INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/acs.2299

A common setting for the design of iterative learning and repetitive controllers with experimental verification

Chris T Freeman*,†, Muhammad Ali Alsubaie, Zhonglun Cai,
Eric Rogers and Paul L Lewin

Electronics and Computer Science, University of Southampton, University Road, Southampton, SO17 1BJ, UK

SUMMARY

Previous work has shown that the structure of repetitive control (RC) and that of iterative learning control (ILC) differ only in the location of an internal model of the disturbance. In this paper, it is shown how this common setting permits derivation of controllers in one domain based on an existing controller in the other. This is illustrated using the following case studies: 1) an RC scheme using a state feedback structure is derived based on an existing ILC scheme, which also uses state feedback (the underlying structure is first extended to comprise both current-error feedback and previous-error feedforward implementations); and 2) an ILC scheme using an output injection structure is developed based on an existing RC scheme, which uses state feedback, but differs from 1) by containing only a single internal model representation. All controllers are shown to have similar equivalent representations, with parameters derived by using linear quadratic regulator analysis. This correspondence enables comparison of the effect of the structure (ILC or RC, state feedback or output injection), number of internal models, and use of error (feedback or feedforward) on subsequent performance. This is undertaken using experimental results obtained using a gantry robot. Copyright © 2012 John Wiley & Sons, Ltd.

Received 15 July 2011; Revised 9 April 2012; Accepted 13 April 2012

KEY WORDS: iterative learning and repetitive control; common design; experimental verification

1. INTRODUCTION

Iterative learning control (ILC) is applicable to systems operating in a repetitive, or trial-to-trial mode with the requirement that a reference trajectory defined over a finite duration is followed to a high precision. A generic example is a pick and place operation where a physical device, such as a robot, is required to undertake the following operations in synchronization with a conveyor system: (i) collect an object from a fixed location; (ii) transfer it over a finite duration; (iii) place it on the moving conveyor; (iv) return to the original location for the next object; and then (v) repeat the previous four steps for as many objects as required. Each execution is known as a trial and the characteristic feature of ILC is that information from the previous trial, or a finite number of them, is used to update the control input to be used on the next trial where, for example, the new control signal is computed in the stoppage time between the completion of a trial and the start of the next one. The aim is to improve performance from trial-to-trial, commonly in the form of the error between a specified reference signal and the output on each trial. One starting point for the literature are the survey papers [1, 2].

Repetitive Control (RC) is concerned with improving the performance of trajectory tracking and set-point control problems in the case where the reference trajectory is repeated ad infinitum or the

*Correspondence to: Chris T Freeman, Electronics and Computer Science, University of Southampton, University Road, Southampton, SO17 1BJ, UK.

†E-mail: cf@ecs.soton.ac.uk

disturbances are repeated and occur in a regularly timed manner, for example, [3]. In this framework, there is no resetting between trials, each following immediately on from the last. This form of control utilizes the error data from previous executions of the task, to modify the input profile, which is supplied to the plant during the next trial, in a way which seeks to reduce the tracking error. Tracking systems in many applications, such as rotating machinery, have to operate in an environment containing such periodic reference and/or disturbance signals.

It is well-known that any periodic signal can be generated by an autonomous system consisting of a positive feedback loop with a time-delay element in the forward path. Furthermore, the internal model principle [4] establishes that such signals can be accommodated by duplicating this model inside a feedback loop. Previous work [5, 6] has shown that both RC and ILC contain an internal model of the reference, but differ in where it is located within the feedback path: in RC, the internal model is located at the system output, and in ILC, it is located at the system input. In [6], this structure was used to derive ILC and RC schemes, which explicitly incorporated current-error feedback, and the use of a single internal model. The placement of the internal model was shown to restrict RC to assuming a state-feedback structure, and ILC to assuming an output injection structure.

The ability to treat RC and ILC in the same framework is appealing as controllers found to operate well in one area can be synthesized for application to the other. However, no experimental comparison has been undertaken to establish the feasibility of this common framework in practice, or the effect of the underlying structural choices on subsequent performance. This paper shows how existing approaches can be used to develop alternative structure RC or ILC schemes in a common setting and considers the effects of the number of internal models on the structure and resulting performance. Hence, the results are of relevance to both domains.

The next section gives the necessary background with particular emphasis on the placement of the internal model in RC and ILC. Section 3 then expands an existing ILC structure and uses the common framework to develop a new RC scheme. Both schemes adopt state feedback to solve the stabilization and hence tracking problem. In Section 4, the case of a single internal model is then considered; first, an existing RC scheme from [6] is taken, and a novel ILC scheme is derived in a common setting. The stipulation of a single internal model is more restrictive, and forces an output injection structure. Compared with the ILC scheme outlined in [6], this has a more general yet simpler overall structure, and leads to a derivation which addresses parameter selection using the same form of cost function as the RC case. All four control schemes have similar equivalent structures and are stabilized using the same form of a linear quadratic regulator (LQR) cost function. Hence, for the first time. this permits transparent comparisons to be performed in terms of ILC/RC state feedback/output injection structure, number of internal models, and presence of error feedback/feedforward control loops. This is conducted in the penultimate section using experimental results with a multi-axis gantry robot, with conclusions given in Section 6.

2. BACKGROUND

The processes considered in this work have m outputs, p inputs, n states and are described by the discrete linear time-invariant state-space model with realization $\{A, B, C, D\}$, with transfer-function matrix description $y(z) = P(z)u(z)$ where

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

in which I_n is the $n \times n$ identity matrix. In ILC, the dynamics are considered over the finite trial duration consisting of N samples, with a constant initial state vector. The system performs a trial, resets to the initial state, and then performs the next. Hence the dynamics on the kth trial can be described by

$$\begin{aligned} x_k(i+1) &= Ax_k(i) + Bu_k(i), \quad i = 0, 1, \cdots N-1, \quad x_k(0) = x_0 \\ y_k(i) &= Cx_k(i) + Du_k(i) \end{aligned} \tag{2}$$

DOI: 10.1002/acs

where no loss of generality arises from setting $x_0 = 0$. By introducing the vectors

$$\begin{aligned} y_k &= \begin{bmatrix} y_k(0)^T & y_k(1)^T & \cdots & y_k(N-1)^T \end{bmatrix}^T \\ u_k &= \begin{bmatrix} u_k(0)^T & u_k(1)^T & \cdots & u_k(N-1)^T \end{bmatrix}^T \end{aligned} \tag{3}$$

the dynamics can also be expressed in equivalent 'lifted' form

$$y_k = Gu_k, \quad k = 0, 1, \cdots \tag{4}$$

where the $Nm \times Np$ lower triangular Toeplitz matrix

$$G = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-2}B & CA^{N-3}B & CA^{N-4}B & \cdots & D \end{bmatrix}. \tag{5}$$

The objective in ILC is to force the output to converge to the reference vector

$$r = \begin{bmatrix} r(0)^T & r(1)^T & \cdots & r(N-1)^T \end{bmatrix}^T. \tag{6}$$

This problem formulation has been used to develop many ILC algorithms such as those of the 'lifted' form, for example, [7, 8],

$$u_{k+1} = u_k + Qe_k \tag{7}$$

where $e_k = r - y_k$ is the error vector on trial k and Q is an $Np \times Nm$ matrix. Using (4), the error evolves as

$$e_{k+1} = (I_{Nm} - GQ)e_k \tag{8}$$

and hence, a necessary and sufficient criterion for convergence to the reference as $k \to \infty$ is

$$\rho(I_{Nm} - GQ) < 1 \tag{9}$$

where $\rho(\cdot)$ denotes the spectral radius of its matrix argument. Note that this imposes the constraint $p \geqslant m$. Many methods are available in the literature to design Q, of which those termed 'past-error feedforward', form a large subset [1, 2, 9], and yield the upper triangular Toeplitz structure

$$Q = \begin{bmatrix} Q_1 & Q_2 & Q_3 & \cdots & Q_N \\ 0 & Q_1 & Q_2 & \cdots & Q_{N-1} \\ 0 & 0 & Q_1 & \cdots & Q_{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & Q_1 \end{bmatrix} \tag{10}$$

where each block element is an $p \times m$ matrix. The p rows of (7) corresponding to the input vector at sample i of trial $k + 1$ can be equivalently written in z-transform form as

$$u_{k+1}(i) = z^{-N}u_{k+1}(i) + z^{-N}\left(Q_1 + zQ_2 + \cdots + z^{N-1}Q_N\right)e_{k+1}(i), \quad i = 0, 1, \cdots N-1 \tag{11}$$

where the z-transform of the sequences e_{k+1} and u_{k+1} is not explicitly stated. Here, the previous trial signals are treated as a set of initial conditions, which are imposed on the shift operator, such that

$$z^{-j}u_{k+1}(i) = u_k(N + i - j), \quad i = 0, 1, \cdots N - 1, \quad i < j \leqslant N \tag{12}$$

$$z^{-j}e_{k+1}(i) = e_k(N + i - j), \quad i = 0, 1, \cdots N - 1, \quad i < j \leqslant N \tag{13}$$

DOI: 10.1002/acs

With the upper triangular structure of (10), the finite trial length also imposes the additional condition

$$z^{-j} e_{k+1}(i) = 0, \quad 0 \leqslant j \leqslant i \tag{14}$$

arising from the fact that an ILC algorithm of the form (7) employs no current trial information in the formation of u_{k+1}. However, the typical structure of Q, as produced by the contraction mapping approaches of [10–13] for example, means that there exists a positive integer s such that $Q_i \approx 0$ for $s < i \leqslant N$, with $s \ll N$. Hence, the resetting action of (14) affects only the last $s - 1$ samples of each trial. Moreover, it can entirely be removed through application of a current-error feedback term, as described later in this section.

