Generalized Iterative Learning Control with Mixed System Constraints: A Gantry Robot based Verification

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

[a]*School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom.*
[b]*School of Electrical Engineering, Zhengzhou University, Science Road 100, Zhengzhou, Henan, 450001, China.*

Abstract

Iterative learning control (ILC) aims at improving the tracking performance of repetitive tasks based on information learnt from past attempts (trials). Modern practical applications demand more flexibility than current frameworks can deliver, in both how the task is specified and how system constraints are applied. To provide these features, an ILC framework is formulated in this paper for a generalized design objective with mixed system constraints, which includes intermediate position and sub-interval tracking as special cases. This is the first framework to combine a generalized ILC task description with constraint handling for continuous time systems. The successive projection method is applied to yield a comprehensive ILC algorithm with attractive convergence properties and computationally efficient implementation. This algorithm is verified experimentally on a gantry robot test platform, whose results reveal its practical efficacy and robustness against model uncertainty.

Keywords: iterative learning control, path following, constraint handling, gantry robot.

1. Introduction

ILC is a high performance control design method, and can improve the tracking accuracy of systems which operate repetitively over a finite time horizon. At the end of each trial, ILC updates the control input for the next trial by learning from prior information, e.g. measured output signals, from previous trials. As the reference remains the same for each trial, ILC can theoretically enable the tracking error to converge to zero after sufficient trials. It hence allows a wide range of practical tasks to be performed with high precision, such as those required by robotic systems [1, 2, 3], chemical batch processing [4, 5] and stroke rehabilitation [6]. ILC has also been combined with other methods, such as PID [7], iterative feedback tuning [8], extremum seeking [9] and projection [10], to efficiently solve the specific tracking tasks in practice. See [11] for a detailed overview of ILC.

Research in ILC, e.g. [12, 13, 14], predominately follows the 'classical' problem setup, whose design objective is to track a given reference trajectory defined over the whole specified finite time horizon. However, this classical ILC task description is overly restrictive and does not capture the various needs of engineering practice. An example is a robotic manipulator's pick-and-place task, whose output trajectory is only critical at a finite number of 'intermediate' time instants t_i. To address this case, an ILC framework termed intermediate point ILC (or point-to-point ILC) was formulated in [15], which leveraged significant control design flexibility by eliminating the unnecessary output tracking requirements. The released control design freedom was further exploited in [16, 17, 18] to employ the optimization of an additional performance index and tracking accuracy. The technique of intermediate point ILC has been implemented in applications consisting of point-to-point tracking, such as high-acceleration positioning tables [19], robotic manipulators [20], two-mass systems [21], electro-mechanical systems [22] and human motor systems [23]. Subsequent research in [24] expanded the intermediate point ILC framework to allow simultaneous tracking of both the reference along sub-intervals $[t_{i-1},\ t_i]$ and the intermediate time instants t_i, in which the tracking error on the latter exhibits more rapid convergence than elsewhere along the time horizon. Note that the authors' previous research in [25, 26] considered the optimal tracking time allocation problem of intermediate point ILC to optimize the control effort of a robotic task. However, the proposed algorithms are restricted to the class of intermediate point ILC problem setup only.

Moreover, the classical ILC task description does not incorporate the system constraints, which is somehow not fully adaptive to the practical task requirement. For example, the input constraints may represent the safety input load range of a power system, beyond which the risk of system damage has a steep rise. Also, the output constraints may represent the acceptable working space for a laser cutting end-effector, and prevent the potential overshoot problem that causes damage to the device and re-

*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), liuyh@zzu.edu.cn (Yanhong Liu)

duces the product quality. Therefore, substantial research on ILC has considered the system constraint handling as an essential part of their control design objectives. The main stream of constrained ILC research, e.g. [27, 28, 29, 30, 31, 32], focused on input constraint handling methods, while other work, e.g. [33, 34, 35], studied the influence of state and output constraints within the ILC design phase. The relevant work on constrained ILC significantly links the theoretical ILC design framework to the practical task requirement, and hence guarantees the reliable implementation of ILC in practice.

Parallel research has also modified the classical ILC task description to tackle the path following tasks, in which the output is required to follow a path defined in space without *a priori* temporal information. Compared to the trajectory tracking requirement of classical ILC, this path following requirement has significantly expands the control design freedom in the time domain, and is ideally suited to automation tasks such as welding, laser cutting and additive manufacturing. The technique of ILC was first used in [36] to improve the corner path following accuracy of a two-axis gimbal system. It proposed an ILC law to update the switching policy (on or off) of the motors (operating in a constant velocity mode) using the minimum corner distance measured at the previous trial. Subsequent research in [37] focused on a switched reluctance motor and redefined the system input and output coordinates with respect to the rotor position. The input voltage is then updated by ILC based on the previous trial torque error to reduce the periodic varying torque ripple. Similarly, the work in [38] redefined the system parameters in spatial coordinates via 2D convolution reconstruction, and reformulated spatial forms of the classical ILC update laws. It increased the task performance of specific applications with spatial steady-state output, e.g. additive manufacturing. Meanwhile, the research in [39] used ILC to update the nominal system model of a path planning problem, which aimed at achieving high accuracy and minimum time of a repetitive 2D robotic path following task. Although the path following problem has been studied in the field of autonomous air or surface vehicle steering, e.g. [40, 41, 42, 43], and robotic motion planning, e.g. [44, 45, 46, 47], they have not considered the ILC setting to improve the path following performance and hence their proposed control algorithms are sensitive to the model uncertainty.

This paper brings together the aforementioned extensions of the classical ILC task description to yield a generalized ILC framework for continuous time systems. This framework addresses high performance repetitive tracking tasks in a wide class of systems with a mixed form of system constraints. To achieve this, a generalized ILC design objective is formulated to embed intermediate position tracking requirements at time instants t_i as well as linear path following requirements on sub-intervals $[t_{i-1},\ t_i]$. The design objective is combined with a mixed form of system constraints to yield a generalized ILC problem. To solve this problem, a comprehensive ILC algorithm with guaranteed convergence properties is proposed based on the successive projection method, such that it can be implemented on a wide range of practical applications. A typical example is the gantry robot, which replicates various application scenarios. For example, It can perform the pick-and-place task, which picks up a payload from a dispenser and places it down onto a moving convey underneath the gantry. Also, it is involved in laser cutting and 3D printing tasks by moving its axis displacement to change the laser head and nozzle positions. To examine the practical performance, this algorithm is verified on a gantry robot test platform to establish its robust performance and practical efficacy.

The authors' recent work in [48] considered a similar problem for linear discrete time systems, where all the signals (e.g. input and output) are finite dimensional. The result however does not apply to the continuous time problems, whose signals are infinite dimensional representing a significant shortcoming. The continuous time problem formulation is necessary as there are applications where only the method of design by emulation (rather than direct digital control) is possible or preferred, e.g. in the control of free electron lasers, as the underlying signals are continuous. In this paper, an infinite dimensional Hilbert space setting is used to address this problem. This setting can describe a range of systems, e.g. continuous linear time invariant systems, continuous linear time varying systems, linear differential systems and linear deferential delay systems. Therefore, it is a substantial generalization to [48] and so can be used on much broader application classes. The Hilbert space theoretical deviation also has substantial difference. In addition, some initial results in this paper were reported in [49]; in this paper a detailed analysis of the proposed algorithm's properties is made with rigorous proof given, and its practical implementation procedure is also provided in detail.

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 Lebesgue 2-space of $\mathbb{R}^\ell$ valued signals 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 $\mathbb{Y}$; the symbol $\preceq$ is the pointwise inequality sign; $P_\Theta(x)$ denotes the projection x to the set Θ in some Hilbert space.

2. Problem formulation

This section first introduces the system dynamics and specifies a general task description. Then, it embeds a mixed form of system constraints into the task description to yield a generalized ILC design objective and formulates a generalized ILC problem.

