UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

Experimental Verification of Constrained Iterative Learning Control Using Successive Projection

Bing Chu*, Zhonglun Cai†, David H Owens*‡§, Eric Rogers*, Chris T Freeman* and Paul L Lewin*
*Electronics and Computer Science,
University of Southampton, Southampton SO17 1BJ, UK
(e-mail: {b.chu, etar, cf, pll}@ecs.soton.ac.uk)
†Faculty of Engineering and the Environment,
University of Southampton, Southampton SO17 1BJ, UK
(e-mail: z.cai@soton.ac.uk)
‡Department of Automatic Control and Systems Engineering,
University of Sheffield, Sheffield, S1 3JD, UK
(e-mail: d.h.owens@shef.ac.uk)
§The Italian Institute of Technology,
Via Morego, 30, 16163-Genova, Italy

Abstract—**In many practical applications, constraints are often present on, for example, the magnitudes of the control inputs. Recently, based on a novel successive projection framework, two constrained iterative learning control (ILC) algorithms were developed with different convergence properties and computational requirements. This paper investigates the effectiveness of these two methods experimentally on a gantry robot facility, which has been extensively used to test a wide range of linear model based ILC algorithms. The results obtained demonstrate the effectiveness of the algorithms in solving one form of the general constrained ILC problem.**

I. Introduction

Iterative Learning Control (ILC) is a technique for controlling systems operating in a repetitive or trial-to-trial mode with the requirement that a reference trajectory $y_{ref}(p)$ defined over a finite interval $0 \leq p \leq \alpha$ is followed to a high precision, where the constant α denotes the finite trial duration or length. The basic idea in ILC is that information from previous trials is used to update the control input for the next trial in order to sequentially improve performance. Moreover, the next trial input is typically computed during the time taken to reset between successive trials.

Since the original work by [1], ILC has developed into an established area in control systems research and applications. Initial sources for the relevant literature are the survey papers [2] and [3]. These show that a wide range of algorithms have been developed, many of which, particularly those based on a linear plant model, have been experimentally tested.

In many practical applications, constraints are present due, for example, to physical limitations or performance requirements. Hence ILC design must take these constraints into account but most of the currently available ILC results are for unconstrained systems and there are relatively few results for the constrained case. One set of results is due to [4] where a novel nonlinear controller for process systems with input constraints is developed where the learning scheme requires relatively little knowledge of the process model. In [5] an ILC problem with soft constraints is studied where Lagrange multiplier methods are used to develop a solution. [6] uses quadratic optimal design to formulate a constrained ILC problem and suggests that a quadratic optimal design has the capability of dealing with constraints. Also [7] uses a constrained convex optimization technique to solve the constrained ILC problem for linear systems with saturation constraints.

Recently, the ILC design problem with general convex input constraints has been considered in [8]. This work shows that the constrained ILC problem can be formulated in a recently developed successive projection framework, which provides an intuitive but rigorous method for system analysis and design. Based on this, a systematic approach for constraints handling is provided and two algorithms to solve this problem developed. The convergence analysis shows that when perfect tracking is possible, both algorithms can achieve this goal whereas the computation of one algorithm is much less than the other at the cost of slightly slower convergence rate. When perfect tracking is not possible, both algorithms converge to asymptotic values representing a ”best fit” solution. Again the more computationally complex algorithm has the best convergence properties. It was also found that the input and output weighting matrices have an interesting effect on the convergence properties of the algorithms.

The main aim of this paper is to give experimental results to verify the effectiveness of the constrained ILC algorithms using a gantry robot facility previously used to test a wide range of ILC algorithms, including Norm Optimal ILC (NOILC) [9]. The paper is organized as follows. In Section 2, the required results from the derivation of the constrained ILC algorithms are given. Then in Sections 3 and 4, the gantry robot facility and the test parameters are described. The experimental results are given in Section 5 and Section 6 gives conclusions and suggestions for further research.

