2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Model Inverse Based Iterative Learning Control Using Finite Impulse Response Approximations

Benjamin T. Fine, Sandipan Mishra, Masayoshi Tomizuka

Abstract— **In Iterative Learning Control, the ideal learning filter is defined as the inverse of the system being learned. Model based learning filters designed from the inverse system transfer function can provide superior performance over single gain, P-type algorithms. These filters, however, can be excessively long if lightly damped zeros are inverted. In this paper, we propose a method for designing model based finite impulse response (FIR) learning filters. Based on the ILC injection point and discrete time system model, these filters are designed using the impulse responses of the inverse transfer function. We compare in simulation the ILC algorithms implemented at two different feedforward injection points and two different modeling methods. We show that the ILC algorithm injected at the reference signal and whose model is generated by discretizing the closed loop continuous time transfer function results in a learning filter with no lightly damped zeros. As a result, the learning filter has only two dominant filter taps much like the PD-type learning filter. We then implement these ILC algorithms on a wafer stage prototype. In this motion control application, we show that the model based ILC algorithm outperforms the P-type system in the plant injection architecture where longer FIR filters are needed for learning stability. We also show that the reference injection architecture provides superior performance to the plant injection for both model based and P-type ILC algorithms.**

I. Introduction

Learning control is a popular feedforward control method for systems undergoing a repetitive process. Information from one run can be used to improve the performance of the next run. Its popularity can be validated by the various surveys on learning control [7], [9], [10]. Iterative Learning Control (ILC), which is discussed in this paper, is one subset of learning control. As one definition describes, ILC is "an approach to improve the transient response performance of an unknown/uncertain hardware system that operates repetitively over a fixed time interval by using the previous actual operation data to compensate for uncertainty [1]." ILC is also a popular research subject as described by its own set of survey papers [1], [3].

The primary challenge to ILC is choosing the learning filter which will influence how and what information will be learned. In fact, the learning filter is usually chosen depending on how well the system is known. Proportional (P-) and proportional plus derivative (PD-) type learning functions are popular because they require very little system information. To improve the ILC convergence rate, the learning gains of these two methods are typically tuned. Thus, every adjustment to the learning gains resets the learning process. Three of the remaining four popular learning algorithms: model inversion, $\mathcal{H}_\infty$ and quadratically optimal are model based [3]. The resulting performance is then dependent on fidelity of the model. Q-filters, μ-synthesis and loop shaping approaches are common methods to address model mismatch. Thus, the design process now focuses on how well the model is known. It is still common to tune these methods as there is a balance among ILC robustness, performance and stability.

For model based learning filters which invert transfer functions, it is well known that discrete time systems constructed by sampling continuous time systems (zero order hold method) can exhibit unstable zeros [2]. These systems will result in an unstable inverse which cannot be used for control. This has been addressed for digital tracking systems [17] as well as in ILC [11], [15]. By implementing an FIR filter, we are looking to construct learning filters which avoid having to invert unstable or lightly damped zeros.

There are also challenges with the structure of model based learning filters. Inverting system models will result in an infinite impulse response (IIR) filter acting on a finite interval. This is generally implementable when the time interval is assumed to be very long and data at the end points is carefully handled. For quadratic based learning methods, the learning filters are designed to be matrices with dimensions as long as the time interval [8]. This leads one to question how important the entire time interval is for each individual update. This has given rise to matrix order reduced by singular value decomposition, though a learning matrix structure is still needed [6]. Here, we are looking to develop an FIR learning filter which contain enough model information without the need of IIR filters or learning matrices.

