Iterative Tuning of Feedforward Controller with Force Ripple Compensation for Wafer Stage

Hoday Stearns
Department of Mechanical Engineering
University of California, Berkeley
Email: hodayx@berkeley.edu

Sandipan Mishra
Department of Mechanical Engineering
University of California, Berkeley
Email: sandipan@me.berkeley.edu

Masayoshi Tomizuka
Professor
Department of Mechanical Engineering
University of California, Berkeley
Email: tomizuka@me.berkeley.edu

***Abstract*— Iterative controller tuning is a method to fine-tune controllers in a repetitive process using only data collected in experiment runs. The controller parameters are chosen so as to minimize a certain cost function. The values of the parameters are updated iteratively based on tracking error in a gradient search. Iterative controller tuning is an advantageous method of controller tuning because no prior plant model is necessary. In this paper, we iteratively tune the feedforward controller of a wafer stage of a photolithography machine. A fixed-structure feedforward controller consisting of the inverted plant model was tuned. In addition, an additive feedforward term for compensation of nonlinear force ripple disturbance was tuned. The feedforward controller is tuned with the objective of minimizing a cost function that is a quadratic function of tracking error. Simulation and experimental results are presented which show that the iterative tuning method effectively reduced tracking error.**

I. Introduction

Photolithography is a step in the fabrication of integrated circuits (ICs). In photolithography, a light source is used to transfer a pattern from a mask onto a chemical photoresist on a silicon wafer. The positioning of the wafer under the mask and light source is done by a wafer stage. The stage performs a step and scan motion to transfer patterns from the mask to the wafer. Higher precision in the positioning of the stage will enable smaller feature sizes to be written. Therefore, improving the speed and accuracy of control of wafer stages will increase the throughput of the stage and enable more ICs to be printed on a wafer.

Iterative methods have been widely used to improve the performance of repetitive processes such as the operation of wafer stages. Past work includes the use of iterative learning control for wafer stages [1] [2] and iterative design of point-to-point input trajectories [3]. In particular, in order to enhance the performance of such high-precision positioning systems, it is desirable to fine-tune the controllers. Iterative feedback tuning (IFT) is a method of automatically tuning controllers [4], [5], [6]. The IFT algorithm iteratively updates the controller parameters so as to minimize a certain cost function. 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. The IFT method has been previously used to tune controllers of systems such as chemical plants, DC-servo systems [6], two-mass torsional motor systems [7], and two-mass spring systems with friction [8]. IFT was applied to tune the feedforward controller of a wafer stage in [9].

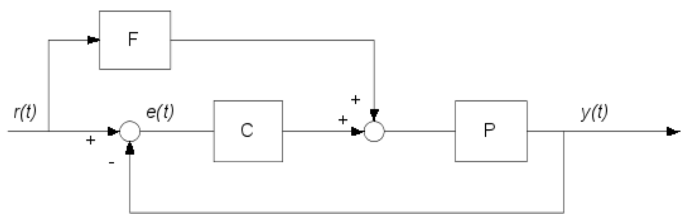

Fig. 1. Block Diagram

In this paper, we use an iterative controller tuning method to tune a fixed-structure feedforward controller for a wafer stage. Controller parameters are updated iteratively to minimize a quadratic function of tracking error. Moving beyond the contribution of [9], we also include tuning of a fixed-structure feedforward controller for compensation of force ripple. Force ripple is a nonlinear periodic disturbance force caused by irregularity in magnet position and inaccuracy of commutation in Linear Permanent Magnet Motors (LPMM) [10]. There exist many control methods for compensating for force ripple. A robust disturbance-observer based approach was developed in [11]. Adaptive control was used for feedforward compensation in [12], [13]. Additive feedforward control of a pre-identified model of the force ripple was used in [14]. Iterative learning control has been applied [15] as well as learning control using neural networks [16]. This paper shows that iterative tuning of a force ripple compensation controller is also an effective way to reduce error caused by force ripple disturbance.

