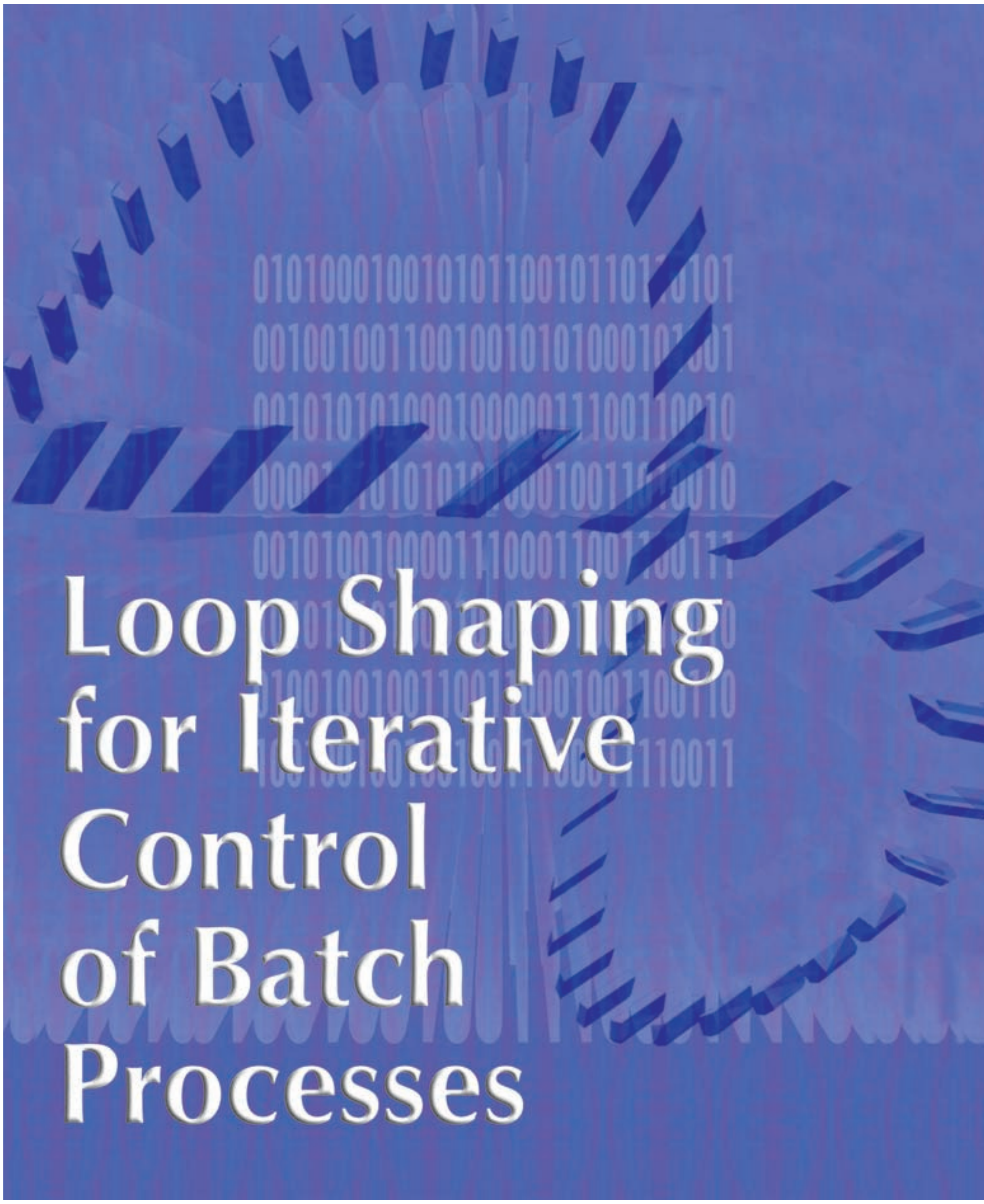
FEATURE



Loop Shaping for Iterative Control of Batch Processes

©IMAGESTATE

By Dimitry Gorinevsky

The idea of iterative control of batch processes has been discussed in the literature for some time. Many related concepts have been published under the names *iterative control*, *learning control*, *run-to-run control*, and others. This article uses the term *iterative learning control* (ILC).

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

The approach described in this article is based on recent progress in the understanding of control for sampled spatially invariant distributed systems [19]-[21]. An ILC update for a linear time-invariant (LTI) plant can be considered a two-dimensional distributed system where local time corresponds to a spatial coordinate and batch number to a time variable. An important characteristic of the ILC problem for a stable LTI plant or other spatially invariant system with spatially distributed control is that it 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 dynamical operators in the local time within a batch run. Run-to-run evolution of a batch process with an ILC update is a system with states comprising plant input and output histories over the batch time. The run-to-run dynamics of the system are a simple one-step delay.

This article makes several contributions to the design of ILC systems:

- It presents a systematic method for the analysis and design of ILC systems. The method is based on a fre-

The author (dimitry.gorinevsky@honeywell.com) is with Honeywell Global Controls Laboratory, CA35-5272H, 1 Results Way, Cupertino, CA 95014, U.S.A.

0272-1708/02/$17.00©2002IEEE

quency domain approach and is related to the operator, linear quadratic (LQ) model predictive, and modal viewpoints. Rigorous ILC design is based on detailed specifications, including robustness, controller nonfragility requirements, and control amplitude constraints. The designed ILC update is implementable through noncausal FIR window operators.

- The robust control design approach described herein assumes that the underlying system is LTI. The nonlinearity effects are attributed to model uncertainty, and a robust controller is designed. An example shows that such a robust design can provide very good performance for a highly nonlinear process.
- The design method is based on a loop-shaping procedure that is *noncausal* in the local time of the batch because the data of the previously completed run are used for control computation. This is inherently simpler than typical control design where the control input is constrained to be causal in the measurements. The loop-shaping controller design is performed by manipulating quadratic penalty weights in an LQ problem. This is closely related to the LQG/LTR (linear-quadratic Gaussian/loop transfer recovery) design method for dynamic controllers [22].
- The article proves that to achieve an optimal performance/robustness tradeoff, the penalty for the run-to-run control increment in the LQ performance index should be set to zero. This is somewhat counterintuitive and contrary to the conventional wisdom for ILC control design.

The article is organized as follows. The next section formulates an ILC design problem and modal analysis approach in the frequency domain that uses the Fourier transform. Following that, the controller design specifications are formulated in a frequency domain form. Next, the LQ/LTR controller design procedure is presented where the quadratic penalty weights are used to recover the design specifications. Finally, a simulated example of ILC in a highly nonlinear batch thermal process illustrates an application of the developed approach.

