

Discrete Predictive Optimal ILC Implemented on a Non-minimum Phase Experimental Test-bed

Chris Freeman, Paul Lewin and Eric Rogers

Abstract—Predictive Optimal ILC has been implemented on a non-minimum phase experimental test facility. Predictive Optimal ILC has first been derived for discrete-time systems, having previously only been formulated for the continuous case. Practical implementation issues have been considered and methods have been proposed to limit the necessary memory and calculation time required. The plant has been described and experimental results then presented for two cases of predictive horizon. The effect of variation of parameters used in the cost function has been described. The convergence rate observed in practice has been compared against a theoretical bound.

I. INTRODUCTION

Iterative learning Control (ILC) can be applied to systems which repetitively perform the same task with a view to sequentially improving accuracy. The task in question is regarded as the tracking of a given reference signal $r(t)$ or output trajectory for an operation on a specified time interval. The object of ILC is to use the repetitive nature of the process to progressively improve the accuracy with which the operation is achieved by updating the control input iteratively from trial to trial. A literature survey of ILC can be found in [1] and there exist textbooks on the subject [2], [3].

Norm Optimal ILC and Predictive Optimal ILC (which includes Norm Optimal ILC as a subset) have been implemented on a non-minimum phase experimental test-bed. Non-minimum phase systems have proved a significant challenge in the field of ILC and the experimental test-bed was designed specifically to include this characteristic. The Norm Optimal algorithm has previously been derived both in continuous time [4] and discrete time [5], [6]. The Predictive Optimal ILC law has been derived in continuous time only [7]. The absence of a discrete formulation of Predictive Optimal ILC necessitates its derivation here before it can be implemented on an experimental non-minimum phase spring-mass-damper system. The formulation of Predictive Optimal ILC for discrete systems closely follows the continuous case, the signal norms used, however, are the same as those used in the discrete derivation of Norm Optimal ILC [6].

II. ALGORITHM DERIVATION

The Predictive Optimal ILC problem is set up as in [6], using k to denote the trial index, and t the elapsed time during the trial. The trial is of length M samples and the

reference is denoted by $r(t)$. The ILC process is said to be convergent if and only if $\{u_k(t)\}_{k \geq 0}$, when applied to the plant, produces an output sequence $\{y_k(t)\}_{k \geq 0}$ with the property that the following limits exist:

$$\lim_{k \rightarrow \infty} y_k(t) = r(t) \quad \lim_{k \rightarrow \infty} u_k(t) = u_\infty(t) \quad (1)$$

$$\forall t \in [0, M]$$

In this paper the following sampled-time system is considered

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) & x(0) &= x_0 \quad 0 \leq t \leq M \\ y(t) &= Cx(t) & x &\in \mathbb{R}^n, \quad u \in \mathbb{R}^m, \quad y \in \mathbb{R}^p \end{aligned} \quad (2)$$

The state-space matrices A , B , C are assumed to be time-invariant for simplicity. Because only finite time intervals are considered in ILC, the output can be written in vector form by defining the supervectors

$$y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(M) \end{bmatrix} \quad u = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(M-1) \end{bmatrix} \quad (3)$$

An equivalent representation of (2) becomes

$$y = y_0 + Gu \quad (4)$$

with the matrix $G \in \mathbb{R}^{(pM) \times (mM)}$ defined as

$$G = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{M-1}B & CA^{M-2}B & \dots & CB \end{bmatrix} \quad (5)$$

with the vector of initial condition response $y_0 = [(CA)^T (CA^2)^T \dots (CA^M)^T]^T x_0$. The matrix G is invertible in the SISO case if and only if $CB \neq 0$, and if it has a delay, can be regularised (see [6]). Consider a tracking problem with reference trajectory $r(t)$, given for $1 \leq t \leq M$ (a relative degree of 1 is assumed for simplicity of representation). The tracking error is defined as

$$e = r - y = r - Gu - y_0 = (r - y_0) - Gu \quad (6)$$

where e and r are supervectors. Without loss of generality, it is possible to replace r by $r - y_0$ in the analysis and thence assume that $y_0 = 0$ or, equivalently, $x_0 = 0$. On completion of the k^{th} trial, the Predictive Optimal algorithm calculates the control input on the $(k+1)^{\text{th}}$ trial as the solution of the minimum norm optimisation problem:

$$u_{k+1} = \arg \min_{u_{k+1}} \{J_{k+1} : e_{k+1} = r - y_{k+1}, y_{k+1} = Gu_{k+1}\} \quad (7)$$