Simulation and experimental results are presented to demonstrate the effectiveness of this method. The results indicated that iterative controller tuning successfully reduced the amount of error and resulted in convergence of controller parameters. Since we are tuning only feedforward controllers, we will forego stability analysis of the closed loop, but we qualitatively consider convergence of controller parameters between iterations.

II. Controller Tuning Methodology

We applied IFT to iteratively tune the feedforward controller of the wafer stage. The block diagram of the wafer stage control system is shown in Figure 1. $P(s)$ represents the transfer function of the wafer stage, which is modeled as a simple mass with damping system

$$P(s) = \frac{k}{ms^2 + bs}.$$

$C(s)$ represents the feedback controller, which in this case is a PID controller. $F(s)$ is the feedforward controller which we wish to tune,

$$F(s) = \rho_1 s^2 + \rho_2 s$$

where ρ_1 is the parameter that corresponds to mass and ρ_2 corresponds to the damping coefficient.

The objective of controller tuning is to find the values of the parameters ρ_1 and ρ_2 such that a cost function is minimized. For the cost function, we have chosen

$$J(\rho) = \int_0^{t_f} e(t)^2 dt \qquad (1)$$

where $e(t) = r(t) - y(t)$ is the error, and $\rho = [\rho_1\ \rho_2]^T$ 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

$$\rho^{k+1} = \rho^k - \gamma R^{-1} \left.\frac{\partial J}{\partial \rho}\right|_{\rho^k}$$

where ρ^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 $\frac{\partial J}{\partial \rho}$ is the gradient of the cost function with respect to controller parameters evaluated at the present value of ρ.

The gradient is estimated by calculations using experimental data. The derivation of the formula for the gradient estimate is as follows. Repeating equation 1, the cost function $J(\rho)$ is given by

$$J(\rho) = \int_0^{t_f} e(t)^2 dt = \int_0^{t_f} (r(t) - y(t))^2 dt.$$

Therefore the gradient is

$$\frac{\partial J}{\partial \rho} = -2 \int (r(t) - y(t)) \frac{\partial y}{\partial \rho} dt. \qquad (2)$$

In the above equation, the unknown quantity that must be calculated is $\frac{\partial y}{\partial \rho}$. Referring to the block diagram in Figure 1, the output $y(t)$ is written as

$$\begin{aligned} y &= \frac{PC}{1+PC} r + \frac{PF}{1+PC} r \\ &= \frac{P(F+C)}{1+PC} r \end{aligned}$$

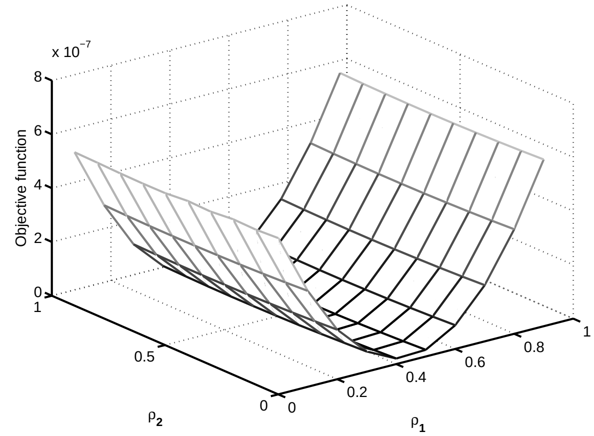

Fig. 2. Cost as a function of $\rho_1 \rho_2$

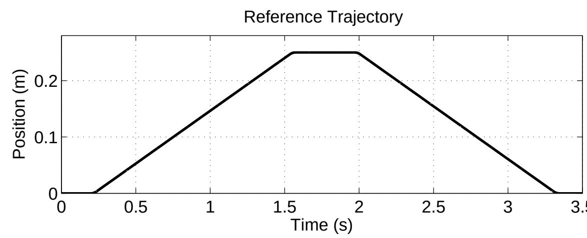

Fig. 3. Reference trajectory for wafer stage

which gives

$$\begin{aligned} \frac{\partial y}{\partial \rho} &= \frac{P}{1+PC} \frac{\partial F}{\partial \rho} r \\ &= \frac{P}{1+PC} \begin{bmatrix} s^2 \\ s \end{bmatrix} r \\ &= \begin{bmatrix} \frac{d^2}{dt^2} \frac{1}{C} \left(\frac{PC}{1+PC} \right) r \\ \frac{d}{dt} \frac{1}{C} \left(\frac{PC}{1+PC} \right) r \end{bmatrix} \\ &= \begin{bmatrix} (y/C)'' \\ (y/C)' \end{bmatrix} \end{aligned}$$

