

# Generalized Iterative Learning Control using Successive Projection: Algorithm, Convergence and Experimental Verification

Yiyang Chen, Bing Chu\* and Christopher T. Freeman

**Abstract**—Iterative learning control (ILC) is a high performance control design method for systems working in a repetitive manner. ILC has traditionally focused on tracking a reference defined at all points over a finite time interval; recent developments have begun to exploit the design freedom unlocked by tracking only a finite number of distinct time instants driven by the needs of e.g. robotic pick-and-place tasks. This paper proposes a generalized ILC paradigm which extends and unifies the scope of existing design frameworks by amalgamating previous task descriptions and embedding system constraints on the input and output. A novel solution is then derived using a successive projection method which provides well defined convergence properties. The proposed design framework is illustrated by applying it to a spatial reference tracking problem with experimental results on a gantry robot testing platform demonstrating its effectiveness.

**Index Terms**—iterative learning control, projection methods, constraint handling, spatial path tracking

## I. INTRODUCTION

Iterative learning control (ILC) is a control design methodology which has focused on improving the tracking accuracy of systems repeatedly tracking a reference signal  $r$  defined over a finite time horizon  $[0, N]$ . By updating the control input based on experimental data, e.g. the output  $y$ , from previous trials, a rich theoretical framework has emerged in [1]–[5] to enable the tracking error  $e = r - y$  to converge to zero after sufficient trials even without using an exact model information. This feature has led ILC to be widely applied to precision industrial tasks, such as robotic systems [6]–[9], chemical batch process [10], [11], medicine [12] and stroke rehabilitation [13]. See [14]–[16] for a detailed overview.

ILC traditionally focuses on tracking a reference defined over the whole trial interval  $[0, N]$  with a range of design algorithms available in the literature [14]–[24]. However in many application domains, the output may only be available at certain time instants, or only certain time instants may be relevant to capture the task requirement, e.g. high-acceleration positioning tables [25], robotic manipulators [26], two-mass systems [27], electro-mechanical systems [28], [29] and human motor system [30]. In recent years, the ‘classical’ ILC problem has been extended to meet the above needs leading to the so-called intermediate point (or point-to-point) ILC. This eliminates the unnecessary output tracking requirements between the intermediate points, and allows significant control

design flexibility to be exploited. Various intermediate point ILC algorithms were proposed to provide explicit algorithmic solutions to this problem. For example, several approaches have updated a ‘complete’ reference passing through all the desired intermediate points along the trial [25]–[27]. Design without explicitly updating the whole reference based on an optimization approach was considered in [31]–[36]. A special case called terminal ILC was studied in [37]–[40], which only tracks a reference defined at the end point of the trial.

This paper proposes a generalized ILC design paradigm that unifies and further extends existing design frameworks to tackle more general tracking requirements, including a mix of intermediate point tracking, sub-interval tracking and constraint satisfaction on system inputs and outputs, problems which cannot be addressed using existing design methods. Examples of such tasks include but are not limited to welding or cutting movements where the tracking is required at certain discrete positions, and the movement between them is restricted to line segments with hard/soft system constraints. To solve this problem, we formulate the design requirements using an abstract Hilbert space setting and derive an algorithm using a successive projection method with well defined convergence properties. Note that the initial concept was reported in [41], which however used a different norm optimal ILC approach and could not incorporate the system constraints that are key to the aforementioned applications.

The proposed generalized ILC is very powerful as it can handle many different types of tracking requirements and a number of existing ILC design methods can be recovered as special cases: e.g. classical ILC [14]–[24], intermediate point ILC [25]–[36], terminal ILC [37]–[40], intermediate and sub-interval ILC [41], [42] and piecewise linear spatial ILC [43]. As a particularly interesting application, it can be applied to solve the challenging problem of tracking a reference defined in space with no *a priori* temporal stipulation by considering the spatial tracking requirement as constraints on the system output. By removing the unnecessary temporal tracking constraints, the design freedom of this ‘spatial’ tracking problem setup can potentially yield significantly better tracking performance in practice. This problem has been studied in the wider non-ILC setting, e.g. in [44], [45] based on feedback linearization and in [46], [47] using optimization techniques, which assume an accurate model. However, the resulting design can be sensitive to model uncertainties and perfect tracking of the spatial reference signal is generally not guaranteed. The spatial tracking problem has also been

Y. Chen, B. Chu and C. T. Freeman are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. ({Yiyang.Chen, b.chu, ctf1 }@soton.ac.uk).

\*The corresponding author (e-mail: b.chu@soton.ac.uk).

considered using ILC. It was first considered in the context of a 2D corner tracking problem in [48], then in [49] to the torque ripple minimization problem in switch reluctance motors, and most recently in [50], [51] for robotic manipulators within a 2D plane, in [52] for a class of nonlinear systems, and in [53] for additive manufacturing. However, all existing implementations of spatial ILC are application specific, and none has proposed a general spatial ILC framework applicable to a broad system class. In addition, all focus on purely achieving the path tracking, and have not fully harnessed the design freedom of spatial ILC to optimize a general additional cost function while following the defined path. As will be seen, the proposed generalized ILC design provides a solution to address these limitations.

The main contributions of the paper are as follows:

- *Rigorous Formulation of the Generalized ILC Design Problem (Section II)*: The design problem is formulated rigorously incorporating the intermediate point and sub-interval tracking requirements, and the constraints on the system input and output. The problem formulation is based on an abstract operator form in Hilbert space and illustrated using a linear discrete time invariant system, which allows the essence of the results to be generalized to other system models without difficulty.
- *Development of a Generalized ILC Algorithm (Sections III, IV and V)*: A generalized ILC algorithm is developed to solve the design problem using a successive projection method. The proposed algorithm has well-defined convergence properties meeting tracking requirements and satisfying the system constraints under mild conditions. Convergence properties and implementation procedures of the proposed algorithm are described in detail.
- *Application to Spatial Reference Tracking Problem (Section VI)*: To illustrate the power of the proposed design method, it is applied to solve a spatial reference tracking problem. It is shown that the spatial reference tracking requirement can be formulated into the generalized ILC framework by defining constraints on the system output. The generalized ILC algorithm developed above can then be applied to track the desired spatial reference. The convergence properties and computational aspects of the solution are discussed.
- *Experimental Verification on a Gantry Robot Test Platform (Section VII)*: The proposed design method is verified experimentally on a gantry robot test platform for the spatial reference tracking problem. The results show that the proposed design achieves accurate tracking of the reference, at the same time satisfying the system constraints. In addition, by exploiting the flexibility in choosing the tracking time allocation along each sub-interval, significant benefit can be obtained in terms of the control effort reduction, simultaneously maintaining high tracking performance. The results also show that the proposed algorithm exhibits a degree of robustness against modeling mismatch/error due to the use of previous data, which is clearly desirable in practical applications.

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;  $l_2^\ell[a, b]$  denotes the space of  $\mathbb{R}^\ell$  valued Lebesgue square-summable sequences defined on an interval  $[a, b]$ ;  $\langle x, y \rangle$  is the inner product of  $x$  and  $y$  in some Hilbert space;  $\mathbb{X} \times \mathbb{Y}$  is the Cartesian product of two spaces  $\mathbb{X}$  and  $\mathbb{Y}$ ;  $P_\Theta(x)$  denotes the projection of  $x$  to the set  $\Theta$  in some Hilbert space;  $\text{Im } M$  and  $\text{Ker } M$  are the image and kernel of an operator  $M$ , respectively. Other notation will be introduced as needed.

## II. PROBLEM FORMULATION

This section introduces the system dynamics and defines the general tracking requirement, together with input and output constraints to yield a generalized ILC problem formulation.

### A. System Dynamics

Consider an  $\ell$ -input,  $m$ -output discrete linear time-invariant system given in state space form

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

where  $k \in \mathbb{N}$  denotes the trial number;  $t \in [0, N]$  is the time index (e.g. sample number) with  $0 < N < \infty$  the 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 on the  $k^{\text{th}}$  trial 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.  $x_k(0) = x_0$ . To facilitate later problem formulation and algorithm design, the system is represented in an equivalent operator form

$$y_k = Gu_k + d, \quad (2)$$

where  $y_k \in l_2^m[0, N]$ ,  $u_k \in l_2^\ell[0, N]$  represent the system output and input, respectively; the input and output Hilbert spaces  $l_2^\ell[0, N]$  and  $l_2^m[0, N]$  are defined with inner products and associated induced norms

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

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

in which  $R \in \mathbb{S}_{++}^\ell$  and  $S \in \mathbb{S}_{++}^m$ ;  $G : l_2^\ell[0, N] \rightarrow l_2^m[0, N]$  is the system operator and  $d \in l_2^m[0, N]$  represents the effect of initial condition taking the following forms

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

Without loss of generality, it is assumed that  $d = 0$  by incorporating it into the reference (see [22] for more detail).

