A Coordinate Descent Approach to Optimal Tracking Time Allocation in Point-to-Point ILC

Yiyang Chen[a], Bing Chu[a,*], Christopher T. Freeman[a]

[a]*School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom.*

Abstract

Iterative learning control (ILC) is designed for applications involving multiple executions of the same task. Existing work has applied ILC to point-to-point motion tasks, but has not fully exploited its design freedom to optimize performance criteria other than the tracking accuracy. This paper extends the task description of the point-to-point ILC framework for discrete-time systems by considering the tracking time instants of desired positions as changing variables (i.e. the temporal location of each position can vary). This extension allows the optimization of an additional performance index while maintaining the tracking accuracy. This optimization problem is solved using a two stage design framework, and an iterative algorithm consisting of a norm optimal ILC update and a coordinate descent approach is then derived to minimize an additional performance index, e.g. control effort, for the point-to-point motion tasks. This algorithm is tested on a gantry robot to verify its effectiveness in the presence of model uncertainty and disturbances.

Keywords: iterative learning control, optimization, gantry robot.

1. Introduction

The technique of ILC was proposed for high performance tracking in applications performing the same task during a finite time horizon over multiple trials. It iteratively updates the next trial's input signal based on the current trial information, e.g. input signal and tracking error. Due to the employment of past data, ILC improves the tracking performance with the aim of enabling zero tracking error after sufficient trials even without using accurate model information. This feature makes ILC typically outperform other traditional feedback control methods in terms of high tracking accuracy, which has led to the application of ILC in a wide range of areas, e.g. robotics [1, 2, 3], wafer stage [4], inkjet printer [5] and rehabilitation [6, 7]. See [8, 9] for a detailed overview.

Existing research, e.g. [10, 11, 12, 13, 14, 15], has studied classical ILC aiming at tracking every point along the whole reference trajectory. To achieve greater performance and flexibility, a point-to-point ILC framework was developed in [16] to handle the class of applications performing point-to-point tracking tasks. In point-to-point ILC, it is only necessary to track a subset of distinct positions, r_i, along the reference trajectory at time instants, t_i. Due to the elimination of unnecessary and redundant tracking requirements on the reference trajectory, significant control design flexibility is provided by point-to-point ILC. This flexibility was exploited in [17, 18] to embed an additional optimal performance index within the point-to-point ILC framework. Researchers have applied point-to-point ILC to increase the practical tracking performance of various robotic applications, e.g. positioning tables [19], manipulators [20, 21], unmanned aerial vehicles [22], two-mass systems [23], electro-mechanical systems [24], stroke rehabilitation [25] and motor systems [26].

In existing point-to-point ILC frameworks, the tracking time allocation of the distinct positions is fixed *a priori*. However, the choice of tracking time allocation directly affects the system performance, such as the required control effort and the machine damage caused by transient acceleration. There exists research attempting to optimize the tracking time allocation over the time horizon of given point-to-point tracking tasks. In [27], the input energy is minimized within a class of point-to-point motion planning problems. However, its solution is obtained using only the nominal system model without any model uncertainties, and has not incorporated any control technique like ILC, which can improve the tracking accuracy by learning from the real plant data. Therefore, its tracking performance is highly dependent on the accuracy of the nominal system model. In addition, the work in [28] minimizes the control effort of point-to-point ILC for continuous time systems using a gradient approach. However, this method is not applicable for discrete time systems, since the gradient is only defined for continuous system.

Due to the limitations of the aforementioned work, this paper formulates the point-to-point ILC task description for discrete time systems to consider the tracking time al-

*Corresponding author

Email addresses: Yiyang.Chen@soton.ac.uk (Yiyang Chen), b.chu@soton.ac.uk (Bing Chu), ctf1@soton.ac.uk (Christopher T. Freeman)

location as a changing variable, which is then optimized over the trial index. The problem is formulated into an optimization problem incorporating the desired optimal cost function and simultaneously preserves high point-to-point tracking accuracy. A two stage design method is used to solve this problem, and an iterative algorithm is then developed using a norm optimal ILC update and a coordinate descent approach. To demonstrate its practical efficacy, this algorithm is verified on a gantry robot test platform with the presence of model uncertainty and disturbances.

The notation used in this paper is standard: $\mathbb{N}$ is the set of non-negative integers; $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote the sets of n dimensional real vectors and $n \times m$ real matrices respectively; $\mathbb{S}^n_{++}$ is the set of all $n \times n$ real positive definite matrices; $\langle x, y\rangle$ is the inner product of x and y in some vector space.

2. Problem formulation

This section presents the system dynamics, and introduces a novel point-to-point ILC framework. The tracking time allocation is incorporated into this framework as an additional variable to formulate an optimization problem, which minimizes a desired cost function while maintaining accurate point-to-point tracking.

2.1. System dynamics

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

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

where the subscript $k \in \mathbb{N}$ denotes the trial number; $t \in [0,\ N]$ is the time index (i.e. sample number) with $N < \infty$ being the finite trial length; $x_k(t) \in \mathbb{R}^n$, $u_k(t) \in \mathbb{R}^\ell$ and $y_k(t) \in \mathbb{R}^m$ are the state, input and output respectively; A, B and C are system matrices of compatible dimensions. At the end of each trial, the state is reset to an identical initial value, i.e. $x_k(0) = x_0$, $\forall k > 0$. The system can be represented in an equivalent matrix form

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

where u_k, y_k represent the system input and output as

$$\begin{aligned} u_k &= [u_k(0),\ u_k(1), \ldots, u_k(N-1)]^\top \in \mathbb{R}^{\ell N},\\ y_k &= [y_k(1),\ y_k(2), \ldots, y_k(N)]^\top \in \mathbb{R}^{mN}, \end{aligned} \tag{3}$$

and the matrix G is denoted as

$$G = \begin{bmatrix} CB & 0 & \cdots & 0\\ CAB & CB & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ CA^{N-1}B & CA^{N-2}B & \cdots & CB \end{bmatrix} \in \mathbb{R}^{mN\times \ell N}. \tag{4}$$

The constant signal $d(t) = CA^t x_0$ represents the effect of initial condition, which can be absorbed as a part of the reference signal without loss of generality, i.e. $d = 0$.

The input and output signals belong to vector spaces $\mathbb{R}^{\ell N}$ and $\mathbb{R}^{mN}$ defined with induced norms

$$\|u\|_{[R]} = \sqrt{\sum_{i=0}^{N-1} u^\top(i) R_i u(i)},\ \ \|y\|_{[S]} = \sqrt{\sum_{i=1}^{N} y^\top(i) S_i y(i)}, \tag{5}$$

in which $R_i \in \mathbb{S}^\ell_{++}$ and $S_i \in \mathbb{S}^m_{++}$.

2.2. Point-to-Point ILC framework

The point-to-point ILC design objective is described as iteratively updating the input signal, u_k, such that the associated output, y_k, ultimately tracks given distinct positions, r_i, $i = 1, \ldots, M$, at a subset of time instants

$$\Lambda = [t_1,\ t_2, \ldots,\ t_M]^\top, \tag{6}$$

which is the tracking time allocation of these distinct positions. Moreover, a secondary objective is stipulated such that the input signal, u_k, converges to a unique value, u^*. Therefore, the point-to-point ILC problem is defined as

$$\lim_{k\to\infty} y_k(t_i) = r_i,\ \ i = 1, \ldots, M,\ \ \lim_{k\to\infty} u_k = u^*. \tag{7}$$

The tracking time allocation Λ belongs to the set

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

which is the admissible set of all possible tracking time allocation. For a matrix (or vector) $\mathcal{M}$, and a vector $\Lambda = [t_1, t_2, ..., t_M]^\top$ consisting of integer components, $\mathcal{M}^\Lambda$ is a matrix (or vector respectively) consisting of the corresponding t_i^{th} (block) rows of $\mathcal{M}$, e.g. for the output vector y, y^Λ is defined as follows:

$$y^\Lambda = \begin{bmatrix} y(t_1)\\ \vdots\\ y(t_M) \end{bmatrix} \in \mathbb{R}^{mM}. \tag{9}$$

