IEEE Conference on Decision and Control, Phoenix, AZ · December 1999

Distributed System Loopshaping Design of Iterative Control for Batch Processes

Dimitry Gorinevsky

Honeywell Technology Center, CA35-5272H
One Results Way, Cupertino, CA 95014
Gorinevsky_Dimitry@HTC.Honeywell.com

Abstract

This paper presents a systematic approach to analysis and design of Iterative Learning Control (ILC). Rigorous ILC design method is developed based on detailed specifications for performance, robustness, controller non-fragility, and control amplitude constraints. The designed ILC update is implementable through non-causal convolution window operators. The controller is designed by solving a linear-quadratic optimization problem and manipulating control penalty weights. This is related in spirit to LQG/LTR design method for dynamic controllers.

1 Introduction

The idea of iterative control of batch processes has been discussed in the automatic control literature for some time. Many related ideas have been published under the names of Iterative Control, Learning Control, Run-to-run Control, and others. In this paper, the name Iterative Learning Control (ILC) will be used. An ILC controller provides feedforward control input to a plant through a batch run. The feedforward control history for an entire batch run is stored in a memory buffer. The plant output in the batch is stored in another memory buffer. The ILC update is computed between the batches and based on the output history buffer data to calculate an update to the feedforward control history buffer data. Only main relevant ideas of the prior work are briefly overviewed below. A detailed bibliography on ILC can be found in [6, 14].

Most of the ILC approaches considered in the literature update the feedforward control as

$$U^{(k)} = U^{(k-1)} - AY^{(k)}, \tag{1}$$

where the superscript k denotes the batch run number, $U^{(k)}$ is a vector containing the sampled feedforward control values through the batch, $Y^{(k)}$ is a vector containing sampled plant output values, and A is a linear update operator.

Initial and best known approaches to the ILC problem used an ILC feedback update operator A of the form similar to standard dynamical controllers, such as P or PD operators acting on the plant output sequence as a local-time function, e.g., see [2]. More complicated feedback operators described by high-order rational transfer functions can be used, e.g., see [15]. The convergence analysis for such operators is conveniently and intuitively performed in the frequency domain such as in [14] and other work.

The main drawback of the described approaches is that they follow standard controller methods and use feedback operators that are causal in local time . At the same time, in the ILC problem the feedback operator does not need to be causal. The papers [11, 12, 13] among many others consider use of noncausal operators and causal/anicausal dynamical filters for shaping ILC feedback.

A different approach to the ILC problem is to consider the plant over a batch time as an operator acting on a feedforward sequence and producing an output sequence. The ILC provides a simple feedback for the operator system. Such ILC feedback operator can be designed by minimizing a quadratic performance index, such as in [7, 16, 21]. This approach is similar to MPC methods used in process industries. The difficulty is in dealing with matrices of very large dimensions if there are many samples in the feedforward and output histories (e.g., hundreds or thousands).

One of the problems with ILC encountered in practice is that even if initial reduction of the error is achieved, after many iterations the error might start growing again [12]. One of the reasons for this is small gain of the plant on high frequencies. Controllability can be improved and computational load reduced by using modal decompositions of the initial ILC problem and droping all but the best controllable modes out of the consideration, e.g., see [5, 10].

An ILC update is inherently ill-conditioned. This problem can be *regularized* [22] by adding a penalty for the control effort to an existing performance index such as to a quadratic plant output error in an ILC batch. An ILC solution for regularized LQ problem has form (1), with a term of the form $-BU^{(k-1)}$ added to the r.h.s. [8]. A 'relaxation' term with $B = rI$ was used in ILC update in a few papers including [1, 3], where it was found necessary to ensure robust convergence.

This paper goal is to develop engineering methods for design of high-performance ILC update that can be easily implemented in practice. The methods should be based on formal specifications for the controller and supported by rigorous robustness, performance, and control amplitude analysis methods. Another goal is to develop a unifying view of the ILC problem that will encompass operator, linear-quadratic model-predictive, regularization, and frequency domain design and analysis approaches.

The approach of this paper is based on recent progress in development of control methods for sampled spatially invariant distributed systems [4, 19, 20]. An ILC update for an LTI plant can be considered as such a system, where local time corresponds to a spatial coordinate and batch number to a time variable. One important characteristic of ILC problem for stable LTI plant or other spatially invariant systems with spatially distributed control is that they can be approximately diagonalized (decomposed into modal subsystems) by a Fourier coordinate transformation. In an ILC problem, the Fourier coordinate transformation corresponds to frequency domain analysis of the local time sequencies.