II. ITERATIVE LEARNING CONTROL FOR CONSTRAINED LINEAR SYSTEMS

Consider the following discrete linear time-invariant system

$$\begin{aligned} x_k(t+1) &= Ax_k(t) + Bu_k(t), \\ y_k(t) &= Cx_k(t), \end{aligned} \tag{1}$$

where t is the time index (i.e. sample number), $k \geq 0$ is the trial index and $x_k(0) = x_0, k = 1, 2, \cdots$ is the same for all trials. The control objective is to track a given reference signal $r(t)$ and $u_k(t), x_k(t), y_k(t)$ are input, state and output vectors, respectively, of the system on trial k. In operation, a trial is completed, the system is reset and a new trial begins. The ILC design uses information from previous trial(s) to compute the control input for the next trial in a manner that improves tracking performance from trial-to-trial.

Before presenting the main results, the operator form of the dynamics is introduced using the well-known lifted-system representation, which provides a straightforward "$N \times N$ matrix" approach in the analysis of discrete-time ILC [10], [11].

Assume, for simplicity, the relative degree of the system is unity, i.e. the generic condition $CB \neq 0$ is satisfied (the case when the system relative degree is greater than one follows as an obvious generalization), then the system state-space model (1) on trial k can be written in the form

$$y_k = Gu_k + d, \tag{2}$$

where G and d are the $N \times N$ and $N \times 1$ matrices

$$\begin{aligned} G &= \begin{bmatrix} CB & 0 & \cdots & 0 & 0 \\ CAB & CB & \ddots & 0 & 0 \\ CA^2B & CAB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & CB & 0 \\ CA^{N-1}B & \cdots & \cdots & CAB & CB \end{bmatrix} \\ d &= \begin{bmatrix} CAx_0 & CA^2x_0 & CA^3x_0 & \cdots & CA^Nx_0 \end{bmatrix}^T. \end{aligned} \tag{3}$$

The $N \times 1$ vectors of input, output and reference time series u_k, y_k and r are defined as

$$\begin{aligned} u_k &= \begin{bmatrix} u_k(0) & u_k(1) & \cdots & u_k(N-1) \end{bmatrix}^T, \\ y_k &= \begin{bmatrix} y_k(1) & y_k(2) & \cdots & y_k(N) \end{bmatrix}^T, \\ r &= \begin{bmatrix} r(1) & r(2) & \cdots & r(N) \end{bmatrix}^T. \end{aligned} \tag{4}$$

Also no loss of generality arises from assuming that $d = 0$ (non-zero d can be incorporated into the reference signal by replacing r with $r - d$). Hence (2) becomes

$$y_k = Gu_k, \tag{5}$$

where G is nonsingular and hence invertible.

Tracking error improvements from trial-to-trial are achieved in ILC by the design of a control law of the following general form

$$u_{k+1} = f\left(e_{k+1}, \ldots, e_{k-s}, u_k, \cdots, u_{k-r}\right). \tag{6}$$

When $s > 0$ or $r > 0$, (6) is termed a higher order updating law. This paper only considers algorithms of the form $u_{k+1} = f\left(e_{k+1}, e_k, u_k\right)$. For higher order algorithms, refer to [12], [13] and the references therein. The ILC design problem can now be stated as finding a control updating law (6) such that the system output has the asymptotic property that $e_k \to 0$ as $k \to \infty$.

There are many design methods to solve the ILC problem. The one used in this paper is based on a quadratic (norm) optimal formulation [14] where, on each trial, a performance index is minimized to obtain the system input time series vector to be used on the next trial. The basis of this paper is NOILC that designs the control input to minimize the performance index

$$J_{k+1}(u_{k+1}) = \|e_{k+1}\|_Q^2 + \|u_{k+1} - u_k\|_R^2, \tag{7}$$

subject to the constraint $e_{k+1} = r - Gu_{k+1}$, where G is the operator representation of the system (1) and Q and R are positive definite weighting matrices. Also $\|e\|_Q^2$ denotes the quadratic form e^TQe and similarly for $\|\cdot\|_R^2$. Solving this optimization problem gives the following optimal choice for the time series vector u_{k+1}

$$u_{k+1} = u_k + R^{-1}G^TQe_{k+1} \tag{8}$$

which, when $k \to \infty$, asymptotically achieves perfect tracking. This well-known NOILC algorithm has many appealing properties including implementation in terms of Riccati state feedback. More details concerning NOILC can be found in [14]–[17].