where $(\cdot)'$ and $(\cdot)''$ denote the first and second time derivatives of $(\cdot)$. Substituting this expression for $\frac{\partial y}{\partial \rho}$ into Equation 2, we have

$$\frac{\partial J}{\partial \rho} = -2 \int_0^{t_f} e(t) \begin{bmatrix} (y/C)'' \\ (y/C)' \end{bmatrix} dt.$$

The interpretation of this equation is that the update of the ρ_1 parameter is proportional to the product of the error and $(y/C)''$, and update of ρ_2 is proportional to the product of the error and $(y/C)'$. We can think of this as the projection of the error profile onto the two directions $(y/C)''$ and $(y/C)'$.

III. Controller Tuning Results

A. Simulation

The controller tuning algorithm was first implemented in simulation. The model of the plant used for simulation is

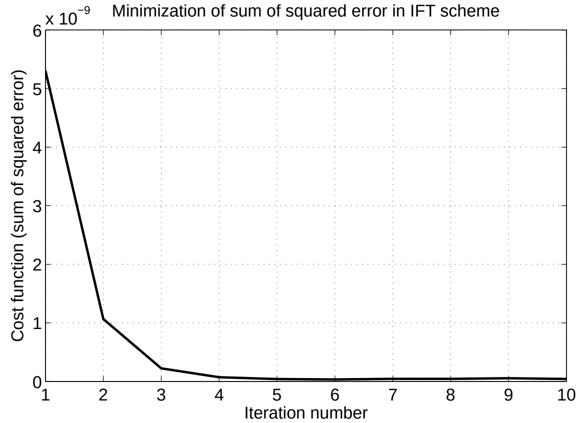

Fig. 4. Minimization of cost function (sum of squared error) in simulation over 10 cycles of tuning