C. Freeman, P. Lewin and E. Rogers are with the School of Electronics and Computer Science, University of Southampton, University Road, Southampton SO17 1BJ, United Kingdom, email: ctf00r@ecs.soton.ac.uk

The discrete form of the cost appearing in [7] is written as

$$J_{k+1,N} = \sum_{i=1}^N \lambda^{i-1} (\|e_{k+i}\|_{\mathbb{Y}}^2 + \|u_{k+i} - u_{k+i-1}\|_{\mathbb{U}}^2) \quad (8)$$

and, with $N = 1$, this reduces to the cost used in [6]. The parameter $\lambda > 0$ is used to determine the importance of future errors. The norms $\|\cdot\|$ are appropriate norms for the input and output \mathbb{U} and \mathbb{Y} spaces respectively. These spaces are l_2 spaces of m and p vectors on $[0, M-1]$ and $[1, M]$. Written out as sums the index becomes

$$\begin{aligned} J_{k+1,N} = & \sum_{i=1}^N \lambda^{i-1} \left(\sum_{t=1}^M [r(t) - y_{k+i}(t)]^T Q(t) [r(t) - y_{k+i}(t)] \right. \\ & \left. + \sum_{t=0}^{M-1} [u_{k+i}(t) - u_{k+i-1}(t)]^T R(t) [u_{k+i}(t) - u_{k+i-1}(t)] \right) \end{aligned} \quad (9)$$

The weighting matrices $Q(t)$ and $R(t)$ must be symmetric and positive definite for all t . The cost function (8) is equivalent to (9) if the norms used are induced from the following inner products

$$\langle y_1, y_2 \rangle_{\mathbb{Y}} = y_1^T Q y_2 = \sum_{t=1}^M y_1(t)^T Q(t) y_2(t) \quad (10)$$

$$\langle u_1, u_2 \rangle_{\mathbb{U}} = u_1^T R u_2 = \sum_{t=0}^{M-1} u_1(t)^T R(t) u_2(t) \quad (11)$$

Following the reasoning of the continuous case [7], it is postulated that $J_{k+1,N}$ is a quadratic form in e_k , that is

$$J_{k+1,N}(u_{k+1}) = \langle e_k, Q_N e_k \rangle_{\mathbb{Y}} \quad (12)$$

where $Q_N \in \mathbb{R}^{(pM) \times (pM)}$ is a symmetric matrix. We can then write

$$\begin{aligned} J_{k+1,N} = & \|e_{k+1}\|_{\mathbb{Y}}^2 + \|u_{k+1} - u_k\|_{\mathbb{U}}^2 \\ & + \lambda \sum_{j=1}^{N-1} \lambda^{j-1} (\|e_{k+1+j}\|_{\mathbb{Y}}^2 + \|u_{k+1+j} - u_{k+j}\|_{\mathbb{U}}^2) \\ = & \|e_{k+1}\|_{\mathbb{Y}}^2 + \|u_{k+1} - u_k\|_{\mathbb{U}}^2 + \lambda J_{k+2,N-1} \end{aligned} \quad (13)$$

The controller on the $(k+1)^{th}$ trial is obtained with vector differential calculus from the required stationary condition,

$$\frac{1}{2} \frac{\partial J_{k+1,N}}{\partial u_{k+1}} = -G^T Q e_{k+1} + R(u_{k+1} - u_k) + \frac{\lambda}{2} \frac{\partial J_{k+2,N-1}}{\partial u_{k+1}} = 0 \quad (14)$$

The postulate (12) allows the substitution $J_{k+2,N-1} = \langle e_{k+1}, Q_{N-1} e_{k+1} \rangle_{\mathbb{Y}}$ so that

$$\frac{\partial J_{k+2,N-1}}{\partial u_{k+1}} = -2G^T Q Q_{N-1} e_{k+1} \quad (15)$$

Inserting this into (14) and rearranging produces

$$\begin{aligned} u_{k+1} &= u_k + R^{-1} G^T Q e_{k+1} + \lambda R^{-1} G^T Q Q_{N-1} e_{k+1} \\ &= u_k + G^* e_{k+1} + \lambda G^* Q_{N-1} e_{k+1} \end{aligned} \quad (16)$$

