2004 ASME International Mechanical Engineering Congress and Exposition
November 13-20, 2004, Anaheim, California USA

IMECE2004-61558

DESIGN AND CONVERGENCE OF A TIME-VARYING ITERATIVE LEARNING CONTROL LAW

Marina Tharayil and Andrew Alleyne,
Department of Mechanical & Industrial Engineering
University of Illinois, Urbana-Champaign
Urbana, IL 61801
tharayil@uiuc.edu, alleyne@uiuc.edu

ABSTRACT

This paper presents a novel linear time-varying (LTV) iterative learning control law that can provide additional performance while maintaining the robustness and convergence properties comparable to those obtained using traditional frequency domain design techniques. Design aspects of causal and non-causal linear time-invariant (LTI), along with the proposed LTV, ILC update laws are discussed and demonstrated using a simplified example. Asymptotic as well as monotonic convergence, robustness and performance characteristics of such systems are considered, and an equivalent condition to the frequency domain convergence condition is presented for the time-varying ILC. Lastly the ILC algorithm developed here is implemented on a Microscale Robotic Deposition system to provide experimental verification.

NOMENCLATURE

j	iteration number
k	discrete time index
n	number of samples in a period
T	length of period
P_m	plant model
P_a	actual plant
L	learning filter
Q	Q-filter
Q_*	Q-filter with cutoff frequency $\omega_c = *$
Q_{tv}	time-varying Q-filter
Q_{zp}	zero phase application of Q
u_∞	converged control
e_∞	converged error
q_i, l_i, p_i	Markov parameters of Q, L, P

1. INTRODUCTION

There is a large class of control applications where the same task is performed repeatedly. An obvious example is robotic applications where the control objective is to track the same command over and over. Typically, in this type of situation, the system would also encounter the same disturbance and nonlinear effects each time. Under these conditions, traditional feedback controllers would generate the same set of errors at each recurrence of the control task. It is obvious that performance of such systems can be improved significantly by using a controller that learns from the previous experience. Ideally, one would expect that a good controller of this type should generate a control effort that improves the system's performance with each iteration. Another natural notion is that such a controller would generate high frequency control effort when the reference (and/or disturbance) has significant high frequency components and the control effort would be of low frequency when the reference/disturbance are relatively constant.

This paper combines these three ideas in proposing a time-varying Iterative Learning Control (ILC) law that can improve the performance of a given system, while retaining good robustness properties, by tuning the control law for the system's reference/disturbance signals. This type of a control law is especially beneficial for systems with localized hard nonlinearities such as stiction, or systems with reference trajectories and/or disturbances that have significant variation in their frequency content during different portions of the period. Monotonic convergence conditions are presented to check when such a design results in improved control with each iteration.

It is assumed that the control is implemented digitally, and therefore all analysis and modeling is performed in the discrete-time domain. The general linear ILC structure and associated

robustness and performance issues are introduced via discrete frequency domain analysis of causal, linear time-invariant (LTI) laws in the following section. As a solution to some of the limitations of such laws, the time-varying ILC update law is presented in Section 3. Next, stability and monotonic convergence conditions for such an update law are studied in a lifted discrete-time domain. Following that, results from applying such a proposed law to the Microscale Robotic Depositioning (MRD) system are presented before concluding the paper.

2. ITERATIVE LEARNING CONTROL

Figure 1 shows a schematic of an ILC scheme. Here the subscript j represents the trial or repetition number, and the reference signal $y_d(t)$ is defined on the interval $[0, T]$. At any given repetition, j, a control input of $u_j(t)$ is applied to the system to produce output $y_j(t)$, $t \in [0\ T]$, where T is the length of the periodic reference. The input and output of the j^{th} trial are stored in memory and used along with the fixed reference to calculate the input for the $j+1^{th}$ trial. In this way, iterative learning control is a feedforward control in the time domain that utilizes feedback from the iteration domain. Thus the goal of the algorithm is to design an update law to produce the lowest possible error as j tends to infinity. In most ILC systems, it is assumed that the plant initial conditions are reset at the start of every period ($x_j(0)=x_0$). Also, the system is assumed to be stable, or stabilized using feedback control.

