

# Iterative Learning Control Applied to a Gantry Robot and Conveyor System

C.T.Freeman\*, P.L.Lewin, E.Rogers and J.D.Ratcliffe

\*BEng, BSc, PhD. Lecturer in applied control. School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK. (email: [cf@ecs.soton.ac.uk](mailto:cf@ecs.soton.ac.uk), phone: +44 (0)23 8059 3486)

## Executive Summary

*Synchronisation is routinely required to coordinate the actions of the various sub-systems involved in process applications. This is commonly achieved through direct mechanical coupling, involving gears, drive belts and cams. Apart from the additional cost incurred, these components are subject to wear, constrain the layout of the plant, and may have limited accuracy. It is shown in this paper that a mechanical linkage between two sub-systems may be replaced by instead implementing a control scheme comprising an iterative learning controller together with a supervisory control loop. To illustrate the approach, two types of iterative learning controller are first implemented on a gantry robot test facility to confirm the high levels of tracking accuracy that may be achieved. The supervisory control loop is then added to synchronise the 'pick and place' action of the robot with a conveyor system moving at constant velocity. Experimental results are provided to confirm both the accurate tracking performance produced by the iterative learning controller, and the high level of synchronisation achieved by the overall scheme.*

## 1. Introduction

A great many industrial automation applications, such as food processing, injection molding, and vehicle assembly lines, consist of several separate machines, or sub-systems, operating simultaneously to achieve an overall goal. An example of this type of plant consists of a robot placing objects on a conveyor, which is travelling at constant velocity. Such applications occur frequently in the food production industry, for instance, when filling containers with fluid. The overall performance of a packaging machine, which has several synchronised axes, will be determined by the relative error between different axes rather than the absolute error in any one axis. Mechanical synchronisation, through the use of belts or cams is possible, but here the feasibility of electronic synchronisation through the additional control of the relative error between synchronised axes is investigated. The approach taken is based around Iterative Learning Control (ILC) which is concerned with trajectory tracking tasks which are of a repeating or cyclic nature. Motivated by human learning, ILC uses data obtained during previous trials/iterations to generate a new plant input which aims to reduce the tracking error at the next iteration<sup>1</sup>. A large number of ILC

algorithms exist which are theoretically capable of reducing the tracking error to zero as the number of iterations tends to infinity, thereby offering the potential for significant performance advantage over the use of conventional controllers<sup>2</sup>.

## 2. Experimental Test Facility

The test facility consists of a three axis gantry robot mounted over a plastic chain conveyor, and is shown in Figure 1.

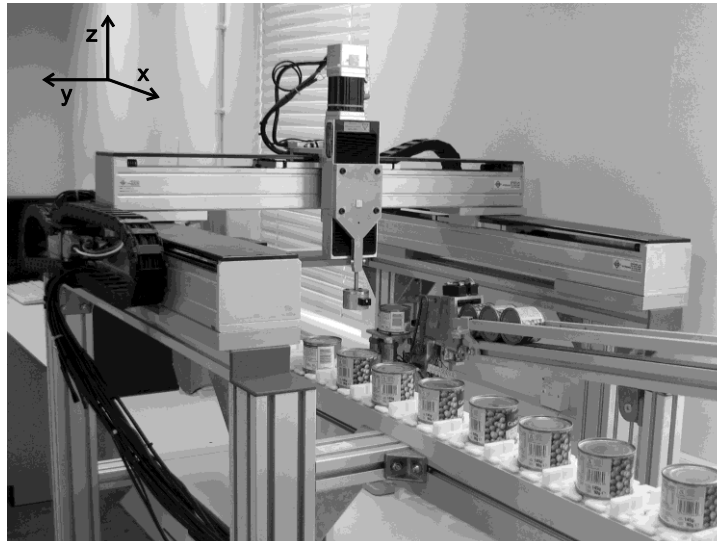


Fig. 1. Gantry robot, conveyor and payload dispenser

The robot has an electromagnet end-effector and is designed to collect payloads from an asynchronous dispenser and transfer them to the conveyor which is moving at constant velocity. The  $x$ -axis comprises the lowest horizontal section of the gantry robot, 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 is detected and recorded with appropriate optical encoder systems. The conveyor beneath the robot is constructed from extruded aluminium beams and provides a useful conveying length of 6 meters. Molded plastic segments are linked together to form a conveying chain, which is driven along the upper and lower surfaces of the extruded beam by a three phase induction motor and reduction gearbox. The induction motor is powered by a pulse width modulated inverter

