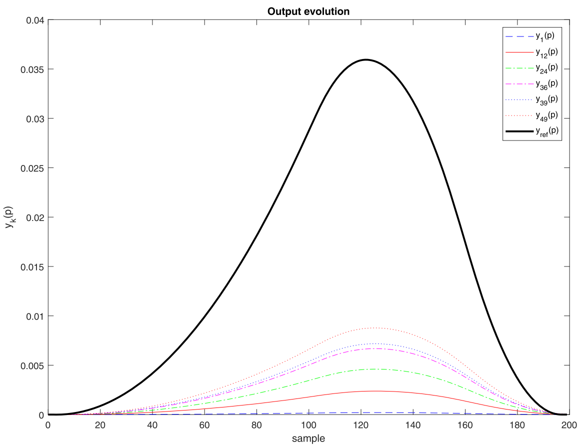


FIGURE 10. Output evolution for $L = 0$.

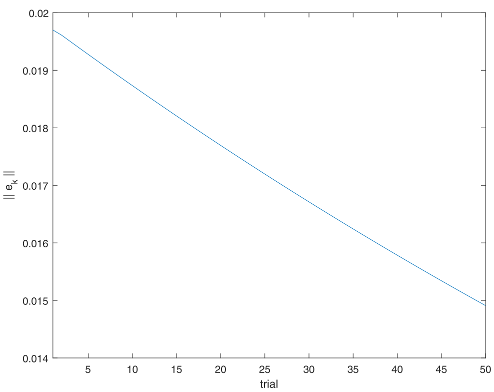


FIGURE 11. Error norm for each trial for the dynamic controller (22) and (47) applied to (46) for $L = 0$.

Solving the LMI of (31) for state-space model matrices of (46) and (48) gives the controller state-space model matrices:

$$\begin{aligned} A_c &= \begin{bmatrix} -0.5525 & 0.4766 \\ 0.1242 & -0.2368 \end{bmatrix}, \\ \hat{A}_{0,c} &= \begin{bmatrix} 62.19 \\ 9.57 \end{bmatrix}, \quad B_{0,c} = \begin{bmatrix} 1.653 \\ 1.801 \end{bmatrix}, \\ C_c &= \begin{bmatrix} 13.04 & -15.06 \end{bmatrix}, \\ D_c &= 97.37, \quad \hat{E}_c = -779.1, \end{aligned} \tag{49}$$

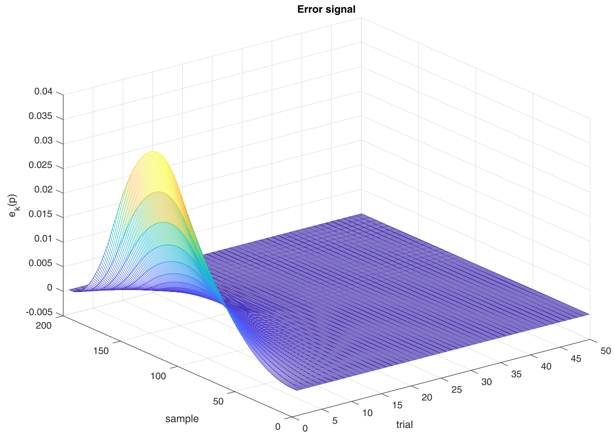


FIGURE 12. Error signal across 50 trials for $L \neq 0$.

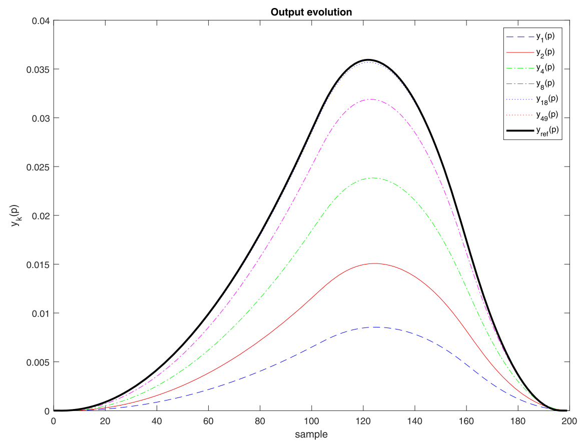


FIGURE 13. Output evolution for $L \neq 0$.

