2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Experimental Comparison of Stochastic Iterative Learning Control Algorithms

Zhonglun Cai, Chris T. Freeman, Paul L. Lewin and Eric Rogers

***Abstract*— A number of iterative learning control algorithms have been developed in a stochastic setting in recent years. The results currently available are in the form of algorithm derivation and the establishment of various fundamental systems theoretic properties. As the crucial, in terms of eventual use in applications, next stage this paper compares their performance when implemented on a gantry robot system.**

I. Introduction

Iterative Learning Control (ILC) is concerned with the performance of systems that operate in a repetitive manner, where the task is to follow some specified trajectory defined over a given finite time interval, also known as a pass or trial, with high precision. The novel principle behind ILC is to use information from previous trials, often in combination with appropriate current trial information, to select the present control input in order to improve the performance from trial-to-trial. In particular, the aim is to sequentially reduce the tracking error (the difference between the output of a trial and the specified reference trajectory) until it is ideally zero.

The original work in this area is credited to [1] and since then there have been substantial developments in both system theoretic and application terms. For an overview of the algorithm development side see, for example, [2], [3] (the second of these references has the additional feature of a categorization of what is a very diverse area). Application areas include robotics, automated manufacturing plants and food processing. For more details, including some areas where there is clear potential for significant added benefit from fully developed ILC, one possible source is the survey article [2].

II. Background

The algorithms considered in this work have arisen from research reported in [4], [5], [6] for discrete linear time varying systems whose dynamics can be described by

$$\begin{aligned} x_k(t+1) &= A(t)x_k(t)+B(t)u_k(t)+\omega_k(t) \\ y_k(t) &= C(t)x_k(t)+\nu_k(t) \end{aligned} \tag{1}$$

where $t \in [0,T]$, where $T < \infty$ denotes the trial length, the system state $x_k(t) \in \Re^n$, the system input $u_k(t) \in \Re^p$, the state disturbance $\omega_k(t) \in \Re^n$, the system measured output $y_k(t) \in \Re^q$ and the system measurement error $\nu_k(t) \in \Re^q$. In this work, however, the interest lies in time invariant dynamics. Attention will focus on the following algorithms from [7], [4], [5]:

D-type ILC — this uses the approximate error differentiation instead of the error derivative considered in [1] for continuous-time systems

$$u_{k+1}(t) = u_k(t)+K_k[e_k(t+1)-e_k(t)] \tag{2}$$

P-type ILC — here

$$u_{k+1}(t) = u_k(t)+K_k e_k(t+1) \tag{3}$$

where K_k is the $(p \times q)$ learning control gain matrix, and $e_k(t)$ is the output error, i.e. $e_k(t) = r(t)-y_k(t)$ where $r(t)$ is the desired reference output trajectory. In each case the learning control gain matrix K_k is given by

$$\begin{aligned} K_k &= \mathbf{P}_{u,k}(CB)^T \cdot [(CB)\mathbf{P}_{u,k}(CB)^T \\ &\quad +(C-CA)\mathbf{P}_{x,k}(C-CA)^T + C\mathbf{Q}_t C^T + \mathbf{R}_t]^{-1} \\ \mathbf{P}_{u,k+1} &= (I-K_kCB)\mathbf{P}_{u,k} \\ \mathbf{P}_{x,k+1} &= A\mathbf{P}_{x,k}A^T + B\mathbf{P}_{u,k}B^T + \mathbf{Q}_t \end{aligned}$$

where $\mathbf{Q}_t = E(\omega(t,k)\omega(t,k)^T)$, $\mathbf{P}_{x,0} = E_k(\delta x(0,k)\delta x(0,k)^T)$ are positive semi-definite matrices, $\mathbf{R}_t = E_k(\nu(0,k)\nu(0,k)^T)$ is a positive definite matrix and $\mathbf{P}_{u,0} = E(\delta u(t,0)\delta u(t,0)^T)$ is a symmetric positive definite matrix (E and E_k are the expectation operators with respect to the time domain and the trial domain respectively, $\delta x(0,k), \delta u(t,0)$ are the initial state error and the input error respectively).

