IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 26, 2019, accepted August 13, 2019, date of publication August 28, 2019, date of current version September 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937972

Energy-Optimal Time Allocation in Point-to-Point ILC With Specified Output Tracking

XINGDING ZHAO[1] AND YOUQING WANG[2], (Senior Member, IEEE)
[1]College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China
[2]College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China

Corresponding author: Youqing Wang (wang.youqing@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61822308 and Grant 61751307, in part by the Natural Science Foundation of Shandong Province under Grant JQ201812, and in part by the Research Fund for the Taishan Scholar Project of Shandong Province of China.

ABSTRACT For the point-to-point (P2P) iterative learning control (ILC), the tracked-points are usually known. However, in many cases, time allocation can be regarded as an optimization variable to achieve optimal control energy. Also, in some situations, only information on certain dimensions in the output is needed to be tracked. Based on the combination of the above two aspects, this paper studies energy-optimal time allocation in P2P ILC with specified output tracking. An algorithm based on a two-stage optimization framework that integrates the norm-optimal ILC and the gradient method is proposed. The proposed algorithm is further extended to a system with input constraints, and its robustness is analyzed. Finally, simulation tests on a gantry robot are performed to validate the performance of the proposed algorithm.

INDEX TERMS Iterative learning control, tracking-time allocation, output tracking, point-to-point.

I. INTRODUCTION

In recent years, reducing the consumption of control energy is becoming more and more important in industrial production [1]. For example, in the operation of industrial robots and the process of picking up and placing items [2], it is necessary to consider how to optimize the control energy. When the system runs a series of point-to-point motions in the tracking of certain elements in outputs, minimizing the loss of control energy is meaningful while achieving control objectives.

Iterative learning control(ILC) is an efficient control strategy suitable for systems with repetitive motion properties to achieve the perfect tracking within a limited time period. Theoretically speaking, updating the control input by using information from previous trials can cause the tracking error to converge to 0 after a sufficient number of trials [3], [4]. ILC is widely used in practice, for instance, in the process of chemical [5], treatment of disease [6], [7], and industrial robot manipulation [2]. In particular, research on ILC has attracted considerable attention from the robotics industry [8].

In some situations, such as in the process of picking up and placing objects on a robotic arm, only the output at certain time points requires consideration [9], [10]. P2P ILC is formulated to complete tracking task only at certain prescribed time points to address control issues. The meaningful freedom obtained by reducing unnecessary constraints on the output of the system is utilized by P2P ILC to deal with additional indexes [11].

Generally, the tracking time points are regarded as priori information [12]. If they are unknown, they can be seen as a variable of an optimization index, which gradually reaches a minimum after multiple cycles [13]. As a result, the optimal time allocation could be computed in accordance with the cost function, which may combine tracking requirement with the optimization of the control input energy. In this case, it extents the P2P ILC framework in existence and allows the ILC algorithm to select tracking points and update their values to optimize overall P2P control tasks [14].

In certain situations, some prescribed elements should be tracked. Element tracking is a significant requirement used to select certain elements or linear combinations of certain elements of the output which are vital to the control tasks at each tracking-time point [15]. Such as in the process of picking up and placing objects on a robotic arm, the manipulator need to move from the given ‘‘pick’’ position to a certain plane. Therefore, only one dimension of information in the ‘‘place’’

The associate editor coordinating the review of this article and approving it for publication was Dong Shen.

position is worthy of attention. Although [14] proposes an approach on P2P ILC with energy-optimal time allocation, but the need for specific output tracking is not considered. And in this passage, according the method in [14], based on a two-stage optimization strategy that included NOILC and the gradient method, the presented framework not only achieves output tracking but also acquires the energy-optimal time allocation with prescribed output tracking. Furthermore, in contrast to the a priori time allocation choice, the framework enables task fulfillment with the minimum performance function. This approach improves production efficiency and reduces unnecessary industrial energy waste in practical operations [16]. At the same time, the framework realizes the perfect tracking of selected time points to implement the purpose of ILC [17].

Compared with the previous conference paper by us [18], this article has been expanded in the following two aspects. On the one hand, the limitations of input and output are widely existed in the process of industrial production control. Therefore, it is of great significance to solve the energy-optimal time allocation of the P2P ILC for prescribed output tracking with input constraints. This paper presents a framework for input-limited problems and a two-stage algorithm is designed to handle it. In the simulation, taking input saturation limitation as an example, the input, the convergence of the input norm, and the output tracking results of the internal points are all presented. On the other hand, the robustness of the algorithm is analyzed. In the case of uncertain parameters in the system, the tracking error can still converge to zero. Surely, the detailed proof is given in the paper.

This study provides the following contributions: Firstly, the selection of time points and output are combined into one optimization framework. Secondly, in regard to the P2P iterative learning control with the prescribed output tracking, a two-stage design framework is proposed to acquire the optimal time allocation that corresponds to the minimum control input energy. Last but not least, the proposed algorithm is extended to a system with input constraints and its robustness is proven.

This paper is structured as the following: In Section 2, the problem formulation is given. A two-stage optimization algorithm on the basis of the NOILC and a gradient method is presented for the solution of the problem proposed in Section 3. As discussed in Section 4, the proposed algorithm is extended to a system with input constraint. Section 5 presents the analysis and relative proof of the robustness of the algorithm. The simulation results are shown in Section 6. In the last Section, the conclusions and further research content in the future are provided.

II. PROBLEM FORMULATION

The problem is rigorously formulated as an optimization problem in the Hilbert space by utilizing an abstract-operator expression of system dynamics [19]–[21]. A continuous-time linear time-invariant system is considered as discussed below.

A. SYSTEM EXPRESSION

First, consider an m-output l-input n-state continuous linear time-invariable system as below:

$$\begin{aligned} \dot{x} &= Ax(t) + Bu(t), \quad x(0) = x_0 \\ y(t) &= Cx(t), \quad t \in [0,\ T] \end{aligned} \tag{1}$$

which can be rewritten as the following operator form:

$$\begin{aligned} y &= Gu + d, \quad G : L_2^l[0, T] \rightarrow L_2^m[0, T] \\ y, d &\in L_2^m[0, T] \quad u \in L_2^l[0, T] \end{aligned} \tag{2}$$

where $L_2^l[0, T]$ and $L_2^m[0, T]$ are the Hilbert spaces of the input and output, respectively.

The system operator G and signal d are defined as follows:

$$(Gu)(t) = \int_0^t Ce^{A(t-s)}Bu(s)\,ds, \quad d(t) = Ce^{At}x_0. \tag{3}$$

For convenience and to avoid missing generality, $x_0 = 0$ is assumed. Therefore, $d = 0$.

Only M tracked time points $t_i (i = 1, \ldots, M)$ are necessary to be considered, and they can be denoted as the following vector form:

$$\Lambda = [t_1, t_2, \cdots, t_M] \tag{4}$$

Consider the following map $\zeta \rightarrow \zeta^p$ with

$$\zeta^p = \begin{bmatrix} F_1\zeta(t_1) \\ \vdots \\ F_M\zeta(t_M) \end{bmatrix} \in R^{f_1+\cdots+f_M}, \tag{5}$$

where each F_j has $f_j \times m$ dimensions and full row rank. The inclusion of F_j is the significant extension used to select at each tracking-time point certain elements or linear combinations of certain elements of the output which are important to the tracking requirement.

By the definition, the dynamics could be modeled as below:

$$y^p = (Gu)^p = G_\Lambda^p u = \begin{bmatrix} G_1 u \\ \vdots \\ G_M u \end{bmatrix}, \tag{6}$$

with a linear operator $G_\Lambda^p : L_2^l[0, T] \rightarrow \Re^{f_1+\cdots+f_M}$.

