2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Weighting Matrix Design for Robust Monotonic Convergence in Norm Optimal Iterative Learning Control

Douglas A Bristow

Abstract—**In this paper we examine the robustness of Norm Optimal ILC with quadratic cost criterion for discrete-time, linear time-invariant, single-input single-output systems. A bounded multiplicative uncertainty model is used to describe the uncertain system and a sufficient condition for robust monotonic convergence is developed. We find that, for sufficiently large uncertainty, the performance weighting can not be selected arbitrarily large, and thus overall performance is limited. To maximize available performance, a time-frequency design methodology is presented to shape the weighting matrix based on the initial tracking error. The design is applied to a nanopositioning system and simulation results are presented.**

I. Introduction

ITERATIVE learning control (ILC) [1]-[3] is used to improve the performance of systems that repeat the same operation many times. ILC uses the tracking errors from previous iterations of the repeated motion to generate a feedforward control signal for subsequent iterations. Convergence of the learning process results in a feedforward control signal that is customized for the repeated motion, yielding very low tracking error.

One popular ILC design method for discrete-time, linear time-invariant (LTI) systems is Norm Optimal ILC [4]-[8], analogous to linear quadratic optimal control for feedback systems. Norm Optimal ILC has successfully been applied to industrial robots [6], wafer steppers [9], and video projection systems [10]. Norm Optimal ILC is an attractive design framework because the designer is supplied with several intuitive tuning knobs in the form of weighting matrices, optimal solutions to the quadratic cost function are straightforward and elegant, and solutions always result in monotonic convergence of the control [4]. Previous work has shown how weighting functions can be designed to prevent actuator saturation [6], decrease noise sensitivity [5], and emphasize certain frequency bands in the learning [7].

One critical limitation of Norm Optimal ILC, however, is that perfect models of the system are assumed. When the model is inaccurate, monotonic convergence of the control is no longer guaranteed. Learning transients including large, rapid growth of the error [11] or even instability can occur, potentially damaging equipment or causing injury. In the first part of this work we examine the monotonic convergence robustness of the optimal solution for bounded frequency-domain model uncertainty. A sufficient condition for robust monotonic convergence is developed, which creates limitations on the weighting matrices. In the second part of this paper we consider the problem of optimally selecting weighting matrices to maximize nominal performance, while maintaining worst-case robustness. We present a design methodology using time-frequency analysis to identify the most challenging portions of the trajectory and design weightings to focus on those portions.

The remainder of this paper is organized as follows. In Section II we present the Norm Optimal ILC for nominal systems. Uncertain systems and robust monotonic convergence are presented in Section III. Our design methodology is developed in Section IV. In Section V we apply our methodology to the design of a robust Norm Optimal ILC for a nanopositioning system. Conclusions are given in Section VI.

II. Norm Optimal ILC

In this work we will consider single-input single-output (SISO) systems for simplicity, although results can be extended to multi-input multi-output systems. Let q be the forward time-shift operator $\mathrm{q}x(k)=x(k+1)$ and q^{-1} be the backward time-shift operator. Consider the discrete-time SISO LTI dynamic system,

$$y_j(k)=P(\mathrm{q})u_j(k)+d(k), \tag{1}$$

where k is the time index, j is the iteration index, y is the output, u is the control input, and d is an iteration-invariant disturbance. $P(\mathrm{q})$ is assumed to be asymptotically stable and can be written as

$$P(\mathrm{q})=p_m\mathrm{q}^{-m}+p_{m+1}\mathrm{q}^{-m-1}+p_{m+2}\mathrm{q}^{-m-2}+\dots, \tag{2}$$

where $p_m\neq 0$ and $\{p_m,p_{m+1},p_{m+2}\dots\}$ is the system impulse response. The system delay, or equivalently relative degree, is given by $m\geq 0$. Repeating disturbances [12], repeated nonzero initial conditions [11], and systems augmented with feedback and feedforward control [12] can be captured in $d(k)$.

Let $y_d(k)$ be the N-sample length desired output for $k=m+1,m+2,\dots,m+N$ and consider the vector-description of signals,

This work was supported in part by the University of Illinois at Urbana-Champaign Nano-CEMMS center NSF Award #0328162.

D.A. Bristow is with the Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO.

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

$$\mathbf{y}_d = \left[y_d(m+1,j) \ \cdots \ y_d(m+N,j) \right]^T$$
$$\mathbf{y}_j = \left[y(m+1,j) \ \cdots \ y(m+N,j) \right]^T,$$
$$\mathbf{u}_j = \left[u(1,j) \ \cdots \ u(N,j) \right]^T,$$
$$\mathbf{d} = \left[d(m+1,j) \ \cdots \ d(m+N,j) \right]^T.$$

The lifted system [13] description is given by $\mathbf{y}_j = \mathbf{P}\mathbf{u}_j + \mathbf{d}$, where,

$$\mathbf{P} = \begin{bmatrix} p_m & 0 & 0 & \cdots & 0 \\ p_{m+1} & p_m & 0 & \ddots & \vdots \\ p_{m+2} & p_{m+1} & p_m & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ p_{m+N} & \cdots & p_{m+2} & p_{m+1} & p_m \end{bmatrix}. \tag{3}$$

The tracking error is given by,

$$\mathbf{e}_j = \mathbf{y}_d - \mathbf{y}_j, \tag{4}$$

or,

$$\mathbf{e}_j = -\mathbf{P}\mathbf{u}_j + \mathbf{e}_0, \text{ for } j \ge 1, \tag{5}$$

where $\mathbf{e}_0 = \mathbf{y}_\mathbf{d} - \mathbf{d}$.

