Conic Input Mapping Design of Constrained Optimal Iterative Learning Controller for Uncertain Systems

Yuanqiang Zhou, Kaihua Gao, Xiaopeng Tang, Huanjia Hu, Dewei Li, and Furong Gao

***Abstract*—In this article, we study the optimal iterative learning control (ILC) for constrained systems with bounded uncertainties via a novel conic input mapping (CIM) design methodology. Due to the limited understanding of the process of interest, modeling uncertainties are generally inevitable, significantly reducing the convergence rate of the control systems. However, huge amounts of measured process data interacting with model uncertainties can easily be collected. Incorporating these data into the optimal controller design could unlock new opportunities to reduce the error of the current trail optimization. Based on several existing optimal ILC methods, we incorporate the online process data into the optimal and robust optimal ILC design, respectively. Our methodology, called CIM, utilizes the process data for the first time by applying the convex cone theory and maps the data into the design of control inputs. CIM-based optimal ILC and robust optimal ILC methods are developed for uncertain systems to achieve better control performance and a faster convergence rate. Next, rigorous theoretical analyses for the two methods have been presented, respectively. Finally, two illustrative numerical examples are provided to validate our methods with improved performance.**

***Index Terms*—Data-driven approach, iterative learning control (ILC), optimization, process control, robust design.**

Manuscript received June 2, 2021; revised October 5, 2021 and January 12, 2022; accepted February 26, 2022. This work was supported in part by the Ministry of Science and Technology of China under Grant 2019YFB1704905 and Grant SQ2019YFB170029; in part by the Hong Kong Research Grant Council under Grant 16208520; in part by the Foshan-HKUST Project under Grant FSUST19-FYTRI01; in part by the Guangdong Scientific and Technological Project under Grant 202002030323 and Grant 2014B050505002; and in part by the SJTU Global Strategic Partnership Fund (2021 SJTU-HKUST). This article was recommended by Associate Editor W. X. Zheng. *(Corresponding author: Yuanqiang Zhou.)*

Yuanqiang Zhou, Kaihua Gao, Xiaopeng Tang, and Huanjia Hu are with the Department of Chemical and Biological Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: yqzhou@ust.hk; kgaoac@connect.ust.hk; xtangai@connect.ust.hk; hhuan@connect.ust.hk).

Dewei Li is with the Department of Automation and the Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: dwli@sjtu.edu.cn).

Furong Gao is with the Department of Chemical and Biological Engineering, Hong Kong University of Science and Technology, Hong Kong, and also with the Guangzhou HKUST Fok Ying Tung Research Institute, Guangzhou 511458, China (e-mail: kefgao@ust.hk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2022.3155754.

Digital Object Identifier 10.1109/TCYB.2022.3155754

I. Introduction

ITERATIVE learning control (ILC) is developed for a class of systems that repetitively perform a given task over a limited duration of time [1], [2]. It has been extensively studied in fundamental theory and practical applications [3], [4]. The main reason includes its ability to modulate the input signal by integrating the input and output information from past iterations/cycles/batches, which leads to gradual improvements of control performance for the systems [5]–[7]. Furthermore, the constant repetition of conducting the same task under the ILC framework makes the system acquires some kind of learning capability from its historical experience. This is the main difference between ILC and other intelligent control techniques [1]. Therefore, it is of great significance to study data-based learning of the system under the ILC strategy.

For common ILC strategies, controller design and stability analysis depend on previous information of the system [8]–[12]. Amann *et al.* [13] proposed an optimization-based ILC to derive gradient-type algorithms and it is a simple and effective control strategy for deterministic and iterative invariant systems. Son *et al.* [8] developed ILC frameworks for tracking problems with specified data points and ILC designs and convergence analyses are based on accurate information from system matrices. Freeman and Dinh [14] developed point-to-point ILC for discrete-time MIMO systems. The monotonic convergence of ILC was studied in [9] for nonlinear systems under globally Lipschitz conditions.

Recent work has focused on combining optimization approaches with ILC design methods. The existing literature on this topic is of various perspectives, including systems with different scenarios [5], [7], and different optimization methods, such as real-time optimization [15], [16], gradient-type optimization [17]–[19], model predictive control (MPC) [20], [21], etc. Therein, MPC has the ability to handle constraints and accommodate the performance objectives in an optimal way [22], [23], so its combination with ILC gives the control system full ability to take constraints into consideration and a certain ability to learn in predictive environments. Li *et al.* [24] proposed a combined design method of real-time feedback-based ILC and MPC for systems with unknown input nonlinearity. Liu *et al.* [25] proposed a combined design of ILC and robust MPC method for nonlinear systems. From a two-dimensional (2-D) system point of view, the combined design issues of ILC and

MPC have been widely studied for repetitive systems or batch processes (see, e.g., [21] and [26]–[28]).