An additional design consideration for feedforward control is determining where to insert the control signal. Feedforward control requires a feedback controller when the plant is unstable. When controlling mechanical systems like servo and linear motors, a feedback controller is commonly needed along with feedforward control. Thus, there are two places to inject a feedforward signal: at the reference and at the control input. There has been recent research studying the effectiveness of these two feedforward control methods. The controller complexity, sensitivity to closed loop dynamics

This work was supported in part by Nikon Research Center of America, and the UC Discovery Grant ele-10197. The authors also graciously thank National Instruments for hardware support

B. Fine is with the University of California, Berkeley. benjamin.t.fine@gmail.com

S. Mishra is with the University of Illinois, Champaign-Urbana sandipan@illinois.edu

M. Tomizuka is with the University of California, Berkeley. tomizuka@me.berkeley.edu

and numerical sensitivity were analyzed in [4] and [5]. In [12], various ILC injection points and learning signals were compared, though only the frequency content of the learning control was analyzed. Both of these works suggest that feedforward control acting on the closed loop system is preferable to injecting control at the plant.

This paper focuses on designing model inverse based ILC algorithms for SISO, LTI systems which have a short filter length and are only noncausal to account for the system delay. In order to do this, we consider the location of the learning algorithm and the model used for learning. We analyze the impulse responses and pole-zero mappings of four models to determine which ILC architecture and system model can generate an easy, implementable FIR learning filter. These learning filters are also implemented on a wafer stage prototype with a position feedback controller. Error convergence of the model based ILC algorithms are then compared to a P-type learning filter.

II. ILC Preliminaries

In this section, we cover the ILC notation used in the remaining sections. Consider the following discrete time, linear time invariant SISO system

$$y_j(k) = G(q)u_j(k) + d(k) \tag{1}$$

where k is the time step and j is the iteration number. The system $G(q)$ is an asymptotically stable, proper rational function of the time shift operator $q : q(u_j(k)) = u_j(k+1)$ with relative degree m and $d(k)$ is an exogenous signal which is iteration invariant. The control objective is to find $u_j(k)$ such that $y_j(k)$ tracks the desired reference $y_d(k)$. At this point, no feedback control has been defined, so $u_j(k)$ is an open-loop control signal.

One common analysis tool used when designing ILC algorithms is to transform (1) into a lifted system representation. This is made possible in part by assuming the reference trajectory is finite and fixed in length. The signals in (1) become

$$\begin{aligned} \mathbf{y}_j &= [y_j(m), y_j(m+1), \ldots, y_j(N+m)]^T \\ \mathbf{u}_j &= [u_j(0), u_j(1), \ldots, y_j(N-1)]^T \\ \mathbf{d} &= [d(m), d(m+1), \ldots, d(N+m)]^T. \end{aligned}$$

Second, $G(q)$ can be expanded as an infinite power series

$$G(q) = g_m q^{-1} + g_{m+1} q^{-2} + g_{m+2} q^{-3} + \ldots.$$

where $\{g_m, g_{m+1}, \ldots\}$ are the impulse response parameters. The relative degree, m, results in an m-step delay between the input and output, so $g_0, ..., g_{m-1} = 0$. Thus, $u_j(k)$ is chosen so $y_j(k+m)$ tracks $y_d(k+m)$ subject to a disturbance $d(k+m)$ where $k \in \{0, 1, ..., N-1\}$. The system matrix $\mathbf{G}$ is lower triangular and Toeplitz

$$\mathbf{G} = \begin{bmatrix} g_m & 0 & \cdots & 0 \\ g_{m+1} & g_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_N & g_{N-1} & \cdots & g_m \end{bmatrix}. \tag{2}$$

Thus, (1) becomes $\mathbf{y}_j = \mathbf{P}\mathbf{u}_j + \mathbf{d}$. Capital and lower case bold letters $(\mathbf{G}, \mathbf{u})$ denote matrices and vectors, respectively, in the lifted domain.