2.1. System dynamics

Consider an ℓ-input, m-output linear time-invariant system given in state space form

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

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

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

The input signal u_k and output signal y_k belongs to the corresponding input and output Hilbert spaces $L_2^\ell[0,\ T]$ and $L_2^m[0,\ T]$ defined with inner products and associated induced norms

$$\langle u,\ v\rangle_R = \int_0^T u^\top(t)Rv(t)\mathrm{d}t,\ \ \|u\|_R = \sqrt{\langle u,\ u\rangle_R}, \tag{3}$$

$$\langle x,\ y\rangle_S = \int_0^T x^\top(t)Sy(t)\mathrm{d}t,\ \ \|y\|_S = \sqrt{\langle y,\ y\rangle_S}, \tag{4}$$

in which $R \in \mathbb{S}_{++}^\ell$ and $S \in \mathbb{S}_{++}^m$. The convolution operator $G:\ L_2^\ell[0,\ T] \to L_2^m[0,\ T]$ takes the form

$$(Gu_k)(t) = \int_0^t Ce^{A(t-s)}Bu_k(s)\mathrm{d}s. \tag{5}$$

The signal $d \in L_2^m[0,\ T]$ represents the effect of initial condition, and has the form

$$d(t) = Ce^{At}x_0. \tag{6}$$

To delimitate the application area of the ILC update algorithm, the following assumptions are made. Firstly, the value of the signal d is considered as $d(t) = 0$ by taking $x_0 = 0$ in later control design. Since the signal d is constant with respect to the identical initial state x_0, it can be absorbed by the reference trajectory $r(t)$ without loss of generality. Secondly, the input and output constrained sets Ω and Φ are considered as convex to fit the convergence proof of successive projection.

2.2. Generalized ILC task description

The classical ILC design objective is to iteratively find the input signal u_k such that the associated output $y_k = Gu_k$ ultimately tracks a given reference trajectory r defined over the whole time horizon, i.e. $\lim_{k\to\infty} y_k = r$.

To broaden the ILC application range, its classical design objective is extended in this paper to subsume both intermediate position tracking at time instants t_i, $i = 0, ..., M$, with ordering constraint

$$0 = t_0 < t_1 < \cdots < t_M = T, \tag{7}$$

and path following on sub-intervals $[t_{i-1},\ t_i]$, $i = 1, ..., M$. To formulate the generalized ILC design objective, the linear mapping $\zeta \mapsto \zeta^e$ is defined to map any signal $\zeta \in L_2^m[0,\ T]$ to the 'extended signal'

$$\zeta^e = \begin{bmatrix} F\zeta \\ P\zeta \end{bmatrix} \in H. \tag{8}$$

Here, the operator F selects the information of the signal ζ at time instants t_i, $i = 0, ..., M$, which is defined as

$$F\zeta = \begin{bmatrix} F_0\zeta(t_0) \\ \vdots \\ F_M\zeta(t_M) \end{bmatrix},\ F_i\zeta(t_i) \in \mathbb{R}^{f_i},\ i = 0, ..., M. \tag{9}$$

Likewise, the operator P extracts the information within the signal ζ to form linear combinations along sub-intervals $[t_{i-1},\ t_i]$, $i = 1, ..., M$, which is defined as

$$\begin{aligned} P\zeta &= \begin{bmatrix} (P\zeta)_1 \\ \vdots \\ (P\zeta)_M \end{bmatrix},\ (P\zeta)_i \in L_2^{p_i}[t_{i-1}, t_i], \\ (P\zeta)_i(t) &= P_i\zeta(t),\ t \in [t_{i-1},\ t_i],\ i = 1, ..., M. \end{aligned} \tag{10}$$

The Hilbert space H in (8) is defined as

$$H = \mathbb{R}^{f_0} \times \cdots \times \mathbb{R}^{f_M} \times L_2^{p_1}[t_0, t_1] \times \cdots \times L_2^{p_M}[t_{M-1}, t_M]$$

with inner product and associated induced norm

$$\begin{aligned} \langle(\omega, \nu),\ (\mu, \lambda)\rangle_{\tilde{Q}} &= \sum_{i=0}^{M} \omega_i^\top Q_i\mu_i + \sum_{i=1}^{M}\int_{t_{i-1}}^{t_i} \nu_i^\top(t)\hat{Q}_i\lambda_i(t)\mathrm{d}t, \\ \|(\omega, \nu)\|_{\tilde{Q}} &= \sqrt{\langle\omega,\ \omega\rangle_{[Q]} + \langle\nu,\ \nu\rangle_{[\hat{Q}]}}, \end{aligned} \tag{11}$$

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

$$\begin{aligned} \omega &= [\omega_0,\ \omega_1, \ldots,\ \omega_M]^\top,\ \mu = [\mu_0,\ \mu_1, \ldots,\ \mu_M]^\top, \\ \nu &= [\nu_1,\ \nu_2, \ldots,\ \nu_M]^\top,\ \lambda = [\lambda_1,\ \lambda_2, \ldots,\ \lambda_M]^\top. \end{aligned}$$

Note that the matrices $F_i \in \mathbb{R}^{f_i\times m}$ and $P_i \in \mathbb{R}^{p_i\times m}$ are both assumed to be full row rank; the intermediate positions $\omega_i,\ \mu_i \in \mathbb{R}^{f_i}$; the signals $\nu_i,\ \lambda_i \in L_2^{p_i}[t_{i-1},\ t_i]$; $\tilde{Q}$ denotes the data sets $\{Q_0, \ldots, Q_M, \hat{Q}_1, \ldots, \hat{Q}_M\}$, where the matrices $Q_i \in \mathbb{S}_{++}^{f_i}$ and $\hat{Q}_i \in \mathbb{S}_{++}^{p_i}$.

According to definitions (9) and (10), the extended signal ζ^e comprises a subset of ζ at intermediate time instants, together with a subset of ζ defined over the sub-intervals. To represent a general ILC tracking requirement, an 'extended system' is formulated with dynamics

$$y_k^e = G^e u_k = (Gu_k)^e = \begin{bmatrix} FGu_k \\ PGu_k \end{bmatrix}, \tag{12}$$

where $G^e : L_2^\ell[0,T] \to H$ is a linear operator. The extended system dynamics (12) capture the relationship between the input signal u_k and the 'extended output' y_k^e. Using the extended system dynamics, the generalized ILC design objective is described as: to find the input signal u_k, such that the associated extended output $y_k^e = G^e u_k$ ultimately tracks the extended reference trajectory r^e for each trial, i.e. $\lim_{k\to\infty} y_k^e = r^e$.

This design objective enables both an intermediate position tracking requirement, $F_i y_k(t_i) = F_i r(t_i)$, at time instants t_i, $i = 0, ..., M$, and a linear relationship between outputs, $P_i y_k(t) = P_i r(t)$, on sub-intervals $[t_{i-1},\ t_i]$, $i = 1, ..., M$ (to enforce tracking along lines or planes without *a priori* temporary constraints).

2.3. Generalized ILC problem with system constraints

In practical implementation, the effect of system constraints must be considered within the control design, as they embed physical limitations or performance requirements of the practical system. The input constraints typically represent the safe input load range, and assume the following forms using the notion of the set Ω:

- Input saturation constraint

$$\Omega = \{u \in L_2^\ell[0,\ T] : |u(t)| \preceq \mathcal{M}(t),\ t \in [0,\ T]\}, \quad (13)$$

- Input effort constraint

$$\Omega = \{u \in L_2^\ell[0,\ T] : \int_0^T u^\top(t)u(t)\mathrm{d}t \leqslant \mathcal{M}\}. \quad (14)$$

- Input gradient constraint

$$\Omega = \{u \in L_2^\ell[0,\ T] : \underline{\mathcal{M}}(t) \preceq \dot{u}(t) \preceq \overline{\mathcal{M}}(t),\ t \in [0,\ T]\}. \quad (15)$$

Similarly, the output constraints restrict the system output to an acceptable region, and assumes the following forms using the notion of the set Φ:

- Output saturation constraint

$$\Phi = \{y \in L_2^m[0,\ T] : |y(t)| \preceq \mathcal{N}(t),\ t \in [0,\ T]\}, \quad (16)$$

- Output polyhedral constraint

$$\Phi = \{y \in L_2^m[0,\ T] : a_i^\top y(t) \leqslant b_i,\ a_i \in \mathbb{R}^m,\ b_i \in \mathbb{R}, \ i = 1, \ldots, M,\ t \in [0,\ T]\}. \quad (17)$$

- Output gradient constraint

$$\Phi = \{y \in L_2^m[0,\ T] : \underline{\mathcal{N}}(t) \preceq \dot{y}(t) \preceq \overline{\mathcal{N}}(t),\ t \in [0,\ T]\}. \quad (18)$$

Note that all these listed input and output constrained sets are convex, which naturally follows the assumption of convex sets Ω and Φ in Section 2.1. Therefore, the generalized ILC objective is combined with the above system constraints to form the following **generalized ILC problem**: to iteratively update the input signal u_k, such that the generalized design objective is achieved and the system constraints are also satisfied, i.e.

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

Remark 1. *The generalized ILC problem (19) collapses to existing ILC problems by setting the values of Q, $\hat{Q}$, F and P appropriately, e.g. 1) $Q_i = 0$, $P_i = I$, classical ILC; 2) $\hat{Q}_i = 0$, $F_i = I$, intermediate point ILC.*

3. ILC algorithm using successive projection

In this section, the generalized ILC problem (19) is first formulated into a standard successive projection framework, and then a comprehensive ILC algorithm with desired convergence properties is derived to solve this problem iteratively using experimental data.