The substitution $G^* = R^{-1} G^T Q$ can be made since $R^{-1} G^T Q$ is equivalent to the adjoint operator with respect to the weighted inner product equations (10,7) [8]. Equation (16) is identical to the continuous time case and, setting $N = 1$, equates to the discrete Norm Optimal solution, since $Q_0 = 0$. Manipulation of (16) yields the error evolution

$$e_{k+1} = L_N e_k, \quad L_N = [I + G G^* (I + \lambda Q_{N-1})]^{-1} \quad (17)$$

where $L_N \in \mathbb{R}^{(pM) \times (pM)}$. To determine Q_N , (16) is used to write

$$\begin{aligned} \|u_{k+1} - u_k\|_{\mathbb{U}} = & (R^{-1} G^T Q (I + \lambda Q_{N-1}) e_{k+1})^T R (R^{-1} G^T Q (I + \lambda Q_{N-1}) e_{k+1}) \\ = & (G^T (I + \lambda Q_{N-1}) e_{k+1})^T Q (R^{-1} G^T Q (I + \lambda Q_{N-1}) e_{k+1}) \\ = & \langle e_{k+1}, (I + \lambda Q_{N-1}) G G^* (I + \lambda Q_{N-1}) e_{k+1} \rangle_{\mathbb{Y}} \end{aligned} \quad (18)$$

so that (13) becomes

$$\begin{aligned} J_{k+1,N} = & \langle e_{k+1}, e_{k+1} \rangle_{\mathbb{Y}} + \lambda \langle e_{k+1}, Q_{N-1} e_{k+1} \rangle_{\mathbb{Y}} \\ & + \langle e_{k+1}, (I + \lambda Q_{N-1}) G G^* (I + \lambda Q_{N-1}) e_{k+1} \rangle_{\mathbb{Y}} \\ = & \langle e_{k+1}, (I + \lambda Q_{N-1}) [I + G G^* (I + \lambda Q_{N-1})] e_{k+1} \rangle_{\mathbb{Y}} \\ = & \langle e_k, L_N (I + \lambda Q_{N-1}) e_k \rangle_{\mathbb{Y}} \end{aligned} \quad (19)$$

Comparison with (12) yields the following recursive equation for Q_N :

$$\begin{aligned} Q_N &= L_N (I + \lambda Q_{N-1}) \\ &= [I + G G^* (I + \lambda Q_{N-1})]^{-1} (I + \lambda Q_{N-1}) \end{aligned} \quad (20)$$

It is then possible to express L_N as a recursive relation by eliminating Q_N

$$L_N(H, \lambda) = [(1 + \lambda)I + H - \lambda L_{N-1}]^{-1}, \quad N = 1, 2, \dots \quad (21)$$

where $L_0 = I$, and $H := G G^*$ has been used for conciseness. $L_N(H, \lambda)$ is a symmetric matrix and, if H is positive and bounded in norm, is positive according to the bound

$$L_N(H, \lambda) \geq (I + \lambda + H)^{-1} > 0, \quad \forall 0 < \lambda < \infty, \quad N = 1, 2, \dots \quad (22)$$

The upper bound of L_N (the norm of L_N) is important to ILC convergence since the norm of the error at the k^{th} trial can be bounded by $\|e_k\| \leq \|L_N\|^k \|e_0\|$. Therefore it is sufficient for convergence that $\|L_N\|$ is less than one. If the plant is bounded below such that

$$\langle e, H e \rangle_{\mathbb{Y}} \geq \sigma^2 \|e\|_{\mathbb{Y}}^2 \quad \forall e \in \mathbb{Y} \quad (23)$$

then $L_N \leq l_N(\sigma^2, \lambda) I$ with

$$l_N(\sigma^2, \lambda) = \frac{1}{1 + \lambda + \sigma^2 - \lambda l_{N-1}(\sigma^2, \lambda)}, \quad l_0 = 1 \quad (24)$$

and the error sequence is bounded by $\|e_{k+1}\| \leq l_N(\sigma^2, \lambda) \|e_k\|$. If $\sigma > 0$ then $\|L_N\| < 1 \quad \forall N \geq 1$ then geometric convergence is assured. In the case of $N = 1$ then $L_1 = [I + H]^{-1}$ and $l_1 = \frac{1}{1 + \sigma^2}$ which agrees with the rate of convergence seen in [6]. In the case of $N = 2$ then $L_2 = [(1 + \lambda)I + H - \lambda(I + H)^{-1}]^{-1}$ and we can write $l_2 = \frac{1}{1 + \lambda + \sigma^2 - \lambda(\frac{1}{1 + \sigma^2})}$ which will later be compared with the convergence rate seen experimentally. As discussed in

[6], σ can be changed by the design rates Q and R since the definition of σ can be written as

$$e^T G^T Q G e \geq \sigma^2 e^T R e \quad \forall e \in \mathbb{Y} \quad (25)$$