## B. Generalized Control Design Objective

A generalized control design objective is now described. In classical ILC the system is required to repeatedly track a desired reference defined on the whole horizon; in intermediate point ILC the system is required to track a given reference defined on finite intermediate points. The general control design objective subsumes both the intermediate point tracking requirement at time instants  $t_i$ ,  $i = 1, \dots, M$ , where

$$0 < t_1 < \dots < t_M = N, \quad (6)$$

and the linear tracking requirement at each sub-interval  $(t_{i-1}, t_i]$ ,  $i = 1, \dots, M$ , where  $t_0 = 0$  is used for ease of notation. To extract the intermediate point and sub-interval tracking requirements, a linear mapping is defined as

$$\zeta \in l_2^m[0, N] \mapsto \zeta^e \in H : \zeta^e = \begin{bmatrix} F\zeta \\ P\zeta \end{bmatrix}, \quad (7)$$

where  $H$  is the Hilbert space defined as

$$H = \mathbb{R}^{f_1} \times \dots \times \mathbb{R}^{f_M} \times l_2^{p_1}(t_0, t_1] \times \dots \times l_2^{p_M}(t_{M-1}, t_M]$$

with inner product and associated induced norm

$$\begin{aligned} \langle (\omega, \nu), (\mu, \lambda) \rangle_{\tilde{Q}} &= \sum_{i=1}^M \{ \omega_i^T Q_i \mu_i + \sum_{j=t_{i-1}}^{t_i} \nu_i^T(j) \hat{Q}_i \lambda_i(j) \}, \\ \|(\omega, \nu)\|_{\tilde{Q}} &= \sqrt{\langle (\omega, \nu), (\omega, \nu) \rangle_{\tilde{Q}}}. \end{aligned} \quad (8)$$

Note that  $(\omega, \nu)$ ,  $(\mu, \lambda) \in H$  have the following forms

$$\begin{aligned} \omega &= (\omega_1, \omega_2, \dots, \omega_M), \quad \mu = (\mu_1, \mu_2, \dots, \mu_M), \\ \nu &= (\nu_1, \nu_2, \dots, \nu_M), \quad \lambda = (\lambda_1, \lambda_2, \dots, \lambda_M), \end{aligned} \quad (9)$$

where  $\omega_i, \mu_i \in \mathbb{R}^{f_i}$ ,  $\nu_i, \lambda_i \in l_2^{p_i}[t_{i-1}, t_i]$ ,  $i = 1, \dots, M$  and  $\tilde{Q}$  denotes the data set  $\{Q_1, \dots, Q_M, \hat{Q}_1, \dots, \hat{Q}_M\}$  in which  $Q_i \in \mathbb{S}_{++}^{f_i}$ ,  $\hat{Q}_i \in \mathbb{S}_{++}^{p_i}$ , for  $i = 1, \dots, M$ .

In the above mapping, operator  $F$  selects the important elements or linear combination of elements of  $\zeta$  at the intermediate time instants  $t_i$ ,  $i = 1, \dots, M$ , and is defined as

$$F\zeta = \begin{bmatrix} F_1\zeta(t_1) \\ \vdots \\ F_M\zeta(t_M) \end{bmatrix}, \quad F_i\zeta(t_i) \in \mathbb{R}^{f_i}, \quad i = 0, \dots, M, \quad (10)$$

with  $F_i \in \mathbb{R}^{f_i \times m}$  a full row rank matrix for  $i = 1, \dots, M$ . Operator  $P$  extracts a linear combination of elements of  $\zeta$  at each sub-interval  $(t_{i-1}, t_i]$ ,  $i = 1, \dots, M$ , as follows

$$P\zeta = \begin{bmatrix} (P\zeta)_1 \\ \vdots \\ (P\zeta)_M \end{bmatrix}, \quad (11)$$

where the component  $(P\zeta)_i \in l_2^{p_i}(t_{i-1}, t_i]$  is defined as

$$(P\zeta)_i(t) = P_i\zeta(t), \quad t \in (t_{i-1}, t_i],$$

in which  $P_i \in \mathbb{R}^{p_i \times m}$  is a full row rank matrix for  $i = 1, \dots, M$ .

From definitions (10) and (11), it follows that the ‘extended output’  $y^e$  comprises a subset of outputs at distinct intermediate points, together with a subset of plant outputs defined over sub-intervals of the task duration. The dynamic relationship

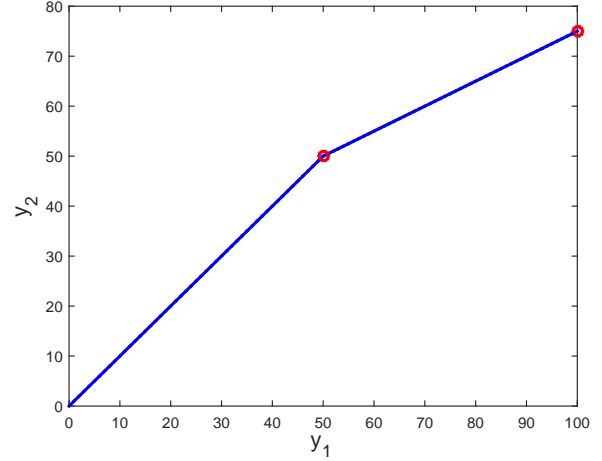


Fig. 1. Exemplary Output  $y$  in  $\mathbb{R}^2$ .

between the system input  $u$  and the extended output  $y^e$  can therefore be modelled by

$$y^e = G^e u = (Gu)^e = \begin{bmatrix} FG u \\ PG u \end{bmatrix}, \quad (12)$$

where  $G^e : l_2^\ell[0, N] \rightarrow H$  is a linear operator. For the generalized control design, the system is required to meet the tracking requirement that the system extended output  $y^e$  (repeatedly) follows a desired reference  $r^e \in H$ , i.e.  $y^e = r^e$ .

To illustrate the relation between  $y$  and  $y^e$ , the output signal represented by the blue line in Figure 1 is considered as an example. This output  $y$  is defined in 2D space ( $m = 2$ ) on the time interval  $[0, 10]$  and has two intermediate time instants ( $M = 2$ ) given by 5 and 10 corresponding to the red circles in the figure. When the matrices are chosen as  $F_1 = F_2 = I$ ,  $P_1 = [1, -1]$  and  $P_2 = [1, -2]$ , it follows that the  $FGu$  component of the extended output  $y^e$  is  $[50, 50, 100, 75]^T$  (i.e. the coordinates of the two red circles), and its  $PGu$  component is expressed as  $(PGu)_1(t) = 0$ , for  $t \in (0, 5]$  and  $(PGu)_2(t) = -50$ , for  $t \in (5, 10]$  (i.e. the line segments connecting the origin and the two red circles).

## C. Input and Output Constraints

In practice, input and output constraints exist widely in control systems due to physical limitations or performance requirements. For example, the input constraint set  $\Omega$  typically assumes one of the following forms:

- Input saturation constraint

$$\Omega = \{u \in l_2^\ell[0, N] : |u(t)| \preceq \mathcal{M}(t), 0 \leq t \leq N\}, \quad (13)$$

where  $\mathcal{M}(t) \geq 0$ ,  $0 \leq t \leq N$  are the (possibly time varying) saturation limits,

- Input effort constraint

$$\Omega = \{u \in l_2^\ell[0, N] : \sum_0^N u^\top(t)u(t) \leq \mathcal{M}\}, \quad (14)$$

where  $\mathcal{M} > 0$  is the total control effort limit,

- Input rate constraint

$$\Omega = \{u \in l_2^\ell[0, N] : |\Delta u(t)| \preceq \mathcal{M}(t), 1 \leq t \leq N\}, \quad (15)$$

where  $\Delta u(t) = u(t) - u(t-1)$  and  $\mathcal{M}(t) \succeq 0$ ,  $1 \leq t \leq N$  are the (possibly time varying) input rate limits. Similarly, the output constraint set  $\Phi$  usually has one of the forms:

- Output saturation constraint

$$\Phi = \{y \in l_2^m[0, N] : |y(t)| \preceq \mathcal{N}(t), 0 \leq t \leq N\}, \quad (16)$$

where  $\mathcal{N}(t) > 0$  represents the output saturation limit, or

- Output polyhedral constraint

$$\Phi = \{y \in l_2^m[0, N] : a_i^\top y(t) \leq b_i, a_i \in \mathbb{R}^m, b_i \in \mathbb{R}, i = 1, \dots, M, 0 \leq t \leq N\}. \quad (17)$$

In particular, the latter constraint restricts the system output to a specified convex region, and can be used to formulate the spatial reference tracking problem as described previously in the introduction.

#### D. Generalized ILC Design Problem

Using the extended output (12) combined with the above constraints, we are now in a position to state the generalized ILC design problem. This problem is defined as follows:

**Definition 1. (Generalized ILC design problem):** The generalized ILC design problem is to find an input update law based on a function of the previous trial's input and tracking error of the form

$$u_{k+1} = \mathcal{F}(u_k, e_k^e), \quad (18)$$

where  $e_k^e = r^e - y_k^e$  is the extended tracking error, such that the tracking error converges to zero as  $k \rightarrow \infty$ , i.e.

$$\lim_{k \rightarrow \infty} e_k^e = 0$$

and the converged input and output satisfy the constraints, i.e.

$$\lim_{k \rightarrow \infty} u_k = u^* \in \Omega, \quad \lim_{k \rightarrow \infty} y_k = y^* \in \Phi. \quad (19)$$

Note that the generalized ILC design problem collapses to specific ILC design problems by setting appropriate values of parameters  $Q$ ,  $\hat{Q}$ ,  $F$  and  $P$ , e.g.  $Q_i = 0$ ,  $P_i = I$ , classical ILC;  $\hat{Q}_i = 0$ ,  $F_i = I$ , intermediate point ILC.

### III. GENERALIZED ILC USING SUCCESSIVE PROJECTION

In this section, the above generalized ILC design problem is formulated using the successive projection framework which was used previously to derive classical and intermediate point ILC algorithms [22], [32]. Based on this formulation, a novel ILC algorithm is proposed to solve this generalized problem.

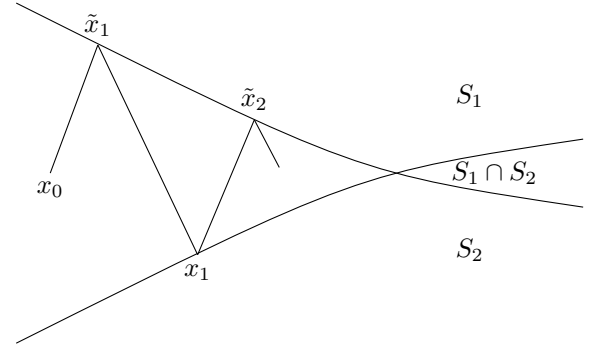


Fig. 2. Illustration of the Successive Projection Algorithm.