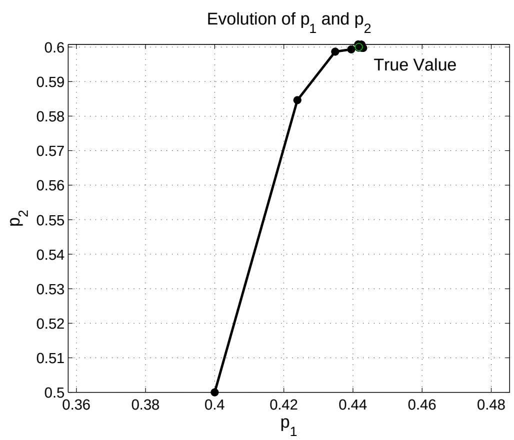

Fig. 5. Convergence of feedforward parameters in simulation

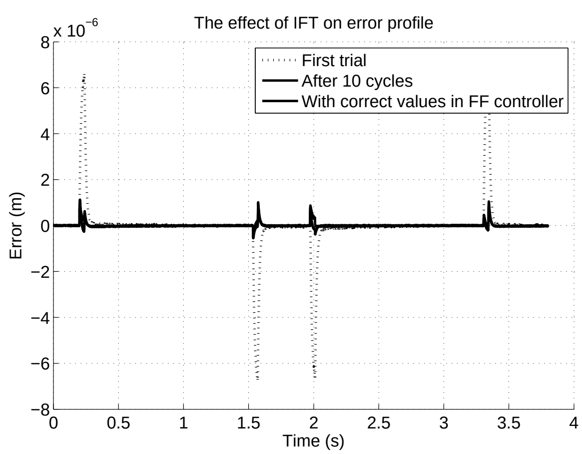

Fig. 6. Reduction of error after 10 cycles of tuning in simulation

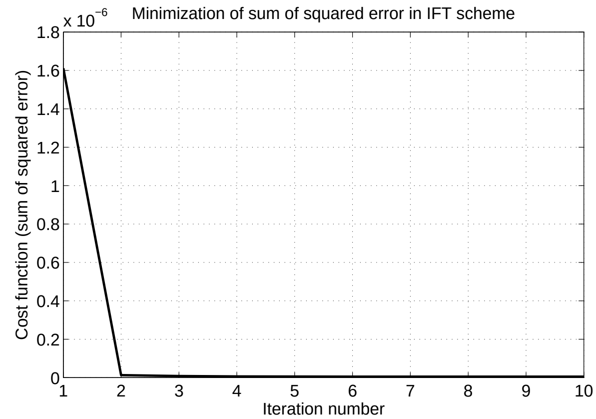

Fig. 7. Minimization of cost function (sum of squared error) in experiment over 10 cycles of tuning

$$P(s) = \frac{12}{5.3s^2 + 7.2s}.$$

Therefore we expect the feedforward parameters ρ_1 and ρ_2 to converge to 0.4417 and 0.6000, respectively.

The shape of the objective function $J(\rho)$ is shown in Figure 2. The shape of $J(\rho)$ depends on the particular reference trajectory used for tuning. The objective function achieves its minimum when ρ_1 and ρ_2 reach their true values corresponding to the true plant parameters, as expected. It can be seen that the value of the objective function depends more heavily on the ρ_1 parameter than on ρ_2 for our particular choice of trajectory.

The controller tuning algorithm was applied to the stage for ten iterations. The reference trajectory used for tuning is a scan and return motion and is shown in Figure 3. Figure 4 shows that the value of the objective function decreases with each iteration as desired. The sum of squared error decreased from 5.30×10^{-9} to 4.26×10^{-11}. The convergence of the feedforward parameters is shown in Figure 5. The parameters appear to converge to their expected values (values from the inverse of the plant model). The error profile after ten cycles of controller tuning is compared with the profile before any tuning in Figure 6. It is seen that the peak error during acceleration is reduced from 7.15×10^{-6} m to 1.13×10^{-6} m.

B. *Experiment*

Next we applied the iterative tuning algorithm experimentally to the wafer stage. The wafer stage was placed in a feedback loop with a PID controller to ensure stability while the feedforward controller was being tuned. Figure 7 is a plot of the cost function at each iteration. The figure shows that the sum of the square of the error decreases from 1.61×10^{-6} to $5.13 \times 10^{-9}\text{m}^2$ at the tenth iteration while the most significant reduction took place at the second iteration. Figure 8 shows a comparison of the error profile in the 1st and 10th iterations. It can be seen that the peak error during acceleration has decreased from 7.49×10^{-5} m to 5.98×10^{-6} m. The plot

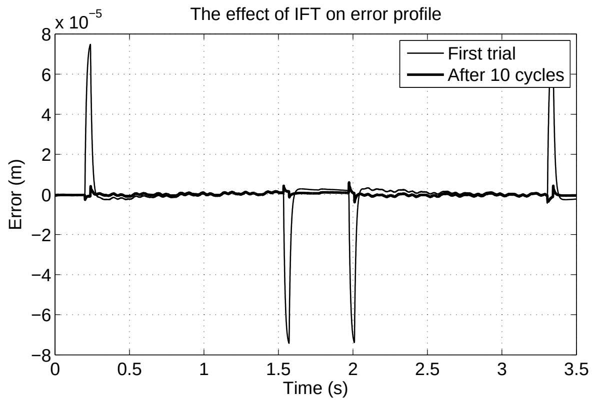

Fig. 8. Reduction of error after 10 cycles of tuning in experiment

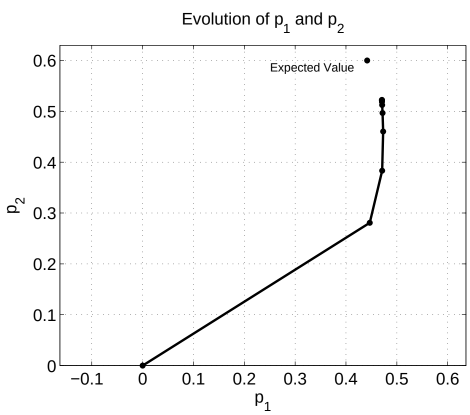

Fig. 9. Convergence of feedforward parameters in experiment with initial values $\rho_1 = 0$ $\rho_2 = 0$