The vector space $\mathbb{R}^{mM}$ is defined with induced norm

$$\|\omega\|_{[Q]} = \sqrt{\sum_{i=1}^{M} \omega_i^\top Q_i \omega_i}, \tag{10}$$

where

$$\omega = [\omega_1, \ldots, \omega_M]^\top \in \mathbb{R}^{mM},$$

and the matrices $Q_i \in \mathbb{S}^m_{++}$ are the weighting parameters with respect to each time instant. Using the above notation (9), the point-to-point ILC design objective (7) can be equivalently described as iteratively updating the sequence of input $\{u_k\}$ such that

$$\lim_{k\to\infty} y_k^\Lambda = r^p,\ \ \lim_{k\to\infty} u_k = u^*, \tag{11}$$

where

$$r^p = [r_1,\ r_2, \ldots,\ r_M]^\top \in \mathbb{R}^{mM}. \tag{12}$$

The point-to-point ILC problem (11) can be solved using the following update law:

$$u_{k+1} = \mathcal{F}(u_k, e_k^p), \tag{13}$$

where $\mathcal{F}$ is an update function involving the previous input, u_k, and $e_k^p = r^p - y_k^\Lambda$ represents the 'point-to-point' tracking error.

2.3. Optimal tracking time allocation problem

In existing point-to-point ILC, the tracking time allocation, Λ, is generally fixed *a priori*. To extend the task description of point-to-point ILC and fully exploit this extra design freedom, the tracking time allocation will now be considered as a changing variable to optimize some desired performance index while guaranteeing accurate point-to-point tracking.

The **Point-to-Point ILC with Optimal Tracking Time Allocation Problem** can be stated as: iteratively updating the sequences of inputs, $\{u_k\}$, outputs, $\{y_k\}$ and tracking time allocation, $\{\Lambda_k\}$ with the asymptotic property that

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

where the corresponding extracted output, y^*, accurately passes through the set of distinct positions, r^p, i.e.

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

and meanwhile a desired cost function, $f(u, y)$, is optimized, i.e. u^*, y^* and Λ^* are solutions of the problem

$$\begin{aligned} \underset{u,y,\Lambda}{\text{minimize}} \quad & f(u,y) \\ \text{subject to} \quad & r^p = G^\Lambda u,\ y = Gu,\ \Lambda \in \Theta. \end{aligned} \tag{14}$$

This problem formulation significantly extends the task description of the point-to-point ILC framework in terms of releasing the flexible choice of the tracking time allocation, Λ, which enables the optimization of an additional cost function whilst maintaining high tracking accuracy.

Remark 1. The desired cost function $f(u, y)$ captures the performance design requirement of many real life applications selected on the basis of specific design needs. For instant, it can be written in the form of

$$f(u, y) = \|u\|_\infty$$

to minimize the peak input. It can also be written as

$$f(u, y) = \|\ddot{y}\|_S$$

to minimize the acceleration of a robotic movement.

However, as will be seen later, this problem formulation complicates the control design, and is addressed in the following section.

3. A two stage design framework

In this section, a two stage design framework is formulated to solve the optimization problem (14). Considering the control effort as an exemplary cost function, the solution of the two stages are given using a norm optimal ILC update and a coordinate descent approach respectively.

3.1. Framework description

Problem (14) can be expressed as

$$\min_{\Lambda\in\Theta}\{\min_u f(u,y), \text{subject to } r^p = G^\Lambda u,\ y = Gu\}. \tag{15}$$

This involves two steps, which first optimize the cost function over u and then optimize over Λ. The problem (15) can be further expressed as

$$\min_{\Lambda\in\Theta}\{\tilde{f}(\Lambda) := f(u_\infty(\Lambda), Gu_\infty(\Lambda))\}, \tag{16}$$

where $u_\infty(\Lambda) : \mathbb{R}^M \to \mathbb{R}^{\ell N}$ is the global minimizer of the inner optimization problem of (15), i.e.

$$\tilde{f}(\Lambda) = \{\min_u f(u,y), \text{subject to } r^p = G^\Lambda u,\ y = Gu\}. \tag{17}$$

The above reformulation gives rise to a two stage design framework solving the optimization problem (14), i.e.

- *Stage One*: Consider tracking time allocation, Λ, as a constant and solve the inner optimization problem

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

- *Stage Two*: Substitute solution, $u_\infty(\Lambda)$, of the inner optimization problem (18) into problem (15) and then solve the problem

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

In this paper, the authors choose the control effort as the desired cost function, i.e. $f(u, y) = \|u\|^2$, which corresponds to the minimum energy. Moreover, there exists a unique global minimum solution for the problem (18) as this cost function is convex. The solutions of the two stages are provided in the following subsection.

3.2. Solution of the proposed framework

1). Solution of Stage One Design: For any constant value of $\Lambda \in \Theta$, Stage One problem (18) equates to finding the minimum control effort needed to perform a point-to-point tracking task. The next theorem illustrates how the norm optimal ILC update is used to solve this problem.

Theorem 1. If the system $S(A, B, C)$ is controllable, C has full row rank and initial input is set as $u_0 = 0$, the norm optimal ILC update

$$u_{k+1} = u_k + G^{\Lambda*}(I + G^\Lambda G^{\Lambda*})^{-1} e_k^p \tag{20}$$

proposed in [29] iteratively solves Stage One problem (18), with analytic solution

$$u_\infty(\Lambda) = \lim_{k\to\infty} u_k = G^{\Lambda*}(G^\Lambda G^{\Lambda*})^{-1} r^p, \tag{21}$$

where $G^{\Lambda*}$ is the adjoint of G^Λ defined as

$$\left\langle r^p,\ G^\Lambda u\right\rangle_Q = \left\langle G^{\Lambda*} r^p,\ u\right\rangle_R. \tag{22}$$

Proof. The norm optimal ILC law (20) is the iterative solution to an optimization problem which has an identical structure to that discussed in [30] except the definitions of the operators, signals and underlying spaces. Similar to the proof of the continuous time version in [29], it follows that the ILC update law (20) solves the Stage One design objective with analytic solution (21) for u_∞. □

2). Solution of Stage Two Design: The next lemma shows how Stage Two problem (19) is simplified using the analytic solution (21).

Lemma 1. According to the analytic solution (21), Stage Two problem (19) is expressed as

$$\begin{aligned}\min_{\Lambda\in\Theta} \tilde{f}(\Lambda) &= \min_{\Lambda\in\Theta} \|u_\infty(\Lambda)\|_R^2 \\ &= \min_{\Lambda\in\Theta} \left\langle r^p,\ (G^\Lambda G^{\Lambda*})^{-1} r^p\right\rangle_Q. \end{aligned}\tag{23}$$

Proof. See Appendix A. □

When there is only one distinct position need to be tracked, i.e. $M = 1$, the problem (23) can be solved analytically, which is shown in the next theorem.

Theorem 2. If $M = 1$, the problem (23) has an analytic solution

$$\Lambda^* = N$$

with the corresponding minimum control effort

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

where

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

Proof. See Appendix B. □

For this special case, the optimal solution, $\Lambda^* = N$, allows the system output to change gradually to the desired position until the terminal time of the time horizon. However, while there exists more than one distinct positions ($M > 1$), the cost function, $\tilde{f}(\Lambda)$, of the problem (23) is generally non-linear and non-convex with respect to Λ. These properties give rise to difficulties in obtaining a numerical solution using standard integer programming solvers, which are mainly designed for linear and quadratic programming problems.

For discrete time systems, the number of elements in the admissible set Θ is finite, which allows the potential application of the blind search method over the whole space. However, the total number of elements in Θ becomes extremely large as the rise of tracking position number M. This aspect makes it generally inefficient to use the blind search method due to the high computational cost. To balance reliability and efficiency, the next theorem provides a coordinate descent approach to problem (23).