In practical applications, system constraints are encountered and of different forms, e.g., input constraints, input rate constraints and state or output constraints. Constraints can be divided into two classes termed hard and soft, respectively. Hard constraints are those on magnitude(s) at each point in time, for example, output limits on actuators. Soft constraints are those that are applied to the whole function rather than its point-wise values e.g. constraints on total energy usage. This paper only considers input constraints.

Suppose the input is constrained to be in a set Ω, which is taken to be a closed convex set in some Hilbert space H. In practice, the set Ω is often of simple structure. For example, the following are often encountered:

- input saturation constraint:
$$\Omega = \{u \in H : |u(t)| \leq M(t)\}$$
- input amplitude constraint:
$$\Omega = \{u \in H : \lambda(t) \leq u(t) \leq \mu(t)\}$$
- input sign constraint:
$$\Omega = \{u \in H : 0 \leq u(t)\}$$
- input energy constraint:
$$\Omega = \{u \in H : \sum_{t=0}^{N-1} u^2(t) \leq M\}$$

If there are no constraints, the ILC design problem is relatively easy to solve and there are many design methods in the literature. However, when constraints are present, the problem becomes more complicated since it is now necessary to decide how to incorporate them into the design process and retain known performance properties. In what follows, two constrained ILC algorithms recently developed in [8] using a novel successive projection framework are summarized.

Algorithm 1: Given any initial input u_0 satisfying the constraint with associated tracking error e_0, the input sequence $u_{k+1}, k = 0, 1, 2, \cdots$, defined by

$$u_{k+1} = \arg\min_{u\in\Omega} \left\{ \|r - Gu\|_Q^2 + \|u - u_k\|_R^2 \right\}, \tag{9}$$

also satisfies the constraint and iteratively solves the constrained ILC problem.

Constrained Algorithm 1 has the following properties:

Theorem 1: Algorithm 1 converges to point u_s^* which is uniquely defined by the following optimization problem

$$u_s^* = \arg\min_{u\in\Omega} \|r - Gu\|_Q^2. \tag{10}$$

Moreover, this convergence is monotonic in the tracking error, that is,

$$\|e_{k+1}\| \le \|e_k\|, k = 0, 1, \cdots . \tag{11}$$

In the case when perfect tracking is possible, Constrained Algorithm 1 will converge to zero tracking error and has desirable properties of monotonic convergence in tracking error norm. However, it requires the solution of a quadratic programming (QP) problem and can be computationally demanding and in [8] two efficient solution methods were developed but are omitted here for brevity.

Another algorithm that is less computationally demanding is the following.

Algorithm 2: Given any initial input u_0 satisfying the constraint with associated tracking error e_0, the input sequence $u_{k+1}, k = 0, 1, 2, \cdots$, defined by the solution of the input unconstrained NOILC optimization problem

$$\tilde{u}_k = \arg\min_{u} \left\{ \|r - Gu\|_Q^2 + \|u - u_k\|_R^2 \right\}, \tag{12}$$

followed by the simple input projection

$$u_{k+1} = \arg\min_{u\in\Omega} \|u - \tilde{u}_k\| \in \Omega, \tag{13}$$

also satisfies the constraint and iteratively solves the constrained ILC problem.