III. CAUSAL ALGORITHM FORMULATION

In this section the cost function (9) is transformed until it is in the form of

$$J_{k+1,N} = \|e_{k+1}\|_{\mathbb{Y}}^2 + \|u_{k+1} - u_k\|_{\mathbb{U}}^2 \quad (26)$$

which is the cost function used in the discrete Norm Optimal ILC derivation [6]. The corresponding causal solution can then be applied to form the necessary update. In order to do this (9) is written out in full as

$$J_{k+1,N} = \sum_{t=1}^M \begin{bmatrix} e_{k+1} \\ e_{k+2} \\ \vdots \\ e_{k+N} \end{bmatrix}^T \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & \lambda Q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda^{N-1} Q \end{bmatrix} \begin{bmatrix} e_{k+1} \\ e_{k+2} \\ \vdots \\ e_{k+N} \end{bmatrix} + \sum_{t=0}^{M-1} \begin{bmatrix} u_{k+1} - u_k \\ u_{k+2} - u_{k+1} \\ \vdots \\ u_{k+N} - u_{k+N-1} \end{bmatrix}^T \begin{bmatrix} R & 0 & \dots & 0 \\ 0 & \lambda R & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda^{N-1} R \end{bmatrix} \begin{bmatrix} u_{k+1} - u_k \\ u_{k+2} - u_{k+1} \\ \vdots \\ u_{k+N} - u_{k+N-1} \end{bmatrix} \quad (27)$$

Using the substitution

$$\begin{bmatrix} u_{k+1} - u_k \\ u_{k+2} - u_{k+1} \\ u_{k+3} - u_{k+2} \\ \vdots \\ u_{k+N} - u_{k+N-1} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{k+1} - u_k \\ u_{k+2} - u_k \\ u_{k+3} - u_k \\ \vdots \\ u_{k+N} - u_k \end{bmatrix} \quad (28)$$

the difference-in-input vector of (27) can be rewritten. The implementation uses a number of parallel plants, only the first of which is the actual plant which produces u_{k+1} . The other ‘virtual’ plants are simulated, their only purpose being to contribute to the calculation of u_{k+1} . The errors are written in vector form as

$$\begin{bmatrix} e_{k+1} \\ e_{k+2} \\ \vdots \\ e_{k+N} \end{bmatrix} = \begin{bmatrix} r \\ r \\ \vdots \\ r \end{bmatrix} - \begin{bmatrix} G & 0 & \dots & 0 \\ 0 & G & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & G \end{bmatrix} \begin{bmatrix} u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+N} \end{bmatrix} \quad (29)$$

According to the block diagonal plant matrix from (29) the extended plant matrices are defined

$$A_N = \text{diag}\{A, A, \dots, A\}, \quad B_N = \text{diag}\{B, B, \dots, B\} \\ C_N = \text{diag}\{C, C, \dots, C\} \quad (30)$$

and also the following extended weight matrices

$$Q_N = \text{diag}\{Q, \lambda Q, \dots, \lambda^{N-1} Q\}, \\ R_N = \begin{bmatrix} (1+\lambda)R & -\lambda R & 0 & \dots & 0 \\ -\lambda R & (\lambda+\lambda^2)R & -\lambda^2 R & \dots & 0 \\ 0 & -\lambda^2 R & (\lambda^2+\lambda^3)R & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda^{N-1} R \end{bmatrix} \quad (31)$$

The optimisation problem is now in the form of the causal LQR problem proposed in [6] which has the following solution

$$\begin{bmatrix} u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+N} \end{bmatrix} = \begin{bmatrix} u_k \\ u_k \\ \vdots \\ u_k \end{bmatrix} - [\{B_N^T K(t) B_N + R_N(t)\}^{-1} B_N^T K(t) \\ \times A_N \left(\begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+N} \end{bmatrix} - \begin{bmatrix} x_k \\ x_k \\ \vdots \\ x_k \end{bmatrix} \right) + R_N^{-1}(t) B_N^T \xi_{k+1,N}(t)] \quad (32)$$

where the state feedback gain matrix $K(t)$ is the solution of the discrete matrix Riccati equation on the interval $t \in [0, M-1]$

$$K(t) = A_N^T K(t+1) A_N + C_N^T Q_N(t+1) C_N - [A_N^T K(t+1) \\ \times B_N \{B_N^T K(t+1) B_N + R_N(t+1)\}^{-1} B_N^T K(t+1) A_N] \quad (33)$$

with the terminal condition $K(M) = 0$. The feedforward term $\xi_{k+1,N}(t)$ is generated by

$$\xi_{k+1,N}(t) = \{I + K(t) B_N R_N^{-1}(t) B_N^T\}^{-1} \\ \times \{A_N^T \xi_{k+1,N}(t+1) + C_N^T Q_N(t+1) \begin{bmatrix} e_k(t+1) \\ e_k(t+1) \\ \vdots \\ e_k(t+1) \end{bmatrix}\} \quad (34)$$

with the terminal condition $\xi(M) = 0$. The algorithm uses $N-1$ models of the plant in parallel with the actual plant for the computation of the optimal input for the N -step predictive algorithm. Only knowledge of the state $x_{k+1}(t)$ of the actual plant is a potential problem as the others are directly available from the simulated plants. If it is not available an observer can be constructed to estimate it.