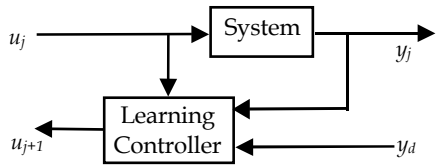


Fig. 1. First Order Learning Control Configuration

A general first order ILC update law is of the form:

$$u_{j+1}(t) = f_L\left(u_j(t), e_j(t), t\right) \tag{1}$$

where

$$e_j(t) = y_d(t) - y_j(t) \tag{2}$$

Three important attributes of such systems are:

1) *Stability / Convergence*: Existence of $u_\infty(t)$ such that

$$\lim_{j\to\infty} u_j(t) = u_\infty(t) \tag{3}$$

2) *Performance*: Existence of $\varepsilon \geq 0$ such that the converged error $e_\infty(t)$ satisfies

$$\left\| e_\infty(t) \right\| < \varepsilon \tag{4}$$

3) *Robustness*: Satisfaction of stability / convergence condition in the presence of plant (P) uncertainties:

$$\widetilde{P} = P(1+\Delta) \tag{5}$$

2.1 LTI Systems

The input-output behavior of a general single-input-single-output (SISO) causal LTI system for the j^{th} iteration can be written in the following form [1-4].

$$y_j(z) = P(z)u_j(z) + d_j(z) \tag{6}$$

Where $P(z)$ is a stable discrete-time transfer function, z^{-1} is the standard delay operator, and the disturbance is assumed the same for all iterations ($d_j(z) = d(z)$). The LTI first order update law can be written using two LTI discrete SISO systems, Q and L, as shown in (7). A schematic of the LTI first order ILC update law is given in Figure 2

$$u_{j+1}(z) = Q(z)\left[u_j(z) + zL(z)e_j(z)\right] \tag{7}$$

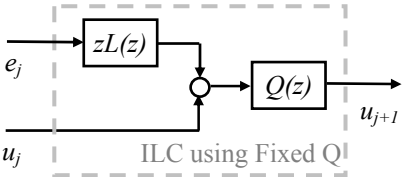


Fig. 2. Causal, LTI ILC update law

Here $L(z)$ is a learning operator that modifies the current control based on the previous error, $Q(z)$ is generally a low pass filter designed to cutoff learning at high frequencies. A one time-step lead is included in the error term to compensate for the relative degree of $P(z)$, assumed to be one here. This is achievable in ILC update laws because the entire error signal from the previous iteration is available at the next iteration. The learning operator $L(z)$ is a discrete filter designed to satisfy the stability condition, (8).

A widely known [1] sufficiency condition for monotonic convergence in the frequency domain for this LTI system is given by

$$\left| Q(e^{i\omega T})\left[1 - e^{i\omega T}L(e^{i\omega T})P(e^{i\omega T})\right] \right| < 1 \qquad \forall \omega \in [0, \pi] \tag{8}$$

For a system that satisfies (8), the converged error is given by

$$e_\infty(z) = \frac{1 - Q(z)}{1 - Q(z)\left[1 - zL(z)P(z)\right]}\left(y_d(z) - d(z)\right) \tag{9}$$

Note from (9) that a necessary condition for convergence to zero error is that $Q(z) \equiv 1$. However, a low pass Q-filter is vital in most practical problems because condition (8) could otherwise easily be violated in the presence of plant uncertainties, which usually come in at high frequencies. From (7) it can be seen that the bandwidth of the Q-filter establishes the control bandwidth, and therefore sets off the trade-off between performance and robustness. Increasing the Q-filter bandwidth enlarges the frequency range where perfect tracking can be achieved, while also increasing the risk of instability due to unmodeled dynamics or uncertainties. This can be clearly seen for the case where Q is a cliff filter, so that $Q(z) \equiv 1$ for $\omega < \omega_c$ and $Q(z) \equiv 0$ for all $\omega \geq \omega_c$. It is evident that $e_\infty(z) = 0$ $\forall \omega < \omega_c$ and $e_\infty(z) \neq 0$ $\forall \omega \geq \omega_c$ in such case.

It has been observed that using a zero-phase Q-filter can provide improved system performance by reducing the error

due to the phase lag in Q [5]. In general, zero phase filters are non-causal in nature. Given an LTI causal filter, one convenient way to eliminate phase lag is by filtering the same signal twice in the following fashion [6]. Filter the signal using the causal low pass filter once, $\tilde{x}(k) = Q(q)x(k)$, and filter the new signal backwards using the same filter, $y(T-k) = Q(q)\tilde{x}(T-k)$. This procedure results in twice the attenuation and none of the phase lag. Such an operation is achievable for ILC systems because the entire signal data from the previous iteration should be available when calculating u_j for the current iteration. Note that this also results in a decrease in the bandwidth as the attenuation increases for each frequency. For MATLAB users, this type of filtering can be performed using the *"filtfilt"* command. It has been shown that the discrete transfer function of such a filter can be written as follows [6].

$$Q_{zp}(z) = Q(z^{-1})Q(z) \tag{10}$$

At this point, an example is introduced to demonstrate some of the convergence, robustness and performance issues encountered in LTI ILC systems. This same example is used to demonstrate the design procedure of the time varying Q-filter based ILC law and the ensuing benefits in Section 3

2.2 Motivating Example

Consider a system as shown in Figure 3a, with noise on the signals and backlash on the output, required to track the trajectory shown in Figure 3b repeatedly. A sample time of 0.01 seconds is used in this example. We assume that the linear model of the plant obtained using frequency response is given by P_m, whereas the actual plant contains additional high frequency dynamics, as in P_a. Figure 4 shows the bode plots of the actual plant and the plant model.