drive. To separate payloads on the conveyor a ‘flight’ protrudes from every third link. The conveyor position is accurately measured by using an incremental encoder attached to the induction motor drive shaft, which is reset every time a flight passes a micro-switch mounted on the extruded beam. The system is designed so that the conveyor travels with a constant velocity of  $0.06\text{ms}^{-1}$ , whilst the gantry robot performs a ‘pick and place’ action with a trial length,  $T$ , of 2 seconds. The reference trajectory for the gantry robot used to achieve this is shown in Figure 2.

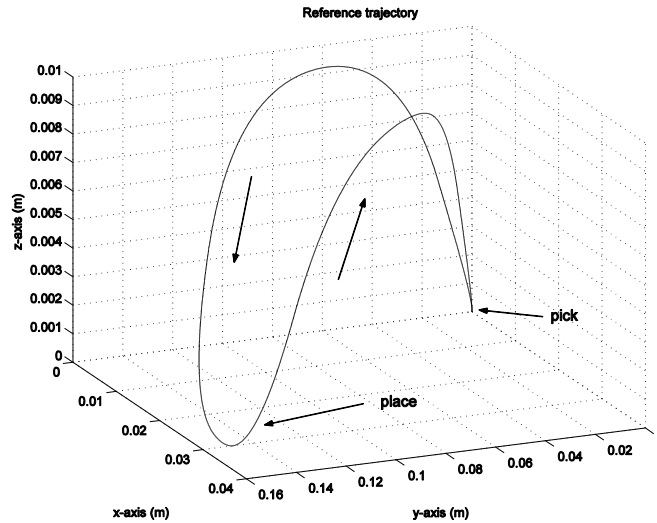


Fig. 2. 3D representation of reference trajectories

The  $x$ ,  $y$  and  $z$ -axis components of this trajectory will be denoted by  $r_x(t)$ ,  $r_y(t)$  and  $r_z(t)$  respectively. To achieve synchronisation, ILC could be implemented on each axis of the gantry and also the conveyor. However, since it is not feasible in practice to achieve zero tracking error, a relative tracking error will exist between these sub-systems and synchronisation will be lost. Also, since the conveyor is only required to travel at constant velocity, the use of ILC on this sub-system is a waste of resources since the most basic proportional feedback controller can achieve this task (at the cost of a large positioning error). Therefore the approach that will be taken is to only implement ILC on the gantry robot, but to develop a supervisory controller which can adjust the reference trajectory used, in order to reduce the relative error between the two sub-systems.

### 3. ILC Methods

Before developing the synchronisation control loop, an ILC scheme is required to enable the gantry to accurately perform the ‘pick and place’ movement. Two ILC methods have been considered for implementation on each of the

gantry's axes; the first has a simple structure and does not need an explicit plant model, the second is a far more complex model-based approach.

### 3.1 P-type ILC

P-type ILC is one of the simplest forms of learning controller and is given by

$$u_{k+1}(t) = u_k(t) + \beta e_k(t) \quad (1)$$

where  $\beta$  (beta) is the scalar learning gain,  $u_k(t)$  is the input on the  $k^{th}$  trial, and the output, reference and error are given by  $y_k(t)$ ,  $r(t)$  and  $e_k(t) = r(t) - y_k(t)$  respectively. P-type ILC alone is poorly suited to integrating plants and is particularly sensitive to non-repeating disturbances, therefore it is often coupled with a feedback controller<sup>3</sup>. For simplicity, a proportional feedback controller has been chosen and a parallel configuration used, in which the controllers operate independently of each other. Both are supplied with the reference trajectory and their output is summed to produce the plant input. The feedback controller provides reasonable tracking performance in the first few iterations and adds robustness to the system, while the learning controller improves the performance at each successive iteration. This hybrid arrangement of two controllers has been found unstable with respect to high frequency noise and low frequency resonances. Therefore an aliasing module has been added to the learning controller within the learning loop. This has the effect of a low pass, zero-phase filter, which allows the convergence conditions to be met<sup>4</sup>. The overall configuration of the learning and feedback controller is shown in Figure 3.

