Point-to-Point Iterative Learning Control with Optimal Tracking Time Allocation: A Coordinate Descent Approach

Yiyang Chen[1], Bing Chu[1], Christopher T. Freeman [1]

1. Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom

E-mail: { yc12u12, b.chu, cf } @ecs.soton.ac.uk

Abstract: Iterative learning control (ILC) is a high performance control technique for systems operating in a repetitive manner. A novel design methodology is developed in this paper to incorporate optimal tracking time allocation within the point-to-point ILC framework for discrete time systems. This leads to significant performance improvements compared to fixed time points (e.g. energy reduction). An optimization problem is formulated based on the point-to-point tracking requirement and the via-point temporal constraints. A two stage design framework is proposed to solve this problem, yielding an algorithm based on norm optimal ILC and the coordinate descent method, which automatically minimizes control effort while maintaining high performance tracking. The proposed algorithm is implemented on a gantry robot experimental test platform, with results verifying its practical effectiveness in the presence of model uncertainty.

Key Words: iterative learning control, optimization, coordinate descent method.

1 Introduction

ILC is a control technique applied to high performance systems which perform the same tracking task over a finite time horizon repetitively. The standard ILC framework can be considered as an open-loop feedforward methodology, which updates the input signal to improve the tracking performance by learning from the data of the previous executions (named trials). Unlike repetitive control, ILC has an initialization procedure at the end of each trial to reset the system states to the same initial values. ILC theoretically enables zero tracking error after sufficient historical trials, which is superior to traditional feedback control. Because of this key feature, ILC is applied to wide range of industrial tasks which require a high accuracy level, e.g. robotic systems [1], [2], chemical batch processing [3], [4] and stoke rehabilitation [5]. See [6] for a detailed overview of ILC.

In certain practical applications, the system output is only critical at a subset of time instants over the whole time horizon, and it is only required to attain a reference position at these isolated time instants. An example is the robotic manipulator's pick-and-place task, as it only concerns the pick and place positions. A new control framework termed point-to-point ILC is thereby formulated in [7] for these control problems, which only enforces output tracking at these special time instants. It removes the unnecessary output constraints, and releases extra freedom in control design to address additional performance objectives. Recent research has been made on optimal problem [8], [9], frequency analysis [7] and constrained conditions [10], [11].

In point-to-point tracking problems, the tracking times of the critical points are typically embedded within some cost functions (e.g. control effort), so these cost functions are highly dependent on the tracking time allocation. Therefore, the reallocation of the tracking times can optimize these cost functions, and translate into significant practical benefits, such as reducing the waste energy in industry, reducing the damage to machine components and increasing the efficiency of production (i.e. throughput). This hence motivates that the point-to-point ILC framework be expanded to allow flexibility in tracking time allocation to embed further optimization of the desired cost function. However, an assumption is made in all previous point-to-point ILC problem formulations that the tracking time allocation is known *a priori*, which limits the potential optimization of these cost functions within the previous framework. Recent research has considered the optimal tracking time allocation of point-to-point robotic motion [12]. However, its optimal solution is computed using a nominal model, therefore it is not robust to model uncertainty and cannot handle high performance tracking tasks in practice.

This paper addresses the full optimal tracking time allocation problem for discrete time systems in which dynamic interaction occurs between the critical points. Note that preliminary results presented in [13] employed the gradient method to address continuous time systems, but these results cannot be applied to discrete time systems and are therefore unsuitable for practical implementation. The main contributions of the paper are as follows:

- The optimal tracking time allocation problem is formulated in Section 2 including both optimization of a cost function as well as point-to-point reference tracking.
- A two stage design framework is proposed in Section 3 to solve the problem, and provides solutions to the minimum control effort problem within the point-to-point ILC framework based on norm optimal point-to-point ILC and the coordinate descent method.
- A practical implementation algorithm is then derived from the solutions of the problem in Section 4 and the practical effectiveness of the algorithm is verified on a gantry robot platform in Section 5.

2 Formulation of the Problem

This section first introduces the system dynamics and defines the point-to-point ILC framework. Then the design problem of point-to-point ILC with optimal tracking time allocation is formulated into an optimization problem.

This work is supported by China Scholarship Council.