In a batch process, the system is repeatedly attempting to follow the same reference trajectory. Plant based disturbances and transient errors from the feedback controller will repeatedly cause tracking errors. These errors cannot be corrected by conventional feedback control. ILC utilizes information from the previous iteration to be used as feedforward control in the next iteration. The task of the ILC algorithm is to determine the feedforward control input $u_j(k)$. A commonly used ILC algorithm is

$$u_{j+1}(k) = Q(q)[u_j(k) + L(q)e_j(k+m)] \tag{3}$$

where $L(q)$ is the learning filter based on $G(q)$ and a Q-filter is added for robustness. Tracking error, $e_j(k) = y_d(k) - y_j(k)$, is used for the learning update. For this paper, we consider stability and performance issues only in the lifted domain. That is, the ILC algorithm is asymptotically stable if and only if the eigenvalues $\rho(\mathbf{Q}(\mathbf{I} - \mathbf{L}\mathbf{G})) < 1$ and monotonically convergent if the largest singular value

$$\overline{\sigma}\left(\mathbf{G}\mathbf{Q}(\mathbf{I} - \mathbf{L}\mathbf{G})\mathbf{G}^{-1}\right) < 1 \tag{4}$$

which also defines the rate of convergence. Proofs of stability can be found in [14].

The two learning filters used in this paper are P-type and model inversion. In P-type ILC,

$$u_{j+1}(k) = Q(q)[u_j(k) + k_p e_j(k+m)] \tag{5}$$

where the learning filter is a proportional gain and is generally tuned to achieve suitable convergence rates. The model inversion method for ILC attempts to invert the system model in order to achieve perfect tracking.

$$u_{j+1}(k) = Q(q)[u_j(k) + \hat{G}(q)^{-1}(e_j(k+m)] \tag{6}$$

In the lifted domain, (6) is equivalent to choosing $\mathbf{L}$ such that $\mathbf{I} - \mathbf{L}\mathbf{G} = \mathbf{0}$. Equation (4) shows that this choice will result in a maximum singular value of zero meaning the error converges in one iteration.

III. ILC in feedback control

Although $G(q)$ is assumed to be asymptotically stable, its inner structure may contain a feedback controller. That is, assume $C(q)$ is designed to stabilize the plant $P(q)$. Figure 1 shows the general feedback control system. The output equation with no learning control becomes

$$y_j(k) = \frac{PC}{1+PC} y_d(k) + \frac{P}{1+PC} d(k) \tag{7}$$

where the argument q is abbreviated. Feedforward control can be injected at two places in this structure: at the reference and control input. Not only are the feedforward controller structure for these two injection points drastically different [4], the ILC structure is also different. In this section, we analyze the two feedforward structures as well as look at two different modeling techniques for developing the model based ILC algorithm. Using a model for a wafer stage prototype, we simulate the impulse response and pole-zero mappings for each combination.

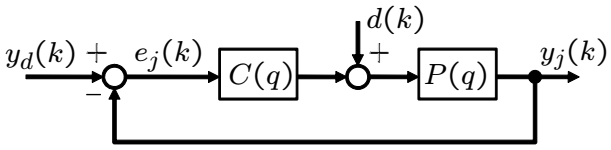

Fig. 1. The system $G(q)$ contains a controller $C(q)$ which stabilizes the plant $P(q)$

A. Two ILC architectures

In the plant injection architecture, shown in Fig. 2, the ILC injection point is placed at the control input. This is commonly called a parallel or plant injection feedforward control because the control acts in parallel to the feedback controller at the control input. In motion control research, having direct control of the plant input is desirable. Feedforward control is used to cancel plant nonlinearities and additional feedback loops like disturbance observers can be used to cancel out low frequency disturbances. The output equation is defined as

$$y_j(k) = \frac{P}{1+PC}\left(d(k) + u_j(k)\right) + \frac{PC}{1+PC} y_d(k) \qquad (8)$$

and the transfer function between the learning control and the output is $\frac{P}{1+PC}$.

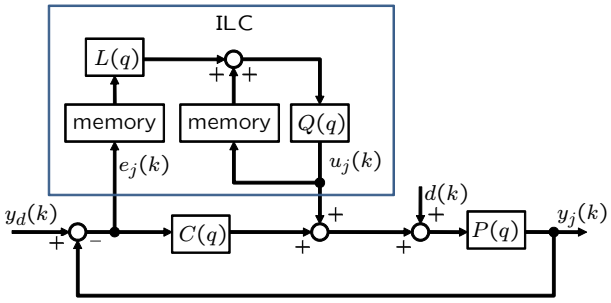

Fig. 2. The plant injection structure has the learning controller act in parallel to the feedback controller and is directly injected into the control input.