Theorem 3. If $M \geqslant 2$, the following update

$$\Lambda_{j+1} = \mathcal{C}(\Lambda_j) \tag{24}$$

generates a sequence $\{\tilde{f}(\Lambda_j)\}$ converging downward to a limit $\tilde{f}^*$ with initial tracking time allocation Λ_0, where $j \in \mathbb{N}$ denotes the coordinate descent trial number, and

$$\Lambda_j = [t_1^j,\ t_2^j, \ldots,\ t_M^j]^\top.$$

The function $\mathcal{C}$ updates each time instant as

$$t_i^{j+1} = \begin{cases} t_i^{j*},\ i = (j+1) \bmod M, \\ t_i^j,\ \text{else}, \end{cases} \tag{25}$$

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

$$\begin{aligned} \underset{t}{\text{minimize}} \quad & \left\langle r^p,\ (G^\Lambda G^{\Lambda*})^{-1} r^p\right\rangle_Q \\ \text{subject to} \quad & \Lambda = [t_1^j, \ldots,\ t_{i-1}^j,\ t,\ t_{i+1}^j, \ldots,\ t_M^j]^\top, \\ & t \in (t_{i-1}^j,\ t_{i+1}^j). \end{aligned} \tag{26}$$

Proof. See Appendix C. □

Remark 2. Note that although the Stage Two optimization problem (19) can be seen as a form of integer programming problem, the performance function is non-linear (and in fact non-convex), preventing the use of standard integer programming solvers, e.g. Matlab **intlinprog** designed for mixed integer linear programming problems.

Remark 3. The coordinate descent approach provides a local optimal solution to the optimization problem (23), and attempts to approximate the global optimal solution by adjusting the tracking time allocation [31].

Remark 4. To accelerate the convergence rate, the coordinate descent approach (24) allows multiple number of time instants to be updated within a single trial.

4. Implementation of design approach

This section first illustrates how to implement the solutions of the two stages in practice, and then combines them together to yield an efficient algorithm which iteratively solves the optimization problem (14).

4.1. Implementation of Stage One

The direct way of implementing the Stage One solution is to simply use the analytic solution (21). For reasons of robustness, this solution should be implemented using feedforward ILC update (20) with the measured error information. Furthermore, the ILC update (20) can be equivalently reformulated into a feedback and feedforward structure to improve the robust performance due to the use of real-time state feedback. This reformulation exploits the special properties of the matrix G^{Λ} and its adjoint $G^{\Lambda *}$. The following lemma is needed.

Lemma 2. The adjoint $G^{\Lambda *} : \omega \in \mathbb{R}^{mM} \to u \in \mathbb{R}^{\ell N}$ has the following analytic form

$$u(t) = R^{-1}B^{\top}p(t), \tag{27}$$

where $p(t)$ is computed along the reverse time axis as

$$p(t) = A^{\top}p(t+1),\ t \in [t_{i-1},\ t_i),\ i = 1, \ldots, M, \tag{28}$$

with boundary conditions

$$\begin{aligned} p(t_i - 1) &= p(t_i) + C^{\top}Q\omega_i,\ i = 1, \ldots, M, \\ p(N) &= 0. \end{aligned} \tag{29}$$

Proof. See Appendix D. □

Based on Lemma 2, the next proposition describes the feedback plus feedforward implementation in detail.

Proposition 1. The ILC update (20) is alternatively implemented using the feedforward plus feedback solution

$$u_{k+1}(t) = u_k(t) + R^{-1}B^{\top}p_{k+1}(t),\ t = 0, \ldots, N, \tag{30}$$

with

$$\begin{aligned} p_{k+1}(t) = -K(t)(I + BR^{-1}B^{\top}K(t))^{-1}A \\ (x_{k+1}(t) - x_k(t)) + \xi_{k+1}(t), \end{aligned} \tag{31}$$

where the Riccati feedback matrix $K(t)$ is defined as

$$K(t) = A^{\top}K(t+1)(I + BR^{-1}B^{\top}K(t+1))^{-1}A, \tag{32}$$

with boundary conditions

$$\begin{aligned} K(t_i - 1) &= K(t_i) + C^{\top}QC,\ i = 1, \ldots, M, \\ K(N) &= 0, \end{aligned} \tag{33}$$

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

$$\xi_{k+1}(t) = (I + K(t)BR^{-1}B^{\top})^{-1}A^{\top}\xi_{k+1}(t+1), \tag{34}$$

with boundary conditions

$$\begin{aligned} \xi_{k+1}(t_i - 1) &= \xi_{k+1}(t_i) + C^{\top}Qe_k(t_i),\ i = 1, \ldots, M, \\ \xi_{k+1}(N) &= 0. \end{aligned} \tag{35}$$

Proof. See Appendix E. □

To improve the robust performance, the feedback plus feedforward solution is preferred for practical implementation of the Stage One solution, as it incorporates the measured error and real-time state feedback. However, the real-time state feedback requires a state observer, which may increase the computation load especially when the number of states is large. Therefore, a balance or tradeoff should be made between the robustness and computational cost while implementing the Stage One solution.

4.2. Implementation of Stage Two

The coordinate descent approach (24) solves the Stage Two problem (19) in several steps, and it starts from an initial tracking time allocation

$$\Lambda_0 = [t_1^0,\ t_2^0, \ldots,\ t_M^0]^T \in \Theta. \tag{36}$$

At each coordinate descent trial, it solves the problem (26) and re-allocates the tracking time instant t_i^j of each position r_i in order to reduce the control effort. In other words, it picks a value of t from the finite number of points in the discrete time interval $(t_{i-1}^j,\ t_{i+1}^j)$ at each trial to minimize the control effort, i.e. $t_{i-1}^j < t < t_{i+1}^j,\ t \in \mathbb{N}$. In this sense, the blind search method can be carried out to solve this sub-problem with a total number of η_i computational attempts, where

$$\eta_i = (t_{i+1}^j - t_{i-1}^j - 1). \tag{37}$$

Remark 5. Within complex systems, the initial choice of tracking time allocation, Λ_0, may affect the convergence performance of the coordinate descent approach. So an appropriate value of Λ_0 is suggested to be selected, which approximates the global minimum solution and reduces the total number of computational update trials.

Remark 6. The coordinate descent approach splits the problem (19) into a series of sub-problems, and its implementation incurs a much smaller computational load than the direct blind search over the whole set Θ.

4.3. An iterative implementation algorithm

The aforementioned implementations of the two stages can be combined together to generate an algorithm (Algorithm 1), which iteratively solves the problem (14). The symbol Λ_0 is a suitably chosen initial tracking time allocation, and $\epsilon > 0$, $\delta > 0$ are small scalars which depend on the tracking accuracy requirement and performance requirement respectively.

In the ideal case, an accurate system model is available. The analytic solution (21) can be used while computing the converged input in Step 2 and 6, and solve the problem (26) in Step 4 with the given value of r^p. However, there exist model uncertainties and disturbances in practice. To embed robustness, it is necessary to extract information from the real plant dynamics. Therefore, it

Algorithm 1 Point-to-Point ILC with Optimal Tracking Time Allocation

Import: Initial tracking time allocation, Λ_0, system model, $S(A,B,C)$, set of distinct positions, r^p, and admissible set, Θ.

1: **initialization:** Coordinate descent trial number $j = 0$
2: Implement Stage One update (20) with $\Lambda = \Lambda_0$ experimentally until convergence, i.e. $\|e_k^p\| < \epsilon\|r^p\|$; record converged input $u_\infty^{ex}(\Lambda_0)$ and control effort $\tilde{f}(\Lambda_0)$.
3: **repeat**
4: Implement Stage Two update (24) based on the measured data, i.e. $r^p \to G^{\Lambda_j} u_\infty^{ex}(\Lambda_j)$.
5: Set $j \to j+1$.
6: Implement Stage One update (20) with $\Lambda = \Lambda_j$ experimentally until convergence, i.e. $\|e_k^p\| < \epsilon\|r^p\|$; record converged input $u_\infty^{ex}(\Lambda_j)$ and control effort $\tilde{f}(\Lambda_j)$.
7: **until** $\left|\tilde{f}(\Lambda_j) - \tilde{f}(\Lambda_{j-1})\right| < \delta \left|\tilde{f}(\Lambda_{j-1})\right|$
8: **return** Λ_j and $u_\infty^{ex}(\Lambda_j)$

is suggested that the norm optimal ILC update in Step 2 and 6 should be implemented experimentally rather than using the analytic solution (21), and the coordinate descent approach in Step 4 should incorporate experimental measured data.