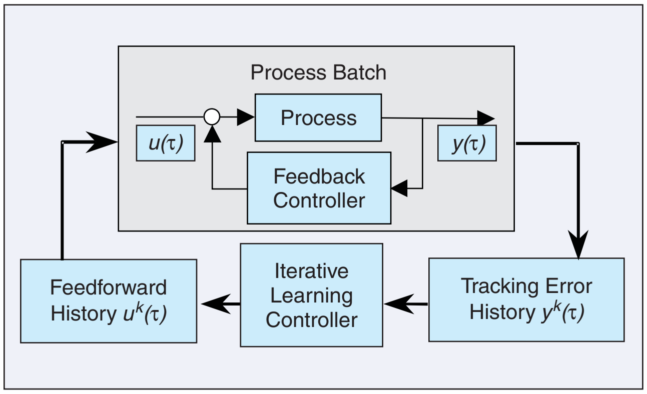

Process Batch
Process
$u(\tau)$
$y(\tau)$
Feedback Controller
Feedforward History $u^k(\tau)$
Iterative Learning Controller
Tracking Error History $y^k(\tau)$

Figure 1. *Iterative learning control diagram.*

Introduction

The main idea of ILC is illustrated in Fig. 1. An ILC controller provides feedforward control input to a plant through a batch run, as shown in Fig. 1, in combination with a feedback controller. The feedforward control history for an entire batch run is stored in a memory buffer, and the plant output in the batch is stored in another memory buffer. The ILC update is computed between the batches and uses the output history buffer data to calculate an update to the feedforward control history buffer data. Conceptually, pure feedback control algorithms consider the tracking error to be completely unknown and continuously attempt to compensate for this error as the process goes through a batch run. Feedback control ignores the fact that the error is mostly reproducible from one batch run to another. An ILC approach uses this knowledge to update the feedforward control sequence. Some relevant ideas of the prior work are briefly summarized below. A more detailed bibliography on ILC can be found in [1] and [2] and in the recent special issue on the topic [3].

Most of the ILC approaches in the literature consider updating the feedforward control as

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

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 tracking error 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 a form similar to standard dynamical controllers, such as P (proportional) or PD (proportional-derivative) operators acting on the plant output sequence as a local-time function (e.g., see [4]). Feedback operators described by high-order rational transfer functions can also be used (e.g., see [5]). The convergence analysis for such operators is conveniently and intuitively performed in the frequency domain, such as in [2], [6] and other works. The main drawback of the described approaches is that they follow standard control design methods and use feedback update operators that are causal in local time. However, in the ILC problem, the error history for an entire past run is available at once, and thus the feedback operator does not need to be causal. Several researchers ([7]-[9] and others) have considered the use of noncausal operators and causal/anticausal dynamical filters for shaping ILC feedback.

In an ILC problem, an operator defined by the plant acts on a feedforward sequence over a batch duration and produces an output sequence. The ILC provides a simple feedback for such an operator system. An ILC feedback operator can be designed by minimizing a quadratic performance index, such as in [10]-[12]. This approach is similar to the model predictive control methods used in process industries. The difficulty is that large sizes of the feedforward and output histories (e.g., hundreds or thousands of samples) result in a necessity to deal with matrices of very large dimensions.

A problem with ILC encountered in practice is that even if the error is initially reduced, it might start growing again after many iterations [8]. One reason for this is a small gain of the plant at high frequencies. This can be seen as poor controllability (ill-conditioning) of the ILC update system. The controllability can be improved and computational load reduced by using modal decompositions of the initial ILC problem and dropping all but the best controllable modes from consideration (e.g., see [13] and [14]). An ILC update is an inherently ill-conditioned system. It can be regularized [15] by adding a penalty for the control effort to an existing performance index, such as a quadratic plant output error in an ILC batch. An ILC solution for a regularized linear-quadratic (LQ) problem has the form (1), with a term of the form $-BU^{(k-1)}$ added to the right-hand side (r.h.s.) [16]. A "relaxation" term with $B = rI$ was used in [17] and [18], where it was found necessary to ensure robust convergence of the ILC update.

Problem Statement

Discrete-time models are adequate for the development of practical ILC algorithms, as the history data for input and output variables must be sampled and stored in a digital computer between the batch runs of the process. For simplicity, a single-input, single-output (SISO) process is considered; however, the analysis and design approaches to follow can be extended to multiple-input, multiple-output (MIMO) processes with little or no modification as long as the underlying assumptions of linearity and time invariance of the process hold.

In this article, when designing and analyzing an ILC update, the underlying batch process is assumed to be LTI. The process nonlinearity effects are incorporated into the model uncertainty and handled in the controller robustness design and analysis. The application example given later shows that this approach allows for an efficient design of the ILC update for a highly nonlinear process. Let $u(t)$ and $e(t)$ be the feedforward input and the process output error, respectively, in the batch, and let $e^0(t)$ be the initial error with no ILC feedforward applied. Consider the following simple model of a SISO batch process with discretized control inputs and measurement outputs

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

$$Y^{(k)} = \begin{bmatrix} e^{(k)}(1) \\ e^{(k)}(2) \\ \vdots \\ e^{(k)}(N) \end{bmatrix}, \; U^{(k)} = \begin{bmatrix} u^{(k)}(1) \\ u^{(k)}(2) \\ \vdots \\ u^{(k)}(N) \end{bmatrix}, \; Y^0 = \begin{bmatrix} e^0(1) \\ e^0(2) \\ \vdots \\ e^0(N) \end{bmatrix}, \tag{3}$$

where N is the batch duration in the sampled time. For a SISO process, the input and output history profile vectors have the same dimension $U \in \Re^N, Y \in \Re^N$. Since the process is LTI, the input/ouput Jacobian $G \in \Re^{N,N}$ in (2) is a Toeplitz matrix. Each column of G is a pulse response of the process initiated at the respective sample.

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

$$U^{(k)} = U^{(k-1)} - AY^{(k-1)} - BU^{(k-1)}, \tag{4}$$

where A and B are linear operators ($N \times N$ matrices). Note that (4) gives the most general form of the state feedback for the ILC problem at hand, where $Y^{(k)}$ is the ILC process state vector and $U^{(k)}$ is 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 of elements (data samples).

As mentioned earlier, the inner loop where the ILC feedforward is applied (see Fig. 1) is assumed to be stable. Thus, the pulse response of this inner loop vanishes exponentially away from the input pulse applied. Toeplitz matrices encountered in this article, such as G, have elements vanishing outside of a narrow diagonal band. For a large duration of the batch run interval N, such matrices can be approximately diagonalized in the Fourier basis such that

$$G \approx F^* \operatorname{diag}\{g_j\} F, \tag{5}$$

$$F_{l,m} = N^{-1/2} e^{-2\pi i l m / N}, \tag{6}$$

where F is a unitary complex discrete Fourier transform (DFT) matrix, F^* is the transposed complex conjugate of F, and $F_{l,m}$ are entries of F. In (5), the array of the modal gains g_j can be obtained as

$$g_j = N^{1/2} (F\hat{g})_j, \tag{7}$$

where g_j is the normalized component j of the product of F and the system pulse response $\hat{g} \in \mathfrak{R}^N$. The latter is the first column of the matrix G.