Remark 1: The first step in Algorithm 2 requires the solution of the input unconstrained NOILC optimization problem (12). Unlike Algorithm 1, which may cause computational problems in solving the large constrained QP problem (9), (12) has a real-time Riccati solution [16]

$$\begin{aligned} u_{k+1}(t) &= u_k(t) - R^{-1}B^T M(t), \\ M(t) &:= K(t)(I + BR^{-1}B^T K(t))^{-1} \times \\ &\quad\; A(x_{k+1}(t) - x_K(t)) - \xi_{k+1}(t), \end{aligned} \tag{14}$$

where $K(t)$ satisfies the Riccati equation

$$\begin{aligned} K(t) = A^T K(t+1)A + C^T QC - A^T K(t+1)B\times \\ (B^T K(t+1)B + R)^{-1} B^T K(t+1)A \end{aligned} \tag{15}$$

with final time condition $K(N) = 0$. Moreover, $\xi_{k+1}(t)$ satisfies the differential equation

$$\begin{aligned} \xi_{k+1}(t) = (I + K(t)BR^{-1}B^T)^{-1}(A^T \xi_{k+1}(t+1) \\ + C^T Q e_k(t+1)), \end{aligned} \tag{16}$$

which is computable in reverse time as it is driven by tracking error from the previous trial k [16].

Remark 2: The second step in Algorithm 2 requires the solution of the problem (13) and would appear to need the application of optimization methods. However, in practice the input constraint Ω is often a point-wise constraint and the solution of (13) can be computed easily. For example, when $\Omega = \{u \in H : |u(t)| \le M(t)\}$, the solution is

$$u_{k+1}(t) = \begin{cases} M(t) & : \tilde{u}_k(t) > M(t) \\ \tilde{u}_k(t) & : |\tilde{u}_k(t)| \le M(t) \\ -M(t) & : \tilde{u}_k(t) < -M(t) \end{cases} \tag{17}$$

for $t = 0, \cdots, N-1$.

Constrained Algorithm 2 requires less computational effort but, unlike Constrained Algorithm 1, it cannot guarantee monotonic convergence of the tracking error norm. Instead, it achieves monotonic convergence of weighted error norm, as shown in the following theorem.

Theorem 2: When perfect tracking is not possible, Algorithm 2 converges to a point u_s^* which is uniquely defined by the following optimization problem,

$$u_s^* = \arg\min_{u\in\Omega} \left\{ \|Ee\|_Q^2 + \|Fe\|_R^2 \right\}. \tag{18}$$

Moreover, this convergence is monotonic with respect to the following performance index

$$J_k = \|Ee_k\|_Q^2 + \|Fe_k\|_R^2, \tag{19}$$

where

$$\begin{aligned} e &= r - Gu \\ E &= I - G\left(G^T QG + R\right)^{-1} G^T Q \\ F &= \left(G^T QG + R\right)^{-1} G^T Q \end{aligned} \; . \tag{20}$$

It was also shown in [8] that when perfect tracking is not possible, the choice of Q and R in Algorithm 2 has an interesting effect on the convergence properties. In particular, there is a compromise between the convergence rate and the tracking performance: using a smaller R will result in faster convergence, however, with a larger final tracking error.

Simulation studies have demonstrated that the two algorithms considered above can solve the constrained ILC problem efficiently, with different convergence properties and computational requirements. The remainder of this paper examines the performance of the algorithms experimentally on a gantry robot that has been used to tests an extensive range of linear model based ILC algorithms [9], [18].

III. GANTRY ROBOT TEST FACILITY

This approach has been experimentally implemented on a 3-axis gantry robot. Figure 1 shows this experimental facility where the robot head performs a 'pick and place' task and is similar to systems which can be found in many industrial applications. These include food canning, bottle filling or automotive assembly, all of which require accurate tracking control, each time the operation is performed, with a minimum level of error in order to maximize production rates. This is an obvious general area for application of ILC.