Using the initial conditions imposed on the memory variable (13), the z-transform representation of (11) can be rewritten as

$$\begin{aligned} u_{k+1}(i) &= \left(I_p - z^{-N} I_p\right)^{-1} z^{-N} \left(Q_1 + zQ_2 + \cdots + z^{N-1} Q_N\right) e_{k+1}(i), \quad (15)\\ &= \left(z^N I_p - I_p\right)^{-1} \left(Q_1 + zQ_2 + \cdots + z^{N-1} Q_N\right) e_{k+1}(i). \quad (16) \end{aligned}$$

If the effect of resetting the plant state at the beginning of each trial is temporarily ignored, the ILC system on the kth trial can be represented by the block diagram representation of Figure 1 in which $\Phi(z) := (z^N I_p - I_p)^{-1}$ and $L(z) = Q_1 + zQ_2 + \cdots + z^{N-1} Q_N$. Suitable selection of the reference ensures that the effect of plant resetting is small and tends to zero as the number of trials increases, as discussed later in this section.

The system $\Phi(z)$ is a $p \times p$ transfer-function matrix, and $L(z)$ is a $p \times m$ transfer-function matrix, and hence, their positions in the feedback system cannot be reversed if $p \neq m$. It is possible to re-derive a z-transform form of the ILC update (7) with the order of blocks $\Phi(z)$ and $L(z)$ reversed, with $\Phi(z)$ now assuming the form $\Phi(z) := (z^N I_m - I_m)^{-1}$. However, the initial conditions relating to the shift operator can no longer directly embed the previous input u_k, but instead embed the previous error, e_k. Hence, if $m < p$ it is not possible to stipulate the system input, and therefore, the ILC system (7) cannot be expressed using the z-transform variable. This conclusion holds for all choices of ILC algorithm, and means that $\Phi(z)$ always operates in the input space of the plant $P(z)$. This conclusion is also supported by results in [5, 6].

In the case of RC, neither the issue of resetting the plant dynamics nor explicitly stipulating the new control input exists. The reference signal $r(i)$ is periodic with period N, that is, $r(i+N) = r(i)$, and the generally assumed structure is shown in Figure 2. Here, the block $\Phi(z)$ implements the recursive relationship

$$v(i + N) = v(i) + e(i) \tag{17}$$

and $R(z)$ denotes an $p \times m$ discrete-time transfer-function matrix which acts upon the $m \times 1$ vector $v(i)$ to produce the plant input $u(i)$. Clearly in this case $\Phi(z)$ operates in the output space, and has the structure $\Phi(z) := (z^N I_m - I_m)^{-1}$. Further details relating the structure of ILC and RC to the forms shown in Figure 1 and Figure 2 respectively are given in [5, 6].

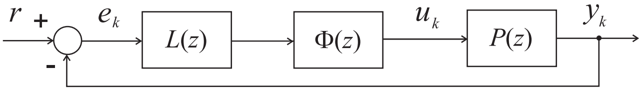

Figure 1. Iterative learning control as a feedback system.

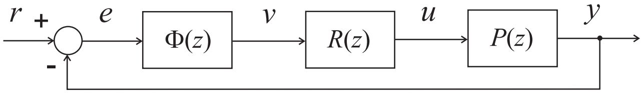

Figure 2. Repetitive control as a feedback system.

Although ILC assumes a fixed initial condition at the start of each new trial, RC has no such resetting and merely assumes that the initial condition at the start of each new period is the final condition at the end of the previous period. In particular, ILC is employed in applications performing pick and place operations whereas RC is applicable to a continuously rotating equipment with a constant periodic task.

The block $\Phi(z)$ in both Figures 1 and 2 is a diagonal transfer-function matrix with each diagonal element having the internal representation shown in Figure 3(a). If $\Phi(z)$ is regarded as a causal dynamic system, it will have N state variables in each channel, and in the RC case, there are m channels as $\Phi(z)$ operates in the output space and for ILC there are p channels as $\Phi(z)$ operates in the input space. The poles of $\Phi(z)$ are given by $z^N = 1$, that is, evenly distributed on the unit circle in the complex plane, and for stability in either case, these poles must be moved inside the unit circle by the action of the feedback loop acting around $\Phi(z)$.

Since ILC and RC have a similar structure, they can be analyzed using the same framework. Stability analysis for both cases can be performed following the method proposed in [3] by first isolating the delay chain, $z^{-N}I$, within the internal model, and then forming the transfer-function, $H(z)$, linking the output of the delay chain to its input. The system formed by the feedback connection of the delay chain and $H(z)$ is termed the 'equivalent system' [3], which is acted upon by the disturbance in order to accomplish the tracking task. Because the delay chain has magnitude equal to one, then by the small gain theorem, a sufficient condition for stability is that

$$\|H(z)\| < 1 \tag{18}$$

where $\|\cdot\|$ denotes the norm induced by the norm imposed on the vector space, although (18) is often not met in practice. Applying this method in the RC case produces the condition

$$\|I_m - P(z)R(z)\| < 1 \tag{19}$$

which depends only on $R(z)$ and $P(z)$ and is independent of N.

The same stability criterion applied to the ILC system (11) yields the sufficient condition for stability

$$\|I_p - L(z)P(z)\| < 1$$

The internal representation of $\Phi(z)$ shown in Figure 3(a) used in both the mentioned ILC and RC approaches is termed the 'past-error feedforward' structure because the current error enters a delay chain before being used to create the control input. An alternative to this form is that shown in Figure 3(b) where the delay chain is applied only to the control input. This is termed the 'current-error feedback' structure, with the corresponding transfer-function $\Phi(z) = (I - z^{-N}I)^{-1}$. Hence, in the ILC case, (16) becomes

$$u_{k+1}(i) = \left(I_p - z^{-N}I_p\right)^{-1} L(z)e_{k+1}(i) \tag{20}$$

which replaces the form of ILC structure previously considered (7) and can be used to implement updates of the 'lifted' form involving the current error e_{k+1}

$$u_{k+1} = u_k + Me_k + Ne_{k+1} \tag{21}$$

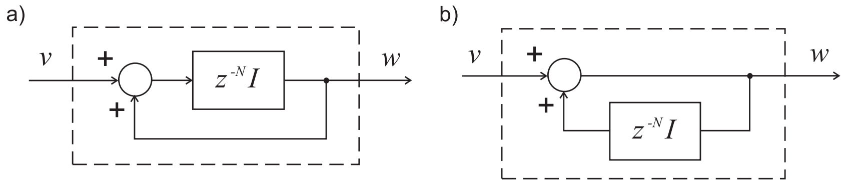

Figure 3. Representations of $\Phi(z)$ for (a) past-error feedforward, and (b) current-error feedback.

where M and N are upper and lower triangular Toeplitz matrices, respectively. Initial conditions on the delay operators are again used to embed previous trial information. For all implementations of ILC as a transfer-function system, the following comments are relevant:

- The structure of $\Phi(z)$ means that the initial conditions imposed on the memory elements on trial $k+1$ are simply the values of these elements at the end of trial k. Hence, no resetting or reinitialization is generally necessary—the recursive memory system is simply 'paused' during the plant resetting.
- Resetting the plant to initial states may cause learning transients during initial trials that are not present in the RC case, but are well-known within the ILC community [14]. However, these can be removed by ensuring that the reference starts and finishes at the same location, and that there is a sufficiently long period at the end of the trial when the reference is fixed (this can be achieved by appending to Reference (6) the vector $\left[r(N-1)^T \cdots r(N-1)^T\right]^T$). In particular, if the plant is controllable and is reset at the beginning of each trial to state $x(0) = 0$, as is commonly the case in practice, then there exists a trial length N and positive integer d such that the choice

$$r(0) = 0, \quad \text{and} \quad r(N-1-i) = 0, \quad i = 0, \dots d \tag{22}$$

guarantees that $x(N-1)$ converges to $x(0)$ through the action of ILC.

Isolating the delay chain in the new $\Phi(z)$ structure of Figure 3(b) and forming the transfer-function $H(z)$ produces the sufficient condition

$$\|(I_p - L(z)P(z))^{-1}\| < 1$$

for stability of the current-error feedback system. Similarly, if the current-error feedback structure is applied in RC, then the sufficient condition for stability is

$$\|(I_m - P(z)R(z))^{-1}\| < 1.$$

The internal model principle can be formulated in a structure where the disturbance model is realized in each channel of either the output or input spaces. The first case implements RC and the second ILC, in either current-error feedback or previous-error feedforward terms, and this property is termed duality in [6].