The Norm Optimal ILC algorithm is to choose the control for the next iteration as the solution to $\arg\min_{\mathbf{u}_{j+1}} \mathbf{J}_{j+1}$, where

$$\mathbf{J}_{j+1} = \mathbf{e}_{j+1}^T \mathbf{W}_\mathbf{e} \mathbf{e}_{j+1} + \mathbf{u}_{j+1}^T \mathbf{W}_\mathbf{u} \mathbf{u}_{j+1} \tag{6}$$

and $\mathbf{W}_\mathbf{e} > 0$, $\mathbf{W}_\mathbf{u} \ge 0$. From (5), (6),

$$\begin{aligned}\mathbf{J}_{j+1} &= \left(-\mathbf{P}\mathbf{u}_{j+1} + \mathbf{e}_0\right)^T \mathbf{W}_\mathbf{e} \left(-\mathbf{P}\mathbf{u}_{j+1} + \mathbf{e}_0\right) + \mathbf{u}_{j+1}^T \mathbf{W}_\mathbf{u} \mathbf{u}_{j+1} \\ &= \mathbf{u}_{j+1}^T \left(\mathbf{P}^T \mathbf{W}_\mathbf{e} \mathbf{P} + \mathbf{W}_\mathbf{u}\right) \mathbf{u}_{j+1} - 2\mathbf{u}_{j+1}^T \mathbf{P}^T \mathbf{W}_\mathbf{e} \mathbf{e}_0 + \mathbf{e}_0^T \mathbf{W}_\mathbf{e} \mathbf{e}_0.\end{aligned}$$

The optimal control obtained from $\partial \mathbf{J}_{j+1} / \partial \mathbf{u}_{j+1} = 0$ is given by,

$$\mathbf{u}_{j+1} = \left(\mathbf{P}^T \mathbf{W}_\mathbf{e} \mathbf{P} + \mathbf{W}_\mathbf{u}\right)^{-1} \mathbf{P}^T \mathbf{W}_\mathbf{e} \mathbf{e}_\mathbf{0}. \tag{7}$$

In the following we replace $\mathbf{P}$ by $\hat{\mathbf{P}}$, to emphasize that the learning algorithm is constructed using the system model $\hat{\mathbf{P}}$, rather than the actual system $\mathbf{P}$. Using the system model (5), we can reformulate (7) into the first-order algorithm (similar to [6]),

$$\mathbf{u}_{j+1} = \mathbf{Q}\left(\mathbf{u}_j + \mathbf{L}\mathbf{e}_j\right), \tag{8}$$

where

$$\mathbf{L} = \hat{\mathbf{P}}^{-1}, \tag{9}$$

is the learning function and,

$$\mathbf{Q} = \left(\hat{\mathbf{P}}^T \mathbf{W}_\mathbf{e} \hat{\mathbf{P}} + \mathbf{W}_\mathbf{u}\right)^{-1} \hat{\mathbf{P}}^T \mathbf{W}_\mathbf{e} \hat{\mathbf{P}}, \tag{10}$$

is the robustifying Q-filter [3]. Thus, we find that the optimal solution is a Q-filtered model-inversion ILC where the weighting matrices are used to shape the Q-filter.

The converged control $\mathbf{u}_\infty \triangleq \lim_{j\to\infty} \mathbf{u}_j$ can be found from (5), (8) as

$$\mathbf{u}_\infty = \left(\hat{\mathbf{P}}^T \mathbf{W}_\mathbf{e} \mathbf{P} + \mathbf{W}_\mathbf{u}\right)^{-1} \hat{\mathbf{P}}^T \mathbf{W}_\mathbf{e} \mathbf{e}_0. \tag{11}$$

The converged error $\mathbf{e}_\infty \triangleq \lim_{j\to\infty} \mathbf{e}_j$ can then be found from (5) as,

$$\mathbf{e}_\infty = \left(\mathbf{I} - \mathbf{P}\left(\hat{\mathbf{P}}^T \mathbf{W}_\mathbf{e} \mathbf{P} + \mathbf{W}_\mathbf{u}\right)^{-1} \hat{\mathbf{P}}^T \mathbf{W}_\mathbf{e}\right) \mathbf{e}_0. \tag{12}$$

An important consideration for safe operation of ILC are the learning transients of the system [11]. In some cases, stable ILC algorithms can generate large control signals for many iterations before convergence [11]. Therefore, we are interested in monotonic convergence of the ILC as defined by the following.

Definition: The ILC is monotonically convergent if $\mathbf{u}_j$ converges and $\left\|\mathbf{u}_\infty - \mathbf{u}_{j+1}\right\|_2 < \left\|\mathbf{u}_\infty - \mathbf{u}_j\right\|_2$.

Remark: Monotonic convergence of $\left\|\mathbf{u}_\infty - \mathbf{u}_j\right\|_2$ does not necessarily give monotonic convergence of the system error, although $\left\|\mathbf{u}_\infty - \mathbf{u}_j\right\|_2$ bounds the error growth. For example, $\left\|\mathbf{e}_\infty - \mathbf{e}_j\right\|_\infty \le \left\|\mathbf{e}_\infty - \mathbf{e}_j\right\|_2 \le \bar{\sigma}\left(\hat{\mathbf{P}}\right)\left\|\mathbf{u}_\infty - \mathbf{u}_j\right\|_2$, where $\bar{\sigma}(\bullet)$ is the maximum singular value of $(\bullet)$.