4.4. A discussion on robustness

In reality, the nominal system model, G, cannot perfectly represent the actual plant, $\hat{G}$, due to model uncertainties. Therefore, it is necessary to perform robustness analysis on Algorithm 1 to ascertain its convergence performance in practice as shown in the next theorem.

Theorem 4. Suppose the actual system matrix, $\hat{G}$, is in an additive form

$$\hat{G} = G + \Delta, \tag{38}$$

where the unknown matrix $\Delta \in \mathbb{R}^{mN\times \ell N}$ represents the model uncertainty. If Δ satisfies

$$\left\|I - \Delta^{\Lambda} G^{\Lambda *}\right\| \leqslant 1, \tag{39}$$

then the sequence $\{e_k^p\}_{k\geqslant 0}$ generated by the update (20), monotonically converges to zero, i.e.

$$\left\|e_{k+1}^p\right\| \leqslant \lambda \left\|e_k^p\right\|, \ \forall\ k \geqslant 0, \tag{40}$$

where $\lambda < 1$ denotes the spectral radius of $(I + G^{\Lambda} G^{\Lambda *})^{-1}$. Moreover, if $G + \Delta$ does not lose rank, the cost function $\tilde{f}(\Lambda)$ in Stage Two problem (19) is bounded above as

$$\tilde{f}(\Lambda) \leqslant \frac{\overline{\delta}^2 \lambda}{\underline{\sigma}^2} \left\|r^p\right\|^2, \tag{41}$$

where $\overline{\delta}$ and $\underline{\sigma}$ are the upper and lower bounds of $\left\|G^{\Lambda}\right\|$ and $\left\|\hat{G}^{\Lambda}\right\|$ respectively.

Proof. See Appendix F. □

Theorem 4 defines robust convergence properties of the Stage One update (20), which guarantee the tracking error converges to zero and perfect point-to-point tracking is achievable in practice. Moreover, the model uncertainties also affect the obtained control effort, $\tilde{f}(\Lambda)$, of the point-to-point tracking task, which has been shown in this theorem to be bounded. However, strong results of the control effort convergence can be guaranteed in specific cases. The next corollary illustrates the robust convergence performance of Algorithm 1 in these specific cases.

Corollary 1. If the model uncertainty is in a scalar form, i.e. $\Delta = \kappa G$, where κ is a constant scalar and satisfies

$$0 \leqslant \kappa \leqslant \frac{2}{\left\|G^{\Lambda}\right\|_\infty^2}, \tag{42}$$

then the sequence $\{e_k^p\}_{k\geqslant 0}$ generated by the update (20), monotonically converges to zero, i.e.

$$\left\|e_{k+1}^p\right\| \leqslant \lambda \left\|e_k^p\right\|, \ \forall\ k \geqslant 0, \tag{43}$$

and the sequence $\{\tilde{f}(\Lambda_j)\}_{j\geqslant 0}$ generated by the update (24) monotonically converges to a limit $\tilde{f}^*$ as

$$\tilde{f}(\Lambda_{j+1}) \leqslant \tilde{f}(\Lambda_j), \ \forall\ j \geqslant 0, \ \lim_{j\to\infty} \tilde{f}(\Lambda_j) = \tilde{f}^*. \tag{44}$$

Proof. See Appendix G. □

The above corollary provides the robust convergence properties of Algorithm 1 for a specific class of systems. In general, it is advisable that users monitor the control effort at each coordinate descent trial, and terminate the algorithm once they observe a rise in the control effort. The robust performance of this algorithm will be illustrated experimentally in the next section.

5. Verification on a gantry robot

In this section, a three-axis gantry robot is used as the test platform to show the performance of Algorithm 1.

5.1. Task specification

The gantry robot shown in figure 1 is required to perform a point-to-point motion task, and the authors focus on the motion of the z-axis among all three axes. This task is specified as enforcing that the end-effector to pass through five distinct positions ($M = 5$) represented by

$$r^p = [0.0048,\ 0.0029,\ -0.0029,\ -0.0048,\ 0]^\top \tag{45}$$

along the z-axis during a given time period, $T = 2s$. The distinct positions are shown in figure 9 as the dots, and the *a priori* tracking time allocation is

$$\Lambda_r = [30, 60, 120, 150, 200]^\top. \tag{46}$$

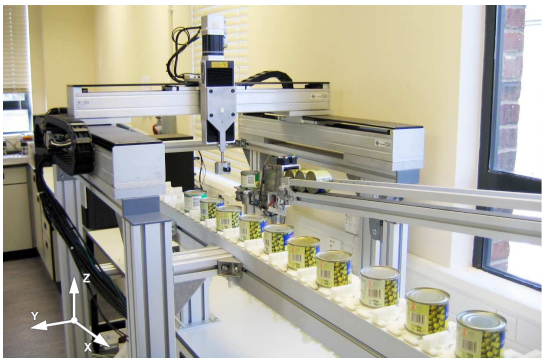

Figure 1: Three-axis gantry robot test platform.

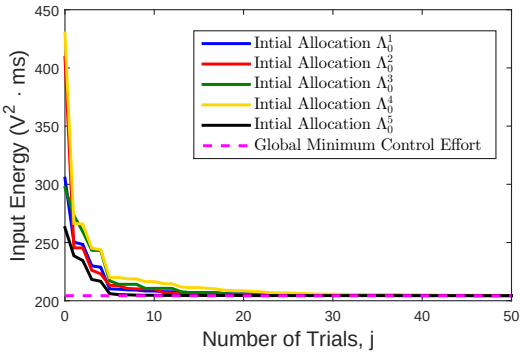

Figure 2: Numerical control effort results at each trial for nominal model test.

It is often expensive and infeasible to perform system identification on practical systems, and the system model might also change due to environmental issues, such as temperature and humidity. Therefore, a nominal model is used to represent the system dynamics of the z-axis, i.e.

$$G_z(z) = \frac{3 \times 10^{-4}}{z - 1}, \tag{47}$$

with a sampling time $T_s = 0.01s$. In addition, the weighting matrices of the ILC update in Step 2 and 6 are taken as diagonal matrices, i.e. $Q_i = qI$ and $R_i = rI$, satisfying $q/r = 1,000,000$ where q and r are positive scalars.

5.2. Numerical tests

A numerical test is first carried out to verify the performance of Algorithm 1 in the ideal case, i.e. the nominal model (47) is sufficiently accurate, $\hat{G} = G_z(z)$. Algorithm 1 is implemented using the nominal model (47) with five choices of initial allocations, Λ_0, i.e.

$$\Lambda_0^1 = [20, 60, 100, 140, 180]^\top,\ \Lambda_0^2 = [10, 50, 100, 150, 190]^\top$$
$$\Lambda_0^3 = [30, 70, 100, 130, 170]^\top,\ \Lambda_0^4 = [10, 50, 90, 130, 170]^\top$$
$$\Lambda_0^5 = [30, 70, 110, 150, 190]^\top,$$

for a total number of 50 coordinate descent trials. Note that the final solutions starting from the five choices all converge to $[49, 68, 128, 148, 200]^\top$, and the corresponding control effort at each trial are plotted in figure 2. For comparison, a blind search has been carried out to check all the possible tracking time allocations to find the global solution $[50, 70, 130, 150, 200]^\top$, and the corresponding global minimum control effort is plotted as the dashed magenta line in the same figure.