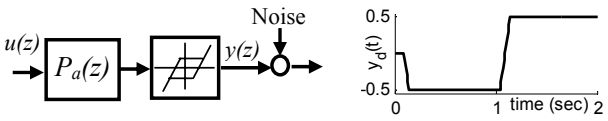


Fig. 3. (a) Example system and (b) reference, y_d

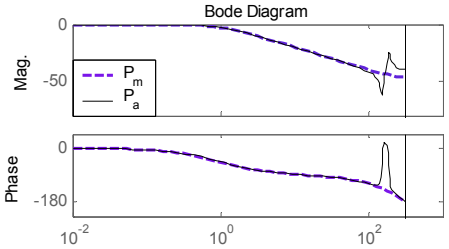


Fig. 4. Bode Plots of P_m and P_a.

$$P_m(z) = \frac{0.00995}{z-0.99}; \quad P_a(z) = 1.35P_m(z)\frac{(z^2-0.1237z+0.9453)}{(z^2+0.5589z+0.8985)} \tag{11}$$

A PD-type learning law can be designed using the plant model to satisfy (8) without a Q-filter. However, such an update law violates the convergence condition in the actual system. This can be seen in Figure 5, where the magnitude of the stability transfer function, $Q(z)\big(1-zL(z)P(z)\big)$, is plotted. Adding a first order butterworth Q-filter with cutoff frequency of 20 rads/sec stabilizes the update law for the actual plant.

$$L(z) = 100.5\frac{z-0.99}{z}; \qquad Q = \frac{0.0912*(z+1)}{z-0.8176} \tag{12}$$

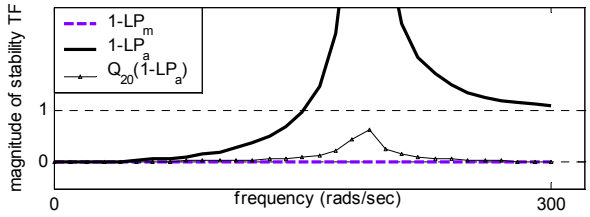


Fig. 5. Condition (8) for $P_m(z)$ and $P_a(z)$

Figure 6 shows the maximum values of the error as a function of iteration for both the actual system and the simplified model, P_m, with and without using the low-pass Q-filter. The performance loss due to the filtering is apparent from the converged error values of P_m. At the same time, the necessity of the added robustness given by the Q-filter is plainly visible by the fact that the actual plant diverges without filtering on u.

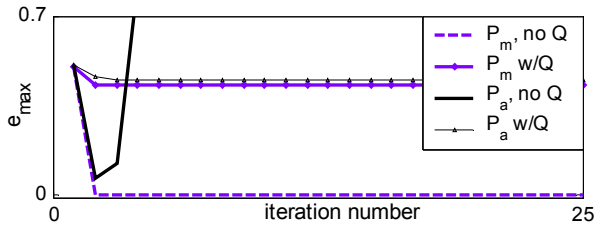


Fig. 6. The maximum value of the error at each iteration.

3. TIME-VARYING ILC DESIGN

As was mentioned earlier, condition (8) is a *sufficient* condition for convergence of the LTI ILC systems, which could be overly restrictive at times. For many cases, one can achieve considerably better performance while maintaining good convergence behavior by increasing the bandwidth of the Q-filter for short intervals of time when necessary. This section uses the example from Section 2.2 to introduce the design procedure for the proposed time-varying ILC law.

Step 1. For a given system, generate the plant model from frequency response data obtained using available tools. Stabilize the plant using feedback control as needed.

Step 2. Design a filter $L(z)$ to satisfy (8), without needing a Q-filter if possible.

Step 3. Apply a Q-filter with cutoff frequency based on the resolution of the frequency response, or as needed to satisfy (8), whichever is lower.

Steps 1, 2, and 3 have been performed for the previous example. It is assumed that the maximum resolution of the frequency response is 20 rads/sec, and therefore a first order butterworth filter with ω_c = 20 rads/sec is set as the base filter.

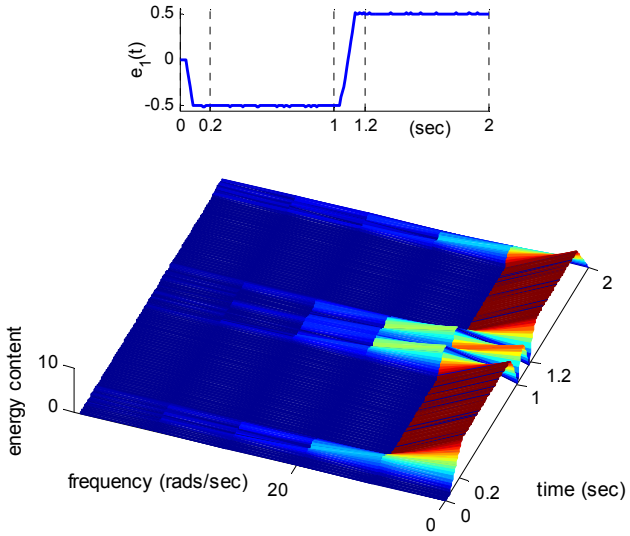


Fig. 7. (a) $e_1(t)$ and (b) the time-frequency distribution of $e_1(t)$.