III. Gantry Robot Test Facility

The algorithms above have been implemented on an industrial gantry robot (see Figure 1) which performs a "pick and place" task and is similar to systems which can be found in industry. Such industrial processes include food canning, bottle filling or automotive assembly, all of which require accurate tracking control with a minimum level of error in order to maximize production rates and minimize loss of product due to faulty manufacture. The gantry robot is constructed from two types of linear motion device. The X-axis comprises the lowest horizontal section, and consists of one brushless linear dc motor and a parallel free running slide. The Y-axis lies directly above this, is perpendicular to the X-axis, and has one end attached to the linear motor and the other end to the slide. The Y-axis comprises a single brushless linear dc motor. The X and Y-axes are 1.02m and 0.91m long respectively. Finally, the vertical Z-axis comprises a short 0.10m travel linear ball-screw stage driven by a rotary brushless dc motor. All axes are powered by matched brushless motor dc amplifiers and axis motion

Zhonglun Cai, Chris T. Freeman, Paul L. Lewin and Eric Rogers are with School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK zc205r@ecs.soton.ac.uk

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

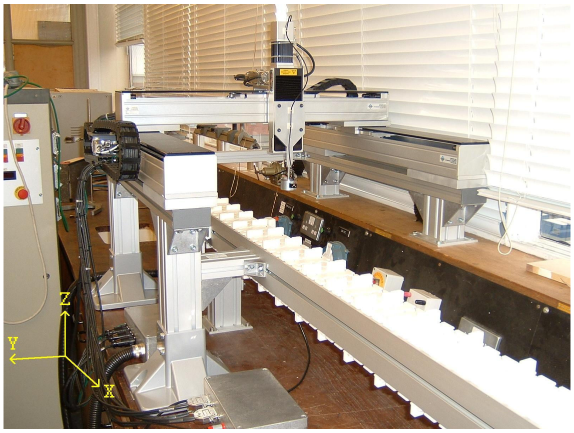


Fig. 1. The multi-axes gantry robot.

is detected and recorded with appropriate optical encoder systems. Each axis has been modeled using the velocity control mode of operation, in which the amplifier receives the encoder data as well as the computer, therefore providing an inner closed loop and an integrating action to the system. The axes dynamics have been determined by performing a series of open loop frequency response tests. From the resulting measurements, linear approximations of the transfer function for each axis were determined and then refined using a non-linear optimization technique. The frequency response obtained for the X-axis is shown in Figure 2 (the responses of the other axes are similar). The X-axis dynamics are approximated by a state-space model of order seven with state, input and output matrices given by

$$A_x = \begin{bmatrix} 2.41 & -0.86 & 0.85 & -0.59 & 0.30 & -0.19 & 0.32 \\ 4.00 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.00 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.50 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.25 & 0 \end{bmatrix}$$

$$B_x = \begin{bmatrix} 0.0313 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C_x = \begin{bmatrix} 0.0095 & -0.0023 & 0.0048 & -0.0027 & 0.0029 \\ -0.0011 & 0.0029 \end{bmatrix}$$

The gantry robot is designed to perform a "pick and place" action with a trial length of 2 seconds, which is equivalent to 30 units per minute (UPM). Figure 3 shows the reference trajectory. The stoppage time between each trial is used to compute the control vector for the subsequent trial. The gantry axes are homed to a predefined point before each iteration begins with an accuracy of ±30 microns in order to minimize the effects of initial state error. A sampling time of $T_s = 0.01$s has been used in all tests.

IV. Control Schemes

Figure 4 shows a block diagram of the control configuration considered in this work. This is a parallel arrangement of a PID feedback controller and an ILC controller, and has been shown to offer advantages over other arrangements, especially in the presence of sudden changes in plant dynamics [8]. When the PID controller is shut down, or switched out, (the gains K_p, K_i, K_d set to zero), the control input update reduces to that of (2).

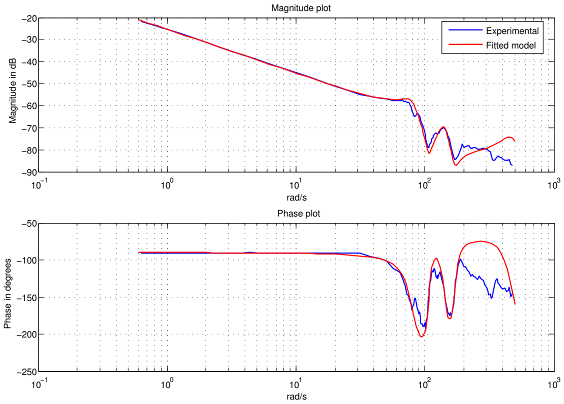


Fig. 2. X-axis Bode gain plot.

