A COMPARATIVE STUDY OF FEEDFORWARD TUNING METHODS FOR WAFER SCANNING SYSTEMS

Hoday Stearns*, Shuwen Yu, Benjamin Fine, Sandipan Mishra, Masayoshi Tomizuka
Department of Mechanical Engineering
University of California
Berkeley, California, 94720
hodayx@berkeley.edu, emmayu@berkeley.edu, sandipan@me.berkeley.edu,
fine@me.berkeley.edu, tomizuka@me.berkeley.edu

ABSTRACT

The use of feedforward control is beneficial for high-performance trajectory tracking in many motion control systems. Three methods of designing and tuning feedforward control signals (Iterative Learning Control, Iterative Controller Tuning, and Adaptive Feedforward Control) for a wafer scanner system are presented and compared. For this application, the main sources of tracking error are due to phase mismatch and nonlinear force ripple disturbance. The objective is to compare the performance of these three methods in compensating for error arising from these sources. The methods are compared based on a set of metrics. Comparison is followed by a discussion on advantages and disadvantages of each method including ability to reduce error during acceleration or scan phases of the trajectory, necessary assumptions, effect of inaccurate modeling,and effect of noise.

INTRODUCTION

Feedforward control has an important role in many industrial positioning systems. In conjunction with feedback control, feedforward control enables plants to achieve more accurate tracking of many motion trajectories than is possible with feedback control alone. Feedforward control allows compensation for errors arising from phase mismatch, in a sense giving the system a faster response to setpoint changes. Additionally, when the same trajectory is to be executed multiple times, feedforward control may be used to compensate for repeated known disturbances. Due to these characteristics, feedforward control is often used in industrial applications today.

This paper presents a comparative study of three different methods of generating a feedforward control signal. We describe three particular methods of feedforward control, namely Iterative Learning Control, feedforward control based on Iterative Feedback Tuning, and Adaptive Feedforward Control. Iterative learning control applies to repetitive processes and involves updating a control law based on data from previous runs of an experiment. Iterative feedback tuning is another iterative method that tries to minimize a cost function by tuning controller parameters. In Adaptive Feedforward Control, plant parameters are estimated in real-time and used to form a control input. These three methods of feedforward control are next implemented on a wafer scanner system, a high-precision machine used in the photolithography process. Then the experimental results from the three methods are compared based on several metrics, such as peak error, 2-norm of error, settling time, control effort, and more. Finally, the results are discussed and interpretations of the findings are offered, including advantages and disadvantages on the three methods. The emphasis of this paper is the comparison of several feedforward control methods through the introduction of a set of metrics by which performance may be compared.

PROBLEM FORMULATION

This paper is concerned with the control of the system shown in Figure 1. The plant is a wafer stage moving along a linear track actuated by a linear permanent magnet motor with feedback information from a laser interferometer and position encoder. The stage is modeled as a single mass with damping

$$m\ddot{y} + b\dot{y} = ku + d \tag{1}$$

*Address all correspondence to this author.

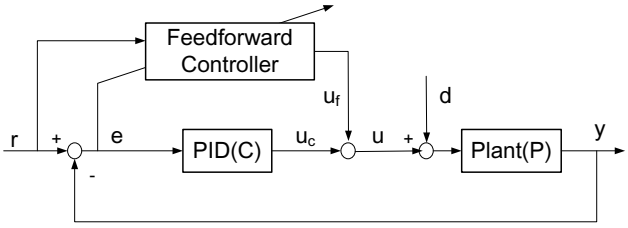

Figure 1. CONTROL SCHEME SETUP.

where y is the position of the stage, m is mass, b is the damping coefficient, u is the voltage applied to the motor actuating the stage, k is a amplifier constant, and d is external disturbances acting on the plant. Therefore, after substituting in the values of identified plant parameters, the transfer function of the plant model becomes

$$P(s) = \frac{11.7}{5.3s^2 + 7.2s}. \tag{2}$$

For evaluation of the feedforward control design techniques, we will set down a basic 2-DOF controller structure shown in Figure 1. The feedback controller C is fixed and designed apriori, considering the bandwidth requirement. A PID controller is used:

$$C(s) = 30000(1 + \frac{2}{s} + 0.012s). \tag{3}$$

The sampling rate is $T_s = 0.0004$ s. The feedforward signal u_f is the design variable.