It is clear from the figure that, although the solution of Algorithm 1 is different form the global solution, it can reduce the control effort of the given task to the value 204.939, which is extremely close to global minimum value 204.807 (only 0.06% difference). This numerical test confirms the argument made in Remark 3, such that the local optimal solution approximates the global one. In addition, Algorithm 1 requires less than 3,000 computational attempts of control effort comparison in total, while the blind search requires more than one billion computational attempts. This test further confirms the statement about the efficiency of this algorithm in Remark 6.

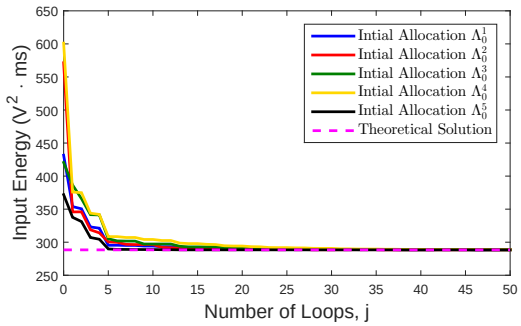

Figure 3: Numerical control effort results at each trial for robust test.

To simulate the robust performance of Algorithm 1, the accurate system model $\hat{G}$ is considered as

$$\hat{G}_z(z) = \frac{2.52 \times 10^{-4}}{z - 1}. \tag{48}$$

The above simulation procedures are repeated using the same nominal model (47). The control effort at each trial with respect to the five choices of initial allocations are plotted in figure 3, and the final solutions all converge to $[48, 67, 128, 148, 200]^\top$. The control effort (288.448) for the theoretical obtained solution $[49, 68, 128, 148, 200]^\top$ is plotted as the dashed magenta line in the same figure. It is clear that the converged control effort (288.416) is 0.011% smaller than that corresponding to the theoretical value.

The input and output at Λ_0^1 and the converged solution are plotted in figure 4, and it is obvious that the output trajectories accurately pass the given distinct positions even with the existence of model uncertainties. These simulation results support the statement that Algorithm 1 has a certain level of robust performance against model uncertainties, which not only reduces the control effort, but also keeps a high level of point-to-point tracking accuracy.

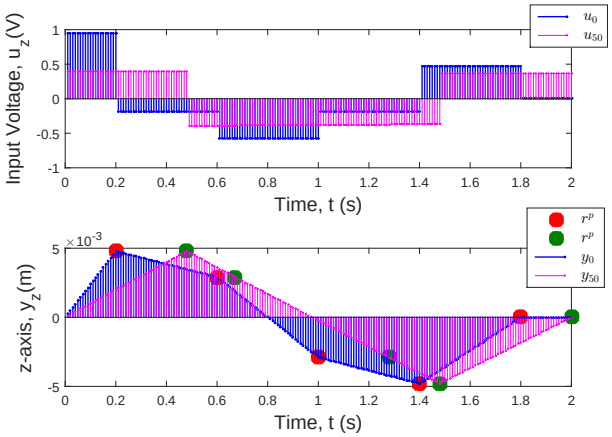

Figure 4: Numerical converged input and output trajectories for initial and final trials in robust test.

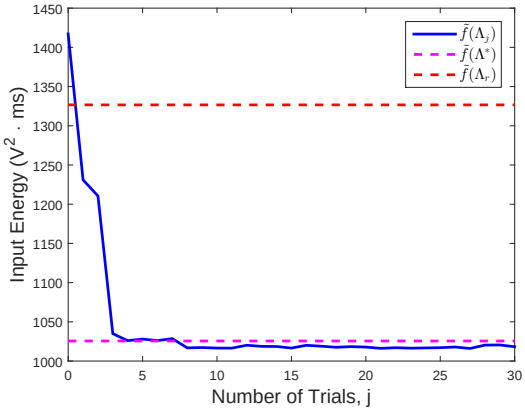

Figure 5: Experimental control effort results at each trial with accurate model.

5.3. Experimental results

Algorithm 1 is next implemented to perform the point-to-point tracking task on the gantry robot. An identified system model

$$\tilde{G}_z(z) = \frac{(36.48z^2 + 3.569z + 0.217) \times 10^{-5}}{z^3 - 0.9941z^2 - 0.005046z - 0.0008451} \quad (49)$$

is obtained to represent the system dynamics of the z-axis in [32] via system identification technique. To provide baseline tracking, disturbance rejection and smooth edge tracking, a feedback proportional controller with a gain of 150 is added to the real plant to give

$$G = \frac{\tilde{G}_z(z)}{1 + K\tilde{G}_z(z)}, \; K = 150, \quad (50)$$

which can be equivalently written in minimal state space form $S(A, B, C)$. The initial allocation is chosen as $\Lambda_0 = [20, 60, 100, 140, 180]^\top$, which is equally distributed along the time horizon, $[0, \; N]$.

A total number of 30 coordinate descent trials are performed until the control effort converges, and the control effort, $\tilde{f}(\Lambda_j)$, at each trial is plotted in figure 5 with the converged value, $\tilde{f}(\Lambda_{30}) = 1018.3$. Also, the point-to-point norm optimal update in [29] is performed with the *a priori* allocation, Λ_r, to obtain the corresponding control effort, $\tilde{f}(\Lambda_r) = 1326.6$, which is plotted in the same figure as the red dashed line. From this figure, it is clear that the converged energy provided by Algorithm 1 is 23.2% less than the control effort respect to the *a priori* allocation, which confirms that this algorithm can significantly reduce the control effort. For more details, the corresponding tracking time allocation at each trial is also plotted in figure 6 with the minimum solution, $\Lambda_{30} = [53, 58, 132, 139, 200]^\top$.

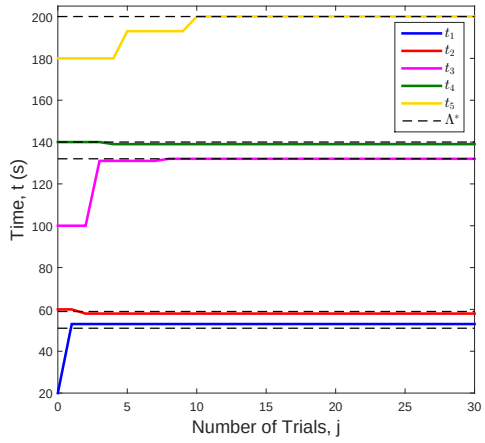

Figure 6: Experimental time-point position results at each trial with accurate model.

The theoretical minimum solution is obtained using the model (50) in simulation as $\Lambda^* = [51, 59, 132, 140, 200]^\top$. This theoretical solution is plotted as the dashed black lines in figure 6, and it is clear that it is close to the experimentally obtained solution, Λ_{30}, as the model uncertainty of identical system model (49) is sufficiently small. For further comparison, the gantry robot's control effort, $\tilde{f}(\Lambda^*) = 1025.6$, with respect to the theoretical optimal tracking time allocation is also plotted in figure 5. There is only 0.7% difference in control effort between the solutions obtained from practical implementation and theoretical calculation.

The above verifications have also been conducted using the nominal model (47) to test the robust performance of Algorithm 1 in practice. The results are plotted in figure 7 and 8. From figure 7, a 23.8% reduction of control effort is achieved via practical implementation compared to the value, $\tilde{f}(\Lambda_r) = 1378.2$, respect to the *a priori* allocation, Λ_r. Also, it is obvious from figure 8 that as the model (49) is not accurate enough, the theoretical solution $\Lambda^* = [48, 58, 130, 140, 200]^\top$ is not exactly the same as the experimentally obtained one, $\Lambda_{30} = [54, 59, 128, 137, 200]^\top$. As a result, in figure 7, the control effort, $\tilde{f}(\Lambda^*) = 1102.1$, at Λ^* is 4.4% less than the control effort, $\tilde{f}(\Lambda_{30}) = 1053.4$, at Λ_{30}. This fact confirms the necessity of experimental

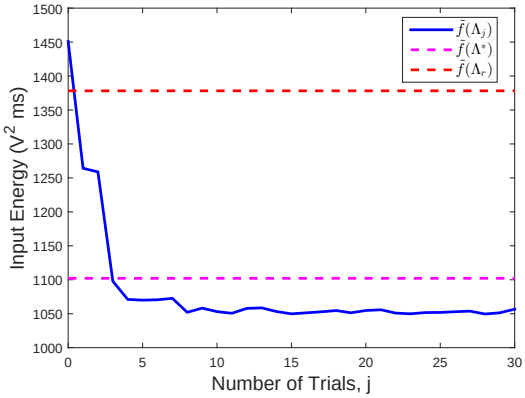

Figure 7: Experimental control effort results at each trial.