2.1 System Dynamics

Consider an ℓ-input, m-output discrete linear time-invariant system given in state space form by $S(A, B, C)$ with unit sample time

$$\begin{aligned} x_k(t+1) &= Ax_k(t) + Bu_k(t),\ x(0) = x_0 \\ y_k(t) &= Cx_k(t),\ 0 \leqslant t \leqslant N,\ N < \infty \end{aligned} \tag{1}$$

where t is the time index (e.g. sample number), $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^\ell$ and $y(t) \in \mathbb{R}^m$ are the state, input and output respectively; A, B and C are system matrices of compatible dimension; $0 < N < \infty$ is the trial length, the subscript $k \in \mathbb{N}$ denotes the ILC trial number. At the end of each trial, the state is reset to initial value x_0. The system can be represented in an equivalent operator form

$$\begin{aligned} & y_k = Gu_k + d,\ G : l_2^\ell[0,\ N] \to l_2^m[0,\ N] \\ & y_k,\ d_k \in l_2^m[0,\ N],\ u \in l_2^\ell[0,\ N] \end{aligned} \tag{2}$$

where the input and output Hilbert spaces $l_2^\ell[0,\ T]$ and $l_2^m[0,\ T]$ are defined with inner products and associated induced norms

$$\langle u,\ v\rangle_R = \sum_{i=0}^{N} u^T(i)Rv(i),\ \ \|u\|_R^2 = \langle u,\ u\rangle_R \tag{3}$$

$$\langle x,\ y\rangle_S = \sum_{i=0}^{N} x^T(i)Sy(i),\ \ \|y\|_S^2 = \langle y,\ y\rangle_S \tag{4}$$

in which $R \in \mathbb{S}_{++}^\ell$ and $S \in \mathbb{S}_{++}^m$ ($\mathbb{S}_{++}^{\cdot}$ denotes the set of all real positive definite matrices with appropriate dimensions). The convolution operator G and signal d (representing the effect of initial condition) take the form

$$(Gu)(t) = \sum_{i=0}^{t-1} CA^{t-i-1}Bu(i),\ d(t) = CA^t x_0 \tag{5}$$

where, without loss of generality, the constant $d(t)$ can be absorbed into the reference to give $x_0 = 0$, $d(t) = 0$.

2.2 Point-to-Point ILC Framework

The point-to-point ILC design objective is to update the input signal, u_k, such that the input signal, u_k, converges to a unique value and the associated output, y_k, ultimately tracks the given reference positions, r_i, $i = 1, \dots, M$, at a sub set of time instants, t_i, $i = 1, \dots, M$, i.e.

$$\lim_{k\to\infty} y(t_i) = r_i,\ i = 1, \dots, M,\ \lim_{k\to\infty} u_k = u^*. \tag{6}$$

From the design objective (6), only the particular output at the tracking time allocation

$$\Lambda = [t_1,\ t_2, \dots,\ t_M]^\top \in \Theta \tag{7}$$

is of interest where

$$\Theta = \{\Lambda \in \mathbb{R}^M : 0 < t_1 < t_2 < \ldots < t_M \leqslant N\} \tag{8}$$

is the admissible set of allocated tracking time instants representing the requirements on enforcing process timing constraints necessary to complete the task. Hence a linear mapping $\beta \in l_2^m[0,\ N] \mapsto \beta^p \in H$ defined by

$$\beta^p = \begin{bmatrix} \beta(t_1) \\ \vdots \\ \beta(t_M) \end{bmatrix} \tag{9}$$

is introduced to extract the output values at the tracking time allocation. Note that H is the Hilbert space denoted by

$$H = \underbrace{\mathbb{R}^m \times \cdots \times \mathbb{R}^m}_{\text{M times}} \tag{10}$$

with inner product and associated induced norm

$$\langle \omega,\ \mu\rangle_{[Q]} = \sum_{i=1}^{M} \omega_i^\top Q_i \mu_i,\ \ \|\omega\|_{[Q]}^2 = \langle \omega,\ \omega\rangle_{[Q]} \tag{11}$$

where

$$\omega = [\omega_1, \dots, \omega_M]^\top \in H,\ \mu = [\mu_1, \dots, \mu_M]^\top \in H;$$

$[Q]$ denotes the set $\{Q_1, \dots, Q_M\}$, and $Q_i \in \mathbb{S}_{++}^m$.

From definition (9), it follows that the 'point-to-point output', y^p, comprises a subset of plant outputs defined over the tracking time allocation Λ. The dynamics of the point-to-point system can therefore be modelled by

$$y^p = G_\Lambda^p u = (Gu)^p = \begin{bmatrix} (Gu)(t_1) \\ \vdots \\ (Gu)(t_M) \end{bmatrix} \tag{12}$$

where $G_\Lambda^p : l_2^\ell[0, N] \to H$ is a linear operator.