As a special case of the result we present in Section III.A, we find that Optimal ILC is monotonically convergent when the system dynamics are accurately modeled as $\hat{\mathbf{P}}$. However, when the model is inaccurate, monotonic convergence may be lost, which is examined in the following section.

III. Robustness of the Optimal Solution

Consider model uncertainty such that the actual system $\mathbf{P}$ belongs to a set of perturbations of the nominal system given by

$$\mathbf{P} = \hat{\mathbf{P}}\left(\mathbf{I} + \mathbf{W}_\Delta \mathbf{\Delta}\right), \tag{13}$$

where $\hat{\mathbf{P}}$, $\mathbf{W}_\Delta$ are known lower-triangular Toeplitz *nxn* matrices and $\mathbf{\Delta}$ is an unknown lower-triangular Toeplitz *nxn* matrix with $\bar{\sigma}(\mathbf{\Delta}) \le 1$. The class of lifted systems (13) includes systems whose frequency-domain description is given by [14],

$$P(z) = \hat{P}(z)\left(1 + W_\Delta(z)\Delta(z)\right), \tag{14}$$

where $\mathbf{W}_\Delta$ is the lifted system description of $W_\Delta(z)$, and $\left\|\Delta(z)\right\|_\infty \le 1$. This description is useful for capturing models with parametric uncertainties and unmodeled dynamics [15].

Robustness results for Q-filtered model-inversion ILC are given in [14]. The following theorem is adapted from [14] for the Norm Optimal solution in (8), (9), (10). This result is the first of its kind (to the author's knowledge) that provides explicit constraints on the weighting matrices for robustness.

Theorem 1: If

$$\bar{\sigma}\left(\left(\mathbf{W_u}+\hat{\mathbf{P}}^T\mathbf{W_e}\hat{\mathbf{P}}\right)^{-1}\hat{\mathbf{P}}^T\mathbf{W_e}\hat{\mathbf{P}}\mathbf{W}_\Delta\right)<1. \quad (15)$$

then the ILC is robustly monotonic for the class of uncertainties (13).

Proof: From (11), (8), and (12),

$$\mathbf{u}_\infty-\mathbf{u}_{j+1}=\mathbf{Q}\left(\mathbf{I}-\mathbf{LP}\right)\left(\mathbf{u}_\infty-\mathbf{u}_j\right),$$

$$\left\|\mathbf{u}_\infty-\mathbf{u}_{j+1}\right\|_2\le\bar{\sigma}\left(\mathbf{Q}\left(\mathbf{I}-\mathbf{LP}\right)\right)\left\|\mathbf{u}_\infty-\mathbf{u}_j\right\|_2.$$

Then, the system is robustly monotonic if $\bar{\sigma}\left(\mathbf{Q}\left(\mathbf{I}-\mathbf{LP}\right)\right)<1$. Substituting (9) and (10) for L and Q, respectively, we arrive at (15). □

A. Robustness of Optimal ILC to Small Uncertainty

When the uncertainty is bounded by $\bar{\sigma}\left(\mathbf{W}_\Delta\right)<1$, then the Optimal ILC is robustly monotonic for all choices of weightings because

$$\begin{aligned}&\bar{\sigma}\left(\left(\mathbf{W_u}+\hat{\mathbf{P}}^T\mathbf{W_e}\hat{\mathbf{P}}\right)^{-1}\hat{\mathbf{P}}^T\mathbf{W_e}\hat{\mathbf{P}}\mathbf{W}_\Delta\right)\\&<\bar{\sigma}\left(\left(\mathbf{W_u}+\hat{\mathbf{P}}^T\mathbf{W_e}\hat{\mathbf{P}}\right)^{-1}\hat{\mathbf{P}}^T\mathbf{W_e}\hat{\mathbf{P}}\right)\\&\le 1,\end{aligned}$$

where the last step is due to positive semidefiniteness of $\mathbf{W_u}$ and positive definiteness of $\mathbf{W_e}$. Note that the uncertainty bound $\bar{\sigma}\left(\mathbf{W}_\Delta\right)<1$ is equivalent [12] to the $\mathcal{H}_\infty$ bound $\left\|W_\Delta(z)\right\|_\infty<1$ for the frequency-domain description (14).

B. Existence of Robustifying Weightings

In the following we show that there always exists weighting matrices that satisfy the robustness condition (15). Consider $\bar{\sigma}\left(\mathbf{W}_\Delta\right)\ge 1$ and the weightings,

$$\mathbf{W_u}=\mathbf{I} \quad (16)$$

and

$$\mathbf{W_e}=c\mathbf{I},\ c>0 \quad (17)$$

Then,

$$\begin{aligned}&\bar{\sigma}\left(\left(\mathbf{W_u}+\hat{\mathbf{P}}^T\mathbf{W_e}\hat{\mathbf{P}}\right)^{-1}\hat{\mathbf{P}}^T\mathbf{W_e}\hat{\mathbf{P}}\mathbf{W}_\Delta\right)\\&=\bar{\sigma}\left(\left(\mathbf{I}+c\hat{\mathbf{P}}^T\hat{\mathbf{P}}\right)^{-1}c\hat{\mathbf{P}}^T\hat{\mathbf{P}}\mathbf{W}_\Delta\right)\\&=\bar{\sigma}\left(\left(\tfrac{1}{c}\hat{\mathbf{P}}^{-1}\hat{\mathbf{P}}^{-T}+\mathbf{I}\right)^{-1}\mathbf{W}_\Delta\right)\\&\le\frac{1}{\underline{\sigma}\left(\tfrac{1}{c}\hat{\mathbf{P}}^{-1}\hat{\mathbf{P}}^{-T}+\mathbf{I}\right)}\bar{\sigma}\left(\mathbf{W}_\Delta\right)\\&\le\frac{\bar{\sigma}\left(\mathbf{W}_\Delta\right)}{\tfrac{1}{c}\underline{\sigma}\left(\hat{\mathbf{P}}^{-1}\hat{\mathbf{P}}^{-T}\right)+1},\end{aligned}$$