Each operator $G_j : L_2^m[0, T] \rightarrow \Re^{f_j}$ is defined as follows:

$$G_j u = F_j \int_0^{t_j} Ce^{A(t_j - s)}Bu(s)\,ds. \tag{7}$$

The extended tracking reference at certain time points can be defined as follows:

$$r^p = \begin{bmatrix} F_1 r_1 \\ F_2 r_2 \\ \vdots \\ F_M r_M \end{bmatrix} \in \Re^{f_1+\cdots+f_M}. \tag{8}$$

In addition, the tracking error is denoted as follows:

$$e^p(t) = r^p - y^p = (r - y)^p. \tag{9}$$

An inner product of the Hilbert space $\Re^{f_1+\cdots+f_M}$ is defined by the following equation:

$$\langle v, w\rangle_{[Q]} = \sum_{i=1}^{M} v_i Q_i w_i, \qquad \|w\|_{[Q]}^2 = \langle w, w\rangle_{[Q]}. \quad (10)$$

where the $f_j \times f_j$ matrices Q_j, $1 \le j \le M$ are symmetric and positive definite. the set $\{Q_1, Q_2, \cdots, Q_M\}$ is denoted by $[Q]$, and v, w are the elements of the space.

An inner product of $L_2^l[0, T]$ is also defined:

$$\langle u_1, u_2\rangle_R = \int_0^T u_1^T(s) R u_2^T(s)\, ds, \quad (11)$$

where the input energy $\|u\|_R^2 = \langle u, u\rangle_R$ and the $l \times l$ symmetric and positive definite matrix R is the weighting matrix of the input space.

B. ENERGY-OPTIMAL TIME ALLOCATION IN P2P ILC

Here, an explicit formulation of the problem is given. First, the admissible set of the time allocation is given. The tracking-time allocation belongs to a set, i.e., $\Lambda \in \Theta$, and the set Θ denotes the permissable range of the tracked time points as follows:

$$\Theta = \left\{\Lambda \in R^M : 0 < t_1{}^- \le t_1 \le t_1{}^+ \le \ldots \le t_M{}^+\right\}, \quad (12)$$

in which $\left[t_i{}^-, t_i{}^+\right]$ denotes the given interval constraints for t_i.

Then, the energy-optimal time allocation problem in P2P ILC with the prescribed output tracking can be denoted as iteratively acquiring an energy-optimal time allocation Λ_k and an input u_k with the tracking requirement that the output values at these time points, i.e., y_k^p, precisely approach a series of reference values r^p, i.e.,

$$\lim_{k\to\infty} y_k^p = r^p. \quad (13)$$

Meanwhile, the objective cost function $f(u) = \|u\|_R^2$, which indicates the input control energy, is also minimized.

III. TWO-STAGE DESIGN FRAMEWORK

The energy-optimal time allocation for prescribed output tracking is designed by the following two-stage optimization framework.

A. FRAMEWORK DESCRIPTION

As discussed in this section, a two-stage framework is developed to obtain the energy-optimal time allocation Λ in P2P iterative learning control for prescribed output tracking. Although the explicit expression of the objective cost function $f(u, y)$ does not include time allocation Λ, it can be related to the tracking condition $G_\Lambda^p(u) = r^p$. In addition, input u is in an infinite dimensional space $L_2^l[0, T]$, and time allocation Λ is in the finite dimensional admissible set Θ.

As a result, the optimization problem can be described as follows:

$$\min_{\Lambda\in\Theta}\left\{\min_u f(u),\ \textit{subject to } G_\Lambda^P u = r^p, y = Gu\right\} \quad (14)$$

The following minimum for u is defined as

$$u_\infty(\Lambda) : \Theta \to L_2^l[0, T]. \quad (15)$$

Then, the expression of the equivalence problem is

$$\min_{\Lambda\in\Theta} f(u_\infty(\Lambda)), \quad (16)$$

which can be resolved by the two-stage framework as below:

Stage one: The optimal input can be solved in a fixed tracking allocation Λ, as follows:

$$u_\infty(\Lambda) = \arg\min_u \|u\|_R^2 \quad \textit{subject to } r^p = G_\Lambda^p u. \quad (17)$$

Stage two: The optimal tracked time allocation can be solved in accordance with the objective cost function, which includes the fixed-time optimal solution $u_\infty(\Lambda)$, as follows:

$$\Lambda^* = \arg\min_{\Lambda\in\Theta} \|u_\infty(\Lambda)\|_R^2. \quad (18)$$

B. IMPLEMENTATION OF FIRST STAGE

Based a P2P NOILC(norm optimal ILC) algorithm, the stage-one optimization problem is formulated to design P2P iterative learning controller in the sense of minimum input control energy.

At the $(k+1)^{th}$ batch, the problem below can be resolved by using P2P NOILC algorithm:

$$u_{k+1} = \arg\min_u \left\{\left\|e^p\right\|_Q^2 + \|u - u_k\|_R^2 : \ e^p = r^p - G_\Lambda^p u\right\} \quad (19)$$

to acquire the control input u_{k+1}. In accordance with the analysis given by [15], the property of this algorithm is clarified in the following theorem:

Theorem 1: For the system (1), if it is state-controllable and the matrix C has full row rank, under the control input (19), then the extended tracking error e_k^p can converge to 0, and the input sequence $\{u_k\}$ has a limit, i.e.,

$$\lim_{k\to\infty} e_k^p = 0, \quad \lim_{k\to\infty} u_k = u_\infty. \quad (20)$$

Proof: First, in the convergence analysis of the tracking error, $G_e G_e^*$ is positive definite in the defined Hilbert space. For the following condition,

$$\left\langle \alpha,\ G_e G_e^* \alpha\right\rangle_{[Q]} = \left\|G_e^* \alpha\right\|^2 = 0, \quad (21)$$

where α is defined as a vector with M components.

Then

$$G_e^* \alpha = 0,\ \forall t \in [0, T]. \quad (22)$$

For the interval (t_{M-1}, T),

$$B^T p_M(t) = 0. \quad (23)$$

can be obtained, where $p_M(t)$ is defined in the following equations (29) and (30).

$\alpha^T = \left[\alpha_1^T,\ \ldots, \alpha_M^T\right]^T$ is written with $\alpha_j \in \Re^{f_j}$ and

$$p_M(t_M) = C^T F_M^T Q_M \alpha_M = 0. \quad (24)$$

Hence, by applying the rank condition,

$$\alpha_M = 0. \tag{25}$$

is obtained.

Then, an inductive demonstration is performed to show the whole $\alpha_j = 0$.

It has done the proof.

The control input can be expressed as below:

$$u_{k+1} = u_k + G_\Lambda^{p^*} e_{k+1}^p = u_k + G_\Lambda^{p^*}(I + G_\Lambda^p G_\Lambda^{p^*})^{-1} e_k^p, \tag{26}$$

where $G_\Lambda^{p^*}$ is the relevant adjoint operator of G_Λ^p.

Next, the computation for the adjoint operator $G_\Lambda^{p^*}$ is shown.