IV. NON-MINIMUM PHASE PLANT

The experimental test-bed has previously been used to evaluate a number of ILC schemes (see [9], [10] for details) and consists of a rotary mechanical system of inertias, dampers, torsional springs, a timing belt, pulleys and gears. The non-minimum phase characteristic is achieved by using the arrangement shown in Figure 1 where θ_i and θ_o are the input and output positions, J_r and J_g are inertias, B_r is a damper, K_r is a spring and G_r represents the gearing. A further spring-mass-damper system is connected to the input in order to increase the relative degree and complexity

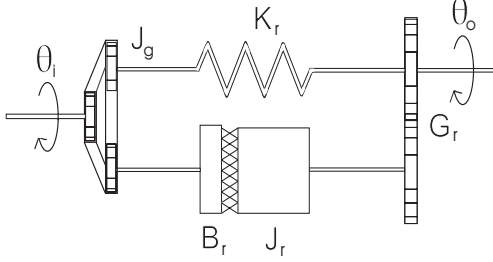


Fig. 1. Non-minimum phase section

of the system. A 1000 pulse/rev encoder records the output shaft position and a standard squirrel cage induction motor drives the load. The continuous time transfer function used to model the plant is given by

$$G(s) = \frac{1.202(4-s)}{s(s+9)(s^2+12s+56.25)} \quad (35)$$

A PID loop around the plant is used in order to act as a pre-stabiliser and provide greater stability. The PID gains used are $K_p = 137$, $K_i = 5$ and $K_d = 3$. The resulting closed-loop system constitutes the system to be controlled. The values of the Kalman covariance matrices, Q_e and R_e , used to construct the observer, were set at 10 and 1 respectively.

V. PRACTICAL IMPLEMENTATION

The implementation of the $N = 1$ case has been achieved using a sampling frequency of 100Hz. The two demands used, appearing as signals in Figures 6 and 7, are each six seconds long, giving $M = 600$. The first three seconds of each demand is zero in order to reduce the large values of initial input that would be necessary to achieve an arbitrary non-zero initial output. This is due to the combination of the presence of a deadzone between the input and output, the non-minimum phase characteristic, and the large time constant of the system. The discrete plant is SISO and has 4 states which means that the matrix $K = [K(0) K(1) \dots K(M-1)]$ has $600 \times 4N \times 4N$ elements and the matrix $\xi = [\xi(0) \xi(1) \dots \xi(M-1)]$ has $600 \times 4N$ elements. A matrix with $600 \times 4N$ elements is also required to store the system states from trial to trial. Each $K(t)$ only needs to be calculated once before the experiment begins. However, in order to reduce the calculations performed, each value can be overwritten by

$$K'(t) = \{I + K(t)B_N R_N^{-1}(t)B_N^T\}^{-1} \quad t \in [0, M-1] \quad (36)$$

which is also iteration independent. The only other occurrence of $K(t)$ occurs in the input update expression (32) and is given by

$$V(t) = \{B_N^T K(t)B_N + R_N(t)\}^{-1} B_N^T K(t)A_N \quad (37)$$

which also contains $600 \times 2N$ elements. Therefore, before overwriting K with K' , $V = [V(0) V(1) \dots V(M-1)]$ is stored in memory. This increased storage is justified by the simpler structure of K' in comparison with K which is a

result of the zero entries in B . For the present plant this allows each value of $K'(t)$ to be written as the sum of a $2N \times 2N$ identity matrix plus a matrix in which only every fourth column is non-zero. In the present case, the memory requirements for storing K' and V can thus be reduced in comparison with storing K by a factor of $\frac{4N}{N+1}$.

Furthermore, if there is the capacity for extra calculation time within each sample instant, the value of $K(t)$ can be calculated and stored only at those sample instants $0 \leq t \leq c$, where $c \leq M-1$. This negates the necessity for the storage of $M-1-c$ values of K' and V . The missing values of these can then be calculated between samples without ever being stored. Even using these methods, it has been necessary to reduce the sampling frequency from 100Hz to 70Hz in the case of $N = 2$.