3.1. Successive projection interpretation

The generalized ILC problem (19) is equivalent to iteratively finding an element (y^{e*}, y^*, u^*) in the intersection of the two sets

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

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

where the set S_1 describes the system dynamics and the set S_2 describes the generalized design objective as well as the system constraints. Since the system dynamics are linear and the constrained sets Ω and Φ defined in (13)-(17) are convex, the above two sets are both convex. Note that $\hat{H}$ is a Hilbert space defined as

$$\hat{H} = \mathbb{R}^{f_0} \times \cdots \times \mathbb{R}^{f_M} \times L_2^{p_1}[t_0,\ t_1] \times \cdots \times L_2^{p_M}[t_{M-1},\ t_M] \times L_2^m[0,\ T] \times L_2^\ell[0,\ T], \quad (22)$$

whose inner product and associated induced norm are naturally derived from (3), (4) and (11).

To address the equivalent problem, the successive projection method is considered in this paper, whose basic scheme is interpreted in Figure 1. The implementation procedure and convergence properties of the successive projection method are given in the next lemma.

Lemma 1. *[50] Let S_1 and S_2 be two closed convex sets in a Hilbert space X. Define the projection operators $P_{S_1}(\cdot)$ and $P_{S_2}(\cdot)$ as*

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

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

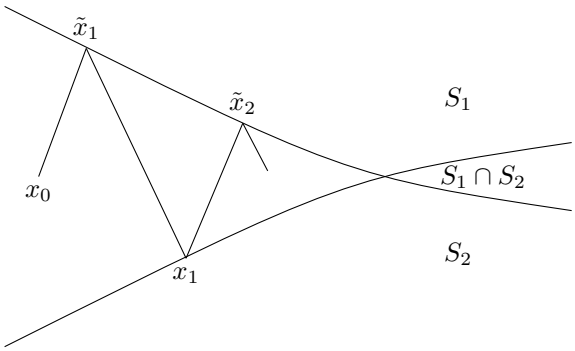

Figure 1: Illustration of the successive projection algorithm: start from an arbitrary point in the space $\hat{H}$, then iteratively project the point to the two convex sets S_1 and S_2 respectively and finally reach the intersection of the two sets.

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

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

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

$$\|\tilde{x}_{k+2} - x_{k+1}\|_X^2 \leqslant \|\tilde{x}_{k+1} - x_k\|_X^2 . \tag{26}$$

For any $\epsilon > 0$, there exists an integer N such that

$$\|\tilde{x}_{k+1} - x_k\|_X^2 < \epsilon,\ \forall\ k > N, \tag{27}$$

and minimum distance between two sets is attained, i.e.

$$\lim_{k\to\infty} \|\tilde{x}_k - x_k\|_X^2 = \inf_{\tilde{x}\in S_1, x\in S_2} \|\tilde{x} - x\|_X^2 . \tag{28}$$

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

$$\|x_{k+1} - x\|_X^2 \leqslant \|x_k - x\|_X^2 ,\ \forall\ x \in S_1 \cap S_2,\ k \geqslant 0. \tag{29}$$

3.2. Comprehensive ILC algorithm

The successive projection method in Lemma 1 can be directly applied to solve the generalized ILC problem (19) by considering $X = \hat{H}$ and the sets S_1 and S_2 to be the forms in (20) and (21). In this sense, a comprehensive ILC algorithm is obtained as follows:

Algorithm 1. *Given system dynamics (1), input constraint set Ω, output constraint set Φ, extended reference r^e, and any initial values $u_0 \in \Omega$, $\tilde{r}_0 \in \Phi$, the input sequence $\{u_k\}$ generated by the ILC update law*

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

with the projections

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

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

iteratively solves the generalized ILC problem (19). The linear operator G^s is defined as

$$G^s u = \begin{bmatrix} G_\Lambda^e u \\ Gu \end{bmatrix} : L_2^\ell[0,\ T] \to \tilde{H}, \tag{33}$$

with its adjoint operator G^{s*}, the error e_k^s is defined as

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

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

$$\begin{aligned} \tilde{H} = \mathbb{R}^{f_0} \times \cdots \times \mathbb{R}^{f_M} \times L_2^{p_1}[t_0,\ t_1] \times \\ \cdots \times L_2^{p_M}[t_{M-1},\ t_M] \times L_2^m[0,\ T] \end{aligned} \tag{35}$$

whose inner product and associated induced norm are naturally derived from (4) and (11).

Proof. See Appendix A. □

3.3. Convergence properties analysis

To analyze the convergence properties of Algorithm 1, a realistic condition is considered such that perfect tracking is possible for problem (19), i.e. $S_1 \cap S_2 \neq \varnothing$. Under this condition, the convergence performance of the successive projection based algorithm within an infinite dimensional Hilbert space is shown in the next theorem.

Theorem 1. *If $S_1 \cap S_2 \neq \varnothing$, Algorithm 1 iteratively achieves the generalized ILC design objective*

$$\lim_{k\to\infty} y_k^e = r^e \tag{36}$$

and the error e_k^s and the input signal u_k converge monotonically with respect to the functions

$$J_k = \|\mathcal{M} e_k^s\|_{[Q]}^2 + \|\mathcal{N} e_k^s\|_R^2 ,\ [Q] = \{\tilde{Q},\ S\}, \tag{37}$$

and

$$\tilde{J}_k = \|\tilde{r}_k - \hat{y}\|_S^2 + \|u_k - \hat{u}\|_R^2 ,\ (r^e, \hat{y}, \hat{u}) \in S_1 \cap S_2, \tag{38}$$

where $\mathcal{M} = (I + G^{s*}G^s)^{-1}$ and $\mathcal{N} = G^{s*}(I + G^{s*}G^s)^{-1}$.

In addition, if the limit of the input signal u_k exits, it follows that

$$\lim_{k\to\infty} u_k = u^* \in \Omega,\ \lim_{k\to\infty} y_k = y^* \in \Phi. \tag{39}$$

Proof. See Appendix B. □

From the above theorem, Algorithm 1 guarantees the desired accuracy for both intermediate point tracking and sub-interval path following, and the monotonic convergence properties of the error e_k^s and input signal u_k with respect to some cost functions. This algorithm is capable of minimizing additional cost functions while maintaining high tracking accuracy which are appealing in practice. However, the tracking error convergence only suggests the

weak convergence of the input signal in an infinite dimensional Hilbert space, and does not explicitly imply its strong convergence. Therefore, an additional assumption is made on the existence of the input limit, and the potential converged values satisfy the system constraints.

While there are no system constraints engaged, Algorithm 1 guarantees minimum control effort of the given task as shown in the next corollary.

Corollary 1. *If $S_1 \cap S_2 \neq \varnothing$, $\Omega = L_2^{\ell}[0,\ T]$, $\Phi = L_2^{m}[0,\ T]$ and $u_0 = 0$, the input u_k converges monotonically with respect to*

$$\hat{J}_k = \langle u_k - \hat{u},\ \mathcal{H}(u_k - \hat{u})\rangle_X\,,\ (r^e, G\hat{u}, \hat{u}) \in S_1 \cap S_2 \quad (40)$$

and the converged input corresponds to the minimum control effort, i.e.

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

where $\mathcal{H} = G^{e*}G^e + I$ and G^{e*} is the Hilbert adjoint operator of G^e.

Proof. See Appendix C. □

Furthermore, consider the extreme scenario that the intersection of the sets S_1 and S_2 is empty, which means perfect tracking is impossible and the problem (19) does not have any solution. However, Algorithm 1 still provides attractive task performance as shown in the next theorem.

Theorem 2. *If $S_1 \cap S_2 = \varnothing$, the distance between the two sequences $\{(\tilde{y}_k^e, \tilde{y}_k, \tilde{u}_k)\}$ and $\{(r^e, \tilde{r}_k, u_k)\}$ updated by Algorithm 1 converges to the minimum distance*

$$\inf_u \{\|r^e - G^e\tilde{u}\|_{\tilde{Q}}^2 + \|\tilde{r} - G\tilde{u}\|_S^2 + \|u - \tilde{u}\|_R^2\},\ \tilde{r} \in \Phi,\ u \in \Omega. \quad (42)$$

between S_1 and S_2, the error e_k^s converges monotonically with respect to J_k defined in (37) and the condition $u_k \in \Omega$ holds for each trial.