Therefore, the point-to-point ILC design objective design objective (6) can be equivalently described as iteratively finding a sequence of input $\{u_k\}$ such that

$$\lim_{k\to\infty} y_k^p = r^p,\ \lim_{k\to\infty} u_k = u^* \tag{13}$$

where

$$r^p = [r_1,\ r_2, \dots,\ r_M]^\top \in H \tag{14}$$

To solve the problem, the point-to-point tracking error $e_k^p = r^p - y_k^p$ is employed within the following updating law:

$$u_{k+1} = \mathcal{F}(u_k, e_k^p) \tag{15}$$

where $\mathcal{F}$ is an updating function involving the previous input and point-to-point tracking error.

2.3 Optimal Tracking Time Allocation Problem

In all existing point-to-point ILC frameworks, the tracking time allocation Λ is assumed to be known as *a priori*, and the extra flexibility in tracking time allocation has not been fully explored. This paper aims to address this **Point-to-Point ILC with Optimal Tracking Time Allocation Problem** by proposing a design framework which automatically provides a tracking time allocation to optimize some desired cost functions, and meanwhile ensures high performance tracking at the tracking time allocation.

The problem design objective is to iteratively find a tracking time allocation Λ_k and an input, u_k, with the asymptotic property that the output values, y_k^p, at the tracking time allocation accurately pass through a set of points, r^p, i.e.

$$\lim_{k\to\infty} y_k^p = r^p,$$

at the same time optimizing a target cost function $f(u, y)$ with respect to the system input, u, and output, y, i.e.

$$\lim_{k\to\infty} (u_k,\ y_k,\ \Lambda_k) = (u_k^*,\ y_k^*,\ \Lambda_k^*)$$

where u_k^*, y_k^* and Λ_k^* are optimal solutions of the problem

$$\begin{aligned} &\underset{u,y,\Lambda}{\text{minimize}} && f(u,y) \\ &\text{subject to} && r^p = G_\Lambda^p u,\ y = Gu,\ \Lambda \in \Theta. \end{aligned} \tag{16}$$

3 A Two Stage Design Framework

While the tracking time allocation Λ does not appear in the performance function $f(u,y)$, they are connected by the constraint $G_\Lambda^p u = r^p$, making the problem (16) non-trivial. In this section, a two stage design framework is formulated to solve this optimization problem.

3.1 Framework Description

Optimization problem (16) can be equivalently written as

$$\min_{\Lambda\in\Theta}\{\min_u f(u,y), \text{subject to } r^p = G_\Lambda^p u,\ y = Gu\} \tag{17}$$

which optimizes over u first and then optimizes over Λ. Define the function $\tilde{f}(\Lambda) : \mathbb{R}^M \to \mathbb{R}$ by

$$\tilde{f}(\Lambda) = \{\min_u f(u,y), \text{subject to } r^p = G_\Lambda^p u,\ y = Gu\} \tag{18}$$

and denote a global minimizer for u of the inner optimization problem as $u_\infty(\Lambda) : \mathbb{R}^M \to l_2^\ell[0,\ N]$. Hence the problem (17) can be equivalently written as

$$\min_{\Lambda\in\Theta}\{\tilde{f}(\Lambda) := f(u_\infty(\Lambda))\}. \tag{19}$$

The above suggests that the optimization problem (16) can be solved by applying a two stage design framework as:

- *Stage One*: Keep the tracking time allocation Λ fixed and solve the optimization problem

$$\min_u f(u,y),\ \text{subject to } r^p = G_\Lambda^p u,\ y = Gu. \tag{20}$$

- *Stage Two*: Substitute the solution $u_\infty(\Lambda)$ into the original problem and then find the optimal solution

$$\min_{\Lambda\in\Theta}\{\tilde{f}(\Lambda) := f(u_\infty(\Lambda))\}. \tag{21}$$

In this paper, the control effort is selected to be the target cost function to exemplify the approach, i.e. $f(u,y) = \|u\|_R^2$. This guarantees the existence of a unique global minimizer for u within (20).

3.2 Solution of the Proposed Framework

1). Solution of Stage One: For a given Λ, Stage One is a point-to-point ILC design problem with minimum control effort requirement. This can be solved using a norm optimal point-to-point ILC algorithm as shown in the next theorem.

