Generalized Norm Optimal Iterative Learning Control: Constraint Handling

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

[*] *Department of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom,*
E-mail: { yc12u12, b.chu, cf } @ecs.soton.ac.uk.

Abstract: This paper proposes a novel control methodology to incorporate constraint handling within generalized iterative learning control (ILC), a overarching methodology which includes intermediate point and sub-interval tracking as special cases. The constrained generalized ILC design objective is first described, and then the design problem is formulated into a successive projection framework. This framework yields a constrained generalized ILC algorithm which embeds system input and output constraints. Convergence analysis of the algorithm is performed and supported by rigorous proofs. The algorithm is verified using a gantry robot experimental platform, whose results reveal its practical efficacy and robustness against plant uncertainty.

Keywords: iterative learning control, constraint handling.

1. INTRODUCTION

ILC is a high performance control design methodology to improve the tracking accuracy of a system repeating the same task over a finite time horizon. By updating the control input based on the data from previous trials, ILC theoretically enables the tracking error to converge to zero after sufficient trials. This feature has led ILC to be widely applied to precision industrial tasks, such as robotic systems (Hladowski et al. (2010); Norrlof (2002)), chemical batch processing (Lee and Lee (2007)) and stroke rehabilitation (Freeman (2016)). See Bristow et al. (2006) for a detailed overview.

In the classical ILC framework, the tracking requirement is to follow a given motion profile defined over a finite time horizon. However, for some applications such as robotic pick-and-place tasks, the output trajectory is only critical at a finite number of 'intermediate' time instants, and by eliminating the unnecessary output constraints, significant control design flexibility can be exploited to embed additional performance. A novel ILC framework termed intermediate point ILC has been proposed to address this problem, and is formulated in Freeman et al. (2011); Freeman (2012); Freeman and Tan (2013); Chu et al. (2015). Subsequent research has expanded the intermediate point ILC framework to allow simultaneous tracking of both the reference along the whole time horizon and the intermediate points, in which the tracking error on the latter has a faster convergence rate than that elsewhere along the time horizon. This ILC framework is named norm optimal ILC with intermediate point weighting and is studied in Owens et al. (2013).

Recent research has expanded the framework to tackle the spatial ILC problem. Here the output is required only to follow a path (i.e. a mapping between output variables), with no specific timing imposed. This problem addresses the needs of automation tasks such as welding, laser cutting and additive manufacturing. The ILC framework was generalized in Owens et al. (2015) to embed tracking of any subset of outputs on defined sub-intervals of the time duration, and at an arbitrary number of intermediate points. Termed 'generalized ILC', it is equivalent to combining intermediate point ILC with sub-interval tracking. This paper then showed how linear constraints between outputs could be used to enforce tracking along lines or planes with no *a priori* timing constraints while also minimizing control effort. This hence provided a formal solution to the spatial tracking problem.

However, neither hard input nor output constraints have been embedded in the generalized ILC framework, which exist widely in practice and have significant impact on industrial manufacture, e.g. to prevent system damage. In particular, the absence of output constraints causes potential overshoot, e.g. the output goes beyond the boundary of its acceptable space. Hence it is critical to embed both input and output constraints in the generalized ILC framework. Input constraints have been studied in classical ILC and intermediate point ILC (Janssens et al. (2013); Mishra et al. (2011); Bolder and Oomen (2016); Freeman et al. (2011)), and the successive projection method proposed in von Neumann (1950) was used to solve the constrained ILC problem in Chu and Owens (2009, 2010); Chu et al. (2015). ILC in high order non-linear systems with output constraints is studied in Jin and Xu (2013). However, out put constraints remain an open problem.

In this paper, constraints are incorporated into generalized ILC. As well as solving the spatial ILC problem, this provides a constrained solution to the special cases of ILC problems embedded within it, e.g. classical ILC, intermediate point ILC, generalized ILC and sub-interval ILC. The successive projection method is used to design a control algorithm which can be easily implemented in practice and automatically provides the solution of the general ILC problem. The proposed algorithm is then verified experimentally on a three-axis gantry robot test platform to establish robust performance and practical effectiveness. This replicates industrial conditions which unavoidably contains model uncertainty.

The paper is organized as follows: Section 2 formulates the generalized constrained ILC problem. This ILC design problem is then solved using successive projection in Section 3, and a practical implementation algorithm is proposed, whose convergence is analyzed rigorously in Section 4. This algorithm is then experimentally verified on a gantry robot platform in Section 5, and finally conclusions and future work are outlined in Section 6.

2. PROBLEM FORMULATION

This section introduces the system dynamics and defines the general tracking requirement. Then input and output constraints are introduced, to yield a general ILC problem formulation for constrained intermediate point and sub-interval tracking.