Proof. See Appendix D. □

Remark 2. *Note that the work in [48] focused on discrete time systems, whose signals were all finite dimensional. In this special case, strong convergence conditions of all signals are available. This paper generalizes the successive projection method to embed elements (y_k^e, y_k, u_k) in an infinite dimensional Hilbert space X, which leads to substantial difference in the derivation of convergence properties. The proposed algorithm guarantees the tracking error to converge to zero, which meets the design objective. However, it only provides a weak convergence condition of the input signal u_k.*

4. Algorithmic implementation procedure

The previous section proposes an algorithm (Algorithm 1) with an ILC update (30)-(32), and proves its convergence. This section links the theoretical design to the practical application, and describes how the proposed ILC update law should be implemented in practice. The iterative implementation procedure consists of the two steps: 1). The input signal update step using (30); 2). The projection step using (31) and (32), and the detailed step solutions are provided respectively.

4.1. Step One implementation solution

Due to the continuous time system problem setup, the signals u_k and e_k^s are defined in infinite dimensional Hilbert spaces and input signal update (30) of Step One cannot be directly implemented as that in [48] using matrix computation with finite elements. Instead of that, this step is suggested to be implemented using the state feedback and feedforward action with differential equations, which is illustrated in the next proposition.

Proposition 1. *The input signal update step (30) is implemented as a feedforward plus feedback solution*

$$u_{k+1}(t) = u_k(t) + R^{-1}B^{\top}p_{k+1}(t) \qquad (43)$$

with the the costate

$$p_{k+1}(t) = -K(t)(x_{k+1}(t) - x_k(t)) + \xi_{k+1}(t), \qquad (44)$$

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

$$\begin{aligned}\dot{K}(t) = (K(t)BR^{-1}B^{\top} - A^{\top})K(t) - K(t)A \\ - C^{\top}\hat{Q}(t)C - C^{\top}SC\end{aligned} \qquad (45)$$

with boundary conditions

$$\begin{aligned}K(t_i-) = K(t_i+) + C^{\top}F^{\top}Q_iFC,\ 0 \leqslant i \leqslant M, \\ K(T) = 0,\end{aligned} \qquad (46)$$

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

$$\begin{aligned}\dot{\xi}_{k+1}(t) = (K(t)BR^{-1}B^{\top} - A^{\top})\xi_{k+1}(t) \\ + C^{\top}\hat{Q}(t)e_k(t) + C^{\top}S\tilde{e}_k(t)\end{aligned} \qquad (47)$$

with boundary conditions

$$\begin{aligned}\xi_{k+1}(t_i-) = \xi_{k+1}(t_i+) + C^{\top}F^{\top}QFe_k(t_i),\ 0 \leqslant i \leqslant M, \\ \xi_{k+1}(T) = 0,\end{aligned} \qquad (48)$$

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

Proof. See Appendix E. □

4.2. Step Two implementation solution

Step Two aims at projecting the unconstrained input $\tilde{u}_{k+1}$ and output $\tilde{y}_{k+1}$ into the constraint sets Φ and Ω using the projections (31) and (32). The input constraint set Ω is usually pointwise in practice, so it is straightforward

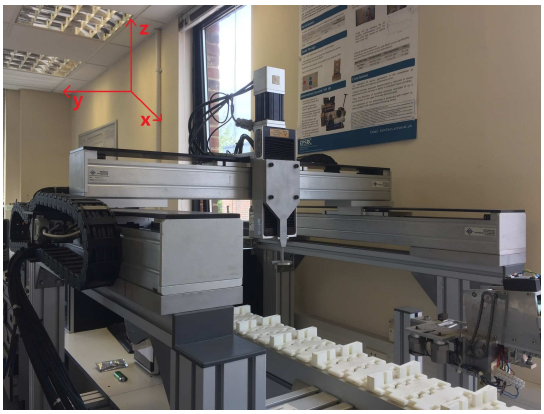

Figure 2: Multi-axis gantry robot test platform.

to obtain the solution of P_Ω. For the saturation form (13), the corresponding solution of $u = P_\Omega(\tilde{u})$ is

$$u(t) = \begin{cases} \mathcal{M}(t), \ \tilde{u}(t) \succ \mathcal{M}(t), \\ \tilde{u}(t), \ -\mathcal{M}(t) \preceq \tilde{u}(t) \preceq \mathcal{M}(t), \\ -\mathcal{M}(t), \ \tilde{u}(t) \prec -\mathcal{M}(t), \end{cases} \tag{49}$$

for $0 \leqslant t \leqslant T$. In addition, the solution of the projection operator $\tilde{r} = P_\Phi(\tilde{y})$ in (32) for the saturation output constraint set (16) is similar to (49).

5. Experimental verification

The practical performance of Algorithm 1 is now verified on a three-axis gantry robot test platform. The corresponding experimental results demonstrate the high tracking performance of this algorithm.

5.1. Test platform specifications

The test platform comprises a multi-axis gantry robot consisting of three perpendicular axes as shown in Figure 2. Note that the x-axis and the y-axis move in the horizontal plane and are driven by linear brush-less DC motors. Moreover, the vertical z-axis is placed on the top of the other two axes, and comprises a linear ball-screw stage driven by a rotary brushless DC motor. The displacement of each axis is measured by an optical incremental encoder installed on the same axis. The hybrid motion of the three axes gives rise to a path of the gantry robot end-effector in 3D space.

The overall structure of the gantry robot test platform is shown in Figure 3. It employs a dSPACE DS1103 microcontroller, Aerotech model BA10 linear amplifiers, and Renishaw RGH22 linear encoders with a resolution of 1 μm for x-axis and y-axis, and Aerotech rotary encoder for z-axis with a resolution of 0.5 μm.

The x-axis and z-axis of the gantry robot have been modelled based on frequency response tests in [51] with

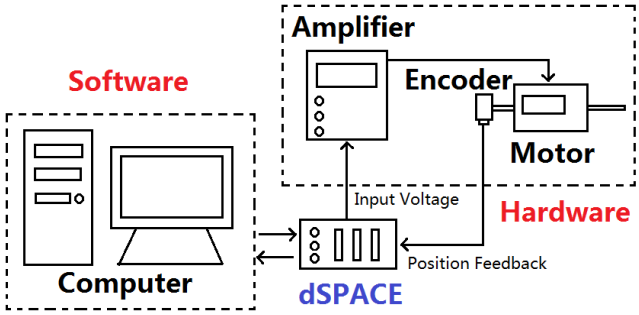

Figure 3: Structure of gantry robot test platform.

transfer functions

$$G_x(s) = \frac{1.67 \times 10^{-5}(s + 500.2)(s + 4.9 \times 10^5)...}{s(s^2 + 24s + 6401)...}$$
$$\frac{(s^2 + 10.58s + 1.145 \times 10^4)(s^2 + 21.98s + 2.9 \times 10^4)}{(s^2 + 21.38s + 2.017 \times 10^4)(s^2 + 139.5s + 2.162 \times 10^5)}$$
$$\text{and } G_z(s) = \frac{15.8869(s + 850.3)}{s(s^2 + 707.6s + 3.377 \times 10^5)}, \tag{50}$$

and a proportional feedback controller with gain 300 is added on the z-axis.

5.2. Task design objectives

The task is specified as using both the x-axis and z-axis ($m = 2$) of the gantry robot to follow a piecewise linear path (the yellow path in Figure 4) composed of five line segments ($M = 5$) during a given trial length $T = 2s$. The path is defined as

$$r(t) = r_{i-1} + \left(\frac{t - t_{i-1}}{t_i - t_{i-1}}\right)(r_i - r_{i-1}), \ t \in [t_{i-1}, \ t_i], \\ i = 1, \ldots, M, \tag{51}$$

where the transition vertices

$$r_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, r_1 = \begin{bmatrix} 0.00345 \\ 0.00476 \end{bmatrix}, r_2 = \begin{bmatrix} 0.00905 \\ 0.00294 \end{bmatrix},$$
$$r_3 = \begin{bmatrix} 0.00905 \\ -0.00294 \end{bmatrix}, r_4 = \begin{bmatrix} 0.00345 \\ -0.00476 \end{bmatrix}, r_5 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

are defined in Cartesian space. The task design objective is to find an input u such that the output y follows the line segment between each pair of transition vertices r_{i-1} and r_i and passes them in ascending order at the intermediate time instants

$$t_0 = 0, \ t_1 = 0.4, \ t_2 = 0.8, \ t_3 = 1.2, \ t_4 = 1.6, \ t_5 = 2.0.$$

The task design objective involves both the intermediate point tracking at time instant t_i and the linear path following along the sub-interval $[t_{i-1}, \ t_i]$, which require

$$(Gu)(t_i) = r_i, \ i = 0, \ldots, M, \tag{52}$$
$$P_i^*(Gu)(t) = P_i^* r(t), \ t \in (t_{i-1}, \ t_i], \ i = 1, \ldots, M, \tag{53}$$

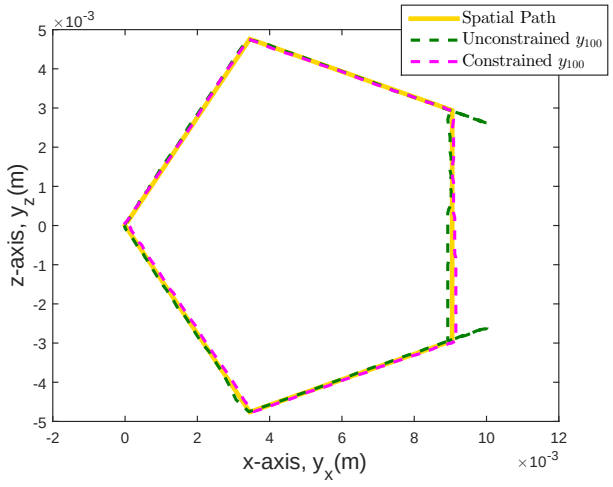

Figure 4: Path profile and converged hybrid output trajectories with and without system constraints.