and (15) is satisfied for,

$$c<\frac{\underline{\sigma}\left(\hat{\mathbf{P}}^{-1}\hat{\mathbf{P}}^{-T}\right)}{\bar{\sigma}\left(\mathbf{W}_\Delta\right)-1}, \quad (18)$$

Thus, it is always possible to pick c small enough to be robust.

In some cases it may not be possible or feasible to obtain explicit bounds on the model uncertainty. In light of (18), a reasonable tuning strategy is to begin with a very small value of c and slowly increasing it while watching the system response for signs of instability or transient growth. When the uncertainty bound is known or can be estimated, the design strategy presented in the following section can be utilized.

IV. Time-Frequency Design of the Weighting Matrices

For design, it suffices to set $\mathbf{W_u}=\mathbf{I}$ and design $\mathbf{W_e}$. Typically, a scalar weighting $c\mathbf{I}$ is used for $\mathbf{W_e}$ and the scalar c is selected by system tuning [5],[6]. As we show in the previous section, it is always possible to find a sufficiently small c to provide robustness. However, for high performance, we would like c to be as large as possible, and thus we might consider selecting a c close to the bound (18).

A. Time-Varying Weighting

The scalar weighting c represents a uniform performance weighting across every time sample in the iteration. However, in many applications, especially in precision motion control [16], performance *challenges* are not uniform across the entire iteration. That is, some portions of the iteration (for instance, those where references and disturbances contain low frequencies) may naturally result in low tracking error with a small weighting. We refer to these low frequency sections as β-segments [16]. Other portions, referred to as α-segments, may contain very high frequencies and naturally result in large errors. α-segments could be the result of rapid changes in the reference signal or nonsmooth, nonlinear disturbances such as friction and backlash. For α-segments, a much larger weighting will be necessary to provide low error. For these applications, we are interested in designing a weighting matrix of the form,

$$\mathbf{W_e}=diag\left\{w(1),w(2),\ldots,w(N)\right\}, \quad (19)$$

which we interpret as a time-varying weighting, since $w(k)$ is the weight on the error at time k.

Our goal is to design $\mathbf{W_e}$ to provide the best performance for the nominal plant, while maintaining robustness to the class of uncertainties (13). Formally, we wish to solve the optimal design problem,

$$\underset{w(1),\ldots,w(N)}{\arg\min}\left\|\mathbf{e}_\infty\big|_{\mathbf{P}=\hat{\mathbf{P}}}\right\|_2 \text{ such that (15)}, \quad (20)$$

where,

$$\mathbf{e}_\infty\big|_{\mathbf{P}=\hat{\mathbf{P}}} = \left(\mathbf{I} - \hat{\mathbf{P}}\left(\hat{\mathbf{P}}^T\mathbf{W}_\mathbf{e}\hat{\mathbf{P}} + \mathbf{I}\right)^{-1}\hat{\mathbf{P}}^T\mathbf{W}_\mathbf{e}\right)\mathbf{e}_0 . \tag{21}$$

An important feature of this problem is that $\mathbf{e}_\infty$ depends on $\mathbf{e}_0 = \mathbf{y}_d - \mathbf{d}$, and thus is specific to the reference and disturbance. The optimal weightings are therefore customized for the specific tracking challenges in $\mathbf{y}_d$ and $\mathbf{d}$. As we show in Section V, this can lead to significant performance improvements over the scalar weighting, *c*.

B. Design Procedure

The design problem (20) can not be solved analytically, so we look to numerical techniques. For long iteration lengths where *N* is large, the number of free variables in (20) could pose computational difficulties. Here, we present a methodology for shaping a low-order weighting profile,

$$w(k) = f(\xi_1, \ldots, \xi_n, k) \tag{22}$$

where ξ_i are variables to be optimized, and $n \ll N$. The methodology uses the initial tracking error $e_0(k)$ and time-frequency analysis tools to identify temporal locations where weightings should be largest. This approach is similar to previous work by the author [16], although the difference here is that the methodology is applied with robustness constraints, and thus results in a *robust*, high performance ILC design.

To determine where tracking challenges will be largest, we begin the design process by measuring $e_0(k)$ from the 0th iteration. $e_0(k)$ is then decomposed into its time and frequency components using any time-frequency decomposition (eg. short-time Fourier Transform, Wigner-Ville decomposition [17], or wavelets). For illustrative purposes, consider $e_0(k)$ from the nanopositioning example in Section V, which is shown in Figure 1. The Wigner-Ville decomposition is given by

$$W(k,\omega) = \frac{1}{\pi}\sum_{\tau=-N}^{N} e_0(k-\tau)e_0(k+\tau)\exp(-i2\omega\tau), \tag{23}$$

and shown for this signal in Figure 2.