Both RC and ILC aim to improve system performance by updating the input recursively through identifying the most useful feedforward input signal that suppresses periodic disturbances. As in [6], the recursion of the repetitive or iterative learning part is represented as feedback. Moreover, convergence of this recursion translates into overall system stability. Hence, the feedback system can be interpreted as the design of a controller for disturbance suppression and an additional learning or repetitive component to suppress periodic disturbances where, in general, the learning or repetitive component will influence the sensitivity function of the complete loop and thereby modify the disturbance suppressing properties of the feedback controller.

This last fact means that fast convergence of the recursion may not be the best solution. A detailed discussion of this point is contained in [6], and it is argued that speed of convergence should be part of the overall feedback control design where the single objective is disturbance suppression. This, in turn, advocates use of the internal model principle to undertake joint design of a feedback controller and the repetitive or learning controller. It is also possible to extend this setting to the often encountered case when a control law has already been applied. As in [6], the remainder of this paper uses the internal model based setting that allows the joint design of a feedback controller and an RC or ILC law with desired properties relative to a single performance measure.

3. CONTROLLER DESIGN USING DUAL INTERNAL MODEL STRUCTURES

Any periodic signal can be generated by a system with no inputs formed by a time-delay in a positive feedback loop. In particular, a discrete-time periodic signal of length N can be generated by the state-space model

$$\begin{aligned} x_w(i+1) &= A_w x_w(i), \quad i=0,1,\cdots N-1, \quad x_w(0)=x_{w0} \\ w(i) &= C_w x_w(i) \end{aligned} \tag{23}$$

with suitable initial state x_{w0}. Here, the $N \times N$ matrix A_w is given by

$$A_w = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \end{bmatrix}$$

and the $1 \times N$ row vector C_w as

$$C_w = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

The poles of A_w are equally spaced on the unit circle in the complex plane. Now, consider the feedback system shown in Figure 4 where d is a periodic input disturbance and r is a periodic reference, each of the form (23). The robust RC problem is to construct a feedback controller $C(z)$ to ensure that the controlled system is exponentially stable and that the tracking error tends to zero asymptotically for all periodic disturbances. The solution is provided by the internal model principle [4], that is, the controller $C(z)$ solves the RC problem provided each of its channels contains a realization of the disturbance generating system, driven by the error e. Furthermore, $C(z)$ is designed such that the feedback connection of plant and controller is internally stable and the previous two properties are robust in the sense that they also hold in the case when the plant dynamics are perturbed. Both RC and ILC solve the periodic control problem and hence the internal model principle provides a solution for both.

For a single channel, $\Phi(z)$ can be represented by the state-space model

$$\begin{aligned} x_w(i+1) &= A_w x_w(i) + B_w v(i), \quad i=0,1,\cdots N-1, \quad x_w(0)=x_{w0} \\ w(i) &= C_w x_w(i) + D_w v(i) \end{aligned} \tag{24}$$

where A_w is given by (3), the $N \times 1$ column vector

$$B_w = \begin{bmatrix} 0 & \cdots & 0 & 0 & 1 \end{bmatrix}^T,$$

C_w by (3) and

$$D_w = \begin{cases} 1 & \text{current-error feedback} \\ 0 & \text{previous-error feedforward} \end{cases}$$

giving the two forms shown in Figure 3(a) and (b). For the multiple-input, multiple-output case introduce

$$\begin{aligned} A_r &= \operatorname{diag}\{A_w, A_w \ldots, A_w\}, \quad & B_r &= \operatorname{diag}\{B_w, B_w, \ldots, B_w\}, \\ C_r &= \operatorname{diag}\{C_w, C_w, \ldots, C_w\}, \quad & D_r &= \operatorname{diag}\{D_w, D_w, \ldots, D_w\} \end{aligned}$$

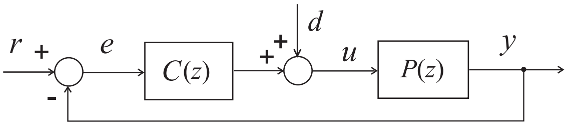

Figure 4. Periodic control problem in which r and d are periodic signals of the form (23).

DOI: 10.1002/acs

where each diagonal block matrix is repeated m times. In a similar way, define $\{A_l, B_l, C_l, D_l\}$ where each diagonal block matrix is repeated p times and the internal models $\Phi(z)$ are

$$C_r(zI_{Nm} - A_r)^{-1}B_r + D_r = \begin{cases} (z^N I_m - I_m)^{-1} & \text{if } D_w = 0 \\ (I_m - z^{-N} I_m)^{-1} & \text{if } D_w = 1 \end{cases}$$

and

$$C_l(zI_{Np} - A_l)^{-1}B_l + D_l = \begin{cases} (z^N I_p - I_p)^{-1} & \text{if } D_w = 0 \\ (I_p - z^{-N} I_p)^{-1} & \text{if } D_w = 1 \end{cases}$$

for RC and ILC, respectively, where in the latter, the initial conditions are realized simply by setting the initial states x_{w0} as the final value, $x_w(N-1)$, on the previous trial. The design of a controller for each case can now begin.

3.1. Iterative learning control design via state feedback

The position of the internal model for ILC is shown in Figure 1, and many designs for $L(z)$ have been proposed to solve the stabilization problem [1,2]. To illustrate the common design setting for RC and ILC considered in this paper, the design approach of [15–17] is now examined. A critical step in deriving the results that follow is the extension of the internal model structures in [6] as summarized in the previous section, to permit them to simultaneously address both current-error feedback and previous-error feedforward implementations.

First, write the series connection between $\Phi(z)$ and $P(z)$ in Figure 1 using the internal model (23) in the state-space form

$$\begin{aligned} \begin{bmatrix} x_{l,k}(i+1) \\ x_k(i+1) \end{bmatrix} &= \begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix} \begin{bmatrix} x_{l,k}(i) \\ x_k(i) \end{bmatrix} + \begin{bmatrix} B_l \\ BD_l \end{bmatrix} \tilde{u}_k(i), \quad i = 0, 1, \cdots N-1 \\ e_k(i) &= r(i) - [DC_l \ \ C] \begin{bmatrix} x_{l,k}(i) \\ x_k(i) \end{bmatrix} - DD_l \tilde{u}_k(i) \end{aligned} \tag{25}$$

where on the kth trial, $x_{l,k}$ is the state of the internal model system realizing $\Phi(z)$, and $\tilde{u}_k$ is its input. The internal model principle guarantees that regulating this system solves the tracking problem, and this is also clear since the input to (25) is the trial-to-trial control difference and its output is the error. To implement the control block $L(z)$, a state feedback scheme will be used, leading to the control input

$$\tilde{u}_k(i) = -K_l \begin{bmatrix} \hat{x}_{l,k}(i) \\ \hat{x}_k(i) \end{bmatrix} \tag{26}$$

To produce the state estimates $\hat{x}_{l,k}$ and $\hat{x}_k$, an observer with state-space vector L_l is required and is given by

$$\begin{aligned} \begin{bmatrix} \hat{x}_{l,k}(i+1) \\ \hat{x}_k(i+1) \end{bmatrix} &= \begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix} \begin{bmatrix} \hat{x}_{l,k}(i) \\ \hat{x}_k(i) \end{bmatrix} - \begin{bmatrix} B_l \\ BD_l \end{bmatrix} K_l \begin{bmatrix} \hat{x}_{l,k}(i) \\ \hat{x}_k(i) \end{bmatrix} \\ &\quad + L_l \left([DC_l \ \ C] - DD_l K_l\right) \left(\begin{bmatrix} \hat{x}_{l,k}(i) \\ \hat{x}_k(i) \end{bmatrix} - \begin{bmatrix} x_{l,k}(i) \\ x_k(i) \end{bmatrix} \right) \end{aligned}$$

The overall scheme is shown in Figure 5, and it remains to select the parameter K_l.