#### A. Successive Projection Interpretation

The design objective of the generalized ILC problem is to iteratively find an input  $u^*$  such that i) the extended output  $y^{e*} = G^e u^*$  tracks the desired reference  $r^e$ , i.e.  $y^{e*} = r^e$ , ii) the output  $y^* = G u^*$  satisfies the constraint  $y^* \in \Phi$  and iii) the input  $u^*$  meets the constraint requirement  $u^* \in \Omega$ . This is equivalent to iteratively finding a point  $(y^{e*}, y^*, u^*)$  in the intersection of the two following convex sets:

$$S_1 = \{(y^e, y, u) \in \hat{H} : y^e = G^e u, y = G u\}, \quad (20)$$

$$S_2 = \{(y^e, y, u) \in \hat{H} : y^e = r^e, y \in \Phi, u \in \Omega\}, \quad (21)$$

where the set  $S_1$  represents the plant dynamics and  $S_2$  represents the tracking requirements and system constraints;  $\hat{H}$  is the Hilbert space defined by

$$\hat{H} = \mathbb{R}^{f_1} \times \dots \times \mathbb{R}^{f_M} \times l_2^{p_1}(t_0, t_1) \times \dots \times l_2^{p_M}(t_{M-1}, t_M) \times l_2^m[0, N] \times l_2^\ell[0, N], \quad (22)$$

whose inner product and associated induced norm are derived naturally from (3), (4) and (8).

The problem of finding a point in the intersection of two sets can be solved by the method of successive projection. The basic successive projection scheme from [54] is shown in Figure 2 with guaranteed convergence performance as shown in the following theorem.

**Theorem 1.** [54], [55] Let  $S_1$  and  $S_2$  be two closed convex sets in a Hilbert space  $X$ . Define projection operators  $P_{S_1}(\cdot)$  and  $P_{S_2}(\cdot)$  as

$$P_{S_1}(x) = \arg \min_{\hat{x} \in S_1} \|\hat{x} - x\|_X^2, \quad (23)$$

$$P_{S_2}(x) = \arg \min_{\hat{x} \in S_2} \|\hat{x} - x\|_X^2, \quad (24)$$

where  $\|\cdot\|$  is the induced norm in  $X$ . Then given the initial estimate  $x_0 \in X$ , the sequences  $\{\tilde{x}_k\}_{k \geq 0}$  and  $\{x_k\}_{k \geq 0}$  generated by

$$\tilde{x}_{k+1} = P_{S_1}(x_k), \quad x_{k+1} = P_{S_2}(\tilde{x}_{k+1}), \quad k \geq 0 \quad (25)$$

are uniquely defined for each  $x_0 \in X$  and satisfy the following monotonic convergence conditions

$$\|\tilde{x}_{k+2} - x_{k+1}\|_X^2 \leq \|\tilde{x}_{k+1} - x_k\|_X^2. \quad (26)$$

If either set is compact or finite dimensional, the sequences  $\{\tilde{x}_k\}_{k \geq 0}$  and  $\{x_k\}_{k \geq 0}$  converge to fixed points  $\tilde{x}^* \in S_1$  and  $x^* \in S_2$ , i.e.

$$\lim_{k \rightarrow \infty} \tilde{x}_k = \tilde{x}^*, \quad \lim_{k \rightarrow \infty} x_k = x^* \quad (27)$$

defining the minimum distance between two sets, i.e.

$$\|\tilde{x}^* - x^*\|_X^2 = \min_{\tilde{x} \in S_1, x \in S_2} \|\tilde{x} - x\|_X^2. \quad (28)$$

Furthermore, if  $S_1 \cap S_2 \neq \emptyset$ , the following convergence condition is satisfied

$$\|x_{k+1} - x\|_X^2 \leq \|x_k - x\|_X^2, \quad \forall x \in S_1 \cap S_2, \quad k \geq 0. \quad (29)$$

*Proof.* See [54], [55] for the detailed proof.  $\square$

### B. Generalized ILC Algorithm

Direct application of Theorem 1 to the generalized ILC design problem (19) with  $X = \hat{H}$  and  $S_1, S_2$  defined in (20) and (21) yields the next algorithm.

*Algorithm 1.* Given system dynamics (1), input constraint set  $\Omega$ , output constraint set  $\Phi$ , extended reference  $r^e$ , any initial values  $u_0 \in \Omega$  and  $\tilde{r}_0 \in l_2^\ell[0, N]$ , the input sequence  $\{u_k\}_{k \geq 0}$  defined by the updating law

$$\tilde{u}_{k+1} = u_k + G^{s*}(I + G^s G^{s*})^{-1} e_k^s \quad (30)$$

followed by the projections

$$u_{k+1} = P_\Omega(\tilde{u}_{k+1}) = \arg \min_{z \in \Omega} \|z - \tilde{u}_{k+1}\|_S^2, \quad (31)$$

$$\tilde{r}_{k+1} = P_\Phi(\tilde{y}_{k+1}) = \arg \min_{z \in \Phi} \|z - \tilde{y}_{k+1}\|_R^2, \quad (32)$$

iteratively solves the generalized ILC problem (19). Note that  $G^{s*}$  is a linear operator defined by

$$G^s u = \begin{bmatrix} G^e u \\ G u \end{bmatrix} : l_2^\ell[0, N] \rightarrow \tilde{H}, \quad (33)$$

whose Hilbert adjoint operator is  $G^{s*}$ ,  $e_k^s$  is the error

$$e_k^s = \begin{bmatrix} e_k^e \\ \tilde{e}_k \end{bmatrix}, \quad e_k^e = r^e - y_k^e, \quad \tilde{e}_k = \tilde{r}_k - y_k, \quad (34)$$

and  $\tilde{H}$  is the Hilbert space

$$\tilde{H} = \mathbb{R}^{f_1} \times \dots \times \mathbb{R}^{f_M} \times l_2^{p_1}(t_0, t_1) \times \dots \times l_2^{p_M}(t_{M-1}, t_M) \times l_2^m[0, N], \quad (35)$$

whose inner product and associated induced norm are naturally derived from (4) and (8).

*Proof.* See Appendix A.  $\square$

## IV. CONVERGENCE PROPERTIES OF THE ALGORITHM

When  $S_1 \cap S_2 \neq \emptyset$ , perfect tracking of the reference is possible. Algorithm 1 iteratively solves the generalized ILC design problem (19) with desirable convergence properties as shown in the next theorem.

*Theorem 2.* If  $S_1 \cap S_2 \neq \emptyset$ , perfect tracking of the reference is possible. In this case Algorithm 1 achieves perfect tracking of the extended reference, i.e.

$$\lim_{k \rightarrow \infty} y_k^e = r^e. \quad (36)$$

In addition, the input  $u_k$  and output  $y_k$  converge as

$$\lim_{k \rightarrow \infty} u_k = u^*, \quad \lim_{k \rightarrow \infty} y_k = y^*, \quad (37)$$

and satisfy the system constraints that  $u_k \in \Omega$ ,  $\forall k \geq 0$ ,  $y^* \in \Phi$ . Furthermore, the input  $u_k$  converges monotonically with respect to the cost function

$$\tilde{J}_k = \|\tilde{r}_k - y^*\|_S^2 + \|u_k - u^*\|_R^2, \quad (38)$$

and the error  $e_k^s$  converges monotonically with respect to the cost function

$$J_k = \|\mathcal{M} e_k^s\|_{[Q]}^2 + \|\mathcal{N} e_k^s\|_R^2, \quad (39)$$

where  $\mathcal{M} = (I + G^{s*} G^s)^{-1}$ ,  $\mathcal{N} = G^{s*}(I + G^{s*} G^s)^{-1}$  and  $[Q] = \{\tilde{Q}, S\}$ .

*Proof.* See Appendix B.  $\square$

The above theorem shows that the proposed algorithm solves the generalized ILC design problem, i.e. perfect tracking is achieved and that the converged input and output satisfy the system constraints. Moreover, this convergence has a specific form of monotonicity with respect to the performance defined above, e.g. weighted error norm as in (39), which is appealing in practice. Furthermore, as a by product, when there are no constraints, the algorithm can be simplified and has the property that it will converge to a minimum norm solution with zero initial input, i.e. the control input with minimum control effort is achieved, as shown in the following corollary.

*Corollary 1.* If  $S_1 \cap S_2 \neq \emptyset$ , in the absence of system constraints, Algorithm 1 control updating law has the following simplified form

$$u_{k+1} = u_k + G^{e*}(I + G^e G^{e*})^{-1} e_k^e,$$

and the algorithm converges monotonically with respect to the performance index

$$\hat{J}_k = \langle u_k - u^*, \mathcal{H}(u_k - u^*) \rangle_X \quad (40)$$

for all  $k \geq 0$ ,  $u_0$  and  $u^*$ , where  $\mathcal{H} = G^{e*} G + I$ , and  $G^{e*}$  is the Hilbert adjoint operator of  $G^e$ . Moreover, the control effort converges to the minimum value with initial condition  $u_0 = 0$ , i.e.

$$\lim_{k \rightarrow \infty} \|u_k\|_R^2 = \min_u \{\|u\|_R^2, \text{ s.t. } r^e = G^e u\}. \quad (41)$$

*Proof.* See Appendix C.  $\square$

On the other hand, perfect tracking of the reference is not possible when  $S_1 \cap S_2 = \emptyset$ . No input exists to satisfy the tracking requirement without violating the system constraints. In this case, this algorithm still attempts to solve the constrained generalized ILC design, as shown in the next theorem.

*Theorem 3.* If  $S_1 \cap S_2 = \emptyset$ , perfect tracking of the reference under the system constraints is not possible. In this case, Algorithm 1 still achieves convergence of the input  $u_k$ , output  $y_k$  and extended output  $y_k^e$  as follows

$$\lim_{k \rightarrow \infty} y_k^e = y^{e*}, \quad \lim_{k \rightarrow \infty} y_k = y^*, \quad \lim_{k \rightarrow \infty} u_k = u^*, \quad (42)$$

where

$$u^* = \arg \min_{u \in \Omega} J_k = \arg \min_{u \in \Omega} \|\mathcal{M}e^s\|_{[Q]}^2 + \|\mathcal{N}e^s\|_R^2, \quad (43)$$

and

$$y^{e*} = G^e u^*, \quad y^* = G u^*$$

in which  $\mathcal{M} = (I + G^{s*} G^s)^{-1}$ ,  $\mathcal{N} = G^{s*} (I + G^{s*} G^s)^{-1}$  and  $[Q] = \{\hat{Q}, S\}$ . In addition, the input  $u_k$  at each trial satisfies the constraint that  $u_k \in \Omega$ . Furthermore, the error  $e_k^s$  converges monotonically with respect to the cost function  $J_k$  defined in (43).