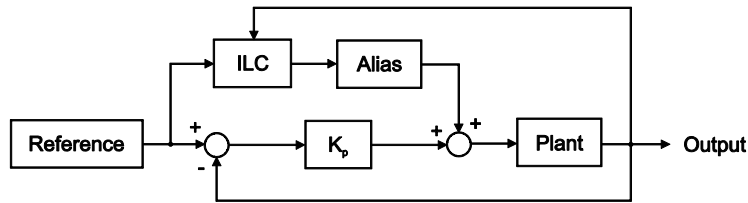


Fig. 3. Controller block diagram: P-type ILC with proportional feedback and aliasing.

### 3.2 Norm-Optimal ILC

Norm-Optimal ILC (NOILC) has received considerable attention in the ILC literature due to its mature theoretic basis<sup>5</sup>. In NOILC the input on the  $(k+1)^{th}$  trial is chosen to minimise

$$J_{k+1}(u_{k+1}) = \|e_{k+1}\|^2 + \|u_{k+1} - u_k\|^2 \quad (2)$$

The appearance of  $u_{k+1} - u_k$  allows control over the possibility of excessive control input signals. The noncausal solution

$$u_{k+1} = u_k + P^* e_{k+1} \quad (3)$$

where  $P^*$  is the plant adjoint, is transformed into a causal implementation incorporating both feedback and feedforward actions. To apply NOILC, (2) is expressed as a linear quadratic cost in  $e_{k+1}$  and  $u_{k+1} - u_k$  using weighting values of  $Q_n$  and  $R_n$  which affect the rate of error reduction and the rate of input change respectively. For ease of implementation,  $Q_n$  and  $R_n$  are generally chosen to be  $q_n I$  and  $r_n I$  respectively, where  $I$  is the identity matrix of appropriate dimensions, and  $q_n$  and  $r_n$  are scalars. As it is the ratio of  $q_n$  to  $r_n$  which adjusts algorithm performance,  $r_n$  has been set equal to unity. The greater control over the algorithm's convergence properties compared with P-type ILC is gained at the expense of significant additional computation, and the need for a full state observer.

#### 4. Experimental Results

Results for the two ILC controllers are compared against those produced using a standard optimal feedback controller. The linear time-invariant model of each axis was first expressed in the state-space form

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} \quad (4)$$

and a standard discrete linear quadratic tracker then implemented using the parameters  $Q$  and  $R$  to adjust the emphasis between error reduction and input energy minimisation respectively<sup>6</sup>. Values of  $R = 1$  and  $Q = q(C^T C)$  were chosen, where  $q$  is a scalar. Results are only provided for the  $x$ -axis since similar performance levels were achieved for the other axes. The gantry robot was reset to the initial conditions between iterations. The results which are presented correspond to the best performance achieved after tuning. In the case of the optimal controller greatest tracking accuracy was found using  $q = 10^9$  (larger values than this causing high frequency vibrations to become excessive). A standard state observer was implemented using a  $\lambda$  (lambda) scalar gain of 3 for each state<sup>7</sup>. A sample time of  $T_s = 0.001s$  was used in all experimental tests.

##### 4.1 P-type ILC

The mean squared error (mse) plot for the hybrid P-type learning and feedback controller during the first 100 iterations is shown in Figure 4.