The adjoint operator in Hilbert space is described as below:

Assuming X and Y are Hilbert spaces and the operator $T : X \to Y$ is defined. So, for any $x \in X, y \in Y$, the adjoint operator $T^* : Y \to X$ meets the condition below:

$$\langle Tx, y\rangle = \langle x, T^* y\rangle . \tag{27}$$

Analogously, for any element $\omega_i\,(i = 1, \ldots, M)$ in the Hilbert space $\Re^{f_1+f_2+\cdots+f_M}$, the equation

$$\left\langle (\omega_1, \ldots, \omega_M), G_\Lambda^p u\right\rangle_{[Q]} = \left\langle G_\Lambda^{p^*} (\omega_1, \ldots, \omega_M), u\right\rangle_R, \tag{28}$$

can be obtained and leads to the following:

$$\left(G_i^* \omega_j\right)(t) = \begin{cases} R^{-1}B^T e^{A^T (t_j - t)} C^T F_j^T Q_j \omega_j; & 0 \le t \le t_j \\ 0; & t > t_j \end{cases} \tag{29}$$

where G_i^* is any one operator of $G_\Lambda^{p^*}$.

This can also be rewritten into the following form:

$$\left(G_i^* \omega_j\right)(t) = R^{-1}B^T p_i(t), \tag{30}$$

where $p_j(t) = 0$ in $\left(t_j, T\right]$, and in $\left[0, t_j\right)$

$$\dot{p}_j(t) = -A^T p_j(t), \quad p_j\left(t_j -\right) = C^T F_j^T Q_j \omega_j. \tag{31}$$

the map $G_\Lambda^{p^*} : (\omega_1, \ldots, \omega_M) \to u(\omega_1, \ldots, \omega_M) \to u$ is defined as follows:

$$u(t) = \sum_{i=1}^{M} \left(G_i^* \omega_i\right)(t) = R^{-1}B^T p(t), \tag{32}$$

where $p(t) = \sum_{i=1}^{M} p_i(t)$.

These equations can be summarized as follows:

$$\begin{cases} u(t) = R^{-1}B^T p(t) \\ \dot{p}(t) = -A^T p(t) \\ p(T) = 0 \\ p\left(t_j -\right) = p\left(t_j +\right) + C^T F_j^T Q_j \omega_j, & 1 \le j < M \end{cases} \tag{33}$$

Finally, an analytic solution for $u_\infty(\Lambda)$ is presented.

The method of Lagrange multipliers is used with the following associated Lagrangian expression to compute the analytic solution of the optimization problem (17):

$$\varphi(u) = \|u\|_R^2 + 2\langle \lambda, G_\Lambda^p u - r^p\rangle_{[Q]}. \tag{34}$$

The global optimal input u_∞ that minimizes φ is defined, and the following conclusion holds:

$$\varphi(u_\infty) \le \varphi(u_\infty + \tau), \quad \forall \tau \in L_2^l[0, T]. \tag{35}$$

Thus,

$$\begin{aligned} &\varphi(u_\infty + \tau) - \varphi(u_\infty) \\ &\quad = \|u_\infty + \tau\|_R^2 - \|u_\infty\|_R^2 \\ &\qquad + 2\langle \lambda, G_\Lambda^p (u_\infty + \tau) - r^p\rangle_{[Q]} - 2\left\langle \lambda, G_\Lambda^p u_\infty - r^p\right\rangle \\ &\quad = \|\tau\|_R^2 + 2\langle u_\infty, \tau\rangle_R + 2\langle \lambda, G_\Lambda^p \tau\rangle_{[Q]} \\ &\quad = \|\tau\|_R^2 + 2\langle u_\infty, \tau\rangle_R + 2\left\langle G_\Lambda^{p^*} \lambda, \tau\right\rangle_R \\ &\quad = \|\tau\|_R^2 + 2\left\langle u_\infty + G_\Lambda^{p^*} \lambda, \tau\right\rangle_R \ge 0, \forall \tau \in L_2^l[0, T]. \end{aligned} \tag{36}$$

Substituting $\tau = -(u_\infty + G_\Lambda^{p^*} \lambda)$ into Eq. (35) yields the following:

$$-\left\| u_\infty + G_\Lambda^{p^*} \lambda \right\|_R^2 \ge 0. \tag{37}$$

Then:

$$u_\infty = -G_\Lambda^{p^*} \lambda. \tag{38}$$

Eq. (38) is then combined with the following equation:

$$G_\Lambda^p u_\infty = r^p, \tag{39}$$

which yields:

$$\lambda = -\left(G_\Lambda^p G_\Lambda^{p^*}\right)^+ r^p \tag{40}$$

where $+$ denotes pseudoinverse.

Finally,

$$u_\infty(\Lambda) = G_\Lambda^{p^*}\left(G_\Lambda^p G_\Lambda^{p^*}\right)^+ r^p. \tag{41}$$

The stage-one optimization problem can also be implemented by using the following method, which includes the implementation of the feedback plus feedforward.

Eq. (26) is equal to

$$u_{k+1}(t) = u_k(t) + G_\Lambda^{P^*} e_{k+1}^p(t). \tag{42}$$

and $G_\Lambda^{p^*} e_{k+1}^p(t)$ can be rewritten as

$$\begin{aligned} G_\Lambda^{p^*} e_{k+1}^p(t) &= R^{-1}B^T p_k(t), \\ \dot{p}_k(t) &= -A^T p_k(t), p_k(T) = 0, \\ p_k(t_j-) &= p_k(t_j+) + C^T {F_j}^T Q_j F_j e_{k+1}(t_j), \end{aligned} \tag{43}$$

in accordance with the costate equation (33).

Therefore, (42) becomes

$$u_{k+1}(t) = u_k(t) + R^{-1}B^T p_k(t), \tag{44}$$

write

$$p_k(t) = -K(t)\left[x_{k+1}(t) - x_k(t)\right] + \xi_{k+1}(t). \tag{45}$$

where $K(t)$, $\xi_{k+1}(t)$ are supposed to be continuously differentiable in each interval (t_j, t_{j+1}) but might be uncontinuous at $t = t_j$ and $K(t)$ is symmetric.

The conditions for $K(t)$ and $\xi_{k+1}(t)$ are acquired through two steps:

1) In accordance with the jump conditions of (43),

$$\begin{aligned}-K(t_j-)[x_{k+1}(t_j)-x_k(t_j)]+\xi_{k+1}(t_j-)\\ =-K(t_j+)[x_{k+1}(t_j)-x_k(t_j)]+\xi_{k+1}(t_j+)\\ +C^TF_j^TQ_jF_je_{k+1}(t_j)\end{aligned}\tag{46}$$

the error $e_{k+1}(t_j)$ can be further expressed as

$$e_{k+1}(t_j)=r_j-y_{k+1}(t_j)=e_k(t_j)-C[x_{k+1}(t_j)-x_k(t_j)].\tag{47}$$

then

$$\begin{aligned}&[K(t_j-)-K(t_j+)][x_{k+1}(t_j)-x_k(t_j)]\\ &\quad+[\xi_{k+1}(t_j-)-\xi_{k+1}(t_j+)]\\ &=C^TF_j^TQ_jF_jC[x_{k+1}(t_j)-x_k(t_j)]+C^TF_j^TQ_jF_je_k(t_j)\end{aligned}\tag{48}$$

which shows the jump condition

$$\begin{aligned}&K(T)=0,\\ &K(t_j-)-K(t_j+)=C^TF_j^TQ_jF_jC,\quad j=1,\ldots,M.\end{aligned}\tag{49}$$

$$\begin{aligned}&\xi_{k+1}(T)=0,\\ &\xi_{k+1}(t_j-)-\xi_{k+1}(t_j+)=C^TF_j^TQ_jF_je_k(t_j),\quad j=1,\ldots,M.\end{aligned}\tag{50}$$