<u>*Step 4*</u>. Analyze the time-frequency content of the error signal generated at the first iteration – i.e. without ILC – to identify effects due to nonlinearities, disturbances, reference, noise, etc. on the actual plant.

Figures 7a and 7b show $e_1(t)$ and the Wigner-Ville time-frequency distribution of $e_1(t)$ respectively. It can be seen that in this case, the significant high frequency energy content is present in the signal during the intervals $t \in (0,0.2)$ and $t \in (1,1.2)$, while most of the energy is contained in the low frequency regions in the remainder of the period.

<u>*Step 5*</u>. If the signal has significant frequency band variations, switch to a higher bandwidth Q-filter locally in time to accommodate the high frequency content in e. That is, at time intervals during which considerable amount of energy is contained in the high frequency bands, switch to a high bandwidth Q-filter, Q_{high}. Details of the filter implementation are given shortly. Effectively, this method increases the bandwidth of the filtered signal locally, and one can plot the effective bandwidth as a function of time, as shown in Figure 8. Ideally, we would like to switch to a fixed high frequency filter at these intervals, as shown in Figure 8a, but in practice it is important to have a smooth bandwidth profile as shown in Figure 8b.

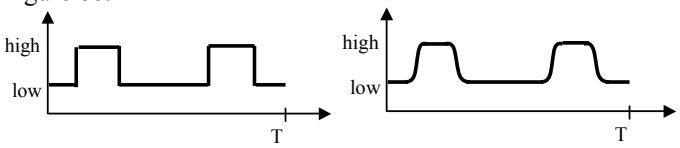


Fig. 8. (a) ideal and (b) actual Q-filter bandwidth profile

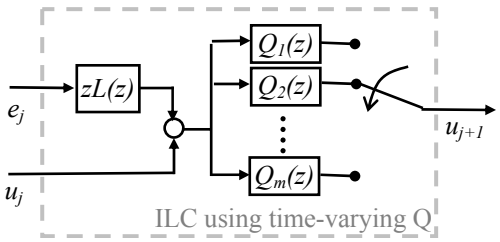


Fig. 9. Schematic of proposed time-varying ILC law.

This is equivalent to the time-varying Q-filter, Q_{tv}, effectively switching between fixed filters Q_1, Q_2,…,Q_m during the course of each iteration. It is assumed that the switching order is fixed for all iterations.

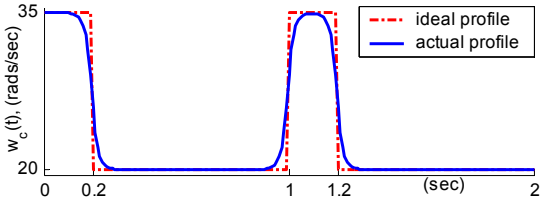


Fig. 10. Bandwidth profile for time-varying Q-filter

For the example, the time-varying Q-filter bandwidth can be designed as follows. As a first try, increase the bandwidth of the Q-filter to 35 rads/sec during the high frequency regions. One can use a zero-phase filter to smoothen this switched bandwidth profile to obtain the continuously varying profile for the time-varying filter, as shown in Figure 10.

<u>*Step 6.*</u> Apply the designed time-varying Q-filter. The system should remain convergent for short enough deviations from the low bandwidth filter.

Figure 11 shows the convergence behavior of the error in the iteration domain for a causal and zero-phase application of the designed Q-filter to the actual plant, P_a. These plots show the convergence using LTI updates laws with (1) the low bandwidth Q-filter, Q_{low} and (2) the high bandwidth Q-filter, Q_{high}, to be compared with (3) the LTV filter Q_{tv} that has bandwidth profile shown in Figure 10, and (4) the zero phase application of this time-varying Q-filter. The converged errors at the 25th iteration are plotted in Figure 12 to show the performance improvement obtained by increasing the Q-filter bandwidth at the regions with considerable energy in the higher frequencies.