The process of interest is the scanning operation in a wafer stage. A typical (single) scanning operation requires a trajectory shown in Fig 2. This trajectory consists of two major phases (a) acceleration to a fixed velocity $\nu = 0.1875$ m/s, (b) scanning at constant velocity ν. During the acceleration phase, we expect high frequency components in the reference trajectory, which may result in tracking error due to phase mismatch, especially close to the bandwidth.

One of the considerations when designing the feedforward signal u_f is the nature of the disturbance d. In a linear permanent magnet motor, the main sources of disturbance include nonlinear forces such as friction force, force ripple, and unmodeled uncertainties. We assume nonlinear friction force is negligible as the stage is supported by air bearings. This means force ripples become the dominant disturbance that causes the tracking error in the constant velocity phase. There has been significant research as well as industrial thrust on compensating the undesired disturbances. An accurate mathematical expression of force ripple [1] is too complicated to implement. A simplified model of force ripples is shown in Eqn. (4) with the assumption that the permanent magnets are equally aligned at pitch of P

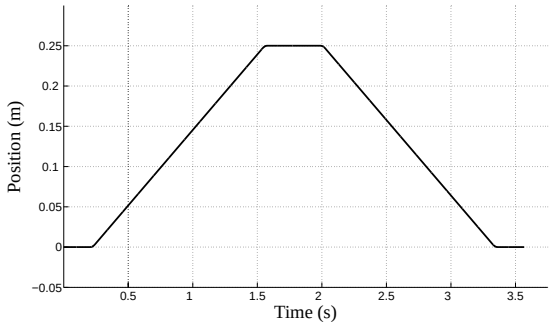

Figure 2. REFERENCE TRAJECTORY.

$$F_{rip}(y) = \sum_{k=1}^{N} A_k cos(k\frac{2\pi}{P}y) + B_k sin(k\frac{2\pi}{P}y) \tag{4}$$

where A_k and B_k are unknown parameters which can be estimated by several methods [2] [3], and N is the number of frequency components used to approximate the force ripple. However, force ripples are not only position dependent but also current dependent [4]. Lee et. al. [5] pointed out that the amplitude of force ripple is highly dependent on position and velocity (which is proportional to the current). A reasonable approximation is suggested and validated in [6] is:

$$F_{rip}(y) = u\left(\sum_{k=1}^{N} A_k cos(k\frac{2\pi}{P}y) + B_k sin(k\frac{2\pi}{P}y)\right). \tag{5}$$

Metrics for Comparison of Controllers

The relative advantages and disadvantages of the three different methods will be compared based on the following metrics:

1. Peak error during constant velocity scan phase.
2. Root mean squared error during constant velocity scan phase.
3. Settling time for velocity error to within 0.01ν(scan velocity).
4. Peak position error during acceleration phases.
5. Mean squared sensitivity function.

$$M_s = \int_0^{\omega_{Nyq}} \left|\frac{E(j\omega)}{R(j\omega)}\right|^2 d\omega \tag{6}$$

6. Control effort energy concentrated beyond bandwidth

$$E = \int_{\omega_B}^{\omega_{Nyq}} |U(j\omega)|^2 d\omega E_f = \int_{\omega_B}^{\omega_{Nyq}} |U_f(j\omega)|^2 d\omega \tag{7}$$

7. Peak feedforward and overall control effort.
8. Computational complexity.

ITERATIVE LEARNING CONTROL

Iterative Learning Control (ILC) is a feedforward control strategy used to improve performance of systems that operate repetitively over a fixed time interval. Based on information gathered about the process from previous iterations of the process, performance in the current cycle can be improved. ILC was originally developed for robot learning and training by Arimoto [7] [8]. Since then it has found applications in many industrial processes [9]. ILC is especially attractive because of its simplicity of design, implementation and analysis. Since the scanning process is repetitive, the ideal feedforward signal can be generated through iterative refinement.

The ILC scheme for generating the feedforward signal is shown in Figure 3. From the figure, we have

$$y_k(j) = H_r(q)r(j) + H_{u_f}(q)u_{f,k}(j) \tag{8}$$

$$e_k(j) = r(j) - y_k(j) \tag{9}$$

$$H_r(z) = \frac{P(z)C(z)}{1+P(z)C(z)} \tag{10}$$

$$H_{u_f}(z) = \frac{P(z)}{1+P(z)C(z)} \tag{11}$$

The subscript k reflects the iteration number of the process. The goal of the ILC scheme is to obtain the ideal $u_f(j)$.