Injecting feedforward control into the reference signal is a common alternative when the there is no direct access to the control input. Figure 3 shows the reference injection structure where the ILC injection point is in series with the feedback control.

$$y_j(k) = \frac{P}{1+PC}\left(d(k)\right) + \frac{PC}{1+PC}\left(y_d(k) + u_j(k)\right). \qquad (9)$$

Now, the ILC algorithm is acting on the inverse sensitivity function $\frac{PC}{1+PC}$.

B. Impulse Response characteristics

In order to develop model based learning filters, we first introduce a mass-damper plant and PID feedback controller. This is a continuous time model of wafer stage prototype used for precision positioning applications. The plant model is

$$P(s) = \frac{1}{0.47s^2 + 0.52s} \qquad (10)$$

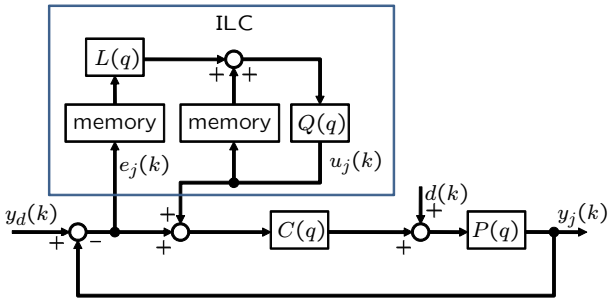

Fig. 3. The reference injection structure acts in series with the feedback control loop and injects the learning signal before the controller.

TABLE I

THE SYSTEM TRANSFER FUNCTIONS ARE ORGANIZED BY THE ILC INJECTION POINT AND SYSTEM MODELING METHOD.

	CT	DT
reference injection: $\frac{PC}{1+PC}$	$G_a(q)$	$G_b(q)$
plant injection: $\frac{P}{1+PC}$	$G_c(q)$	$G_d(q)$

and is stabilized with a continuous time controller

$$C(s) = 50000(1 + 2/s + 0.012s). \qquad (11)$$

Though first modeled in continuous time, implementation and ILC design is performed in discrete time:

$$P(z) = \frac{1.702e-007z + 1.702e-007}{z^2 - 2z + 0.9996} \qquad (12)$$

$$C(z) = 50000(1 + 2\frac{T_s}{1-z^{-1}} + 0.012\frac{1-z^{-1}}{Ts}) \qquad (13)$$

with a sampling time $T_s = 0.0004$ sec. The discrete time plant, $P(z)$, is found using the zero order hold of $P(s)$. The closed loop system transfer functions can then be computed for both ILC architectures.

A second modeling technique is also considered. Here, the continuous time, closed loop systems are first computed and then converted to discrete time. Table I summarizes the four different system models which are being designed: two different ILC architectures and two modeling methods. Before designing learning filters for the four systems (G_a through G_d), we first look at the impulse responses.

The system impulse responses are shown in fig. 4. Although not entirely apparent, all four discrete time systems have a relative degree of one. Also, the impulse responses of the reference injection architecture, (a) and (b), take far fewer time steps to settle than the plant injection (c) and (d). The peak response (a) occurs at the first time step after the delay where (b), (c) and (d) require more than one step to reach the peak amplitude.