The approximation (5), (7) is valid for the same reason that frequency domain analysis can be used for LTI dynamical, signal processing, and control systems. From a rigorous mathematical standpoint, the error of the approximation (5) is caused by considering the ILC update on a finite interval of local time $[1,\ N]$. For large N, a band-diagonal matrix G is close to a circulant matrix. For circulant matrices, (5), (7) holds exactly. An excellent introductory tutorial and further references on circulant matrices and circulant approximations of Toeplitz matrices can be found in [23]. The analysis using circulant matrices is employed in related two-dimensional control problems in [20] and [21]. Another possible mathematical derivation is to assume that the ILC update interval $[1,\ N]$ is such that the system is in the steady state and the tracking error is zero on the interval ends. Alternatively, the update interval might be embedded into a much larger (infinite) interval where such conditions hold. Essentially, the issue of validity of the approximation (5), (7) is the same as that of frequency domain analysis of LTI systems on a finite time interval, such as in analyzing performance of an LTI feedback controller by tracking a step or a finite-duration ramp. Such approximations are common in control systems engineering.

It is further assumed that the feedback matrices A and B in (4) can also be approximately diagonalized by the same Fourier transformation (6) as matrix G (5) such that

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

An iterative learning controller provides feedforward control input to a plant through a batch run.

This assumption holds if A and B correspond to localized LTI operators in the local time within a batch run. In other words, the action of A and B on the local time data sequence can be approximately represented as a convolution of the sequence with a localized pulse response.

Let $\hat{g}(z)$, $\hat{a}(z)$, and $\hat{b}(z)$ be transfer functions (z-transforms) computed for the pulse responses corresponding to the operators G, A, and B, respectively. The spectra in (5) and (8) can then be approximately represented as [23]

$$a_j = \hat{a}\left(e^{i\omega_j}\right), \quad b_j = \hat{b}\left(e^{i\omega_j}\right), \quad g_j = \hat{g}\left(e^{i\omega_j}\right), \tag{9}$$

where $\omega_j = 2\pi j / N$ is the local time frequency corresponding to the respective eigenvalues of the operators (5) and (8). For noncausal operators A and B, the transfer functions $\hat{a}(z)$ and $\hat{b}(z)$ in (9) are noncausal transfer functions obtained as two-sided z-transforms of the respective localized pulse responses [24]. The approximations (5) and (8) are valid asymptotically for $N \to \infty$. For a finite batch length N, the errors of the approximation can be found numerically. The approximation (5), (7) 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 $\tilde{Y}^0 = FY^0$, system (2) takes the form

$$\tilde{y}_j^{(k)} = g_j \tilde{u}_j^{(k)} + \tilde{y}_j^0, \quad (j = 1, \ldots, N), \tag{10}$$

where $\tilde{y}_j^{(k)}, \tilde{u}_j^{(k)}$, and $\tilde{y}_j^0$ are components of the vectors $\tilde{U}, \tilde{Y}$, and $\tilde{Y}^0$, respectively. In the frequency domain representation (10) of the ILC system (as well as in (9)), the frequencies ω_j correspond to the local time. Each frequency harmonic (mode) in (10) evolves and can be controlled independently from run to run.

The modal control values compensating for the initial error $\tilde{y}_j^0$ in (10) are $\tilde{u}_{ss,j} = -\tilde{y}_j^0 / g_j$. For poorly controllable modes (i.e., for modes with vanishingly small modal gains g_j), control that exactly cancels out the initial error can have extremely large magnitude. One way of dealing with an inverse (control) problem for ill-defined systems such as (10) is offered by the *regularization* theory [15]. The regularization consists of adding a penalty for large values of control to the problem. This can be done conveniently by designing the control as a solution to an LQ problem with a quadratic control penalty in the performance index.

Analysis Approach

This section presents the control analysis approach for the ILC system (2)-(4), which consists of studying robust stability, nominal performance, and actuator move magnitude for the control loop. Unfortunately, standard control-theoretic methods of robust analysis are not directly applicable to the problem in question. This is because all uncertainty in the run-to-run evolution of the ILC system is related to the local time dynamics model (i.e., to the parameter matrices in the run-to-run dynamics). Such uncertainty is highly structured and warrants derivation of customized robustness conditions, which is presented in this section.

Robust Stability

By substituting (2) into (4), we obtain the following closed-loop dynamics equation:

$$U^{(k)} = U^{(k-1)} - AGU^{(k-1)} - BU^{(k-1)} - AY^0. \tag{11}$$

It follows from (11) that an ILC update converges exponentially provided that

$$\|I - AG - B\| < 1, \tag{12}$$

where $\|\cdot\| = \overline{\sigma}(\cdot)$ is the maximum singular value of a matrix and I is the $N \times N$ identity matrix.

The errors of the approximation (5), (8) can be taken into account by presenting the matrices G, A, and B in the form

$$G = \hat{G} + \delta G, \quad A = \hat{A} + \delta A, \quad B = \hat{B} + \delta B, \tag{13}$$

where $\hat{G}$, $\hat{A}$, and $\hat{B}$ satisfy conditions (5), (8) exactly; δG, δA, and δB are approximation errors. The uncertainties δG, δA, and δB are assumed to be completely unstructured. Only norm bounds on these operators are used in the analysis to follow. In particular, it is further assumed that

$$\|\delta G\| \le g_0, \tag{14}$$

where g_0 is a scalar uncertainty parameter.

Substituting (13) into (12) gives the following sufficient condition for robust convergence:

$$\left\| I - \hat{A}\hat{G} - \hat{B} \right\| + \left\| \hat{A}\delta G \right\| + \left\| \delta AG + \delta B \right\| < 1. \tag{15}$$

The following controller approximation error bound (nonfragility condition) is further assumed:

$$\| \delta AG + \delta B \| < \sigma_1 < 1, \tag{16}$$

where σ_1 can be considered as one of the controller design parameters. By using (5), (8), (14), and (16), the frequency domain condition for the inequality (15) to hold is

$$\max_{j=1,\ldots,N} \left| 1 - a_j g_j - b_j \right| + g_0 \max_{j=1,\ldots,N} \left| a_j \right| < 1 - \sigma_1. \tag{17}$$

The conditions (16) and (17) together ensure robustness of the ILC update convergence to mismatches between the controller, the plant, and their models (approximations) used in the controller design. Ensuring that (16) and (17) hold makes it possible to design the controller using the diagonal approximation (5), (8).

Actuator Move Magnitude

From (11), the steady-state control can be obtained as

$$U_{ss} = -(AG + B)^{-1} AY^0, \tag{18}$$

where, in accordance with (12), the operator $AG + B$ is invertible.