Theorem 1. If the system $S(A,B,C)$ is controllable and C has full row rank, the solution of Stage One problem (20) is given by u_∞, which can be found using the norm-optimal point-to-point ILC algorithm

$$u_{k+1} = u_k + G_\Lambda^{p*}(I + G_\Lambda^p G_\Lambda^{p*})^{-1} e_k^p \tag{22}$$

proposed in [8] with initial input $u_0 = 0$ such that

$$u_\infty(\Lambda) = \lim_{k\to\infty} u_k. \tag{23}$$

Furthermore, an analytic solution can be obtained for $u_\infty(\Lambda)$ as follows:

$$u_\infty(\Lambda) = G_\Lambda^{p*}(G_\Lambda^p G_\Lambda^{p*})^{-1} r^p. \tag{24}$$

Note that $G_\Lambda^{p*} : \omega \in H \to u \in l_2^\ell[0,\ N]$ is the Hilbert adjoint operator of G_Λ^p given by

$$\begin{aligned} &u(t) = R^{-1}B^\top p(t),\ p(N) = 0,\ \tilde{t}_i = t_i - 1, \\ &p(t) = A^\top p(t+1),\ t \in [t_{i-1},\ \tilde{t}_i],\ i = 1,\ldots,M, \\ &p(\tilde{t}_i^-) = p(\tilde{t}_i^+) + C^\top Q_i \omega_i,\ i = 1,\ldots,M. \end{aligned} \tag{25}$$

Proof. It follows from [8] that the final converged input u_∞ of ILC update (24) provides the solution of the problem (20). The analytical solution of the problem (20) is obtained from the associated Lagrangian with Lagrange multiplier $\lambda \in H$

$$\mathcal{L}(u) = \|u\|_R^2 + 2 < \lambda,\ G_\Lambda^p u - r^p >_{[Q]} \tag{26}$$

which has a unique stationary point $u_\infty = G_\Lambda^{p*}\lambda$ and $r^p = G_\Lambda^p G_\Lambda^{p*}\lambda$. As the system $S(A,B,C)$ is controllable and C has full row rank, the matrix $G_\Lambda^p G_\Lambda^{p*}$ is invertible and the analytical solution is given by (24). The relevant adjoint operator G_Λ^{p*} is obtained from the Hilbert inner product form

$$< \omega,\ G_\Lambda^p u >_{[Q]} = < (G_\Lambda^{p*}\omega,\ u >_R \tag{27}$$

which gives rise to (25). □

Remark 1. Note that the system's state controllable condition is not restrictive as a state controllable model can always be constructed for a given system.

2). Solution of Stage Two: Using analytical solution (24), optimization problem (21) can be further reduced as shown in the next lemma.

Lemma 1. Based on the analytical solution (24) of Stage One optimization problem, the Stage Two optimization problem (21) can be expressed as

$$\min_{\Lambda\in\Theta}\|u_\infty(\Lambda)\|_R^2 = \min_{\Lambda\in\Theta}\left\langle r^p,\ (G_\Lambda^p G_\Lambda^{p*})^{-1} r^p\right\rangle_{[Q]}. \tag{28}$$

Proof. Substituting the analytical solution (24) into the optimization problem (21) and using the property of adjoint operator gives

$$\begin{aligned} \min_{\Lambda\in\Theta}\|u_\infty(\Lambda)\|_R^2 &= \min_{\Lambda\in\Theta}\ \langle(u_\infty(\Lambda),\ u_\infty(\Lambda)\rangle_R \\ &= \min_{\Lambda\in\Theta}\ \left\langle(G_\Lambda^{p*}(G_\Lambda^p G_\Lambda^{p*})^{-1}r^p, G_\Lambda^{p*}(G_\Lambda^p G_\Lambda^{p*})^{-1}r^p\right\rangle_R \\ &= \min_{\Lambda\in\Theta}\ \left\langle G_\Lambda^p G_\Lambda^{p*}(G_\Lambda^p G_\Lambda^{p*})^{-1}r^p, (G_\Lambda^p G_\Lambda^{p*})^{-1}r^p\right\rangle_{[Q]} \\ &= \min_{\Lambda\in\Theta}\ \left\langle r^p,\ (G_\Lambda^p G_\Lambda^{p*})^{-1}r^p\right\rangle_{[Q]} \end{aligned}$$

which completes the proof. □

This optimization problem, however, is non-trivial except for the special case of $M = 1$, i.e. there is only one tracking point, where the solution can be obtained analytically, as shown in the following theorem.