2 Models and Problems

This paper considers a discrete-time ILC problem. Discrete-time models are adequate for development of practical ILC algorithms, since the history data for input and output variables has to be discretized and stored in a digital computer between the batch runs of the process. For the sake of the presentation simplicity, a SISO process is considered. However the analysis and design approaches to follow can be extended to MIMO processes with little or no modification as long as the underlying assumptions of linearity and time-invariance of the process hold.

Let $u(t)$ and $e(t)$ be the feedforward input and the process output error respectilely in the batch, $e^0(t)$ be the initial error with no ILC feedforward applied. Consider the following simple model of a SISO batch process with the discretized control inputs and measurement outputs

$$Y^{(k)} \quad = \quad GU^{(k)} + Y^0, \tag{2}$$

where $Y^{(k)}$, $U^{(k)}$, and Y^0 are vectors with the components $e^{(k)}(t)$, $u^{(k)}(t)$, and $e^0(t)$ respectively, $t = 1, \ldots, N$.

Since the process is LTI $G \in \Re^{N,N}$ in (2) is a Toeplitz matrix. Much of the analysis to follow will be based on properties of Toeplitz matrices.

In what follows, a linear ILC feedback of the following form is considered

$$U^{(k)} = U^{(k-1)} - AY^{(k-1)} - BU^{(k-1)}, \tag{3}$$

where A and B are linear operators ($N \times N$ matrices). Note that (3) gives the most general form of the state feedback for the control problem in hand, where $Y^{(k)}$ is the ILC process state vector and $U^{(k)}$, the controller state vector. In the ILC problem framework this means iterative elimination of the initial error. The time histories Y and U can contain hundreds or thousands elements - data samples.

The analysis to follow uses properties of Toeplitz matrices. The Toeplitz matrices encountered in this paper, such as G, have elements vanishing outside of a narrow diagonal band. Asymptotically for large duration of the batch run interval N such matrices are close to circulant matrices and can be approximately diagonalized in the same Fourier basis [9]. In particular

$$G \approx F^* \mathrm{diag}\,\{g_j\}\, F, \tag{4}$$
$$F_{j,k} = N^{-1/2} e^{-2\pi ijk/N}, \tag{5}$$

where F is a unitary complex Discrete Fourier Transfom (DFT) matrix, F^* is transposed complex conjugate of F, and $F_{j,k}$ are entries of F. Further, in (4) the array of the diagonal elements g_j can be obtained as

$$g_j \approx N^{1/2} F_j \hat{g}, \tag{6}$$

where F_j is the j-th column of the DFT matrix F and $\hat{g} \in \Re^N$ has elements $\hat{g}_1 = G_{1,1}$, $\hat{g}_j = G_{j,1} + G_{j-1,N}$, ($j = 2, \ldots, N$). The vector $\hat{g}$ is a first column of the circulant approximation for G .

The approximate equalities (4), (6) are valid asymptotically for $N \to \infty$. For any finite N, the errors of the approximation can be found or estimated numerically as discussed further on. The approximation (4), (6) makes the analysis results intuitive and shows explicitly how the results depend on frequency-domain properties of the dynamical operators acting on local-time signals within a batch run.

By changing variables in (2) to $\tilde{U} = FU$, $\tilde{Y} = FY$, and $\tilde{Y}^0 = FY^0$, the system (2) takes the form

$$\tilde{y}_j^{(k)} = g_j \tilde{u}_j^{(k)} + \tilde{y}_j^0, \quad (j = 1, \ldots, N), \tag{7}$$

where $y_j^{(k)}$, $\tilde{u}_j^{(k)}$, and $\tilde{y}_j^0$ are components of the vectors $\tilde{U}$, $\tilde{Y}$, and $\tilde{Y}^0$ respectively.

The modal subsystems in (7) are decoupled and each can be controlled independently. The modal control values compensation for the initial error $\tilde{y}_j^0$ in (7) are $\tilde{u}_{ss,j} = \tilde{y}_j^0/g_j$. For poorly controllable modes, i.e., for the modes with vanishingly small modal gains g_j, control that cancels out the initial error can have extremely large magnitude. One way of dealing with an inverse (control) problems for ill-defined system like (7) is offered by the *regularization* theory [22]. The regularization consists of adding a penalty for large values of control to the problem. This can be conveniently done by designing control as a solution to a Linear Quadratic (LQ) problem with performance index including a quadratic control penalty.