In this and the following 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 (18), 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 control amplitude specifications in the ILC update have the form $\left|(\tilde{U}_{ss})_j\right| < u_{\max}$, $(j = 1,\ldots,N)$. Constraining the modal components $(\tilde{U}_{ss})_j$ instead of the instantaneous values $(U_{ss})_j$ is computationally convenient. The control frequency domain magnitude constraint used herein has the form

$$\left| \frac{a_j}{a_j g_j + b_j} \right| y_0 < u_{\max}, \tag{19}$$

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

Nominal Performance

The performance of an ILC controller can be quantified by the residual steady-state error obtained after the iterative control update converges. This steady-state error can be computed from (2) and (18) as

$$Y_{ss} = \left[I - G(AG + B)^{-1} A \right] Y^0. \tag{20}$$

It is convenient to analyze modal components of the error (20), the components of the vector $\tilde{Y}_{ss} = FY_{ss}$. By assuming (5) and (8) hold exactly, (20) gives the modal components of the steady-state error in the form

$$\left(\tilde{Y}_{ss}\right)_j = \frac{b_j}{a_j g_j + b_j} \tilde{Y}_j^0. \tag{21}$$

The control design should ensure that the errors $(\tilde{Y}_{ss})_j$ are possibly small without violating the robustness conditions (16), (17) and control amplitude constraint (19). The formulation of this section allows performing control design and analysis in the frequency domain considering one modal component at a time. This is similar to the standard control theory and practice of frequency domain analysis of control loops. Yet the frequency domain analysis of this section differs from standard frequency analysis of dynamical systems and control loops. This is because in ILC the control does not need to depend on the measurement in a causal way as far as the local time dependencies are concerned. Also, the analysis is concerned with the system evolution from run to run, as opposed to the local time dynamics.

LQ/LTR Design of ILC

A straightforward approach to designing an ILC controller for system (2) is by minimizing a quadratic performance index. The index, including penalties for the next step error $\hat{Y}^{(k)}$ (predicted using the system model (2)) and the control effort, has the form

$$J = \hat{Y}^{(k)T} \hat{Y}^{(k)} + U^{(k)T} SU^{(k)} + \Delta U^{(k)T} R \Delta U^{(k)} \to \min, \tag{22}$$

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

LQ Controller

By minimizing (22) subject to the plant model (2), the controller can be obtained in the form (4), where

$$A = D\hat{G}^T, \quad B = DS, \quad D = \left(\hat{G}^T \hat{G} + S + R \right)^{-1}. \tag{23}$$

The penalty weight matrices S and R will be chosen to recover the loop robustness and other specifications similar to the transfer loop sensitivity recovery in the LQG/LTR loop-shaping procedure [22]. Therefore, the name LQ/LTR ILC is used herein for both the controller design procedure and the designed controller. The LQ/LTR ILC approach of this section is closely related to the approaches for control of distributed-parameter processes, such as paper manufacturing processes [20], [21].

In the ILC problem, the error history for an entire past run is available at once, and the feedback operator does not need to be causal.

As a first step toward selecting the matrices S and R, consider the following fact.

Proposition 1: Consider the plant model (2) where matrices A, B, and G are given by (13). Consider further a model-based LQ controller (4), (22). Suppose that (16) holds. Then the left-hand side of the robust convergence condition (15), 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: With matrices (23) substituted in (15), the norm $\left\| I - \hat{A}\hat{G} - \hat{B} \right\| = \|DR\|$ achieves its minimal (zero) value for $R = 0$. The norm $\left\| \hat{A}\delta G \right\| = \left\| D\hat{G}\delta G \right\|$ does not depend on R as long as $S + R$ is fixed. Finally, the estimate of $\|\delta AG + \delta B\|$ is given by (16) and depends neither on R nor on S. ■

The problem of the form (22) was considered in several ILC papers, and in most cases, formulations with $S = 0$ and $R \neq 0$ have been studied. Proposition 1 shows that one should instead select $S \neq 0$ and $R = 0$. The result of Proposition 1 is somewhat counterintuitive because it seems that increasing penalty on the control move $U^{(k)} - U^{(k-1)}$ in (22) should reduce feedback gain and thus improve the robustness. Proposition 1 states that this is not the case. The reason is the absence of uncertainty in the run-to-run dynamics of the system (2)-(4). These dynamics are a simple one-step delay. In accordance with (13), all uncertainty is concentrated in the spatial (local batch time) operators A, B, and G.

Note that with (23) and for a perfect model $\hat{G} = G$, the steady-state error (20) becomes

$$\begin{aligned} Y_{ss} &= \left(I - G\left[D(G^T G + S) \right]^{-1} DG^T \right) Y^0 \\ &= \left(I - G(G^T G + S)^{-1} G^T \right) \\ &= Y^0. \end{aligned}$$

Clearly, Y_{ss} does not depend on R and increases for larger S. Therefore, and in accordance with Proposition 1, the detailed design of the LQ/LTR ILC described below sets the control increment penalty to $R = 0$. The penalty S for the accumulated control profile value should be selected to achieve a tradeoff between the robustness and nominal performance (steady-state error) requirements.

LQ/LTR ILC Design Specifications

Consider controller (23) where $R = 0$. The penalty matrix S in (23) will be taken to be diagonal in the Fourier basis

$$S = F^* \mathrm{diag}\{s_j\} F, \quad s_j \geq 0, \quad R = 0. \tag{24}$$

By substituting (5) and (24) into (23), the nominal controller can be presented in the form (8), where

$$a_j = g_j \Big/ \left(s_j + |g_j|^2 \right), \quad b_j = s_j \Big/ \left(s_j + |g_j|^2 \right). \tag{25}$$

Robust Stability

With (24) and (25), the robust stability condition (17) leads to

$$\max_{j=1,\ldots,N} \frac{|g_j|}{s_j + |g_j|^2} < \frac{1-\sigma_1}{g_0}. \tag{26}$$

Condition (26) can be presented in the equivalent form to obtain an explicit lower bound on the modal penalty s_j depending on the modal gain g_j:

$$s_j > \frac{g_0}{1-\sigma_1}|g_j| - |g_j|^2, \quad (j = 1,\ldots,N). \tag{27}$$

Fragility

Consider the nonfragility (loop robustness to controller modeling error) condition (16). The designed controller matrices $\hat{A}$ and $\hat{B}$ (25) can differ from the implemented matrices A and B as described in (13). In particular, approximation might be required because it is practically desirable to implement finite window convolution operators instead of the matrix multiplications. The controller (4), (23) can be implemented as

$$U^{(k+1)} = U^{(k)} - \hat{D}\left(\hat{G}^T Y^{(k)} + \hat{S} U^{(k)} \right). \tag{28}$$

The matrix $\hat{D}$ in (28) corresponds to a finite window convolution operator that approximates D in (23) such that

$$\hat{D}^{-1} = D^{-1} - \delta R, \tag{29}$$

where δR is the approximation error. By substituting (29) into (28) and comparing to (4) and (23), the approximation errors in (13) can be obtained as

$$\delta A = \left(\hat{D} - D\right)\hat{G}^T, \quad \delta B = \left(\hat{D} - D\right)S. \tag{30}$$

By using (29) and (30) and neglecting the second-order approximation error, (16) can be presented in the form

$$\left\|\delta A\hat{G} + \delta B\right\| \equiv \left\|\hat{D}\delta R\right\| < \sigma_1. \tag{31}$$

Note that in accordance with (29), $\left\|\hat{D}\right\| \le \left[\underline{\sigma}(D^{-1}) - \|\delta R\|\right]^{-1}$, where it is assumed that $\|\delta R\| < \underline{\sigma}(D^{-1})$. Therefore, a sufficient condition for (31) to hold is $\|\delta R\| / \left(\underline{\sigma}(D^{-1}) - \|\delta R\|\right) < \sigma_1$, where $\underline{\sigma}(\cdot)$ denotes the minimum singular value of a matrix. This can be rewritten in the equivalent form

$$s_j > \|\delta R\|\left(1 + \sigma_1^{-1}\right) - \left|g_j\right|^2. \tag{32}$$

The nonfragility condition (32) gives an explicit lower bound on the modal penalty s_j. This bound depends on the modal gain g_j and the controller approximation error $\|\delta R\|$.