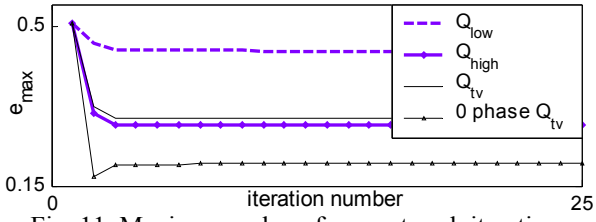


Fig. 11. Maximum value of error at each iteration

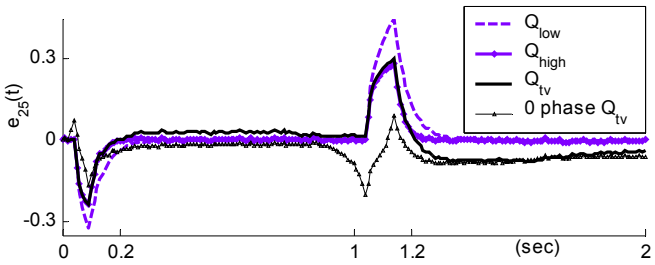


Fig. 12. Converged error plots.

The added robustness of the time-varying Q-filter is more apparent when the maximum cutoff frequency in Figure 10 is raised to 200 rads/sec. Figure 13 plots the convergence behavior for this case. Note the improved performance

obtained by increasing the maximum bandwidth of the Q-filter. However, the LTI Q-filter of cutoff frequency 200 rads/sec results in errors as high as $e_{max} \cong 3*10^7$ by the 25th iteration. Clearly convergence behavior such as this is unacceptable in practical applications.

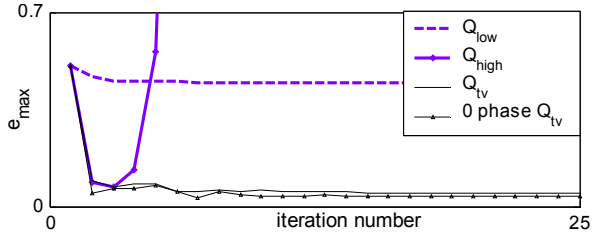


Fig. 13. Maximum value of error at each iteration

From this example, it is seen that some filters can violate the sufficiency condition (8) and yet give good convergence behavior, while others result in system divergence. The following section gives necessary and sufficient conditions for convergence of time-varying and LTI ILC systems, as well as conditions for monotonic convergence.

4. CONVERGENCE CONDITIONS

This section develops convergence conditions for LTI and LTV update laws in the iteration domain using matrix representation of the time-domain system dynamics. This way, the otherwise 2D ILC problem can be written as a 1D problem in the iteration domain.

The LTI plant, *P(z)*, defined in (6) has state space representation (13), and its input output behavior can be written in the form of a convolution sum. Here *k* is the discrete time index and *n* is the number of time steps in a period.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) && k = 0,1,\cdots,n-1 \\ y(k) &= Cx(k) + d(k) && k = 1,\cdots,n \end{aligned} \qquad (13)$$

$$y(k) = CA^k x(0) + d(k) + \sum_{i=1}^{k-1} CA^{k-i-1} Bu(i);\ k = 0,1,\cdots,n$$

Assuming zero initial conditions, *P(z)* can be written as follows [1],[2],[4],[7]. Here p_i, sometimes called the Markov parameters of the plant, are given by $CA^{i-1}B$.

$$P(z) = C(zI - A)^{-1} B = p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3} + \cdots \qquad (14)$$

Note that it is assumed the plant is of relative degree 1 here. For a transfer function of relative degree *r*, the first nonzero element will be multiplied by z^{-r}. Define vectors $\hat{y}_j$ and $\hat{u}_j$, and matrix *P* as follows:

$$\hat{y}_j = \begin{bmatrix} y_j(1) \\ y_j(2) \\ \vdots \\ y_j(n) \end{bmatrix}; \hat{u}_j = \begin{bmatrix} u_j(0) \\ u_j(1) \\ \vdots \\ u_j(n-1) \end{bmatrix};\ P = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ p_2 & p_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_n & p_{n-1} & \cdots & p_1 \end{bmatrix} \qquad (15)$$

Then the linear plant (6) can now be written as follows:

$$\hat{y}_j = P\hat{u}_j + \hat{d} \qquad (16)$$

where $\hat{d}$ is a vector of the form (15) containing the effects of periodic disturbance. The general ILC update law can be written in the form:

$$\hat{u}_{j+1} = Q(\hat{u}_j + L\hat{e}_j) \qquad (17)$$

where *Q* and *L* are *nxn* matrices that determine the updated control vector of the $j+1^{th}$ iteration based on the current control and error vectors.