This paper presents a systematic LQ design approach that yields practical ILC controllers implementable through FIR window operators. This is done through the Fourier modal analysis (4), (5) and formal specifications for the controller robust stability, performance, and control amplitude.

3 Design and Analysis Approach

This section describes the control design and analysis approach for the ILC system (2)–(3) in general terms. The main assumption is that the matrices G, A, and B can be all approximately diagonalized by the same Fourier transformation, (5) as matrix G (4) such that

$$A \approx F^* \mathrm{diag}\,\{a_j\}\, F, \quad B \approx F^* \mathrm{diag}\,\{b_j\}\, F, \tag{8}$$

In this section the following controller properties are formally analyzed: Robust Stability, Nominal Performance, and Actuator Magnitude

3.1 Robust Stability

By subsituting (2) into (3), we obtain the following closed-loop dynamics equation

$$U^{(k)} = U^{(k-1)} - AGU^{(k-1)} - BU^{(k-1)} - AY^0 \tag{9}$$

It follows from (9) that an ILC update converges exponentially provided that

$$||I - AG - B|| < 1, \tag{10}$$

where $||\cdot|| = \bar{\sigma}(\cdot)$ denotes the operator norm (maximal singular value) of a matrix and I is the $N \times N$ unity matrix.

The errors of the approximation (8) can be taken into account by presenting the matrices G, A, and B in the form

$$G = \hat{G} + \delta G, \quad A = \hat{A} + \delta A, \quad B = \hat{B} + \delta B, \tag{11}$$

where $\hat{G}$, $\hat{A}$, and $\hat{B}$ satisfy conditions (4), (8) exactly; δG, δA, and δB are approximation errors. It will be further assumed that

$$||\delta G|| \leq w_0, \tag{12}$$

where w_0 is a scalar uncertainty parameter.

By substituting (11) into (10), the following sufficient condition of the robust convergence can be obtained

$$||I - \hat{A}\hat{G} - \hat{B}|| + ||\hat{A}\delta G|| + ||\delta AG + B|| < 1, \qquad (13)$$

To guarantee the robustness condition (13), it will be further assumed that the following controller approximation error bound (non-fragility condition) holds

$$\hat{\sigma}(\delta AG + \delta B) < \sigma_1 < 1 \qquad (14)$$

By using (4), (8), (12), and (14), the inequality (13) can be guaranteed if

$$\max_{j=1,\ldots,N} |1 - a_j g_j - b_j| + w_0 \max_{j=1,\ldots,N} |a_j| \ < 1 - \sigma_1 \qquad (15)$$

The conditions (14) and (15) together ensure robustness of the ILC update convergence to a mismatch between the controller, plant and their models (approximations) used in the controller design. By ensuring that (14) and (15) hold, it becomes possible to design controller using the diagonal approximation (4), (8).

3.2 Actuator Magnitude

From (9), the steady state control can be obtained as

$$U_{ss} = -(AG + B)^{-1} AY^0, \qquad (16)$$

where we assume the operator $AG + B$ to be invertible. In case $AG + B$ is not invertible, the control input can theoretically grow without a limit in the updates, which is undesirable.

In this and the next sections, we consider control design specifications for the nominal plant. It will be assumed that (8) holds exactly for the nominal plant. Consider the steady state control vector transformed into the modal coordinates $\tilde{U}_{ss} = FU_{ss}$. From (8) and (16) the modal components of the control can be found as $(\tilde{U}_{ss})_j = \tilde{Y}_j^0 a_j/(a_j g_j + b_j)$. The assumed constraint on control amplitude in the ILC update has the form: $|(\tilde{U}_{ss})_j| < u_{MAX}$, $(j = 1, \ldots, N)$. Constraining the modal components $(\tilde{U}_{ss})_j$ instead of the instantaneous values $(U_{ss})_j$ can be considered as a 'soft' constraint technique. It gives a designer convenient tuning knobs for achieving the desirable control input amplitude. The control magnitude constraint used herein has the form

$$\left| \frac{a_j}{a_j g_j + b_j} \right| y_0 < u_{MAX}, \qquad (17)$$

where it is assumed that $|\tilde{Y}_j^0| \le y_0$. This soft constraint approach follows standard practice of feedback control design where original control amplitude constraints are commonly replaced by frequency domain constraints.