2.1 System Dynamics

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

$$\begin{aligned} \dot{x}_k(t) &= Ax_k(t) + Bu_k(t), \quad x_k(0) = x_0 \\ y_k(t) &= Cx_k(t),\ t \in [0,\ T],\ T < \infty \end{aligned} \tag{1}$$

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

$$\begin{aligned} &y_k = Gu_k + d,\ G : L_2^\ell[0,\ T] \to L_2^m[0,\ T] \\ &y_k,\ d \in L_2^m[0,\ T],\ u_k \in L_2^\ell[0,\ T] \end{aligned} \tag{2}$$

where the input and output Hilbert spaces $L_2^\ell[0,\ T]$ and $L_2^m[0,\ T]$ are defined with inner products and associated induced norms

$$\langle u,\ v\rangle_R = \int_0^T u^\top(t)Rv(t)\mathrm{d}t,\ \|u\|_R^2 = \langle u,\ u\rangle_R \tag{3}$$

$$\langle x,\ y\rangle_S = \int_0^T x^\top(t)Sy(t)\mathrm{d}t,\ \|y\|_S^2 = \langle y,\ y\rangle_S \tag{4}$$

in which $R \in \mathbb{S}_{++}^\ell$ and $S \in \mathbb{S}_{++}^m$ ($\mathbb{S}_{++}^n$ denotes the set of all $n \times n$ real positive definite matrices). The convolution operator G and signal d take the form

$$(Gu_k)(t) = \int_0^t Ce^{A(t-s)}Bu_k(s)\mathrm{d}s,\ d(t) = Ce^{At}x_0 \tag{5}$$

where, without loss of generality, the constant $d(t)$ can be absorbed into the reference to give $x_0 = 0$, $d(t) = 0$ while performing tracking tasks and see Chu et al. (2015) for more information.

2.2 Generalized ILC Design Objective

The classical ILC design objective is to update the input signal, u_k, such that the associated output, y_k, ultimately tracks a given reference trajectory, r, defined over the whole time horizon, i.e.

$$\lim_{k\to\infty} y_k = r \tag{6}$$

To solve the classical ILC problem, the tracking error $e_k = r - y_k$ is employed within the following updating law:

$$u_{k+1} = f(u_k, e_k). \tag{7}$$

In contrast, the intermediate point ILC design objective is to update the input signal, u_k, such that the output, y_k, tracks the reference, r, at a subset of time instants, i.e.

$$\lim_{k\to\infty} y_k(t_i) = r(t_i),\ i = 1, \ldots, M \tag{8}$$

where

$$0 = t_0 < t_1 < \cdots < t_M = T \tag{9}$$

are distinct intermediate time instants in $[0,\ T]$. The solution of the intermediate point ILC problem can be also obtained using an updating law of form (7).

The generalized ILC framework proposed in Owens et al. (2015) subsumes both by embedding intermediate point and sub-interval tracking as follows: for any signal $\beta \in L_2^m[0,\ T]$ in output space define the linear mapping

$$\beta \mapsto \beta^e : \beta^e = \begin{bmatrix} F\beta \\ P\beta \end{bmatrix} \in H \tag{10}$$

where H is the Hilbert space denoted by

$$H = \mathbb{R}^f \times \cdots \times \mathbb{R}^f \times L_2^p[t_0, t_1] \times \cdots \times L_2^p[t_{M-1}, t_M] \tag{11}$$

with inner product and associated induced norm

$$\begin{aligned} \langle(\omega, \nu),\ (\mu, \lambda)\rangle_{\tilde{Q}} &= \sum_{i=1}^M \omega_i^\top Q_i \mu_i + \sum_{i=1}^M \int_{t_{i-1}}^{t_i} \nu_i^\top(t)\hat{Q}_i\lambda_i(t)\mathrm{d}t \\ \|(\omega, \nu)\|_{\tilde{Q}}^2 &= \langle\omega,\ \omega\rangle_{[Q]} + \langle\nu,\ \nu\rangle_{[\hat{Q}]} \end{aligned} \tag{12}$$

in which

$$\begin{aligned} &\omega = [\omega_1,\ \omega_2, \ldots,\ \omega_M]^\top,\ \mu = [\mu_1,\ \mu_2, \ldots,\ \mu_M]^\top \\ &\nu = [\nu_1,\ \nu_2, \ldots,\ \nu_M]^\top,\ \lambda = [\lambda_1,\ \lambda_2, \ldots,\ \lambda_M]^\top \\ &\omega_i,\ \mu_i \in \mathbb{R}^f,\ \nu_i,\ \lambda_i \in L_2^p[t_{i-1},\ t_i],\ i = 1, \ldots, M \\ &\tilde{Q} = \{Q_1,\ Q_2, \ldots,\ Q_M,\ \hat{Q}_1,\ \hat{Q}_2, \ldots,\ \hat{Q}_M\} \end{aligned}$$