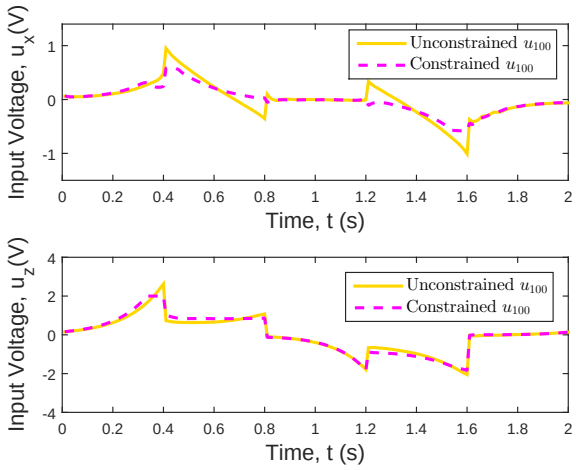

Figure 5: Converged input trajectories along the time horizon with and without system constraints.

and the output constraint

$$\Phi = \{y \in L_2^m[0,\ T] : a_i^\top r_{i-1} \leqslant a_i^\top y(t) \leqslant a_i^\top r_i, \quad t \in (t_{i-1},\ t_i],\ i = 1, \ldots, M\}, \quad (54)$$

where $P_i^* = [(r_i^2 - r_{i-1}^2),\ -(r_i^1 - r_{i-1}^1)]$ and $a_i = r_i - r_{i-1}$.

These objectives are addressed by Algorithm 1 by setting the projection matrices F_i and P_i in (9) and (10) as

$$F_i = I,\ P_i = 100 \cdot [(r_i^2 - r_{i-1}^2),\ -(r_i^1 - r_{i-1}^1)], \quad (55)$$

and embedding the output constraint (54), which yields the solution of $\tilde{r} = P_\Phi(\tilde{y})$ as follows:

$$\tilde{r}(t) = \begin{cases} \tilde{y}(t) + \Delta_{i,i}(t), & a_i^\top \tilde{y}(t) > a_i^\top r_i, \\ \tilde{y}(t), & a_i^\top r_{i-1} \leqslant a_i^\top \tilde{y}(t) \leqslant a_i^\top r_i, \\ \tilde{y}(t) + \Delta_{i,i-1}(t), & a_i^\top \tilde{y}(t) < a_i^\top r_{i-1}, \end{cases} \quad (56)$$

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

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

Also, the input voltages of the two axes have the saturation constraint form (13) with $\mathcal{M}(t) = [0.6, 2]^\top$. For simplicity, the weighting matrices Q_i, $\hat{Q}_i$, S and R are chosen to be diagonal matrices.

5.3. Performance of the proposed algorithm

First, the performance of Algorithm 1 for the unconstrained case is considered. The system constraints are provisionally removed by setting $\Omega = L_2^\ell[0,\ T]$ and $\Phi = L_2^m[0,\ T]$. The algorithm is then applied to this task for 100 ILC trials using $\hat{Q}_i = 100,000I$, $Q_i = 500,000I$, $S = 10,000I$ and $R = I$. The final converged hybrid output trajectory is plotted in Figure 4, and it is clear that the converged hybrid output trajectory accurately follows each line segment of the path. However, the overshoot problem occurs at two particular places on the path. The overshoot of the end-effector beyond its given working space might lead to the collision with the gantry frame. Also, the final converged inputs of the two axes are plotted in Figure 5, and it is obvious that at certain time sub-intervals the inputs exceed the given input saturation constraints of the gantry robot. These unconstrained results illustrate the necessity of applying the system constraints in practice.

To satisfy the practical limits of the given task, the input saturation constraint defined in (13) together with the hard output constraint defined in (54) are now added. Note that the parameters are chosen as the same values as those in the unconstrained case, and again 100 ILC trials are carried out. The corresponding final converged hybrid output trajectory is plotted as the dashed magenta trajectory in Figure 4. This result addresses the effect of output constraints on overcoming the overshoot problem, and also verifies the feasibility of the proposed algorithm to provide perfect tracking of the piecewise linear path. In addition, the dashed magenta trajectories are plotted in Figure 5 to represent the final converged input voltages of both axes. Compared to the previous unconstrained results, the employment of the input constraints guarantees the input voltages to stay within their saturation limits. These converged input and output results reveal that Algorithm 1 can guarantee high performance tracking as well as handle a mixed form of system constraints.

Experiments of Algorithm 1 are performed multiple times to further analyze and compare its convergence properties along the ILC trials. For each set of experiments, the parameters $\hat{Q}_i = 100,000I$, $S = 10,000I$ and $R = I$ are kept as constant values, but the value of Q_i is selected as $200,000I$, $300,000I$, $500,000I$, $800,000I$, and $1,200,000I$ respectively. Again, 100 ILC trials are carried out for each value of Q_i. To evaluate the accuracy of the proposed algorithm with a relative fair judgment, the concept of mean square error is used in this paper to represent the average path following square error at each unit time interval of

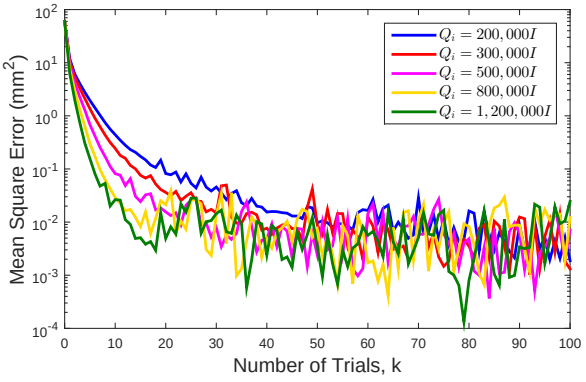

Figure 6: Mean square tracking error over 100 ILC trials with system constraints.

Table 1: Control effort comparison for generalized and classical ILC.

	Control Effort $(V^2 \cdot ms)$
$Q_i = 200,000I$	1,819.9
$Q_i = 300,000I$	1,820.3
$Q_i = 500,000I$	1,812.8
$Q_i = 800,000I$	1,810.9
$Q_i = 1,200,000I$	1,807.8
Classical ILC	2,146.9

$0.01s$. The mean square value of the error e_k^s is recorded for each trial, and then plotted in Figure 6. The convergence rate has a proportional relationship with respect to the weighting value Q_i. Moreover, all curves converge to around 0.01 in terms of mean square error, which also shows the ability of this algorithm to improve tracking accuracy and maintain robustness along the trials.

The converged control effort with different values of Q_i are listed in Table 1. Due to the fact that the end-effector of many practical applications, e.g. laser cutting and welding, moves at a constant speed along its path, a classical ILC setup assuming constant moving speed is considered and applied to the same task, and its control effort is also listed in the same table for comparison. This table shows that an approximate 16% control effort reduction is achieved using the proposed algorithm comparing to classical ILC, as its problem formulation releases significant design freedom. Furthermore, experiments with other combinations of Q_i, $\hat{Q}_i$, S and R yield similar convergence performance to the results in Figure 6. For brevity, these results are omitted.

6. Conclusion and future work

In this paper, a generalized framework with substantial novelty is formulated for a wide class of ILC tracking problems, which involve intermediate position tracking, subinterval path following and mixed system constraints handling. Within this framework, a generalized ILC algorithm is proposed based on a successive projection scheme, and attractive convergence properties have been highlighted. A computationally efficient causal feedback plus feedforward implementation of this algorithm has also been formulated. This algorithm is verified on a gantry robot test platform by performing a piecewise linear path following task with stipulated input and output constraints. The experimental results illustrate the practical efficacy of the proposed algorithm, which provides the first general ILC solution for continuous time systems.