Each axis of the gantry robot has been modeled based on frequency response tests where, since the axes are orthogonal, it is assumed that there is minimal interaction between them. Here we first consider the X-axis (the one parallel to the conveyor in Figure 1) and frequency response tests (via Bode approximate gain plots in Figure 2) result in a 7th order continuous-time transfer-function as an adequate model of the dynamics on which to base control systems design.

$$G_X(s) = \frac{13077183.4436(s+113.4)}{s(s^2+61.57s+1.125\times 10^4)} \times \cdots \\ \frac{(s^2+30.28s+2.13\times 10^4)}{(s^2+227.9s+5.647\times 10^4)(s^2+466.1s+6.142\times 10^5)}. \tag{21}$$

The dynamics have been sampled at $T_s = 0.01$ seconds to yield the discrete state space model which can be used to compute the model matrix G according to (3). The same procedure has been applied to the Y-axis and Z-axis. For the details see [9].

Fig. 1. The multi-axis gantry robot.

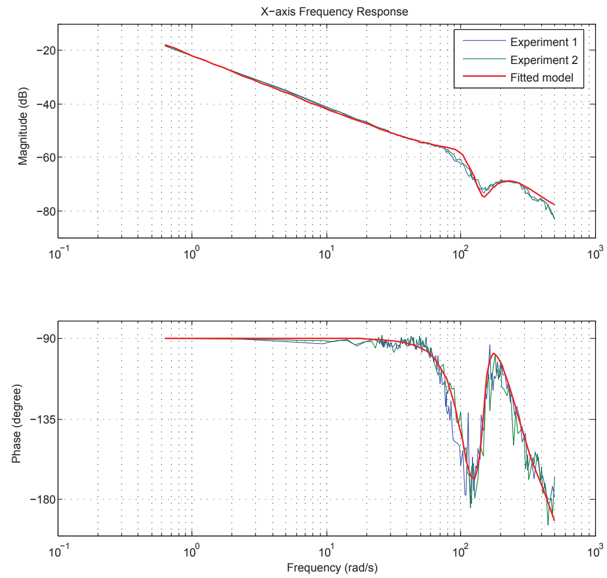

Fig. 2. Frequency response test results and fitted model

$$A_x =$$
$$\begin{bmatrix} 0.035 & 1.000 & 0 & 0 & 0 & 0 & 0 \\ -0.008 & 0.035 & -0.071 & 0.156 & 0 & 0 & 0 \\ 0 & 0 & -0.158 & 1.000 & 0 & 0 & 0 \\ 0 & 0 & -0.078 & -0.158 & 0.569 & 0.698 & 1.319 \\ 0 & 0 & 0 & 0 & 0.388 & 1.000 & 0 \\ 0 & 0 & 0 & 0 & -0.390 & 0.388 & 1.080 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.000 \end{bmatrix},$$

$$B_x = \begin{bmatrix} 0 & 0 & 0 & 0.0164 & 0 & 0.0134 & 0.0197 \end{bmatrix}^T,$$

$$C_x = \begin{bmatrix} -0.0003 & -0.0054 & -0.0145 & 0.0316 & 0 & 0 & 0 \end{bmatrix}.$$

IV. TEST PARAMETERS

The gantry robot is designed to repeatedly complete a pick-and-place motion in synchronization with a moving conveyor. A reference trajectory for the gantry movement has been pre-defined with the purpose of synchronizing its motion with that of the conveyor, which is running at a constant speed. Each axis is controlled individually and has its own reference trajectory and they are combined to form the 3D reference trajectory given in Figure 3 which clearly shows the 'pick and place' action. The signal duration is 2 seconds.

V. EXPERIMENTAL RESULTS

This section gives the results of experimental performance of the two constrained ILC algorithms given in Section 2. Input constraints were imposed such that for each axis, the amplitude of the input voltage was limited to be 90% that of the optimal input u^*, which produces performance tracking. Note that under these constraints perfect tracking is not possible, which is more practically relevant than the trivial case where perfect tracking is achievable).