*Proof.* See Appendix D.  $\square$

*Remark 1.* It is worth pointing out that the scenario in the above theorem is not well-posed. In practice an appropriate tracking requirement  $r^e$  should be specified to avoid this impossible tracking task, i.e.  $S_1 \cap S_2 = \emptyset$ .

## V. IMPLEMENTATION OF THE ALGORITHM

Implementation of Algorithm 1 consists of two steps: the ILC update (30) and projection steps (31), (32). The update (30) can be directly implemented as a feedforward solution using  $u_k$  and  $e_k^s$  to construct  $\tilde{u}_{k+1}$ . Alternatively, it can be implemented using a causal feedback plus feedforward structure by employing the state feedback to further embed robust performance in practice. This exploits the special properties of the linear operator  $G^e$  and its adjoint operator  $G^{e*}$ . To formulate the causal feedback plus feedforward solution, the following lemma is needed.

*Lemma 1.* The Hilbert adjoint operator  $G^{s*} : (\omega, \nu, y) \in \tilde{H} \rightarrow u \in l_2^s[0, N]$  has the following analytic form

$$u(t) = R^{-1} B^\top p(t), \quad (44)$$

where  $p(t)$  is computed in reverse time as follows

$$p(t) = A^\top p(t+1) + C^\top (P_i^\top \hat{Q}_i \nu_i(t+1) + S y(t+1)), \\ t \in [t_{i-1}, t_i), \quad i = 1, \dots, M, \quad (45)$$

with boundary conditions

$$p(t_i - 1) = p(t_i) + C^\top F_i^\top Q_i \omega_i, \quad i = 1, \dots, M, \\ p(N) = 0. \quad (46)$$

*Proof.* The proof follows that of [56] with modifications applied to the extended operator in Hilbert space.  $\square$

Using Lemma 1, the feedback plus feedforward implementation is given in the next proposition.

*Proposition 1.* The ILC update (30) in Algorithm 1 can be implemented in a feedforward plus feedback solution

$$u_{k+1}(t) = u_k(t) + R^{-1} B^\top p_{k+1}(t), \quad (47)$$

with

$$p_{k+1}(t) = -K(t)(I + BR^{-1}B^\top K(t))^{-1} A \\ (x_{k+1}(t) - x_k(t)) + \xi_{k+1}(t), \quad (48)$$

where  $K(t)$  is the solution of the Riccati equation

$$K(t) = A^\top K(t+1)(I + BR^{-1}B^\top K(t+1))^{-1} A \\ + C^\top (S + \hat{Q}(t+1))C, \quad (49)$$

with boundary conditions

$$K(t_i - 1) = K(t_i) + C^\top F_i^\top Q_i F_i C, \quad i = 1, \dots, M, \\ K(N) = 0, \quad (50)$$

and  $\xi_{k+1}(t)$  denotes the feedforward term at the  $(k+1)^{th}$  trial generated by the difference equation

$$\xi_{k+1}(t) = (I + K(t)BR^{-1}B^\top)^{-1} (A^\top \xi_{k+1}(t+1) \\ + C^\top \hat{Q}(t+1)e_k(t+1) + C^\top S \tilde{e}_k(t+1)), \quad (51)$$

with boundary conditions

$$\xi_{k+1}(t_i - 1) = \xi_{k+1}(t_i) + C^\top F_i^\top Q_i F_i e_k(t_i), \\ \xi_{k+1}(N) = 0, \quad (52)$$

in which  $\hat{Q}(t) = P_i^\top \hat{Q}_i P_i$  for  $t \in (t_{i-1}, t_i]$ ,  $i = 1, \dots, M$ .

*Proof.* See Appendix E.  $\square$

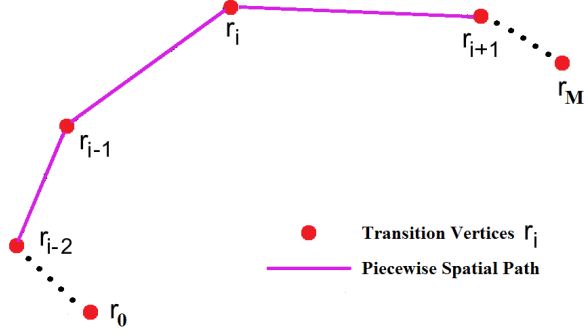
The ILC update (30) is then followed by the projection steps in (31) and (32) to project the unconstrained input  $\tilde{u}_{k+1}$  and output  $\tilde{y}_{k+1}$  into the input and output constraint sets  $\Phi$  and  $\Omega$  respectively which are usually straightforward. For example, the input constraint set  $\Omega$  is usually a pointwise constraint in practice, so the solution of the projection operator  $P_\Omega$  in (31) is straightforward. As another example, when the input constraint set  $\Omega$  has the saturation form (13), the solution of  $u = P_\Omega(\tilde{u})$  is given by

$$u(t) = \begin{cases} \mathcal{M}(t), & \tilde{u}(t) \succ \mathcal{M}(t), \\ \tilde{u}(t), & -\mathcal{M}(t) \preceq \tilde{u}(t) \preceq \mathcal{M}(t), \\ -\mathcal{M}(t), & \tilde{u}(t) \prec -\mathcal{M}(t) \end{cases} \quad (53)$$

for  $0 \leq t \leq N$ . Also the solution of the projection operator  $P_\Phi$  in (32) is guaranteed to be unique. Consider the saturation output constraint set (16) for example, the solution of  $\tilde{r} = P_\Omega(\tilde{y})$  is given by

$$\tilde{r}(t) = \begin{cases} \mathcal{N}(t), & \tilde{y}(t) \succ \mathcal{N}(t), \\ \tilde{y}(t), & -\mathcal{N}(t) \preceq \tilde{y}(t) \preceq \mathcal{N}(t), \\ -\mathcal{N}(t), & \tilde{y}(t) \prec -\mathcal{N}(t), \end{cases} \quad (54)$$

for  $0 \leq t \leq N$ .


 Fig. 3. Piecewise Spatial Path in  $\mathbb{R}^2$ .

## VI. PIECEWISE LINEAR SPATIAL PATH TRACKING

As described in the introduction, the spatial tracking problem requires the output trajectory to follow a continuous path defined in space with no temporal constraints. The framework developed in this paper can be applied to produce the first spatial ILC algorithm capable of converging to an optimal solution to the problem for a general class of systems. We consider the common class of piecewise spatial tracking tasks defined as follows:

*Definition 2.* The piecewise linear spatial tracking problem is to design an input such that the output travels between each pair of vertices  $r_{i-1}$  and  $r_i$  in ascending order, i.e. having reached  $r_{i-1}$  at  $t_{i-1}$ , remain in the interval between  $r_{i-1}$  and  $r_i$  until  $r_i$  is reached at  $t_i$  and then repeat the process for the next pair  $r_i$  and  $r_{i+1}$ .

Note that the vertices  $r_i$ ,  $i = 0, \dots, M$  are defined in Cartesian space  $\mathbb{R}^m$  as  $r_i = [r_i^1, r_i^2, \dots, r_i^m]^\top$ , and a special example of the piecewise spatial path, i.e.  $m = 2$ , is shown in Figure 3.

The next theorem illustrates how Algorithm 1 solves the spatial problem of Definition 2 as a special case.

*Theorem 4.* The piecewise linear spatial path tracking problem is solved by Algorithm 1 with

$$r^e = \begin{bmatrix} Fr \\ Pr \end{bmatrix}, \quad (55)$$

where  $r \in l_2^m[0, N]$  is given by

$$r(t) = r_i, \quad t \in (t_{i-1}, t_i], \quad i = 1, \dots, M, \quad (56)$$

the operator  $F$  is defined by  $F_i = I$ ,  $i = 1, \dots, M$  and the operator  $P$  by  $P_i \in R^{(m-1) \times m}$  a full rank matrix satisfying

$$\text{Ker } P_i = \text{Im } a_i, \quad (57)$$

in which  $a_i = r_i - r_{i-1}$ , for  $i = 1, \dots, M$ . The input constraint set  $\Omega$  is chosen according to the power capability of the plant, and the output constraint set is

$$\Phi = \{y \in l_2^m[0, N] : a_i^\top r_{i-1} \leq a_i^\top y(t) \leq a_i^\top r_i, \\ t \in (t_{i-1}, t_i], \quad i = 1, \dots, M\}, \quad (58)$$

which prevents overshoot beyond each linear segment of the path. Using Algorithm 1 the input  $u_k$  and output  $y_k$  both converge as

$$\lim_{k \rightarrow \infty} u_k = u^* \in \Omega, \quad \lim_{k \rightarrow \infty} y_k = y^*. \quad (59)$$

Furthermore, if  $S_1 \cap S_2 \neq \emptyset$ , the output  $y_k$  and extended output  $y_k^e$  have the following convergence properties

$$\lim_{k \rightarrow \infty} y_k = y^* \in \Phi, \quad \lim_{k \rightarrow \infty} y_k^e = r^e. \quad (60)$$

That is, perfect tracking of the spatial path is achieved.

*Proof.* See Appendix F.  $\square$

The above theorem states that the proposed Algorithm 1 solves the high performance piecewise linear spatial tracking problem. From Corollary 1, Algorithm 1 also fully exploits the spatial tracking task's inherent design freedom in the temporal domain, resulting in the spatial tracking problem being solved with minimum control effort for a general class of systems, under mild conditions when  $S_1 \cap S_2 \neq \emptyset$  and the system constraints do not engage (e.g. if overshoot can be tolerated or does not occur).