and $[Q]$, $[\hat{Q}]$ denotes the data sets $\{Q_1,\ Q_2, \ldots,\ Q_M\}$, $\{\hat{Q}_1,\ \hat{Q}_2, \ldots,\ \hat{Q}_M\}$ where for $i = 1, \ldots, M$ each $Q_i \in \mathbb{S}_{++}^f$ and $\hat{Q}_i \in \mathbb{S}_{++}^p$. Operator F is defined as

$$F\beta = \begin{bmatrix} F_1\beta(t_1) \\ \vdots \\ F_M\beta(t_M) \end{bmatrix},\ F_i\beta(t_i) \in \mathbb{R}^f,\ 1 \leqslant i \leqslant M \tag{13}$$

where F_i is an $f \times m$ matrix of full row rank which selects components of β that are important in the problem to be specified at $t = t_i$. Also, operator P is defined as

$$P\beta = \begin{bmatrix} (P\beta)_1 \\ \vdots \\ (P\beta)_M \end{bmatrix},\ (P\beta)_i \in L_2^p[t_{i-1},\ t_i],\ 1 \leqslant i \leqslant M \tag{14}$$

with

$$(P\beta)_i = P_i\beta(t),\ t \in [t_{i-1},\ t_i],\ 1 \leqslant i \leqslant M$$

where P_i is an $p \times m$ matrix of full row rank which extracts the required linear combination of components of β.

From definitions (13) and (14), it follows that the 'extended output' y^e comprises a subset of plant outputs defined over sub-intervals of the task duration, together with a subset of outputs at distinct intermediate points. The dynamics of the extended system can therefore be modelled by

$$y^e = G^e u = (Gu)^e = \begin{bmatrix} FGu \\ PGu \end{bmatrix} \quad (15)$$

where $G^e : L_2^\ell[0,T] \to H$ is a linear operator. In addition, the 'extended reference' r^e is defined as

$$r^e = \begin{bmatrix} Fr \\ Pr \end{bmatrix} \in H \quad (16)$$

which represents the requirement of intermediate point plus sub-interval tracking.

Using the extended output (15) and reference (16), the generalized ILC design objective is defined as

$$\lim_{k\to\infty} y_k^e = r^e \quad (17)$$

and its solution can be obtained using a generalized norm optimal ILC updating law as follows:

$$u_{k+1} = \arg\min_u \|r^e - G^e u\|_{\hat{Q}}^2 + \|u - u_k\|_R^2, \ k \geqslant 0. \quad (18)$$

Note that the above updating law is the extension of norm optimal ILC. This is not trivial due to the special structure of linear operator G^e. See Owens et al. (2015) for more implementation information.

Remark 1. By setting the values of Q, $\hat{Q}$, F and P appropriately, the generalized ILC framework collapses to well known ILC frameworks, e.g. 1) $Q_i = 0$, $P_i = I$, classical ILC; 2) $\hat{Q}_i = 0$, $F_i = I$, intermediate point ILC.

2.3 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 Ω typically assumes one of the following forms:

Input saturation constraint

$$\Omega = \{u(t) \in \mathbb{R}^\ell : |u(t)| \preceq M(t), \ t \in [0, \ T]\}, \quad (19)$$

Input energy constraint

$$\Omega = \{u(t) \in \mathbb{R}^\ell : \int_0^T (u(t))^\top u(t) \mathrm{d}t \leqslant M, \ t \in [0, \ T]\}, \quad (20)$$

and similarly the output constraint set Φ has the forms:

Output saturation constraint

$$\Phi = \{y(t) \in \mathbb{R}^m : |y(t)| \preceq N(t), \ t \in [0, \ T]\}, \quad (21)$$

Output polyhedral constraint

$$\Phi = \{y(t) \in \mathbb{R}^m : a_i^\top y(t) \leqslant b_i, \ a_i \in \mathbb{R}^m, \ b_i \in \mathbb{R}, \ i = 1, \ldots, M, \ t \in [0, \ T]\}. \quad (22)$$

In particular, the latter constraint restricts the system output to a specified convex region, and can be used to solve the overshoot problem which currently limits spatial ILC. Overshoot beyond the desired region may cause damage to the device and reduce the product quality, e.g. laser cutting. The above constraints, combined with the generalized ILC framework form a powerful comprehensive solution to constrained ILC. The generalized design objective accordingly founds a combined updating law in form of (7) such that

$$\lim_{k\to\infty} y_k^e = r^e, \ \lim_{k\to\infty} y_k \in \Phi, \ u_k \in \Omega. \quad (23)$$

3. AN ILC DESIGN FRAMEWORK USING SUCCESSIVE PROJECTION

A constrained generalized ILC algorithm based on successive projection is now formulated to solve (23).

