Control of a Microscale Deposition Robot Using a New Adaptive Time-Frequency Filtered Iterative Learning Control

Douglas A. Bristow, Andrew G. Alleyne, and Danian Zheng

Abstract– **A Robocasting manufacturing process and robotic deposition machine are presented. The process requires that the machine be able to track 3-D trajectories with high precision. Iterative Learning Control (ILC) is presented as a viable strategy to meet these demands. Typically, practical implementation of ILC requires some type of Q-filtering that creates an inherent tradeoff between performance and robustness. This tradeoff can be minimized by using a time-varying Q-filter that has been tailored to the system and reference trajectory. A new adaptive time-frequency Q-filtered ILC algorithm is presented to adaptively construct a tailored time-varying Q-filter. Further, because the approach is adaptive, the performance is not limited by overly conservative uncertainty models. A simulation example is presented to demonstrate that, when designed for a nominal plant, the adaptive Q-filtered ILC has performance comparable to that of a standard, fixed-bandwidth Q-filtered ILC. When a perturbation of the plant is introduced, the adaptive Q-filtered ILC adapts to maintain stability, whereas the fixed-bandwidth Q-filtered ILC becomes unstable. The adaptive algorithm is applied to the robotic deposition machine to demonstrate the ability of the algorithm to achieve high precision in this application.**

I. Introduction

Robocasting [1] is a "direct write" robotic deposition manufacturing technique used for the construction of complex prototype parts with small feature sizes. In this technique, a "3-D ink" [2] is extruded through a nozzle and deposited onto a substrate. A robot is used to continuously position the nozzle in 3-D space so that the ink is deposited appropriately to build the part. The Robocasting technique is traditionally used to construct parts with feature sizes in the mesoscale (100 μm – 2 mm), though development of new inks [3] is expanding the operating range to allow for the construction of parts with feature sizes in the microscale (0.1 μm – 100 μm).

An *H-drive* XYZ robot that is used to investigate Robocasting for meso- and micro- length scales was presented in [4]. A schematic of this system is shown in Fig. 1. The X and Y axes of this robot are linear motors with lubricated ball bearing slides. The Z axis is a rotary motor connected via belt to a precision ball screw.

Nominal discrete linear models were acquired for the X and Y axes for a sampling period of 1 ms and given below in (3) and (4). As with all physical systems, the true system behavior is nonlinear. In particular, the resonant frequencies are dependent upon the positions of the other axes, there is sliding and pre-sliding friction in the bearings, and motor force-ripple results in a position-dependency of the plant dc gain [4].

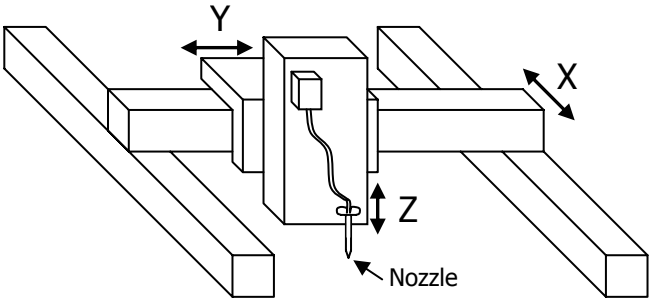

Fig. 1. Schematic of robotic deposition machine.

Linear feedback controllers were designed for each of the axes individually. For the X and Y axes, the double lead controllers,

$$C_X(z) = \frac{40.86(z-0.996)(z-0.935)}{(z-0.877)(z-0.839)} \quad (1)$$

and

$$C_Y(z) = \frac{20.463(z-0.929)(z-0.89)}{(z-0.822)(z-0.821)} \quad (2)$$

were designed to provide stability and an initial level of performance.

However, as the feature sizes of parts decrease from the mesoscale into the microscale, the tracking error tolerances become very tight – maximum error should be less than 5 μm. Achieving this level of transient tracking performance with feedback controllers alone would require very high gain. The repetitive nature of manufacturing, however, makes the process well suited to the "plug-in" Iterative Learning Control (ILC) strategy.