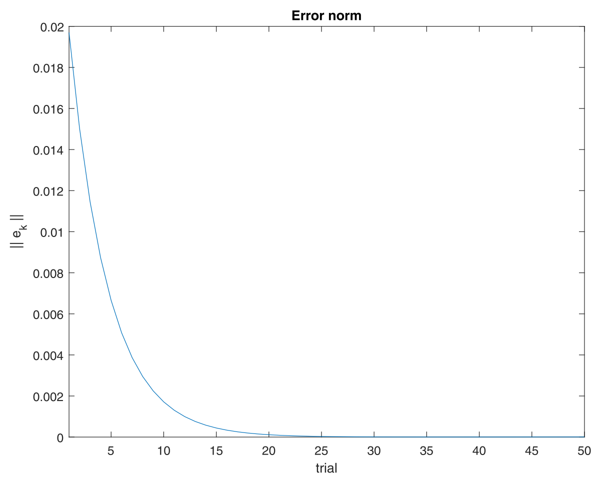


FIGURE 14. Norm of error for each trial for $L \neq 0$.

This yields the simulation results shown on Figs. 12–14. From Fig. 13, 18 trials are enough to achieve very high quality tracking.

V. CONCLUSION

This paper has reported substantial new results on the design of dynamic controllers for ILC applications. The new design removes the need to store state dynamics for implementation and reduces data storage needed for implementation. Controller design uses repetitive process stability theory as a starting point and leads to an LMI based condition for controller design. The benefits of this design have been compared with the previously known design of this structure, and the possibility of experimental validation is now feasible. This work is conditional on a choice to apply dynamic output ILC. The relative benefits against alternative control law structures are a matter for judgement based on knowledge of the particular application under consideration. Possible future research could focus on improving the quality of the solution emerging from the LMI solver. As a necessary first step, a physically motivated optimization procedure has been developed and its benefits demonstrated. Further development could consider design in the presence of noise and also robust design.

REFERENCES

[1] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robot. Syst.*, vol. 1, no. 2, pp. 123–140, 1984.

[2] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.

[3] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1099–1121, Nov. 2007.

[4] P. M. Sammons, M. L. Gegel, D. A. Bristow, and R. G. Landers, "Repetitive process control of additive manufacturing with application to laser metal deposition," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 2, pp. 566–577, Mar. 2019.

[5] L. G. Dekker, J. A. Marshall, and J. Larsson, "Experiments in feedback linearized iterative learning-based path following for center-articulated industrial vehicles," *J. Field Robot.*, vol. 36, no. 5, pp. 955–972, Aug. 2019.

[6] S.-E. Sakariya, C. T. Freeman, and K. Yang, "Iterative learning control of functional electrical stimulation in the presence of voluntary user effort," *Control Eng. Pract.*, vol. 96, pp. 1–11, Mar. 2020.

[7] T. Seel, C. Werner, J. Raisch, and T. Schauer, "Iterative learning control of a drop foot neuroprosthesis—Generating physiological foot motion in paretic gait by automatic feedback control," *Control Eng. Pract.*, vol. 48, pp. 87–97, Mar. 2016.

[8] J. E. Kurek and M. B. Zaremba, "Iterative learning control synthesis based on 2-D system theory," *IEEE Trans. Autom. Control*, vol. 38, no. 1, pp. 121–125, Jan. 1993.

[9] E. Rogers, K. Galkowski, and D. H. Owens, *Control Systems Theory and Applications for Linear Repetitive Processes* (Lecture Notes in Control and Information Sciences), vol. 349. Berlin, Germany: Springer-Verlag, 2007.

[10] W. Paszke, E. Rogers, K. Galkowski, and Z. Cai, "Robust finite frequency range iterative learning control design and experimental verification," *Control Eng. Pract.*, vol. 21, no. 10, pp. 1310–1320, 2013.

[11] J. Bolder and T. Oomen, "Inferential iterative learning control: A 2D-system approach," *Automatica*, vol. 71, pp. 247–253, Sep. 2016.

[12] L. Hladowski, K. Galkowski, Z. Cai, E. Rogers, C. T. Freeman, and P. L. Lewin, "Output information based iterative learning control law design with experimental verification," *ASME J. Dyn. Syst., Meas., Control*, vol. 134, no. 2, pp. 021012-1–021012-10, Mar. 2012.