3.3 Nominal Performance

The performance of an ILC controller can be quantified by the residual steady state error obtained after the iterative control update convereges. This steady state error can be computed from (2) and (16) as

$$Y_{ss} = [I - G(AG + B)^{-1} A]Y^0 \qquad (18)$$

It is convenient to analyze modal components of the error (18), i.e., components of the transformed vector $\tilde{Y}_{ss} = FY_{ss}$. By assuming that the plant is nominal so that (8) holds exactly, (18) gives the modal components of the steady state error

$$(\tilde{Y}_{ss})_j = \frac{b_j}{a_j g_j + b_j} \tilde{Y}_j^0 \qquad (19)$$

The control design should ensure that the errors $(\tilde{Y}_{ss})_j$ are possibly small without vilolating the robustness conditions (14), (15) and control amplitude constraint (17). The formulation of this section allows to perform control design and analysis in a decoupled way considering one modal component at a time. This is similar to standard control theory and practice of frequency domain analysis of control loops.

Note that the frequency analysis presented in this section differs from standard (local time) frequency analysis of the dynamical system and control loop underlying the ILC system. This is because, unlike standard control theory, in ILC the control does not need to depend on the measurement in causal way as far as the local time dependencies are concerned.

4 LQ/LTR Design of ILC

A straightforward approach to designing an ILC controller for the system (2) is by minimizing a quadratic performance index. Consider the following performance index including penalties for the next step error $Y^{(k)}$ and control effort

$$J = {Y^{(k)}}^T Y^{(k)} + {U^{(k)}}^T S U^{(k)} + {\Delta U^{(k)}}^T R \Delta U^{(k)} \to \min, \quad (20)$$

where $\Delta U^{(k)} = U^{(k)} - U^{(k-1)}$, S and R are symmetric semidefinite positive penalty weight matrices. In what follows, the penalty weight matrices S and R will be used as tuning parameters and chosen such that the design specifications of Section 3 are satisfied.

4.1 LQ controller

By minimizing (20) subject to the plant model (2) the LQ controller can be obtained in the form (3), where

$$A = D\hat{G}^T, \quad B = DS, \quad D = \left(\hat{G}^T \hat{G} + S + R\right)^{-1} \qquad (21)$$

The penalty weight matrices S and R will be chosen to recover the loop robustness and other specifications similar in spirit to the transfer loop sensitivity recovery in the LQG/LTR loopshaping procedure [18]. Therefore, a name LQ/LTR ILC is used herein for both the controller design procedure and the designed controller. The LQ/LTR ILC approach presented in this section is closely related to the approaches for control of distributed parameter processes, such as paper manufacturing processes, presented in [19, 20].

As a first step towards selecting the penalty weight matrices S and R, consider the following fact

Theorem 1 *Consider the plant model (2) where matrices A, B, and G are given by (11). Consider futher a model-based LQ controller (3), (20). Suppose that (13) holds. Then, the left hand side of the robust convergence condition (14), where $S + R$ is a fixed matrix, is minimized for $R = 0$. In other words, assuming $R = 0$ while keeping $S + R$ fixed provides the best stability margin.*

Proof. In the l.h.s. of (14) the first norm achieves its minimal (zero) value for $R = 0$ and the second term does not depend on R as long as $S + R$ is fixed. ■

The problem of the form (3) was considered in a number of ILC papers and in most cases the problem formulations with $S = 0$ and $R \neq 0$ have been considered. Theorem 1 shows that one should rather select $S \neq 0$ and $R = 0$. Result of Theorem 1 is somewhat counterintuitive because it seems that increasing penalty on the control move $U^{(k)} - U^{(k-1)}$ in (20) should reduce feedback gain and thus increase the robustness. The reason why Theorem 1 result is valid is because there is no uncertainty in the run-to-run dynamics of the system (2)–(3). These dynamics are given by a simple one-step delay. In accordance with (11) all uncertainty is concentrated in the spatial (local batch time) operators A, B, and G.

Based on Theorem 1, the detailed design of the LQ/LTR ILC below sets the control increment penalty to $R = 0$. The penalty S for the accumulated control profile value is selected to achieve the design specifications.

4.2 Detailed LQ/LTR ILC design

Consider controller (21) where $R = 0$. The penalty matrix S in (20) will be taken to be circulant - diagonal in the Fourier basis.

$$S = F^* \mathrm{diag}\{s_j\} F, \quad s_j \geq 0, \qquad R = 0. \tag{22}$$

By subsituting (4) and (22) into (21), the nominal controller can be presented in the form (8), where

$$a_j = \bar{g}_j/(s_j + |g_j|^2), \quad b_i = s_j/(s_j + |g_j|^2), \tag{23}$$

The nominal designed matrices $\hat{A}$ and $\hat{B}$ can differ from the implemented matrices A and B in (11). One of the reasons is that the designed controller will be implemented through finitie convolution window operator. This can lead to truncation errors as in the simulation example below.