Stability analysis can be performed as in [3] by isolating the block $z^{-N} I_p$ within the internal model, and expressing the relationship between its output and input in terms of the transfer-function $H(z)$. A sufficient condition for stability is then given by (18) although this is often overly conservative. Figure 6 shows the system represented in transfer-function form for both previous-error feedforward, and current-error feedback cases, where

$$L(z) = -K_l \left(zI_{Np+n} - \begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix} + \begin{bmatrix} B_l \\ BD_l \end{bmatrix} K_l - L_l \left([DC_l \ \ C] - DD_l K_l\right) \right)^{-1} L_l. \tag{27}$$

DOI: 10.1002/acs

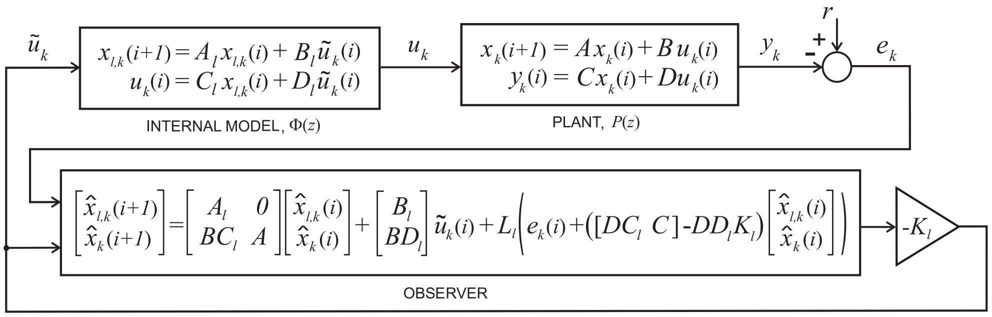
$\tilde{u}_k$
$x_{l,k}(i+1) = A_l x_{l,k}(i) + B_l \tilde{u}_k(i)$
$u_k(i) = C_l x_{l,k}(i) + D_l \tilde{u}_k(i)$
INTERNAL MODEL, $\Phi(z)$
u_k
$x_k(i+1) = Ax_k(i) + Bu_k(i)$
$y_k(i) = Cx_k(i) + Du_k(i)$
PLANT, $P(z)$
y_k
r
+
-
e_k
$\begin{bmatrix} \hat{x}_{l,k}(i+1) \\ \hat{x}_k(i+1) \end{bmatrix} = \begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix} \begin{bmatrix} \hat{x}_{l,k}(i) \\ \hat{x}_k(i) \end{bmatrix} + \begin{bmatrix} B_l \\ BD_l \end{bmatrix} \tilde{u}_k(i) + L_l\left(e_k(i) + ([DC_l \;\; C] - DD_l K_l)\begin{bmatrix} \hat{x}_{l,k}(i) \\ \hat{x}_k(i) \end{bmatrix}\right)$
$-K_l$
OBSERVER

Figure 5. Iterative learning control scheme.

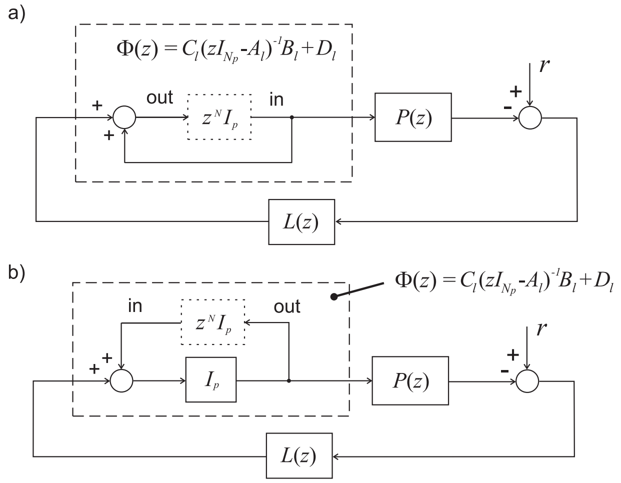
a)
$\Phi(z) = C_l(zI_{N_p} - A_l)^{-1}B_l + D_l$
+
+
out
$z^N I_p$
in
$P(z)$
+
-
r
$L(z)$
b)
$\Phi(z) = C_l(zI_{N_p} - A_l)^{-1}B_l + D_l$
in
$z^N I_p$
out
+
+
I_p
$P(z)$
+
-
r
$L(z)$

Figure 6. Internal model-based iterative learning control scheme for (a) previous-error feedforward, and (b) current-error feedback.

The transfer-function $H(z)$ connects the signals `in` and `out` in Figure 6 and for the case of previous-error feedforward is given by

$$H(z) = \frac{\texttt{out}}{\texttt{in}}(z) = (I_p - L(z)P(z)) = (G(z) + P(z))G(z)^{-1} \tag{28}$$

where

$$G(z) = [DC_l \;\; C]\left(zI_{Np+n} - \begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix} + \begin{bmatrix} B_l \\ BD_l \end{bmatrix} K_l\right)^{-1} \begin{bmatrix} B_l \\ BD_l \end{bmatrix} + DD_l \tag{29}$$

and for current-error feedback

$$H(z) = \frac{\texttt{out}}{\texttt{in}}(z) = (I_p + L(z)P(z))^{-1} = G(z)(G(z) - P(z))^{-1}. \tag{30}$$

Design of K_l is achieved by consideration of the equivalent system representation, which is the feedback connection of $H(z)$ and the $z^{-N}I_p$ block within the internal model [3]. It is straightforward to show that in both cases, the resulting feedback path dynamics are governed by the system

$$\begin{bmatrix} \tilde{x}_{l,k}(i+1) \\ \tilde{x}_k(i+1) \end{bmatrix} = \underbrace{\begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix}}_{\text{system matrix}} \begin{bmatrix} \tilde{x}_{l,k}(i) \\ \tilde{x}_k(i) \end{bmatrix} - \underbrace{\begin{bmatrix} B_l \\ BD_l \end{bmatrix}}_{\text{input matrix}} \overbrace{K_l}^{\text{state feedback}} \begin{bmatrix} \tilde{x}_{l,k}(i) \\ \tilde{x}_k(i) \end{bmatrix} \tag{31}$$

DOI: 10.1002/acs

which operates in the presence of the periodic disturbance to achieve the tracking task. The LQR approach is selected to calculate a suitable K_l gain to regulate the states and thereby drive the error to zero, with cost function

$$J = \sum_{i=0}^{\infty} \left\{ \eta(i)^T \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \eta(i) + \mu(i)^T R \mu(i) \right\} \tag{32}$$

where Q_1 and Q_2 are symmetric positive definite weighting matrices on the states and R is a symmetric positive definite weighting matrix on the control effort, and the system (31) has been rewritten in the form

$$\begin{aligned} \eta(i+1) &= \begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix} \eta(i) - \begin{bmatrix} B_l \\ BD_l \end{bmatrix} \mu(i) \\ \mu(i) &= K_l \eta(i). \end{aligned} \tag{33}$$

Solving (32) yields a solution for K_l via the corresponding Ricatti equation. Note that this is solved once before the experimental tests begin, and the separation principle permits independent design of K_l and L_l to result in a stable system. As stated in Section 2, in the ILC framework, the system shown in Figure 5 is run for N samples starting with the plant reset to its initial states, and the initial states of the internal model equal to their final values in the previous trial.

The use of state feedback to provide stabilization differs from the output injection structure previously associated with ILC in the duality framework [6]. This difference is due to the presence of an additional internal model within the state estimator system (27).

3.2. Repetitive control design via state feedback

Using the common framework described in Section 2, an RC counterpart to the previous ILC design is now derived. The implementation of an internal model within the feedback loop enables exactly the same state feedback structure to be assumed, hence simplifying controller development.

Define the series connection of $P(z)$ and $\Phi(z)$ shown in Figure 2 using the internal model (23) with state x_r to obtain

$$\begin{bmatrix} x_r(i+1) \\ x(i+1) \end{bmatrix} = \begin{bmatrix} A_r & -B_r C \\ 0 & A \end{bmatrix} \begin{bmatrix} x_r(i) \\ x(i) \end{bmatrix} + \begin{bmatrix} -B_r D \\ B \end{bmatrix} u(i) + \begin{bmatrix} B_r \\ 0 \end{bmatrix} r(i) \tag{34}$$

The internal model principle again guarantees that regulation of the system achieves the tracking task, and hence the same approach as in Section 3.1 will be applied. Accordingly, the controller $R(z)$ is implemented using state feedback, leading to the control input

$$u(i) = -K_r \begin{bmatrix} \hat{x}_r(i) \\ \hat{x}(i) \end{bmatrix} \tag{35}$$

For this structure, an observer with state-space vector L_l produces the estimated states

$$\begin{aligned} \begin{bmatrix} \hat{x}_r(i+1) \\ \hat{x}(i+1) \end{bmatrix} = {} & \begin{bmatrix} A_r & -B_r C \\ 0 & A \end{bmatrix} \begin{bmatrix} \hat{x}_r(i) \\ \hat{x}(i) \end{bmatrix} - \begin{bmatrix} -B_r D \\ B \end{bmatrix} K_r \begin{bmatrix} \hat{x}_r(i) \\ \hat{x}(i) \end{bmatrix} \\ & + L_r \left(v(i) - \left(\begin{bmatrix} C_r & -D_r C \end{bmatrix} + D_r D K_r \right) \begin{bmatrix} \hat{x}_r(i) \\ \hat{x}(i) \end{bmatrix} \right) \end{aligned}$$

and the overall scheme is shown in Figure 7.