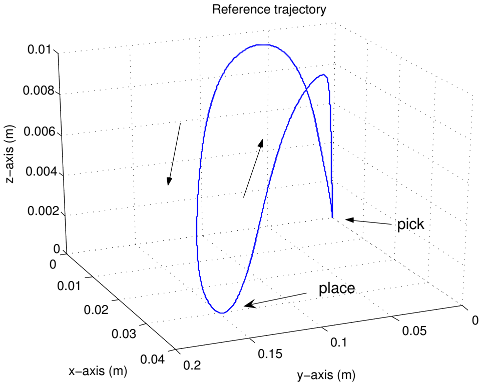


Fig. 3. 3D Reference trajectories

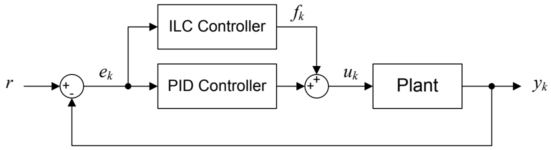


Fig. 4. Block diagram of the controller structure and arrangement.

When the PID controller is turned on, a feedback/feedforward scheme is implemented. Let f_k be the feedforward signal from the ILC controller, for the D-type algorithm (2) this is given by

$$f_{k+1}(t) = f_k(t) + K_k[e_k(t+1) - e_k(t)] \tag{4}$$

and for the P-type algorithm, by

$$f_{k+1}(t) = f_k(t) + K_k e_k(t+1) \tag{5}$$

Hence for the D-type algorithm

$$u_k(t) = f_k(t) + [K_p e_k(t) + K_i \int_0^t e_k(\tau)\mathrm{d}\tau + K_d \frac{\mathrm{d}e_k(t)}{\mathrm{d}t}] \tag{6}$$

Defining $\text{PID}[e_k(t)] \triangleq K_p e_k(t) + K_i \int e_k(t)\mathrm{d}t + K_d \frac{\mathrm{d}e_k(t)}{\mathrm{d}t}$ allows this to be written as

$$\begin{aligned} u_k(t) &= f_k(t) + \text{PID}[e_k(t)] \\ u_{k+1}(t) &= f_{k+1}(t) + \text{PID}[e_{k+1}(t)] \\ &= u_k(t) + \text{PID}[e_{k+1}(t) - e_k(t)] \\ &\quad + K_k[e_k(t+1) - e_k(t)] \end{aligned}$$

and similarly for the P-type algorithm

$$u_{k+1}(t) = u_k(t) + \text{PID}[e_{k+1}(t) - e_k(t)] + K_k e_k(t+1) \qquad (7)$$

With a PID feedback controller, the input update is consequently the sum of the ILC update plus the response of the PID controller to the difference between the current and previous trial outputs. Note that, after sampling with period T_s, (6) becomes

$$\begin{aligned} u_k(t) &= f_k(t) + [K_p e_k(t) + K_i \sum_{\tau=1}^{t} \frac{(e_k(\tau) + e_k(\tau-1))T_s}{2} \\ &\quad + K_d \frac{e_k(t) - e_k(t-1)}{T_s}] \quad t \in [0, T] \end{aligned} \qquad (8)$$

V. Simulations and Experimental Results

A. Basic Design and Simulation Studies

Prior to experimental implementation, a series of designs have been completed and their performance evaluated in simulation. Figure 5 shows results using the D-type algorithm with a variety of PID controller parameters, where PID$=\{0,0,0\}$ corresponds to neglecting this control loop from the scheme. X-axis mean square error (mse) results are shown, and it can be seen that higher PID gains are able to give improved performance only on the first trial, but do not assist the ILC controller in subsequent error reduction.

Figures 6 and 7 show the X-axis mse results obtained by varying $P_{u,0}$ (the initial value of $\mathbf{P}_{u,k}$) and $\mathbf{Q}_t$ where

$$\mathbf{Q}_t = Q_t I \qquad (9)$$

$$\mathbf{P}_{x,k} = P_{x,k} I \qquad (10)$$

here Q_t and $P_{x,k}$ are scalars and I is the identity matrix of appropriate dimensions. $\mathbf{R}_t$ is set to the mean value of the white noise, this being extracted from the error vector after each completed trial. These results indicate that selecting a larger $P_{u,0}$ and a smaller Q_t gives better tracking performance both in terms of convergence speed and final error.