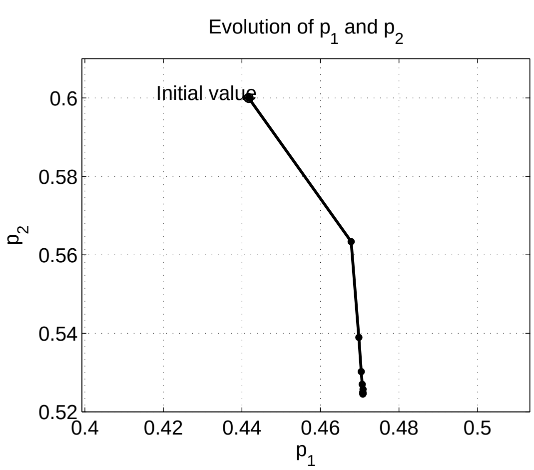

Fig. 10. Convergence of feedforward parameters in experiment with initial values $\rho_1 = 0.4417$ $\rho_2 = 0.6000$.

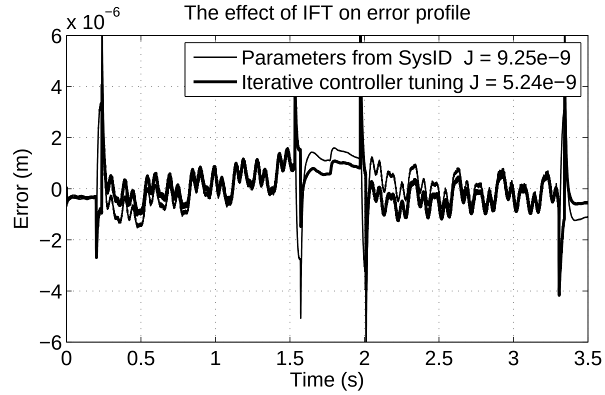

Fig. 11. Comparison of two feedforward controllers : 1. using parameters from iterative tuning vs. 2. using parameters corresponding to inverse of plant obtained through system identification

of the feedforward parameters over each iteration is shown in Figure 9. Both parameters converge.

Theoretically, the best performance should be achieved when the feedforward controller is the inverse of the plant. However, it is interesting to note that the parameters in Figure 9 converge to values different from the expected (i.e. values from the inverse of the plant model obtained through system identification). Even if the controller tuning is started with the parameters' initial values set to the expected values, after tuning, the parameters still converge to the same values as before (compare Figures 9 and 10). Using the new values of the parameters in the feedforward controller results in lower sum of squared error than using the expected values as seen in Figure 11. Therefore, applying iterative tuning is still beneficial even when a model of the plant already exists, because the plant model always has inaccuracies and the plant changes slightly over time.

IV. Force Ripple Compensation

One implementation issue arises due to disturbances caused by force ripple of the linear motor [10], [17]. The effect of the force ripple can be seen in Figure 8. Force ripple is a nonlinear periodic disturbance that arises in permanent magnet linear motors due to misalignment of coils and magnets. The force ripple is a nonlinear disturbance dependent on stage position and controller voltage output and is represented as a sum of sinusoids

$$F_{ripple}(y) = u(y)\left[\sum_{k=1}^{N} a_k \sin(k\omega_o y) + \sum_{k=1}^{N} a_k \cos(k\omega_o y)\right]$$

where y is stage position, $F_{ripple}(y)$ is the force caused by the force ripple disturbance, u is the controller output voltage, and ω_o is the basic frequency of the ripple which is known beforehand.