VI. RESULTS

Figures 2 and 3 show results obtained for a sinewave and a repeating sequence demand. The total error incurred during each trial over the course of 200 iterations is plotted against the trial number in each case. The normalised error (NE) is simply the total error produced in a trial multiplied by a scalar chosen so that a constant zero plant output produces a NE value of unity. Without learning, a NE of 1 would constantly be incurred. Tests have been stopped if excessive vibration of the plant output makes it unsafe to continue, this can occur at fairly low levels of NE. Different values of Q are used, the value of R being fixed at unity. It can be seen that increasing Q increases the convergence speed but leads to large transients and divergence of error if set too high. The repeating sequence demand, which includes higher frequency components, exhibits slower convergence, greater transients, and a greater value of the minimum error. In both cases a value of $Q = 2$ produces good results. Figures 4 and 5 show similar results, but

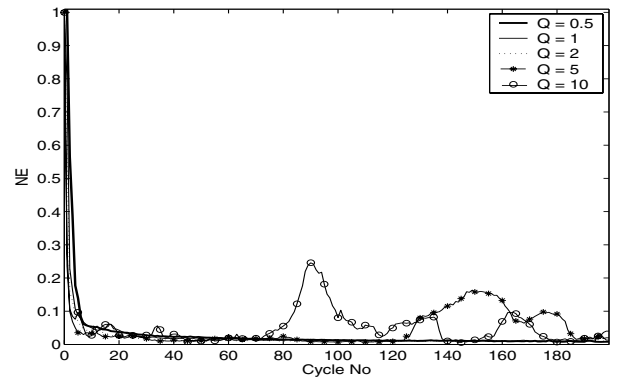


Fig. 2. Cycle error results for sinewave demand, $N = 1$

using a predictive horizon, $N = 2$. In this case the number of iterations performed has been increased to 400. Three values of λ are used, each with two different values of Q . R is again fixed at unity throughout. It is clear that lower values of λ lead to a reduced convergence rate and reduced

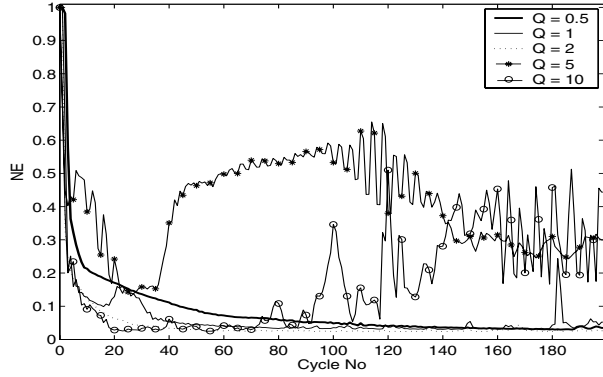


Fig. 3. Cycle error results for repeating sequence demand, $N = 1$

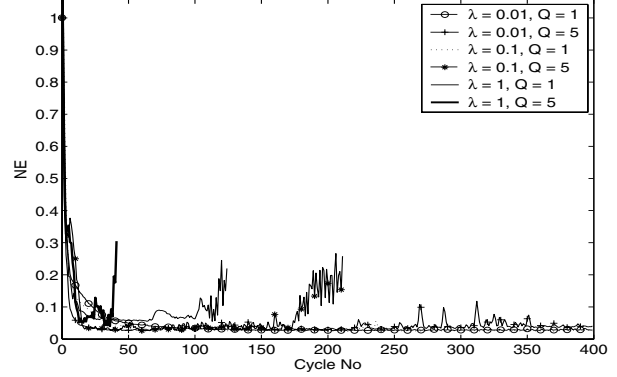


Fig. 5. Cycle error results for repeating sequence demand, $N = 2$

transients. This is expected since λ dictates the amount of data used from the future trials. In practice the increase of λ produces a large change in $u(t)$ at the beginning of each trial meaning that frequently tests must be discontinued due to the possibility of damage to the plant. This effect is due to the presence of a deadzone in the plant and occurs whether the plant output position is reset inbetween trials, or whether the next trial follows immediately on from the preceding one. Figures 6 and 7 show the reference,