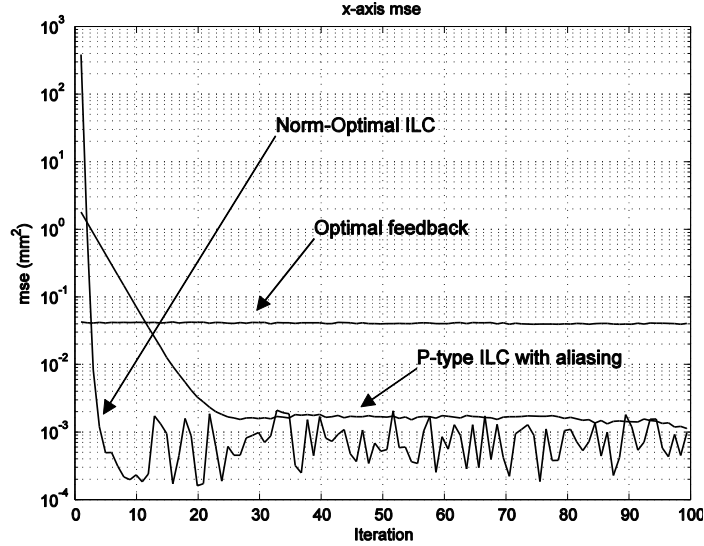


Fig. 4. Comparison of the mse for optimal feedback ( $q = 10^9$ ), ILC with aliasing and Norm-Optimal ILC ( $q_n = 10^5$ )

The best overall performance was achieved using a proportional feedback gain,  $K_p$ , equal to 600 and a learning gain,  $\beta$ , equal to 100. Increasing the former gain reduced the initial error, but led to greater fluctuation in the error mse over later iterations. The effect of  $K_p$  on the convergence rate and final level of error reached was small. Increasing the learning gain increased the convergence rate and reduced the final error level, however values greater than 100 led to vibrations in the gantry robot which quickly became excessive and potentially damaging. This figure also provides plots corresponding to the best performance that has been achieved using both NOILC and optimal control. The tracking performance using P-type ILC during the first iteration is quite poor, but this rapidly improves and by iteration 25, the tracking is excellent. Results have shown that 5000 iterations may be run with no sign of instability<sup>4</sup>.

#### 4.2 Norm-Optimal ILC

The best tracking performance using Norm-Optimal ILC was achieved using  $q_n$  and  $r_n$  values of  $10^5$  and 1 respectively, and the associated mse plot for the first 100 iterations appears in Figure 4. The convergence is very rapid and the minimum mse is significantly smaller than either that of P-type ILC or optimal control. Note that the vertical logarithmic scale accentuates the error fluctuation, and that the first trial error is greater than P-type ILC due to the absence of a feedback controller. Again 5000 iteration tests have been performed using this algorithm without any indication of instability<sup>8</sup>.

As well as reducing the tracking accuracy by over an order of magnitude compared with more traditional approaches, ILC can also adapt to changes in the plant dynamics, due to wear, temperature or environmental conditions. In the remaining experimental work, the P-type learning controller will be adopted since the absence of a plant model means it can be more readily applied to bulk manufacturing processes with minimal installation and set-up time. It is also simple to implement and requires minimal design and installation time. An additional advantage of P-type ILC is that due to its lack of computation, it can be implemented with no interval between iterations, thereby maximising the unit rate. In this case P-type ILC equates to its repetitive control equivalent.

### 5. Dual-loop Supervisory Control

Having selected an ILC controller for the gantry robot, and a simple feedback controller for the conveyor, the next objective is to develop a supervisory controller which reduces the relative error between these sub-systems by adjusting the reference trajectory of the gantry robot. It is important to define what is meant by relative error. During the great majority of the motion performed by each axis, exact synchronisation between the conveyor and robot is unimportant since the robot is either transporting a payload towards the conveyor or is returning to the dispenser to collect another. The conveyor simply moves a fixed distance at constant velocity. The relative error only becomes critical at the moment when the two machines must interact i.e., when the robot places the payload onto the conveyor. Essentially, the moment of placement is defined by that single sample instant,  $\mu$  ( $\mu$ ), when the payload is transferred from the robot to the conveyor. The relative error at all other sample instants can be ignored. In addition, the reference trajectories of the three robot axes must be synchronised, otherwise the robot cannot perform the correct 'pick and place' action. This implies that the relative error only needs to be measured between the conveyor axis and one of the robot axes. The obvious choice is the  $x$ -axis, because this is the only component of the robot which moves in the same direction of travel as the conveyor. The relative error will therefore be measured between the conveyor and the  $x$ -axis at sample instant  $\mu$ . Appropriate selection of this sampling instant is now required.