3.1 Successive Projection Interpretation

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

$$S_1 = \{(y^e, y, u) \in \hat{H} : y^e = G^e u, \ y = Gu\} \quad (24)$$

$$S_2 = \{(y^e, y, u) \in \hat{H} : y^e = r^e, \ u \in \Omega, \ y \in \Phi\} \quad (25)$$

where S_1 represent the plant dynamics and S_2 represents the constrained tracking signal requirements; $\hat{H}$ is the Hilbert space denoted by

$$\hat{H} = \mathbb{R}^f \times \cdots \times \mathbb{R}^f \times L_2^p[t_0, \ t_1] \times \cdots \times L_2^p[t_{M-1}, \ t_M] \times L_2^m[0, \ T] \times L_2^\ell[0, \ T] \quad (26)$$

whose inner product and associated induced norm are naturally derived from (3), (4) and (12).

The problem can be solved by the method of successive projection. The basic successive projection scheme is illustrated in Figure 1, with convergence performance and conditions defined in the next theorem.

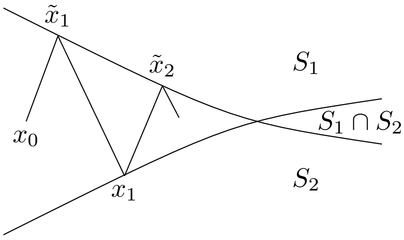

Fig. 1. Illustration of the successive projection algorithm.

Theorem 2. Owens and Jones (1978) Let $S_1 \subset \hat{H}$, $S_2 \subset \hat{H}$, be two closed convex sets in a Hilbert space $\hat{H}$ with $S_1 \cap S_2$ nonempty. Define orthogonal projection operators $P_{S_1} : \hat{x} \in \hat{H} \to x \in S_1$ and $P_{S_2} : \hat{x} \in \hat{H} \to x \in S_2$ as

$$P_{S_1}(x) = \arg \inf_{\hat{x}\in S_1} \|\hat{x} - x\|_X^2, \quad (27)$$

$$P_{S_2}(x) = \arg \inf_{\hat{x}\in S_2} \|\hat{x} - x\|_X^2. \quad (28)$$

Then given the initial guess $x_0 \in \hat{H}$, the sequences $\{\tilde{x}_k\}$ and $\{x_k\}$ generated by

$$\tilde{x}_{k+1} = P_{S_1}(x_k), \ x_{k+1} = P_{S_2}(\tilde{x}_{k+1}), \ k \geqslant 0 \quad (29)$$

are uniquely defined for each $x_0 \in \hat{H}$, and for each $\epsilon > 0$, there exists an integer N such that for $k > N$

$$\|\tilde{x}_{k+1} - x_k\|_X^2 < \epsilon. \quad (30)$$

Furthermore, if the intersection of the two sets is empty, i.e. $S_1 \cap S_2 = \varnothing$, the distance between the two sequences

$\{\tilde{x}_k\}$ and $\{x_k\}$ converges to the minimum distance between the two sets S_1 and S_2, i.e.

$$\inf_{\tilde{x}\in S_1, x\in S_2} \|\tilde{x} - x\|_X^2 . \tag{31}$$

3.2 Generalized ILC with Constraint Handling

The direct application of Theorem 2 to the constrained generalized ILC problem (23) yields the next algorithm.

Algorithm 1. Given system dynamics $S(A, B, C)$, input constraint set Ω, output constraint set Φ, extended reference r^e, any initial input signal $u_0 \in \Omega$, initial value $\tilde{r}_0 \in \Phi$, the input sequence $\{u_k\}$ defined by the updating law

$$\tilde{u}_{k+1} = u_k + G^{s*}(I + G^s G^{s*})^{-1} e_k^s \tag{32}$$

followed by the projections

$$u_{k+1} = P_\Omega(\tilde{u}_{k+1}),\ \tilde{r}_{k+1} = P_\Phi(\tilde{y}_{k+1}) \tag{33}$$

iteratively solves the constrained generalized ILC problem, where P_Φ and P_Ω are the orthogonal projection operators to the sets Φ and Ω defined as

$$P_\Phi(y) = \arg\inf_{z\in\Phi} \|z - y\|_S^2 ,\ P_\Omega(u) = \arg\inf_{z\in\Omega} \|z - u\|_R^2 \tag{34}$$

respectively, the linear operator $G^s : L_2^\ell[0,\ T] \to H \times L_2^m[0, T]$ and the error e_k^s are defined as

$$G^s u = \begin{bmatrix} G^e u \\ Gu \end{bmatrix},\ e_k^s = \begin{bmatrix} r^e - y_k^e \\ \tilde{r}_k - y_k \end{bmatrix} \tag{35}$$

and G^{s*} is the Hilbert adjoint operator of G^s..