Theorem 2. When there is only one tracking point, optimization problem (28) has analytical solution

$$\Lambda^* = N.$$

The corresponding minimum energy is

$$\min_{\Lambda\in\Theta} \|u_\infty(\Lambda)\|_R^2 = \left\langle r^p,\ \Psi_N^{-1} r^p \right\rangle_{[Q]}$$

where

$$\Psi_t = \sum_{i=1}^{t} CA^{t-i}BR^{-1}(CA^{t-i}B)^\top.$$

Proof. In this case, the optimization problem (28) becomes

$$\min_t \left\langle r^p,\ \Psi_t^{-1} r^p \right\rangle_{[Q]},\ \text{subject to } t\in[1,\ N]. \quad (29)$$

It is clear that Ψ_t is a positive operator such that

$$\Psi_t \leqslant \Psi_N,\ \forall t\leqslant N \quad (30)$$

which hence yields

$$\left\langle r^p,\ \Psi_N^{-1} r^p \right\rangle_{[Q]} \leqslant \left\langle r^p,\ \Psi_t^{-1} r^p \right\rangle_{[Q]},\ \forall t\leqslant N. \quad (31)$$

It follows that $t = N$ is the global optimizer of the problem (29), which completes the proof. □

Theorem 2 shows that when $M = 1$, $\Lambda^* = N$ is always the optimal choice in terms of minimizing the control input energy - this is not surprising as this allows the system output to change gradually to the desired position. However, when $M > 1$, i.e. there is more than one tracking point, the cost function is generally non-linear and non-convex with respect to the time point set Λ. These aspects lead to the difficulties in obtaining an analytical solution of the problem (28) at general cases.

In the case of discrete time systems, the set Θ has a finite number of elements and this permits the use of a blind search over the whole set. However, this carries a high computational load especially when the number of tracking points M becomes large. Therefore, an efficient alternative method is proposed in the next theorem to give an approximate solution of the Stage Two problem (28).

Theorem 3. Consider the coordinate descent method with initial estimate Λ_0

$$\Lambda_{j+1} = \mathcal{C}(\Lambda_j) \quad (32)$$

with $\Lambda_j = [t_1^j, t_2^j, \dots,\ t_M^j]^\top$, $j\in\mathbb{N}$ denotes the coordinate descent trial number and each time point is updated by the function $\mathcal{C}$ as

$$t_{i,j+1} = \begin{cases} t_i^{j*},\ i = (j+1) \bmod M \\ t_i^j,\ \text{else} \end{cases} \quad (33)$$

where t_i^{j*} is the optimizer of the optimization problem

$$\begin{aligned} \underset{t}{\text{minimize}} \quad & \left\langle r^p,\ (G_\Lambda^p G_\Lambda^{p*})^{-1} r^p \right\rangle_{[Q]} \\ \text{subject to} \quad & \Lambda = [t_1^j, \dots,\ t_{i-1}^j,\ t,\ t_{i+1}^j, \dots,\ t_M^j]^\top, \\ & t\in(t_{i-1}^j,\ t_{i+1}^j). \end{aligned} \quad (34)$$

The sequence $\{\tilde f(\Lambda_j)\}$ based on the coordinate descent update (32) converges downward to a limit $\tilde f^*$.

Proof. The coordinate descent method divides the optimization problem (28) into a number of intermediate trials. At each trial, it performs a local optimization (34) to update a single time point $t_{i,j}$ with other points keeping the same values. Therefore, it follows that the sequence $\{\tilde f(\Lambda_j)\}$ monotonically decreases as

$$\tilde f(\Lambda_{j+1}) \leqslant \tilde f(\Lambda_j). \quad (35)$$

In addition, as the function $\tilde f(\Lambda)$ is bounded below, the sequence $\{\tilde f(\Lambda_j)\}$ converges to a non-negative value $\tilde f^*$. □

Remark 2. Although blind search is still used in each coordinate descent trial to update a single time point t_i^j, the coordinate descent method requires much less computation time than the pure blind search method over the whole set Θ to provide an optimal solution to the same optimization problem. This is because of the desirable property of the coordinate descent method which splits the original problem into several sub-problems.

Remark 3. In most optimization problems the coordinate descent method does not guarantee a global optimal solution, but yields a suitable local optimal solution which approximates the global optimal solution [14].

Remark 4. Gradient method can be applied to problem (28) in the case of continuous time systems [13]. However, the gradient is unavailable in discrete time systems, so this approach is infeasible.

Remark 5. The coordinate descent method described in (32) updates a single time instant at each trial, however, multiply number of time instants can be also considered for updating.

4 Implementation of the Design Approach

This section develops an algorithm to efficiently implement the two stage framework.

4.1 Implementation of Stage One

The general solution of Stage One using (22) can be either computed directly using the analytical solution (24) or implemented experimentally using a combined feedback and feedforward solution illustrated in the next proposition.