Although the experimental results reveal that the algorithm has a certain degree of robustness against model uncertainty and disturbance, a rigorous robust analysis like that in [26] will next be performed on this algorithm. Other alternative path following tasks besides the piecewise linear case will also be applied. These constitute the focus of future research and will be reported separately.

7. Acknowledgment

This project is partially funded by the China Scholarship Council, the ZZU-Southampton Collaborative Research Project under Grant 16306/01, the Royal Society International Exchanges Award under Grant IE161369, the National Natural Science Foundation of China under Grant 61473265, and the Science and Technology Innovation Research Team Support Plan of Henan Province under Grant 17IRTSTHN013.

Appendix A. Proof of Algorithm 1

Since $X = \hat{H}$ and $x = (y^e, y, u)$, the projection operator P_{S_1} can be computed as

$$
\begin{aligned}
P_{S_1}(x) &= \arg\min_{\hat{x}\in S_1} \|\hat{x} - x\|_X^2 \qquad \text{(A.1)}\\
&= \arg\min_{(\hat{y}^e,\hat{y},\hat{u})\in\hat{H}} \left\| \begin{pmatrix} \hat{y}^e \\ \hat{y} \\ \hat{u} \end{pmatrix} - \begin{pmatrix} y^e \\ y \\ u \end{pmatrix} \right\|^2_{\{\tilde{Q},\ S,\ R\}},\\
&\text{s.t. } \hat{y}^e = G^e\hat{u},\ \ \hat{y} = G\hat{u}\\
&= \arg\min_{(\hat{y}^e,\hat{y},\hat{u})\in\hat{H}} \|\hat{y}^e - y^e\|_{\tilde{Q}}^2 + \|\hat{y} - y\|_S^2 + \|\hat{u} - u\|_R^2,\\
&\text{s.t. } \hat{y}^e = G^e\hat{u},\ \ \hat{y} = G\hat{u}\\
&= \arg\min_{\hat{u}} \|G^e\hat{u} - y^e\|_{\tilde{Q}}^2 + \|G\hat{u} - y\|_S^2 + \|\hat{u} - u\|_R^2,
\end{aligned}
$$

which gives rise to the solution $\hat{u} = u^*$ with respect to

$$u^* = u + G^{s*}(I + G^s G^{s*})^{-1} \begin{bmatrix} y^e - G^e u \\ y - Gu \end{bmatrix}. \qquad \text{(A.2)}$$

It follows from the definition (20) such that

$$P_{S_1}(x) = (G^e u^*, Gu^*, u^*), \qquad \text{(A.3)}$$

where u^* is in form of (A.2). Similarly, the projection operator P_{S_2} can be computed as

$$\begin{aligned} P_{S_2}(x) &= \arg\min_{\hat{x}\in S_2} \|\hat{x}-x\|_X^2 \\ &= \arg\min_{(\hat{y}^e,\hat{y},\hat{u})\in\hat{H}} \left\| \begin{pmatrix} \hat{y}^e \\ \hat{y} \\ \hat{u} \end{pmatrix} - \begin{pmatrix} y^e \\ y \\ u \end{pmatrix} \right\|^2_{\{\tilde{Q},\ S,\ R\}}, \\ &\text{s.t. } \hat{y}^e = r^e,\ \hat{u}\in\Omega,\ \hat{y}\in\Phi \\ &= \arg\min_{(\hat{y}^e,\hat{y},\hat{u})\in\hat{H}} \|\hat{y}^e-y^e\|_{\tilde{Q}}^2 + \|\hat{y}-y\|_S^2 + \|\hat{u}-u\|_R^2, \\ &\text{s.t. } \hat{y}^e = r^e,\ \hat{u}\in\Omega,\ \hat{y}\in\Phi. \end{aligned} \tag{A.4}$$

In above problem (A.4), the variables $\hat{y}^e$, $\hat{y}$ and $\hat{u}$ are independent of one another, which means that the solution can be obtained by solving separate problems. Using (31) and (32), it follows that

$$P_{S_2}(x) = (r^e, P_\Phi(y), P_\Omega(u)). \tag{A.5}$$

Note that the elements $\tilde{x}_{k+1}$ and x_{k+1} are updated using projection operators P_{S_1} and P_{S_2} at the k^{th} trial. Based on solutions of the projection operators, the update (25) in Lemma 1 can be obtained with $x_k = (r^e, \tilde{r}_k, u_k)$ and $\tilde{x}_k = (\tilde{y}_k^e, \tilde{y}_k, \tilde{u}_k)$. For $\tilde{x}_{k+1} = P_{S_1}(x_k)$, it follows from the solution (A.3) that

$$\begin{aligned} \tilde{u}_{k+1} &= u_k + G^{s*}(I+G^sG^{s*})^{-1}\begin{bmatrix} r^e - y_k^e \\ \tilde{r}_k - y_k \end{bmatrix}, \\ \tilde{y}_{k+1}^e &= G^e\tilde{u}_{k+1},\ \tilde{y}_{k+1} = G\tilde{u}_{k+1},\ k\geqslant 0, \end{aligned} \tag{A.6}$$

and for $x_{k+1} = P_{S_2}(\tilde{x}_{k+1})$, it follows from (A.5) that

$$\tilde{r}_{k+1} = P_\Phi(\tilde{y}_{k+1}),\ u_{k+1} = P_\Omega(\tilde{u}_{k+1}),\ k\geqslant 0. \tag{A.7}$$

Therefore, the successive projection method combines the solutions (A.6) and (A.7) to yield Algorithm 1, which updates the input sequence $\{u_k\}$ along the trial under 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 1

If $S_1 \cap S_2 \neq \varnothing$, there exists intersection between the two sets, minimum distance between the two sets is zero and perfect tracking under constraints is possible. From the condition (28), the two sequences $\{(\tilde{y}_k^e, \tilde{y}_k, \tilde{u}_k)\}_{k\geqslant 0}$ and $\{(r^e, \tilde{r}_k, u_k)\}_{k\geqslant 0}$ attain a distance of zero, i.e.

$$\lim_{k\to\infty} \|\tilde{y}_k^e - r^e\|_{\tilde{Q}}^2 + \|\tilde{y}_k - \tilde{r}_k\|_S^2 + \|\tilde{u}_k - u_k\|_R^2 = 0. \tag{B.1}$$

It follows that

$$\lim_{k\to\infty} y_k^e = \lim_{k\to\infty} G^e(u_k + \tilde{u}_k - u_k) = \lim_{k\to\infty} \tilde{y}_k^e = r^e, \tag{B.2}$$

which gives rise to the design objective (36). 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 $J_{k+1} \leqslant J_k$.

When the limit of the input signal u_k exists, the constrained condition (39) is satisfied by the definition of the projection operators P_Ω and P_Φ. Moreover, substitute $x^* = (r^e, y^*, u^*)$ and $x_k = (r^e, \tilde{r}_k, u_k)$ into the monotonic convergence condition (29) to give $\tilde{J}_{k+1} \leqslant \tilde{J}_k$, which completes the proof.

Appendix C. Proof of Corollary 1

In the absence of system constraints, the sets S_1 and S_2 collapse to

$$\begin{aligned} S_1 &= \{(y^e, y, u) \in H \times L_2^\ell[0,\ T] : y^e = G^e u,\ y = Gu\}, \\ S_2 &= \{(y^e, y, u) \in H \times L_2^\ell[0,\ T] : y^e = r^e\}. \end{aligned}$$

Substitute $\hat{x} = (r^e, G\hat{u}, \hat{u}) \in S_1 \cap S_2$ and $x_k = (r^e, Gu_k, u_k)$ into monotonic convergence condition (29), which yields $\hat{J}_{k+1} \leqslant \hat{J}_k,\ \forall\ k \geqslant 0$. The proof of minimum control effort follows from Theorem 1 in [52] by considering the Lagrangian associated with the minimum control effort as

$$\mathcal{L}(u, \lambda) = \|u\|_R^2 + 2\langle \lambda,\ r^e - G^e u\rangle_{\tilde{Q}}, \tag{C.1}$$

where λ is the Lagrange multiplier. The above problem has a unique stationary point $u_\infty = G^{e*}\lambda$ and $r^e = G^e u_\infty$, which gives rise to $r^e = G^eG^{e*}\lambda$. The stationary point solution solves the minimum control effort problem. If there are no system constraints engaged and initial input is zero, the update law of input u_k is

$$\begin{aligned} u_k &= G^{e*}\sum_{i=1}^{k} X^i e_0^e = G^{e*}\sum_{i=0}^{k-1} X^i(I-X)\lambda_0 \\ &= G^{e*}(I-X^k)\lambda_0, \end{aligned} \tag{C.2}$$

where $X = (I + G^eG^{e*})^{-1}$ and $e_0^e = r^e = G^eG^{e*}\lambda_0$. It follows that u_k converges to an input $\hat{u}_\infty = G^{e*}\lambda_0$ and $r^e = G^eG^{e*}\lambda_0$, which is exactly 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 its minimum value, which completes the proof.