Since ILC design may be anti-causal along-the-pass, zero-phase filtering (see, for example, [9]) is feasible and here a low-pass filter of this form has been used to mitigate the effects of high frequency noise (note that it has been reported [10] that excessive phase shift can cause the ILC controller to incorrectly compensate for the error and lead to an unstable system). Here, a third order, low-pass, zero-phase Chebychev filter with a 20 dB attenuation at 15 Hz (94.25 rad/sec) has been used with discrete-time transfer function

$$H(z) = \frac{0.102693 + 0.002934z^{-1} + 0.002934z^{-2} + 0.102693z^{-3}}{1 - 1.644597z^{-1} + 1.091881z^{-2} - 0.236029z^{-3}} \qquad (11)$$

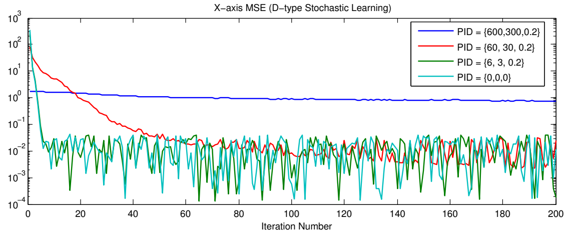


Fig. 5. Simulation results with various PID gains. ($P_{u,0} = 1000, Q_t = 0.001, P_{x,0} = 0.1$, Disturbance: 0.0002)

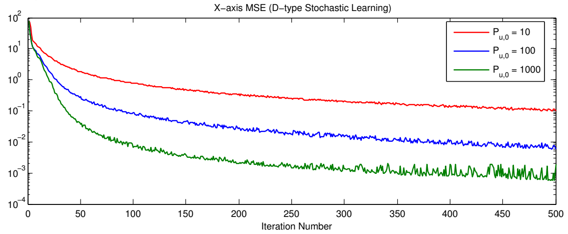


Fig. 6. Simulation results with varying $P_{u,0}$. (PID $= \{60, 30, 0.2\}, Q_t = 0.001, P_{x,0} = 0.1$, Disturbance: 0.00005)

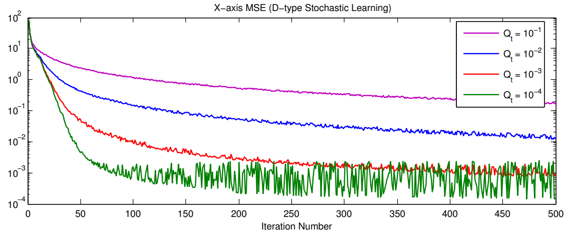
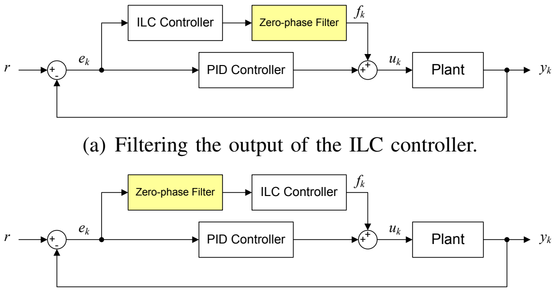


Fig. 7. Simulation results with varying Q_t. (PID $= \{60, 30, 0.2\}, P_{u,0} = 800, P_{x,0} = 0.1$, Disturbance: 0.00005)

Moreover, since the filter placement in the overall scheme is also critical for convergence and tracking performance, two arrangements have been considered: In Figure 8(a) the filter is applied to the feedforward signal and in Figure 8(b) it is applied to the error signal prior to computation of the control signal for the next trial. Experiments have shown that the latter arrangement provides the superior performance, and so it will be adopted in the tests which follow.



(a) Filtering the output of the ILC controller.

(b) Filtering error vector before the ILC update

Fig. 8. Filter arrangements.

B. Experimental Results — D-type

The ILC controller has been applied to all three axes of the gantry robot, and the zero-phase filter given by (11) has been used in all experiments. The reference trajectories (see Figure 3) are the same as those used in all previously reported results in which ILC algorithms have been implemented on the gantry robot (to enable the broadest possible comparison to be made).

Figure 9 shows the resulting errors for all axes without a PID feedback controller, for various values of $P_{u,0}$. In contrast to the simulation results, use of larger values of $P_{u,0}$ does not lead to appreciable differences in the levels of error produced.