The presence of force ripple adversely affects the convergence of controller parameters. During experimental tuning, we reduced the effect of the force ripple on the gradient

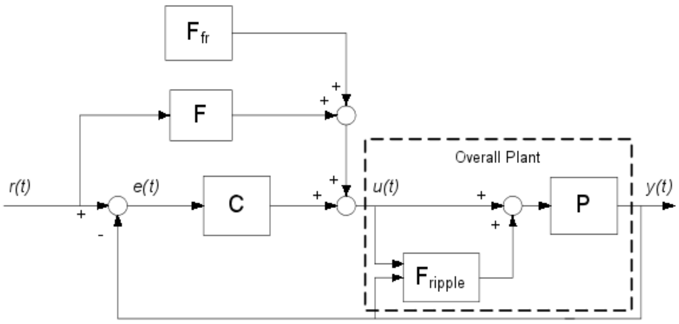

Fig. 12. Block Diagram with Force Ripple

calculation by approximating the error profile by a smooth polynomial during segments where force ripple is present.

However, instead of ignoring the effect of force ripple, now we try to cancel it out using additive feedforward control. We will show that the same iterative controller tuning method used earlier can be used to tune the parameters of the force ripple compensation feedforward controller. This method is still effective even when applied to eliminate nonlinear force ripple.

A. Methodology

We investigated the use of feedforward methods to compensate for force ripple. In this section we present the iterative tuning method which we apply to tune the force ripple compensator controller. The derivation is very similar to that presented in Section II.

We desire to cancel out the force ripple by additive feedforward control of the form

$$F_{fr}(t) = k_c v(t) \Bigg[\sum_{k=1}^{N} a_k \sin(k\omega_o r(t)) + \sum_{k=1}^{N} a_k \cos(k\omega_o r(t)) + \gamma_0 + \gamma_1 r(t) \Bigg]$$

where $r(t)$ is reference position, $v(t)$ is reference velocity, and k_c is a constant. The reason for approximating $y(t)$ and $u(t)$ in equation by $r(t)$ and $k_c v(t)$ is so that the feedforward data F_{fr} can be computed offline and remains constant from run to run.

The objective of the iterative tuning is to tune the coefficients a_k and b_k to minimize a cost function which is a quadratic function of tracking error, the same as in equation . The gradient of the cost function with respect to controller parameters is similarly identical to equation . However, now we must recalculate the gradient $\frac{\partial y}{\partial \rho}$.

From the block diagram in Figure 12, we see that with the addition of feedforward force ripple compensation, the output is given by

$$y(t) = \frac{PC}{1+PC} r(t) + \frac{PF}{1+PC} r(t) + F_{ripple} + \frac{P}{1+PC} k_c v(t) \left[\sum_{n=1}^{N} a_k \sin(k\omega_n n) + \sum_{n=1}^{N} b_k \cos(k\omega_o n) \right].$$

Taking the partial derivatives with respect to a_k and b_k, we have

$$\begin{bmatrix} \frac{\partial y}{\partial a_k} \\ \frac{\partial y}{\partial b_k} \end{bmatrix} = \begin{bmatrix} \frac{P}{1+PC} k_c v(t) \sin(k\omega_o n) \\ \frac{P}{1+PC} k_c v(t) \cos(k\omega_o n) \end{bmatrix}.$$

However, assuming that the force ripple disturbances are at low frequencies, we can approximate $\frac{PC}{1+PC}|_{j\omega_o} \approx 1$, so we will use simply

$$\begin{bmatrix} \frac{\partial y}{\partial a_k} \\ \frac{\partial y}{\partial b_k} \end{bmatrix} = \begin{bmatrix} \frac{1}{C} k_c v(t) \sin(k\omega_o n) \\ \frac{1}{C} k_c v(t) \cos(k\omega_o n) \end{bmatrix}$$

for the partial derivative of y. Using this approximation will eliminate the need to do a preliminary experiment to calculate $\frac{P}{1+PC} k_c v(t) \sin(k\omega_o n)$ for each harmonic frequency $k\omega_o$ of the force ripple.