These results can be explained by the pole-zero map shown in fig. 5. The figure inserts are a large magnification of the poles and zeros near $z = 1$. It shows that all four systems have some pole-zero cancelations though the relative degree of each system model is still one. In the plant

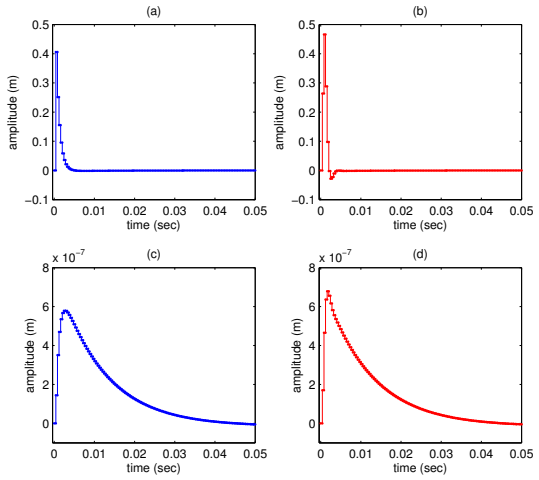

Fig. 4. The impulse response of the systems are separated by injection point and the method by which the transfer function was created.

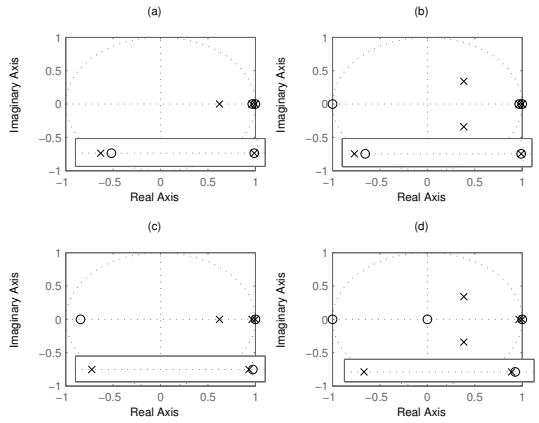

Fig. 5. The poles and zeros are shown for each system model.

injection architectures, there is one lightly damped pole that is uncanceled which causing a slower response.

IV. ILC FIR FILTER DESIGN

Using the lifted domain, ILC based on the model inversion method is equivalent to choosing $\mathbf{L}$ such that $\mathbf{I} - \mathbf{LG} = \mathbf{0}$. One way to compute $\mathbf{L}$ is to realize that $\mathbf{G}$, (2), is a full rank, lower triangular Toeplitz matrix. Thus, its inverse is also a full rank lower triangular Toeplitz matrix. In fact, this matrix can be computed algebraically by only looking at the first column of each matrix. Also note that the first column of each matrix also corresponds to the first N impulse response parameters and contains the largest causal set of impulse response parameters.

The impulse response of $L(q)$ for $q : \{0, 1, \ldots, N-1\}$ is

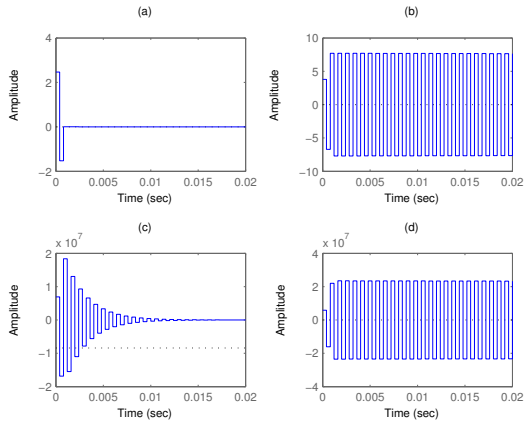

Fig. 6. The impulse response of the learning filters are separated by injection location and modeling approach.

computed as:

$$\begin{aligned} g_m l_0 &= 1 \\ g_{m+1} l_0 + g_m l_1 &= 0 \\ g_{m+2} l_0 + g_{m+1} l_1 + g_m l_0 &= 0 \\ \vdots & \quad \vdots \\ \sum_{n=0}^{N-1} g_{m+N-1-n} l_n &= 0 \end{aligned}$$

The filter impulse response at the hth time step, l_h, is found recursively from previous values of g and l.