2) Next, (45) is differentiated at any point t other than the tracked time points in Λ, $\dot{x}_k$ and $\dot{x}_{k+1}$ are replaced to yield the general Riccati and predictive differential equations

$$\dot{K}(t)+A^TK(t)-K(t)BR^{-1}B^TK(t)+K(t)A=0,\tag{51}$$

$$\dot{\xi}_{k+1}(t)+A^T\xi_{k+1}(t)-K(t)BR^{-1}B^T\xi_{k+1}(t)=0.\tag{52}$$

with the jump condition above defined at time instants $t\in\{t_j\}_{1\le j\le M}$. In general, stage-one optimization problem is implemented as follows:

1) Calculate the Riccati feedback matrix offline from the following equations:

$$\begin{aligned}&\dot{K}(t)+A^TK(t)-K(t)BR^{-1}B^TK(t)+K(t)A=0,\\ &K(T)=0,\\ &K(t_j-)-K(t_j+)=C^TF_j^TQ_jF_jC,\quad j=1,\ldots,M.\end{aligned}\tag{53}$$

2) Calculate the predictive feedforward term offline from the following equations:

$$\begin{aligned}&\dot{\xi}_{k+1}(t)+A^T\xi_{k+1}(t)-K(t)BR^{-1}B^T\xi_{k+1}(t)=0,\\ &\quad\xi_{k+1}(T)=0,\\ &\xi_{k+1}(t_j-)-\xi_{k+1}(t_j+)\\ &\qquad=C^TF_j^TQ_jF_je_k(t_j),\quad j=1,\ldots,M.\end{aligned}\tag{54}$$

3) Carry out the control law:

$$u_{k+1}(t)=u_k(t)+R^{-1}B^T[-K(t)[x_{k+1}(t)-x_k(t)]+\xi_{k+1}(t)].\tag{55}$$

In the implementation of the first stage, both analytic solution and updating algorithm can be used. In fact, the updating algorithm is more robust due to the inclusion of state feedback. In addition, an updating algorithm is also used to obtain the control input with batch variations.

C. IMPLEMENTATION OF SECOND STAGE

In the second stage, a gradient method is carried out to acquire an approximate optimal solution. Start with an initial choice of time allocation Λ_0, the time allocation $\Lambda_j=\left[t_1^j,t_2^j,\cdots,t_M^j\right]$ is iteratively calculated to the j^{th} loop by using the gradient-descent algorithm [22] as below:

$$\Lambda_j=\Lambda_{j-1}-\gamma\frac{\partial f\left(\Lambda_{j-1}\right)}{\partial\Lambda_{j-1}}=\Lambda_{j-1}-\gamma\begin{bmatrix}\frac{\partial f\left(\Lambda_{j-1}\right)}{\partial t_1^{j-1}}\\ \frac{\partial f\left(\Lambda_{j-1}\right)}{\partial t_2^{j-1}}\\ \vdots\\ \frac{\partial f\left(\Lambda_{j-1}\right)}{\partial t_M^{j-1}}\end{bmatrix},\tag{56}$$

where γ is a scalar step width.

If the step width γ satisfies the condition as below,

$$\gamma\le 2/\bar{\sigma}\left(H\left(f\left(\Lambda_{j-1}\right)\right)\right).\tag{57}$$

where $\bar{\sigma}\left(H\left(f\left(\Lambda_{j-1}\right)\right)\right)$ is the largest singular value of the Hessian matrix $H\left(f\left(\Lambda_{j-1}\right)\right)$, the value of the object cost function will decrease monotonically [23]. Therefore, it can be seen as a reference for the selection of the step width.With the appropriate step width, a solution to the second stage optimization problem could be acquired by using the gradient method [16], [17], [24].

Lemma 1 [14]: By using the analytical solution (24) to the first stage optimization problem, the second stage optimization problem (16) can be expressed as below:

$$\|u_\infty(\Lambda)\|_R^2=r_p^T\left(G^p(\Lambda)\,G^p(\Lambda)^*\right)^+r_p.\tag{58}$$

Obtaining the optimal time point set through the gradient method requires knowledge of the gradient of the input energy function at a given time-point set, i.e., $\partial f/\partial\Lambda$, which can be directly computed by using Eq. (56).

However, in brief, the convergence property of the algorithm depends on the choice of the initial value. Therefore, the choice of initial time allocation is crucial. In this study, the central time allocation is selected as the initial value, in which all the initial tracked time instants are in the middle of corresponding time intervals. The initial time allocation is defined as below:

$$\Lambda_0=\left[t_1^0,t_2^0,\cdots,t_M^0\right]^T,\tag{59}$$

where $t_i^0=\left(t_i^-+t_i^+\right)/2$.

D. ITERATIVE ALGORITHM IMPLEMENTATION

First of all, an initial time allocation Λ_0 is selected, and the steps of first and second stages are then performed to obtain the following algorithm:

Step 1 At the start of the j^{th} circle, the first stage design step is performed, and the terminal control input energy value of $f\left(\Lambda_{j-1}\right)$ is obtained.

Step 2 The second stage design step is performed by using gradient method to renew the time allocation:

$$\Lambda_j = \Lambda_{j-1} - \gamma \frac{\partial f\left(\Lambda_{j-1}\right)}{\Lambda_{j-1}}. \quad (60)$$

Step 3 Let $j \to j+1$, turn back to the first procedure until the boundary condition

$$\left|f\left(\Lambda_j\right) - f\left(\Lambda_{j-1}\right)\right| < \delta \left|f\left(\Lambda_{j-1}\right)\right|. \quad (61)$$

is satisfied. Parameter δ can be selected in accordance with tracking precision and performance requirements [19], [25].

IV. INPUT CONSTRAINT PROBLEMS

As discussed in the previous section, a two-stage optimization algorithm was designed. The previous optimization algorithm must be extended to the system with input constraints.

A. PROBLEM FORMULATION WITH INPUT CONSTRAINTS

System input limitations are commonly encountered in actual production and daily life because of physical limitations and performance specifications. In summary, several types of input restrictions exist as follows [26]:

1) Input saturation constraint

$$\Omega = \left\{u(t) \in R^l : \ |u(t)| \le M(t), \quad t \in [0, T]\right\}. \quad (62)$$

2) Input amplitude constraint

$$\Omega = \left\{u(t) \in R^l : \ \lambda(t) \le u(t) \le \eta(t), \quad t \in [0, T]\right\}. \quad (63)$$

3) Input sign constraint

$$\Omega = \left\{u(t) \in R^l : \ 0 \le u(t), \quad t \in [0, T]\right\}. \quad (64)$$

Hence, the optimization problem with input constraint is

$$\min_{\Lambda \in \Theta} \left\{\min_u f(u), \ subject\ to\ G_\Lambda^p u = r^p, \ y = Gu, \ u \in \Omega\right\} \quad (65)$$

B. TWO-STAGE ALGORITHM WITH INPUT CONSTRAINTS

For optimization problem (65), a two-stage algorithm with input constraints is developed as follows:

Stage 1: In a fixed tracking allocation, the optimal input can be obtained from:

$$\begin{aligned} &\min_u \quad \|u\|_R^2 \\ &subject\ to\ \ r^p = G_\Lambda^p u. \end{aligned} \quad (66)$$

where the solution is defined as $\hat{u}_\infty(\Lambda)$.