With the rapid development of data science and machine learning, various data-based learning methods have enriched the ILC methodology. Hou *et al.* [29] pointed out that data-driven control methods are effective and applicable when the plant models are unavailable; even if the physical models are available, they may be too complex to be tractable for controller design. Lu *et al.* [30] reviewed different levels of learning mechanisms: control input, model parameter, and tracking reference, and the key point behind those learning mechanisms is to utilize a collection of inputs and/or outputs data to adjust the corresponding unknown information. The notion of learnability of a control system was introduced in [31] and some data-based learning schemes were developed using repetitiveness of the control system. An MPC-based ILC design method was proposed in [20] with a data-driven approach for uncertain batch processes and the historical input data were used in [32] to modulate the control input using an optimization-based ILC strategy. Two data-driven ILC schemes were developed in [33] for nonlinear discrete-time MIMO systems based on an equivalent dynamic linearized model of an unknown ideal learning controller, where the historical input and tracking error data were used to directly modulate the iterationwise control input. The aforementioned results have proved that with the integration of data-driven learning methods, system performance can be significantly improved and the convergence rate can be accelerated.

In this article, we address a data-enhanced controller design methodology for uncertain systems, which takes ILC, optimization, data-driven approach, robustness, and constraints handling into consideration. First, we present some existing optimal ILC and robust optimal ILC methods for uncertain systems. Most of the existing optimal ILC strategies are designed based on the nominal systems using the lifted techniques [34], [35] and robust convergence is guaranteed with a given matrix inequality related to the boundary of the uncertainties. Following the results, some extensions to robust optimal ILC methods are developed for uncertain systems. Several results about the method need to give an adequate uncertainty description of linear or nonlinear plants. Using ellipsoidal uncertainty and polytopic uncertainty description of the system, a robust ILC with a linear matrix inequality (LMI) approach is developed in [36] based on the worst case performance index. For an uncertain system, if knowledge of the compact set that contains any possible discrepancies is available, the robust optimal ILC strategy developed in [37] can be used based on the minimization of a dynamic upper bound on tracking error.

The main objective of this article is to develop a new data-driven learning design methodology that integrates historical process data into the design of ILCs with a focus on optimizing uncertain systems. For uncertain systems, the data-driven approach has some advantages, as historical input and/or output data contain information that interacts with uncertainties. Thus, historical process data can be compensated for the current trail optimization. Using cone theory to incorporate process data from previous cycles into current trail optimization, this article develops a new data-based design methodology, called conical input mapping (CIM). CIM uses the process data for the first time by utilizing convex cone theory and mapping the data into the optimal control input. With the CIM methodology incorporated into the optimization-based ILC, the optimization of control input can be transformed into the optimization of coefficients associated with the data, which improves the accuracy of the optimization model and reduces the error of the current trail optimization, ultimately improving robustness and control performance. Using CIM-based optimal ILC strategies, we attempt to accelerate the convergence rate of norm-optimal ILC algorithms and improve control performance for uncertain systems.

TABLE I
COMPARISONS BETWEEN BASIC RESULTS, ALGORITHMS 1 AND 2

	Section	Nominal model	Worst case	CIM
Basic results	Subsection II-B1	✓	×	×
	Subsection II-B2	×	✓	×
Algorithm 1	Section III	✓	×	✓
Algorithm 2	Section IV	×	✓	✓

The contribution of this article is threefold. First, different from the existing data-driven mechanisms [20], [33], [38], [39], we, for the first time, develop a CIM methodology for uncertain systems. This new methodology imposes admissible constraints in the domain of the input variables for data-driven ILC, thereby reducing the errors of the current trail optimization. Second, by employing the methodology, two different data-driven optimal ILC algorithms are developed: the former is a new CIM-based optimal ILC method and the latter is a new CIM-based robust optimal ILC method. Third, rigorous theoretical analyses for the two proposed algorithms are presented, respectively, demonstrating the feasibility of the algorithms and the monotonic convergence of control systems.