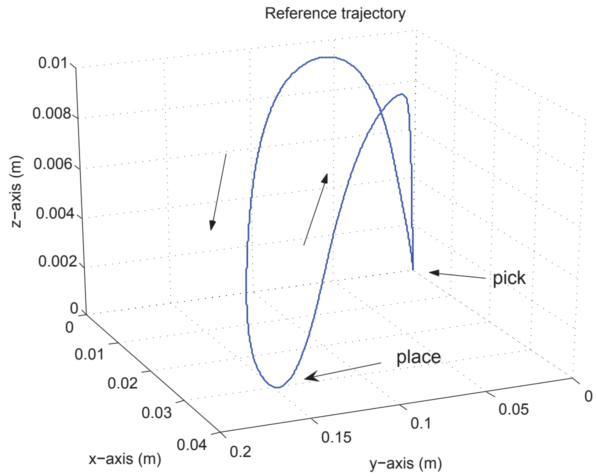

Fig. 3. 3-D combined reference trajectory.

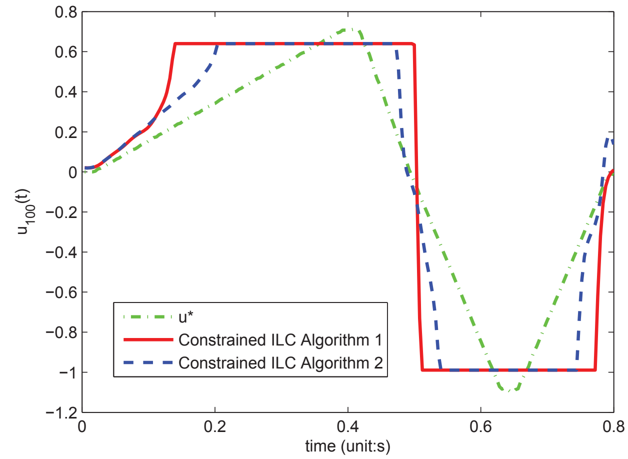

Fig. 5. Input on the 100^{th} trial for the X-axis

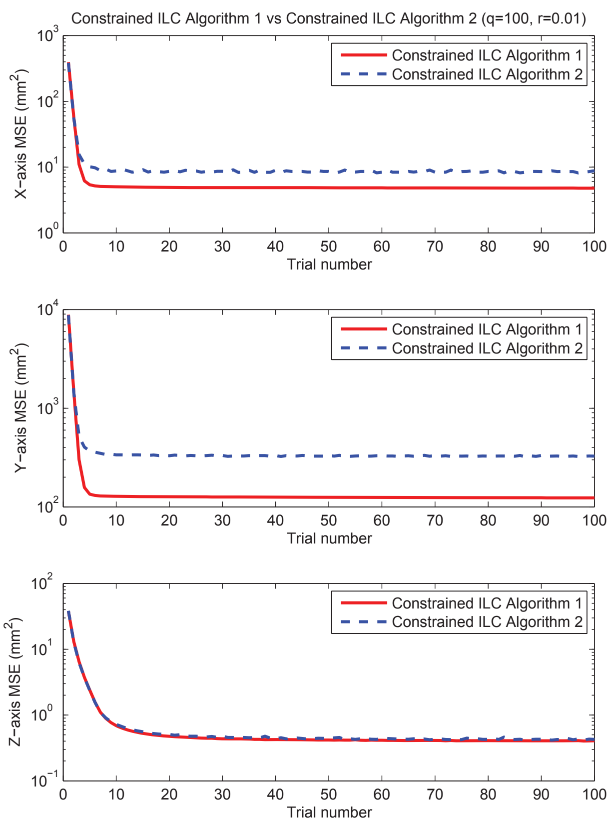

Fig. 4. Comparison of Convergence with $Q = 100I, R = 0.01I$

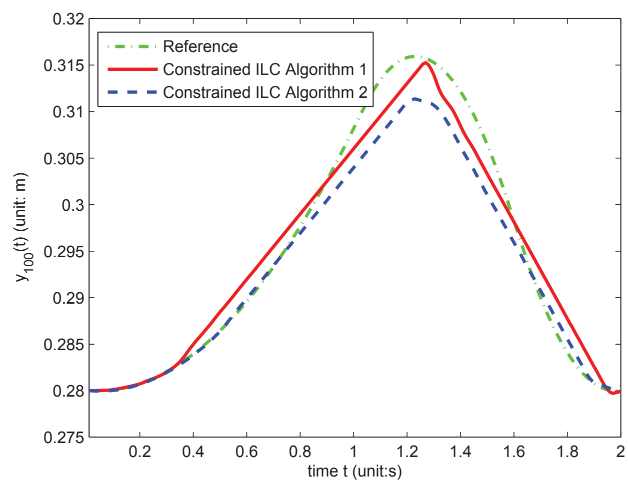

Fig. 6. Output on the 100^{th} trial for the X-axis

The cost function weighting matrices are chosen as diagonal matrices with common diagonal entries of 100 and 0.01 for all three axes and the experimentally results are given in Figure 4. These results confirm that both algorithm solve the constrained ILC problem and converge to some final values. The convergence of Constrained ILC Algorithm 1 is monotonic in the tracking error norm, whereas Constrained ILC Algorithm 2 is not. Moreover, the final tracking error norm of Constrained ILC Algorithm 1 is smaller than that of Algorithm 2, which is consistent with the theoretical predictions.

To expand the discussion of these results , the input, output and tracking error on 100^{th} trial for X-axis are given in Figures 5-7. Using Constrained ILC Algorithm 1, a smaller tracking error (on average) is obtained, compared to Constrained ILC Algorithm 2. Note also that the input computed by Algorithm 2 does not just enforce saturation on the original input but adds some compensation (Figure 5).