Proof. To apply Theorem 2 to constrained generalized ILC problem (23), we first compute the necessary projections. From the definition of Hilbert space $\hat{H}$ in (26), denote $x = (y^e, y, u)$ to be an element belonging to $\hat{H}$. The projection operator P_{S_1} in Theorem 2 is hence

$$\begin{aligned}
P_{S_1}(x) &= \arg\inf_{\hat{x}\in S_1} \|\hat{x} - x\|_X^2 \\
&= \arg\inf_{(\hat{y}^e,\hat{y},\hat{u})\in\hat{H}} \left\| \begin{pmatrix} \hat{y}^e \\ \hat{y} \\ \hat{u} \end{pmatrix} - \begin{pmatrix} y^e \\ y \\ u \end{pmatrix} \right\|_{\{\tilde{Q},\ S,\ R\}}^2 , \\
&\quad \text{s.t. } \hat{y}^e = G^e\hat{u},\ \hat{y} = G\hat{u} \\
&= \arg\inf_{(\hat{y}^e,\hat{y},\hat{u})\in\hat{H}} \|\hat{y}^e - y^e\|_{\tilde{Q}}^2 + \|\hat{y} - y\|_S^2 + \|\hat{u} - u\|_R^2 , \\
&\quad \text{s.t. } \hat{y}^e = G^e\hat{u},\ \hat{y} = G\hat{u}. \\
&= \inf_{\hat{u}} \|G^e\hat{u} - y^e\|_{\tilde{Q}}^2 + \|G\hat{u} - y\|_S^2 + \|\hat{u} - u\|_R^2 .
\end{aligned} \tag{36}$$

Optimization problem (36) yields solution $\hat{u} = u^*$ with

$$u^* = u + G^{s*}(I + G^s G^{s*})^{-1} \begin{bmatrix} y^e - G^e u \\ y - Gu \end{bmatrix}. \tag{37}$$

It follows from the definition (24) that

$$P_{S_1}(x) = \begin{pmatrix} G^e u^* \\ Gu^* \\ u^* \end{pmatrix} \tag{38}$$

where u^* is given by (37). Performing a similar procedure for projection operator P_{S_2} yields

$$\begin{aligned}
P_{S_2}(x) &= \arg\inf_{\hat{x}\in S_2} \|\hat{x} - x\|_X^2 \\
&= \arg\inf_{(\hat{y}^e,\hat{y},\hat{u})\in\hat{H}} \left\| \begin{pmatrix} \hat{y}^e \\ \hat{y} \\ \hat{u} \end{pmatrix} - \begin{pmatrix} y^e \\ y \\ u \end{pmatrix} \right\|_{\{\tilde{Q},\ S,\ R\}}^2 , \\
&\quad \text{s.t. } \hat{y}^e = r^e,\ \hat{u} \in \Omega,\ \hat{y} \in \Phi \\
&= \arg\inf_{(\hat{y}^e,\hat{y},\hat{u})\in\hat{H}} \|\hat{y}^e - y^e\|_{\tilde{Q}}^2 + \|\hat{y} - y\|_S^2 + \|\hat{u} - u\|_R^2 , \\
&\quad \text{s.t. } \hat{y}^e = r^e,\ \hat{u} \in \Omega,\ \hat{y} \in \Phi.
\end{aligned} \tag{39}$$

In optimization problem (39), the elements $\hat{y}^e$, $\hat{y}$ and $\hat{u}$ are independent of one another, which means this solution can be obtained separately. Using (34) it follows that

$$P_{S_2}(x) = \begin{pmatrix} r^e \\ P_\Phi(y) \\ P_\Omega(u) \end{pmatrix}. \tag{40}$$