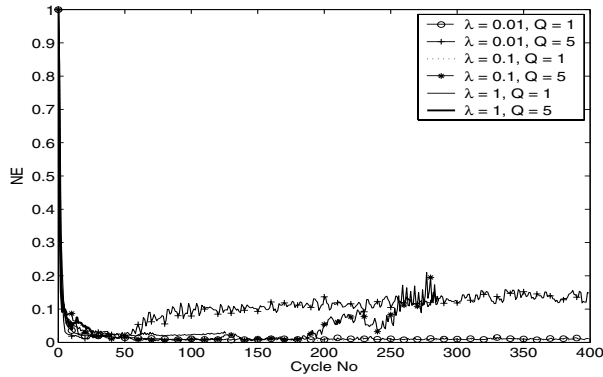


Fig. 4. Cycle error results for sinewave demand, $N = 2$

input and output signals for the sinewave and repeating sequence demands respectively. These were recorded on the 400th cycle of the test which used the parameters $N = 2$, $\lambda = 0.1$ and $Q = 1$. The demand is seen to be tracked accurately without excessive control action. Figures 8 and 9 compare results for varying amounts of prediction, using $Q = 1$ and $Q = 5$ respectively. Only the repeating sequence demand is used and the $N = 1$ result is compared with the $N = 2$ used in conjunction with varying λ . It can be seen that prediction ($N = 2$) is capable of adding robustness to the algorithm with a suitable value of λ since the error shows reduced transients and more trials can be performed before the tests are halted. Only a slight increase in convergence rate is evident from the experimental results. The following statements can hence be made concerning

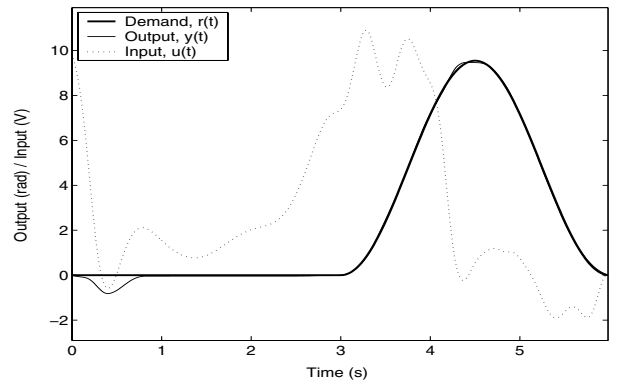


Fig. 6. Signals at 400th cycle for sinewave demand, $N = 2$, $\lambda = 0.1$, $Q = 1$

the design parameters:

- Use of prediction can increase robustness
- The level of Q dictates convergence speed for a fixed R but causes instability if excessively high
- The convergence of the repeating sequence demand is slower than for the sinewave demand
- Increasing the predictive horizon increases the convergence speed
- Increasing λ causes large input changes at the start of a trial

The bound on the H given by (23) does not provide useful information for the plant in question since σ^2 is very large in this case. However using the l_∞ norm in place of the l_2 norm allows the approximation

$$\sigma^2 \approx \frac{1}{2 \min_w |G(w)|} \quad (38)$$

for the case that $Q = R = 1$, which is considered next. If the assumption is made that frequencies present in the demand dominate the learning process, then the bound on H alters depending on the demand. The sinewave demand has negligible frequency content above 3.8 rad^{-1} and so the magnitude plot of $G(w)$ can be consulted to give an approximate value of $\sigma^2 = 1.25$. Similarly the repeating

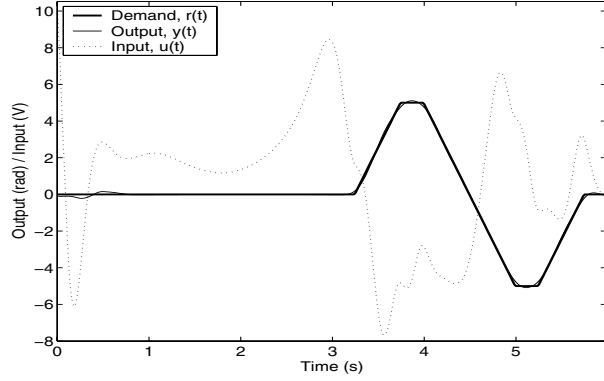


Fig. 7. Signals at 400th cycle for repeating sequence demand, $N = 2$, $\lambda = 0.1$, $Q = 1$