Figures 8 shows the effects of varying the selection of the weighting matrices keeping the diagonal structure but changing the control input weighting to 0.1. Compared to the results in Figure 4, this new choice puts larger weighting on the input change, leading to slower convergence. Varying this weighting value has no effect on the final tracking error for Algorithm 1

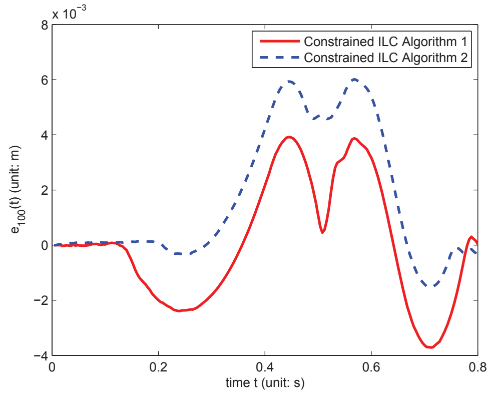

Fig. 7. Tracking error on the 100^{th} trial for the X-axis

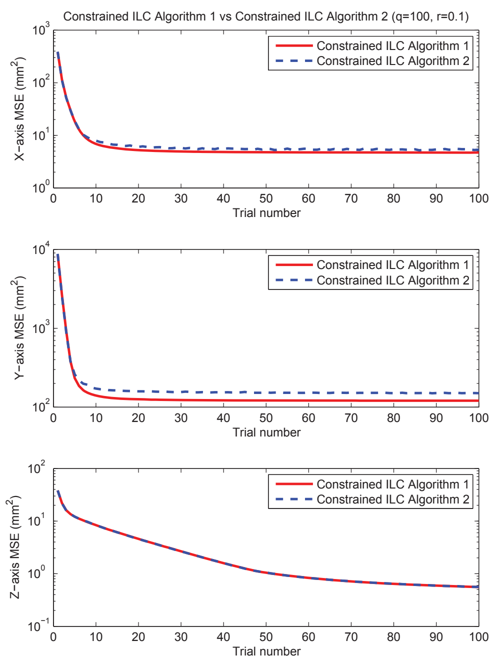

Fig. 8. Comparison of Convergence with control weighting increased to 0.1.

but in Algorithm 2 it results in a better (almost optimal) tracking accuracy, which verifies the theoretical results of Section 2.

Using these results, it can be concluded that Constrained ILC Algorithm 1 performs better than Algorithm 2. However, this is achieved at the expense of higher computational load, which may be not acceptable in some applications but Algorithm 2 achieves nearly optimal performance using a quite simple computation, which is equally (if not more) important in many cases.