Appendix D. Proof of Theorem 2

The condition $S_1 \cap S_2 = \varnothing$ implies that perfect tracking is impossible under given system constraints. However, it still follows Theorem 1 that the two sequences $\{\tilde{x}_k\}$ and $\{x_k\}$ attain the minimum distance between the two sets as shown in (42).

In addition, the input u_k generated by projection operator P_Ω belongs to the input constraint set Ω and the proof of monotonic convergence with respect to J_k follows from a similar proof to that of Theorem 1.

Appendix E. Proof of Proposition 1

To formulate the solution, the next lemma is needed.

Lemma 2. *The Hilbert adjoint operator* $G^{s*}:(\omega,\nu,y)\in\tilde{H}\to u\in L_2^\ell[0,\ T]$ *is defined by the inner product*

$$\langle(\omega,\nu,y),\ G^s u\rangle_{\{\tilde{Q},S\}}=\langle G^{s*}(\omega,\nu,y),\ u\rangle_R\,, \tag{E.1}$$

and has the analytic form

$$u(t)=R^{-1}B^\top p(t), \tag{E.2}$$

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

$$\dot{p}(t)=A^\top p(t)+C^\top(P_i^\top\hat{Q}_i\nu_i(t)+Sy(t)), \tag{E.3}$$

for $t\in[t_{i-1},\ t_i]$, $i=1,\ldots,M$, *with boundary conditions*

$$\begin{aligned}&p(t_i^-)=p(t_i^+)+C^\top F_i^\top Q_i\omega_i,\ i=0,\ldots,M,\\ &p(T)=0.\end{aligned} \tag{E.4}$$

See Appendix F for the proof. The above lemma addresses the analytic representation of the adjoint operator G^{s*}. Using this lemma, the feedback plus feedforward implementation of (30) can hence be worked out. The ILC update (30) is equivalently written as

$$u_{k+1}(t)=u_k(t)+G^{s*}\tilde{e}^s_{k+1}(t), \tag{E.5}$$

where $\tilde{e}^s_{k+1}=[e^e_{k+1},\ \hat{e}_{k+1}]^\top$ and $\hat{e}_{k+1}=\tilde{r}_k-y_{k+1}$. From Lemma 2, $G^{s*}\tilde{e}^s_{k+1}(t)$ is computed using

$$G^{s*}\tilde{e}^s_{k+1}(t)=R^{-1}B^\top p_{k+1}(t), \tag{E.6}$$

where the costate $p_{k+1}(t)$ is computed in reverse time as

$$\dot{p}_{k+1}(t)=-A^\top p_{k+1}(t)-C^\top(P_i^\top\hat{Q}_iP_ie_{k+1}(t)+S\hat{e}_{k+1}(t)),$$

for $t\in[t_{i-1},\ t_i]$, $i=1,\ldots,M$, with boundary conditions

$$\begin{aligned}&p_{k+1}(t_i-)=p_{k+1}(t_i+)+C^\top F_i^\top Q_iF_ie_{k+1}(t_i), i=0,\ldots,M,\\ &p_{k+1}(T)=0.\end{aligned} \tag{E.7}$$

Substituting (E.6) into (E.5) yields the solution (43).

Assuming full state knowledge, the costate equation (E.6) yields a causal implementation

$$p_{k+1}(t)=-K(t)(x_{k+1}(t)-x_k(t))+\xi_{k+1}(t). \tag{E.8}$$

Then use the method proposed in [53] such that it follows from (1), (43) and (E.8) such that

$$\begin{aligned}\dot{x}_{k+1}(t)-\dot{x}_k(t)&=A(x_{k+1}(t)-x_k(t))+B(u_{k+1}(t)-u_k(t))\\ &=A(x_{k+1}(t)-x_k(t))+BR^{-1}B^\top p_{k+1}(t)\\ &=(A-BR^{-1}B^\top K(t))(x_{k+1}(t)-x_k(t))\\ &\quad+BR^{-1}B^\top\xi_{k+1}(t).\end{aligned} \tag{E.9}$$

Then substitute (E.8) and (E.9) into the costate equation (E.6) to yield an equation of the form

$$\begin{aligned}&\mathcal{H}(A,B,C,S,\hat{Q}_i,P_i,R,K(t),\dot{K}(t))[x_{k+1}(t)-x_k(t)]\\ &=\mathcal{G}(A,B,C,S,\hat{Q}_i,P_i,R,\xi_{k+1}(t),\dot{\xi}_{k+1}(t),e_k(t),\hat{e}_k(t)),\end{aligned}$$

where $\mathcal{H}(\cdot)$ and $\mathcal{G}(\cdot)$ are functions of their arguments and independent of the states. If both functions are both zero, the above equation holds independently of the difference of in states. This yields the Riccati equation $K(t)$ and the optimal predictor $\xi_{k+1}(t)$ in (45) and (47) respectively. Considering the boundary conditions in (E.7), there is an extra term at the end, i.e. $C^\top F_i^\top Q_iF_ie_{k+1}(t_i)$. Note that

$$\begin{aligned}e_{k+1}(t_i)&=r_i-Cx_{k+1}(t_i)\\ &=e_k(t_i)-C(x_{k+1}(t_i)-x_k(t_i)),\end{aligned} \tag{E.10}$$

which yields

$$\begin{aligned}C^\top F_i^\top Q_iF_ie_{k+1}(t_i)=&C^\top F_i^\top Q_iF_iC(x_{k+1}(t_i)-x_k(t_i))\\ &+C^\top F_iQ_iF_ie_k(t_i),\end{aligned} \tag{E.11}$$

and gives rise to the boundary conditions in (46) and (48).

Appendix F. Proof of Lemma 2

To address the properties of the relevant adjoint operator G^{s*}, its inner product form

$$\langle(\omega,\nu,y),\ G^s u\rangle_{[Q]}=\langle G^{s*}(\omega,\nu,y),\ u\rangle_R \tag{F.1}$$

is needed, where $[Q]=\{\tilde{Q},\ S\}$. Note that G^s consists of the three parts FG, PG and G respectively, and their adjoints are computed separately as follows:

1). Adjoint Operator of FG: The operator FG has the following structure

$$FGu=[G_0u,\ldots,G_Mu]^\top, \tag{F.2}$$

where

$$G_iu=F_i\int_0^{t_i}Ce^{A(t_i-t)}Bu(t),\ i=1,\ldots,M. \tag{F.3}$$

Consider operator $G_i:L_2^\ell[0,\ T]\to\mathbb{R}^{f_i}$ via equation

$$\begin{aligned}\omega_i^\top Q_iG_iu&=\omega_i^\top Q_iF_i\int_0^{t_i}CA^{t_i-t}Bu(t)\\ &=\int_0^{t_i}(R^{-1}B(A^\top)^{t_i-t}C^\top F_i^\top Q_i\omega_i)^\top Ru(t),\end{aligned} \tag{F.4}$$

and equation

$$\omega_i^\top Q_iG_iu=\int_0^T((G_i^*\omega_i)(t))^\top Ru(t) \tag{F.5}$$

held by the definition of adjoint operator, i.e.

$$\langle\omega_i,\ G_iu\rangle_{Q_i}=\langle G_i^*\omega_i,\ u\rangle_R\,. \tag{F.6}$$

Hence, these equations give rise to the adjoint of G_i as

$$(G_i^*\omega_i)(t) = \begin{cases} R^{-1}Be^{A^\top(t_i-t)}C^\top F_i^\top Q_i\omega_i, & t \leqslant t_i, \\ 0, & t > t_i, \end{cases}$$

which can be further written as

$$(G_i^*\omega_i)(t) = R^{-1}B^\top p_i(t), \tag{F.7}$$

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

$$\dot{p}_i(t) = A^\top p_i(t),\ p_i(t_i) = C^\top Q_i\omega_i. \tag{F.8}$$

2). Adjoint Operator of G: The adjoint of G is defined by the inner product form

$$\langle y,\ Gu\rangle_S = \langle G^*y,\ u\rangle_R \tag{F.9}$$

as the map $u = G^*y$, i.e.