Control Amplitude

By using the frequency domain representations (5) and (25), the control magnitude constraint can be presented as the following inequality for the penalty weight s_j:

$$s_j > y_0 \cdot \left|g_j\right| / u_{\max} - \left|g_j\right|^2, \quad (j = 1, \ldots, N). \tag{33}$$

The lower bound (33) on acceptable penalty weight s_j depends on the modal gain g_j.

Nominal Performance

In accordance with (21), the performance is defined by modal components of the steady-state error. Substituting the frequency domain gains a_j and b_j from (25) into (21) gives these error components in the form

$$\left(\tilde{Y}_{ss}\right)_j = \frac{s_j}{s_j + \left|g_j\right|^2}\tilde{Y}_j^0 \to \min. \tag{34}$$

Since the r.h.s. of (34) is a monotonic increasing function of s_j, the performance is optimized if the penalty weights s_j (24) are as small as possible:

$$s_j \to \min, \quad (j = 1, \ldots, N). \tag{35}$$

Loop-Shaping Design of an LQ/LTR ILC

The performance optimality condition (35), together with the design constraints (27), (32), (33), allows a designer to choose the penalties s_j that satisfy all the design specifications and solve the LQ/LTR ILC design problem. Since the index j spans the local time frequencies, such design of the controller is a loop-shaping design. In classical loop-shaping approaches, conditions on sensitivity function, complementary sensitivity, and the like are considered. In contrast, herein the loop-shaping design constraints have been formulated as explicit conditions on the penalty weights s_j; these conditions depend on the modal frequency via the modal gain g_j only. The design is graphically illustrated in Fig. 2, where the upper curve shows designed penalties s_j versus the modal gain g_j. The performance optimality condition (35) requires selecting each point on this curve as low as possible in the upper half plane. The inequalities (27), (32), (33) provide constraints on the value of s_j depending on g_j. These constraints are shown as shaded areas in Fig. 2.

The robust stability bound in (27) is an upturned parabola with one zero and one positive root. Both inequalities (32) and (33) have the form $s_j > \text{const} - \left|g_j\right|^2$, where either constant does not depend on the modal gain g_j. For controller design, only one of these two conditions, the one with the larger constant, needs to be considered. This condition appears in the diagram of Fig. 2 as an upturned symmetric parabola.

The shaded patches below the parabolas show the prohibited values for the penalty weights s_j. The s_j versus g_j curve should pass above the union of these areas. Notably, the general shapes of the parabolic design constraints in Fig. 2 do not depend on a particular ILC problem (operator G). Only the positive roots of the parabolas might change depending on the problem parameters.

Example: Control of Rapid Thermal Processing

This section considers an example describing two closely related thermal batch processes: rapid thermal processing (RTP) and annealing. RTP is presently one of the mainstream semiconductor manufacturing technologies. In a batch RTP process, a silicon wafer is placed into a chamber and rapidly heated to a high temperature. During the heating, the wafer temperature must closely follow a target (ref-

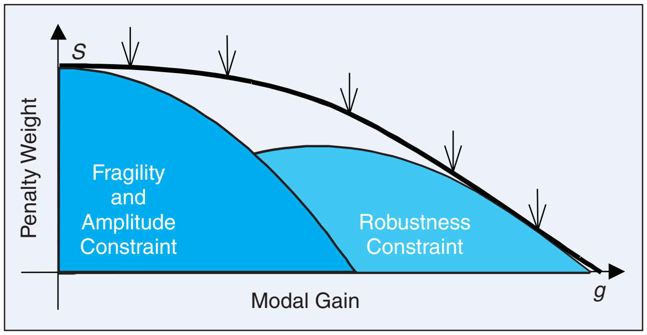


Figure 2. *Graphical design of LQ/LTR controller.*

erence) temperature profile. The annealing process considered here is used in the carbonization of parts in a high-temperature furnace.

In both thermal processing applications, tight and rapid control of the temperature profile is critical for reducing processing time and improving process quality (thermal deformations caused by nonuniform heating) and run-to-run consistency. As a result, several sophisticated advanced process control (APC) approaches have been developed for RTP process control (e.g., see [25] and [26]). Performance of such APC feedback algorithms is subject to fundamental limitations caused by the process dynamics and delays. In addition, the APC algorithms require identification of high-fidelity models and are expensive to set up and maintain in the field.

A run-to-run control application is being developed by Honeywell for annealing and rapid thermal processing as a practical implementation of the ILC technology. The controller is designed using the robust LQ/LTR ILC algorithms, which work despite the process nonlinearity, identification errors, and other deficiencies of the simple linear process model used in the ILC update design.

Thermal Processing Model

To illustrate an application of the ILC technique, consider an annealing process model. The model and the simulation results below are closely related to the models and results for the RTP published in several recent papers (e.g., see [25] and [26]). Most RTP models ignore the thermal inertia of the furnace, since RTP processes are designed to minimize its impact on the process. The furnace thermal inertia is an important factor in the annealing process. The furnace thermal inertia increases the dynamical order of the system, making it more difficult to control, and is taken into account in the simulations below.

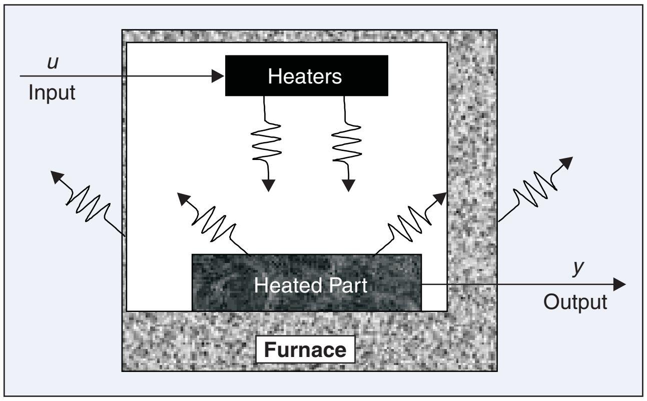


Figure 3. *RTP process overview.*

The thermal processing setup is illustrated schematically in Fig. 3. A controlled heater lamp heats the part in the furnace and the furnace chamber. The heat transfer mechanisms include the linear effects of convection and diffusion and nonlinear irradiation between the heater and the furnace, the part and the furnace, as well as the furnace and the environment.