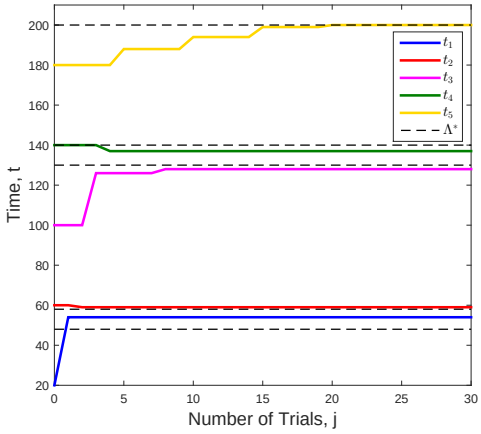

Figure 8: Experimental time-point position results at each trial.

implementation and its advantages over theoretical computation using a nominal model.

To check the tracking performance under the existence of model uncertainties, the input and output of the initial and final trials have been selected and plotted in figure 9. From this figure, the converged output trajectories of both trials accurately pass through the given distinct positions marked as red and green dots, i.e. perfect tracking is achieved (note that there only exists 0.00053 mean square point-to-point tracking error at the final trial). Therefore, Algorithm 1 can still reduce the control effort and guarantee high point-to-point tracking accuracy at the same time against model uncertainties. This test of Algorithm 1 on the gantry robot platform further confirms its robust performance in practice.

Experiments with other initial tracking time allocations and other R_i, Q_i values provide varying convergence speed but similar monotonic convergence performance to the above results. For brevity, these results are omitted.

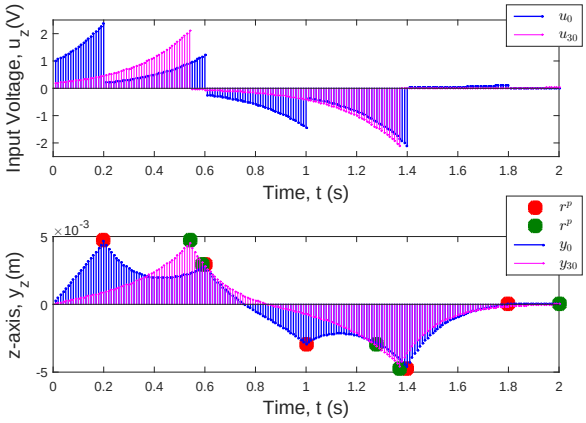

Figure 9: Experimental converged input and output trajectories for initial and final trials.

6. Conclusion and future work

This paper significantly extends the task description of point-to-point ILC by enabling the tracking time allocation of distinct positions as a changing variable. This allows the employment of an additional optimal performance index, e.g. control effort, within the point-to-point ILC framework. A point-to-point tracking problem has been formulated in this paper to achieve accurate point-to-point tracking as well as optimization of an additional cost function. A Two Stage design framework is derived to solve this problem in two steps. This framework has been exemplified to obtain the minimum control effort, which yields an iterative algorithm based on a norm optimal ILC update and a coordinate descent approach. To verify its practical performance, a gantry robot is used to test the performance of the proposed algorithm in practice.

From the experimental results, a certain degree of robustness has been confirmed within the scope of the proposed algorithm in terms of reducing control effort. However, the Two Stage design framework can optimize other cost functions than control effort, and corresponding algorithms will be developed in the future. The ILC problem formulation in this paper can be generalized to other systems, such as continuous linear time-invariant systems and discrete linear time varying systems, without difficulty by extending the signal space definition to Hilbert space formalism. Moreover, the system constraints will be considered in the design problem. The above constitutes part of our future research and will be reported separately.

Appendix A. Proof of Lemma 1

As the control effort is chosen as the desired cost function, it follows that

$$\tilde{f}(\Lambda) = \|u_\infty(\Lambda)\|_R^2. \tag{A.1}$$

Then, substitute the analytic solution (21) into the problem (19), and explore the properties of adjoint in vector space to give

$$\begin{aligned}\min_{\Lambda\in\Theta}\tilde{f}(\Lambda) &= \min_{\Lambda\in\Theta}\|u_\infty(\Lambda)\|_R^2 = \min_{\Lambda\in\Theta}\ \langle u_\infty(\Lambda),\ u_\infty(\Lambda)\rangle_R \\ &= \min_{\Lambda\in\Theta}\ \left\langle G^{\Lambda*}(G^\Lambda G^{\Lambda*})^{-1}r^p, G^{\Lambda*}(G^\Lambda G^{\Lambda*})^{-1}r^p\right\rangle_R \\ &= \min_{\Lambda\in\Theta}\ \left\langle G^\Lambda G^{\Lambda*}(G^\Lambda G^{\Lambda*})^{-1}r^p, (G^\Lambda G^{\Lambda*})^{-1}r^p\right\rangle_Q \\ &= \min_{\Lambda\in\Theta}\ \left\langle r^p,\ (G^\Lambda G^{\Lambda*})^{-1}r^p\right\rangle_Q,\end{aligned}$$

which completes the proof.

Appendix B. Proof of Theorem 2

At this special case, there is a single distinct position, i.e. $M = 1$ and $\Lambda = t_1$. Denote $\Psi_{t_1} = G^{t_1}G^{t_1*}$ which is explicitly written as

$$\Psi_{t_1} = \sum_0^{t_1} Ce^{A(t_1-t)}BR^{-1}(Ce^{A(t_1-t)}B)^\top. \tag{B.1}$$

The optimization problem (23) becomes

$$\min_{\Lambda\in\Theta}\|u_\infty(\Lambda)\|_R^2 = \min_{\Lambda\in\Theta}\ \left\langle r^p,\ \Psi_{t_1}^{-1}r^p\right\rangle_Q. \tag{B.2}$$

For any x, there exists

$$\begin{aligned}&\langle x,\ (\Psi_N - \Psi_{t_1})x\rangle_Q \\ &= \langle x,\ (\sum_{t_1}^{N} Ce^{A(t_1-t)}BR^{-1}(Ce^{A(t_1-t)}B)^\top)x\rangle_Q \geqslant 0,\end{aligned}$$

which gives rise to

$$\Psi_{t_1} \leqslant \Psi_T,\ \forall\ t_1^- \leqslant t_1 \leqslant t_1^+ = N.$$

The above positive properties yield $\Psi_{t_1}^{-1} \geqslant \Psi_N^{-1}$, and thus

$$\left\langle r^p,\ \Psi_{t_1}^{-1}r^p\right\rangle_Q \geqslant \left\langle r^p,\ \Psi_T^{-1}r^p\right\rangle_Q. \tag{B.3}$$

It follows from (B.3) that $t_1 = t_1^+ = N$ is the minimum solution of the problem (23), which completes the proof.