*Remark 2.* The implementation of the aforementioned spatial path tracking design is exactly the same as that discussed in previous sections, in particular the unique solution  $\tilde{r} = P_\Phi(\tilde{y})$  is given as

$$\tilde{r}(t) = \begin{cases} \tilde{y}(t) + \Delta_{i,i}(t), & a_i^\top \tilde{y}(t) > a_i^\top r_i, \\ \tilde{y}(t), & a_i^\top r_{i-1} \leq a_i^\top \tilde{y}(t) \leq a_i^\top r_i, \\ \tilde{y}(t) + \Delta_{i,i-1}(t), & a_i^\top \tilde{y}(t) < a_i^\top r_{i-1}, \end{cases} \quad (61)$$

for  $t \in (t_{i-1}, t_i]$ ,  $i = 1, \dots, M$  where

$$\Delta_{i,j}(t) = (a_i^\top a_i)^{-1} a_i^\top (r_j - \tilde{y}(t)) a_i.$$

*Remark 3.* It is worth pointing out that the spatial path tracking problem considered in this section is still time dependent as: the input depends implicitly on the transition time instants  $t_i$ ,  $i = 0, \dots, M$  and the input has to be implemented on the time interval  $[0, N]$ . A different ILC approach was developed in [53] for additive manufacturing, where no temporal dynamics was involved, but it is not suitable for spatial path tracking. Please refer to [57], [58] for more information.

## VII. EXPERIMENTAL VERIFICATION ON A GANTRY ROBOT

In this section, the proposed algorithm is validated experimentally on a three-axis gantry robot test platform to demonstrate its effectiveness.

### A. Test Platform Specification

The multi-axis gantry robot shown in Figure 4 is employed as test platform. The gantry robot's input are the voltages applied to the three motors controlling the motions of the robot over  $x, y, z$  directions, and the output are the displacements of the three axes. The control design objective is to use both the



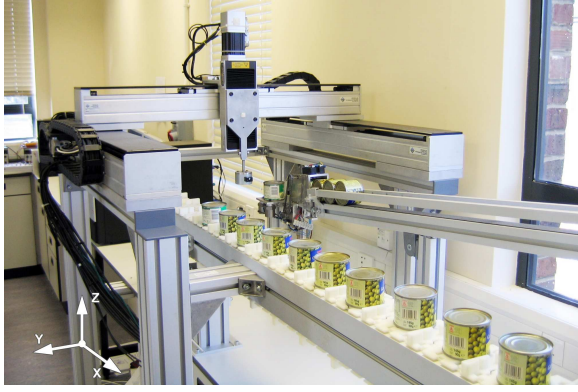


Fig. 4. Multi-axis Gantry Robot Test Platform.

x-axis and z-axis ( $m = 2$ ) to track a piecewise linear spatial path composed of five line segments ( $M = 5$ ) with

$$r_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, r_1 = \begin{bmatrix} 0.00476 \\ 0.00345 \end{bmatrix}, r_2 = \begin{bmatrix} 0.00294 \\ 0.00905 \end{bmatrix},$$

$$r_3 = \begin{bmatrix} -0.00294 \\ 0.00905 \end{bmatrix}, r_4 = \begin{bmatrix} -0.00476 \\ 0.00345 \end{bmatrix}, r_5 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

as shown in Figure 5 (the yellow line), during the given tracking time  $T = 2s$ . The tracking time instants of the intermediate points are given as

$$t_1 = 0.4, t_2 = 0.8, t_3 = 1.2, t_4 = 1.6, t_5 = 2.0. \quad (62)$$

The x-axis and z-axis are modelled based on frequency response tests in [59] with state space parameters

$$A_x = \begin{bmatrix} 1.0673 & -0.0242 & 0.0097 & -0.0047 & 0.0028 & -0.0009 & 0.0005 \\ 0.0827 & 0.999 & 0.0004 & -0.0002 & 0.0001 & 0 & 0 \\ 0.0016 & 0.04 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0.0004 & 0.02 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.0001 & 0.01 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0025 & 1 \end{bmatrix}$$

$$B_x = \begin{bmatrix} 0.403 & 0.0159 & 0.0001 & 0.0002 & 0 & 0 & 0 \end{bmatrix}^\top \times 10^{-4},$$

$$C_x = \begin{bmatrix} 0.0013 & -0.0005 & 0.0001 & 0 & -0.0001 & 0.0001 & -0.0001 \end{bmatrix},$$

$$A_z = \begin{bmatrix} 1.0291 & -0.0077 & 0.0035 & -0.0011 \\ 0.0406 & 0.998 & 0.0001 & 0 \\ 0.0002 & 0.001 & 1 & 0 \\ 0 & 0 & 0.005 & 1 \end{bmatrix}, B_z = \begin{bmatrix} 0.3957 \\ 0.0079 \\ 0 \\ 0 \end{bmatrix} \times 10^{-4},$$

$$C_z = \begin{bmatrix} 0.0026 & -0.0004 & -0.0004 & 0.0005 \end{bmatrix}, \quad (63)$$

and a proportional feedback gain 300 is added on the z-axis. The system is sampled with a zero-order hold at  $0.01s$ . The input voltage has the saturation constraint in the form (13) with  $\mathcal{M}(t) = [0.5, 4]^\top$ . The parameters  $F_i$ ,  $P_i$  and  $r(t)$  are chosen according to the values in (55). For simplicity, the weighting matrices  $Q_i$ ,  $\hat{Q}_i$ ,  $S$  and  $R$  are chosen to be diagonal.

### B. Performance of the Proposed Algorithm

Firstly, we artificially remove the system constraints (i.e. setting  $\Omega = l_2^\ell[0, N]$  and  $\Phi = l_2^m[0, N]$ ) and apply the proposed algorithm to this task on the gantry robot for 100

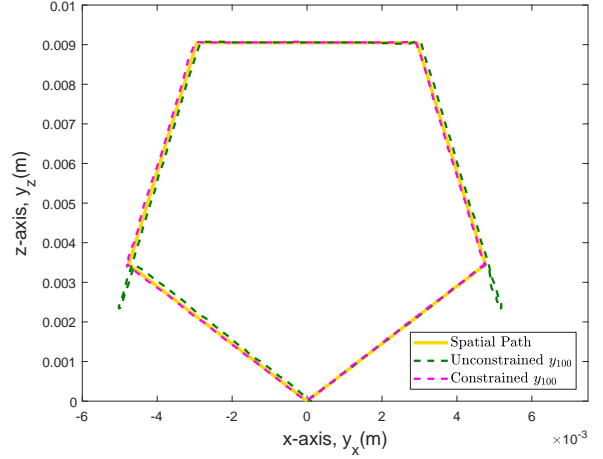


Fig. 5. Spatial Path and Converged Hybrid Output Trajectories with and without Output Constraints.

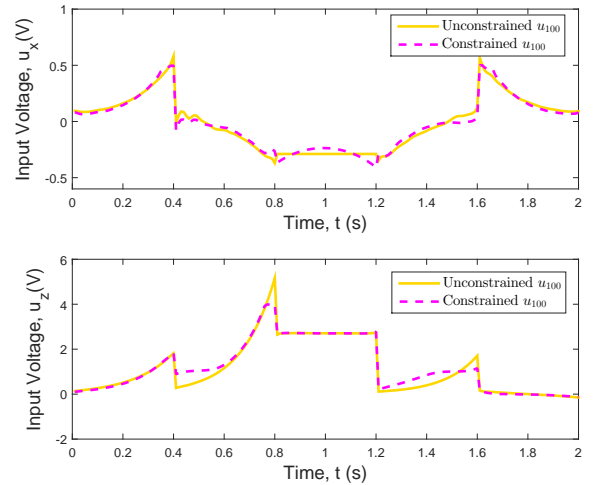


Fig. 6. Converged Input Trajectories with and without Input Constraints.

trials to see the performance for the unconstrained case. The final converged hybrid output and inputs of the two axes are plotted in Figure 5 and Figure 6 respectively. Although the converged hybrid output performs near perfect tracking along the piecewise linear reference path, it is clear that the overshoot problem takes place. In the gantry robot test platform, the overshoot problem may lead to collision between the end-effector and the frame, which causes damage to the machine. In addition, the corresponding inputs of the two axes exceeds the input constraint set  $\Omega$  defined in (58) at certain time intervals, which is allowed not in practice. Therefore, it is necessary to apply the system constraints in practice to meet the actual system requirement of the design task.

To avoid the above problems, we next apply Algorithm 1 to the same task with the input saturation constraint defined in (13) and the hard output constraint defined in (58). In the experiment, we choose  $\hat{Q}_i = 100,000I$ ,  $Q_i = 500,000I$ ,  $S = 10,000I$  and  $R = I$ , and a total of 100 update trials are performed. The final converged hybrid output of the two



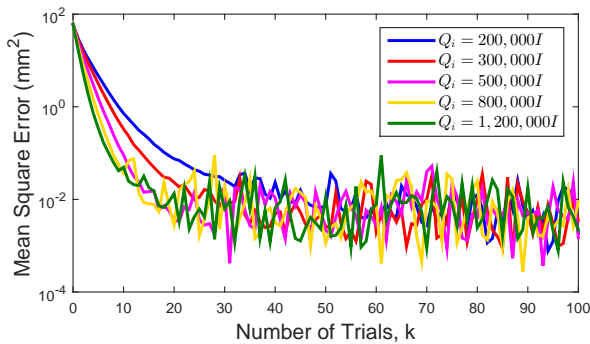


Fig. 7. Mean Square Tracking Error over 100 Trials with System Constraints.

axes is plotted as the dashed magenta trajectory in Figure 5. Compared to the previous overshoot result in the same figure, it is clear that Algorithm 1 not only achieves the generalized tracking requirement, but keeps the hybrid output trajectory within the output constraint set  $\Phi$  defined in (58), i.e. this algorithm solves the overshoot problem. Furthermore, the final converged input voltages of the two axes are plotted as the dashed magenta trajectories in Figure 6, and it is clear that both stay within the input constraint set  $\Omega$  defined in (58). Note that the inputs of both axes increase while approaching the transition vertices as more control effort is needed to change the moving direction of the end-effector. Therefore, it is clear from Figure 5 and Figure 6 that Algorithm 1 not only guarantee high performance spatial tracking, but also handles the system constraints well.

We further apply the proposed algorithm with different parameters to compare convergence properties. We keep  $\hat{Q}_i = 100,000I$ ,  $S = 10,000I$  and  $R = I$  constant, and  $Q_i$  is selected to take the values  $200,000I$ ,  $300,000I$ ,  $500,000I$ ,  $800,000I$ , and  $1,200,000I$ . A total of 100 update trials are performed for each value of  $Q_i$ , and the mean square error  $e_k^s$  at each trial is plotted in Figure 7. From this figure, it is obvious that the convergence rate increases as we increase the weighting value  $Q_i$ . It is noted that all plots converge to below 0.01 mean square error, which verifies accurate tracking in practice despite of model uncertainty and random disturbance. It is noted that there are no particular concerns about the fluctuation in the figure as the mean square errors all converge and satisfy the practical tracking requirement.