Figure 8 shows the system represented in transfer-function form for both previous-error feedforward, and current-error feedback cases. As in the ILC case, analysis focuses on the feedback path about $z^{-N} I_m$, which is defined by the transfer-function, $H(z)$, linking the signals `in` and `out` appearing in Figure 8. where

$$\begin{aligned} R(z) = -K_r \Bigg(& z I_{Nm+n} - \begin{bmatrix} A_r & -B_r C \\ 0 & A \end{bmatrix} + \begin{bmatrix} B_r D \\ -B \end{bmatrix} K_r \\ & + L_r \left(\begin{bmatrix} C_r & -D_r C \end{bmatrix} + D_r D K_r \right) \Bigg)^{-1} L_r \end{aligned} \tag{36}$$

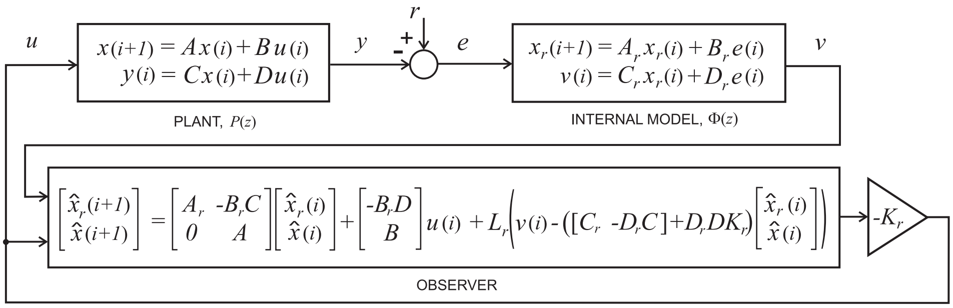
u
$x(i+1) = Ax(i) + Bu(i)$
$y(i) = Cx(i) + Du(i)$
PLANT, $P(z)$
y
r
$+$
$-$
e
$x_r(i+1) = A_r x_r(i) + B_r e(i)$
$v(i) = C_r x_r(i) + D_r e(i)$
INTERNAL MODEL, $\Phi(z)$
v
$\begin{bmatrix} \hat{x}_r(i+1) \\ \hat{x}(i+1) \end{bmatrix} = \begin{bmatrix} A_r & -B_rC \\ 0 & A \end{bmatrix}\begin{bmatrix} \hat{x}_r(i) \\ \hat{x}(i) \end{bmatrix} + \begin{bmatrix} -B_rD \\ B \end{bmatrix} u(i) + L_r\left(v(i) - ([C_r \;\; -D_rC] + D_rDK_r)\begin{bmatrix} \hat{x}_r(i) \\ \hat{x}(i) \end{bmatrix}\right)$
OBSERVER
$-K_r$

Figure 7. Synthesized repetitive control scheme.

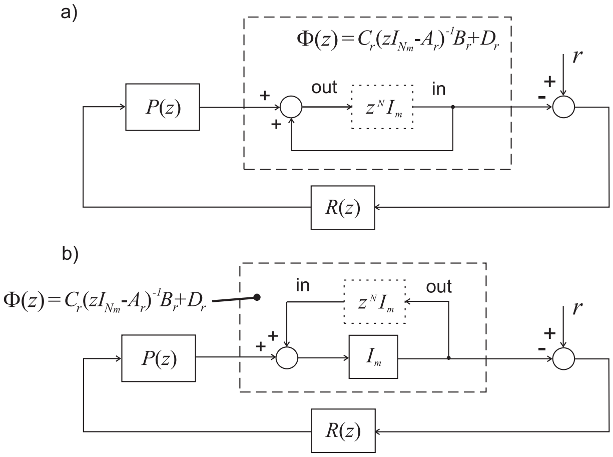

Figure 8. Internal model-based repetitive control scheme for (a) previous-error feedforward and (b) current-error feedback.

For the previous-error feedforward and current-error feedback cases, the relationships in (28) and (30), respectively, are again obtained, where

$$G(z) = \begin{bmatrix} C_r & -D_rC \end{bmatrix}\left(zI - \begin{bmatrix} A_r & -B_rC \\ 0 & A \end{bmatrix} + \begin{bmatrix} B_rD \\ -B \end{bmatrix} K_r\right)^{-1} \begin{bmatrix} -B_rD \\ B \end{bmatrix} - D_rD$$

These provide sufficient conditions for stability when combined with (28). The equivalent system representation is again used to design a suitable K_r, and encompasses the feedback connection of $H(z)$ and the $z^{-N}I_m$ block within the internal model [3]. The equivalent system has the feedback path dynamics given by

$$\begin{bmatrix} \tilde{x}_r(i+1) \\ \tilde{x}(i+1) \end{bmatrix} = \underbrace{\begin{bmatrix} A_r & -B_rC \\ 0 & A \end{bmatrix}}_{\text{system matrix}} \begin{bmatrix} \tilde{x}_r(i) \\ \tilde{x}(i) \end{bmatrix} - \underbrace{\begin{bmatrix} -B_rD \\ B \end{bmatrix}}_{\text{input matrix}} \overbrace{K_r}^{\text{state feedback}} \begin{bmatrix} \tilde{x}_r(i) \\ \tilde{x}(i) \end{bmatrix} \tag{37}$$

To design the state feedback matrix K_r, the LQR approach is applied to (37) and this leads to minimization of (32) with

$$\begin{aligned} \eta(i+1) &= \begin{bmatrix} A_r & -B_rC \\ 0 & A \end{bmatrix} \eta(i) - \begin{bmatrix} -B_rD \\ B \end{bmatrix} \mu(i) \\ \mu(i) &= K_r\eta(i) \end{aligned} \tag{38}$$

Hence, an RC scheme has been designed using the same LQR cost function and state feedback structure as in the ILC case.

DOI: 10.1002/acs

4. CONTROLLER DESIGN USING SINGLE INTERNAL MODEL STRUCTURES

Having illustrated the utility of the common framework underlying ILC and RC for controller design, another case study is now examined. This is motivated by the observation that the controllers in Section 3 contained two internal models: one comprising $\Phi(z)$ and one within the estimator and hence embedded within $R(z)$ or $L(z)$. This duplication allowed a simplification in overall control structure, but may lead to large control inputs because of the necessity that both internal models be simultaneously stabilized. The development of schemes based on a single internal model is therefore undertaken in this section, and enables experimental comparison of structures to be carried out in Section 5.

4.1. Repetitive control design via state feedback

Results in [5, 6] consider the case in which a single internal model appears in the control scheme, and it is shown that this restricts the structure associated with each case: RC must use state feedback, and ILC must employ output injection. In this section, the starting point is the RC scheme given in [6], and from this, an ILC scheme based on the common framework is developed, which includes the option of current-error feedback or previous-error feedforward. Note that the RC scheme of [6] is more restrictive and includes an explicit current-error feedback path that cannot be broken.

Modifying the RC scheme of Figure 2 to include an explicit current-error feedback path produces the system shown in Figure 9. In [6], the current-error feedback was realized through feedback of the plant states. Furthermore, the term C_r in the internal model representation was used to implement the filter $R(z)$. The resulting control input is given by

$$u(i) = K_r x_r(i) + K\hat{x}(i) \tag{39}$$

with state observer

$$\hat{x}(i+1) = A\hat{x}(i) + Bu(i) + L_r(e(i) + C\hat{x}(i) + Du(i)) \tag{40}$$

where L_r is the state observer matrix and K_r and K are the state feedback matrices for the internal model and the plant, respectively. The overall structure is shown in Figure 10.

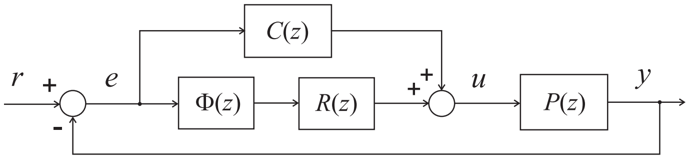

Figure 9. Repetitive controller with existing feedback controller $C(z)$.

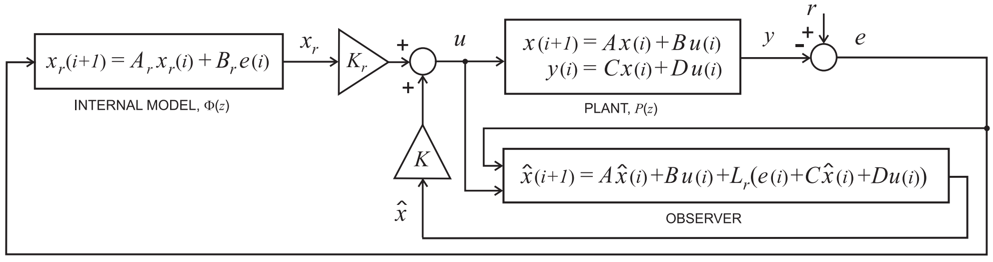

Figure 10. Repetitive controller with current-error feedback.