$$l_h = \frac{-\sum_{n=0}^{h-1} g_{m+h-n} l_n}{g_m}. \tag{14}$$

Thus, the FIR learning filter with length $h \leq N$ is defined as

$$L(q) = l_0 + l_1 q^{-1} + l_2 q^{-2} + \ldots + l_h q^{-h} \tag{15}$$

The learning filters can now be determined using the systems defined in Section III-B. The learning filter impulse responses which are the FIR filter coefficients are shown in fig. 6. The impulse responses in (b) and (d) do converge, however, they require thousands of time steps to do so. Referring back to fig. 5, the slow convergence of systems (b) through (d) can be seen by observing the lightly damped zero near $z = -1$. These zeros, when inverted, cause the weakly damped impulse responses. Fig. 5(a) is the only model that does not have a zero nearby, and converges in two time steps.

This zero is due to continuous time zeros at $s = -\infty$ being mapped to $z = -1$. Both G_b and G_d were designed using the discrete time equivalent $P(z)$. The continuous time plant $P(s)$ has a relative degree of two which means that $P(z)$ will exhibit the sampling zero effect will have a zero at $z = -1$ [2]. G_c is based on the $P(s)$ and $C(s)$, though the transfer function $\frac{P}{1+PC}$ still maintains the relative degree of two. The discrete time equivalent system will still map at least one continuous time zero to $z = -1$. G_a is the only

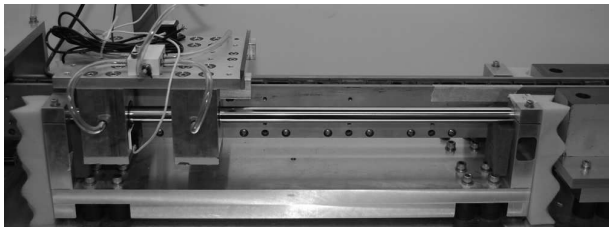

Fig. 7. A linear stage is used for wafer positioning

discrete time system where the discrete time transfer function does not have a lightly damped zero.

It can be seen that $G_a(q)$ will result in the the most desirable learning filter. The filter can be approximated with only two taps since the trailing points are orders of magnitude smaller. Interestingly, fig. $6(a)$ is very much like a PD-type ILC where learning is a function of two time steps: $L(q) = k_p + k_d q^{-1}$. The plant injection architecture which uses the continuous time model, fig. $6(c)$, can also be easily implemented, though considerably longer filters will be needed to approximate the impulse response. In the following section, these two learning filters are implemented and convergence rates are compared as the filter lengths are varied. Additionally, P-type ILC experimental results are provided to compare a model based learning control to one that requires tuning.

V. Experimental Results

ILC algorithms for both reference and plant injection methods were implemented on a linear motor used for wafer stage control [13]. The stage prototype, shown in fig. 7, is supported by air bearings and is modeled using eqns. (10) - (13).

The high controller gains in (11) and (13) reflect the need for a high performance control system. The system model (10) was obtained through sine sweep and iterative feedback tuning methods specifically for a third order trajectory shown in fig. 8 [16]. The scanning velocity is 0.1875 m/s with a maximum acceleration of 5 m/s^2. The length of the scan is 0.1m.

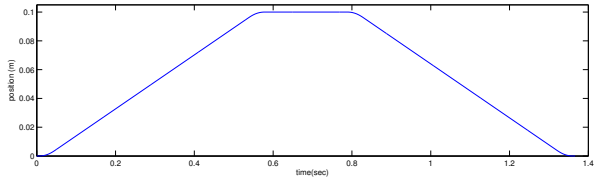

Fig. 8. A third order trajectory is designed for tracking. The scan velocity is 0.1875 m/s.

An 11-tap zero phase Butterworth filter with 150Hz cuttoff frequency was designed for the ILC Q-filter was designed and implemented on all learning controllers. Of the four models, only $G_a(q)$ and $G_c(q)$ were implemented due to the excessive filter length required for the other two models.