To reduce the relative error, it is proposed that a second learning loop is used to shift the reference trajectory of one machine in the time domain, so that the reference tracking errors of both machines are compensated for, and the resulting payload placement is correctly synchronised. This second loop will be applied to the gantry robot control system, so that the simple proportional feedback controller operating on the conveyor, can still be implemented

unaltered. The point of payload placement in the gantry trajectories (shown in Figure 2), occurs at sample interval 1000 (halfway through the trial). This sample instant is now referred to as  $\lambda$  (*lambda*), the expected interaction sample. While the learning capabilities of the robot controller suggest that there will be minimal robot tracking error at sample  $\lambda$ , the structure of the conveyor proportional feedback controller results in a constant position error, which is consistent across all repetitions. Effectively, at sample  $\lambda$  the conveyor is not yet in the correct position to receive the payload. An undefined period of time must pass until the conveyor has reached the required position, by which time the gantry robot has already placed the payload and is returning to the dispenser for the next. Sample instant  $\mu$  is chosen to be the sample when the error between the conveyor position and its reference at sample  $\lambda$  is minimised, i.e.

$$\varepsilon_k(\mu) = \min_{1 \leq n \leq N} (r_c(\lambda) - y_{c,k}(n)) \quad (5)$$

where  $\varepsilon_k$  (epsilon) is the conveyor relative error,  $r_c$  is the conveyor reference trajectory,  $y_{c,k}$  is the conveyor position for the  $k^{\text{th}}$  iteration of the task,  $n$  is the sample number and  $N=T/T_s$  is the number of samples per trial. Ideally,  $\varepsilon_k(\mu)$  is equal to zero. Depending upon the shape of the reference trajectories, it is possible that more than one value of  $\mu$  exists, in which case the one closest to the value of  $\lambda$  will generally be taken. As a consequence, the relative error for the  $x$ -axis can be defined for the same values of  $\mu$  and  $\lambda$  as

$$\chi_k = r_x(\lambda) - y_{x,k}(\mu) \quad (6)$$

where  $r_x$  and  $y_{x,k}$  are the  $x$ -axis reference trajectory and outputs respectively. The  $x$ -axis relative error  $\chi_k$  (chi) is used to drive the second learning loop, which shifts the robot trajectories by an integer number  $\zeta$  (zeta). For each iteration, the value of  $\zeta$  is fixed and it is only updated at the start of the next trial. The learning loop can be defined as

$$\zeta_{k+1} = \zeta_k + \lfloor \psi \chi_k \rfloor \quad (7)$$

where  $\psi$  (psi) is the supervisory learning gain and  $\lfloor \bullet \rfloor$  indicates the ‘floor’ function. Then the new robot reference trajectories for the current iteration are given by

$$\begin{aligned} r_x(n) &= r_x(n + \zeta_{k+1}) \\ r_y(n) &= r_y(n + \zeta_{k+1}) \\ r_z(n) &= r_z(n + \zeta_{k+1}) \end{aligned} \quad (8)$$



where  $r_y$  and  $r_z$  denote the  $y$  and  $z$ -axis references respectively. When  $N < (n + \zeta_{k+1}) < 1$  the reference wraps around i.e. for  $N < (n + \zeta_{k+1})$ , the shift is defined by  $(n + \zeta_{k+1} - N)$ , while when  $(n + \zeta_{k+1} < 1)$ , the shift is defined by  $(n + \zeta_{k+1} + N)$ . Figure 5 shows the program flowchart logic used to implement the dual-loop supervisory control. The basic concept is to identify the sample,  $\mu$ , when the conveyor has reached the payload placement position. Then the error between the  $x$ -axis position at sample  $\mu$  and the  $x$ -axis reference at  $\lambda$  is computed and used in the second learning loop to shift the robot trajectories in a way which seeks to minimise the relative error.