Robust stability With (22) and (23) the robust stability condition (15) leads to

$$\max_{j=1,\ldots,N} \frac{|g_j w_j|}{s_j + |g_j|^2} < 1 - \sigma_1 \tag{24}$$

The condition (24) can be presented in the equivalent form

$$s_j > w_0|g_j|/(1-\sigma_1) - |g_j|^2, \quad (j = 1, \ldots, N) \tag{25}$$

Note that (25) gives an explicit lower bound on the modal penaly s_j. This bound depends on the modal gain g_j.

Fragility Consider now the non-fragility (loop robustness to controller modeling error) condidtion (14). This condition is important because the designed circulant controller matrices $\hat{A}$ and $\hat{B}$ (23) can differ from the implemented matrices A and B in accordance with (11). In particular, the designed controller matrices need to be approximated because it is practically desirable to implement finite window convolution operators instead of the matrix multiplications. The controller (3), (21) can be implemented as

$$U^{(k+1)} = U^{(k)} - \hat{D}\left(\hat{G}^T Y^{(k)} + \hat{S} U^{(k)}\right) \tag{26}$$

The matrices $\hat{D}$ and $\hat{S}$ in (26) correspond to finite window convolution operators such that

$$\hat{D}^{-1} = D^{-1} - \delta R, \;\; D = \left(\hat{G}^T \hat{G} + S\right)^{-1}, \;\; \hat{S} = S - \delta S, \tag{27}$$

where δR and δS are approximation errors. By substituting (27) into (26) and comparing to (3), (21), the approximation errors in (11) can be obtainted as

$$\delta A = (\hat{D} - D)\hat{G}, \qquad \delta B = \hat{D}\hat{S} - DS \tag{28}$$

By using (27) and (28) and neglecting second order approximation error, this condition can be presented in the form

$$||\delta A \hat{G} + \delta B|| \equiv ||\hat{D}(\delta R + \delta S)|| < \sigma_1 \tag{29}$$

Note that in accordance with (27) $||\hat{D}|| \leq \left[\underline{\sigma}(D^{-1}) - ||\delta R||\right]^{-1}$. Therefore, a sufficient condition for (29) to hold is $(||\delta R|| + ||\delta S||)/(\underline{\sigma}(D^{-1}) - ||\delta R||) < \sigma_1$, where $\underline{\sigma}(\cdot)$ denotes the minimal sigular value of a matrix. This can be equvalently re-written in the form

$$s_j > (||\delta R|| + ||\delta S||)/\sigma_1 + ||\delta R|| - |g_j|^2 \tag{30}$$

The non-fragility condition (30) gives an explicit lower bound on the modal penaly s_j. This bound depends on the modal gain g_j and the controller approximation errors $||\delta R||$, $||\delta S||$.

Control amplitude By using the diagonal representations (4) and (23), the control magnitude constraint can be presented as the following inequality for the penalty weight s_j

$$s_j \;>\; y_0 \cdot |\bar{g}_j|/u_{MAX} - |g_j|^2, \quad (j = 1, \ldots, N) \tag{31}$$

The lower bound (31) on acceptable penalty weight s_j depends on the modal gain g_j.

Nominal performance In accordance with (19), the performance is defined by modal components of the steady state error. Substituting the diagonal controller gains a_j and b_j from (23) into (19) gives these error components in the form

$$(\tilde{Y}_{ss})_j = \frac{s_j}{s_j + |g_j|^2} \tilde{Y}_j^0 \;\rightarrow\; \min, \tag{32}$$

Since the r.h.s. of (32) is a monotonic increasing function of s_j, the performance is optimized if the penalty weights s_j (22) are as small as possible

$$s_j \;\rightarrow\; \min, \quad (j = 1, \ldots, N) \tag{33}$$

4.3 Graphical design of LQ/LTR ILC

The performance optimality condition (33) together with the inequalities (25), (30), (31) allow to choose the penalties s_j that satisfy all the design specifications and solve the LQ/LTR ILC design problem. Since all of the above conditions define requirements on the modal penalty s_j depending on the modal gain g_j such design can be graphically illustrated in a s_j vs. g_j diagram.

The design is illustrated in a digram of Figure 1, where the upper curve plots the designed penalties s_j vs. the modal gain g_j. In accordance with the performance optimality condition (33), each point on this curve should be selected as low as possible in the upper half plane. This is prevented by the inequalities (25), (30), (31) that provide lower bounds on the value of s_j depending on g_j. These inequalities are shown as shaded areas in Figure 1.