Standard ILC Problem Formulation

The standard *P-type* ILC design problem can then be formulated as:

$$u_{f,k+1}(j) = u_{f,k}(j) + \alpha e_k(j+n_d) \tag{12}$$

where α must be designed so that the ILC loop is stable. This is guaranteed if $|1-\alpha \cdot h(n_d)| < 1$, where $h(\cdot)$ is the impulse response of the system from $u_f(j)$ to $y_k(j)$, i.e. $H_{u_f}(z)$, and n_d is the index of the first nonzero impulse response, i.e. the delay of the system.

For analysis of the general ILC scheme, we introduce a lifted formulation of the ILC problem below [10]. Lifting refers to stacking up the various signals in a cycle into vectors of dimension $\Re^N$, where N is the period of the (discrete) repetitive process. The advantage of lifting lies in the fact that a linear time varying (or time invariant) dynamic system can be converted into a static $N \times N$ matrix equation. Although the dimension of this matrix is large (typically orders can reach $N \approx 10000$), it is easy to design, analyze and categorize the behavior the lifted system from iteration to iteration. We can rewrite the above equations in lifted form by stacking all the signals into $N \times 1$ vectors [11].

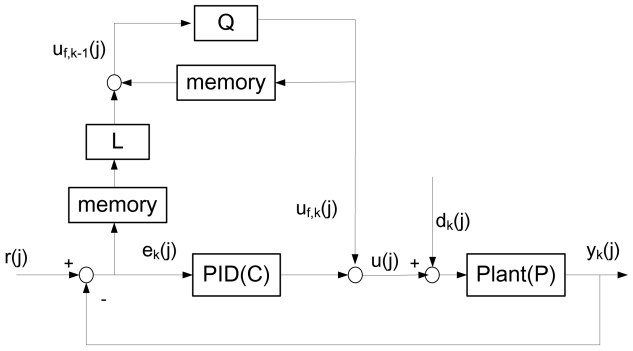

Figure 3. ILC CONTROL SCHEME.

Assuming zero initial conditions, we get the following lifted formulation of the ILC system.

$$\begin{aligned} \mathbf{y}_k &= T_r\mathbf{r} + T_{u_f}\mathbf{u}_f && \text{Lifted Plant Equations} \\ \mathbf{e}_k &= \mathbf{r} - \mathbf{y}_k && \text{Lifted Error Equations} \\ \mathbf{u}_{f,k+1} &= \mathbf{u}_{f,k} + \alpha\mathbf{N}^{n_d}\mathbf{e}_k && \text{Lifted Learning Equations} \end{aligned} \tag{13}$$

where $\mathbf{N}$ is a nilpotent matrix and T_r, T_{u_f} are Toeplitz matrices as below

$$T_r = \begin{bmatrix} g(0) & 0 & \dots & 0 \\ g(1) & g(0) & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ g(N-1) & g(N-2) & \dots & g(0) \end{bmatrix} \tag{14}$$

where $g(i)$ is the i^{th} term of the impulse response of $H_r(z)$.

$$T_{u_f} = \begin{bmatrix} h(0) & 0 & \dots & 0 \\ h(1) & h(0) & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ h(N-1) & h(N-2) & \dots & h(0) \end{bmatrix} \tag{15}$$

where $h(i)$ is the i^{th} term of the impulse response of $H_{u_f}(z)$.

In a more general ILC design case, we have

$$\mathbf{u}_{f,k+1} = \mathbf{u}_{f,k} + L\mathbf{e}_k \text{ Lifted Learning Equation} \tag{16}$$

The ILC scheme is stable if $\rho\left(I_N - T_{u_f}L\right) < 1$. Further, the ILC scheme is *monotonically stable* in the sense of the 2-norm in $\Re^N$ if $\bar{\sigma}\left(I_N - T_{u_f}L\right) < 1$. This is a desirable property to have if we wish to avoid poor learning transients.