ILC is a powerful technique that can provide high precision transient performance when operating on repeated trajectories. In this technique, a feedforward control signal is developed iteratively by 'learning' from the system's error on each iteration to improve the performance on the next iteration. ILC is oftentimes referred to as a plug-in controller because the feedforward signal it generates can easily be added to an existing closed-loop controller either at the plant input or reference input. In either case, the

This work was supported in part by NSF DMI-0140466 and under an NSF Graduate Research Fellowship.

D. Bristow and A. Alleyne are with the University of Illinois at Urbana-Champaign, Urbana, IL 61801. (email: alleyne@uiuc.edu)

D. Zheng is with the GE Global Research Center, Niskayuna, NY 12309.

$$P_X(z) = \frac{(8.771e-4)(z+0.961)(z^2-1.981z+0.992)(z^2-1.875z+0.975)}{(z-0.999)(z-1)(z^2-1.978z+0.989)(z^2-1.739z+0.8688)} \quad (3)$$

$$P_Y(z) = \frac{(1.946e-3)(z+0.999)(z^2-1.984z+0.991)(z^2-1.873z+0.955)}{(z-0.999)(z-1)(z^2-1.983z+0.991)(z^2-1.87z+0.954)} \quad (4)$$

discrete closed-loop feedback controlled system can be written as

$$\begin{aligned} x_i(k+1) &= Ax_i(k) + Bu_i(k) + w(k) \\ y_i(k) &= Cx_i(k) \end{aligned} \quad (5)$$

where x are the system states, y is the output, u is the feedforward input, and w contains the reference signal and all disturbances which repeat each iteration. The discrete time index is $k \in [0, N]$ and $i \in [0, \infty)$ is the iteration index. The A matrix contains the closed-loop dynamics and is therefore assumed stable.

The desired reference for the system to track is denoted by $y_d(k)$ and is invariant for all iterations. The error, e, is then defined as

$$e_i(k) = y_d(k) - y_i(k). \quad (6)$$

A popular discrete ILC learning function is the simple PD-type ILC [5] given by

$$u_{i+1}(k) = u_i(k) + k_p e_i(k) + k_d\left(e_i(k+1) - e_i(k)\right) \quad (7)$$

where k_p and k_d are the learning gains.

The following assumptions are typically made for ILC systems:

(i) Initial error is zero

$$e_i(0) = 0 \quad \forall i \quad (8)$$

(ii) Initial states are constant

$$x_i(0) = x_0 \quad \forall i \quad (9)$$

(iii) w does not vary with iteration

$$w_i(k) = w(k) \quad \forall i \quad (10)$$

It is important to note that in ILC signals exist in two domains: time *and* iteration. Because the system in (5) is internally feedback stabilized, stability in the time domain is guaranteed as long as u is bounded. To ensure stability of the ILC system in the iteration domain, we must ensure that (7) is stable. That is, stability of the ILC system is equivalent to the existence of a u^* such that

$$\lim_{i\to\infty} u_k(k) = u^*(k), \quad (11)$$

thus guaranteeing u is bounded. It is well known ([6] for instance) that (11) will be satisfied if, and only if,

$$\left|I - k_d CB\right| < 1. \quad (12)$$

However, as pointed out in [6], learning transients are as important an issue as stability. In many cases, stable learning laws may result in error transients that grow to several orders of magnitude larger than the original error. In most practical applications, such transients are unacceptable. A method is presented in [5] for choosing learning gains to achieve monotonic convergence, so that poor learning transients can be avoided. However, with plant uncertainty, it is unlikely that a set of gains which guarantee monotonic convergence for all possible perturbations can be found.

The addition of a low-pass Q-filter [7] is one means of improving robustness and preventing large learning transients. The filtered learning function is

$$u_{i+1}(k) = Q(q)\left[u_i(k) + k_p e_i(k) + k_d\left(e_i(k+1) - e_i(k)\right)\right] \quad (13)$$

where $Q(q)$ is the discrete Q-filter. The Q-filter has several desirable properties including improved robustness, better transient convergence behavior, and decreased sensitivity to noise. In most cases, implementation of ILC in practice will require some type of Q-filter.