Stage 2: The optimal time allocation can be solved by:

$$\min_{\Lambda \in \Theta} \left\{\left\|\hat{u}_\infty(\Lambda)\right\|^2\right\}. \quad (67)$$

Stage one is a constrained optimization problem. An analytical solution is difficult to acquire for the optimization problem in stage one. However, the problem can be solved by the method given by [27]–[29]. This method is divided into two steps:

First, a general norm-optimal ILC optimization problem with unconstrained input

$$\bar{u}_{k+1} = \arg\min_u \left\{\left\|e^p\right\|_{[Q]}^2 + \|u - u_k\|_{[R]}^2\right\}. \quad (68)$$

is solved.

Then, the input is projected into the constraint set

$$u_{k+1} = \arg\min_{u \in \Omega} \|u - \bar{u}_{k+1}\|. \quad (69)$$

This approach is easier to be implemented and calculated than the constrained optimization problem.

V. ANALYSIS OF THE ROBUSTNESS

The robustness of the algorithm is discussed in this section. Tracking error can still converge to 0 even if uncertain parameters exist in the system [30]–[33].

Consider a linear time-invariant system with the form:

$$y = G(\lambda)u. \quad (70)$$

$\lambda \in \Omega$, where Ω is a bounded set, which means there exists $M < \infty$ such that $\|\lambda\|_\infty \le M$.

The control input u_{k+1} can be resolved using the norm-optimal ILC algorithm as below:

$$u_{k+1} = \arg\min_u \left[\left\|e_{k+1}^p\right\|_L^2 + \|u - u_k\|_S^2\right]. \quad (71)$$

where L is the weighting matrix of the output, and S is the weighting matrix of the input. They are symmetric and positive definite. e_{k+1}^p is the tracking error on the $(k+1)^{th}$ iteration.

Theorem 2: For the system (70), using the control input law (71), if the tracking reference $r \in R(G)$ or $R(G)$ is dense in Y, then the tracking error can converge to 0.

Proof: The quadratic performance criterion of the norm optimal iterative learning control (NOILC) is

$$\begin{aligned} J_{k+1} &= \left\|e_{k+1}^p\right\|_L^2 + \|u_{k+1} - u_k\|_S^2 \\ &= \left(e_{k+1}^p(\lambda)\right)^T L e_{k+1}^p(\lambda) + \Delta u_{k+1}^T S \Delta u_{k+1} \\ &= \left(e_k^p(\lambda)\right)^T L e_k^p(\lambda) - 2\left(e_k^p(\lambda)\right)^T LG(\lambda)\Delta u_{k+1} \\ &\quad + \Delta u_{k+1}^T \left(S + G(\lambda)^T LG(\lambda)\right)\Delta u_{k+1}. \end{aligned} \quad (72)$$

The NOILC algorithm has the following property:

$$\left\|e_{k+1}^p\right\|^2 \le J_{k+1}(u_{k+1}) \le \left\|e_k^p\right\|^2. \quad (73)$$

This property follows from optimality and the fact that the nonoptimal choice of $u_{k+1} = u_k$ would lead to

$$J_{k+1}(u_k) = \left\|e_k^p\right\|^2. \quad (74)$$

Obviously, the algorithm is going in the direction of decreasing the error along the iteration k. In addition, equality holds if and only if $u_{k+1} = u_k$, that is, if the algorithm has converged and input-updating no longer occurs.

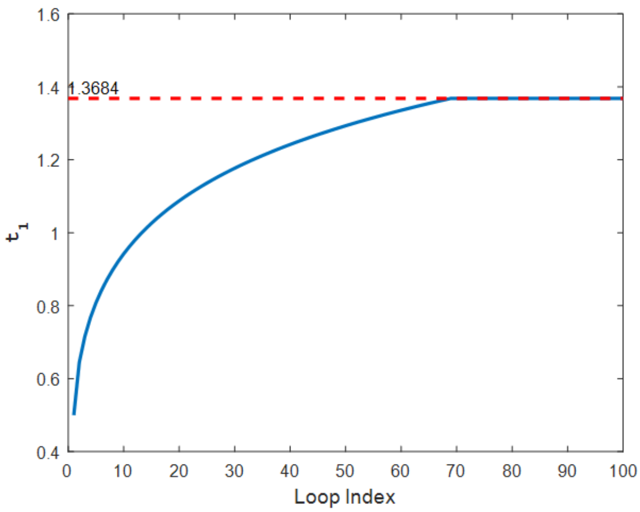

FIGURE 1. Time-point position results for t_1 at each loop.

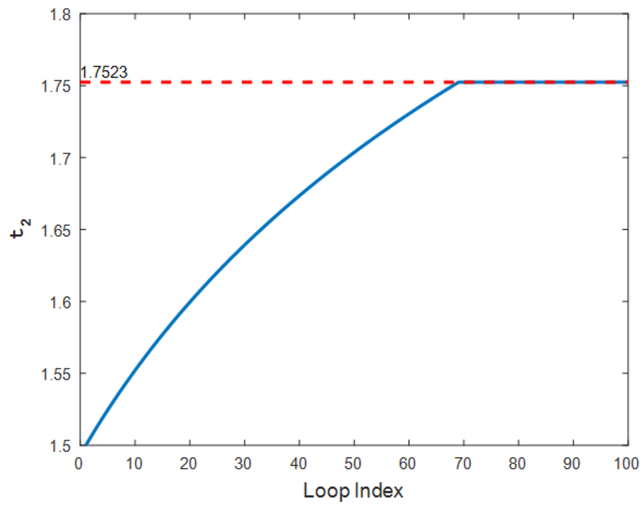

FIGURE 2. Time-point position results for t_2 at each loop.

The input update model of the system is

$$u_{k+1} = u_k + G^*(\lambda)e^p_{k+1}. \tag{75}$$

where G^* is the relevant adjoint operator of G. The error update model of the system is

$$\begin{aligned} e^p_{k+1} &= e^p_k - G(\lambda)\,\Delta u_{k+1} \\ &= \left[I + G(\lambda)G^*(\lambda)\right]^{-1} e^p_k. \end{aligned} \tag{76}$$

The monotonicity of the algorithm shows that the following limit exists

$$\lim_{k\to\infty} \left\|e^p_k\right\|^2 = \lim_{k\to\infty} J_k(u_k) = J_\infty \geq 0. \tag{77}$$

Using the following inequality

$$\|y\| \leq \|G(\lambda)\|\,\|u\|. \tag{78}$$

It yields the relations

$$\sum_{k\geq 0} \|u_{k+1} - u_k\|^2 < \left\|e^p_0\right\|^2 - J_\infty < \infty. \tag{79}$$

$$\sum_{k\geq 0} \left\|e^p_{k+1} - e^p_k\right\|^2 < \|G(\lambda)\|^2 \left(\left\|e^p_0\right\|^2 - J_\infty\right) < \infty. \tag{80}$$

and hence

$$\lim_{k\to\infty} \|u_{k+1} - u_k\|^2 = 0. \tag{81}$$

$$\lim_{k\to\infty} \left\|e^p_{k+1} - e^p_k\right\|^2 = 0. \tag{82}$$

Let $u \in U$ be arbitrary. Also, note from the property of asymptotically slow variation (81) that $u_{k+1} - u_k \to 0$ in norm as $k \to \infty$.