The remainder of this article is organized as follows. In Section II, we present the problem formulation and provide some basic optimal ILC methods. In Section III, we present the CIM design methodology, and then, a new CIM-based optimal ILC method (Algorithm 1) is developed and monotonic convergence of the system is analyzed. In Section IV, a novel CIM-based robust optimal ILC method using min–max optimization (Algorithm 2) is developed and monotonic convergence of the system is analyzed. In Section V, we provide two numerical examples to illustrate the efficacy of the two algorithms. Finally, we draw conclusions and discuss future orientations of the research in Section VI. For the sake of clarity and readability, the comparisons between our main results are illustrated in Table I.

The notation used in this article is fairly standard. Specifically, $\mathbb{R}$ denotes the real space, $\mathbb{N}$ denotes the collection of all non-negative natural numbers, and $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. We write $\mathbf{1}_N$ to denote the vector $[1 \ \cdots \ 1]^{\mathrm{T}} \in \mathbb{R}^N$ with N components. For a matrix $A \in \mathbb{R}^{n\times n}$, we write $A \succ 0$ and $A \succeq 0$ to denote that A is positive definite and positive semidefinite, respectively. For a vector $x \in \mathbb{R}^n$, x^{T} denotes its transpose, $\|x\|$ denotes the Euclidean norm, $\|M\|$ denotes the induced matrix norm for a

real matrix $M \in \mathbb{R}^{n\times m}$, and $\|x\|_P^2$ denotes the quadratic form $x^{\mathrm{T}}Px$ for a real symmetric and positive semidefinite matrix P. Furthermore, we use $\otimes$ to indicate the Kronecker product, and the symbol $\mathbb{I}_{[a,b]}$, $a, b \in \mathbb{N}$, is defined to be the integer set $\{a, a+1, \ldots, b\}$ with $a \le b$. Finally, we define $\mathbf{max}(x, y) = [\max(x_1, y_1), \max(x_2, y_2), \ldots, \max(x_n, y_n)]^{\mathrm{T}} \in \mathbb{R}^n$ and $\mathbf{min}(x, y) = [\min(x_1, y_1), \min(x_2, y_2), \ldots, \min(x_n, y_n)]^{\mathrm{T}} \in \mathbb{R}^n$, where $x = [x_1, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^n$ and $y = [y_1, \ldots, y_n]^{\mathrm{T}} \in \mathbb{R}^n$ are both n-dimensional real vectors.

II. Preliminaries

A. Problem Formulation

Consider a MIMO discrete-time system given by

$$x_k(t+1) = (A + \Delta_a(t))x_k(t) + (B + \Delta_b(t))u_k(t) \quad (1)$$

$$y_k(t) = Cx_k(t) \quad (2)$$

where $t \in \mathbb{I}_{[0,N]}$ is the time index, $k \in \mathbb{I}_{[1,\infty)}$ is the iteration index, N is the trail length of each iteration; $x_k(t) \in \mathbb{R}^n$, $u_k(t) \in \mathbb{R}^m$, and $y_k(t) \in \mathbb{R}^p$ are the state vector, control input, and measured output, respectively, at time t in the kth iteration; A, B, and C are the system matrices with appropriate dimensions; and $\Delta_a(t)$ and $\Delta_b(t)$ denote the unknown uncertainties. For the industrial process described as (1) and (2), we consider the following input constraints as:

$$u_{i,k}(t) \in \mathbb{U}_i \triangleq \left[\underline{u}_i, \overline{u}_i\right], \quad \underline{u}_i < \overline{u}_i \quad (3)$$

$$u_{i,k}(t) - u_{i,k}(t-1) \in \delta\mathbb{U}_i \triangleq \left[\underline{\delta u}_i, \overline{\delta u}_i\right], \underline{\delta u}_i < 0 < \overline{\delta u}_i \quad (4)$$

$$u_{i,k}(t) - u_{i,k-1}(t) \in \Delta\mathbb{U}_i \triangleq \left[\underline{\mathrm{d}u}_i, \overline{\mathrm{d}u}_i\right], \underline{\mathrm{d}u}_i < 0 < \overline{\mathrm{d}u}_i \quad (5)$$

where $u_{i,k}(t)$ denotes the ith component of $u_k(t)$, that is, $u_k(t) = [u_{1,k}(t), \ldots, u_{i,k}(t), \ldots, u_{m,k}(t)]$, $i \in \mathbb{I}_{[1,m]}$.