The tradeoff for the improved robustness properties provided by the Q-filter is in performance. Whereas the unfiltered learning function of (7) will theoretically converge to zero error, the Q-filtered learning law of (13) will have some non-zero converged error. Using the Z-transform to examine (13) in the frequency domain, we get

$$U_{i+1}(z) = Q(z)\left[U_i(z) + \left(k_p + k_d(z-1)\right)E_i(z)\right]. \quad (14)$$

So, if Q is an ideal low-pass filter, then the ILC will only learn to compensate for the portion of the error whose frequency content is below the bandwidth of the Q-filter. Because Q must be chosen to satisfy stability conditions for the worst case uncertainty, [7], the performance will be limited by plant uncertainty when using a fixed bandwidth Q-filter.

In this paper, an adaptive time-frequency Q-filtered Iterative Learning Control algorithm is presented. An initial version of this algorithm was presented in [8]. This algorithm adaptively develops a time-varying Q-filter to provide high bandwidth where performance is needed and lower bandwidth elsewhere for improved robustness and noise rejection. Because the algorithm is adaptive, performance is not limited by worst-case uncertainty, nor does it require explicit knowledge of uncertainty bounds. Simulations will show that the adaptive Q-filtered ILC algorithm achieves performance on par with the standard fixed-bandwidth Q-filtered ILC when operating on a

'nominal plant' while simultaneously being robust to a larger class of perturbations.

The rest of the paper is organized as follows. Section II presents the adaptive Q-filter algorithm. A simulation example is presented in Section III. In Section IV the algorithm is applied to the robotic deposition machine described above and results are presented. Conclusions are presented in Section V.

II. Adaptive Time-Frequency Filtered ILC

As mentioned in the introduction, the choice of bandwidth for the Q-filter results in a tradeoff between stability and performance. By using a time-varying Q-filter, this tradeoff can be made at each time instant resulting in an overall improvement in both performance and robustness versus that of a fixed-bandwidth Q-filter. It is the goal of the adaptive Q-filter method to develop a time-varying bandwidth profile from which the Q-filter will be generated to achieve the simultaneous improvement in performance and robustness. It is assumed that the learning parameters, k_p and k_d, have been selected *a priori* by the designer. These parameters remain fixed and are not modified by the adaptive algorithm.

The Q-filter can be selected as any causal low-pass filter, though in many cases it is advantageous to use a filter of the form

$$Q(q)=\frac{\alpha_m q^m+\alpha_{m-1}q^{m-1}+\cdots+\alpha_0+\cdots+\alpha_m q^{-(m)}}{2\alpha_m+\cdots+2\alpha_1+\alpha_0} \quad (15)$$

because of its zero-phase property. Note that the filter is non-causal, but that this is not problematic because the filtering occurs on data which is one iteration old. One possible choice for filter coefficients is that of the Gaussian filter such that

$$\alpha_k(\sigma)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{k^2}{2\sigma^2}\right). \quad (16)$$

This filter has the advantage that σ can be related to the bandwidth by approximating the filter in continuous time. In continuous time, the Gaussian filter has the form

$$g(t,\tau)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(t-\tau)^2}{2\sigma^2}\right). \quad (17)$$

The magnitude of the frequency response can be found by solving the convolution with a sinusoid of constant frequency as in,

$$|G(\omega)|=\int_{-\infty}^{\infty}\cos(\omega\tau)g(0,\tau)d\tau=\exp\left(-\frac{\omega^2\sigma^2}{2}\right) \quad (18)$$

Solving for the frequency, ω, at which $|G(\omega)|=\sqrt{1/2}$, the "half-power" bandwidth, Ω, is found as,

$$\Omega=\frac{\sqrt{\ln 2}}{\sigma}. \quad (19)$$

So, using (16) and (19), the Q-filter coefficients for any arbitrary bandwidth, Ω, can be efficiently calculated.