DOI: 10.1002/acs

This corresponds to that of Figure 9, with $R(z) = K_r(zI_{Nm} - A_r)^{-1}B_r(I_m - z^{-N}I_m)(I_p + K(zI_n - \bar{A})^{-1}(B + L_rD))$, $C(z) = K(zI_n - \bar{A})^{-1}L_r$ where $\bar{A} = A + BK + L_r(C + DK)$. The feedback connection of $H(z)$ and $z^{-N}I_m$ is shown in [6] to have the equivalent realization

$$\begin{bmatrix} x_r(i+1) \\ \tilde{x}(i+1) \end{bmatrix} = \underbrace{\begin{bmatrix} A_r & -B_rC \\ 0 & A \end{bmatrix}}_{\text{system matrix}} \begin{bmatrix} x_r(i) \\ \tilde{x}(i) \end{bmatrix} + \underbrace{\begin{bmatrix} -B_rD \\ B \end{bmatrix}}_{\text{input matrix}} \overbrace{\begin{bmatrix} K_r & K \end{bmatrix}}^{\text{state feedback}} \begin{bmatrix} x_r(i) \\ \tilde{x}(i) \end{bmatrix} \tag{41}$$

As with the ILC system of Section 3, state feedback is used to provide stabilization, and hence solve the tracking problem. Accordingly $[K_r\ K]$ is determined by minimizing (32) with

$$\begin{aligned} \eta(i+1) &= \begin{bmatrix} A_r & -B_rC \\ 0 & A \end{bmatrix} \eta(i) + \begin{bmatrix} -B_rD \\ B \end{bmatrix} \mu(i) \\ \mu(i) &= \begin{bmatrix} K_r & K \end{bmatrix} \eta(i). \end{aligned}$$

The above equivalent system and LQR structure is identical to those derived in Section 3.2, revealing the same underlying dynamics [3]. However the asymptotic tracking performance is determined by how the periodic disturbance interacts with this system, which differs for the structures that have been considered because the disturbance enters at different points. Hence, the role of multiple internal models on the subsequent performance can be transparently examined in Section 5 through stipulation of the same symmetric positive definite weighting matrices Q_1, Q_2 and R in (32).

4.2. Iterative learning control design via output injection

A disturbance observer structure will now be developed for the ILC case in order to compare the two implementations (state feedback versus output injection). This differs from the ILC implementation considered in [6] because no explicit current-error feedback is included, the more general option of either current-error feedback or previous-error feedforward instead incorporated into the internal model. Moreover, the resulting structure is considerably simpler, and analysis is conducted to place the solution of the problem within the same LQR framework as previously considered. The two implementations of ILC, however, do have the same equivalent structure.

To develop an ILC scheme with a single internal model, the starting point is again the series connection of plant and internal model given by (34). Because it is known that a disturbance observer structure is necessary for stabilization [5], an estimator is designed for an additive periodic disturbance assumed to exist at the plant input, and the estimator disturbance signal is then applied at the same point in order to cancel it. This differs from the observer implementations of Section 3, which estimated the system states. As the disturbance is removed from the system, the additive input reduces to zero, and the plant output converges to the demanded trajectory. This results in the output injection structure shown in Figure 11. The system is given by

$$\begin{bmatrix} x_{l,k}(i+1) \\ x_k(i+1) \end{bmatrix} = \begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix} \begin{bmatrix} x_{l,k}(i) \\ x_k(i) \end{bmatrix} + \begin{bmatrix} L_l \\ BD_l \end{bmatrix} \tilde{u}_k(i) \tag{42}$$

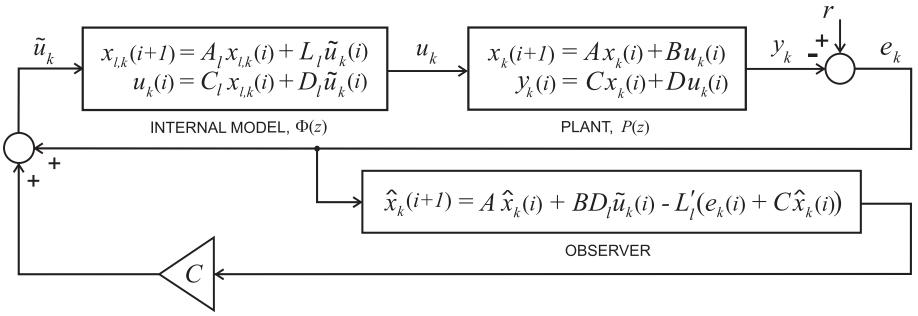
$\tilde{u}_k$
$x_{l,k}(i+1) = A_l x_{l,k}(i) + L_l\tilde{u}_k(i)$
$u_k(i) = C_l x_{l,k}(i) + D_l\tilde{u}_k(i)$
INTERNAL MODEL, $\Phi(z)$
u_k
$x_k(i+1) = Ax_k(i) + Bu_k(i)$
$y_k(i) = Cx_k(i) + Du_k(i)$
PLANT, $P(z)$
y_k
r
+
-
e_k
+
+
$\hat{x}_k(i+1) = A\hat{x}_k(i) + BD_l\tilde{u}_k(i) - L_l'(e_k(i) + C\hat{x}_k(i))$
OBSERVER
C

Figure 11. Synthesized iterative learning control scheme.

DOI: 10.1002/acs

with the observer

$$\hat{x}_k(i+1) = A\hat{x}_k(i) + BD_l\tilde{u}_k(i) - L_l'(e_k(i) + C\hat{x}_k(i)) \tag{43}$$

which produces an output $C\hat{x}_k(i)$ that is used as a correction term for the internal model. This system is shown in Figure 12 in transfer-function form for both previous-error feedforward and current-error feedback cases.

Stability analysis again considers the feedback path about $z^{-N}I_p$, which is the transfer-function matrix, $H(z)$, linking signals `in` and `out` in Figure 12, which shows both internal model structures. Here

$$L(z) = C\left(zI_n - A - BD_lC + L_l'C\right)^{-1}(BD_l - L_l') + I_m \tag{44}$$

and $L'(z)$ is such that

$$C_l(zI_{Np} - A_l)^{-1}B_l + D_l = (I_p - z^{-N}I_p)^{-1}L'(z)$$

yielding

$$L'(z) = C_lB_lz^{-1} + C_lA_lB_lz^{-2} + \cdots + C_lA_l^{-N-1}B_lz^{-N}. \tag{45}$$

For the case of previous-error feedforward

$$H(z) = \frac{\texttt{out}}{\texttt{in}}(z) = G(z)(G(z) - L'(z))^{-1} \tag{46}$$

where

$$G(z) = -[DC_l \;\; C]\left(zI_{Np+n} - \begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix} + \begin{bmatrix} L_l \\ L_l' \end{bmatrix}[DC_l \;\; C]\right)^{-1}[C_l \;\; 0]^T + D_l \tag{47}$$

and for current-error feedback

$$H(z) = \frac{\texttt{out}}{\texttt{in}}(z) = (G(z) - I_m)(G(z) - I_m - L'(z))^{-1} \tag{48}$$

A sufficient condition for stability is provided in each case by (18), and the feedback connection of $H(z)$ and $z^{-N}I_m$ can be shown to have equivalent realization

$$\begin{bmatrix} \tilde{x}_{l,k}(i+1) \\ \tilde{x}_k(i+1) \end{bmatrix} = \underbrace{\begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix}}_{\text{system matrix}}\begin{bmatrix} \tilde{x}_{l,k}(i) \\ \tilde{x}_k(i) \end{bmatrix} - \underbrace{\begin{bmatrix} L_l \\ L_l' \end{bmatrix}}_{\text{observer gain}}\overbrace{[DC_l \;\; C]}^{\text{output matrix}}\begin{bmatrix} \tilde{x}_{l,k}(i) \\ \tilde{x}_k(i) \end{bmatrix} \tag{49}$$

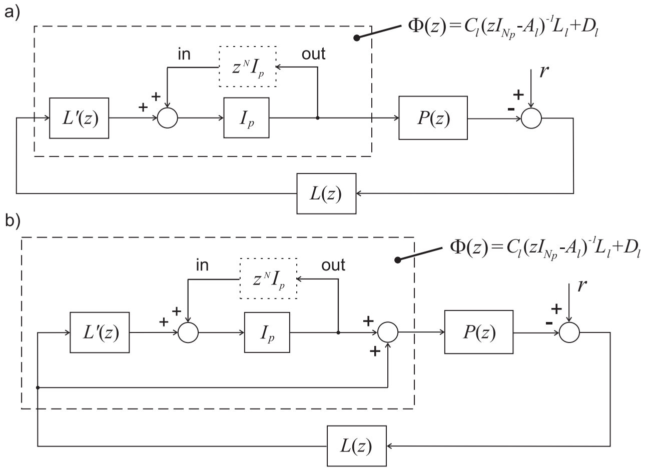
a)
$\Phi(z)=C_l(zI_{N_p}\text{-}A_l)^{-1}L_l+D_l$
in
z^NI_p
out
+
+
$L'(z)$
I_p
$P(z)$
r
+
-
$L(z)$
b)
$\Phi(z)=C_l(zI_{N_p}\text{-}A_l)^{-1}L_l+D_l$
in
z^NI_p
out
+
+
$L'(z)$
I_p
+
+
$P(z)$
r
+
-
$L(z)$

Figure 12. Internal model-based iterative learning control scheme for (a) previous-error feedforward and (b) current-error feedback.

DOI: 10.1002/acs

where $\tilde{x}_l = x_l$ is the internal model state, and $\tilde{x} = x - \hat{x}$ is the difference between plant state and disturbance state, which then feeds into the internal model. The system (49) confirms the output injection structure reported in [5, 6] when only a single internal model is used. This structure is the dual of (31), which used state feedback. A solution to the design of the observer gain is found by considering the dual of a general state-space model, as discussed in, for example, [18].