In order to separate out non-repetitive disturbances and improve robustness to model uncertainty, the learning law is modified using a Q-filter, as below.

$$\mathbf{u}_{f,k+1} = Q(\mathbf{u}_{f,k} + L\mathbf{e}_k) \text{ Lifted Learning Equation} \tag{17}$$

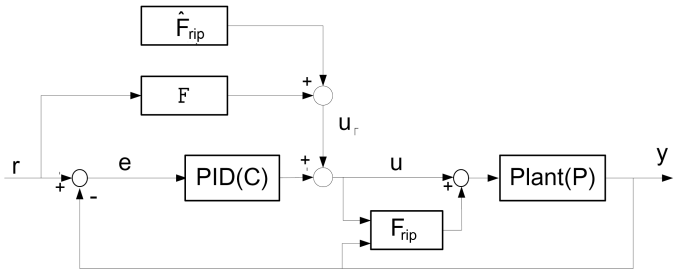

Figure 4. IFT CONTROL SCHEME.

The filtered ILC scheme is stable if $\rho\left(Q(I_N - LT_{u_f})\right) < 1$.

The design of the learning matrices Q and L are based on robustness and convergence specification for the system.

ITERATIVE CONTROLLER TUNING

The second method of designing a feedforward control input is based on iterative feedback tuning (IFT). IFT is a method of automatically tuning controller parameters iteratively so as to minimize a certain cost function [12], [13], [14]. Normally, such an optimization would require a complete model of the plant and a full-order controller, but in IFT, the optimization is performed iteratively by using an estimate of the gradient of the cost function that is calculated based on data collected from experiment iterations. IFT has been applied to tune the controller of a wafer stage in [15].

Here we use an iterative controller tuning method to tune a fixed-structure feedforward controller for the wafer stage. Controller parameters are updated iteratively to minimize a quadratic function of tracking error. In addition, a fixed-structure feedforward controller for compensation of force ripple is also tuned.

Controller Tuning Methodology

IFT is applied to iteratively tune the feedforward controller only of the wafer stage. The structure of the feedforward control scheme is shown in Fig. 4. The feedforward controllers we wish to tune are represented by $F(s)$ and the feedforward force ripple compensator $\hat{F}_{rip}$. $F(s)$ is the feedforward controller whose structure is the inverse of the plant model,

$$F(s) = \hat{\theta}_1 s^2 + \hat{\theta}_2 s \tag{18}$$

where $\hat{\theta}_1$ is the parameter that corresponds to mass and $\hat{\theta}_2$ corresponds to the damping coefficient.

In addition, we added a second feedforward controller $\hat{F}_{rip}$ to compensate for force ripple disturbance. Force ripple is modeled as in Eqn. (5). We desire to cancel out the force ripple by additive feedforward control of the form

$$\hat{F}_{rip}(t) = k_c v(t) \left[\sum_{k=1}^{N} \hat{A}_k \sin(k\frac{2\pi}{p} r(t)) + \sum_{k=1}^{N} \hat{B}_k \cos(k\frac{2\pi}{p} r(t)) + \hat{\gamma}_0 + \hat{\gamma}_1 r(t) \right] \tag{19}$$

where $r(t)$ is reference position, $v(t)$ is reference velocity, and k_c is a constant, and $\hat{A}_k$, $\hat{B}_k$, and $\hat{\gamma}_k$ are the parameters to be tuned. The reason for approximating $y(t)$ and $u(t)$ in Eqn. (5) by $r(t)$ and $k_c v(t)$ is so that the feedforward data u_f can be computed offline.

The objective of controller tuning is to find the values of the parameters $\hat{\theta}_1$, $\hat{\theta}_2$, $\hat{A}_k$, $\hat{B}_k$, and $\hat{\gamma}_k$ such that a cost function is minimized. The cost function we have chosen is

$$J(\hat{\theta}) = \int_0^{t_f} e(t)^2 dt \tag{20}$$

where $e(t) = r(t) - y(t)$ is the error, and $\hat{\theta} = [\hat{\theta}_1\ \hat{\theta}_2\ \hat{A}_1 ... \hat{A}_N\ \hat{B}_1 ... \hat{B}_N\ \hat{\gamma}_k]^T$ is a vector of controller parameters.

The values for the controller parameters are updated iteratively in a gradient-based search based on the error profile from the previous iteration. The update equation is

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \gamma R^{-1} \left. \frac{\partial J}{\partial \hat{\theta}} \right|_{\hat{\theta}_k} \tag{21}$$

where $\hat{\theta}_k$ is the parameter estimate in the kth iteration, γ is a scalar parameter to control step size, R is a matrix to modify the search direction, and $\frac{\partial J}{\partial \theta}$ is the gradient of the cost function with respect to controller parameters evaluated at the present value of $\hat{\theta}$.