Appendix C. Proof of Theorem 3

The coordinate descent approach splits the problem (23) into several sub-problems. At each trial, it solves a sub-problem (26) to update a single time instant $t_{i,j}$ while keeping the others as fixed values. The definition of each sub-problem gives rise to the monotonically decreasing property of the sequence $\{\tilde{f}(\Lambda_j)\}$, i.e.

$$\tilde{f}(\Lambda_{j+1}) \leqslant \tilde{f}(\Lambda_j). \tag{C.1}$$

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

Appendix D. Proof of Lemma 2

The matrix G^Λ has the following structure

$$G^\Lambda u = [G_1u, \ldots, G_Mu]^\top, \tag{D.1}$$

where

$$G_iu = \sum_{t=0}^{t_i-1} CA^{t_i-t-1}Bu(t),\ i = 1, \ldots, M. \tag{D.2}$$

Consider the matrix $G_i \in \mathbb{R}^{m\times\ell N}$ via equation

$$\begin{aligned}{\omega_i}^\top QG_iu &= \omega_i^\top QF_i\sum_{t=0}^{t_i-1} CA^{t_i-t-1}Bu(t) \\ &= \sum_{t=0}^{t_i-1}(R^{-1}B(A^\top)^{t_i-t-1}C^\top F_i^\top Q\omega_i)^\top Ru(t),\end{aligned} \tag{D.3}$$

and the condition

$$\omega_i^\top QG_iu = \sum_{t=0}^{N}((G_i^*\omega_i)(t))^\top Ru(t), \tag{D.4}$$

due to the definition of the adjoint, i.e.

$$\langle\omega_i,\ G_iu\rangle_Q = \langle G_i^*\omega_i,\ u\rangle_R. \tag{D.5}$$

Therefore, (D.3) and (D.4) give rise to

$$(G_i^*\omega_i)(t) = \begin{cases} R^{-1}B(A^\top)^{t_i-t-1}C^\top F_i^\top Q\omega_i, & t < t_i, \\ 0, & t \geqslant t_i,\end{cases}$$

which can be further written as

$$(G_i^*\omega_i)(t) = R^{-1}B^\top p_i(t), \tag{D.6}$$

where $p_i(t) = 0$ on $[t_i,\ N]$, and on $[0,\ t_i)$

$$p_i(t) = A^\top p_i(t+1),\ p_i(t_i-1) = C^\top Q\omega_i. \tag{D.7}$$

Due to linearity, the adjoint $G^{\Lambda*}$ can be written as the sum of adjoints, $G_1^*, \ldots, G_M^*$, i.e.

$$G^{\Lambda*}\omega = \sum_{i=1}^{M}(G_i^*\omega_i)(t) = R^{-1}B^\top p(t) \tag{D.8}$$

as shown in (27), where $p(t) = \sum_{i=1}^M p_i(t)$. Therefore, the costate $p(t)$ in (28) with its boundary conditions (29) are derived from (D.6), which completes the proof.

Appendix E. Proof of Proposition 1

The ILC update (20) has the feedforward form

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

From Lemma 2, $G^{\Lambda*}e_{k+1}^p(t)$ is computed using the analytic form

$$G^{\Lambda*}e_{k+1}^p(t) = R^{-1}B^\top p_{k+1}(t), \tag{E.2}$$

where $p_{k+1}(t)$ denotes the costate in reverse time, i.e.

$$p_{k+1}(t) = -A^\top p_k(t+1),\ t \in [t_{i-1},\ t_i),\ i = 1, \ldots, M,$$

with boundary conditions

$$\begin{aligned} p_{k+1}(t_i - 1) &= p_{k+1}(t_i) + C^\top Q e_{k+1}(t_i),\ i = 1, \ldots, M, \\ p_{k+1}(N) &= 0. \end{aligned} \tag{E.3}$$

Substituting (E.2) into (E.1) yields the solution (30).

Assuming full state knowledge, the costate equation (E.2) yields a causal implementation

$$\begin{aligned} p_{k+1}(t) = -K(t)(I + BR^{-1}B^\top K(t))^{-1}A \\ (x_{k+1}(t) - x_k(t)) + \xi_{k+1}(t). \end{aligned} \tag{E.4}$$

According to (1), (30) and (E.4), the following equation

$$\begin{aligned} &x_{k+1}(t+1) - x_k(t+1) \\ &\quad = A(x_{k+1}(t) - x_k(t)) + B(u_{k+1}(t) - u_k(t)) \\ &\quad = A(x_{k+1}(t) - x_k(t)) + BR^{-1}B^\top p_{k+1}(t) \\ &\quad = (I + BR^{-1}B^\top K(t))^{-1}A(x_{k+1}(t) - x_k(t)) \\ &\qquad + BR^{-1}B^\top \xi_{k+1}(t) \end{aligned} \tag{E.5}$$

is derived from [33]. Then substitute (E.4) and (E.5) into the costate (E.2), which yields an equation of the form

$$\begin{aligned} &\mathcal{H}(A, B, R^{-1}, K(t), K(t+1))[x_{k+1}(t+1) - x_k(t+1)] \\ &\quad = \mathcal{G}(A, B, R^{-1}, K(t), \xi_{k+1}(t), \xi_{k+1}(t+1)), \end{aligned} \tag{E.6}$$

where $\mathcal{H}(\cdot)$ and $\mathcal{G}(\cdot)$ are functions of their arguments and independent of the states. It is clear that both functions should be set to zero to make the equation (E.6) independent of the current state difference. This hence yields the definition of the discrete Matrix Riccati equation $K(t)$ and the optimal predictor $\xi_{k+1}(t)$ in (32) and (34).

Considering the boundary conditions, there exists an additive term $C^\top Q e_{k+1}(t_i)$ in equation (E.3). Thus

$$e_{k+1}(t_i) = e_k(t_i) - C(x_{k+1}(t_i) - x_k(t_i)), \tag{E.7}$$

which yields

$$C^\top Q e_{k+1}(t_i) = C^\top QC(x_{k+1}(t_i) - x_k(t_i)) + C^\top Q e_k(t_i),$$

and gives rise to the boundary conditions in (33) and (35).

Appendix F. Proof of Theorem 4

Based on the addictive form (38) and update (20), it follows from the derivations in [34] such that the monotonic convergence condition (40) holds, and the sequence $\{e_k^p\}$ converges to zero.

As the error converges to zero, the tracking design objective is achieved as

$$r^p = \hat{G}^\Lambda u_\infty^{ex}(\Lambda) \tag{F.1}$$

for each coordinate descent trial. From Algorithm 1, the value $\hat{r}^p$ used in Stage Two update (24) is computed based on the measured data as

$$\hat{r}^p = G^\Lambda u_\infty^{ex}(\Lambda). \tag{F.2}$$

Since $G+\Delta$ does not lose rank, the lower bound $\underline{\sigma}$ of $\left\|\hat{G}^\Lambda\right\|$ is non-zero, which gives

$$\|u_\infty^{ex}(\Lambda)\| \leqslant \frac{1}{\underline{\sigma}} \cdot \|r^p\|. \tag{F.3}$$

From the definition of the matrix G^Λ, its norm is bounded above and there exists

$$\|\hat{r}^p\| \leqslant \overline{\delta} \cdot \|u_\infty^{ex}(\Lambda)\|. \tag{F.4}$$

The above inequalities yield

$$\|\hat{r}^p\| \leqslant \frac{\overline{\delta}}{\underline{\sigma}} \cdot \|r^p\|. \tag{F.5}$$

Using (F.5), the upper bound of the cost function is obtained as

$$\tilde{f}(\Lambda) = \left\langle \hat{r}^p,\ (G^\Lambda G^{\Lambda *})^{-1}\hat{r}^p \right\rangle \leqslant \lambda \cdot \langle \hat{r}^p,\ \hat{r}^p \rangle \leqslant \frac{\overline{\delta}^2 \cdot \lambda}{\underline{\sigma}^2} \cdot \|r^p\|^2, \tag{F.6}$$

which completes the proof.

Appendix G. Proof of Corollary 1

From the definition of the norms, it follows that

$$\left\|G^\Lambda\right\|^2 \leqslant \left\|G^\Lambda\right\|_\infty^2. \tag{G.1}$$

As the matrix $G^\Lambda G^{\Lambda *}$ is positive definite, for any κ satisfies the condition (42), these exists

$$0 \leqslant \kappa \leqslant \frac{2}{\|G^\Lambda\|_\infty^2} \leqslant \frac{2}{\|G^\Lambda\|^2}, \tag{G.2}$$

which implies

$$\left\|I - \kappa G^\Lambda G^{\Lambda *}\right\| \leqslant 1. \tag{G.3}$$

For $\Delta = \kappa G$, the condition (39) collapses to (G.3), so the proof of the monotonic convergence condition (43) follows from that in Theorem 4.