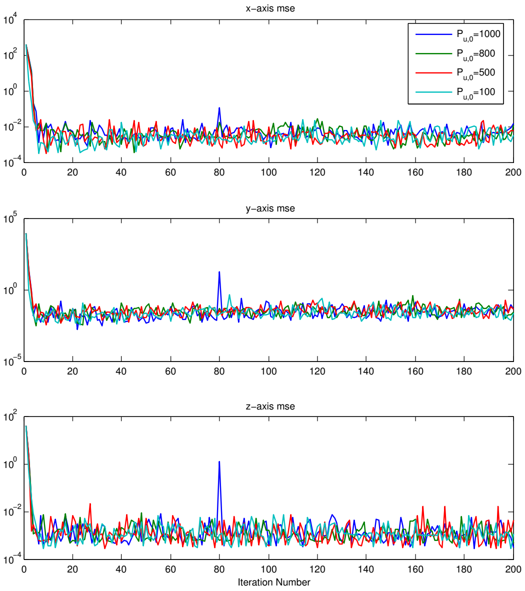


Fig. 9. Experimental results (PID=$\{0,0,0\}, Q_t = 0.001, P_{x,0} = 0.1$).

Figure 10 shows the errors for all axes without a PID feedback controller, using various values of Q_t. With smaller values of Q_t, the performance is improved, especially for the Y and Z-axes. However, reducing Q_t further provides progressively less advantage in terms of performance. From the mse curves, it can be seen that the value of Q_t significantly influences the learning speed.

It has been found that use of the PID feedback controller provides a higher level of tracking performance over initial trials. This is illustrated by Figure 11 in which error plots are given for all axes which show that, with small PID gains, the ILC controller is able to cooperate more effectively with the PID controller. Without the PID controller, the convergence rates for all axes are higher, but the performance in terms of the final level of error is diminished, especially for the Y-axis. One possible reason for this is that the reference signal for this axis is much higher than for the other two.

With the PID feedback controller in place, another series of experiments have been conducted in order to compare the effects of varying $P_{u,0}$. The results are given in Figure 12, and it can be seen that the performance for different initial values of $P_{u,0}$ is generally quite similar. Exceptions occur in the case of the Y-axis, where the initial values of $P_{u,0} = 500$ or $P_{u,0} = 800$ clearly improve over those using $P_{u,0} = 100$. These experimental results as a whole suggest that the algorithm has significant robustness and disturbance rejection potential. Note also that very large unexpected errors can arise on some trials (see for example, Figure 12, around trial 100 with $P_{u,0} = 100$, and around trial 120 with $P_{u,0} = 200$) but overall they do not lead to long-lasting negative effects.

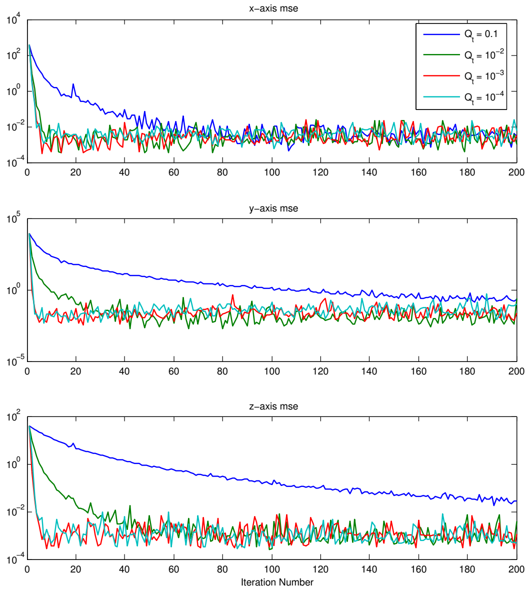


Fig. 10. Experimental results (PID=$\{0,0,0\}, P_{u,0} = 100, P_{x,0} = 0.1$).