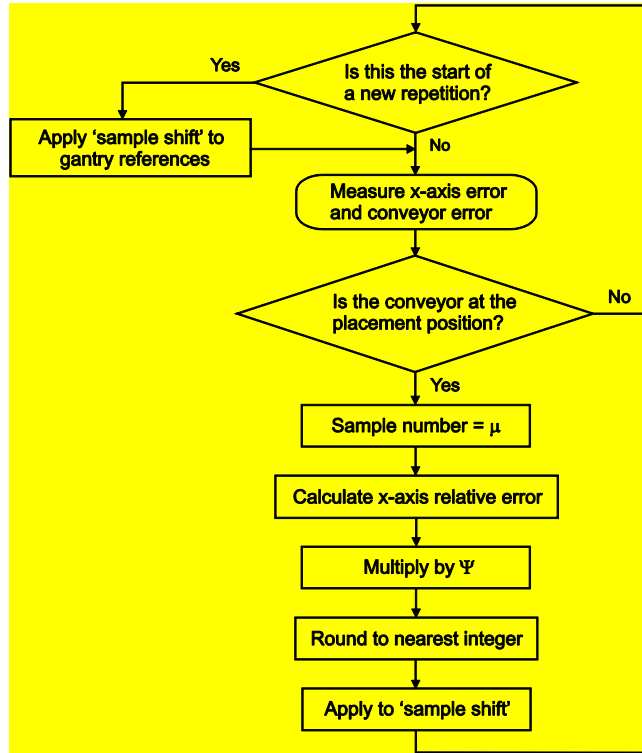


Fig. 5. Supervisory learning control, program flowchart

A simple approximation for the maximum value of  $\psi$  can be derived by assuming that the robot axes track their references accurately. Then,  $y_{x,k}(\mu) = r_x(\mu)$  and the desired reference shift  $\zeta_{k+1} = \mu - \lambda$ . It is then possible to determine the value of  $\psi$  which will achieve the required shift within one iteration. The analysis is as follows:

$$\begin{aligned} \chi_r &= r_x(\lambda) - r_x(\mu) \\ \mu - \lambda &= \zeta_{k+1} \end{aligned} \tag{9}$$

and because for the first iteration,  $\zeta_k = 0$ ,

$$\mu - \lambda = \psi(r_x(\lambda) - r_x(\mu))$$

$$\psi = \frac{\mu - \lambda}{r_x(\lambda) - r_x(\mu)}$$
(10)

$r_x$  and  $\lambda$  are known values and  $\mu$  can be found by operating the conveyor for one repetition. In practice,  $y_{x,k}$  tends to lag behind  $r_x$ . Therefore, the required shift is generally less than  $\mu - \lambda$  implying that  $\psi$  must be smaller than calculated.

Relative error results are now given to illustrate the benefits of using a dual loop supervisory system. Figure 6 displays the reference trajectories and measured displacement of the conveyor and  $x$ -axis, when the conveyor is operated with proportional feedback control and the robot operates with PID feedback control.