To determine the time-varying frequency content of the tracking error, the Wigner-Ville time-frequency (*t-f*) distribution [9] is used. The discrete, truncated *t-f* distribution for a real signal is given as

$$W(k,\omega)=\frac{1}{\pi}\sum_{\tau=-N}^{N}e(k-\tau)e(k+\tau)\exp(-j2\omega\tau) \quad (20)$$

and yields an energy distribution of the signal, *e*, simultaneously in time, *i*, and frequency, ω. Defining the auto-correlation of *e* as

$$R_k(\tau)=e(k-\tau)e(k+\tau) \quad (21)$$

and scaling ω simplifies (20) to

$$W\left(k,\frac{\omega}{2}\right)=\frac{1}{\pi}\sum_{\tau=-N}^{N}R_i(\tau)\exp(-j\omega\tau). \quad (22)$$

Holding the time, *k* fixed, (22) is simply the Fourier Transform of the auto-correlation. Therefore, the distribution can be efficiently calculated at each time instant using Fast Fourier Transform methods [10].

For example, given the example error signal shown in Fig. 2, the *t-f* distribution is calculated using (22) with N=256 samples and plotted over the time-frequency plane as shown in Fig. 2. To estimate the frequency content of the signal, a level set a distance, *c*, from, and parallel to, the time-frequency plane is used. A bounding function, $F(k)$, is defined from the intersection of the level set and the *t-f* distribution as

$$F(k)=\begin{cases}0, & \text{if } W(k,\omega)\neq c\ \forall\omega\\ \max\omega\ s.t.\ W(k,\omega)=c, & \text{otherwise}\end{cases} \quad (23)$$

The level set is shown in Fig. 3 and the bounding function is plotted against time in Fig. 4. The bounding function provides a very useful interpretation of the frequency content of the signal and is the primary adaptation mechanism in the algorithm.

Selection of the level set height, *c*, is important for correct interpretation of the *t-f* distribution. The goal is to pick a level set height that provides a meaningful frequency distribution. The utility of a specific level set height will vary as the energy in *e* varies. Therefore, it is best to scale the level set height, *c*, with respect to the energy level of the distribution. For instance, the level set height in Fig. 3 is 10% of the peak energy value in the distribution. Additionally, it is important to note that when there is noise in the system, the noise will appear in the *t-f* distribution. It is desirable only to learn the deterministic error in the system and so the level set should always be kept above the level of noise. Generally, noise will have low energy and high frequency, so examination of high-frequency peaks in the *t-f* distribution of a typical error signal can provide an estimate for the noise level.

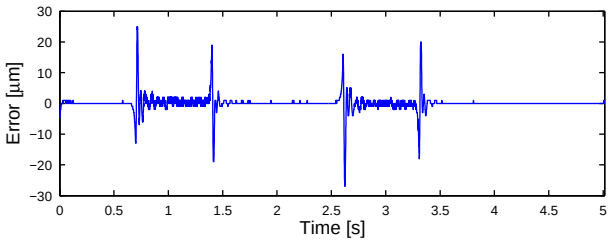

Fig. 2. Example error signal.

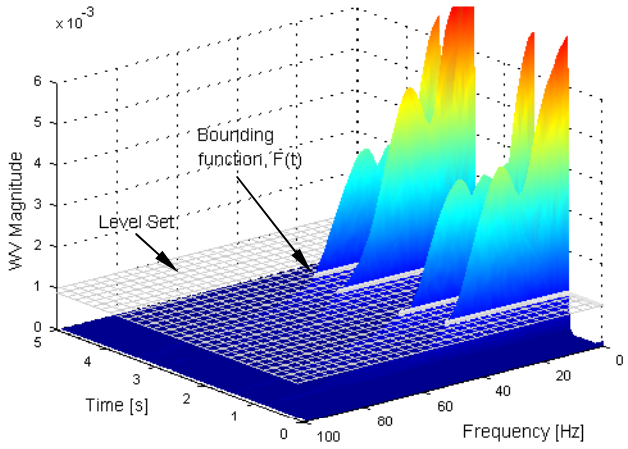

Fig. 3 Time-frequency distribution and level set.

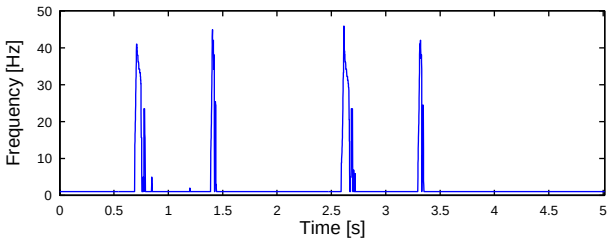

Fig. 4. Bounding function, $F(i)$.