$$\begin{aligned} &(G^*y)(t) = R^{-1}B^\top p_{M+1}(t),\ p_{M+1}(N) = 0, \\ &\dot{p}_{M+1}(t) = A^\top p_{M+1}(t) + C^\top Sy(t). \end{aligned} \tag{F.10}$$

3). Adjoint Operator of PG: Note that PG is simply the composite map G and the map P such that

$$y(t) \to P_iy(t),\ t \in [t_{i-1},\ t_i],\ i = 1, \ldots, M. \tag{F.11}$$

It follows that the adjoint operator P_i^* is computed as

$$y(t) = S^{-1}P_i^\top \hat{Q}_i\nu_i(t), t \in [t_{i-1}, t_i], i = 1, \ldots, M, \tag{F.12}$$

from the inner product form

$$\langle \nu_i,\ P_iy\rangle_{\hat{Q}_i} = \langle P_i^*y,\ y\rangle_S\,. \tag{F.13}$$

Hence, the adjoint of PG defined by the relation $u = (PG)^*(\nu_1, \ldots, \nu_M)$ can be written as $(PG)^* = G^*P^*$, and computed as

$$\begin{aligned} &((PG)^*\nu)(t) = R^{-1}B^\top p_{M+2}(t),\ p_{M+2}(N) = 0 \\ &\dot{p}_{M+2}(t) = A^\top p_{M+2}(t) + C^\top P_i^\top \hat{Q}_i\nu_i(t), \\ &t \in [t_{i-1},\ t_i],\ i = 1, \ldots, M, \end{aligned} \tag{F.14}$$

by substituting (F.12) into (F.10).

The adjoint operator G^{s*} is the map $(\omega, \nu, y) \mapsto u$. Due to linearity, the adjoint operator G^{s*} can be expressed as the sum of the adjoints of FG, PG and G, i.e.

$$\begin{aligned} G^{s*}(\omega, \nu, y) &= \sum_{i=1}^{M}(G_i^*\omega_i)(t) + (G^*y)(t) + ((PG)^*\nu)(t) \\ &= R^{-1}B^\top p(t) \end{aligned} \tag{F.15}$$

as shown in (E.2), where $p(t) = \sum_{i=1}^{M+2} p_i(t)$. According to the representations of G_i^*, G^* and $(PG)^*$, the costate $p(t)$ in (E.2) is obtained together with its boundary conditions (E.4). These together give rise to the definition of the adjoint operator G^{s*}, which completes the proof.

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] J. H. Lee, K. S. Lee, Iterative learning control applied to batch processes: An overview, Control Engineering Practice 15 (2007) 1306–1318.

[5] H. Tao, W. Paszke, E. Rogers, H. Yang, K. Galkowski, Iterative learning fault-tolerant control for differential time-delay batch processes in finite frequency domains, Journal of Process Control 56 (2017) 112–128.

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

[7] K. Tan, S. Zhao, J.-X. Xu, Online automatic tuning of a proportional integral derivative controller based on an iterative learning control approach, IET Control Theory and Applications 1 (1) (2007) 90–96.

[8] S. Preitl, R.-E. Precup, Z. Preitl, S. Vaivoda, S. Kilyeni, J. K. Tar, Iterative feedback and learning control. servo systems applications, IFAC Proceedings Volumes.

[9] S. Z. Khong, D. Nesic, M. Krstic, Iterative learning control based on extremum seeking, Automatica 66 (2016) 238–245.

[10] T. Gluck, M. Blank, D. Buchl, A. Kugi, Convex constrained iterative learning control using projection: Application to a smart power switch, IEEE Transactions on Control Systems Technology 26 (5) (2018) 1818–1825.

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

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

[13] 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.

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

[15] 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.

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

[17] C. T. Freeman, Y. Tan, Iterative learning control with mixed constraints for point-to-point tracking, IEEE Transactions on Control Systems Technology 21 (3) (2013) 604–616.

[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] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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 (5) (2018) 1685–1698.

[26] Y. Chen, B. Chu, C. T. Freeman, A coordinate descent approach to optimal tracking time allocation in point-to-point ILC, Mechatronics 59 (2019) 25–34.

[27] P. Janssens, G. Pipeleers, J. Swevers, A data-driven constrained norm-optimal iterative learning control framework for LTI systems, IEEE Transactions on Control Systems Technology 21 (2) (2013) 546–551.

[28] M. Volckaert, M. Diehl, J. Swevers, Generalization of norm optimal ilc for nonlinear systems with constraints, Mechanical Systems and Signal Processing (2013) 280–296.

[29] S. Mishra, U. Topcu, M. Tomizuka, Optimization-based constrained iterative learning control, IEEE Transactions on Control Systems Technology 19 (6) (2011) 1613–1621.

[30] J.-X. Xu, J. Xu, T. H. Lee, Iterative learning control for systems with input deadzone, IEEE Transactions on Automatic Control 50 (9) (2005) 1455–1459.

[31] B. Chu, D. H. Owens, Iterative learning control for constrained linear systems, International Journal of Control 83 (7) (2010) 1397–1413.

[32] B. Chu, C. T. Freeman, D. H. Owens, A novel design framework for point-to-point ILC using successive projection, IEEE Transactions on Control Systems Technology 23 (3) (2015) 1156–1163.

[33] J.-X. Xu, X. Jin, State-constrained iterative learning control for a class of MIMO systems, IEEE Transactions on Automatic Control 58 (5) (2013) 1322–1327.

[34] X. Jin, Nonrepetitive leader-follower formation tracking for multiagent systems with LOS range and angle constraints using iterative learning control, IEEE Transactions on Cybernetics.

[35] X. Jin, Fault tolerant nonrepetitive trajectory tracking for MIMO output constrained nonlinear systems using iterative learning control, IEEE Transactions on Cybernetics.

[36] K. L. Moore, M. Ghosh, Y. Q. Chen, Spatial-based iterative learning control for motion control applications, Meccanica 42 (2007) 167–175.

[37] S. K. Sahoo, S. K. Panda, 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, FL, 2007, pp. 1–7.

[38] D. J. Hoelzle, K. L. Barton, On spatial iterative learning control via 2-D convolution: Stability analysis and computational efficiency, IEEE Transactions on Control Systems Technology 24 (4) (2016) 1504–1512.

[39] P. Janssens, W. V. Loock, G. Pipeleers, F. Debrouwere, J. Swevers, Iterative learning control for optimal path following problems, in: 52nd IEEE Conference on Decision and Control, Florence, Italy, 2013, pp. 6670 – 6675.

[40] K. D. Do, Z. P. Jiang, J. Pan, Robust adaptive path following of underactuated ships, Automatica 40 (2004) 929–944.

[41] A. P. Aguiar, J. P. Hespanha, Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty, IEEE Transactions on Automatic Control 52 (8) (2007) 1362–1379.

[42] D. R. Nelson, D. B. Barber, T. W. McLain, R. W. Beard, Vector field path following for miniature air vehicles, IEEE Transactions on Robotics 23 (3) (2007) 519–529.

[43] T. I. Fossen, K. Y. Pettersen, R. Galeazzi, Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces, IEEE Transactions on Control Systems Technology 23 (2) (2015) 820–827.

[44] D. Verscheure, B. Demeulenaere, J. Swevers, J. de Schutter, M. Diehl, Time-optimal path tracking for robots: A convex optimization approach, IEEE Transactions on Automatic Control 54 (10) (2009) 2318–2327.

[45] T. Lippa, S. Boyd, Minimum-time speed optimisation over a fixed path, International Journal of Control 87 (6) (2014) 1297–1311.

[46] Q.-C. Pham, S. Caron, P. Lertkultanon, Y. Nakamura, Admissible velocity propagation: Beyond quasi-static path planning for high-dimensional robots, International Journal of Robotics Research 36 (1) (2017) 44–67.

[47] H. Pham, Q.-C. Pham, A new approach to time-optimal path parameterization based on reachability analysis, IEEE Transactions on Robotics 34 (3) (2018) 645–659.

[48] Y. Chen, B. Chu, C. T. Freeman, Generalized iterative learning control using successive projection: Algorithm, convergence and experimental verification, IEEE Transactions on Control Systems Technology Early Access (2019) 1–13.

[49] Y. Chen, B. Chu, 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.

[50] D. H. Owens, R. P. Jones, Iterative solution of constrained differential/algebraic systems, International Journal of Control 27 (6) (1978) 957–964.

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

[52] D. H. Owens, C. T. Freeman, B. Chu, Generalized norm optimal iterative learning control with intermediate point and sub-interval tracking, International Journal of Automation and Computing 12 (3) (2015) 243–253.

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