In addition, the converged control effort for different values of  $Q_i$  are shown in Table I. If performed within the classical ILC framework, the end-effector would commonly be assumed to move at a constant speed. This classical setup has been implemented, and the control effort needed is also shown in this table for comparison. The table shows that the proposed algorithm fully exploits the design freedom in the temporal domain in terms of an approximate 18% control effort reduction from that provided by classical ILC.

Experiments with other combinations of  $Q_i$ ,  $\hat{Q}_i$ ,  $S$  and  $R$  yield similar convergence performance to the results in Figure 7. For brevity, these results are omitted.

TABLE I  
CONTROL EFFORT COMPARISON.

	Control Effort ( $V^2 \cdot ms$ )
$Q_i = 200,000I$	5,462
$Q_i = 300,000I$	5,467
$Q_i = 500,000I$	5,461
$Q_i = 800,000I$	5,459
$Q_i = 1,200,000I$	5,458
Classical ILC	6,649

## VIII. CONCLUSION

This paper develops a novel unified ILC design framework capable of solving tracking requirements defined on both intermediate point and sub-intervals as well as handling a mixed form of system constraints, which substantially extends the existing ILC research. To solve this problem, the paper proposes a new ILC algorithm using successive projection method with well defined convergence properties. A particularly powerful feature of the proposed algorithm is that it can handle many different types of tracking requirements. As an example, it is applied to solve a challenging problem of high performance spatial reference tracking. The algorithm achieves high tracking accuracy as well as fully exploits the design freedom in the temporal domain to minimize some cost functions, e.g. minimum control effort. The algorithm's convergence properties have been analyzed rigorously, and then verified on a gantry robot platform by tracking a spatial path with stipulated input and output constraints, which demonstrates its practical efficacy.

The experimental results suggest that the proposed algorithm has a degree of robustness against model uncertainties. A rigorous robustness analysis of the proposed algorithm will be undertaken. In addition, the Hilbert space setting of the problem formulation means the results in this paper can be in principle extended to many other situations of interest, e.g. continuous linear state space models, linear time varying systems and switched linear systems. Some initial results on continuous linear state space models were recently reported in [60]. However, more research is needed to fully characterize its convergence properties and thoroughly validate its performance. This paper considers piecewise linear path tracking problem with prescribed transition time instants. Future work includes fully unlocking the design freedom to incorporate more ILC design freedom, e.g. the optimal tracking time allocation problem [35], curved spatial paths and more general ILC design problems. These constitute part of our future research and will be reported separately.

## APPENDIX A PROOF OF ALGORITHM 1

To apply Theorem 1 to generalized ILC problem (19), we first compute the necessary projections. From the definition of Hilbert space  $\hat{H}$  in (22), denote  $x = (y^e, y, u)$  to be an element

belonging to  $\hat{H}$ . The projection operator  $P_{S_1}$  is hence

$$\begin{aligned}
 P_{S_1}(x) &= \arg \min_{\hat{x} \in S_1} \|\hat{x} - x\|_X^2 \\
 &= \arg \min_{(\hat{y}^e, \hat{y}, \hat{u}) \in \hat{H}} \left\| (\hat{y}^e, \hat{y}, \hat{u}) - (y^e, y, u) \right\|_{\{\hat{Q}, S, R\}}^2, \\
 &\quad \text{s.t. } \hat{y}^e = G^e \hat{u}, \quad \hat{y} = G \hat{u} \\
 &= \arg \min_{(\hat{y}^e, \hat{y}, \hat{u}) \in \hat{H}} \|\hat{y}^e - y^e\|_{\hat{Q}}^2 + \|\hat{y} - y\|_S^2 + \|\hat{u} - u\|_R^2, \\
 &\quad \text{s.t. } \hat{y}^e = G^e \hat{u}, \quad \hat{y} = G \hat{u}. \\
 &= \arg \min_{\hat{u}} \|G^e \hat{u} - y^e\|_{\hat{Q}}^2 + \|G \hat{u} - y\|_S^2 + \|\hat{u} - u\|_R^2. \quad (64)
 \end{aligned}$$

The optimization problem (64) yields solution  $\hat{u} = u^*$  with

$$u^* = u + G^{s*} (I + G^s G^{s*})^{-1} \begin{bmatrix} y^e - G^e u \\ y - G u \end{bmatrix}. \quad (65)$$

It follows from the definition (20) that

$$P_{S_1}(x) = (G^e u^*, G u^*, u^*), \quad (66)$$

where  $u^*$  is given by (65). Performing a similar procedure for projection operator  $P_{S_2}$  yields

$$\begin{aligned}
 P_{S_2}(x) &= \arg \min_{\hat{x} \in S_2} \|\hat{x} - x\|_X^2 \\
 &= \arg \min_{(\hat{y}^e, \hat{y}, \hat{u}) \in \hat{H}} \left\| (\hat{y}^e, \hat{y}, \hat{u}) - (y^e, y, u) \right\|_{\{\hat{Q}, S, R\}}^2, \\
 &\quad \text{s.t. } \hat{y}^e = r^e, \quad \hat{u} \in \Omega, \quad \hat{y} \in \Phi \\
 &= \arg \min_{(\hat{y}^e, \hat{y}, \hat{u}) \in \hat{H}} \|\hat{y}^e - y^e\|_{\hat{Q}}^2 + \|\hat{y} - y\|_S^2 + \|\hat{u} - u\|_R^2, \\
 &\quad \text{s.t. } \hat{y}^e = r^e, \quad \hat{u} \in \Omega, \quad \hat{y} \in \Phi. \quad (67)
 \end{aligned}$$

In the optimization problem (67), the elements  $\hat{y}^e$ ,  $\hat{y}$  and  $\hat{u}$  are independent of each other, which means this solution can be obtained separately. Using (31) and (32), it follows that

$$P_{S_2}(x) = (r^e, P_\Phi(y), P_\Omega(u)). \quad (68)$$

Consider update (25) in Theorem 1, and also let  $x_k = (r^e, \tilde{r}_k, u_k)$  and  $\tilde{x}_k = (\tilde{y}_k^e, \tilde{y}_k, \tilde{u}_k)$ . At the  $k^{\text{th}}$  trial, the elements  $\tilde{x}_{k+1}$  and  $x_{k+1}$  are updated using projection operators  $P_{S_1}$  and  $P_{S_2}$ . For  $\tilde{x}_{k+1} = P_{S_1}(x_k)$ , it follows from the solution (66) that

$$\tilde{u}_{k+1} = u_k + G^{s*} (I + G^s G^{s*})^{-1} \begin{bmatrix} r^e - y_k^e \\ \tilde{r}_k - y_k \end{bmatrix}, \quad (69)$$

$$\tilde{y}_{k+1}^e = G^e \tilde{u}_{k+1}, \quad \tilde{y}_{k+1} = G \tilde{u}_{k+1}, \quad k \geq 0, \quad (70)$$

and for  $x_{k+1} = P_{S_2}(\tilde{x}_{k+1})$ , it follows from (68) that

$$\tilde{r}_{k+1} = P_\Phi(\tilde{y}_{k+1}), \quad u_{k+1} = P_\Omega(\tilde{u}_{k+1}), \quad k \geq 0, \quad (71)$$

which directly illustrates how the input  $u_{k+1}$  and the reference  $\tilde{r}_{k+1}$  are updated by  $P_{S_2}$ . Therefore, Theorem 1 can be applied to problem (19) with the solutions (69) and (71) to yield Algorithm 1, which updates the input sequence  $\{u_k\}_{k \geq 0}$  along the trial under the initial condition  $x_0 = (r^e, \tilde{r}_0, u_0) \in S_2$ , i.e.  $\tilde{r}_0 \in \Phi$ ,  $u_0 \in \Omega$ .

## APPENDIX B PROOF OF THEOREM 2

If  $S_1 \cap S_2 \neq \emptyset$ , there exists intersection between the two sets and perfect tracking under system constraints is possible. As both  $S_1$  and  $S_2$  are finite dimensional closed convex sets in the Hilbert space  $\hat{H}$ , it follows from equation (27) in Theorem 1 that the two sequences  $\{(\tilde{y}_k^e, \tilde{y}_k, \tilde{u}_k)\}_{k \geq 0}$  and  $\{(r^e, \tilde{r}_k, u_k)\}_{k \geq 0}$  both converge as

$$\begin{aligned}
 \lim_{k \rightarrow \infty} (\tilde{y}_k^e, \tilde{y}_k, \tilde{u}_k) &= (\tilde{y}^{e*}, \tilde{y}^*, \tilde{u}^*), \\
 \lim_{k \rightarrow \infty} (r^e, \tilde{r}_k, u_k) &= (r^e, \tilde{r}^*, u^*), \quad (72)
 \end{aligned}$$

which attain the minimum distance between the two sets, i.e.

$$\|r^e - \tilde{y}^{e*}\|_{\hat{Q}}^2 + \|\tilde{r}^* - \tilde{y}^*\|_S^2 + \|u^* - \tilde{u}^*\|_R^2 = 0. \quad (73)$$

So we obtain  $y^{e*} = r^e$  for perfect tracking and we also know that the  $y^*$  belongs to the output constraint set  $\Phi$  as  $y^* = \tilde{r}^* \in \Phi$ . Hence all these give rise to (37). In addition, the input  $u_k$  is obtained from the projection operator  $P_\Omega$  at each trial so  $u_k \in \Omega$ . Then, substitute  $\tilde{x}_k = (\tilde{y}_k^e, \tilde{y}_k, \tilde{u}_k)$  and  $x_k = (r^e, \tilde{r}_k, u_k)$  with the update solution (30) into the monotonic convergence condition (26) to give  $J_{k+1} \leq J_k$ , and substitute  $x^* = (r^e, y^*, u^*) \in S_1 \cap S_2$  and  $x_k = (r^e, \tilde{r}_k, u_k)$  into the monotonic convergence condition (29) to give  $\tilde{J}_{k+1} \leq \tilde{J}_k$ , which completes the proof.