Consider update (29) in Theorem 2, and let $x_k = (r_k^e, \tilde{r}_k, u_k)$ and $\tilde{x}_k = (\tilde{y}_k^e, \tilde{y}_k, \tilde{u}_k)$. At the k^{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 (38) that

$$\tilde{u}_{k+1} = u_k + G^{s*}(I + G^s G^{s*})^{-1} \begin{bmatrix} r_k^e - y_k^e \\ \tilde{r}_k - y_k \end{bmatrix}, \tag{41}$$

$$\tilde{y}_{k+1}^e = G^e\tilde{u}_{k+1},\ \tilde{y}_{k+1} = G\tilde{u}_{k+1},\ k \geqslant 0 \tag{42}$$

and for $x_{k+1} = P_{S_2}(\tilde{x}_{k+1})$, it follows from (40) that

$$r_{k+1}^e = r^e, \tilde{r}_{k+1} = P_\Phi(\tilde{y}_{k+1}), u_{k+1} = P_\Omega(\tilde{u}_{k+1}), k \geqslant 0 \tag{43}$$

which directly illustrates how the input u_{k+1} and the reference $\tilde{r}_{k+1}$ are updated by P_{S_2}. Then, choose an initial guess $x_0 = (r_0^e, \tilde{r}_0, u_0)$ such that

$$r_0^e = r^e,\ \tilde{r}_0 \in \Phi,\ u_0 \in \Omega \tag{44}$$

which together with (43) yields

$$r_k^e = r^e,\ k \geqslant 0. \tag{45}$$

Substituting (45) into (41) gives rise to (32), which updates the input $\tilde{u}_{k+1}$ using P_{S_1}. □

The update (32) can be alternatively implemented in a causal feedback plus feedforward solution to further embed robust performance in practice by exploiting the special properties of the linear operator G^e and its adjoint operator G^{e*}. The implementation details are omitted here for brevity.

Remark 3. Input constraint set Ω is usually a pointwise constraint in practice, so the solution of the projection operator P_Ω is straightforward. Also the solution of the projection operator P_Φ is guaranteed to be unique, otherwise the convergence properties may not hold.

4. CONVERGENCE PROPERTIES

When perfect tracking is possible for the generalized constrained ILC problem (23), i.e. $S_1 \cap S_2 \neq \varnothing$, Algorithm 1 iteratively solves the constrained generalized ILC problem (23) and has desirable convergence properties in practice as shown in the following theorem.

Theorem 4. If perfect tracking is possible, Algorithm 1 yields a sequence $\{(y_k^e, y_k, u_k)\}$ such that

$$\lim_{k\to\infty} y_k^e = r^e \tag{46}$$

which guarantees perfect tracking. Furthermore, the input u_k at each trial satisfies

$$u_k \in \Omega. \tag{47}$$

and the limit of the output y_k (if it exists) satisfies

$$\lim_{k\to\infty} y_k \in \Phi \tag{48}$$

Proof. As both S_1 and S_2 are closed convex sets in the Hilbert space $\hat{H}$ and $S_1 \cap S_2 \neq \varnothing$, it follows from equation (30) in Theorem 2 that for a sufficiently small $\epsilon > 0$, there exists an integer N such that for $k > N$

$$\|r^e - \tilde{y}^e_{k+1}\|^2_{\hat{Q}} + \|\tilde{r}_k - \tilde{y}_{k+1}\|^2_S + \|u_k - \tilde{u}_{k+1}\|^2_R < \epsilon \tag{49}$$

which is equivalent to

$$y^e_k = r^e,\ y_k = \tilde{r}_k,\ u_k = \tilde{u}_{k+1}, \forall k > N. \tag{50}$$

As the sequence $\{y^e_k\}$ converges to a constant value r^e, the limit (46) holds. In addition, the input u_k and reference $\tilde{r}_k$ are obtained from the projection operators P_Ω and P_Φ respectively. Therefore, the input u_k at each trial belongs to the input constraint set and if the sequence $\{y_k\}$ converges we know that its limit belongs to the output constraint set Φ due to $y_k = \tilde{r}_k$, which gives rise to (47) and (48). □

On the other hand, if the intersection of the sets S_1 and S_2 is empty, perfect tracking is impossible. However, the algorithm still attempts to meet the constrained tracking requirement with the given system dynamics. The next theorem illustrate this property.

Theorem 5. If perfect tracking is not possible, the distance between the two sequences $\{(\tilde{y}^e_k, \tilde{y}_k, \tilde{u}_k)\}$ and $\{(r^e_k, \tilde{r}_k, u_k)\}$ yielded by Algorithm 1 converges to the minimum distance between S_1 and S_2, i.e.

$$\inf_u \|r^e - G^e u\|^2_{\hat{Q}} + \|\tilde{r} - Gu\|^2_S + \|\tilde{u} - u\|^2_R\,,\ \ \tilde{r} \in \Phi,\ \ \tilde{u} \in \Omega. \tag{51}$$

Furthermore, the input u_k at each trial satisfies

$$u_k \in \Omega. \tag{52}$$

Proof. From Theorem 2, as both S_1 and S_2 are closed convex sets in Hilbert space $\hat{H}$ and $S_1 \cap S_2 = \varnothing$, it follows that the distance between the two sequences $\{(\tilde{y}^e_k, \tilde{y}_k, \tilde{u}_k)\}$ and $\{(r^e_k, r_k, u_k)\}$ converges to the minimum distance between two sets. Similar to the previous theorem, the condition (52) holds for the input u_k at each trial as it is obtained from the projection operator P_Ω. □

Remark 6. In practice, an appropriate extended reference r^e should be designed to avoid an impossible tracking task, i.e. $S_1 \cap S_2 = \varnothing$. More information on the application of successive projections is omitted here for brevity.