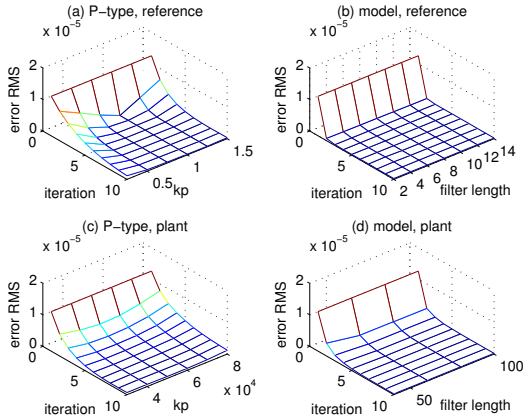

Fig. 9. A comparison of RMS error convergence for different ILC architectures and learning filters

TABLE II

The ILC filter choices for comparison are based on the previous figures.

injection point	P-type, k_p	Model based, filter length
reference	1	14
plant	50000	100

Two more ILC algorithms based on the P-type methods were also compared. These four ILC algorithms:

- plant injection, P-type
- plant injection, model inversion
- reference injection, P-type
- reference injection, model inversion

were compared based on their tuning parameters. For P-type ILC, the single parameter k_p was varied. The length of the model based filter was also varied. The optimal gain for the reference injection P-type learning filter in fig. 9(a) is shown to be $k_p = 1$ while the P-type plant injection ILC in fig. 9(c) is $k_p = 50000$. This gain is also the proportional gain of the feedback controller. The model inversion ILC algorithms are shown in fig. 9(b) and (d). The plant injection architecture shown in (d) requires a considerably longer filter length than the reference injection architecture in (b). For the plant injection architecture, the ILC algorithm became unstable when the filter length was set to 30. For both figures, the convergence rates vary little among different filter lengths.

The error convergence of the best performing ILC algorithm are shown in fig. 10 and learning filter parameters are summarized in Table II. It can be seen that the reference based injection architecture provides superior performance to the plant injection methods. In the plant injection architecture, the model based algorithm considerably outperforms the P-Type ILC algorithm while the two learning filters perform similarly in the reference injection architecture.

These performance differences of the two ILC algorithms is largely due to designing P-type ILC algorithms which model the system inverses. While the plant injection system

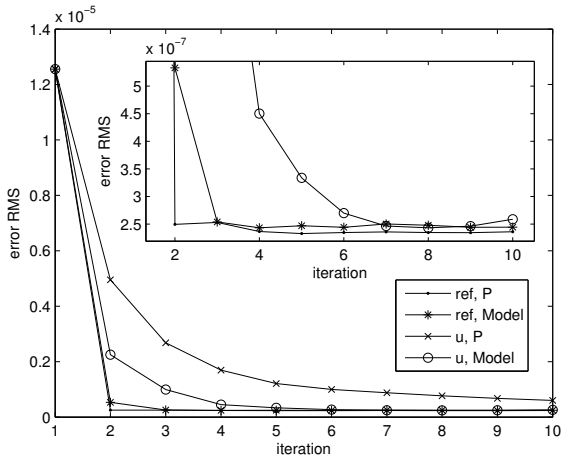

Fig. 10. The four ILC methods are compared

G_c requires at least a 100 tap learning filter, a 14 tap filter is sufficient for the reference injection system G_a. Even more, the P-type ILC algorithms can be considered single tap model based learning filters. As shown in figure in fig. 6(a), the reference injection architecture closely resembles a single tap filter where a single tap filter poorly estimates the plant injection impulse response in (c). This is apparent in fig. 10 as the reference based, P-type ILC algorithm is the slowest to converge.

The system transfer functions also provide insight as to why the reference injection ILC architecture outperforms the plant injection. The reference system is based on the closed loop transfer function $\frac{PC}{1+PC}$ which is designed to have zero gain at zero frequency. Therefore, the inverse also has zero gain at low frequencies. This is makes P-type algorithms with $k_p = 1$ a good choice. Any high frequency differences between the model and the learning filter are dealt with by the Q-filter.