C. Experimental Results — P-type

Several experiments have been completed using the P-type stochastic learning algorithm. These have firstly been conducted using the same controller arrangement, the same filtering method and smaller values of initial parameters as used in the case of D-type ILC, however this did not lead to a high level of performance. This is illustrated in Figure

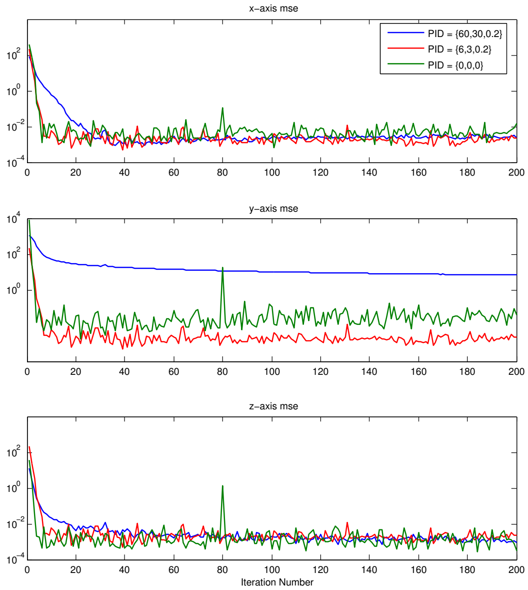


Fig. 11. Experimental results ($P_{u,0} = 1000, Q_t = 0.001, P_{x,0} = 0.1$).

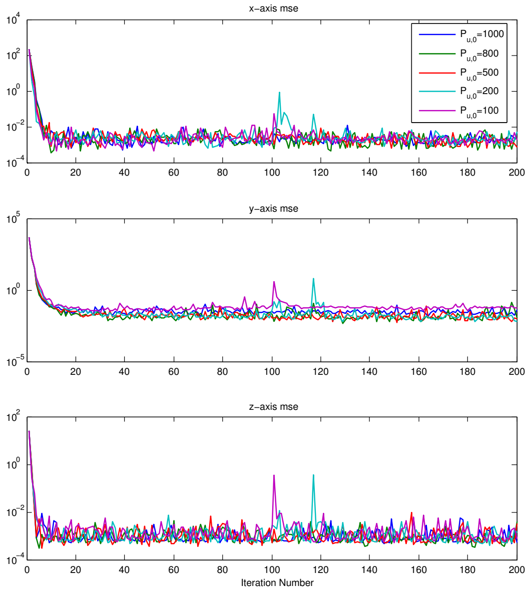


Fig. 12. Experimental results (PID=$\{6,3,0.2\}, Q_t = 0.001, P_{x,0} = 0.1$).

13, where the upper figure shows the tracking error over all iterations. Only 10 trials were completed and the experiment was then stopped due to the presence of instability over a narrow frequency band. A frequency analysis was applied to the error signal (see Figure 13 lower figure), and a frequency of approximately 11-12Hz can be seen to build up. Another 4^{th} order Chebychev filter (see (12)) which has a smaller cut off frequency of around 5Hz has therefore been designed to solve this problem, and is given by

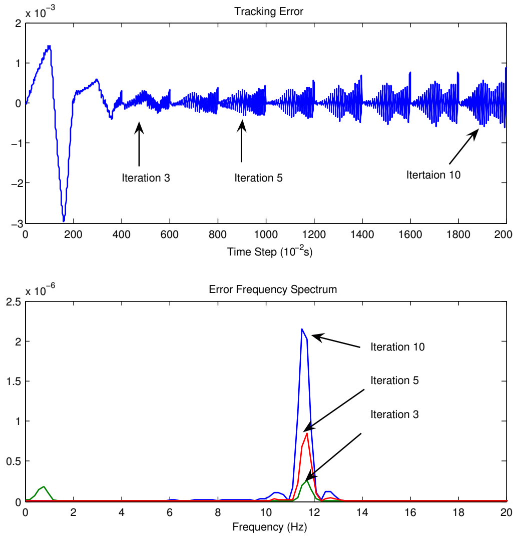


Fig. 13. The frequency spectrum of tracking error for some trials.

$$H(z) = \frac{0.0002+0.0007z^{-1}+0.0011z^{-2}}{1-3.5328z^{-1}+4.7819z^{-2}}\cdots$$
$$\cdots\frac{+0.0007z^{-3}+0.0002z^{-4}}{-2.9328z^{-3}+0.6868z^{-4}} \qquad (12)$$

Figure 14 shows a series of experimental results for all axes obtained with the modified filter in place. It is clear that larger initial values of $P_{u,k}$, $P_{u,0}$ provide superior performance, especially for the Y-axis and Z-axis.