5. 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.

5.1 *Test Platform Specification*

The multi-axis gantry robot shown in Figure 2 is employed as test platform. The control design objective is to use both the x-axis and z-axis ($m = 2$) to perform a generalized ILC tracking task during the given tracking time $T = 2s$. The x-axis and z-axis have been modelled based on frequency response tests in Ratcliffe (2005) with transfer functions

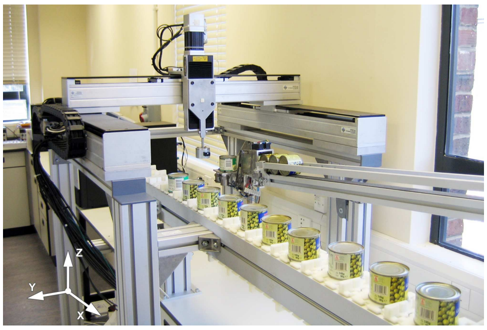

Fig. 2. Multi-axis Gantry Robot Test Platform.

$$G_x(s) = \frac{1.67\times10^{-5}(s+500.2)(s+4.9\times10^5)...}{s(s^2+24s+6401)...}$$
$$\frac{(s^2+10.58s+1.145\times10^4)(s^2+21.98s+2.9\times10^4)}{(s^2+21.38s+2.017\times10^4)(s^2+139.5s+2.162\times10^5)}$$
$$\text{and } G_z(s) = \frac{15.8869(s+850.3)}{s(s^2+707.6s+3.377\times10^5)} \tag{53}$$

and a proportional feedback controller with gain 300 is added on the z-axis. The saturation constraint for the input voltage has form (19) with $M(t) = [3, 2]^\top$. A special tracking task is selected, with a piecewise linear path composed of five line segments ($M = 5$) defined as

$$r(t) = r_{i-1} + \left(\frac{t - t_{i-1}}{t_i - t_{i-1}}\right)(r_i - r_{i-1}),\ t \in [t_{i-1},\ t_i],$$
$$i = 1, \ldots, M \tag{54}$$

where r_i are the vertices in Cartesian space defined by

$$r_1 = \begin{bmatrix} 0.00345 \\ 0.00476 \end{bmatrix}, r_2 = \begin{bmatrix} 0.00905 \\ 0.00294 \end{bmatrix}, r_3 = \begin{bmatrix} 0.00905 \\ -0.00294 \end{bmatrix},$$
$$r_4 = \begin{bmatrix} 0.00345 \\ -0.00476 \end{bmatrix}, r_5 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{55}$$

as shown in later figures. The spatial problem is solved by selecting the projection matrices F_i and P_i in (13) and (14) to be

$$F_i = I,\ P_i = 100 \cdot [(r^2_i - r^2_{i-1}),\ \ -(r^1_i - r^1_{i-1})]. \tag{56}$$

To avoid overshoot problem, the output constraint is defined as

$$\Phi = \{y \in L^m_2[0,\ T] : a_i^\top r_{i-1} \leqslant a_i^\top y(t) \leqslant a_i^\top r_i,$$
$$t \in [t_{i-1},\ t_i],\ i = 1, \ldots, M\},\ a_i = r_i - r_{i-1}. \tag{57}$$

For implementational simplicity, the weighting matrices Q_i, $\hat{Q}_i$, S and R are chosen to be diagonal matrices. The above parameters define (15) such that objective (23) enables spatial ILC tracking using minimum control effort.

5.2 *Performance of the Proposed Algorithm*

Firstly, the generalized ILC update is applied to this task on the gantry robot for 100 trials without constraints to illustrate the limitation of existing method. This corresponds to Algorithm 1 with $\Omega = L^\ell_2[0,\ T]$ and $\Phi = L^m_2[0,\ T]$ so that $P_\Omega(\tilde{u}_{k+1}) = \tilde{u}_{k+1}$ and $P_\Phi(\tilde{y}_k) = \tilde{y}_k$. The final converged hybrid output of the two axes is plotted in Figure 3, and the red dots are the intermediate points, i.e.

r_i, $i = 1, \ldots, M$. Although the converged hybrid output performs perfect tracking along the piecewise linear reference path, it is clear from the figure that the overshoot problem takes place and hence the unconstrained generalized ILC objective cannot solve this problem. In the gantry robot test platform, the overshoot problem may lead to the collision between the end-effector and the frame, which definitely cause damage to the machine.