The gradient $\frac{\partial J}{\partial \theta}$ is estimated from data collected in experimental runs. The derivation of the formula for the gradient estimate can be found in [16].

ADAPTIVE FEEDFORWARD CONTROLLER FOR FORCE RIPPLE COMPENSATION

In this section, an adaptive feedforward controller for online force ripple estimation and compensation is described. Figure 5 shows the overall control structure, which consists of a feedback controller and an adaptive feedforward compensator. In the adaptive feedforward block, a parameter adaptation algorithm is used to estimate the unknown weighting parameters of force ripple and the feedforward controller is used to reduce the effects of the force ripple.

The adaptive algorithm for the feedforward controller is described in the following. First, a sliding surface is defined as

$$S = \dot{e} + \lambda e \tag{22}$$

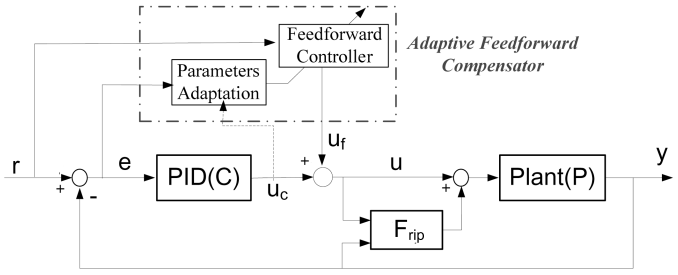

Figure 5. BLOCK DIAGRAM OF OVERALL CONTROL SCHEME.

where $e(t) = r(t) - y(t)$ is the tracking error, r and y represent the desired and measurement position, and λ is any positive gain which provides the stability of the sliding surface. By noting Eqn. (1) and differentiating Eqn. (22), we get:

$$\dot{S} = \ddot{r} - \frac{1}{M}(ku - F_{rip}) + \lambda(\dot{r} - \dot{y}) \tag{23}$$

Here, the control input u is expressed as $u = u_c + u_f + \nu$ where u_c represents the feedback control input, ν the synthesis control input, and u_f the feedforward control input. The aim of the feedback controller is to achieve good transient response and enhance robustness of the system. ν is set to $-\eta S$, so as to avoid chattering effects. The aim of the feedforward control input is to eliminate the nonlinear force ripple. Perfect compensation occurs when the feedforward input, u_f, is equal to actual force ripples. Therefore, we designed the feedforward input as:

$$u_f = \frac{1}{k} u \hat{\theta}^T \phi(r) \tag{24}$$

where $\hat{\theta} = [\xi, \hat{A_1}, \hat{B_1}, ..., \hat{A_N}, \hat{B_N}]$, $\omega = \frac{2\pi}{P}$, and $\phi(r) = [1, \cos(\omega r), \sin(\omega r), ..., \cos(N\omega r), \sin(N\omega r)]$. The additional term, ξ, is utilized to eliminate the bias estimation of $\hat{\theta}$. By a standard adaptation law, $\hat{\theta}$ is updated by Eqn. (25).

$$\dot{\hat{\theta}} = -\frac{1}{\rho} u_c \phi(y) S \tag{25}$$

where ρ is the parameter which is determined by the rate of convergence. For the digital implementation, the continuous adaptation law is approximated by discrete adaptation law. To assure that the estimated parameters lie in the predetermined range, $\hat{\theta}$ should satisfy the condition defined in Eqn. (26).

$$\hat{\theta}_i[(k+1)T_s] = \begin{cases} \theta_{iMIN}, \; if \; \hat{\theta}_i[(k+1)T_s] < \theta_{iMIN} \\ \theta_{iMAX}, \; if \; \hat{\theta}_i[(k+1)T_s] > \theta_{iMAX} \end{cases} \tag{26}$$

where T_s is the sampling time, θ_{iMIN} is the minimum boundary value, and θ_{iMAX} is the maximum boundary value for the i^{th} parameter of the sinusoidal harmonics.