The time-frequency decomposition in Figure 2 shows a high frequency peak at 0.1 seconds followed by lower frequency content at approximately 0.11 seconds. We identify these peaks as α-segments and shape the weighting vector $w(k)$ to permit larger weights during these segments. To do so, we select a shaping function centered at the α-segment t_i with one or more tunable parameters. For example the rectangle function,

$$rect(a,b,t_i,k) = \begin{cases} a, & for\ |k-t_i| \le b \\ 0, & o.w. \end{cases}, \tag{24}$$

where *a* is the function height and 2*b* is its width can serve this purpose. The weighting profile is then constructed as,

$$\begin{aligned} w(k) &= f\left(c, a_1, b_1, \ldots, a_{N_\alpha}, b_{N_\alpha}, k\right) \\ &= c + \sum_{i=1}^{N_\alpha} rect(a_i, b_i, t_i, k), \end{aligned} \tag{25}$$

where *c* is the β-segment weighting. The weighting profile for this example is shown in Figure 3.

To optimally select the shaping parameters a_i, b_i, and *c*, we use a numerical search to minimize $\left\|\mathbf{e}_\infty\big|_{\mathbf{P}=\hat{\mathbf{P}}}\right\|_2$, subject to the robustness constraint (15). The design procedure is summarized in the flowchart in Figure 4.

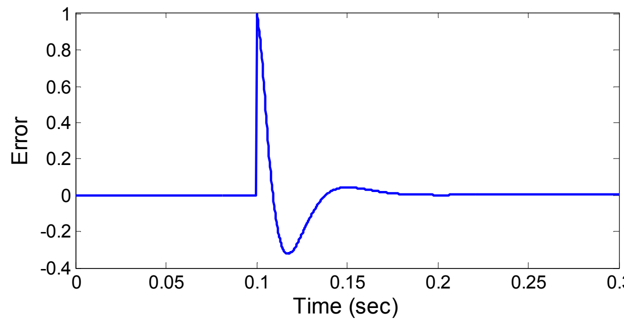

Fig. 1. Initial error, $e_0(k)$.

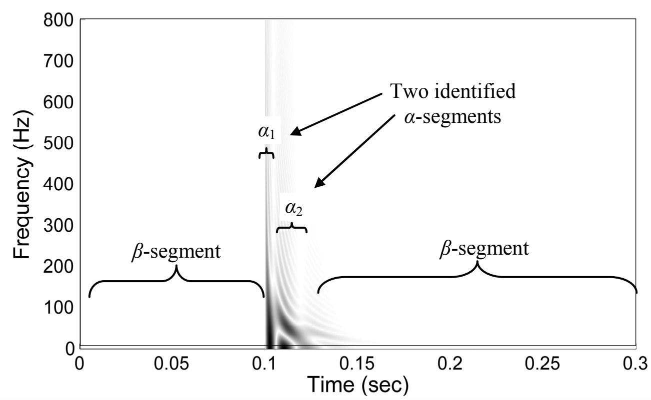

Fig. 2. Wigner-Ville time-frequency decomposition of $e_0(k)$.

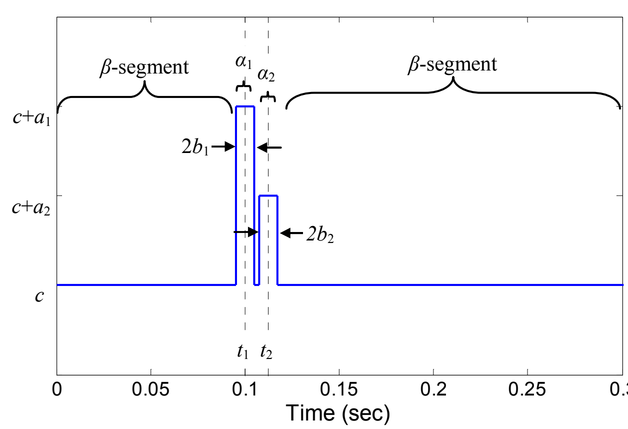

Fig. 3. Parameterized weighting profile, $w(k)$.

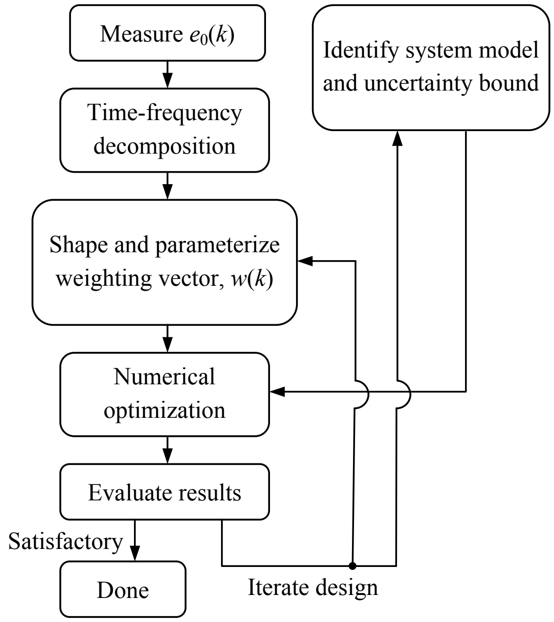

Fig. 4. Time-frequency design process.