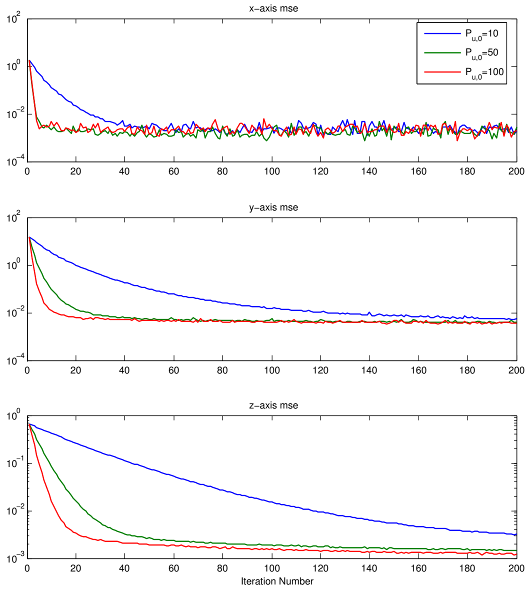


Fig. 14. Experimental results (PID=$\{600,300,0.2\}, Q_t = 0.1, P_{x,0} = 0.1$).

Figure 15 shows a series of experimental results using various values of Q_t. Although simulation studies indicated that smaller values of Q_t lead to improved performance, experimental results for the X-axis show little difference in practice. Furthermore, the smaller values were not able to provide superior results initially. For the Y-axis and Z-axis, smaller values of Q_t do lead to a reduced level of final error. However, as with the X-axis results, too small a value of Q_t produces poorer performance in initial trials.

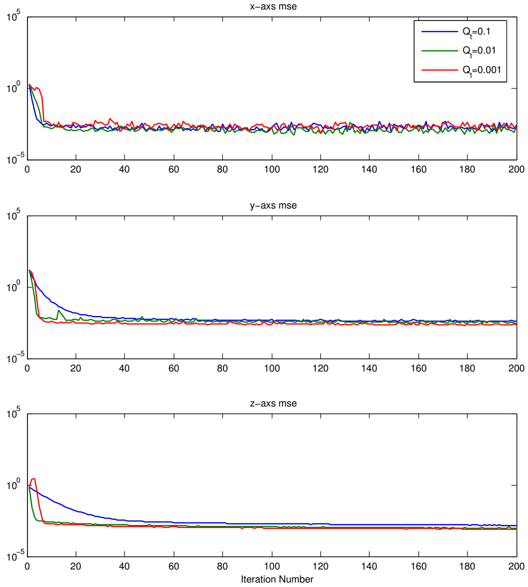


Fig. 15. Experimental results (PID=$\{600,300,0.2\}, P_{u,0} = 50, P_{x,0} = 0.1$).

VI. Comparison of Experimental Results

A. Comparison of D-type and P-type

The best performing results achieved for both the D-type and P-type stochastic learning algorithms are compared in Figure 16. This shows that the P-type algorithm produces superior performance for the Y-axis compared with the use of the D-type update. The P-type algorithm slightly improves on results of the D-type when applied to the X-axis. However, the performance is approximately equal in the case of the Z-axis, but the P-type algorithm holds the final error with significantly less fluctuation.

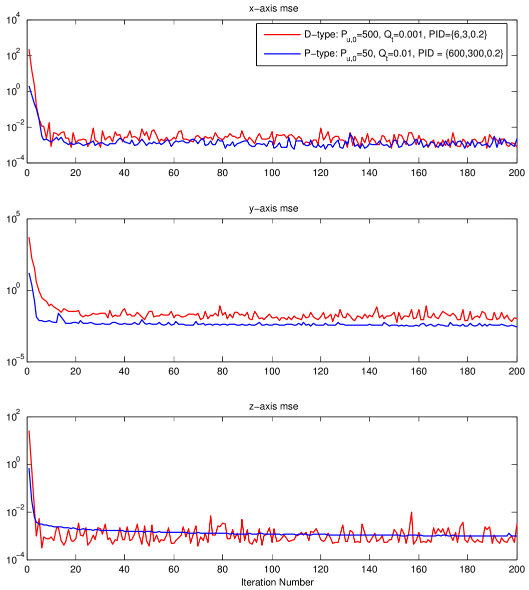


Fig. 16. A comparison between the D-type and P-type stochastic learning algorithms.

B. Comparison with Other Algorithms

Compared with other algorithms that have been implemented on the gantry robot, which include the basic P-type algorithm with an aliasing filter, the inverse algorithm, and norm-optimal ILC (see [11], [8], [12] for details), a similar performance is achieved using the discrete stochastic learning algorithms. Figure 17 shows that the convergence speed for the stochastic learning algorithms is much more rapid than the P-type with an signal aliasing filter, and slightly slower than when using norm-optimal ILC and the inverse algorithm. In terms of tracking error, the stochastic learning algorithms outperform the inverse algorithm, but are unable to reach the level attained by norm-optimal ILC. P-type with signal aliasing outperforms the remaining methods in terms of reducing the fluctuation of the final error.