Consider first

$$\begin{aligned} x(i+1) &= Ax(i) + Bu(i) \\ y(i) &= Cx(i) + Du(i) \end{aligned} \tag{50}$$

and take the dual to produce

$$\begin{aligned} x^*(i+1) &= A^T x^*(i) + C^T u(i) \\ y(i) &= B^T x^*(i) + D^T u(i) \end{aligned} \tag{51}$$

Now, apply exponentially stabilizing state feedback of the form $u = Fx^*$, thereby giving rise to the system

$$x^*(i+1) = \left(A^T + C^T F\right) x^*(i) \tag{52}$$

and taking the dual of this last system gives

$$x(i+1) = \left(A^T + C^T F\right)^T x(i) = \left(A + F^T C\right) x(i) = Ax(i) + F^T Cx(i) \tag{53}$$

whose stabilization is also guaranteed as the system is exponentially stable if and only if its dual is stable [18]. Therefore, design of state feedback for the system (51) provides the output feedback vector for the system (53). Accordingly, comparing (53) with (49), the system (51) is

$$\begin{aligned} \begin{bmatrix} x^*_{l,k}(i+1) \\ x^*_k(i+1) \end{bmatrix} &= \begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix}^T \begin{bmatrix} x^*_{l,k}(i) \\ x^*_k(i) \end{bmatrix} + [DC_l \;\; C]^T u_k(i) \\ &= \begin{bmatrix} A_l^T & C_l^T B^T \\ 0 & A^T \end{bmatrix} \begin{bmatrix} x^*_{l,k}(i) \\ x^*_k(i) \end{bmatrix} + \begin{bmatrix} C_l^T D^T \\ C^T \end{bmatrix} \begin{bmatrix} L_l^T \;\; L_l'^T \end{bmatrix} \begin{bmatrix} x^*_{l,k}(i) \\ x^*_k(i) \end{bmatrix} \end{aligned}$$

LQR theory will again be applied to calculate $\left[L_l^T \; L_l'^T\right]$ in order to regulate this system, which requires $\left[L_l^T \; L_l'^T\right]$ such that (32) is minimized, with

$$\begin{aligned} \eta(i+1) &= \begin{bmatrix} A_l^T & C_l^T B^T \\ 0 & A^T \end{bmatrix} \eta(i) + \begin{bmatrix} C_l^T D^T \\ C^T \end{bmatrix} \mu(i) \\ \mu(i) &= \begin{bmatrix} L_l^T \; L_l'^T \end{bmatrix} \eta(i). \end{aligned} \tag{54}$$

This controller completes the set of RC and ILC schemes incorporating one and two internal models respectively, and in common with those in Section 3, can be configured to use either current-error feedback or previous-error feedforward to achieve the tracking task. Sections 3 and 4 have each illustrated how the common underlying structure of Section 2 provides a straightforward route for novel controller derivation. All four control schemes use the same LQR cost function (32), and hence enable transparent comparison of performance to be undertaken experimentally in the next section.

5. EXPERIMENTAL RESULTS USING GANTRY ROBOT

The developed RC and ILC approaches have been experimentally tested using a three-axis gantry robot test facility shown in Figure 13 (see [19] for full details of design and operation). The combined displacement reference trajectories for each axis produce a ‘pick and place’ action, designed to collect payloads and place them on a moving conveyor. The overall movement is shown in Figure 14. The time taken for each cycle is fixed at 2 s, and a sample frequency of 100 Hz is used,

DOI: 10.1002/acs

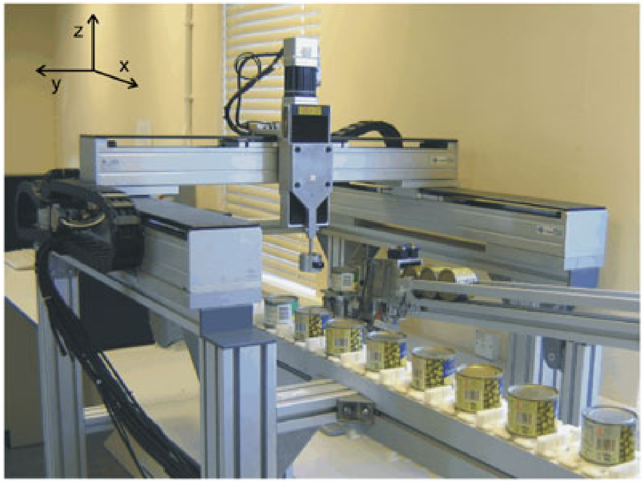

Figure 13. Gantry robot test facility.

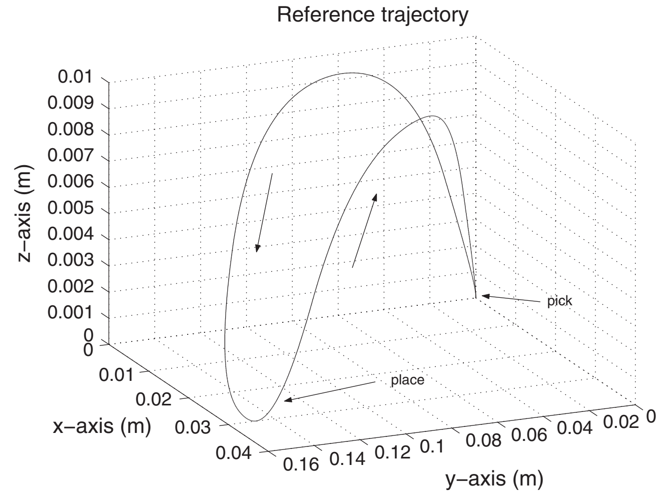

Figure 14. Three-dimensional representation of reference trajectories.

giving $N = 200$. Owing to their similarity, results for a single axis alone are given. The associated discrete-time transfer-function model is

$$P(z) = \frac{0.00029625(z^2 - 0.1162z + 0.1911)(z^2 - 0.9333z + 0.9225)(z^2 + 0.0619z + 0.8731)}{(z-1)(z^2 + 0.1141z + 0.2479)(z^2 - 1.247z + 0.7866)(z^2 - 0.2767z + 0.8075)}$$

For each of the controllers derived in this paper, 200 trials of the tracking task have been performed, where in the RC case, these are looped with no resetting between trials, and in the ILC case resetting of the plant occurs between trials. Figures 15 and 16, respectively, show the mean square error (MSE) results for the ILC scheme developed in Section 3.1 in both current-error feedback and previous-error feedforward implementations. A variety of weighting in the cost function (32) with (33) have been used to find a compromise between convergence and stability. The error MSE is reduced by several orders of magnitude, resulting in extremely accurate tracking. Figures 17 and 18 show MSE results for the RC scheme developed in Section 3.2 in both current-error feedback and previous-error feedforward implementations. A range of weighting parameters for the cost (32)

DOI: 10.1002/acs

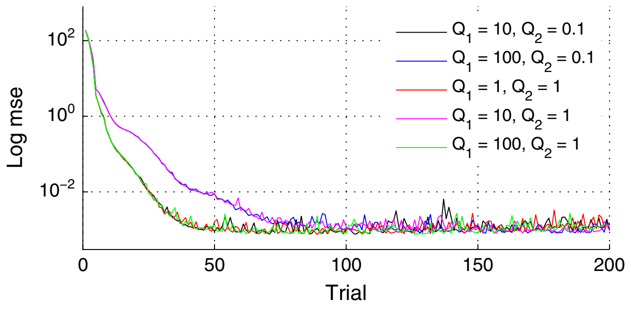

Figure 15. Mean square error plot for current-error feedback iterative learning control scheme (two internal models).

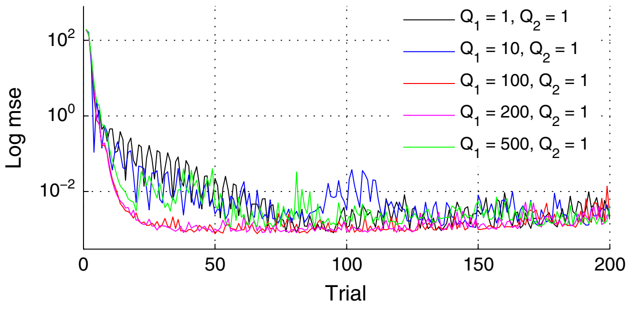

Figure 16. Mean square error plot for previous-error feedforward iterative learning control scheme (2 internal models).

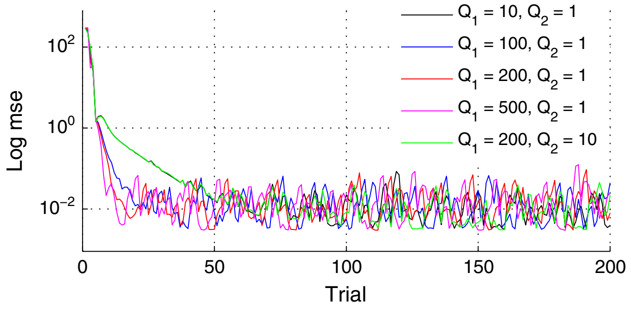

Figure 17. Mean square error plot for current-error feedback repetitive control scheme (two internal models).