[13] L. Hladowski, K. Galkowski, W. Nowicka, and E. Rogers, "Repetitive process based design and experimental verification of a dynamic iterative learning control law," *Control Eng. Pract.*, vol. 46, pp. 157–165, Jan. 2016.

[14] L. Hladowski, K. Galkowski, Z. Cai, E. Rogers, C. T. Freeman, and P. L. Lewin, "Experimentally supported 2D systems based iterative learning control law design for error convergence and performance," *Control Eng. Pract.*, vol. 18, no. 4, pp. 339–348, 2010.

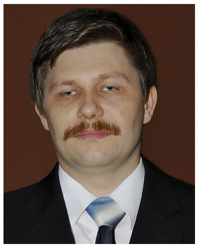
[15] M. Norrlof and S. Gunnarsson, "A frequency domain analysis of a second order iterative learning control algorithm," in *Proc. 38th IEEE Conf. Decis. Control*, Dec. 1999, pp. 1587–1592.

[16] J. D. Ratcliffe, J. J. Hätönen, P. L. Lewin, E. Rogers, T. J. Harte, and D. H. Owens, "P-type iterative learning control for systems that contain resonance," *Int. J. Adapt. Control Signal Process.*, vol. 19, no. 10, pp. 769–796, 2005.

[17] D. De Roover, O. H. Bosgra, and M. Steinbuch, "Internal-model-based design of repetitive and iterative learning controllers for linear multivariable systems," *Int. J. Control*, vol. 73, no. 10, pp. 914–929, 2000.

[18] L. Hladowski, K. Galkowski, and E. Rogers, "2D systems based dynamic iterative learning control design with experimental validation on a 3D crane model," *IFAC Papers Online*, vol. 52, no. 29, pp. 332–337, Jan. 2019.

[19] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM J. Optim.*, vol. 9, no. 1, pp. 112–147, 1998.



LUKASZ HLADOWSKI received the Ph.D. degree from the University of Zielona Góra, Poland, in 2001. His publication record counts over 35 papers in the world-known journals and conference proceedings, such as the *International Journal of Applied Mathematics and Computer Science*, *Multidimensional Systems, Signals Processing*, and the *International Journal of Control*. His research interests include multidimensional (nD) systems, iterative learning control, linear repetitive processes, and the practical application of numerical and optimization methods to control problems for ILC and nD systems.

KRZYSZTOF GALKOWSKI (Member, IEEE) is currently a Professor of technical sciences with the University of Zielona Góra. He has been on the same level with Nicolaus Copernicus University, Torun, Poland. He has authored over 100 articles in leading peer-reviewed journals and over 200 papers in the proceedings of major international conferences. His research interests include the multidimensional (nD) systems and control theory and applications for repetitive processes, including application to iterative learning control with experimental validation. He is also a member of the editorial boards of the *International Journal of Control* and *Multidimensional Systems and Signal Processing*. He is also the coauthor/an editor of four research monographs/books and also a Guest Editor of special issues in leading international journals, including the *International Journal of Control*, Multidimensional Systems and Signal Processing, and the Asian Journal of Control. He is also a Co-Founder of the international workshops on multidimensional systems (NDS). He is also a Visiting Professor with the School of Electronics and Computer Science, University of Southampton, U.K., and was three times a Gerhard Mercator Guest Professor, funded by DFG, in The University of Wuppertal, Germany, (September 2004–August 2005), (October 2006–September 2007), and (March 2011–February 2012). From 2011 to 2014, he was a Chair Professor with the East China University of Science and Technology, Shanghai, China.

ERIC ROGERS (Life Member, IEEE) was born in Northern, Ireland. He received the bachelor's degree in mechanical engineering from Queen's University Belfast and the master's and Ph.D. degrees in control systems from The University of Sheffield, U.K. He is currently a Professor in control systems theory and design with the School of Electronics and Computer Science, University of Southampton, U.K. His current research interests include multidimensional systems theory and applications, iterative learning control and applications, including application to robotic-assisted stroke 123 multidimensional systems and signal processing (2019) (30:2185–2197 2197) rehabilitation with supporting clinical trials and smart rotor control for wind turbines and estimation and control of autonomous underwater vehicles with particular interests in long range autonomy and close sea bed following. He is also the Editor-in-Chief of the *International Journal of Control and Multidimensional Systems and Signal Processing*.