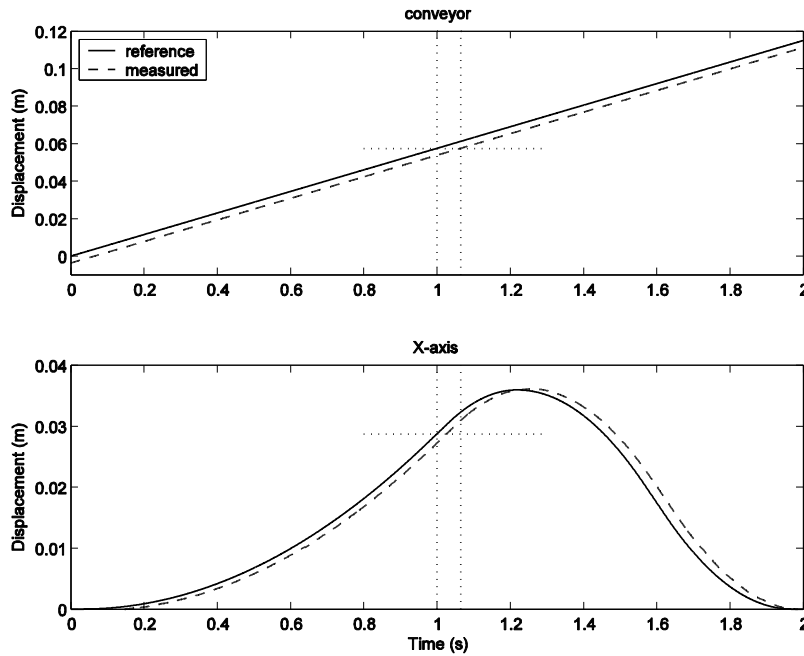


Fig. 6. References and output with PID feedback controller

The dotted construction lines have been added as a means of emphasising the relative error between the axes. The vertical construction line at time equal to 1 second corresponds to sample instant  $\lambda$ , the expected interaction sample. The horizontal construction line represents the reference demand at sample  $\lambda$ . The second vertical construction line corresponds to the point at which the measured conveyor output reaches the demand specified at  $\lambda$ . This is sample instant  $\mu$ , the actual interaction sample and is extended down to the  $x$ -axis plot. For zero relative error, the measured  $x$ -axis displacement must intersect the point described by the  $x$ -axis horizontal reference demand and the vertical

actual interaction sample. Clearly, for the feedback controllers, this does not occur. Both axes experience position tracking error and relative placement error.

The use of ILC causes the  $x$ -axis reference to be tracked extremely closely, which has the effect of actually increasing the relative position error beyond that achieved by feedback control alone. To compensate for the increase in relative error, the supervisory loop is added to the robot controller. The relative tracking performance with  $\psi$  equal to 10000 at repetition 1000 is displayed in Figure 7. The  $x$ -axis reference displayed in this figure is the original un-shifted reference, the actual shifted  $x$ -axis reference supplied to the controller resembles the measured output profile very closely. Though both axes experience a significant level of tracking error relative to their own original references, the relative error at the placement instant has been minimised.

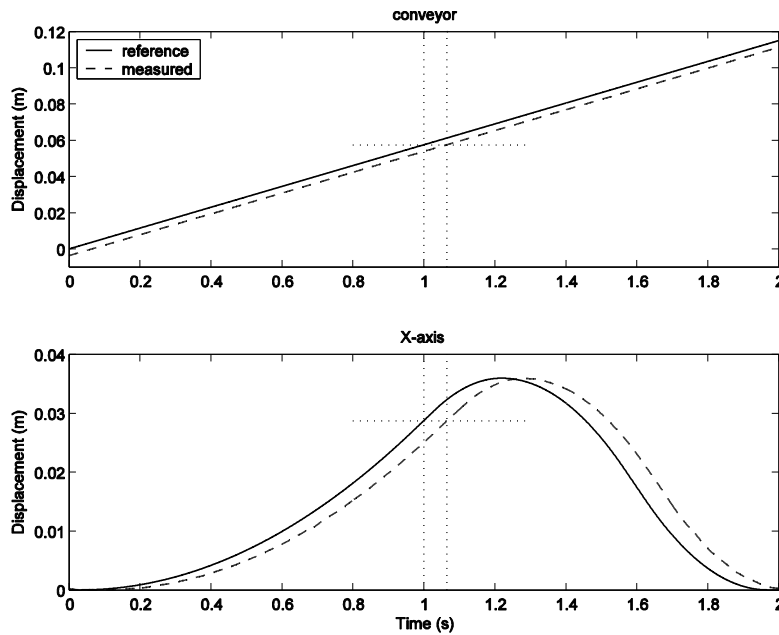


Fig. 7. References and output at iteration 1000 with  $\psi = 10000$

The effect of varying  $\psi$  on relative error reduction has been investigated experimentally. Figure 8 shows the data for relative error reduction. For all values of  $\psi$ , the other test parameters are held constant. As the supervisor gain increases, it is evident that the rate of relative error reduction changes. Higher gain implies faster convergence to minimal error. Note that when  $\psi$  is equal to zero, the relative error actually increases, as previously highlighted.