V. Numerical Example: Nanopositioning System

In this section we apply the methodology presented in the previous section to the design of an Optimal ILC for a nanopositioning system [18]. The nanopositioning system is operated with a feedback controller, as shown in Figure 5. The sampling rate is 4 kHz and the plant model $G(\mathrm{q})$ and feedback controller $C(\mathrm{q})$ are given in (23) and (24), respectively, at the bottom of the next page. The closed loop transfer function from u to e is given by $\hat{P}(\mathrm{q}) = G(\mathrm{q})/(1+C(\mathrm{q})G(\mathrm{q}))$, and its Bode plot is shown in Figure 6. To illustrate the design approach and robustness limitations on performance, we assume that the model for P has the multiplicative uncertainty bound,

$$W_\Delta(\mathrm{q}) = \frac{1.4\ (\mathrm{q} - 0.9975)}{\mathrm{q} - 0.9656}.$$

A Bode plot of P and W_Δ is shown in Figure 7.

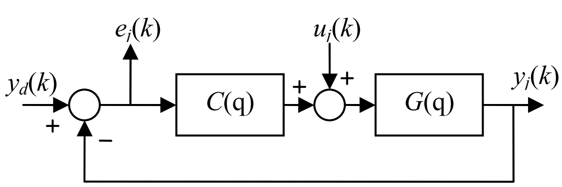

Fig. 5. Nanopositioning system control structure [18].

We choose our reference as a unit step at 0.1 seconds with an iteration length of 0.3 seconds. The initial error and time-frequency decomposition are given in figures 1 and 2, respectively. As illustrated in Figure 3, two rectangular shape functions (24), are selected at t_1=0.1 seconds and t_2=0.11 seconds where the two dominant lobes in the time-frequency decomposition are centered. Rectangular shape functions are used here as the simplest shape function to illustrate the design approach. Other shapes could be selected by the designer and may yield improved performance.

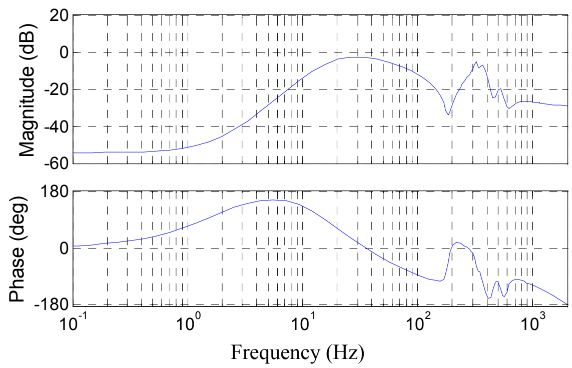

Fig. 6. Bode plot of $\hat{P}(\mathrm{q})$.

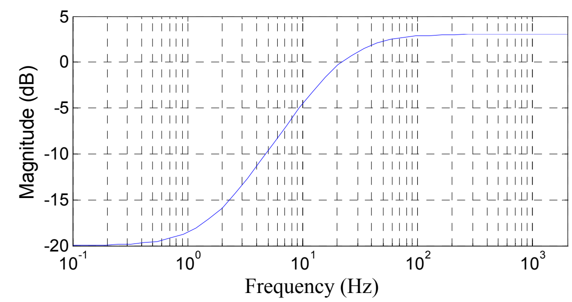

Fig. 7. Bode plot of $W_\Delta(\mathrm{q})$.

A. *Nominal System Performance*

Using MATLAB's `fmincon`, the optimal parameters are found as

$$\begin{array}{lll} a_1 = 178.7 & a_2 = 0 & c = 30.4 \\ b_1 = 80 & b_2 = N/A & \end{array}.$$

Interestingly, the optimization has determined that the second α-segment is not significant, and set the weighting for that segment to 0. For comparison purposes, we also search for the optimal scalar weighting and find it as,

$$c_{scalar} = 67.7 .$$

The resulting weighting vector for each are plotted in Figure 8. The learning algorithm is constructed for both sets of weightings and the tracking performance is simulated using the nominal plant model, $\hat{P}(\mathrm{q})$. In both cases convergence occurs in one iteration because of the perfect model inversion in **L**. The converged error is shown in Figure 9. The performance results are listed in Table 1.

Both weighting vectors are robust to uncertainties bounded by W_Δ. However, the optimization results show that better performance is achieved when the weighting is decreased in the β-segments and reallocated to the first α-segment.

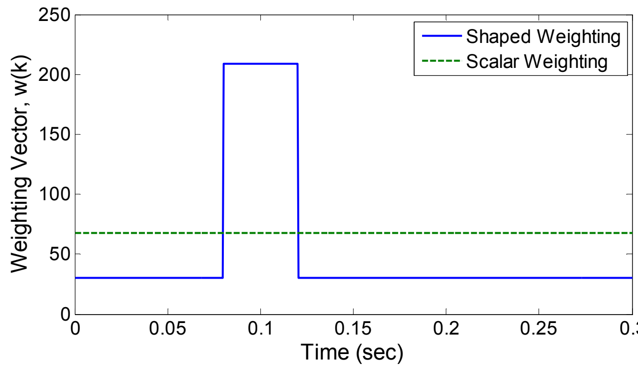

Fig. 8. Optimized weighting vector, $w(k)$.