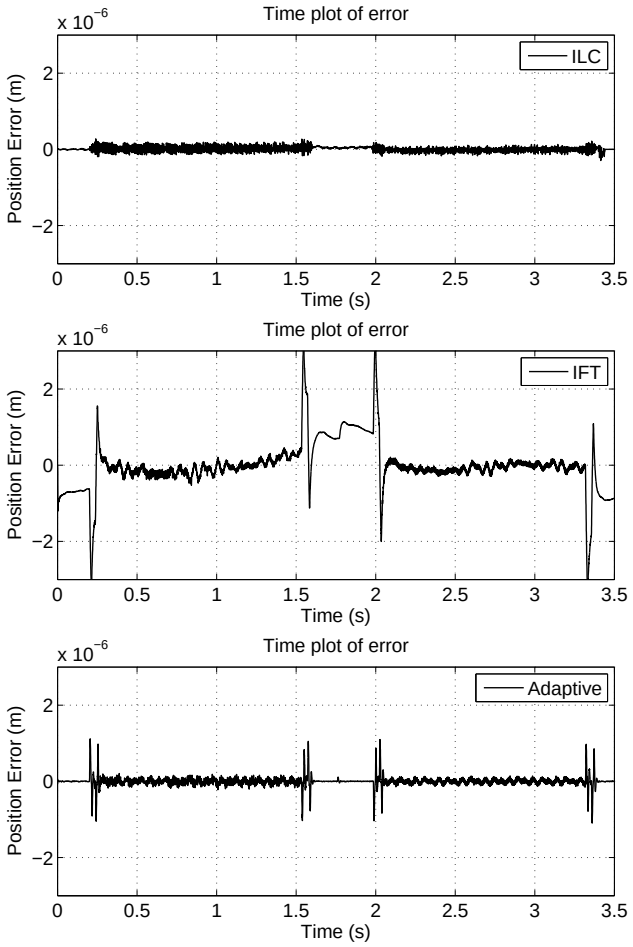

Figure 6. COMPARISON OF TRACKING ERROR.

COMPARISON OF FEEDFORWARD CONTROL METHODS

The three different methods of feedforward control generation, ILC, IFT, and adaptive control, are compared in terms of the metrics described in the previous section. The results are compiled in Table 1. The error signals and control effort signals of the three methods are plotted and compared in figures 6, 7, and 8. The results shown for ILC are from after 9 cycles of learning, and the results for IFT are after 10 cycles. The adaptive control scheme is not an iterative scheme so the results shown are from a single independent run.

Upon examination of Table 1, it can be seen that ILC has the best performance in terms of metrics 1, 4, 7b, and 8. ILC was the most effective in eliminating error during the acceleration phase and also peak error during the constant velocity scan phase. However, ILC produces a control input signal that is the noisiest of the three methods, as can be seen from visual inspection of Figure 8. The input signal has the highest energy content beyond bandwidth (row 6b). It also had the highest mean squared sensitivity (row 5). However, the ILC scheme has the lowest

Table 1. COMPARISON OF FEEDFORWARD CONTROL METHODS

	ILC	IFT	Adaptive
1. Peak error during scan phase (m)	1.83×10^{-7}	5.25×10^{-7}	2.18×10^{-7}
2. RMS error during scan phase (m)	1.4×10^{-9}	3.28×10^{-9}	1.18×10^{-9}
3. Settling time (s)	0	0.063	0.056
4. Peak error (m)	2.92×10^{-7}	3.46×10^{-6}	1.12×10^{-6}
5. Mean squared sensitivity	1.88×10^{-12}	8.78×10^{-13}	1.16×10^{-12}
6a. Control effort energy beyond bandwidth (overall) (V^2)	3.89×10^4	4.32×10^4	4.78×10^4
6b. Control effort energy beyond bandwidth (ff only) (V^2)	1.03×10^4	6.36×10^2	7.74×10^2
7a. Peak control effort (overall) (V)	2.51	2.47	2.46
7b. Peak control effort (feedforward only) (V)	2.44	2.44	2.47
8. Computational complexity (number of operations)†	$2NL$	$(20N + 210)L$	$40N$

† Computed in terms of number of scalar additions and multiplications. N = number of samples, L = number of iterations

computational complexity required per iteration out of the three methods. Further, ILC makes no assumptions about the structure of the plant nor the disturbances and so is a very general method that can be applied in a wide variety of situations.