Proposition 1. The ILC update (22) can be implemented using the feedforward plus feedback implementation

$$u_{k+1}(t) = u_k(t) + R^{-1}B^\top p_k(t),\ t = 0,\dots,N \quad (36)$$

with

$$\begin{aligned} p_k(t) = -K(t)(I + BR^{-1}B^\top K(t))^{-1}A \\ (x_{k+1}(t) - x_k(t)) + \xi_{k+1}(t) \end{aligned} \quad (37)$$

where $K(t)$ denotes the Riccati feedback matrix

$$\begin{aligned} & K(t) = A^\top K(t+1)(I + BR^{-1}B^\top K(t+1))^{-1}A, \\ & K(N) = 0,\ K(\tilde t_i^-) = K(\tilde t_i^+) + C^\top Q_i C \end{aligned} \quad (38)$$

and $\xi_{k+1}(t)$ denotes the predictive feedforward term at the $(k+1)^{th}$ ILC trial

$$\begin{aligned} & \xi_{k+1}(t) = (I + K(t)BR^{-1}B^\top)^{-1}A^\top \xi_{k+1}(t+1), \\ & \xi_{k+1}(N) = 0,\ \xi_{k+1}(\tilde t_i^-) = \xi_{k+1}(\tilde t_i^+) + C^\top Q_i e_k(t_i). \end{aligned} \quad (39)$$

Proof. The ILC update (22) can be equivalently written into

$$u_{k+1}(t) = u_k(t) + (G_\Lambda^{p*} e_{k+1}^p)(t). \tag{40}$$

Note that $(G_\Lambda^{p*} e_{k+1}^p)(t)$ can be computed according to the costate equation (25), and hence (40) give rise to (36).

When assuming full state knowledge, a causal implementation is derived by transforming the costate equation (25) into (37). Then use the method proposed in [15] to derive the discrete Matrix Riccati equation $K(t)$ and the optimal predictor $\xi_{k+1}(t)$ as shown in (38) and (39). Substitute

$$e_{k+1}(\tilde{t}_i) = e_k(\tilde{t}_i) - C(x_{k+1}(\tilde{t}_i) - x_k(\tilde{t}_i)) \tag{41}$$

into the jump condition at $\tilde{t}_i$ in costate equation (25), and it follows that both $K(t)$ and $\xi_{k+1}(t)$ have the jump conditions at $\tilde{t}_i$ for $i = 1, \dots, M$ shown in (38) and (39). $\square$

The experimental implementation of norm-optimal ILC using feedback and feedforward solution provides an optimal solution to Stage One problem (20) based on real plant dynamics. Due to the real time state feedback at the current ILC trial, this implementation method has a certain degree of robustness against model uncertainties.

4.2 Implementation of Stage Two

The coordinate descent method introduced in Theorem 3 re-arranges the tracking time allocation to minimize the control effort. It starts from an initial tracking time allocation

$$\Lambda_0 = [t_1^0, t_2^0, \dots, t_M^0]^T \in \Theta. \tag{42}$$

At each coordinate descent trial, it only updates a single time point t_i^j by solving the optimization problem (34). This is equivalent to finding the optimal element along the finite interval (t_{i-1}^j, t_{i+1}^j) with respect to a cost function. Therefore, blind search methods can be applied to this problem with a total computation number $\eta_i = (t_{i+1}^j - t_{i-1}^j - 1)$.

Remark 6. As the solution of the coordinate descent method is a local solution, it is necessary to choose a suitable initial tracking time allocation Λ_0 to give a solution which approximates the global one, especially when the system is complex. If no information is available, Λ_0 can be chosen arbitrarily, and alternatively it can be selected using different methods such as performing grid search with a large sample time.

4.3 An iterative implementation algorithm

Combining the implementation of Stage One and Stage Two designs leads to an iterative implementation of the two stage design framework - Algorithm 1. Note that Λ_0 is a suitably chosen initial tracking time allocation, and $\epsilon > 0$, $\delta > 0$ are small scalars which depend on the tracking precision requirement and performance requirement, respectively.

In Algorithm 1, we require that Step 2 and 6 (i.e. norm-optimal ILC algorithm) is implemented experimentally and Step 4 uses real data $u_\infty(\Lambda_j)$ obtained from experiments. These requirements are not necessary when an accurate system model is known. However when there exists model mismatches/uncertainties, the proposed algorithm will have attractive robustness properties as the algorithm 'learns' information about the real plant dynamics through use of experimental data. This will be demonstrated using experimental results in the next section.