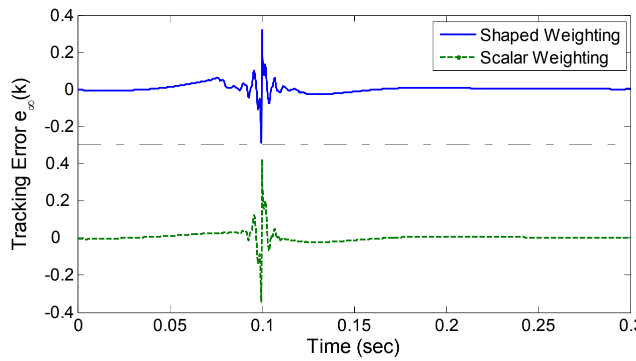

Fig. 9. Converged error $e_\infty(k)$ on the nominal plant $\hat{P}(\mathrm{q})$.

Table 1. Performance results.

	$\lVert \mathbf{e}_\infty \rVert_2$	$\max \lvert e_\infty(k) \rvert$
Scalar Weighting	1.0419	0.4222
Shaped Weighting	0.8717	0.3214
Improvement	**16%**	**24%**

B. Perturbed Plant Performance

In this section we present simulation results for the perturbed system, $P(\mathrm{q}) = \hat{P}(\mathrm{q})\left(1 + W_\Delta(\mathrm{q})\right)$. Because the learning function **L** does not invert the perturbed system, convergence is slower than for the nominal system. As shown in Figure 10, the control convergences monotonically on the perturbed system. Figure 11 shows that the shaped weighting also outperforms the scalar weighting on the perturbed system. Although this is not guaranteed by the design, we might reasonably expect this result since the perturbed dynamics do not change the location of the α-segment.

Finally, to demonstrate the robustness limitations of large weightings in Optimal ILC, we create a third weighting vector with the scalar weighting $c_{scalar} = 200$. A check of (15) shows that this weighting is not robustly monotonic for the perturbed system. The first 100 iterations are simulated using this weighting and plotted in Figure 12. The results show large transient growth with approximately 7x amplification of the initial error after 100 iterations. It is interesting that this ILC is actually stable as determined by checking the system eigenvalues [3], and thus we expect the control to convergence after sufficient iterations. Clearly, however, the large transient growth is undesirable, which reinforces the necessity of robust monotonic convergence, rather than the weaker condition of robust stability.

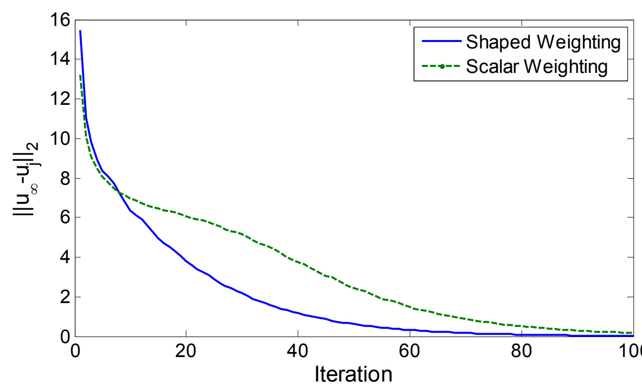

Fig. 10. Monotonic convergence of the control on the perturbed plant P.

$$G(\mathrm{q}) = \frac{0.058593(\mathrm{q}-0.5318)(\mathrm{q}^2-1.89\mathrm{q}+0.9716)(\mathrm{q}^2-1.703\mathrm{q}+0.9595)(\mathrm{q}^2-1.461\mathrm{q}+0.9264)(\mathrm{q}^2-1.11\mathrm{q}+0.8759)}{(\mathrm{q}-0.6193)(\mathrm{q}-0.9251)(\mathrm{q}^2-1.294\mathrm{q}+0.5162)(\mathrm{q}^2-1.723\mathrm{q}+0.9719)(\mathrm{q}^2-1.589\mathrm{q}+0.9055)(\mathrm{q}^2-1.268\mathrm{q}+0.8754)} \tag{23}$$

$$C(\mathrm{q}) = \frac{\begin{gathered}0.21546(\mathrm{q}-0.7394)(\mathrm{q}-0.9252)(\mathrm{q}-0.08362)(\mathrm{q}^2-1.974\mathrm{q}+0.9744)(\mathrm{q}^2-1.201\mathrm{q}+0.3644)(\mathrm{q}^2-1.338\mathrm{q}+0.543)\\ \cdot(\mathrm{q}^2-1.72\mathrm{q}+0.97)(\mathrm{q}^2-1.573\mathrm{q}+0.8973)(\mathrm{q}^2-1.262\mathrm{q}+0.8723)(\mathrm{q}^2-0.4859\mathrm{q}+0.8786)\end{gathered}}{\begin{gathered}\mathrm{q}(\mathrm{q}-0.8988)(\mathrm{q}-0.9969)^3(\mathrm{q}-0.5403)(\mathrm{q}^2-0.8642\mathrm{q}+0.2344)(\mathrm{q}^2-1.784\mathrm{q}+0.869)\\ \cdot(\mathrm{q}^2-1.704\mathrm{q}+0.9596)(\mathrm{q}^2-1.439\mathrm{q}+0.87)(\mathrm{q}^2-1.189\mathrm{q}+0.8285)(\mathrm{q}^2-0.4483\mathrm{q}+0.8109)\end{gathered}} \tag{24}$$

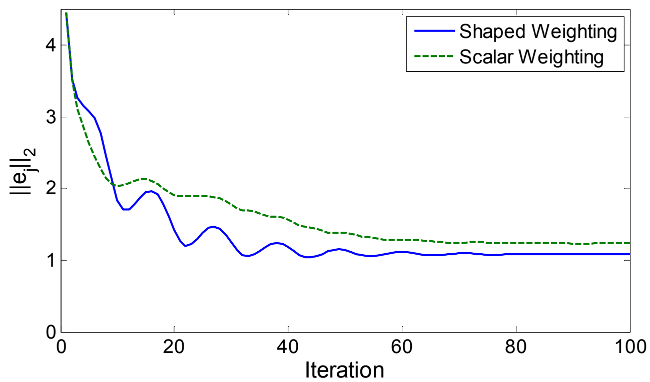

Fig. 11. Error convergence on perturbed plant $P(q)$.