IFT produces a smooth control input signal (Figure 8). Of the three methods, the feedforward control input of IFT contains the lowest high frequency energy (row 6b) and avoids exciting high frequency resonances, which results in a low mean squared sensitivity value (row 5). However, it performed the poorest in terms of reducing peak error during scan phase, acceleration phase, and RMS error (row 1,2,4). The IFT feedforward signal is overly simple and isn't effective in compensating error. In order to apply IFT, it is necessary to make assumptions about the plant structure and disturbance structure, and inaccurate modeling of both will degrade the potential performance of IFT. IFT performance may be improved by increasing the accuracy of modeling and complexity of the feedforward controller. In addition, IFT has higher computation complexity than ILC. However, one advantage of IFT is that the feedforward controller is applicable even for different reference trajectories, while in ILC, the feedforward signal is only valid for one particular trajectory (However, the force ripple compensator signal is not applicable to different trajectories).

An adaptive algorithm achieved good performance in terms of reducing peak error and RMS error, and in fact achieved the best reduction of RMS error during the scan phase (row 2). The adaptive method is the most effective in eliminating error during the constant velocity scan phase arising from force ripple. As can be seen in Figure 8, the adaptive control input looks similar to the ILC input but is less noisy. Consequently, the control effort energy beyond bandwidth of the ff signal is lower than ILC (row 6b). Another advantage of the adaptive method is that high performance is achieved in only one run, instead of after multiple iterations needed in ILC and IFT.

CONCLUSION

Three different methods for generation of feedforward control inputs, ILC, IFT, and adaptive control, were presented. The performances of the three methods were evaluated based on a set of criteria that considered peak error, RMS error, sensitivity, peak control effort, control effort energy beyond bandwidth, and computational complexity. ILC has the least computational complexity of the three methods and was the most effective and reducing peak errors, but produced a noisy control signal and error. IFT performed the worst at reducing both peak error and RMS error, but produced a smooth signal with low high-frequency component. Adaptive control performed almost as well as ILC at reducing peak error and was the most effective at reducing RMS error during the constant velocity scan phase.

ACKNOWLEDGMENT

This research is made possible through the support of Nikon Research Corporation of America and the U.C. Discovery Grant.

REFERENCES

[1] B. Armstrong-Helouvery, P. D., and de Wit, C. C., 1994. "A survey of models, analysis tools and compensation methods for the control of machines with friction". In Automatica, Vol. 30, pp. 1083–1138.

[2] Tan, K., and Zhao, S., 2002. "Adaptive force ripple suppression in iron-core permanent magnet linear motors". In

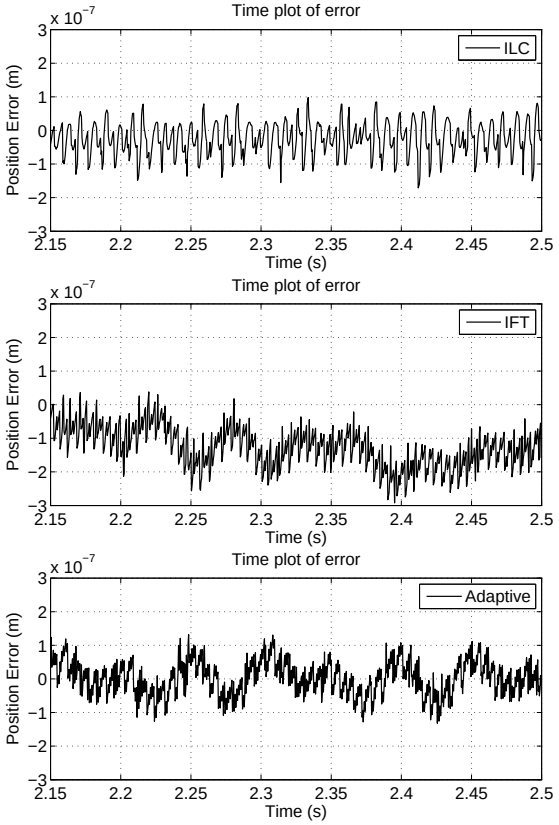

Figure 7. COMPARISON OF TRACKING ERROR DURING SCAN (MAGNIFICATION OF FIG. 6)