The control objective is to steer the system output to track a given reference. We denote the desired reference as $y_{\mathrm{d}}(t)$, $t \in \mathbb{I}_{[0,N]}$ and assume that $y_{\mathrm{d}}(t)$ is realizable; that is, there is a suitable initial state $x_{\mathrm{d}}(0)$ and unique input sequence $u_{\mathrm{d}}(t) \in \mathbb{U} \triangleq \prod_{i=1}^m \mathbb{U}_i \subset \mathbb{R}^m$ and $\prod_{i=1}^m \delta\mathbb{U}_i$ such that

$$x_{\mathrm{d}}(t+1) = Ax_{\mathrm{d}}(t) + Bu_{\mathrm{d}}(t) \quad (6)$$

$$y_{\mathrm{d}}(t+1) = Cx_{\mathrm{d}}(t+1). \quad (7)$$

In addition, suppose that the matrix CB is of full-column rank, then the desired input $u_{\mathrm{d}}(t)$ can be well defined as $u_{\mathrm{d}}(t) = [(CB)^{\mathrm{T}}CB]^{-1}(CB)^{\mathrm{T}}(y_{\mathrm{d}}(t+1) - CAx_{\mathrm{d}}(t))$ and $x_{\mathrm{d}}(t+1) = Ax_{\mathrm{d}}(t) + Bu_{\mathrm{d}}(t)$ with the desired initial value $x_{\mathrm{d}}(0)$.

Assumption 1: For all $t \in \mathbb{I}_{[0,N]}$, the uncertainties $\Delta_a(t)$ and $\Delta_b(t)$ are bounded; that is, $\max_t \|\Delta_a(t)\| \le \beta_a$, $\max_t \|\Delta_b(t)\| \le \beta_b$.

Assumption 2: The identical initialization condition holds for all iterations; that is, $x_k(0) = 0\ \forall k \in \mathbb{I}_{[1,\infty)}$.

For (1) and (2), the majority of existing optimal ILC formulations focus on designing an optimal timewise control input increment $\delta u_k(t) \triangleq u_k(t) - u_k(t-1)$ instead of the control input $u_k(t)$. Thus, we rewrite (1) and (2) in the general augmented state-space form given by

$$\mathbf{x}_k(t+1) = (\mathbf{A} + \Delta_{\mathbf{a}}(t))\mathbf{x}_k(t) + (\mathbf{B} + \Delta_{\mathbf{b}}(t))\delta u_k(t) \quad (8)$$

$$y_k(t+1) = \mathbf{C}\mathbf{x}_k(t+1) \quad (9)$$

where $\mathbf{x}_k(t) = [x_k^{\mathrm{T}}(t), u_k^{\mathrm{T}}(t-1)]^{\mathrm{T}} \in \mathbb{R}^{n+m}$ and the augmented matrices $\mathbf{A} \in \mathbb{R}^{(n+m)\times(n+m)}$, $\Delta_{\mathbf{a}}(t) \in \mathbb{R}^{(n+m)\times(n+m)}$, $\mathbf{B} \in \mathbb{R}^{(n+m)\times m}$, $\Delta_{\mathbf{b}}(t) \in \mathbb{R}^{(n+m)\times m}$, and $\mathbf{C} \in \mathbb{R}^{p\times(n+m)}$ are given by

$$\mathbf{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \quad \Delta_{\mathbf{a}}(t) = \begin{bmatrix} \Delta_a(t) & \Delta_b(t) \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} B \\ I \end{bmatrix}, \quad \Delta_{\mathbf{b}}(t) = \begin{bmatrix} \Delta_b(t) \\ 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} C & 0 \end{bmatrix}.$$

To facilitate the following design for (8) and (9), we define $\mathbf{u}_k = [\delta u_k^{\mathrm{T}}(0), \ldots, \delta u_k^{\mathrm{T}}(N-1)]^{\mathrm{T}}$, $\mathbf{y}_k = [y_k^{\mathrm{T}}(1), \ldots, y_k^{\mathrm{T}}(N)]^{\mathrm{T}}$, $\mathbf{u}_{\mathrm{d}} = [\delta u_{\mathrm{d}}^{\mathrm{T}}(0), \ldots, \delta u_{\mathrm{d}}^{\mathrm{T}}(N-1)]^{\mathrm{T}}$, $\mathbf{y}_{\mathrm{d}} = [y_{\mathrm{d}}^{\mathrm{T}}(1), \ldots, y_{\mathrm{d}}^{\mathrm{T}}(N)]^{\mathrm{T}}$, the lifted system matrix

$$\mathcal{H} \triangleq \begin{bmatrix} \mathbf{CB} & 0 & \cdots & 0 \\ \mathbf{CAB} & \mathbf{CB} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} & \cdots & \mathbf{CB} \end{bmatrix} \quad (10)$$

and the lifted uncertainty

$$\Xi_k \triangleq \left[\xi_k^{\mathrm{T}}(1), \ldots, \xi_k^{\mathrm{T}}(N)\right]^{\mathrm{T}} \quad (11)$$

with $\xi_k(t)$, $t = 1, \ldots, N$, defined as

$$\xi_k(t) \triangleq y_k(t) - \mathbf{CA}^t\mathbf{x}_k(0) - \sum_{\tau=0}^{t-1} \mathbf{CA}^{t-1-\tau}\mathbf{B}\delta u_k(\tau). \quad (12)$$