The bound given by the robust stability condition in (25) is a parabola with branches down that has one zero and one

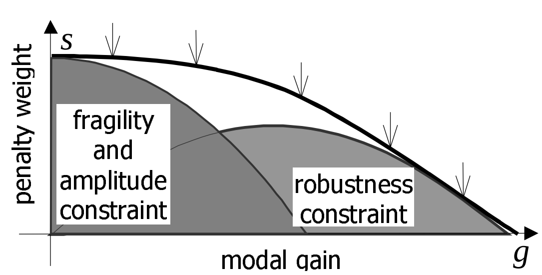

Figure 1: Graphical design of LQ/LTR controller

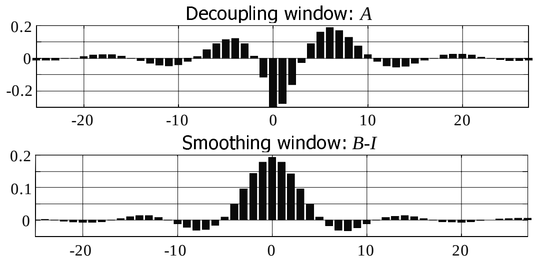

Figure 2: Designed convolution windows for an LQ/LTR ILC controller

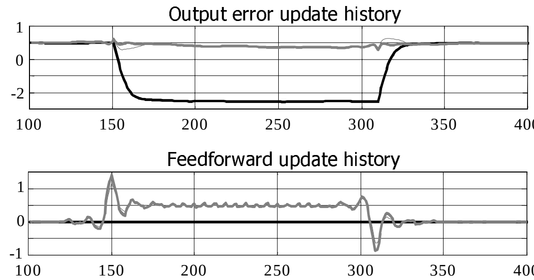

Figure 3: ILC simulation results for the designed LQ/LTR ILC controller

positive root. Both inequalities (30) and (31) have the form $s_j > \text{const} - |g_j|^2$, where either constant does not depend on the modal gain g_j. For controller design only one of these two conditions - one with a larger constant - needs to be considered. This condition appears in the diagram of Figure 1 as an upturned symmetric parabola.

The shaded patches below the parabolas show the prohibited values for the penalty weights s_j. The s_j vs. g_j curve should pass above the union of these areas. Notably, the general shapes of the parabolic design constraints in Figure 1 do not depend on a particular ILC problem (operator G). Only the positive roots of the parabolas might change depending on the problem parameters.

5 Simulation Example

As an example of applying the developed LQ/LTR ILC design approach, consider an ILC control problem for the following simple discrete-time system

$$y(t) = \frac{g}{1-az^{-1}}[v(t)+u(t)],\ e(t) = y(t) - y_d(t) \quad (34)$$

$$v(t) = -\left(\frac{0.15}{1-z^{-1}} + 1.15\right)e(t), \quad (35)$$

where (34) defines the controlled plant and (35) defines a PI feedback controller. The discrete Laplace variable z^{-1} can be interpeted as a unit delay operator; $v(t)$ is the feedback control; $y(t)$ is the plant output; $u(t)$ is the feedforward control computed by the ILC controller; and $e(t)$ is the tracking error observed by the ILC controller. The system evolves on the interval $0 \le t \le 400$. The continuos desired trajectory $y_d(t)$ is ramped on the time interval $t \in [151,\ 310]$ with a unit slope and is constant outside of this interval. The plant parameters in (34) are as follows: $g = 1$ and $a = e^{-1/T_r}$, where $T_r = 6$. These are nominal parameters that will be used in the ILC controller design.

5.1 LQ/LTR ILC design

The LQ/LTR ILC design was performed as discussed in the previous subsection. The initial error amplitude estimate for control design was chosen to be uniform across the modes, $|\tilde{Y}_j^0| \le y_0 = 0.25$. The plant uncertainty was assumed to be uniform across the modes $w_0 = 0.25$. The approximation error of implementing controller was assumed to be $\sigma_1 = 0.15$. This error is related to FIR convolution window implementation of the designed circulant controller matrices.

The designed LQ/LTR ILC finite window convolution operators A and B are illustrated in Figure 2. The designed windows consist of 51 elements each and are convolved with the error profile in the update (3).

The designed LQ/LTR ILC controller was tested in the closed loop simulations. The plant gain and time constant parameters were perturbed by 20% in the simulation. In addition to that, a bounded random noise with an amplitude of ± 0.001 was added to the plant output. The simulation results for first 10 consecutive ILC update steps are illustrated in Figure 3. The lower plot shows the feedforward histories (profiles) in these iterations, the upper plot shows the error profiles. As one can see, the error converges to small final residual error profile in 2-3 iterations.