A well-known *necessary and sufficient* (*N&S*) condition for *asymptotic convergence* of $\hat{u}_j$ to $\hat{u}_\infty$ as $j \to \infty$ is given by [1], [7]:

$$\left|\lambda_i\left(Q(I - LP)\right)\right| < 1 \quad \forall i = 1,2,\cdots,n \qquad (18)$$

Here the matrix *Q*(*I-LP*) is the stability matrix that determines the iteration dynamics of a given ILC system. A *sufficient* (*S*) condition for *monotonic* convergence of the system in a given norm $\|.\|_i$ is given by:

$$\left\|Q(I - LP)\right\|_i < 1 \qquad (19)$$

Here $\|.\|_i$ is the induced norm of the matrix. For a system that satisfies (18), the converged error is given by:

$$\hat{e}_\infty = \lim_{j\to\infty} \hat{e}_j = \left[I - P\left[I - Q + QLP\right]^{-1} QL\right]\left(\hat{y}_d - \hat{d}\right) \qquad (20)$$

Where $\hat{y}_d$ is a vector of the form (15) that represents the desired trajectory. Note again that in general, perfect tracking and disturbance rejection can only be obtained if *Q* = *I*.

The matrices *Q* and *L* in (15) were of the most general form. For an ILC update law described by causal, LTI discrete transfer functions, matrices *Q* and *L* will be of similar form as *P*.

$$L = \begin{bmatrix} l_0 & 0 & \cdots & 0 \\ l_1 & l_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n-1} & l_{n-2} & \cdots & l_0 \end{bmatrix};\ Q = \begin{bmatrix} q_0 & 0 & \cdots & 0 \\ q_1 & q_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ q_{n-1} & q_{n-2} & \cdots & q_0 \end{bmatrix} \qquad (21)$$

Here it is assumed the transfer functions *L(z)* and *Q(z)* are of relative degree zero, or the matrices have been shifted accordingly. In this case, the stability transfer function *Q*(*I-LP*) is lower triangular, with identical values across its diagonal. Hence the *N&S* condition (18) simplifies to a scalar condition.

$$\left|q_0(1 - l_0 p_1)\right| < 1 \qquad (22)$$

A time-varying Q-filter that switches between two causal LTI filters $Q_a(z)$ and $Q_b(z)$ can be written as shown in (23) [7]. Here $q_{a,i}$ and $q_{b,i}$ represent the Markov parameters of filter Q_a and Q_b respectively

$$Q_{tv} = \begin{bmatrix} q_{a,0} & 0 & 0 & \cdots & & & 0 \\ q_{a,1} & q_{a,0} & 0 & \cdots & & & 0 \\ \vdots & \vdots & \ddots & \ddots & & & \vdots \\ q_{b,p} & \cdots & q_{b,1} & q_{b,0} & 0 & & \\ \vdots & \cdots & \vdots & \vdots & \ddots & \ddots & \\ q_{a,n-2} & \cdots & & \cdots & q_{a,1} & q_{a,0} & 0 \\ q_{a,n-1} & q_{a,n-2} & \cdots & & \cdots & q_{a,1} & q_{a,0} \end{bmatrix} \begin{matrix} \} \text{Qa} \\ \\ \} \text{Qb} \\ \\ \} \text{Qa} \end{matrix} \quad (23)$$

The N&S conditions for convergence of the ILC system defined by (16) using the update law (17) with Q filter of the form (23) is given by:

$$\left|q_{a,0}(1-l_0 p_1)\right| < 1 \text{ and } \left|q_{b,0}(1-l_0 p_1)\right| < 1 \quad (24)$$

It can be seen that this result can easily be extended to a system switching between *m* Q-filters, as shown in (25). The sufficient condition for monotonic convergence in a given norm remains the same, (19).

$$\max_{i=i,2,\cdots,m} \left|q_{i,0}(1-l_0 p_1)\right| < 1 \quad (25)$$

It is noted here that the use of non-causal filters results in full matrices Q and L, and (18) and (19) give the *N&S* and *S* conditions for convergence in such case. Typically, zero phase filters have band-diagonal form. A Q-filter that filters backwards and forwards in time, such as described by (10), can be written as follows, where Q contains the Markov parameters of *Q(z)*.

$$Q_{zp} = Q^T Q \quad (26)$$

Similarly, a time-varying zero-phase filter contains parameters of the individual filter being used at a given time instant in the corresponding row. In summary, for a given ILC system, the general absolute condition for convergence (stability) is given by (18), which simplifies to (22, 24, 25) for causal update laws. Monotonic convergence behavior can be guaranteed for a given system if (19) is satisfied. However, (19) is a *sufficiency* condition, which can be violated sometimes and yet result in monotonic convergence. It was shown in [1] that LTI systems satisfying (8) will also satisfy the lifted monotonic convergence condition, (19) for the 2-norm. In fact, they are equivalent in the limit, as n approaches ∞. Therefore, writing the time-varying Q-matrices in forms (23), or the equivalent zero-phase form, allows the user to perform a convergence test equivalent to (8) for time-variant filters using the 2-norm check given by (19).