## APPENDIX C PROOF OF COROLLARY 1

In the absence of system constraints, the sets  $S_1$  and  $S_2$  can be simplified as follows.

$$S_1 = \{(y^e, u) \in H \times l_2^\ell[0, N] : y^e = G^e u\}, \quad (74)$$

$$S_2 = \{(y^e, u) \in H \times l_2^\ell[0, N] : y^e = r^e\}. \quad (75)$$

Substitute  $x^* = (r^e, u^*)$  and  $x_k = (r^e, u_k)$  into monotonic convergence condition (29), which hence gives rise to  $\hat{J}_{k+1} \leq \hat{J}_k$ ,  $\forall k \geq 0$ . The proof of control effort follows from Theorem 1 in [41] by considering the Lagrangian associated with the minimum control effort

$$\mathcal{L}(u, \lambda) = \|u\|_R^2 + 2 \langle \lambda, r^e - G^e u \rangle_{\hat{Q}}, \quad (76)$$

where  $\lambda$  is the Lagrange multiplier. The problem has a unique stationary point  $u_\infty = G^{e*} \lambda$  and  $r^e = G^e u_\infty$  which hence leads to  $r^e = G^e G^{e*} \lambda$ . The stationary point solution solves the minimum control effort problem. In the absence of system constraints, the input  $u_k$  is

$$\begin{aligned}
 u_k &= G^{e*} \sum_{i=1}^k X^i e_0^e = G^{e*} \sum_{i=0}^{k-1} X^i (I - X) \lambda_0 \\
 &= G^{e*} (I - X^k) \lambda_0, \quad (77)
 \end{aligned}$$

where  $X = (I + G^e G^{e*})^{-1}$  and  $e_0^e = r^e = G^e G^{e*} \lambda_0$  as  $u_0 = 0$ . It follows that  $u_k$  converges in norm to an input  $\hat{u}_\infty = G^{e*} \lambda_0$  and  $r^e = G^e G^{e*} \lambda_0$ , which is the unique stationary point of the Lagrangian. It is clear that  $\hat{u}_\infty = u_\infty$  and  $\lambda_0 = \lambda$ , and hence the control effort converges to the minimum control effort. This completes the proof.

APPENDIX D  
PROOF OF THEOREM 3

If  $S_1 \cap S_2 = \emptyset$ , perfect tracking is not possible under system constraints. As both  $S_1$  and  $S_2$  are finite dimensional closed convex sets in the Hilbert space  $\hat{H}$ , it also follows Theorem 1 that the two sequences  $\{(\tilde{y}_k^e, \tilde{y}_k, \tilde{u}_k)\}_{k \geq 0}$  and  $\{(r^e, \tilde{r}_k, u_k)\}_{k \geq 0}$  still converge as shown in (72) yielding (42), and attain the minimum distance between the two sets, i.e.

$$\min_{\tilde{u}, u \in \Omega, \tilde{r} \in \Phi} \|r^e - G^e \tilde{u}\|_Q^2 + \|\tilde{r} - G\tilde{u}\|_S^2 + \|u - \tilde{u}\|_R^2, \quad (78)$$

which can be further written as

$$\min_{u \in \Omega, \tilde{r} \in \Phi} \left\{ \min_{\tilde{u}} \|r^e - G^e \tilde{u}\|_Q^2 + \|\tilde{r} - G\tilde{u}\|_S^2 + \|u - \tilde{u}\|_R^2 \right\}. \quad (79)$$

The inner optimization problem in (79) yields

$$\tilde{u} = u + \mathcal{N}e^s, \quad (80)$$

and hence the solution (43) for  $u^*$  is obtained by substituting (80) into (79). Also, the input  $u_k$  at each trial belongs to the input constraint set  $\Omega$  as it is obtained from the projection operator  $P_\Omega$ . The proof of the monotonic convergence with respect to  $J_k$  follows from the similar proof in Theorem 2. The proof is now complete.

APPENDIX E  
PROOF OF PROPOSITION 1

The ILC update (30) is equivalent to

$$u_{k+1}(t) = u_k(t) + G^{s*} \tilde{e}_{k+1}^s(t), \quad (81)$$

where  $\tilde{e}_{k+1}^s = [e_{k+1}^e, \hat{e}_{k+1}]^\top$  and  $\hat{e}_{k+1} = \tilde{r}_k - y_{k+1}$ . From Lemma 1,  $G^{s*} \tilde{e}_{k+1}^s(t)$  can be computed using

$$G^{s*} \tilde{e}_{k+1}^s(t) = R^{-1} B^\top p_{k+1}(t), \quad (82)$$

where the costate  $p_k(t)$  is computed in reverse time as

$$p_{k+1}(t) = -A^\top p_{k+1}(t+1) - C^\top (P_i^\top \hat{Q}_i P_i e_{k+1}(t+1) + S \hat{e}_{k+1}(t+1)), \quad t \in [t_{i-1}, t_i], \quad i = 1, \dots, M, \quad (83)$$

with boundary conditions

$$p_{k+1}(t_i - 1) = p_{k+1}(t_i) + C^\top F_i^\top Q_i F_i e_{k+1}(t_i), \\ p_{k+1}(N) = 0. \quad (84)$$

Substituting (82) into (81) yields the solution (47).

Assuming full state knowledge, the costate equation (82) yields a causal implementation

$$p_{k+1}(t) = -K(t)(I + BR^{-1}B^\top K(t))^{-1}A \\ (x_{k+1}(t) - x_k(t)) + \xi_{k+1}(t). \quad (85)$$

Then use the method proposed in [61] such that it follows from (1), (47) and (85) that

$$x_{k+1}(t+1) - x_k(t+1) \\ = A(x_{k+1}(t) - x_k(t)) + B(u_{k+1}(t) - u_k(t)) \\ = A(x_{k+1}(t) - x_k(t)) + BR^{-1}B^\top p_{k+1}(t) \\ = (I + BR^{-1}B^\top K(t))^{-1}A(x_{k+1}(t) - x_k(t)) \\ + BR^{-1}B^\top \xi_{k+1}(t). \quad (86)$$

Then substitute (85) and (86) into the costate equation (82) to yield an equation of the form

$$\mathcal{H}(A, B, C, S, \hat{Q}_i, P_i, R^{-1}, K(t), K(t+1)) \\ [x_{k+1}(t) - x_k(t)] = \mathcal{G}(A, B, C, S, \hat{Q}_i, P_i, R^{-1}, \\ K(t), \xi_{k+1}(t), \xi_{k+1}(t+1), e_k(t+1), \tilde{e}_k(t+1)), \quad (87)$$

where  $\mathcal{H}(\cdot)$  and  $\mathcal{G}(\cdot)$  are functions of their arguments and independent of the states. If both functions are set to zero, the equation (87) holds independently of the difference of the current difference of in state. Doing this yields the discrete Matrix Riccati equation  $K(t)$  and the optimal predictor  $\xi_{k+1}(t)$  in (49) and (51).

Comparing to the boundary conditions in equation (84), there is one more term  $C^\top F_i^\top Q_i F_i e_{k+1}(t_i)$  added at the end. Note that

$$e_{k+1}(t_i) = e_k(t_i) - C(x_{k+1}(t_i) - x_k(t_i)), \quad (88)$$

which yields

$$C^\top F_i^\top Q_i F_i e_{k+1}(t_i) = C^\top F_i^\top Q_i F_i C(x_{k+1}(t_i) - x_k(t_i)) \\ + C^\top F_i Q_i F_i e_k(t_i), \quad (89)$$

and gives rise to the boundary conditions in (50) and (52).

APPENDIX F  
PROOF OF THEOREM 4

According to Definition 2, the piecewise spatial tracking requirement is to track the transition vertices  $r_i$  at time instants  $t_i$  and the linear path between  $r_{i-1}$  and  $r_i$  during the sub-interval  $(t_{i-1}, t_i]$ . Therefore, the piecewise spatial path tracking problem is

$$(Gu)(t_i) = r_i, \quad i = 1, \dots, M, \\ Gu(t) \in \mathcal{R}_i, \quad t \in (t_{i-1}, t_i], \quad i = 1, \dots, M, \quad (90)$$

where  $\mathcal{R}_i$  is the set of all points along each linear sub-path, i.e.  $\mathcal{R}_i = \{y \in R^m : y = r_i - \gamma(r_i - r_{i-1}), \gamma \in [0, 1]\}$ . Note that all the elements in  $\mathcal{R}_i$  have the linear relationship

$$P_i y = P_i r_i, \quad \forall y \in \mathcal{R}_i, \quad (91)$$

where  $P_i \in R^{(m-1) \times m}$  is a full row rank matrix. It follows that

$$P_i(r_i - \gamma(r_i - r_{i-1})) = P_i r_i, \quad (92)$$

which yields  $P_i a_i = 0$ , and further gives rise to the condition (57). In addition, the hard output constraint set defined in (58) is used to prevent overshoot. Therefore, a piecewise spatial ILC problem is formulated as

$$\lim_{k \rightarrow \infty} y_k = y^* \in \Phi, \quad \lim_{k \rightarrow \infty} y_k(t_i) = r_i, \quad i = 0, \dots, M, \\ \lim_{k \rightarrow \infty} P_i y_k(t) = P_i r_i, \quad t \in (t_{i-1}, t_i], \quad i = 1, \dots, M. \quad (93)$$

Therefore, it is clear that the generalized ILC problem (19) is reduced to exactly the problem (93) when the parameters are set to the values described in this theorem. Therefore, Algorithm 1 can solve the problem (90) as a special case. As the piecewise spatial path tracking problem is a special case of generalized ILC problems, the proofs of the convergence properties in (59) and (60) follow exactly from the proofs of Theorem 2 and 3.

## ACKNOWLEDGMENT

This work was partially funded by the China Scholarship Council (CSC), the ZZU-Southampton Collaborative Research Project under Grant 16306/01, the Royal Society International Exchanges Award under Grant IE161369 and Natural Science Foundation of China under Grant 61773232.