Since Δ is in form of scalar, it follows that

$$\hat{G} = (1 + \kappa)G \tag{G.4}$$

based on the addictive form (38). Using (F.1), (F.2) and (G.4), the relationship between the constant value r^p and the numerical computed value $\hat{r}^p$ is yielded as

$$r^p = \hat{G}^\Lambda u_\infty^{ex}(\Lambda) = (1 + \kappa)G^\Lambda u_\infty^{ex}(\Lambda) = (1 + \kappa)\hat{r}^p, \tag{G.5}$$

which means that the value $\hat{r}^p$ at each coordinate descent trial is proportional to the value r^p.

According to the coordinate descent update (24), the cost function $\left\langle \hat{r}^p,\ (G^{\Lambda}G^{\Lambda *})^{-1}\hat{r}^p \right\rangle$ is optimized at each coordinate descent trial. Based on (G.5), the optimized cost function is also proportional to $\tilde{f}(\Lambda)$ as

$$\begin{aligned}\left\langle \hat{r}^p,\ (G^{\Lambda}G^{\Lambda *})^{-1}\hat{r}^p \right\rangle &= \frac{1}{(1+\kappa)^2}\cdot\left\langle r^p,\ (G^{\Lambda}G^{\Lambda *})^{-1}r^p \right\rangle, \\ &= \frac{1}{(1+\kappa)^2}\cdot\tilde{f}(\Lambda). \qquad \text{(G.6)}\end{aligned}$$

Since the function $\tilde{f}(\Lambda)$ is non-negative (bounded below), the sequence $\{\tilde{f}(\Lambda_j)\}$ monotonically converges to a limit $\tilde{f}^*$, which completes the proof.

Acknowledgment

This work was supported by the ZZU-Southampton Collaborative Research Project under Grant 16306/01.

References

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

[2] M. Norrlof, An adaptive iterative learning control algorithm with experiments on an industrial robot, IEEE Transactions on Robotics and Automation 19 (2) (2002) 245–251.

[3] X. Jin, Fault-tolerant iterative learning control for mobile robots non-repetitive trajectory tracking with output constraints, Automatica 94 (2018) 63–71.

[4] T. Oomen, C. R. Rojas, Sparse iterative learning control with application to a wafer stage: Achieving performance, resource efficiency, and task flexibility, Mechatronics 47 (2017) 134–147.

[5] J. Bolder, T. Oomen, S. Koekebakker, M. Steinbuch, Using iterative learning control with basis functions to compensate medium deformation in a wide-format inkjet printer, Mechatronics 24 (8) (2014) 944–953.

[6] C. T. Freeman, Newton-method based iterative learning control for robot-assisted rehabilitation using FES, Mechatronics 24 (8) (2014) 934–943.

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

[8] D. Bristow, M. Tharayil, A. Alleyne, A survey of iterative learning control, IEEE Control Systems Magazine 26 (3) (2006) 96–144.

[9] H.-S. Ahn, Y. Chen, K. L. Moore, Iterative learning control: Brief survey and categorization, IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews 37 (6) (2007) 1099–1121.

[10] R. W. Longman, Iterative learning control and repetitive control for engineering practice, International Journal of Control 73 (10) (2000) 930–954.

[11] A. Tayebi, M. B. Zaremba, Robust iterative learning control design is straightforward for uncertain LTI systems satisfying the robust performance condition, IEEE Transactions on Automatic Control 48 (1) (2003) 101–106.

[12] K. L. Moore, Y. Q. Chen, V. Bahl, Monotonically convergent iterative learning control for linear discrete-time systems, Automatica 41 (9) (2005) 1529–1537.

[13] S. Mishra, M. Tomizuka, Projection-based iterative learning control for wafer scanner systems, IEEE/ASME Transactions on Mechatronics 14 (3) (2009) 388–393.

[14] B. Altin, K. Barton, Robust iterative learning for high precision motion control through L1 adaptive feedback, Mechatronics 24 (6) (2014) 549–561.

[15] J. Bolder, T. Oomen, Inferential iterative learning control: A 2D-system approach, Automatica 71 (2016) 247–253.

[16] C. T. Freeman, Z. Cai, E. Rogers, P. L. Lewin, Iterative learning control for multiple point-to-point tracking application, IEEE Transactions on Control Systems Technology 19 (3) (2011) 590–600.

[17] C. T. Freeman, Constrained point-to-point iterative learning control with experimental verification, Control Engineering Practice 20 (5) (2012) 489–498.

[18] T. D. Son, H.-S. Ahn, K. L. Moore, Iterative learning control in optimal tracking problems with specified data points, Automatica 49 (5) (2013) 1465–1472.

[19] H. Ding, J. Wu, Point-to-point control for a high-acceleration positioning table via cascaded learning schemes, IEEE Transactions on Industrial Electronics 54 (5) (2007) 2735–2744.

[20] J. Park, P. H. Chang, H. S. Park, E. Lee, Design of learning input shaping technique for residual vibration suppression in an industrial robot, IEEE/ASME Transactions on Mechatronics 11 (1) (2006) 55–65.

[21] A. Tayebi, S. Abdul, M. B. Zaremba, Y. Ye, Robust iterative learning control design: Application to a robot manipulator, IEEE/ASME Transactions on Mechatronics 13 (5).

[22] K. Barton, D. Kingston, Systematic surveillance for UAVs: A feedforward iterative learning control approach, in: American Control Conference, 2013, pp. 5917–5922.

[23] J. van de Wijdeven, O. Bosgra, Residual vibration suppression using Hankel iterative learning control, International Journal of Robust Nonlinear Control 18 (10) (2008) 1034–1051.

[24] C. T. Freeman, T. V. Dinh, Experimentally verified point-to-point iterative learning control for highly coupled systems, International Journal of Adaptive Control and Signal Processing 29 (2015) 302–324.

[25] C. T. Freeman, A.-M. Hughes, J. H. Burridge, P. H. Chappell, P. L. Lewin, E.Rogers, Iterative learning control of FES applied to the upper extremity for rehabilitation, Control Engineering Practice 17 (3) (2009) 368–381.

[26] S.-H. Zhou, Y. Tan, D. Oetomo, C. T. Freeman, E. Burdet, I. Mareels, Modeling of endpoint feedback learning implemented through point-to-point learning control, IEEE Transactions on Control Systems Technology 25 (5) (2017) 1576–1585.

[27] P. Janssens, G. Pipeleers, M. Diehl, J. Swevers, Energy optimal time allocation of a series of point-to-point motions, IEEE Transactions on Control Systems Technology 22 (6) (2014) 2432–2435.

[28] Y. Chen, B. Chu, C. T. Freeman, Point-to-point iterative learning control with optimal tracking time allocation, IEEE Transactions on Control Systems Technology 26 (4) (2018) 1685–1698.

[29] D. H. Owens, C. T. Freeman, T. V. Dinh, Norm-optimal iterative learning control with intermediate point weighting: theory, algorithms, and experimental evaluation, IEEE Transactions on Control Systems Technology 21 (3) (2013) 999–1007.

[30] N. Amann, D. Owens, E. Rogers, Iterative learning control using optimal feedback and feedforward actions, International Journal of Control 65 (2) (1996) 277–293.

[31] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Steinn, Introduction to Algorithms, 3rd Edition, The MIT Press, Massachusetts Institute of Technology, 2009.

[32] J. D. Ratcliffe, Iterative learning control implemented on a multi-axis system, Ph.D. thesis, University of Southampton, Sothampton (June 2005).

[33] N. Amann, Optimal algorithms for iterative learning control, Ph.D. thesis, University of Exeter, UK (1996).

[34] D. H. Owens, C. T. Freeman, B. Chu, An inverse-model approach to multivariable norm optimal iterative learning control with auxiliary optimisation, International Journal of Control 87 (8) (2014) 1646–1671.