VII. Conclusion

Stochastic learning algorithms have been implemented on a multi-axis gantry robot and their performance assessed. The ILC controller has been combined with a PID feedback controller in a parallel arrangement. Experimental results have shown that the highest level of performance is achieved when the PID controller is tuned using small parameter values. Moreover, it is found that use of a robustness filter applied to the error signal prior to the ILC update greatly improves upon the performance of previous implementations. In comparing the two approaches it has been found that the performance of the P-type stochastic learning algorithm was slightly superior to that of the D-type stochastic learning algorithm. Furthermore, when compared with other ILC algorithms implemented on the same system, it has been found that the performance of the stochastic learning algorithm compares favourably, it only being eclipsed by the far more computationally intensive norm-optimal ILC algorithm.

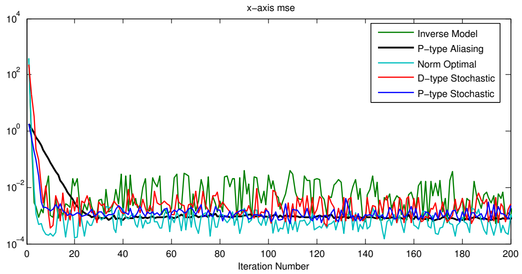


Fig. 17. Comparison of MSE for x-axis with other algorithms.

References

[1] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operations of robots by learning,” *Journal of Robotic Systems*, vol. 1, pp. 123–140, 1984.

[2] D. A. Bristow, M. Tharayil, and A. G. Alleyne, “A survey of iterative learning control a learning-based method for high-performance tracking control,” *IEEE control systems magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[3] H.-S. Ahn, Y. Chen, and K. L. Moore, “Iterative learning control: brief survey and categorization 1998–2004,” *IEEE Transactions on Systems Man and Cybernetics Part C*, 2007, in Press.

[4] S. S. Saab, “A discrete-time stochastic learning control algorithm,” *IEEE Transactions On Automatic Control*, vol. 46, no. 6, pp. 877–887, June 2001.

[5] ——, “On a discrete-time stochastic learning control algorithm,” *IEEE Transactions On Automatic Control*, vol. 46, no. 8, pp. 1333–1335, August 2001.

[6] ——, “Stochastic p-type/d-type iterative learning control algorithms,” *International Journal of Control*, vol. 76, no. 2, pp. 139–148, 2003.

[7] ——, “A discrete-time learning control algorithm for a class of linear time-invariant systems,” *IEEE Transactions On Automatic Control*, vol. 40, no. 6, pp. 1138–1142, June 1995.

[8] J. D. Ratcliffe, J. J. Hätönen, P. L. Lewin, E. Rogers, J. Harte, and D. H. Owens, “P-type iterative learning control for systems that contain resonance,” *International Journal of Adaptive Control and Signal Processing*, vol. 19, no. 10, pp. 769–796, 2005.

[9] A. M. Plotnik and R. W. Longman, “Subtitles in the use of zero-phase low-pass filtering and cliff filtering in learning control,” *Advances in the Astronautical Sciences*, vol. 103, pp. 673–692, 1999.

[10] K. Chen and R. Longman, “Stability issues using FIR filtering in repetitive control,” *Advances in the Astronautical Sciences*, vol. 112, no. 2, pp. 1321–1340, 2002.

[11] J. D. Ratcliffe, T. J. Harte, J. J. Hätönen, P. L. Lewin, E. Rogers, and D. H. Owens, “Practical implementation of a model inverse optimal iterative learning controller on a gantry robot,” in *Proceedings of IFAC Workshop on Adaptive and Learning in Control and Signal Processing and the IFAC Workshop on Periodic Control Systems*, Yokohama, Japan, 2004, pp. 687–692.

[12] J. D. Ratcliffe, P. L. Lewin, E. Rogers, J. J. Hätönen, and D. H. Owens, “Norm-optimal iterative learning control applied to gantry robots for automation applications,” *IEEE Transactions On Robotics*, vol. 22, no. 6, pp. 1303–1307, December 2006.