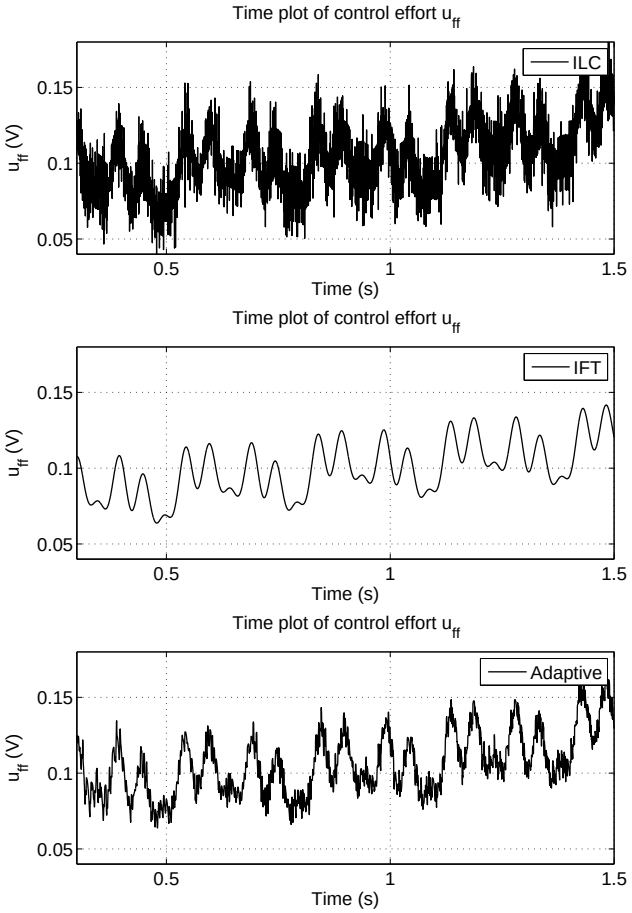

Figure 8. COMPARISON OF CONTROL EFFORT.

Proceedings of the 2002 IEEE International Symposium on Intelligent Control.

[3] Xu, L., and Yao, B., 2000. "Adaptive robust precision motion control of linear motors with ripple force compensations: Theory and experiments". In Proceedings of the 2000 IEEE International Conference on ControlApplications.

[4] Rohrig, C., and Jochheim, A., 2001. "Identification and compensation of force ripple in linear permanent magnet motors". In American Control Conference, 2007., Vol. 3, pp. 2161–2166.

[5] Tan, K.K., H. S., and Lee, T., 2002. "Robust adaptive numerical compensation for friction and force ripple in permanent-magnet linear motors". In IEEE Transactions on Magnetics, Vol. 38, pp. 221–228.

[6] Yu, S., Mishra, S., and Tomizuka, M., 2007. "On-line force ripple identification and compensation in precision positioning of wafer stage". In ASME International Mechanical Engineering Congress and Exposition, 2007.

[7] Arimoto, S., Kawamura, S., and Miyazaki, F., 1984. "Bettering operation of robots by learning". *J. of Robotic Systems*, **1**(2), pp. 123–140.

[8] Togai, M., and Yamano, O., 1985. "Analysis design of an optimal learning control scheme for industrial robots: a discrete system approach". In Proc. of the 24th Conf. on Decision and Control, pp. 1399–1404.

[9] Garimella, S., and Srinivasan, K., 1998. "Application of iterative learning control to coil-to-coil control in rolling". *IEEE Trans. on Cont. Sys. Tech*, **6**(2), pp. 281–293.

[10] Gunnarsson, S., and Norrlof, M., 2001. "On the design of ilc algorithms using optimization". *Automatica*, **37**(1), pp. 2011–2016.

[11] Dijkstra, B., and O.H.Bosgra, 2002. "Extrapolation of optimal lifted system ilc solution, with application to a waferstage". *American Control Conference*, pp. 2595 – 2600.

[12] Hjalmarsson, H., Gunnarsson, S., and Gevers, M., 1994. "A convergent iterative restricted complexity control design scheme". In Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on, Vol. 2, pp. 1735–1740.

[13] Hjalmarsson, H., Gevers, M., Gunnarsson, S., and Lequin, O., 1998. "Iterative feedback tuning: theory and applications". *IEEE Control Systems Magazine*, **18**(4), Aug., pp. 26–41.

[14] Hjalmarsson, H., 2002. "Iterative feedback tuning-an overview". *Int.J.Adapt.Control Signal Process*, **16**,

pp. 373–395.

[15] van der Meulen, S., Tousain, R., and Bosgra, O., 2007. “Fixed structure feedforward controller tuning exploiting iterative trials, applied to a high-precision electromechanical servo system”. In American Control Conference, 2007. ACC ’07, pp. 4033–4039.

[16] Stearns, H., Mishra, S., and Tomizuka, M., 2008. “Iterative tuning of feedforward controller with force ripple compensation for wafer stage”. In Proceedings of 10th International Conference on Advanced Motion Control.