Algorithm 1 Two Stage Design Framework Implementation

```
Import: $\Lambda_0$, $S(A, B, C)$, $r^p$, $\Theta$
1: initialization: Coordinate descent trial number $j = 0$
2: Implement Stage One update (22) with $\Lambda = \Lambda_0$ experimentally
   until convergence, i.e. $\|e_k^p\| < \epsilon\|r^p\|$; record converged input
   $u_\infty^{ex}(\Lambda_0)$ and input energy $\tilde{f}(\Lambda_0)$.
3: repeat
4:   Implement Stage Two update (32) with $r^p = G_{\Lambda_j}^p u_\infty^{ex}(\Lambda_j)$.
5:   Set $j \to j + 1$.
6:   Implement Stage One update (22) with $\Lambda = \Lambda_j$ experimen-
     tally until convergence, i.e. $\|e_k^p\| < \epsilon\|r^p\|$; record con-
     verged input $u_\infty^{ex}(\Lambda_j)$ and input energy $\tilde{f}(\Lambda_j)$.
7: until $\left|\tilde{f}(\Lambda_j) - \tilde{f}(\Lambda_{j-1})\right| < \delta \left|\tilde{f}(\Lambda_{j-1})\right|$
8: return $\Lambda_j$ and $u_\infty^{ex}(\Lambda_j)$
```

5 Experimental Verification on a Gantry Robot

The proposed design framework is now validated experimentally on a three-axis gantry robot test facility to demonstrate its effectiveness on a widely used industrial platform.

5.1 Test Platform Specification

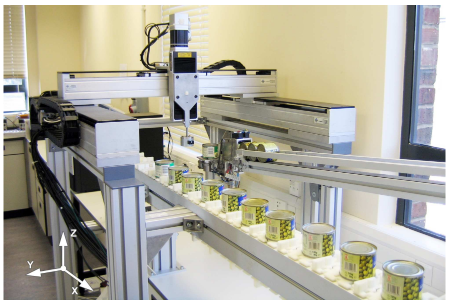

Fig. 1: Multi-axis Gantry Robot Test Platform.

The multi-axis gantry robot shown in Figure 1 is employed as test platform. The control design objective is to use the z-axis ($m = 1$) to perform a point-to-point ILC tracking task during the given tracking time $T = 2s$ with only five special tracking points ($M = 5$) given as

$$r^p = [0.0048,\ 0.0029,\ -0.0029,\ -0.0048,\ 0]^\top \tag{43}$$

which are shown in Figure 3. The *a priori* tracking time allocation is given as $\Lambda_r = [20, 60, 100, 140, 180]^\top$.

5.2 Experimental Results using an Inaccurate Model

In industrial environments obtaining an accurate model is generally infeasible and we hence assume that only an approximate model is available for the z-axis as follows:

$$\hat{G}_z(s) = \frac{0.03}{s} \tag{44}$$

with a 150 proportional feedback controller which is sampled with a zero-order hold at $0.01s$.

A 30 coordinate descent trial updating procedure of Algorithm 1 is performed on the gantry robot platform with initial tracking time allocation $\Lambda_0 = \Lambda_r$. For implementational simplicity, the weighting matrices in Step 2 and 6 are taken as $Q_i = qI$ and $R = rI$ where q and r are positive scalars which satisfy $q/r = 1,000,000$. From the experiment, an experimental optimal tracking time allocation $\Lambda_{30} = [54, 59, 128, 137, 200]^\top$ is obtained.

The input energy $\tilde{f}(\Lambda_j)$ at each trial is plotted in Figure 2, which shows that the input energy converges to a limit energy $\tilde{f}(\Lambda_{30}) = 1053.4$ along the trial. From the figure, the limit input energy $\tilde{f}(\Lambda_{30})$ is 27.4% less than the input energy $\tilde{f}(\Lambda_r)$ at Λ_r, which confirms the robustness of the algorithm against model uncertainties. The theoretical optimal tracking time allocation $\Lambda^* = [48, 58, 130, 140, 200]^\top$ is computed in simulation using the system model, and the corresponding operation energy $\tilde{f}(\Lambda^*) = 1102.1$ at Λ^* is also plotted in Figure 2 as the dashed magenta line. Note that the experiential energy solution $\tilde{f}(\Lambda_{30})$ is 4.4% less than the theoretical one $\tilde{f}(\Lambda^*)$, which demonstrates the advantage of implementing the algorithm experimentally rather than using the nominal model in simulation.

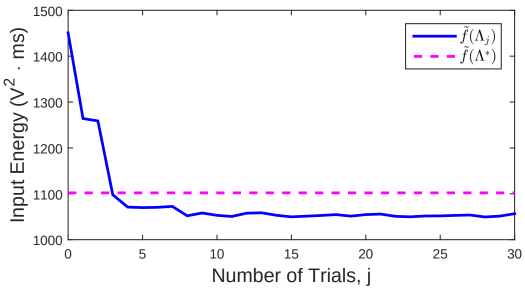

Fig. 2: Experimental Input Energy Results at Each Trial.