5.2 Robust PD ILC design

The designed LQ/LTR ILC was compared with a more traditional design of the ILC update. In particular a PD ILC controller as considered in [2] was designed and tested for the same process. The ILC update for this controller has the form

$$u^{(k)}(t) = (1-\rho)u^{(k-1)}(t) - k_P e^{(k)}(t) - k_D \Delta e^{(k)}(t), \quad (36)$$

where $\Delta e^{(k)}(t) = e^{(k)}(t+1) - e^{(k)}(t)$; t denotes local time within a batch run; k is the batch (ILC update) number; ρ, k_P, and k_D are positive parameters of the controller. To ensure robust convergence of the ILC iterations this PD controller has a "relaxation" term defined by the parameter ρ as suggested in [2, 3, 1]. The parameter ρ is chosen large enough to ensure robustness for poorly controllable high-frequency modes but small enough so that the overall ILC controller performance does not deteriorate too much. Further on, the ILC update (36) will be refered to as a Robust PD (R-PD) ILC update.

The controller (36) can be presented in the form (9), where A and B are Toeplitz matrices. These matrices, as well as the closed-loop plant matrix G, are Toeplitz band-diagonal matrices so that the approximation (4), (6) can be used and the coordinate transformations in (8) are given by the Discrete Fourier Transform (5). The parameters a_j and b_j in (8) can be computed in straightforward way as diagonal elements of the matrices FAF^* and FBF^*, where F is the unitary DFT matrix. This allows to apply the tools of Section 3 to analysis of the R-PD controller.

In simulations, the proportional gain k_P, the derivative gain k_P, and ρ were such that the robust stability margin in (15) for the R-PD ILC controller is close to the margin for the designed LQ/LTR ILC controller. At the same time, an attempt was made to optimize performance of the R-PD ILC controller. As a result, the following parameter values were selected

$$k_P = 0.00095, \qquad k_D = 0.003, \qquad \rho = 0.02 \quad (37)$$

The R-PD ILC and LQ/LTR ILC controllers can be compared by observing their steady state performance charac-

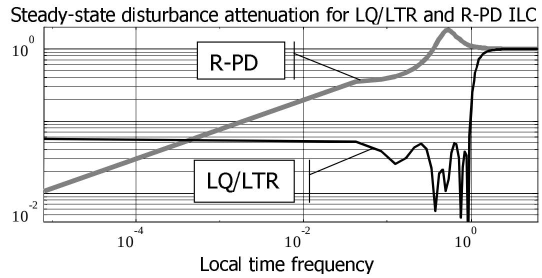

Figure 4: Nominal performance - disturbance attenuation gain for LQ/LTR and R-PD ILC controller

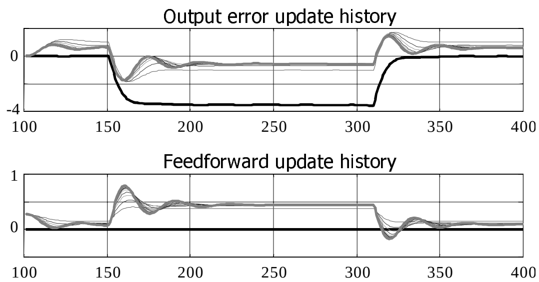

Figure 5: ILC simulation results for the R-PD ILC controller

teristics as defined by the disturbance attenuation multiplier $h_j = |b_j|/|a_j g_j + b_j|$ in (19). Figure 4 illustrates the disturbance attenuation gain h_j computed for both ILC controllers. The gain h_j in Figure 4 is plotted vs. the local-time frequency $\nu_j = 2\pi j/N$. Figure 4 shows that the nominal disturbance attenuation gain achieved by the LQ/LTR ILC controller is uniformly small for lower local-time frequencies ν_j and increases approaching 1 for higher frequencies outside of the plant bandwidth, where the gain g_j is small. The disturbance attenuation gain for the R-PD ILC controller is larger than that for the LQ/LTR ILC, except for the zero frequency (DC) component of error. In fact, the R-PD ILC even amplifies some disturbance components by a factor of up to 1.81 in the vicinity of the frequency $\nu = 0.55$ rad/sample.