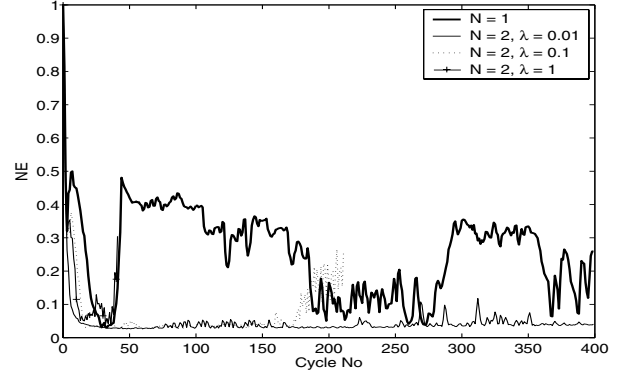


Fig. 9. Cycle error results for repeating sequence demand, $Q = 5$

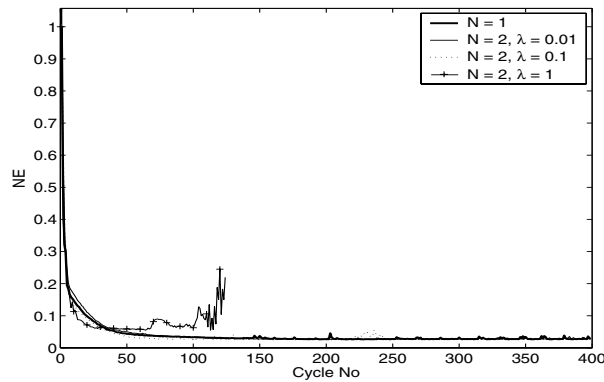


Fig. 8. Cycle error results for repeating sequence demand, $Q = 1$

sequence demand yields $\sigma^2 = 0.56$. For the case of $N = 1$ these values, when inserted in (24), lead to convergence rates of $l_1 = 0.44$ and 0.6 respectively, and $Q = 1$. These are close to the initial learning rates observed in practice. For the case of $N = 2$ the convergence rates for the two sequences become $l_2 = 0.443$ and 0.64 for $\lambda = 0.01$, $l_2 = 0.43$ and 0.625 for $\lambda = 0.1$, and $l_2 = 0.36$ and 0.51 for $\lambda = 1$. The convergence rates are plotted in Figure 10 for the repeating sequence. Comparison with the experimental rates seen in Figure 8 shows these theoretical bounds are accurate models for early trials. The lack of robustness with respect to high frequency modelling then reduce the learning rates significantly.

VII. CONCLUSIONS

Predictive Optimal ILC has been derived for discrete-time systems. The algorithm has been used on an experimental non-minimum phase spring-mass-damper system. Two values of predictive horizon have been investigated. Excellent results have been achieved and the effect of parameter variation has been investigated. Practical implementational issues have been discussed and methods for increasing the computational efficiency have been proposed. The convergence rates observed have been compared to theoretical values.

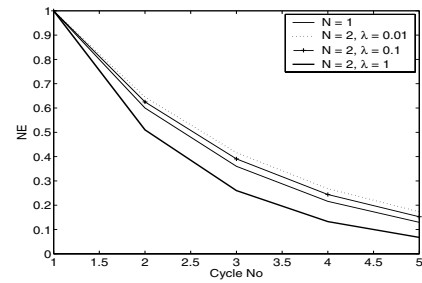


Fig. 10. Theoretical convergence rates for repeating sequence with $Q = 1$

VIII. FUTURE WORK

The effect of varying the sampling frequency is a natural area of future study. The results presented here will also be compared with those achieved using the same algorithm on an industrial gantry robot. The algorithm robustness is an open area for further study.

REFERENCES

- [1] K. L. Moore, "Iterative learning control - an expository overview," *Applied and Computational Controls, Signal Processing and Circuits*, no. 1, pp. 151 – 214, 1998.
- [2] —, *Iterative Learning Control for Deterministic Systems*. Springer-Verlag, 1993.
- [3] Z. Bien and J. Xu, *Iterative Learning Control, Analysis, Design, Integration and Applications*. Kluwer Academic Publishers, 1998.
- [4] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control using optimal feedback and feedforward actions," *International Journal of Control*, vol. 65, no. 2, pp. 277–293, 1996.
- [5] —, "Iterative learning control for discrete-time systems using optimal feedback and feedforward actions," in *Proceedings of the 34th Conference on Decision and Control*, December 1995, pp. 1696–1701.
- [6] —, "Iterative learning control for discrete-time systems with exponential rate of convergence," in *IEE Proceedings - Control Theory Applications*, vol. 143, no. 23, March 1996, pp. 217–224.
- [7] —, "Predictive optimal iterative learning control," *International Journal of Control*, vol. 69, no. 2, pp. 203–226, 1998.
- [8] D. G. Luenberger, *Optimization by vector space methods*. John Wiley and Sons, New York, 1969.
- [9] C. Freeman, P. Lewin, and E. Rogers, "Experimental comparison of extended simple structure ILC algorithms for a non-minimum phase plant," *Submitted to ASME*, 2004.
- [10] —, "Experimental evaluation of ILC algorithms for non-minimum phase systems," *Accepted: International Journal of Control*, 2004.