Hence, as stated in Eq.(81),

$$\begin{aligned} \lim_{k\to\infty} \langle u,\ u_{k+1} - u_k\rangle_U &= \lim_{k\to\infty} \left\langle u,\ G^* e^p_{k+1}\right\rangle_U \\ &= \lim_{k\to\infty} \left\langle Gu, e^p_{k+1}\right\rangle_Y = 0. \end{aligned} \tag{83}$$

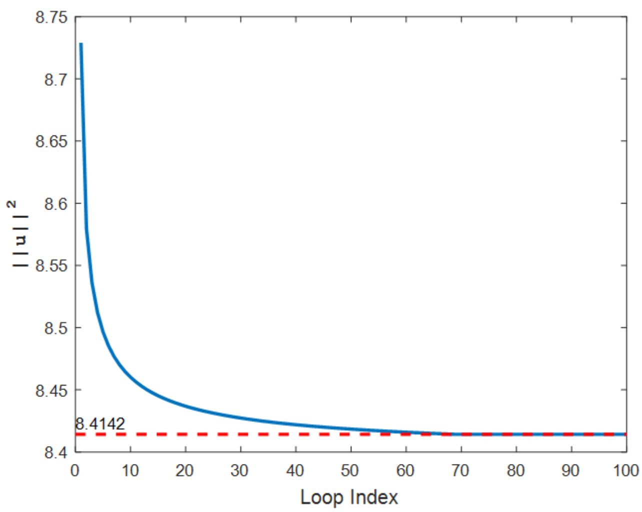

FIGURE 3. Input energy results.

Eq. (83) means that

$$\lim_{k\to\infty} \left\langle y, e^p_k\right\rangle_Y = 0. \tag{84}$$

From the Eps. (72) and (75)

$$\begin{aligned} J_{k+1} &= \left\|e^p_{k+1}(\lambda)\right\|^2_Y + \left\|G^*(\lambda)e^p_{k+1}(\lambda)\right\|^2_U \\ &= \left\langle e^p_{k+1}(\lambda), \left[I + G(\lambda)G^*(\lambda)\right] e^p_{k+1}(\lambda)\right\rangle_Y \end{aligned} \tag{85}$$

Define the self-adjoint operator

$$H = \left[I + G(\lambda)G^*(\lambda)\right]. \tag{86}$$

By induction from (76)

$$e^p_k = H^j e^p_{k+j}. \tag{87}$$

Hence

$$e^p_k = H^{-k} e^p_0. \tag{88}$$

$$e^p_{k+1} = H^k e^p_{2k+1}. \tag{89}$$

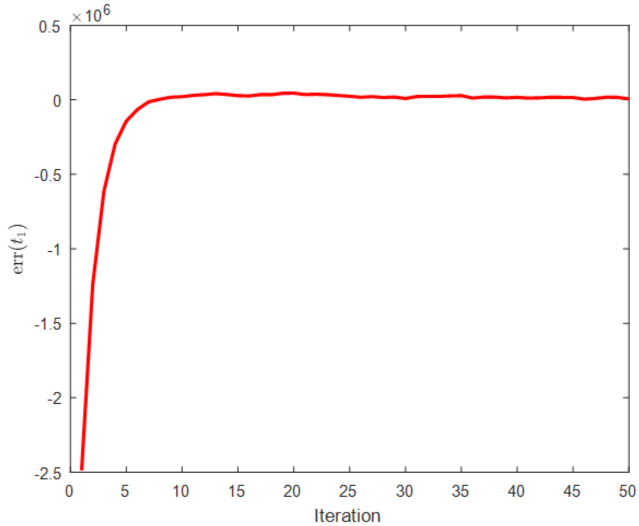

FIGURE 4. Tracking error at t_1.

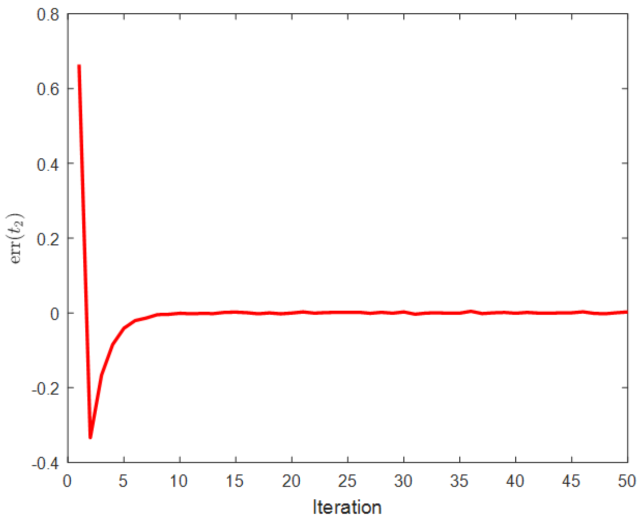

FIGURE 5. Tracking error at t_2.

Applying this relation yields

$$J_{k+1} = \langle e_k^p, e_{k+1}^p \rangle = \left\langle H^{-k} e_0^p, H^k e_{2k+1}^p \right\rangle = \langle e_0^p, e_{2k+1}^p \rangle . \quad (90)$$

If $r \in R(G)$, then $e_0^p = r^p - Gu_0$ is in the range of G. By writing $e_0^p = Gu$, the limit of J_{k+1} can be obtained from the Eq. (84)

$$\lim_{k \to \infty} \langle y, e_k^p \rangle_Y = 0. \quad (91)$$

According to the equation

$$\lim_{k \to \infty} \| e_k^p \|^2 = \lim_{k \to \infty} J_k (u_k) = J_\infty \geq 0. \quad (92)$$

The following equation is satisfied

$$\lim_{k \to \infty} \| e_k^p \|^2 = 0. \quad (93)$$

This finishes the proof.

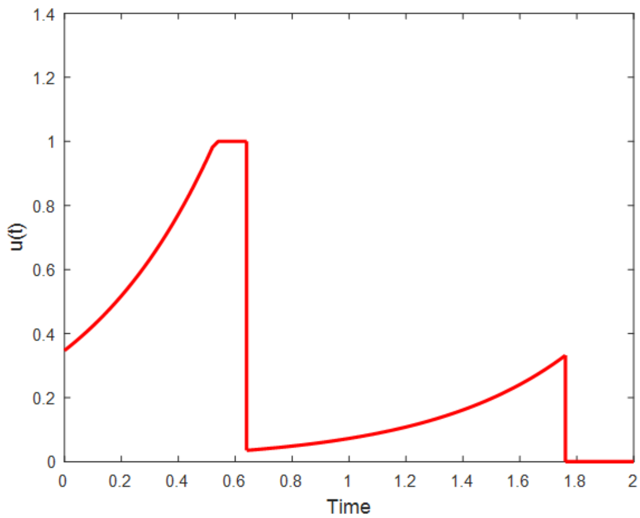

FIGURE 6. Results for the input under constraints.

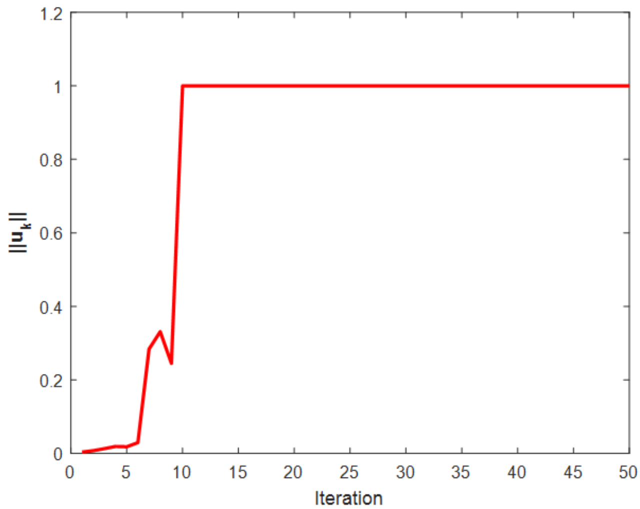

FIGURE 7. The result of the input norm.