B. Experiment

Next, the feedforward force ripple compensator parameters were tuned for the wafer stage. The results are shown in Figures 13, 14, and 15. The ∞-norm of the error during constant velocity scan phase decreased from $1.51 \times 10^{-}6$ m to $6.33 \times 10^{-}7$ m, and the sum of squared error decreased from $1.98 \times 10^{-}9$ to $1.52 \times 10^{-}10$.

The residual force ripple seen in Figure 14 is thought to be a result of the inaccuracy of the approximation of the disturbance signal from the nonlinear force ripple model with the additive feedforward control signal $F_{fr}(t)$. The DC-offset and slowly varying trend in the error profile is a result of the disturbance forces from the cables connected to the wafer stage, including power supply cables and sensor cables.

Figure 13 shows the convergence of parameters a_k, b_k. The oscillations in the estimates of parameters γ_0, γ_1, which correspond to the cable force disturbance, may be reduced by better modeling of cable forces, or by choosing a smaller step-size in the optimization algorithm.

V. Conclusion

Iterative controller tuning is an effective method for automated tuning of controller parameters for the purpose of minimizing a cost function. Iterative controller tuning is particularly applicable to fine-tuning controllers for plants with unknown or uncertain models. We applied the tuning algorithm to iteratively tune the parameters of a feedforward controller for a wafer stage including force ripple compensation. Tuning resulted in decrease in sum of squared error, convergence of controller parameters, and reduction of peak error during acceleration.

Although iterative controller tuning will still work without any plant model, including information already known about

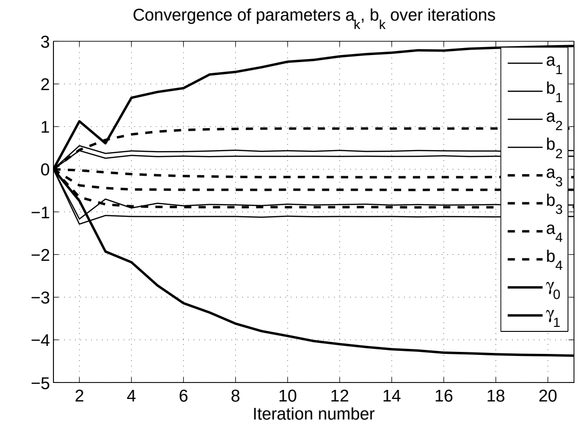

Fig. 13. Convergence of Feedforward Parameters in Experiment

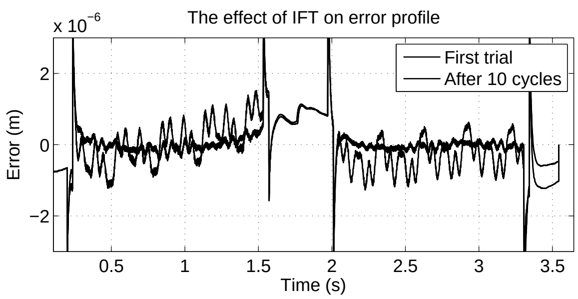

Fig. 14. Reduction of Error After 10 Cycles of Tuning in Experiment

the plant would be beneficial. Future research will include how to use prior information (for example, the plant transfer function)to improve iterative controller tuning. In addition, although iterative tuning of feedforward controllers works in the sense that it reduces sum of squared error, this research shows that the results are limited by the complexity of the feedforward controller, and could be improved by using a more complex structure for the controller. For example, the simple second-order feedforward controller cannot compensate for force ripple regardless of how well it is tuned. Therefore, future work should include investigation of more complex feedforward controllers to compensate for unmodeled plant dynamics and force ripple. Future work will also include designing and tuning a feedforward compensator for cable

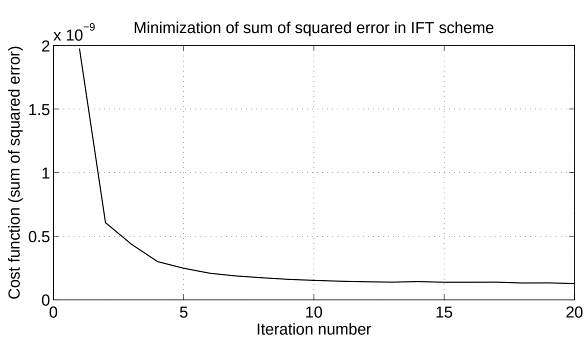

Fig. 15. Iterative Minimization of Cost Function in Experiment

force disturbances.