VI. Conclusions

In this paper, two constrained ILC algorithms developed based a successive project framework in [19] have been tested on a gantry robot facility. The results confirm that both algorithms can solve the constrained ILC problem efficiently, while the computation of one algorithm is much less than the other at the cost of slightly sacrificed convergence performance. This requires a compromise between the performance/accuracy and the computational cost.

The experimental results in this paper are based on a linear model of the gantry robot, where nonlinearities are neglected and can be treated as model uncertainty. The results clearly demonstrate certain robustness of the algorithms used. However, further theoretical robustness analysis still needs to be done and constitutes part of planned future research.

References

[1] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operations of robots by learning,” *Journal of Robotic Systems*, vol. 1, no. 1, pp. 123–140, 1984.

[2] H. S. Ahn, Y. Chen, and K. L. Moore, “Iterative learning control: brief survey and categorization,” *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 37, no. 6, pp. 1109–1121, 2007.

[3] D. Bristow, M. Tharayil, and A. 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.

[4] C. Chen and S. Peng, “Learning control of process systems with hard input constraints,” *Journal of Process Control*, vol. 9, no. 2, pp. 151–160, 1999.

[5] S. Gunnarsson and M. Norrlof, “On the design of ILC algorithms using optimization,” *Automatica*, vol. 37, no. 12, pp. 2011–2016, 2001.

[6] J. Lee, K. Lee, and W. Kim, “Model-based iterative learning control with a quadratic criterion for time-varying linear systems,” *Automatica*, vol. 36, no. 5, pp. 641–657, 2000.

[7] S. Mishra, U.Topcu, and M. Tomizuka, “Iterative learning control with saturation constraints,” in *Proceedings of American Control Conference*, 2009, pp. 943–948.

[8] B. Chu and D. H. Owens, “Iterative learning control for constrained linear systems,” *International Journal of Control*, vol. 83, no. 7, pp. 1397–1413, 2010.

[9] J. D. Ratcliffe, P. L. Lewin, E. Rogers, J. J. Hatonen, 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, 2006.

[10] J. Hatonen, “Issues of algebra and optimality in iterative learning control,” Ph.D. dissertation, University of Oulu, Finland, 2004.

[11] J. Hatonen, D. Owens, and K. Moore, “An algebraic approach to iterative learning control,” *International Journal of Control*, vol. 77, no. 1, pp. 45–54, 2004.

[12] Z. Bien and K. Huh, “Higher-order iterative learning control algorithm,” *IEE Proceedings, Part D: Control Theory and Applications*, vol. 136, no. 3, pp. 105–112, 1989.

[13] J. Hatonen, D. Owens, and K. Feng, “Basis functions and parameter optimisation in high-order iterative learning control,” *Automatica*, vol. 42, no. 2, pp. 287–294, 2006.

[14] N. Amann, D. 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.

[15] N. Amann, “Optimal algorithms for iterative learning control,” Ph.D. dissertation, University of Exeter, UK, 1996.

[16] N. Amann, D. Owens, and E. Rogers, “Iterative learning control for discrete-time systems with exponential rate of convergence,” *IEE Proceedings: Control Theory and Applications*, vol. 143, no. 2, pp. 217–224, 1996.

[17] ——, “Predictive optimal iterative learning control,” *International Journal of Control*, vol. 69, no. 2, pp. 203–226, 1998.

[18] B. Chu, Z. Cai, D. H. Owens, E. Rogers, C. T. Freeman, and P. L. Lewin, “Experimental verification of accelerated norm-optimal iterative learning control,” in *UKACC International Conference on Control*. Coventry, UK, 2010, pp. 211–216.

[19] B. Chu and D. H. Owens, “Accelerated norm-optimal iterative learning control algorithms using successive projection,” *International Journal of Control*, vol. 82, no. 8, pp. 1469–1484, 2009.