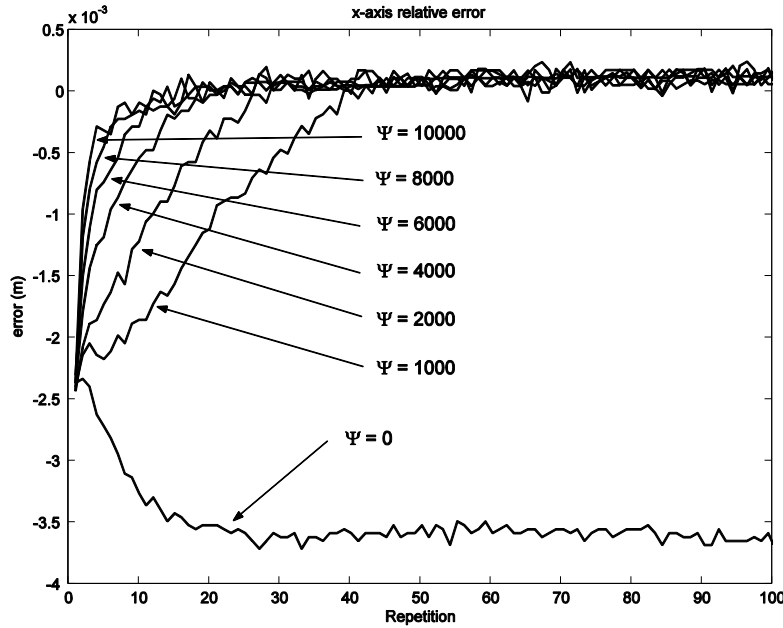


Fig. 8. Relative error reduction for different values of  $\psi$

Stability of the dual-loop supervisory system has been investigated by operating the combined controller for 1000 repetitions. Both learning loops indicate rapid, monotonic convergence to minimum error and good stability properties.

## 6. Conclusions

A combined ILC and supervisory control scheme has been proposed for the coordination of sub-systems involved in multi-axis automation. The tracking accuracy of two ILC algorithms has first been compared experimentally on a gantry robot, and then further results have confirmed the ability of an additional supervisory loop to synchronise its action to that of a conveyor. The reduction in both relative error and axis tracking error has been found to be considerable. Future work will focus on extending the methods for use with a greater number of sub-systems, and increasing the complexity of the interaction between them.

## References

1. Arimoto, S., Kawamura, S. and F. Miyazaki. Bettering operation of dynamic systems by learning : A new control theory for servomechanism or mechatronic systems. *Proceedings of 23rd IEEE Conference on Decision and Control, Las Vegas, NV*, 1064–1069, December 1984.

2. Kawamura, S., Miyazaki, F. and S. Arimoto. Applications of learning method for dynamic control of robot manipulators. *Proceedings of the 24th IEEE Conference on Decision and Control, Ft. Lauderdale, FL*, 1381–1386, December 1985.
3. Longman, R.W., Iterative learning control and repetitive control for engineering practice. *International Journal of Control*, **73**(10), 930–954, 2000.
4. Ratcliffe, J.D., Hätonen, J.J., Lewin, P.L., Rogers, E., Harte, T.J. and D.H. Owens. P-type iterative learning control for systems that contain resonance. *International Journal of Adaptive Control and Signal Processing*, **19**(10), 769–796, December 2005.
5. Amann, N., Owens, D.H. and E. Rogers. Iterative learning control using optimal feedback and feedforward actions. *International Journal of Control*, **65**(2), 277–293, 1996.
6. Jacquot, R.G., *Modern Digital Control Systems*, Marcel Dekker, INC., 1981. ISBN 0-8247-1322-2.
7. Luenberger, D.G., An introduction to observers. *IEEE Transactions on Automatic Control*, **16**(6), 596–602, December 1971.
8. Ratcliffe, J.D., Duinkerken, L., Lewin, L., Rogers, E., Hätonen, J.J., Harte, T.J. and D.H. Owens. Fast norm-optimal iterative learning control for industrial applications. *Proceedings of the 24th American Control Conference, Portland, USA*, 1951–1956, 2005.