## REFERENCES

- [1] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *International Journal of Control*, vol. 73, no. 10, pp. 930–954, 2000.
- [2] M. Norrlof and S. Gunnarsson, "Time and frequency domain convergence properties in iterative learning control," *International Journal of Control*, vol. 75, no. 14, pp. 1114–1126, 2002.
- [3] A. Tayebi and M. B. Zaremba, "Robust iterative learning control design is straightforward for uncertain LTI systems satisfying the robust performance condition," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 101–106, 2003.
- [4] K. L. Moore, Y. Q. Chen, and V. Bahl, "Monotonically convergent iterative learning control for linear discrete-time systems," *Automatica*, vol. 41, no. 9, pp. 1529–1537, 2005.
- [5] J. Bolder and T. Oomen, "Inferential iterative learning control: A 2D-system approach," *Automatica*, vol. 71, pp. 247–253, 2016.
- [6] 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.
- [7] L. Wang, C. T. Freeman, and E. Rogers, "Predictive iterative learning control with experimental validation," *Control Engineering Practice*, vol. 53, pp. 24–34, 2015.
- [8] X. Li, Q. Ren, and J.-X. Xu, "Precise speed tracking control of a robotic fish via iterative learning control," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 65–76, 2016.
- [9] X. Jin, "Fault-tolerant iterative learning control for mobile robots non-repetitive trajectory tracking with output constraints," *Automatica*, vol. 94, pp. 63–71, 2018.
- [10] 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.
- [11] T. Liu and Y. Q. Wang, "A synthetic approach for robust constrained iterative learning control of piecewise affine batch processes," *Automatica*, vol. 48, no. 11, pp. 2762–2775, 2012.
- [12] Y. Wang, J. Zhang, F. Zeng, N. Wang, X. Chen, B. Zhang, D. Zhao, W. Yang, and C. Cobelli, "learning" can improve the blood glucose control performance for type 1 diabetes mellitus," *Diabetes Technology and Therapeutics*, vol. 19, no. 1, pp. 41–48, 2017.
- [13] C. T. Freeman, *Control System Design for Electrical Stimulation in Upper Limb Rehabilitation*. Springer International Publishing, 2016.
- [14] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *Control Systems, IEEE*, vol. 26, no. 3, pp. 96–144, 2006.
- [15] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Transaction On Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1099–1121, 2007.
- [16] D. H. Owens and J. Hatonen, "Iterative learning control - An optimization paradigm," *Annual Reviews in Control*, vol. 29, no. 1, pp. 57–70, 2005.
- [17] P. Janssens, G. Pipeleers, and J. Swevers, "A data-driven constrained norm-optimal iterative learning control framework for LTI systems," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 546–551, 2013.
- [18] M. Volckaert, M. Diehl, and J. Swevers, "Generalization of norm optimal ILC for nonlinear systems with constraints," *Mechanical Systems and Signal Processing*, pp. 280–296, 2013.
- [19] S. Mishra, U. Topcu, and M. Tomizuka, "Optimization-based constrained iterative learning control," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 6, pp. 1613–1621, 2011.
- [20] J.-X. Xu, J. Xu, and T. H. Lee, "Iterative learning control for systems with input deadzone," *IEEE Transactions on Automatic Control*, vol. 50, no. 9, pp. 1455–1459, 2005.
- [21] B. J. Driessen and N. Sadegh, "Multi-input square iterative learning control with input rate limits and bounds," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 32, no. 4, pp. 545–550, 2002.
- [22] 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.
- [23] J.-X. Xu and X. Jin, "State-constrained iterative learning control for a class of MIMO systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 5, pp. 1322–1327, 2013.
- [24] X. Jin and J.-X. Xu, "Iterative learning control for output-constrained systems with both parametric and nonparametric uncertainties," *Automatica*, vol. 49, 2013.
- [25] H. Ding and J. Wu, "Point-to-point control for a high-acceleration positioning table via cascaded learning schemes," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 5, pp. 2735–2744, 2007.
- [26] J. Park, P. H. Chang, H. S. Park, and E. Lee, "Design of learning input shaping technique for residual vibration suppression in an industrial robot," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 1, pp. 55–65, 2006.
- [27] J. van de Wijdeven and O. Bosgra, "Residual vibration suppression using Hankel iterative learning control," *International Journal of Robust Nonlinear Control*, vol. 18, no. 10, pp. 1034–1051, 2008.
- [28] C. T. Freeman and T. V. Dinh, "Experimentally verified point-to-point iterative learning control for highly coupled systems," *International Journal of Adaptive Control and Signal Processing*, vol. 29, pp. 302–324, 2015.
- [29] 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.
- [30] S.-H. Zhou, Y. Tan, D. Oetomo, C. T. Freeman, E. Burdet, and I. Mareels, "Modeling of endpoint feedback learning implemented through point-to-point learning control," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 5, pp. 1576–1585, 2017.
- [31] C. T. Freeman, "Constrained point-to-point iterative learning control with experimental verification," *Control Engineering Practice*, vol. 20, no. 5, pp. 489–498, 2012.
- [32] B. Chu, C. T. Freeman, and D. H. Owens, "A novel design framework for point-to-point ILC using successive projection," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1156–1163, 2015.
- [33] 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.
- [34] T. D. Son, 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.
- [35] Y. Chen, B. Chu, and C. T. Freeman, "Point-to-point iterative learning control with optimal tracking time allocation," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1685–1698, 2018.
- [36] —, "A coordinate descent approach to optimal tracking time allocation in point-to-point ILC," *Mechatronics*, vol. 59, pp. 25–34, 2019.
- [37] G. Gauthier and B. Boulet, "Robust design of terminal ILC with mixed sensitivity approach for a thermoforming oven," *Journal of Control Science and Engineering*, vol. 2008, 2008, Article ID 289391.
- [38] Y. Wang and Z. Hou, "Terminal iterative learning control based station stop control of a train," *International Journal of Control*, vol. 84, no. 7, pp. 1263–1277, 2011.
- [39] J.-X. Xu, Y. Chen, T. Lee, and S. Yamamoto, "Terminal iterative learning control with an application to RTPCVD thickness control," *Automatica*, vol. 35, no. 9, pp. 1535–1542, 1999.
- [40] J.-X. Xu and D. Huang, "Initial state iterative learning for final state control in motion systems," *Automatica*, vol. 44, no. 12, pp. 3162–3169, 2008.
- [41] D. H. Owens, C. T. Freeman, and B. Chu, "Generalized norm optimal iterative learning control with intermediate point and sub-interval tracking," *International Journal of Automation and Computing*, vol. 12, no. 3, pp. 243–253, 2015.
- [42] 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.
- [43] Y. Chen, B. Chu, and C. T. Freeman, "Spatial path tracking using iterative learning control," in *55th IEEE Conference on Decision and Control*, Las Vegas, US, 2016, pp. 7189–7194.
- [44] M. Böck and A. Kugi, "Constrained model predictive manifold stabilization based on transverse normal forms," *Automatica*, vol. 74, pp. 315–326, 2016.
- [45] A. Hladio, C. Nielsen, and D. Wang, "Path following for a class of mechanical systems," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2380–2390, 2013.

- [46] D. Verscheure, B. Demeulenaere, J. Swevers, J. de Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.
- [47] T. Lippa and S. Boyd, "Minimum-time speed optimisation over a fixed path," *International Journal of Control*, vol. 87, no. 6, pp. 1297–1311, 2014.
- [48] K. L. Moore, M. Ghosh, and Y. Q. Chen, "Spatial-based iterative learning control for motion control applications," *Meccanica*, vol. 42, pp. 167–175, 2007.
- [49] S. K. Sahoo, S. K. Panda, and J.-X. Xu, "Application of spatial iterative learning control for direct torque control of switched reluctance motor drive," in *IEEE Power Engineering Society General Meeting*, Tampa, US, 2007, pp. 1–7.
- [50] P. Janssens, W. V. Loock, G. Pipeleers, F. Debrouwere, and J. Swevers, "Iterative learning control for optimal path following problems," in *52nd IEEE Conference on Decision and Control*, Florence, Italy, 2013, pp. 6670 – 6675.
- [51] A. Steinhauser and J. Swevers, "Iterative learning of feasible time-optimal trajectories for robot manipulators," in *The 20th World Congress of the International Federation of Automatic Control*, Toulouse, France, 2017.
- [52] J. Liu, X. Dong, D. Huang, and M. Yu, "Composite energy function-based spatial iterative learning control in motion systems," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1834–1841, 2018.
- [53] D. J. Hoelzle and K. L. Barton, "On spatial iterative learning control via 2-D convolution: Stability analysis and computational efficiency," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1504–1512, 2016.
- [54] D. H. Owens and R. P. Jones, "Iterative solution of constrained differential/algebraic systems," *International Journal of Control*, vol. 27, no. 6, pp. 957–964, 1978.
- [55] L. Gubin, B. Polyak, and E. Raik, "The method of projections for finding the common point of convex sets," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 6, pp. 1–24, 1967.
- [56] D. H. Owens, C. T. Freeman, and B. Chu, "Multivariable norm optimal iterative learning control with auxiliary optimisation," *International Journal of Control*, vol. 86, no. 6, pp. 1026–1045, 2013.
- [57] I. Lim, D. J. Hoelzle, and K. L. Barton, "A multi-objective iterative learning control approach for additive manufacturing applications," *Control Engineering Practice*, vol. 64, pp. 74–87, 2017.
- [58] Z. Wang, C. P. Pannier, K. Barton, and D. J. Hoelzle, "Application of robust monotonically convergent spatial iterative learning control to microscale additive manufacturing," *Mechatronics*, vol. 56, no. 268-276, 2018.
- [59] J. D. Ratcliffe, "Iterative learning control implemented on a multi-axis system," Ph.D. dissertation, University of Southampton, Southampton, 2005.
- [60] Y. Chen, B. Chu, and C. T. Freeman, "Generalized norm optimal iterative learning control: Constraint handling," in *The 20th World Congress of the International Federation of Automatic Control*, Toulouse, France, 2017, pp. 13396–13401.
- [61] N. Amann, "Optimal algorithms for iterative learning control," Ph.D. dissertation, University of Exeter, Exeter, 1996.