A simple nonlinear continuous-time model for such thermal processing has two states: furnace temperature T_F and part temperature T_P:

$$\dot{T}_F = b_u u - c_1\left(T_F^4 - T_P^4\right) - c_2(T_F - T_A), \qquad (36)$$

$$\dot{T}_P = c_3\left(T_F^4 - T_P^4\right), \qquad (37)$$

where u is the control input (heating intersity), the system output $y = T_P$ is the part temperature that is assumed to be directly measured, and T_A is the ambient temperature. The following values of the parameters were assumed in the simulation: $b_u = 1000$, $c_1 = 1.1 \cdot 10^{-10}$, $c_2 = 0.8$, and $c_3 = 1.5 \cdot 10^{-9}$. The control goal is to ramp the part temperature from $T_A = 300$ to 600 °C in 2 min, maintain it at 600 °C for 3 min, then ramp the temperature up to 900 °C in 2 min, stay there for 2 min, and finally, ramp the temperature down to 600 °C in 2 min and stay there for 2 min.

In simulation, system (36) was subject to sampled-time control with the sampling time $T_s = 1.2$ s. The sampled time feedback controller was set up as a PID controller with gain scheduling to compensate for more than a sixfold process gain change within a batch run. Let $e_y(t) = y(t) - r(t)$ be the temperature profile tracking error: the difference between the part temperature $y(t) = T_P(t)$ and the reference temperature $r(t)$. The feedback control was computed as

$$u_{fb}(t) = -g(t) \times\left(K_P + \frac{1}{1-z^{-1}}K_I + (1-z^{-1})K_D\right)e_y(t) \qquad (38)$$

$$g(t) = (T_{\max} - T_{\min})/(r(t) - T_{\min}), \qquad (39)$$

where t is the discrete local time within a batch run, the discrete Laplace variable z^{-1} can be interpeted as a unit delay operator, $g(t)$ is the gain scheduling factor, and the

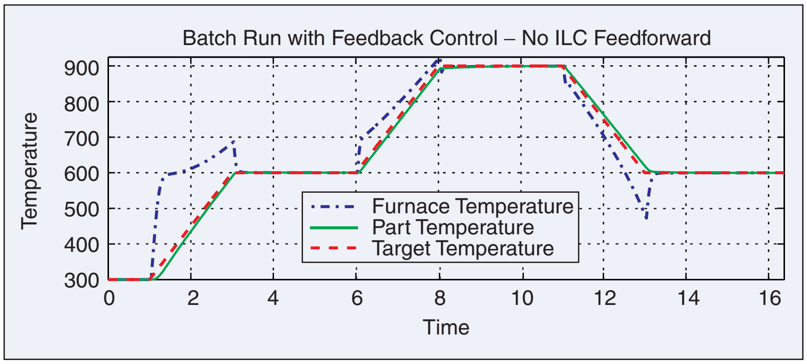


Figure 4. *Reference temperature tracking with feedback control only.*

controller parameters were chosen as follows: $K_P = 0.02$, $K_I = 0.0002$, $K_D = 0.5$, $T_{\min} = 160$, and $T_{\max} = 900$.

Fig. 4 illustrates the results obtained with feedback control (39). The part temperature (solid line) generally follows the target temperature profile $r(t)$ in the batch process—a piecewise linear function. To achieve this, the furnace temperature (dash-dotted line) exibits much larger swings. The part temperature tracking error can exceed 15° over significant intervals. This feedback control accuracy is not good enough for the problem. To improve this error, an ILC feedforward update is implemented.

The feedforward action in the ILC controller was set up as a modification of the reference trajectory $r(t)$ for the PID controller (38), (39):

$$r(t) = T_d(t) + u_{ff}(t), \tag{40}$$

where $T_d(t)$ is the fixed target temperature profile in the RTP process and $u_{ff}(t)$ is the feedforward sequence modified by the ILC controller. The manipulated variables $u_{ff}(t)$ for the batch are collected in a vector U and the control variables (temperature profile tracking errors) in a vector Y, as follows:

$$\begin{aligned} U &= [u_{ff}(1) \ldots u_{ff}(N)]^T, \\ Y &= [y(1) - r(1) \ldots y(N) - r(N)]^T. \end{aligned} \tag{41}$$

LQ/LTR ILC Design

The process described by (36)-(40) is highly nonlinear. At the same time, the LQ/LTR ILC design approach presented in the previous section uses an LTI model of the form (2), (5) to relate vectors (41). This approach allows designing an ILC update that is robust to the model uncertainty and thus can be applied to a nonlinear process.

To illustrate robustness requirements in the thermal processing application, consider linearized pulse responses of the process (36)-(40) to small perturbations of control applied at different times in a batch run. These pulse responses correspond to different columns of matrix G in (2) and are illustrated in Fig. 5 (upper plot). The gain and time constant of these closed-loop responses with the PID controller differ by a factor of four or more. The controller is designed for an "average" pulse response of the closed-loop process, and its robustness is chosen such that it is capable of handling the significant model variation caused by the nonlinearity. The lower plot in Fig. 5 illustrates the frequency domain uncertainty associated with these responses. The solid line corresponds to the average frequency response, and the dashed lines show deviation of individual responses from the average.

The LQ/LTR ILC design was performed as described in the previous section. The initial error amplitude estimate in (19) was chosen to be uniform across the modes $\left|\tilde{Y}_j^0\right| \le y_0 = 0.25$. This roughly corresponds to $N^{-1/2}Y_{\max}$, where