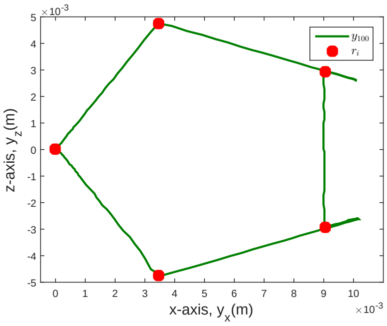

Fig. 3. Final Converged Hybrid Output Trajectory using Generalized ILC without Constraints.

We next apply Algorithm 1 to handle the system constraints. 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 corresponding output paths over different trials are shown in Figure 4 with red dots denoting the intermediate point positions, r_i. Compared to the overshoot results in Figure 3, it is clear from Figure 4 that Algorithm 1 not only achieves the generalized tracking requirement, but keeps the hybrid output trajectory within the constraint set Φ defined in (57), i.e. this algorithm solves the overshoot problem. Furthermore, the input voltages of the two axes over different trials are plotted in Figure 5, and it is clear that both converge and stay within the imposed bounds.

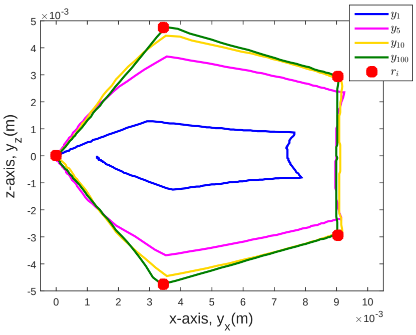

Fig. 4. Hybrid Output Trajectories at Trial 1, 5, 10 and 100 with Output Constraints.

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 corresponding mean square error, e_k^s, at each trial is plotted in Figure 6. 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.

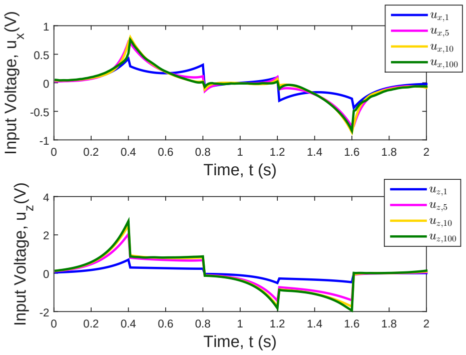

Fig. 5. Input Trajectories at Trial 1, 5, 10 and 100 with Output Constraints.

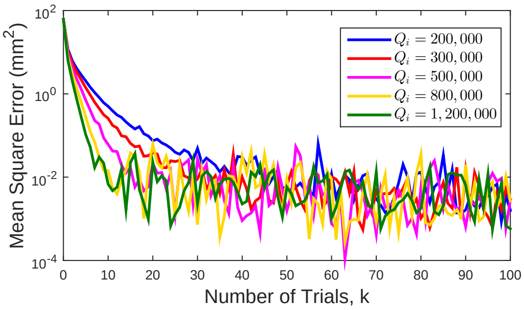

Fig. 6. Mean Square Tracking Error over 100 Trials with Output Constraints.

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

In this paper, a novel ILC framework is developed to solve the generalized (intermediate point and sub-interval tracking) ILC problems with input and output constraints. This is the first algorithm capable of handling these form of constraints, and in so doing solves the output constrained classical ILC problem. It hence has substantial novelty. The algorithm is verified on a gantry robot platform by tracking a given piecewise linear reference with stipulated output constraints, and is founded to provide the first general solution for spatial ILC.

Although the experimental results reveal that the algorithm has a high degree of robustness against model uncertainty and random disturbance, a rigorous analysis will be performed on this algorithm. Other alternative spatial path tracking tasks besides the piecewise linear case will be applied. These constitute the focus of future research and will be reported separately.

REFERENCES

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

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

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

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

Chu, B. and Owens, D.H. (2009). Accelerated norm-optimal iterative learning control algorithms using successive projection. *International Journal of Control*, 82(8), 1469–1484.

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

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

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

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

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

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

Jin, X. and Xu, J.X. (2013). Iterative learning control for output-constrained systems with both parametric and nonparametric uncertainties. *Automatica*, 49.

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

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

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

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

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

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

Ratcliffe, J.D. (2005). *Iterative Learning Control Implemented on a Multi-Axis System*. Ph.D. thesis, University of Southampton, Sothampton.

von Neumann, J. (1950). *Functional Operators*, volume 2. Princeton University Press.