Example System Asymptotic Convergence: The values of the various p_i, l_i, and q_i used in the example are given in (27). For the causal update laws, it is apparent by inspection that *N&S* condition (18) is satisfied for both P_m and P_a using any given filter. So in theory, the apparently divergent systems shown in Figures 6 and 13 would have converged eventually, without the presence of nonlinearities and other external effects. This can explain why some of the update laws which used filters violating condition (8) converged, while others did not. As was mentioned before, fulfilling (18) ensures eventual convergence, but not monotonic convergence.

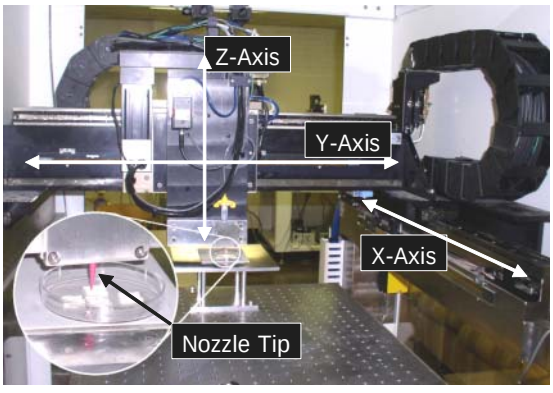
$$\begin{aligned} P_m(z) &= .01z^{-1} + .0099z^{-2} + .0098z^{-3} + .0097z^{-4} + \cdots \\ P_a(z) &= .0134z^{-1} + .0041z^{-2} + .0098z^{-3} + .0148z^{-4} + \cdots \\ L(z) &= 100.5 - 99.5z^{-1} \\ Q_{20}(z) &= .0912 + .1657z^{-1} + .1355z^{-2} + .1108z^{-3} + \cdots \\ Q_{35}(z) &= .1502 + .2553z^{-1} + .1786z^{-2} + .1249z^{-3} + \cdots \\ Q_{200}(z) &= .6090 + .4762z^{-1} - .1038z^{-2} + .0226z^{-3} + \cdots \end{aligned} \quad (27)$$

Example System Monotonic Convergence: Monotonic convergence condition (19) was satisfied by all update laws for the plant model, P_m, under the 2-norm, as per prediction. Next, condition (19) was checked for the actual plant using the 2-norm, and the results are shown below. An ideal switched filter of the form (23), i.e. with effective bandwidth profile of the form Figure 8a, was assumed in these calculations. Here, one can see how (19) forms an equivalent condition as (8) for LTI as well as LTV update ILC systems. Also apparent is the advantage of using the LTV update law rather than an LTI filter using the highest cut-off frequency in the LTV law. Monotonic convergence can be guaranteed for the time-varying update law with $\omega_{c,max} = 35$, but not the LTI Q-filter using the same cut-off frequency, although both achieved similar performance. This shows that one can generate significantly better performing ILC update laws using the proposed method while retaining not only convergence, but also monotonic convergence properties.

$$\begin{aligned} \left\|Q_{20}(I-LP_a)\right\|_2 &= 0.6914 & \left\|Q_{tv\_35}(I-LP_a)\right\|_2 &= 0.9045 \\ \left\|Q_{35}(I-LP_a)\right\|_2 &= 1.2115 & \left\|Q_{tv\_200}(I-LP_a)\right\|_2 &= 4.7242 \\ \left\|Q_{200}(I-LP_a)\right\|_2 &= 7.0666 & & \end{aligned} \quad (28)$$

5. EXPERIMENTAL RESULTS

The Microscale Robotic Deposition (μRD) system uses robotic positioning to deposit an 'ink' for 3-D construction of complex parts of small dimensions [8]. The tracking stage used and a schematic of the system are shown in Figure 14a and 14b. Very precise X-Y-Z axis positioning of the robot end effector is required for the accurate manufacturing of the desired parts. The causal algorithm described in this paper is applied to control the X-axis position here. The X-axis is driven by a linear motor with lubricated ball bearing slides. A double lead feedback controller is implemented in addition to the ILC controller.



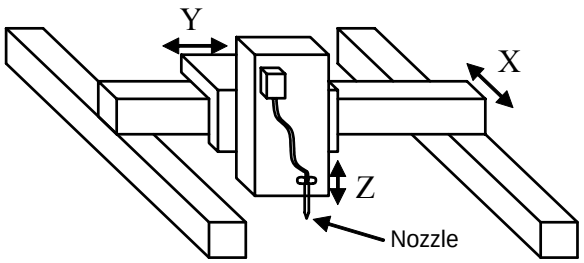


Fig. 14. (a) The actual μRD tracking stage, and (b) a schematic.