The ILC simulation results for the R-PD ILC controller (36), (37) are displayed in Figure 5 that is similar in format to Figure 3. As expected from the performance analysis illustrated in Figure 4, much larger tracking error is observed for R-PD in Figure 5 compared to the LQ/LTR ILC update in Figure 3. The convergence speed for R-PD is also much slower. Figure 5 plots results for each 12th ILC iteration while Figure 3 plots every ILC iteration. It takes 20-30 iterations for the R-PD ILC to converege, compared to 3-4 iterations for the LQ/LTR ILC.

References

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

[2] Arimoto, S. "Learning control theory for robotic motion," *Int. J. of Adaptive Control, and Signal Proc.*, vol. 4, 1990, pp.453–564.

[3] Arimoto, S., Naniwa, T., and Suzuki H. "Robustness of P-type learning control with a forgetting factor for robotic motions," In *29th IEEE Conf. on Decision and Control*, pp. 2640–2645, Honolulu, Hawaii, USA, Dec. 1990.

[4] Bamieh, B., Paganini, F., and Dahleh, M. "Distributed Control of Spatially-Invariant Systems," *IEEE Trans.on Automatic. Contr.*, to appear, 1999.

[5] Cheng, W., and Wen, J.T. "Output trajectory tracking based on feedforward learning," *American Control Conf.*, pp. 1747-1751, Baltimore, MD, June 1994

[6] Chen, Yangquan. *A Bibliographical Library for ILC Research,* /http:ilc.ee.nus.edu.sg/ILC/, 1998

[7] Gorinevsky, D., Torfs, D, and Goldenberg, A. A. "Learning approximation of feedforward control dependence on the task parameters," *IEEE Trans. on Robotics and Automation*, vol. 13, no. 4, 1997, pp. 567-581

[8] Gorinevsky, D. "On regularized feedback update of distributed-parameter systems in control and nonlinear estimation applications," *American Control Conf.*, Albuquerque, NM, June 1997

[9] Gray, R.M. *Toeplitz and Circulant Matrices: A Review*, /http:www.isl.stanford.edu/ gray/toeplitz.pdf, 1998

[10] Gugliemo, K., and Sadeh, N. "Experimental evaluation of a new robot learning controller," *1991 IEEE Int. Conf. on Robotics and Autom.*, pp. 734–739, Sacramento, CA, April 1991.

[11] Ishihara, T., Abe, K., Takeda, H. "A discrete-time design of robust iterative learning algorithm," *IEEE Trans. on Syst. Man and Cybern.*, vol. 22, no. 1, 1992, pp. 74–84.

[12] Longman, R. W., and Wirkander, S.-L. "Automated tuning concepts for iterative learning and repetitive control laws," *37th IEEE Conf, on Decision and Control*, pp. 192–197, Tampa, FL, 1998.

[13] Messner, W. et al. "A new adaptive learning rule," *IEEE Trans. on Automat. Cont.*, vol. 36, no. 2, pp. 188–197, 1991.

[14] Moore, K. L. *Iterative learning control for deterministic systems.* Advances in Industrial Control. Springer-Verlag, 1993.

[15] Padiew, F., and Su, R. "An H_∞ approach to learning control systems," *Int. J. of Adaptive Cont. and Signal Proc.*, vol. 4, pp.465–474, 1990.

[16] Phan, M.Q. and Juang, J.N. "Designs of learning controllers based on an auto-regressive representation of a linear system," *AIAA J. of Guidance, Control, and Dynamics*, vol. 19, no. 2, 1996, pp. 355–362.

[17] Rogers, E. and Owens, D. H.. *Stability Analysis for Linear Repetitive Processes.* Springer-Verlag, Berlin, 1992.

[18] Stein G., and Athans, M. "The LQG/LTR procedure for multivariable feedback control design," *IEEE Tr. on Automatic Control*, vol. 32, no. 2, 1987, pp. 105–114.

[19] Stewart, G. E., Gorinevsky, D., and Dumont, G. A. "Design of a practical robust controller for a sampled distributed parameter system," *37th IEEE Conf. on Decision and Control*, Tampa, FL, December 1998

[20] Stewart, G.E., Gorinevsky, D.M., and Dumont, G.A. "Spatial loopshaping: A case study on cross-directional profile control," *American Control Conf.*, pp. 3098–3103, San Diego, CA, 1999.

[21] Tao, K.M., Kosut, R.L., and Aral, G. "Learning feedforward control," *American Contr. Conf*, pp.2575–2579, Baltimore, MD, 1994.

[22] Tikhonov, A.N., and Arsenin, V.Ya. *Solutions of Ill-Posed Problems.* Halsted Press, Washington, 1977.