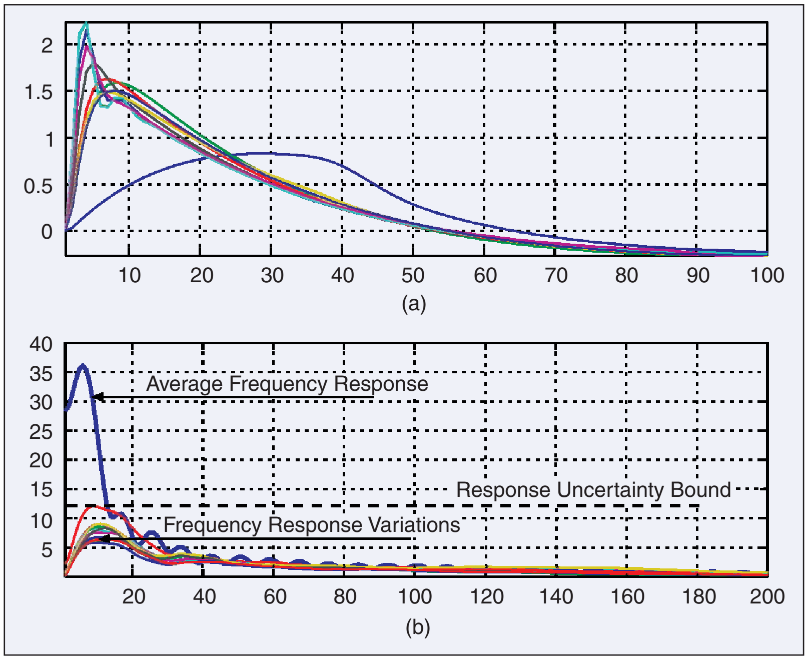


Figure 5. *(a) Pulse response and (b) frequency response uncertainty.*

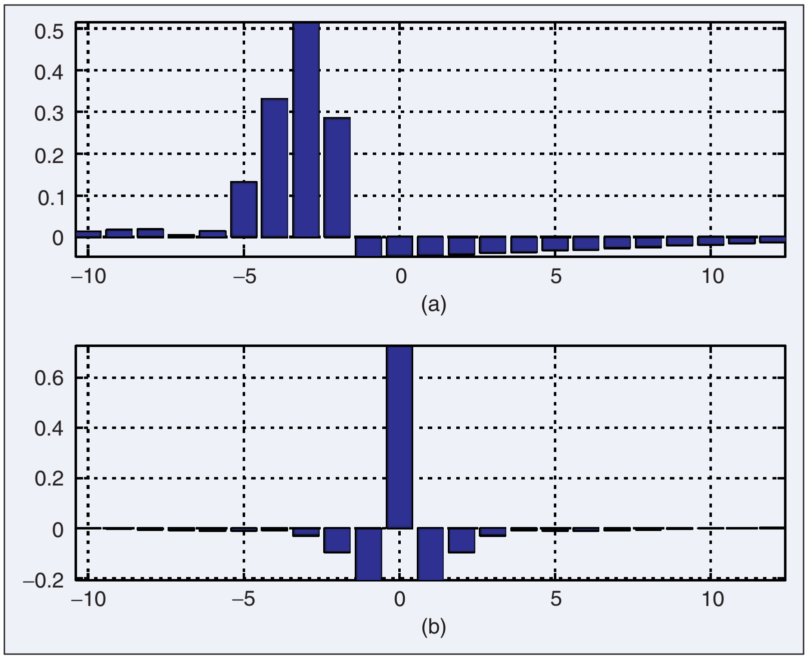


Figure 6. *Designed convolution windows for an LQ/LTR ILC controller: (a) decoupling window and (b) smoothing window.*

$N = 800$ is the number of samples in the run and $Y_{\max} = 7$ is the average amplitude of the tracking error in Fig. 4. In accordance with Fig. 5, a 30% uncertainty in (14), (26), $g_0 = 0.3$, was assumed. The approximation error (31) of implementing the controller was assumed to be $\sigma_1 = 0.15$. This error is related to the FIR convolution window implementation of the designed circulant controller matrices and was evaluated numerically.

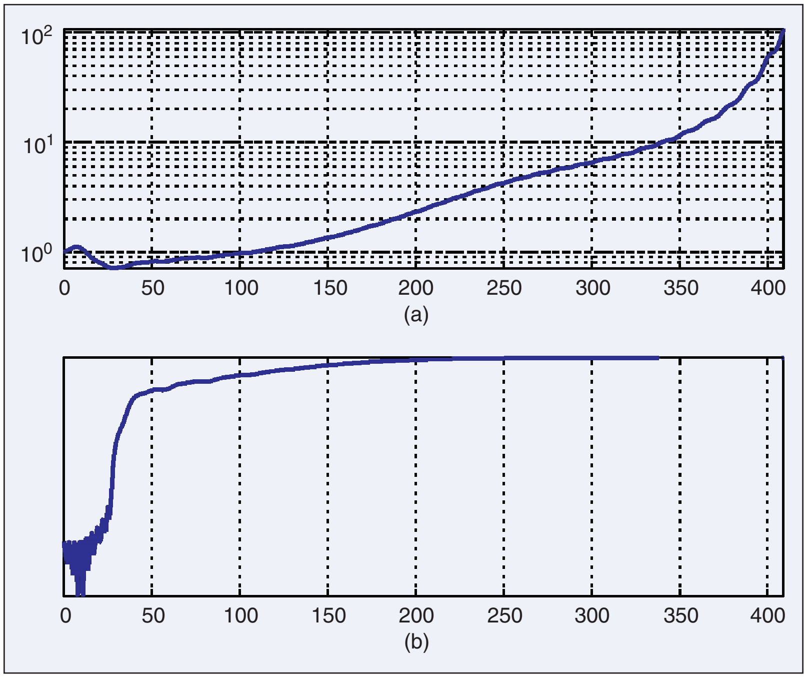


Figure 7. *Nominal performance. Disturbance attenuation gain for LQ/LTR ILC controller: (a) diagonal uncertainty margin and (b) steady-state disturbance attenuation.*

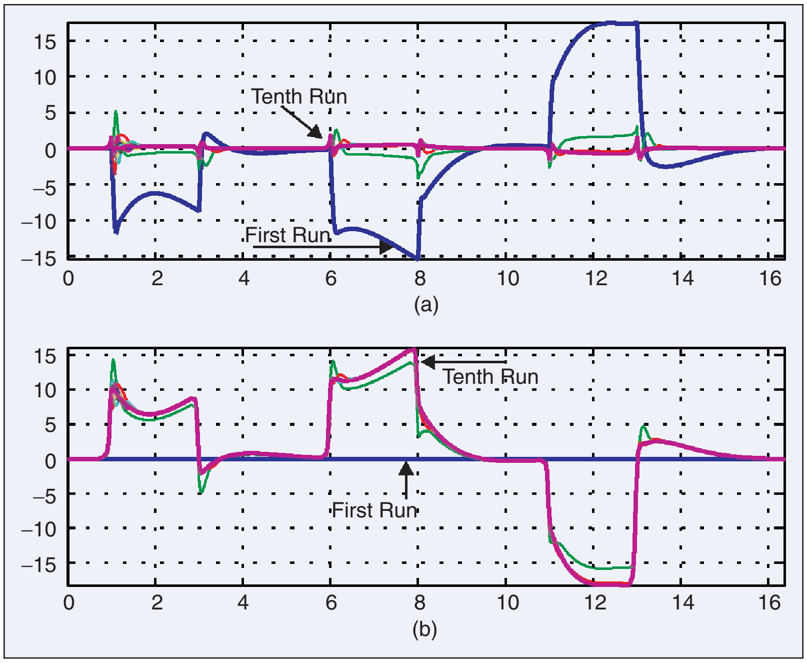


Figure 8. *(a) Tracking error and (b) ILC feedforward control in the first ten ILC updates.*

The designed ILC update (4) is very simple and practical to implement. It computes an update of a feedforward sequence between batch runs by applying noncausal finite window convolution operators A and B to the tracking error and feedforward sequences used in the previous iteration. The designed operators are illustrated in Fig. 6: A in the upper plot and B in the lower plot.

The designed LQ/LTR ILC update can be analyzed by observing its steady-state performance characteristics as defined by the disturbance attenuation multiplier $h_j = |b_j| / |a_j g_j + b_j|$ in (21). The lower plot in Fig. 7 illustrates the disturbance attenuation gain h_j computed for the designed ILC controller. The gain h_j in Fig. 7 is plotted versus the local-time frequency $\nu_j = 2\pi j / N$. The disturbance attenuation gain achieved by the LQ/LTR ILC controller is uniformly small for lower local-time frequencies ν_j and increases, approaching one for higher frequencies outside of the plant bandwidth, where the gain g_j is small. The upper plot in Fig. 7 shows the uncertainty margin, the left-hand side in (27), achieved with the designed controller. Its lowest value is about 0.7, which ensures robust converegence in accordance with the design specifications.