The discrete linear model used for the closed loop X-axis positioning system dynamics using a sampling period of 1ms is shown in (29). The actual system, in addition to having high frequency resonances, experiences friction and other nonlinear effects.

$$P_X(z) = \frac{8.64\times10^{-4}(z+0.9993)(z-0.8609)(z-0.8605)}{(z^2-1.91z+0.9126)(z^2-1.747z+0.8113)} \quad (29)$$

The reference trajectory shown in Figure 15a is used for a 1mm change in y. The ILC update algorithm uses a PD learning law as described in [8] with K_{pX}=1.779 and K_{dX}=111. Error at the first iteration and its Wigner-Ville time-frequency distribution are plotted in Figures 15b and 16a. The causal time-varying Q-filter used has effective bandwidth profile shown is Figure 16b, and satisfies conditions (18) and (19) using the plant model.

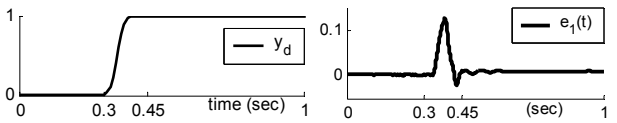


Fig. 15. (a) Desired trajectory, $y_d(t)$, and (b) $e_1(t)$

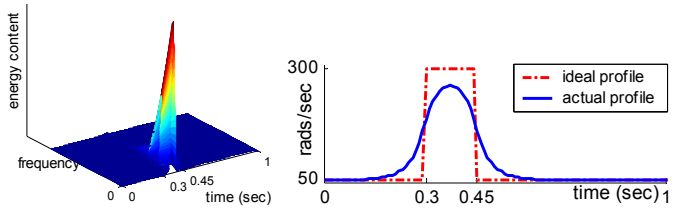


Fig. 16. (a) T-F distribution of $e_1(t)$, and (b) bandwidth profile

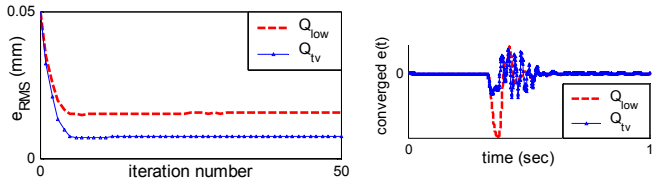


Fig. 17. (a) Experimentally obtained RMS values of $e(t)$, and (b) converged $e(t)$ at the 50th iteration.

$$\begin{aligned} q_{low,0}(1-l_0p_1) &= 0.0221 \quad & \|Q_{50}(I-LP)\|_2 &= 0.6010 \\ q_{high,0}(1-l_0p_1) &= 0.1187 \quad & \|Q_{tv}(I-LP)\|_2 &= 0.6814 \end{aligned} \quad (30)$$

Figures 17 demonstrates the improved performance obtained by increasing the bandwidth for a short period, $t\in[0.3, 0.45]$.

6. CONCLUSIONS

Convergence, robustness, and performance issues are studied for ILC systems. The design of a novel time-varying ILC algorithm is presented. This algorithm is shown to circumvent the traditional trade-off that arises between system performance and robustness to nonlinearities and model uncertainties for a class of systems. A framework is set up for convergence analysis for such systems. Finally, the proposed algorithm is applied on a system with nonlinear effects and high frequency resonances to demonstrate its effectiveness.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to Douglas Bristow for help with obtaining the experimental data. The authors would also like to acknowledge the NSF support for this work under the grant NSF DMI-0140466.

REFERENCES

[1] M. Norrlof and Gunnarsson, S., "Time and frequency domain convergence properties in iterative learning control", *International Journal of Control,* 2002, Vol. 75, No. 14, pp. 1114 – 1126.

[2] M. Phan, Longman, R., and Moore, K., "Unified Formulation of Linear Iterative Learning Control", *Advances in Astronautical Sciences*, 2000, Vol. 106, pp. 93-111.

[3] R. Longman, "Iterative learning control and repetitive control for engineering practice", *International Journal of Control*, 2000, Vol. 73, No. 10, pp. 930 – 954.

[4] Moore, K. L., "Iterative Learning Control: An Expository Overview", *Applied and Computational Controls, Signal Processing, and Circuits*, 1998, Vol. 1, No. 1, pp. 425 – 488.

[5] H. Elci, Longman, R., Phan, M., Juang, J., and Ugoletti, R., "Simple Learning Control Made Practical by Zero-Phase Filtering: Application to Robotics", *IEEE Transactions on Circuits and Systems –I: Fundamental Theory and Applications*, June 2002, Vol. 49, No. 6, pp. 753 – 767.

[6] T. Songchon and Longman, R., "Iterative Learning Control and the Waterbed Effect", *Proceedings of the 2000 AIAA/AIAA Astrodynamics Conference,* Denver, CO, August 2000, pp. 444 – 453.

[7] M. Tharayil and Alleyne, A., "A Time-Varying Iterative Learning Control Scheme", *Proceedings of the 2004 American Control Conference,* 2004.

[8] D. Bristow and Alleyne, A, "Control of a Microscale Deposition Robot Using a New Adaptive Time-Frequency Filtered Iterative Learning Control", *Proceedings of the 2004 American Control Conference,* 2004.