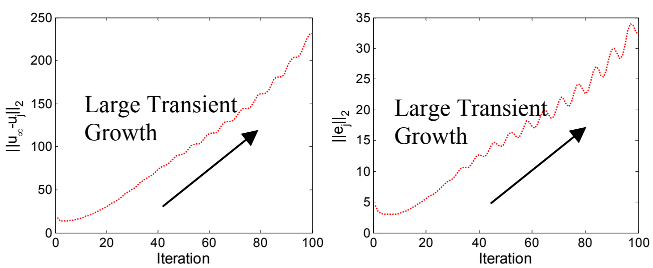

Fig. 12. Large transient growth for non-robust weighting, $c_{scalar} = 200$.

VI. Conclusions

In this work we examined the robustness of Optimal ILC with respect to monotonic convergence. Sufficient conditions for robust monotonic convergence were developed and found to impose constraints on weighting matrix selection for sufficiently large uncertainties. These constraints translate to performance limitations. A time-frequency design methodology was presented to optimize the performance within the robustness constraints by shaping the weightings based on the initial tracking error.

The methodology was applied to the design of an Optimal ILC for a nanopositioning system. Simulations showed that the shaped weighting resulted in 16% improvement in converged error 2-norm and 24% improvement in peak error versus unshaped weighting.

References

[1] Moore, K.L., Iteration Learning Control for Deterministic Systems, Springer-Verlag, 1993.

[2] Bien, Z. and J.-X. Xu, Iterative Learning Control: Analysis, Design, Integration and Applications, Kluwer Academic Publishers, 1998.

[3] Bristow, D.A., M. Tharayil, and A.G. Alleyne, "A Survey of Iterative Learning Control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96-114, 2006.

[4] Amann, N., D.H. Owens, and E. Rogers, "Iterative Learning Control for Discrete-Time Systems With Exponential Rate of Convergence," *IEE Proceedings: Control Theory and Applications*, vol. 143, no. 2, pp. 217-224, 1996.

[5] Lee, J.H., K.S. Lee, and W.C. Kim, "Model-Based Iterative Learning Control With a Quadratic Criterion for Time-Varying Linear Systems," *Automatica*, vol. 36, no. 5, pp. 641-657, 2000.

[6] Gunnarsson, S. and M. Norrlof, "On the Design of ILC Algorithms Using Optimization," *Automatica*, vol. 37, no. 12, pp. 2011-2016, 2001.

[7] Gorinevsky, D., "Loop Shaping for Iterative Control of Batch Processes," *IEEE Control Systems Magazine*, vol. 22, no. 6, pp. 55-65, 2002.

[8] Dijkstra, B.G. and O.H. Bosgra, "Convergence Design Considerations of Low Order Q-ILC for Closed Loop Systems, Implemented on a High Precision Wafer Stage," *Proceedings of the Conference on Decision and Control,* 2002, pp. 2494-2499.

[9] Dijkstra, B.G. and O.H. Bosgra, "Extrapolation of Optimal Lifted System ILC Solution, With Application to a Waferstage," *Proceedings of the American Control Conference,* 2002, pp. 2595-2600.

[10] Tousain, R. and D. van Casteren, "Iterative Learning Control in a Mass Product: Light on Demand in DLP Projection Systems," *Proceedings of the American Control Conference,* 2007, pp. 5478-5483.

[11] Longman, R.W., "Iterative Learning Control and Repetitive Control for Engineering Practice," *International Journal of Control*, vol. 73, no. 10, pp. 930-954, 2000.

[12] Norrlof, M. and S. Gunnarsson, "Time and Frequency Domain Convergence Properties in Iterative Learning Control," *International Journal of Control*, vol. 75, no. 14, pp. 1114-1126, 2002.

[13] Norrlof, M. and S. Gunnarsson, "Experimental Comparison of Some Classical Iterative Learning Control Algorithms," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 636-641, 2002.

[14] Bristow, D.A. and A.G. Alleyne, "Monotonic Convergence of Iterative Learning Control for Uncertain Systems Using a Time-Varying Filter," *IEEE Transactions on Automatic Control*, vol. 53, no. 2, pp. 582-585, 2008.

[15] Skogestad, S. and I. Postlethwaite, Multivariable Feedback Control: Analysis and Design, John Wiley & Sons, 1996.

[16] Bristow, D.A. and A.G. Alleyne, "Optimizing Learning Convergence Speed and Converged Error for Precision Motion Control," *to appear Journal of Dynamic Systems, Measurement, and Control*.

[17] Cohen, L., Time-Frequency Analysis, Prentice Hall, 1995.

[18] Helfrich, B.E., C. Lee, D.A. Bristow, X. Xiaohui, J. Dong, A.G. Alleyne, S.M. Salapaka, and P.M. Ferreira, "A Framework for Combined H-Infinity and Iterative Leraning Control Design With Application to Nanopositioning Systems," *to appear the American Control Conference*, 2008.