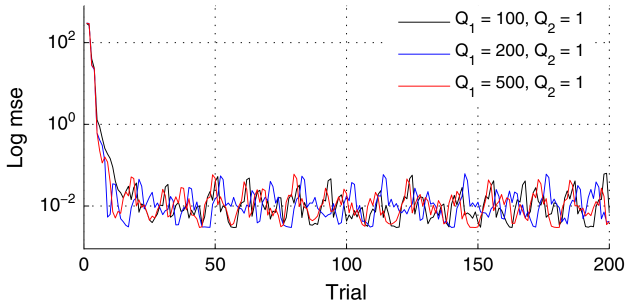

Figure 18. Mean square error plot for previous-error feedforward repetitive control scheme (two internal models).

DOI: 10.1002/acs

with (38) have been used. Similar error levels have been obtained for each case, although there is evidence of increased error fluctuation for the current-error feedback implementation, and slightly slower convergence. RC does not quite deliver as high tracking accuracy as ILC because of the lack of resetting.

Figure 19 shows results from the single internal model scheme of Section 4.1, which has an explicit error feedback loop. Figures 20 and 21, respectively, show MSE results for the ILC output injection scheme developed in Section 4.2, which also has a single internal model. There is evidence that the previous-error feedforward structure provides faster convergence at the expense of increased transients compared with the current-error feedback structure. From the cost functions used in the controller derivation for all schemes, it is clear that the use of two internal models requires significantly greater weighting values, which shows that more emphasis is required on the tracking objective as opposed to reducing the input amplitude. Hence, more control effort is needed to stabilize two internal models rather than a single internal model.

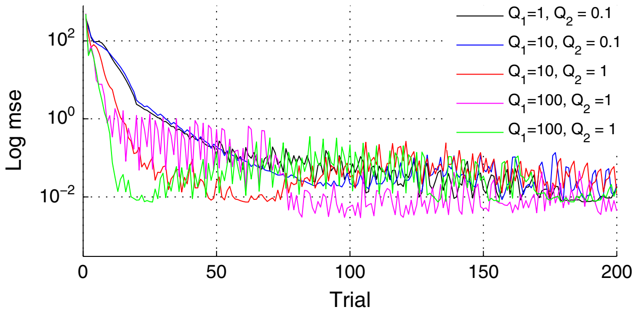

Figure 19. Mean square error plot for explicit current-error feedback repetitive control scheme (one internal model).

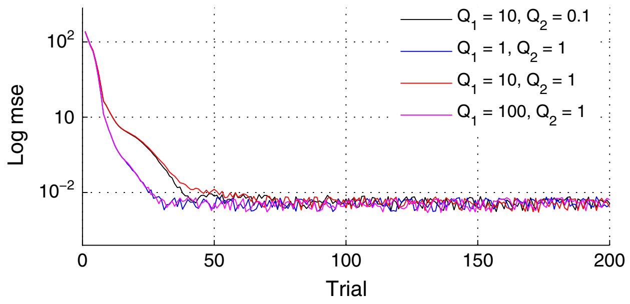

Figure 20. Mean square error plot for current-error feedback iterative learning control scheme (one internal model).

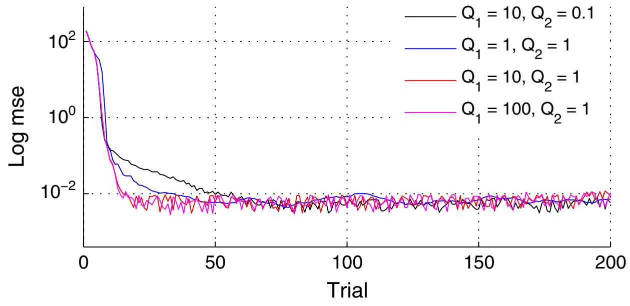

Figure 21. Mean square error plot for previous-error feedforward iterative learning control scheme (one internal model).

The results show the general trends: (i) ILC attains lower error because of the resetting action between trials; (ii) current-error feedforward leads to slightly reduced error transients at the expense of convergence speed; (iii) two internal models lead to slightly reduced error; and (iv) two internal models lead to increased control effort. All controllers provide highly accurate tracking of the reference trajectories.

6. CONCLUSIONS

The common framework connecting RC and ILC has been shown to allow derivation of controllers in one setting on the basis of an existing control structure in the other. This provides significant utility because schemes that work well in one domain may be used to derive similar controllers in the other. This paper addresses the lack of performance comparison in the literature to investigate the feasibility of such an approach, or the effect of structural considerations such as the number of internal models, form of stabilization, or error feedback/feedforward implementation.

This paper firstly illustrated the power of the common framework for synthesis of controllers between domains, introducing two case studies: first, an RC scheme incorporating two internal models on the basis of an existing RC scheme, and second, an ILC scheme incorporating a single internal model on the basis of an existing RC scheme. An industrially relevant robotic test facility was used to provide quantitative comparison of all the key design issues, as well as verification of the high levels of performance possible. Hence, the exploitation of a common framework for controller development has been confirmed and investigated, yielding results and conclusions that are relevant to both domains.

REFERENCES

1. Bristow DA, Tharayil M, Alleyne AG. A survey of iterative learning control a learning-based method for high-performance tracking control. *IEEE Control Systems Magazine* 2006; **26**(3):96–114.
2. Ahn HS, Chen Y, Moore KL. Iterative learning control: brief survey and categorization. *IEEE Transactions on Systems, Man and Cybernetics–Part C: Applications and Reviews* 2007; **37**(6):1099–1121.
3. Hara S, Yamamoto Y, Omata T, Nakano M. Repetitive control system: a new type servo system for periodic exogenous signals. *IEEE Transactions on Automatic Control* 1988; **AC-33**:659–668.
4. Francis BA, Wonham WM. The internal model principle for linear multivariable regulators. *Journal of Applied Mathematics and Optimization* 1975; **2**:170–194.
5. de Roover D, Bosgra OH. Dualization of the internal model principle in compensator and observer theory with application to repetitive and learning control. *Proceedings of the 1997 American Control Conference*, Albuquerque, New Mexico, 1997; 3902–3906.
6. de Roover D, Bosgra OH, Steinbuch M. Internal-model-based design of repetitive and iterative learning controllers for linear multivariable systems. *International Journal of Control* 2000; **73**(10):914–929.
7. Freeman CT, Alsubaie M, Cai Z, Rogers E, Lewin PL. Model and experience-based initial input construction for iterative learning control. *International Journal of Adaptive Control and Signal Processing* 2011; **25**(5):430–447.
8. van de Wijdeven J, Bosgra OH. Residual vibration suppression using Hankel iterative learning. *International Journal of Robust and Nonlinear Control* 2008; **18**(10):1034–1051.
9. Hätönen JJ, Owens DH, Moore KL. An algebraic approach to iterative learning control. *International Journal of Control* 2004; **77**(1):45–54.
10. Owens DH, Hätönen JJ, Daley S. Robust monotone gradient-based discrete-time iterative learning control. *International Journal of Robust and Nonlinear Control* 2009; **19**:634–661.
11. Kinosita K, Sogo T, Adachi N. Iterative learning control using adjoint systems and stable inversion. *Asian Journal of Control* 2002; **4**(1):60–67.
12. Furuta K, Yamakita M. The design of a learning control system for multivariable systems. *Proceedings of the IEEE International Symposium on Intelligent Control*, Philadelphia, PA, 1987; 371–376.
13. Freeman CT, Lewin PL, Rogers E. Further results on the experimental evaluation of iterative learning control algorithms for non-minimum phase plants. *International Journal of Control* 2007; **80**(4):569–582.
14. Longman RW. Iterative learning control and repetitive control for engineering practice. *International Journal of Control* 2000; **73**:930–954.
15. Hätönen JJ, Owens DH, Ylinen R. A new optimality based repetitive control algorithm for discrete-time systems. *Proceedings of the European Control Conference (ECC'03)*, Cambridge UK, 2003. CD–Rom.
16. Freeman CT, Hätönen JJ, Lewin PL, Rogers E, Owens DH. Experimental evaluation of a new repetitive control algorithm on a non-minimum phase spring-mass-damper system. *Proceedings of the IFAC Workshop on Adaption and*

DOI: 10.1002/acs

Learning in Control and Signal Processing and the IFAC Workshop on Periodic Control Systems, Yokohama, Japan, 2004; 681–686.

17. Freeman CT, Lewin PL, Rogers E, Owens DH, Hätönen JJ. An optimality based repetitive control algorithm for discrete-time systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 2008; **55**(1):412–423.
18. Kwakeraak H, Sivan R. *Linear Optimal Control Systems*. John Wiley and Sons, Inc.: New York, 1972.
19. Ratcliffe JD, Lewin PL, Rogers E, Hätönen JJ, Owens DH. Norm-optimal iterative learning control applied to gantry robots for automation applications. *IEEE Transactions On Robotics* 2006; **22**(6):1303–1307.

DOI: 10.1002/acs