VI. SIMULATION TEST OF GANTRY ROBOT

A multi axis gantry robot platform is used to validate the proposed method [34]–[36]. It includes the three vertical axes above the moving conveyor-belt. The equipment contains three independent axes on a moving conveyor-belt. The X- and Y axes are in the horizontal plane, while the Z axis is moving in the vertical direction. The robot uses the robotic arm to pick up the item from the bottom and place it on the conveyor belt, and the axis position data corresponding to the motion trail of the robot arm are recorded by the optical device during the process. In the manipulator trajectory of the entire space, it is only necessary to pay attention to the motion behavior of one or two axes of the robot arm in many cases. In the example below, only the position information of the X-axis and the Y-axis needs to be recorded, and the spatial position of the tracked points moves on a certain plane. The control goal is to ensure the two tracked points move on a specific plane.

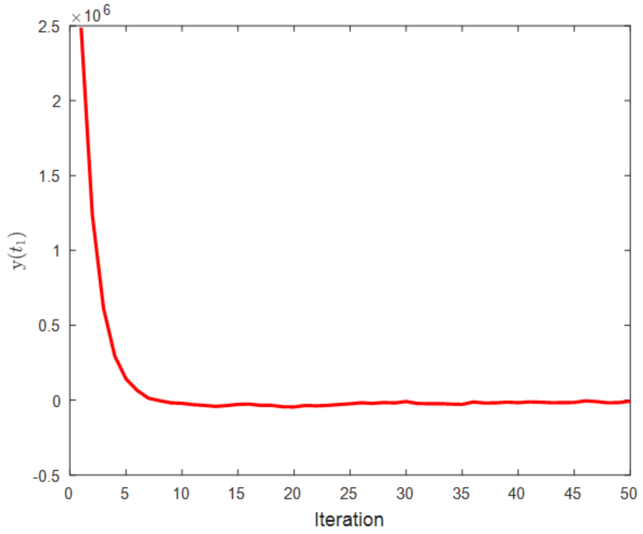

FIGURE 8. Output at t_1.

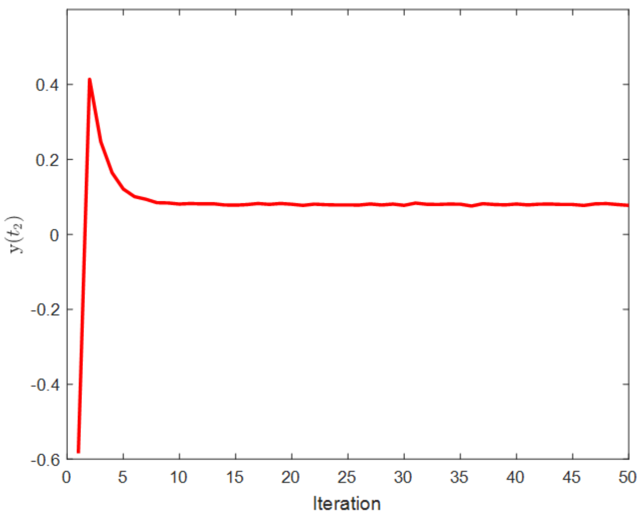

FIGURE 9. Output at t_2.

Consider the system below:

$$G(s) = \frac{1}{s+2}. \tag{94}$$

only two intermediate points are needed to be tracked ($M = 2$), and the constraint of the tracked time points is $0 < t_1 < 1.5s < t_2 < 2s$.

The reference vector for each intermediate point is severally $r_1 = [0 \quad 0.5]^T$ and $r_2 = [0.6 \quad 0.1]^T$. $F_1 = [1 \quad 0]$ and $F_2 = [0 \quad 1]$ are defined for the selection of important elements of the output. As such, $r^p = [0 \quad 0.1]^T$. The following parameters are set: initial state value $x_0 = 0$, initial input value $u_0 = 0$, the weighting matrix of the input space $R = 1$, $Q_1 = Q_2 = 500,000$, step length $\gamma = 0.1$, and the value in the terminal condition $\delta = 0.00002$.

The algorithm is computed in continuous time and then discretized with the sampling interval $T_s = 0.001$.

The optimal tracking-time points are $t_1 = 1.3684$ s and $t_2 = 1.7523$ s as illustrated in Figs.1 and 2.

The control input energy monotonously decreases, and the minimum energy corresponding to the optimal time allocation is shown in Fig.3, which can be obtained as follows by using the proposed method:

$$\min_{\Lambda} \|u_\infty(\Lambda)\|^2 = 8.4142(J). \tag{95}$$

The tracking errors of the two internal points are shown in Figs. 4 and 5. As shown in the figures, the optimal time allocation is successfully computed, and good tracking performance is obtained.

Then, the input saturation constraint is incorporated into the system with $M(t) = 1$. The input constraint is ensured as shown in Fig. 6.

The convergence of the input norm along the batch axis is presented in Fig. 7. The limit of the input norm is

$$\lim_{k\to\infty} \|u_k\| = 1. \tag{96}$$

The outputs of the two tracked points that vary with the iteration are plotted in Figs. 8 and 9. Output tracking is achieved in accordance with output tracking requirements of the two intermediate points. i.e. $F_1 r_1 = 0$ and $F_2 r_2 = 0.1$, the output tracking is achieved.

VII. CONCLUSION

In this paper, to solve the energy-optimal time allocation for P2P ILC with prescribed output tracking, the selection of time points and output are combined into one framework to be optimized. Meanwhile, in view of NOILC and a gradient method, a two-stage iterative algorithm is designed. Then, a system with input constraint is considered. Finally, the robustness of the algorithm, particularly its robustness against parameter uncertainty, is discussed.

The performance of the algorithm when facing other forms of model uncertainties warrants further analysis. In addition, widespread system constraints exist in practice. Thus, future studies on the inclusion of additional constraints, such as output constraints and input energy constraints, into the given formulation are needed. Besides, other overall optimization objectives can also be selected.

REFERENCES

[1] H.-Q. Guo, C.-Z. Liu, J.-W. Yong, X.-Q. Cheng, and F. Muhammad, "Model predictive iterative learning control for energy management of plug-in hybrid electric vehicle," *IEEE Access*, vol. 7, pp. 71323–71334, 2019.

[2] S. Arimoto, S. Kawamura, and F. Miyazaki, "Convergence, stability, and robustness of learning control schemes for robot manipulators," in *Recent Trends in Robotics: Modeling, Control and Education*, M. J. Jamishidi, L. Y. Luh, and M. Shahinpoor, Eds. New York, NY, USA: Elsevier, 1986, pp. 307–316.

[3] D. Shen, W. Zhang, Y. Wang, and C.-J. Chien, "On almost sure and mean square convergence of P-type ILC under randomly varying iteration lengths," *Automatica*, vol. 63, pp. 359–365, Jan. 2016.

[4] D. Shen and Y. Wang, "Survey on stochastic iterative learning control," *J. Process Control*, vol. 24, no. 12, pp. 64–77, Dec. 2014.