To aid in determining whether or not the adaptation is improving performance, the Short Time Error Norm (STEN) for the i^{th} iteration, defined as

$$S_i(k)=\frac{1}{2N_S}\sum_{\tau=-N_S}^{N_S} e_k^2(k-\tau), \tag{24}$$

is used. The STEN will be used in the adaptation to determine if changes in the bandwidth are improving the error local to a specific time, t. The STEN will also be used to scale the bounding function so that time instants with large STEN are emphasized in the adaptation.

We make one additional observation that it may not be desirable to update the Q-filter bandwidth each iteration because the adaptation can become confounded with convergence transients. Therefore, the ILC update, (13), and the Q-filter bandwidth adaptation should be separated into two iteration scales – fast and slow. The ILC update will continue to occur each iteration (fast dynamics), as denoted by i, the iteration index. A new subscript, $j=p*i$, where $p \in N$ is the iteration-scale separation, will index the slow dynamics of the filter adaptation.

It is now possible to state the Q-filter bandwidth adaptation law as (25) where Ω is the Q-filter bandwidth, L is a low-pass filter, and F and S are defined as before as the bounding function, (23), and STEN, (24), respectively. Together, F and S determine the magnitude change in the bandwidth, Ω. The argument in the signum function determines whether the bandwidth should be increased or decreased based on the previous performance. For instance, if on the previous update the bandwidth was increased and resulted in an increase in STEN, then the next update will decrease the bandwidth. The bandwidth is filtered by the low-pass filter, L, similar to Q in (15) so that the bandwidth profile is smooth to prevent discrete jumps in the feedforward signal.

III. Simulation Example

A simulation example will be used to demonstrate the advantages of the adaptive Q-filtered ILC algorithm over the fixed-bandwidth Q-filtered ILC. Each algorithm will be designed for a nominal plant and performance will be compared for that plant and a perturbation to that plant. The nominal plant is a 2nd order underdamped system with a 10 Hz bandwidth given by

$$P(z)=\frac{0.0017496(z+0.9721)}{(z^2-1.915z+0.9187)}. \tag{26}$$

The perturbed plant includes a multiplicative resonance at 30 Hz and is given by

$$\hat{P}(z)=\frac{0.0017511(z+0.9729)(z^2-1.964z+0.9964)}{(z^2-1.915z+0.9187)(z^2-1.967z+0.9989)}. \tag{27}$$

Fig. 5 shows a frequency plot of the two plants. The PD gains for both systems are selected as k_p=.21 and k_d=30.6. For the fixed Q-filter ILC, a bandwidth of 40 Hz is selected. For the adaptive Q-filtered ILC, an initial bandwidth of 10 Hz is selected. The iteration-scale separation parameter, p, is chosen as 20 to ensure that the ILC has converged before the filter bandwidth is updated. Because the simulation will not include noise, the minimum t-f level set height is selected relatively low as 1e-5. The STEN window width is selected as, N_S=100 and the Ω is filtered with the Gaussian filter L having a bandwidth of 10 Hz.

The algorithms are tested on the reference trajectory shown in Fig. 6. The reference is generated using smooth sigmoids, though large accelerations are used at the corners to test the performance of the two algorithms.

$$\Omega_{j+1}(k)=L(q)\left[\Omega_j(k)-\mathrm{sgn}\left(\left(\Omega_j(k)-\Omega_{j-1}(k)\right)\left(S_j(k)-S_{j-1}(k)\right)\right)\cdot F_j(i)\cdot\frac{S_j(k)}{\max_k\left(S_j(k)\right)}\right] \tag{25}$$

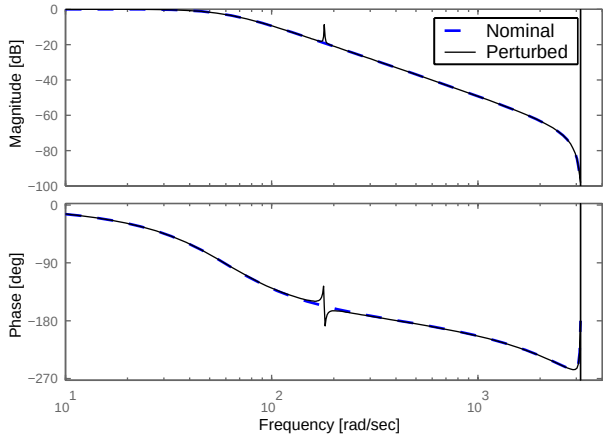

Fig. 5. Nominal and perturbed plants.