Note that if $\Delta_a(t) = 0$ and $\Delta_b(t) = 0$ for all $t \in \mathbb{I}_{[0,N]}$ in (1) and (2), we obtain that $\xi_k(t) = 0\ \forall t$. Now, the lifted form of the system (1) and (2) and the desired system (6) and (7) can be expressed as

$$\mathbf{y}_k = \mathcal{M}\mathbf{x}_k(0) + \mathcal{H}\mathbf{u}_k + \Xi_k \quad (13)$$

$$\mathbf{y}_{\mathrm{d}} = \mathcal{M}\mathbf{x}_{\mathrm{d}}(0) + \mathcal{H}\mathbf{u}_{\mathrm{d}} \quad (14)$$

where $\mathcal{M} = [(\mathbf{CA})^{\mathrm{T}}, \ldots, (\mathbf{CA}^N)^{\mathrm{T}}]^{\mathrm{T}}$.

Next, we consider the exact form of the lifted uncertainty $\xi_k(t)$ using the uncertainty matrices $\Delta_{\mathbf{a}}(t)$ and $\Delta_{\mathbf{b}}(t)$. First, for any time t, using induction, we obtain

$$\xi_k(t) = \mathbf{C}\left[\prod_{\tau=0}^{t-1}(\mathbf{A} + \Delta_{\mathbf{a}}(\tau)) - \mathbf{A}^t\right]\mathbf{x}_k(0) + \mathbf{C}\sum_{\tau=0}^{t-1}\left[\left(\prod_{s=\tau+1}^{t-1}(\mathbf{A} + \Delta_{\mathbf{a}}(s))\right)(\mathbf{B} + \Delta_{\mathbf{b}}(\tau)) - \mathbf{A}^{t-1-\tau}\mathbf{B}\right]\delta u_k(\tau).$$

Using Assumption 2 and $u_k(-1) = 0$, we obtain that $\mathbf{x}_k(0) = [x_k^{\mathrm{T}}(0), u_k^{\mathrm{T}}(-1)]^{\mathrm{T}} = 0$. Then, we can simplify $\xi_k(t)$ as

$$\xi_k(t) = \sum_{\tau=0}^{t-1}\left[\mathbf{C}\left(\prod_{s=\tau+1}^{t-1}(\mathbf{A} + \Delta_{\mathbf{a}}(s))\right)(\mathbf{B} + \Delta_{\mathbf{b}}(\tau)) - \mathbf{CA}^{t-1-\tau}\mathbf{B}\right]\delta u_k(\tau) \quad (15)$$

which is a linear combination of the unknown matrices $\{\Delta_{\mathbf{a}}(\tau), \Delta_{\mathbf{b}}(\tau), \tau \in \mathbb{I}_{[0,t-1]}\}$ and the input sequence $\{\delta u_k(0), \ldots, \delta u_k(t-1)\}$.

Considering $t = 0, 1, \ldots, N-1$ in (15) and by concatenating to form (11), we end up with

$$\Xi_k = \Delta\mathcal{H}\mathbf{u}_k \tag{16}$$

where $\Delta\mathcal{H} \in \mathbb{R}^{Np\times Nm}$ is a lower triangular coefficient matrix determined by $\{\Delta_{\mathbf{a}}(t), \Delta_{\mathbf{b}}(t), t \in \mathbb{I}_{[0,N]}\}$ and each block-entry of $\Delta\mathcal{H}$ is given by

$$[\Delta\mathcal{H}]_{i,j} \triangleq \begin{cases} \mathbf{C}\Big(\prod_{s=j}^{i}(\mathbf{A}+\Delta_{\mathbf{a}}(s))\Big)(\mathbf{B}+\Delta_{\mathbf{b}}(\tau)) \\ \quad -\mathbf{C}\mathbf{A}^{i-j}\mathbf{B}, & \text{if } i \geq j \\ \mathbf{0}, & \text{otherwise} \end{cases}$$

where $i, j \in \mathbb{I}_{[1,N]}$. Note that if $\Delta_{\mathbf{a}}(t) = 0$ and $\Delta_{\mathbf{b}}(t) = 0$ for all $t \in \mathbb{I}_{[0,N]}$ in (8) and (9), we obtain that $[\Delta\mathcal{H}]_{i,j} = 0$ for all $i, j \in \mathbb{I}_{[1,N]}$, indicating that $\Delta\mathcal{H} = \mathbf{0}$.