Also, it was seen that convergence of the parameters in the feedforward controller is dependent on choosing suitable values of the optimization algorithm parameters. Investigation should be made on how to use plant information to choose these optimization parameters to result in convergence of tuning.

ACKNOWLEDGMENT

The authors would like to thank the U.C. Discovery Grant and Nikon Research Corporation of America.

REFERENCES

[1] S. Mishra and M. Tomizuka, "Projection-based iterative learning control for wafer scanner systems."

[2] B. G. Dijkstra and O. H. Bosgra, "Extrapolation of optimal lifted system ILC solution, with application to a waferstage," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 4, 2002, pp. 2595–2600.

[3] B. Dijkstra and O. H. Bosgra, "Exploiting iterative learning control for input shaping, with application to a wafer stage," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 6, Jun. 2003, pp. 4811–4815.

[4] H. Hjalmarsson, S. Gunnarsson, and M. Gevers, "A convergent iterative restricted complexity control design scheme," in *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, vol. 2, Lake Buena Vista, FL, dec 1994, pp. 1735–1740.

[5] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative feedback tuning: theory and applications," *IEEE Control Systems Magazine*, vol. 18, no. 4, pp. 26–41, Aug. 1998.

[6] H. Hjalmarsson, "Iterative feedback tuning - an overview," *Int.J.Adapt.Control Signal Process*, vol. 16, pp. 373–395, 2002.

[7] I. Nilkhamhang and A. Sano, "Iterative tuning of feedforward and feedback controllers with physical parameter identification for two-mass system," in *IEEJ Trans. EIS*, vol. 126, no. 4, 2006.

[8] K. Hamamoto, T. Fukuda, and T. Sugie, "Iterative feedback tuning of controllers for a two-mass spring system with friction," in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 3, Sydney, NSW, Dec. 2000, pp. 2438–2443.

[9] S. van der Meulen, R. Tousain, and O. Bosgra, "Fixed structure feedforward controller tuning exploiting iterative trials, applied to a high-precision electromechanical servo system," in *American Control Conference, 2007. ACC '07*, New York, NY, USA, Jul. 2007, pp. 4033–4039.

[10] C. Rohrig and A. Jochheim, "Identification and compensation of force ripple in linear permanent magnet motors," in *American Control Conference, 2007.*, vol. 3, Arlington, VA, USA, 2001, pp. 2161–2166.

[11] E. Schrijver and J. van Dijk, "H $_\infty$ design of disturbance compensators for cogging forces in a linear permanent magnet motor," *Journal A*, vol. 140, pp. 36–41, 1999.

[12] K. Tan and S. Zhao, "Adaptive force ripple suppression in iron-core permanent magnet linear motors," in *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, 2002.

[13] L. Xu and B. Yao, "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*, 2000.

[14] C. Rohrig and A. Jochheim, "Motion control of linear permanent magnet motors with force ripple compensation," in *American Control Conference, 2001. Proceedings of the 2001*, vol. 3, 2001, pp. 2161–2166.

[15] T. Lee, K. Tan, S. Lim, and H. Dou, "Iterative learning control of permanent magnet linear motor with relay automatic tuning," *Mechatronics*, vol. 10, no. 1, pp. 169–190, 2000.

[16] G. Otten, T. J. de Vries, and J. v. Amerongen, "Linear motor motion control using a learning feedforward controller," *IEEE/ASME Transactions on Mechatronics*, vol. 2, no. 3, September 1997.

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