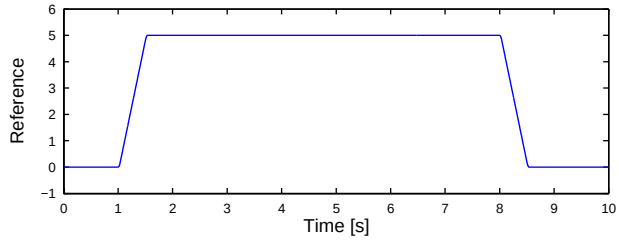

Fig. 6. Reference trajectory.

The maximum error for each algorithm with the nominal plant, (26), are plotted against one another for the first 150 iterations in Fig. 7. For the nominal plant, both algorithms converge with similar performance. The converged Q-filter bandwidth profile for the adaptive Q-filtered ILC is shown in Fig. 8. The algorithm has clearly identified the high acceleration portions of the reference as having high-frequency content and increased the Q-filter bandwidth only at these locations.

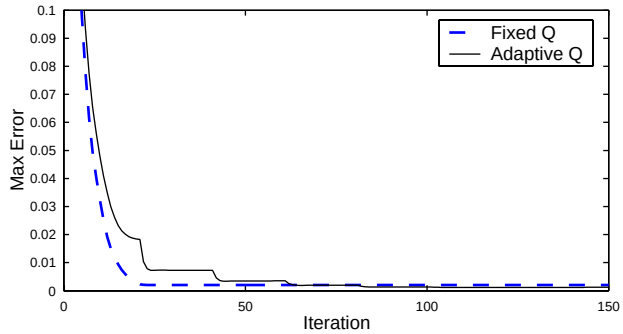

Fig. 7. Max error convergence for nominal plant.

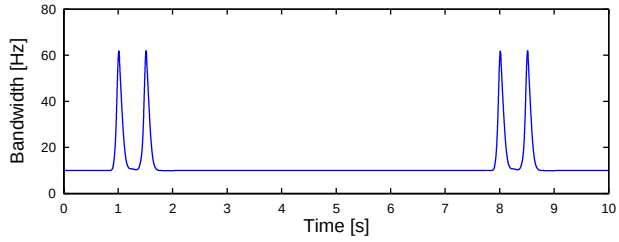

Fig. 8. Converged Q-filter bandwidth for the adaptive algorithm.

Both algorithms are now applied to the perturbed plant, (27). The maximum error on each iteration for both algorithms are again plotted against one another in Fig. 9. In this case, the adaptive Q-filtered ILC algorithm converges to a low error whereas the fixed-bandwidth Q-filtered ILC is unstable demonstrating the enhanced robustness of the adaptive algorithm. The bandwidth of the fixed-bandwidth Q-filtered ILC could be lowered to 30 Hz to maintain stability for the perturbed plant in (27). However, in doing so, the maximum error for the nominal plant (Fig. 7) increases by more than 50% which demonstrates how worst-case uncertainty limits the performance of the fixed-bandwidth Q-filter.

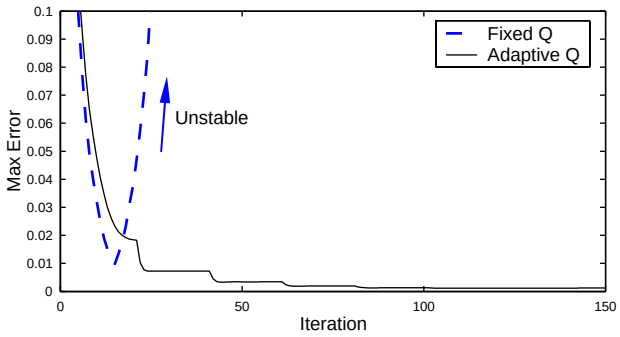

Fig. 9. Max error convergence for perturbed plant.