The experimental final converged results for the initial and final trials are compared in Figure 3 with the reference r^p marked as red and green dots. It is obvious that the final converged output perform perfect point-to-point tracking at the critical time points, e.g. the mean square error is 0.00053 at the final trial. So the algorithm not only optimizes the input energy but also maintains high tracking performance.

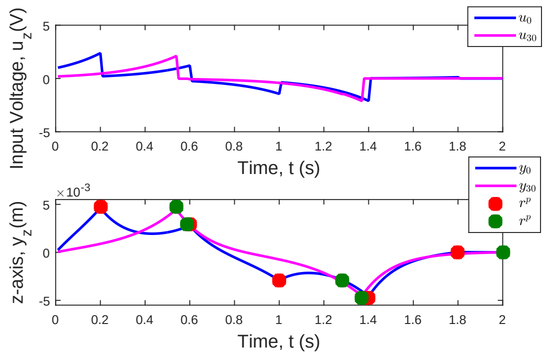

Fig. 3: Experimental Converged Input and Output Trajectories for Initial and Final Trials.

Experiments with other initial tracking time allocations provide similar convergence performance to the results in Figure 2. For brevity, these results are omitted.

6 Conclusion

This paper exploits the flexibility of choosing the tracking time allocation in the point-to-point ILC framework, which affects system performance. An optimization problem is formulated, and a two stage design framework is used to solve this problem with minimum control effort. Distinct to previous work, this framework is suitable for discrete time systems. Stage One solution is obtained via a norm optimal point-to-point ILC algorithm, and Stage Two solution is computed using a coordinate descent method. The solutions yield an iterative algorithm, which is verified on a gantry robot test platform, whose results reveals practical efficacy.

The experimental results demonstrate the effectiveness of the algorithm in practice, but a rigorous robustness analysis on the algorithm is needed. In addition, system constraints can be incorporated into the proposed design framework. Furthermore, in principle the proposed design framework can be extended to optimize other cost functions besides control effort. All these constitute part of our future research and will be reported separately.

References

[1] L. Hladowski, K. Galkowski, Z. Cai, E. Rogers, C. T. Freeman, and P. L. Lewin, "Experimentally supported 2D systems based iterative learning control law design for error convergence and performance," *Control Engineering Practice*, vol. 18, pp. 339–348, 2010.

[2] M. Norrlof, "An adaptive iterative learning control algorithm with experiments on an industrial robot," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 245–251, 2002.

[3] X. Yu, Z. Xiong, D. Huang, and Y. Jiang, "Model-based iterative learning control for batch processes using generalized hinging hyperplanes," *Industrial & Engineering Chemistry Research*, vol. 52, no. 4, pp. 1627–1634, 2013.

[4] J. H. Lee and K. S. Lee, "Iterative learning control applied to batch processes: An overview," *Control Engineering Practice*, vol. 15, pp. 1306–1318, 2007.

[5] C. T. Freeman, *Control System Design for Electrical Stimulation in Upper Limb Rehabilitation*. Springer International Publishing, 2016.

[6] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *Control Systems, IEEE*, vol. 26, no. 3, pp. 96–144, 2006.

[7] C. T. Freeman, Z. Cai, E. Rogers, and P. L. Lewin, "Iterative learning control for multiple point-to-point tracking application," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 590–600, 2011.

[8] D. H. Owens, C. T. Freeman, and T. V. Dinh, "Norm-optimal iterative learning control with intermediate point weighting: theory, algorithms, and experimental evaluation," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 999–1007, 2013.

[9] T. D. Sona, H.-S. Ahn, and K. L. Moore, "Iterative learning control in optimal tracking problems with specified data points," *Automatica*, vol. 49, no. 5, pp. 1465–1472, 2013.

[10] C. T. Freeman and Y. Tan, "Iterative learning control with mixed constraints for point-to-point tracking," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 604–616, 2013.

[11] C. T. Freeman, "Constrained point-to-point iterative learning control with experimental verification," *Control Engineering Practice*, vol. 20, no. 5, pp. 489–498, 2012.

[12] P. Janssens, G. Pipeleers, M. Diehl, and J. Swevers, "Energy optimal time allocation of a series of point-to-point motions," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2432–2435, 2014.

[13] Y. Chen, B. Chu, and C. T. Freeman, "Point-to-point iterative learning control with optimal tracking time allocation," in *54th IEEE Conference on Decision and Control*, Osaka, Japan, 2015, pp. 6089–6094.

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Steinn, *Introduction to Algorithms*, 3rd ed. Massachusetts Institute of Technology: The MIT Press, 2009.

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