[5] Y. Wang, E. Dassau, and F. J. Doyle, III, "Closed-loop control of artificial pancreatic β-cell in type 1 diabetes mellitus using model predictive iterative learning control," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 2, pp. 211–219, Feb. 2010.
[6] Y. Wang, J. Zhang, F. Zeng, N. Wang, X. Chen, B. Zhang, D. Zhao, W. Yang, and C. Cobelli, "'Learning' can improve the blood glucose control performance for type 1 diabetes mellitus," *Diabetes Technol. Therapeutics*, vol. 19, no. 1, pp. 41–48, 2017.
[7] Y. Wang, J. Shi, D. Zhou, and F. Gao, "Iterative learning fault-tolerant control for batch processes," *Ind. Eng. Chem. Res.*, vol. 45, no. 26, pp. 9050–9060, 2006.
[8] M. Norrlof, "An adaptive iterative learning control algorithm with experiments on an industrial robot," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 245–251, Apr. 2002.
[9] C. T. Freeman, "Constrained point-to-point iterative learning control with experimental verification," *Control Eng. Pract.*, vol. 20, no. 5, pp. 489–498, 2012.
[10] R. Chi, Z. Hou, S. Jin, and B. Huang, "An improved data-driven point-to-point ILC using additional on-line control inputs with experimental verification," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 4, pp. 687–696, Apr. 2019.
[11] T. D. Son, H.-S. Ahn, and K. L. Moore, "Iterative learning control in optimal tracking problems with specified data points," *Automatica*, vol. 49, no. 5, pp. 1465–1472, 2013.
[12] C. T. Freeman, Z. Cai, E. Rogers, and P. L. Lewin, "Iterative learning control for multiple point-to-point tracking application," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 3, pp. 590–600, May 2011.
[13] P. Janssens, G. Pipeleers, M. Diehl, and J. Swevers, "Energy-optimal time allocation of a series of point-to-point motions," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 6, pp. 2432–2435, Nov. 2014.
[14] Y. Chen, B. Chu, and C. T. Freeman, "Point-to-point iterative learning control with optimal tracking time allocation," in *Proc. 54th IEEE Conf. Decis. Control*, Osaka, Japan, Dec. 2015, pp. 6089–6094.
[15] D. H. Owens, C. T. Freeman, and T. V. Dinh, "Norm-optimal iterative learning control with intermediate point weighting: Theory, algorithms, and experimental evaluation," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 999–1007, May 2013.
[16] J. Hätönen, "Issues of algebra and optimality in iterative learning control," Ph.D. dissertation, Dept. Process Environ. Eng., Univ. Oulu, Oulu, Finland, 2004.
[17] J. Hätönen, D. H. Owens, and K. L. Moore, "An algebraic approach to iterative learning control," *Int. J. Control*, vol. 77, no. 1, pp. 45–54, 2004.
[18] X. Zhao and Y. Wang, "Optimal time allocation of point-to-point iterative learning control with specifiewed output tracking," in *Proc. 7th IEEE Conf. Data Driven Control Learn. Syst.*, Enshi, China, 2018, pp. 390–395.
[19] J. Hätönen and D. H. Owens, "Convex modifications to an iterative learning control law," *Automatica*, vol. 40, no. 7, pp. 1213–1220, 2004.
[20] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control using optimal feedback and feedforward actions," *Int. J. Control*, vol. 65, no. 2, pp. 277–293, 1996.
[21] B. Chu and D. H. Owens, "Accelerated predictive norm-optimal iterative learning control," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 225, no. 6, pp. 744–759, 2011.
[22] D. P. Bertsekas, "On the Goldstein-Levitin-Polyak gradient projection method," *IEEE Trans. Autom. Control*, vol. AC-21, no. 2, pp. 174–184, Apr. 1976.
[23] A. A. Goldstein, *Constructive Real Analysis*, New York, NY, USA: Dover, 2012.
[24] D. H. Owens and J. Hätönen, "Iterative learning control—An optimization paradigm," *Annu. Rev. Control*, vol. 29, no. 1, pp. 57–70, 2005.
[25] L. Hladowski, K. Galkowski, Z. Cai, E. Rogers, C. T. Freeman, and P. L. Lewin, "A 2D systems approach to iterative learning control for discrete linear processes with zero Markov parameters," *Int. J. Control*, vol. 84, no. 7, pp. 1246–1262, 2011.
[26] C. T. Freeman and Y. Tan, "Iterative learning control with mixed constraints for point-to-point tracking," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 604–616, May 2013.
[27] B. Chu, C. T. Freeman, and D. H. Owens, "A novel design framework for point-to-point ILC using successive projection," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 3, pp. 1156–1163, May 2015.
[28] B. Chu and D. H. Owens, "Accelerated norm-optimal iterative learning control algorithms using successive projection," *Int. J. Control*, vol. 82, no. 8, pp. 1469–1484, 2009.
[29] N. Amann, "Optimal algorithms for iterative learning control," Ph.D. dissertation, Univ. Exeter, Exeter, England, 1996.
[30] D. H. Nguyen and D. Banjerdpongchai, "A convex optimization approach to robust iterative learning control for linear systems with time-varying parametric uncertainties," *Automatica*, vol. 47, no. 9, pp. 2039–2043, Sep. 2011.
[31] Y. Wang, H. Zhang, S. Wei, D. Zhou, and B. Huang, "Control performance assessment for ILC-controlled batch processes in a 2-D system framework," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1493–1504, Sep. 2018.
[32] M. Yu and C. Li, "Robust adaptive iterative learning control for discrete-time nonlinear systems with time-iteration-varying parameters," *IEEE Trans. Syst., Man., Cybern. Syst.*, vol. 47, no. 7, pp. 1737–1745, Jul. 2017.
[33] L. Huang, Q. Zhang, L. Sun, and Z. Sheng, "Robustness analysis of iterative learning control for a class of mobile robot systems with channel noise," *IEEE Access*, vol. 7, pp. 34711–34718, 2019.
[34] Z. Cai, C. T. Freeman, P. L. Lewin, and E. Rogers, "Iterative learning control for a non-minimum phase plant based on a reference shift algorithm," *Control Eng. Pract.*, vol. 16, no. 6, pp. 633–643, 2008.
[35] C. T. Freeman, P. L. Lewin, and E. Rogers, "Experimental evaluation of iterative learning control algorithms for non-minimum phase plants," *Int. J. Control*, vol. 78, no. 11, pp. 826–846, 2005.
[36] J. D. Ratcliffe, P. L. Lewin, E. Rogers, J. J. Hatonen, and D. H. Owens, "Norm-optimal iterative learning control applied to gantry robots for automation applications," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1303–1307, Dec. 2006.

XINGDING ZHAO received the B.S. degree from Xidian University, Xi'an, China, in 2015. He is currently pursuing the Ph.D. degree with the Beijing University of Chemical Technology. His research interests include iterative learning control, fault-tolerant control, and system modeling and optimization.

YOUQING WANG (M'09–SM'12) received the B.S. degree from Shandong University, Jinan, China, in 2003, and the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2008. He was a Research Assistant with the Department of Chemical Engineering, Hong Kong University of Science and Technology, from February 2006 to August 2007. From February 2008 to February 2010, he was a Senior Investigator with the Department of Chemical Engineering, University of California at Santa Barbara, Santa Barbara, CA, USA. From August 2015 to November 2015, he was a Visiting Professor with the Department of Chemical and Materials Engineering, University of Alberta, Canada. He is currently a Professor with the Shandong University of Science and Technology and the Beijing University of Chemical Technology. His research interests include fault-tolerant control, state monitoring, modelling and control of biomedical processes (e.g., artificial pancreas system), and iterative learning control. He holds membership of two IFAC Technical Committees (TC6.1 and TC8.2). He was a recipient of several research awards, including the *Journal of Process Control* Survey Paper Prize and ADCHEM2015 Young Author Prize. He is also an (Associate) Editor of *Multidimensional Systems and Signal Processing* and the *Canadian Journal of Chemical Engineering*.