IV. Application to the Robotic Deposition Machine

The adaptive Q-filtered ILC algorithm is now applied to the robotic deposition machine described in Section I. In this case, the ILC feedforward signal will be added to the feedback control signal. The transfer function from ILC input to plant output is therefore

$$\frac{Y_X(z)}{U_X(z)} = \frac{P_x(z)}{1 + C_X(z)P_X(z)} \tag{26}$$

for the X axis and similarly for the y axis. Simulations using the nominal plant models are used to select the PD gains of k_{pX}=1.68, k_{dX}=115.9, k_{pY}=2.32, and k_{dY}=43.3 to provide a reasonable tradeoff between convergence rate and aggressiveness of the learning.

The initial Q-filter bandwidth used for each axis is 5 Hz, which is known to be quite stable. The other parameters are chosen similarly to those used in the simulation. The iteration-scale separation parameter, p, is chosen as 15. The STEN window width is chosen as 100 and the Ω profile is filtered by L having a bandwidth of 5 Hz. Because of the noise present in the physical system, the minimum *t-f* level set is set to 1e-4.

To test the performance of the adaptive Q-filtered ILC algorithm on the X and Y axes of the robotic deposition machine, the trajectory for a simple lattice is used. The lattice row spacings are 250 μm and the trajectory is shown in Fig. 10. The trajectory is constructed using smooth sigmoids like those used in the simulation example of

Section III. The velocity is 1 mm/s and acceleration at the corners is 250 mm/s^2.

The maximum and RMS errors for each iteration are shown in Fig. 11. The vertical gridlines mark the iterations where the Q-filter bandwidth profile was updated. After 4 filter updates, the adaptive Q-filtered ILC has converged with maximum errors less than 5 μm. The converged contour tracking error at each point on the trajectory is shown in Fig. 10 via the color shading of the trajectory. The maximum contour error here is 3 μm which is even better than the individual axis errors.

Using the adaptive algorithm, a lattice with 4 times as many rows and columns as the one in Fig. 10 is built using $BaTiO_3$ ink [3] and a deposition nozzle diameter of 100 μm. After convergence of the Q-filter, the part is deposited. The constructed part is shown in Fig. 12.

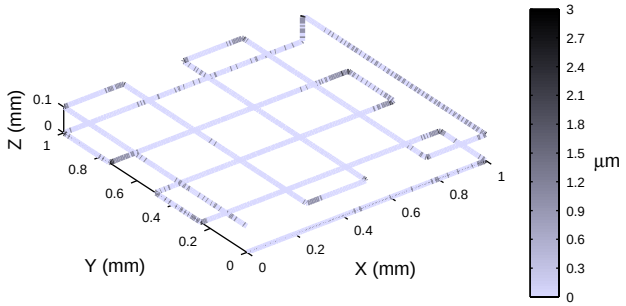

Fig. 10. Converged 2D contour error.

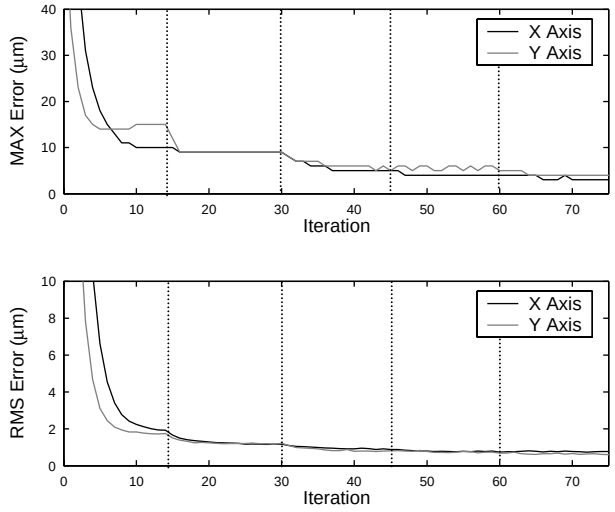

Fig. 11. Maximum and RMS error convergence.