The closed-loop simulations with the designed LQ/LTR ILC controller are illustrated in Fig. 8. The control update converges rapidly after four or five runs, and the maximal error of less than one degree is achieved (Fig. 8(a)). The feedforward signal, the temperature set-point sequence for the PID controller, is shown in Fig. 8(b). Only small and fairly smooth modification of the set point is required to achieve this outstanding accuracy.

In [27], the described approach is compared against a more traditional design of the ILC update. In particular, a PD ILC controller was designed and tested for the same process. To ensure robust convergence of the ILC iterations, this PD controller has a “relaxation” term as suggested in [4], [18], [28]. A comparative analysis of performance and robustness for the robust PD update with the “relaxation” term (R-PD ILC) against the LQ/LTR ILC controller is performed in [27]. The comparison shows that R-PD ILC has a much larger tracking error compared to the LQ/LTR ILC if their robustess is made comparable. The convergence speed for R-PD is also much slower, by a factor of 20.

Conclusions

This article has presented consistent methods for the analysis and design of ILC update for batch processes. The anaylsis results were formulated in the frequency domain. The engineering approach to robust LQ loop-shaping design of the ILC controller demonstrated herein is based on formal design specifications formulated in the frequency domain and including controller performance, robust stability margin, and actuator move maginitude constraints. The controller design uses a process FIR pulse response model and pulse response uncertainty estimate. The designed ILC update is implementable through convolution window operators that are noncausal in the local time of a batch. In the design process, penalty weights in the LQ problem are adjusted to recover robustness margins and other specifications for the closed loop, which is similar to the LQG/LTR approach for the design of dynamical controllers. The presented LQ/LTR ILC design methods can be illustrated using intuitive graphical diagrams and are based on industrial FIR models of the controlled plant.

The controller was demonstrated in an application to batch thermal processing. Outstanding accuracy, robustness to nonlinearity, and convergence speed were demonstrated in simulation for this difficult, highly nonlinear process application.

Acknowledgment

The author gratefully acknowledges Jan Jelinek and Anoop Mathur of Honeywell Laboratories in Minneapolis for useful discussions and for providing the thermal processing simulation model for this work.

References

[1] Y. Chen. (1999). *A Bibliographical Library for Iterative Learning Control (ILC) Research* [Online]. Available: http://cicserver.ee.nus.edu.sg/~ilc/ILC/ilcref.html

[2] K.L. Moore, *Iterative Learning Control for Deterministic Systems*. New York: Springer-Verlag, 1993.

[3] *Int. J. Control* (Special Issue on Iterative Learning Control), vol. 73, no. 10, 2000.

[4] S. Arimoto, "Learning control theory for robotic motion," *Int. J. Adapt. Control Signal Process.*, vol. 4, pp. 453-564, 1990.

[5] F. Padiew and R. Su, "An H_∞ approach to learning control systems," *Int. J. Adapt. Control Signal Process.*, vol. 4, pp. 465-474, 1990.

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

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

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

[9] W. Messner, R. Horowitz, W.-W. Kao, and M. Boals, "A new adaptive learning rule," *IEEE Trans. Automat. Contr.*, vol. 36, no. 2, pp. 188-197, 1991.

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

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

[12] K.M. Tao, R.L. Kosut, M. Ekblad, and G. Aral, "Feedforward learning applied to RTP of semiconductor wafers," in *Proc. 33th IEEE Conf. Decision and Control*, Lake Buena Vista, FL, Dec. 1994, pp. 67-72.

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

[14] K. Gugliemo and N. Sadeh, "Experimental evaluation of a new robot learning controller," in *Proc. IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, Apr. 1991, pp. 734-739.

[15] A.N. Tikhonov and V.Ya. Arsenin, *Solutions of Ill-Posed Problems*. Washington, D.C.: Halsted, 1977.

[16] D. Gorinevsky, "On regularized feedback update of distributed-parameter systems in control and nonlinear estimation applications," in *Proc. American Control Conf.*, vol. 3, Albuquerque, NM, June 1997, pp. 1823-1827.

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

[18] S. Arimoto, T. Naniwa, and H. Suzuki, "Robustness of P-type learning control with a forgetting factor for robotic motions," in *Proc. 29th IEEE Conf. Decision and Control*, Honolulu, HA, Dec. 1990, pp. 2640-2645.

[19] B. Bamieh, F. Paganini, and M. Dahleh, "Distributed control of spatially-invariant systems," *IEEE Trans. Automat. Contr.*, vol. 47, pp. 1091-1107, July 2002.

[20] G.E. Stewart, D. Gorinevsky, and G.A. Dumont, "Design of a practical robust controller for a sampled distributed parameter system," in *Proc. 37th IEEE Conf. Decision and Control*, Tampa, FL, Dec. 1998, pp. 3156-3161.

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

[22] G. Stein and M. Athans, "The LQG/LTR procedure for multivariable feedback control design," *IEEE Trans. Automat. Contr.*, vol. 32, no. 2, pp. 105-114, 1987.

[23] R.M. Gray. (1999). *Toeplitz and Circulant Matrices: A Review* [Online]. Available: http://www.isl.stanford.edu/~gray/toeplitz.pdf

[24] A.V. Oppenheim, R.W. Schafer, and J.R. Buck, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[25] J.L. Ebert, A. Emami-Naeini, H. Aling, and R.L. Kosut, "Thermal modeling and control of rapid thermal processing systems," in *Proc. 34th IEEE Conf. Decision and Control*, New Orleans, LA, Dec. 1995, pp. 1304-1309.

[26] Y.M. Cho and P. Gyugyi, "Control of rapid thermal processing: A system theoretic approach," *IEEE Trans. Contr. Syst. Technol.*, vol. 5, no. 6, pp. 644-653, 1997.

[27] D.M. Gorinevsky, "Distributed system loopshaping design of iterative control for batch processes," in *Proc. 38th IEEE Conf. Decision and Control*, Phoenix, AZ, Dec. 1999, pp. 203-208.

[28] D.H. Owens, N. Amann, and E. Rogers, "Iterative learning control-an overview of recent algorithms," *Appl. Math. Comput. Sci.*, vol. 5, no. 3, pp. 425-438, 1995.

Dimitry Gorinevsky is a senior staff scientist with Honeywell Global Controls Laboratory, Cupertino, CA. He received his Ph.D. from Moscow Lomonosov University. He was with the Russian Academy of Sciences in Moscow, an Alexander von Humboldt Fellow in Munich, and with the University of Toronto. Before joining Honeywell Labs he was with Honeywell-Measurex and was an adjunct professor of electrical engineering at the University of British Columbia, Vancouver, Canada. His interests are in advanced information and control systems applications across many industries. He has authored a book and more than 110 reviewed technical papers. He is an associate editor of *IEEE Transactions on Control Systems Technology* and a Senior Member of the IEEE.