VI. Conclusion

In this paper, two injection architectures and two methods for system modeling have been presented for model inversion based ILC. The ILC algorithms were designed so that the learning filters were finite in length and only noncausal to account for the system delay. The reference based ILC algorithms were shown to have better convergence rates than the plant injection algorithms because the closed loop transfer function (and its inverse) has good low frequency characteristics. Because of the unit gain at low frequency, this resulted in $k_p = 1$ being a choice for P-type ILC. For the plant injection architecture, faster convergence became a function of filter length. It was shown that considerably longer filters were needed to improve convergence though neither P-type nor model based algorithms performed as well as their reference injection counterparts.

In addition to injection location, the model choice also had an effect on ILC learning filter design. The discrete time models based on the closed loop continuous time transfer functions limited the effects of sampling zeros. This allowed for direct implementation of the FIR learning filters. Even more, the reference injection architecture resulted in a discrete time system with no sampling zeros at all. The learning filter closely resembled the two tap PD-type learning filter and needed a far shorter filter for implementation.

References

[1] H.-S. Ahn, Y. Chen, and K. L. Moore. Iterative learning control: Brief survey and categorization. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1099–1121, 2007.

[2] K. Åström and B. Wittenbark. *Computer Controlled Systems: Theory and Design*. Prentice-Hall, 2nd edition, 1990.

[3] D. A. Bristow, M. Tharayil, and A. G. Alleyne. A survey of iterative learning control. *Control Systems Magazine, IEEE*, 26(3):96–114, 2006.

[4] J. A. Butterworth, L. Y. Pao, and D. Y. Abramovitch. Architectures for tracking control in atomic force microscopes. In *Proc. IFAC World Congress*, Seoul, Korea, 2008.

[5] J. A. Butterworth, L. Y. Pao, and D. Y. Abramovitch. The effect of nonminimum-phase zero locations on the performance of feedforward model-inverse control techniques in discrete-time systems. In *American Control Conference, 2008*, pages 2696–2702, 2008.

[6] B. G. Dijkstra and O. H. Bosgra. Extrapolation of optimal lifted system ilc solution, with application to a waferstage. volume 4, pages 2595–2600 vol.4, 2002.

[7] K.-S. Fu. Learning control systems–review and outlook. *IEEE Transactions on Automatic Control*, 15(2):210–221, 1970. 0018-9286.

[8] S. Gunnarsson and M. Norrlf. On the design of ilc algorithms using optimization. *Automatica*, 37(12):2011–2016, 2001.

[9] H. Hjalmarsson. Iterative feedback tuning-an overview. *International Journal of Adaptive Control and Signal Processing*, 16(5):373–395, 2002. 10.1002/acs.714.

[10] R. Horowitz. Learning control of robot manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, 115:402–411, 1993.

[11] K. Kinosita, T. Sogo, and N. Adachi. Iterative learning control using adjoint systems and stable inversion. *Asian Journal of Control*, 4(1):60–67, 2002.

[12] S. Mishra. *Fundamental Issues in Iterative Learning Controller Design: Convergence, Robustness, and Steady State Performance*. PhD thesis, University of California, Berkeley, 2008.

[13] S. Mishra, J. Coaplen, and M. Tomizuka. Precision positioning of wafer scanners segmented iterative learning control for nonrepetitive disturbances. *Control Systems Magazine, IEEE*, 27(4):20–25, 2007.

[14] M. Norrlof and S. Gunnarsson. Time and frequency domain convergence properties in iterative learning control. *International Journal of Control*, 75:1114–1126, 2002.

[15] T. Sogo. Stable inversion for nonminimum phase sampled-data systems and its relation with the continuous-time counterpart. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 4, pages 3730–3735 vol.4, 2002.

[16] H. Stearns, S. Mishra, and M. Tomizuka. Iterative tuning of feedforward controller with force ripple compensation for wafer stage. In *Advanced Motion Control, 2008. AMC '08. 10th IEEE International Workshop on*, pages 234–239, 2008.

[17] M. Tomizuka. On the design of digital tracking controllers. *Journal of Dynamic Systems, Measurement, and Control*, 115:412–418, 1993.