V. Conclusions

A new adaptive Q-filtered ILC algorithm has been presented. By incorporating a time-varying Q-filter rather than a fixed-bandwidth Q-filter, a better tradeoff between performance and robustness can be made. By tailoring the time-varying Q-filter based on time-frequency properties of the error signal, performance is improved. By examining changes in STEN, the adaptive Q-filtered ILC algorithm is able to maintain robustness by reducing Q-filter bandwidth when instabilities grow. Further, by adaptively tailoring the time-varying Q-filter, the adaptive algorithm is not limited by worst-case uncertainty models when true perturbations are much less. A simulation example using a nominal plant and a slightly perturbed plant was used to highlight these advantages.

The adaptive Q-filtered ILC algorithm was applied to a robotic deposition manufacturing machine. Experimental results show that the algorithm was indeed quite stable for the system and capable of achieving very high precision in a real manufacturing process.

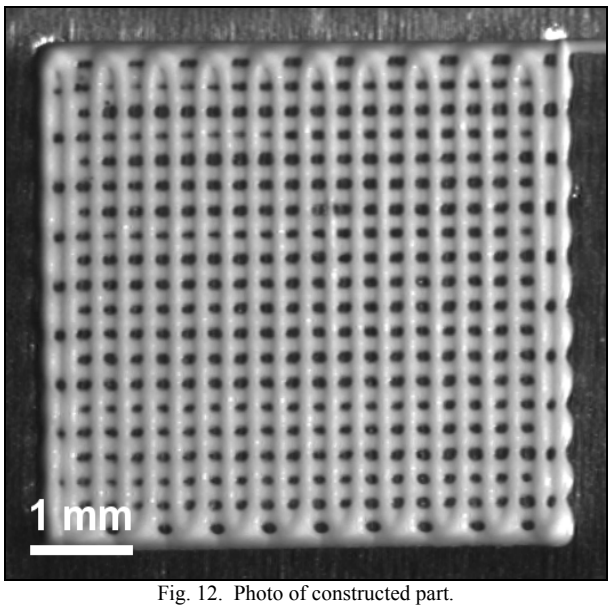

Fig. 12. Photo of constructed part.

References

1. Cesarano, J., R. Segalman, and P. Calvert, "Robocasting Provides Moldless Fabrication from Slurry Deposition", *Ceramic Industry*, pp. 94-102, 1998.
2. Smay, J., J. Cesarano, and J. Lewis, "Colloidal Inks for Directed Assembly of 3-D Periodic Structures", *Langmuir*, vol. 18, pp. 5429-5437, 2002.
3. Li, Q. and J. Lewis, "Nanoparticle Inks for Directed Assembly of Three-Dimensional Periodic Structures", *Advanced Materials*, vol. 15, no. 19, pp. 1639-1643, Oct 2003
4. Bristow, D. and A. Alleyne, "A Manufacturing System for Microscale Robotic Deposition", *Proceedings of the 2003 American Controls Conference*, pp. 2620-2625, 2003.
5. Chen, Y. and K.L. Moore, "An Optimal Design of PD-type Iterative Learning Control with Monotonic Convergence", *Proceedings of the 2002 International Symposium on Intelligent Control*, pp. 55-60, 2002.
6. Huang, Y-C. and R.W. Longman, "Source of the Often Observed Property of Initial Convergence Followed by Divergence in Learning and Repetitive Control", *Advances in Astronautical Sciences*, vol. 90, pp. 555-572, 1996.
7. Tsao, T-C. and M. Tomizuka, "Robust Adaptive and Repetitive Digital Tracking Control and Application to a Hydraulic Servo for Noncircular Machining", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 116, pp. 24-32, 1994.
8. Zheng, D. and Alleyne, A., "Adaptive Iterative Learning Control for Systems with Non-smooth Nonlinearities", *Proceedings of the 2001 ASME IMECE Conference*, pp. 71-78, 2001.
9. L. Cohen, Time-Frequency Analysis, Prentice Hall, NJ, 1995.
10. A. Armbardar, Analog and Digital Signal Processing, International Thomson Publishing, Boston, MA, 1995.