Using Assumption 1, we obtain that $\Delta_{\mathbf{a}}(t)$ and $\Delta_{\mathbf{b}}(t)$ are bounded with $\|\Delta_{\mathbf{a}}(t)\| \leq \beta_a + \beta_b$ and $\|\Delta_{\mathbf{b}}(t)\| \leq \beta_b$ for all $t \in \mathbb{I}_{[0,N]}$. Thus, $\Delta\mathcal{H}$ in (16) is bounded; that is, there exists $\beta_{\Delta\mathcal{H}} > 0$, which is a function of β_a and β_b such that

$$\|\Delta\mathcal{H}\| \leq \beta_{\Delta\mathcal{H}} \triangleq \max_{i,j}\Big\|[\Delta\mathcal{H}]_{i,j}\Big\|. \tag{17}$$

In addition, $\Delta\mathcal{H}$ is stationary and independent along the iteration axis.

Assumption 3: The initial error is bounded, that is, $\|x_k(0) - x_{\mathrm{d}}(0)\| \leq \beta_0$.

Using (13)–(14) and (16), the entire tracking error of the kth iteration is given by

$$\mathbf{e}_k \triangleq \mathbf{y}_{\mathrm{d}} - \mathbf{y}_k = \mathcal{M}\mathbf{x}_{\mathrm{d}}(0) + \mathcal{H}\delta\mathbf{u}_k - \Delta\mathcal{H}\mathbf{u}_k \tag{18}$$

where $\delta\mathbf{u}_k = \mathbf{u}_{\mathrm{d}} - \mathbf{u}_k$. In this case, the transition model for tracking error trajectory is obtained as

$$\mathbf{e}_k = \mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k - \Delta\mathcal{H}\Delta\mathbf{u}_k \tag{19}$$

where $\Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$ denotes the cyclewise control input along the iteration axis, that is

$$\Delta\mathbf{u}_k = \left[\Delta\delta u_k^{\mathrm{T}}(0), \ldots, \Delta\delta u_k^{\mathrm{T}}(N-1)\right]^{\mathrm{T}} \tag{20}$$

where $\Delta\delta u_k(t) \triangleq \delta u_k(t) - \delta u_k(t-1)$.

Next, all the constraints given by (3)–(5) can be rewritten as

$$\Gamma\Delta\mathbf{u}_k \leq \Lambda_k \tag{21}$$

where $\Gamma \in \mathbb{R}^{4Nm\times Nm}$ and $J \in \mathbb{R}^{Nm\times Nm}$ are given by

$$\Gamma = \begin{bmatrix} I \\ -I \\ J \\ -J \end{bmatrix}, \quad J = \begin{bmatrix} I & 0 & \cdots & 0 & 0 \\ -I & I & \cdots & 0 & 0 \\ 0 & -I & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -I & I \end{bmatrix}$$

and I is the identity matrix with appropriate dimension, the vector $\Lambda_k \in \mathbb{R}^{4Nm}$ is given by

$$\Lambda_k = \begin{bmatrix} \mathbf{min}\big(\overline{\mathbf{u}} - \mathbf{u}_{k-1}, \overline{\mathrm{d}\mathbf{u}}\big) \\ -\mathbf{max}\big(\underline{\mathbf{u}} - \mathbf{u}_{k-1}, \underline{\mathrm{d}\mathbf{u}}\big) \\ \overline{\delta\mathbf{u}} - J\mathbf{u}_{k-1} \\ -\underline{\delta\mathbf{u}} + J\mathbf{u}_{k-1} \end{bmatrix}$$

where $\underline{\mathbf{u}} = \mathbf{1}_N \otimes \underline{u}$, $\overline{\mathbf{u}} = \mathbf{1}_N \otimes \overline{u}$, $\underline{\mathrm{d}\mathbf{u}} = \mathbf{1}_N \otimes \underline{\mathrm{d}u}$, $\overline{\mathrm{d}\mathbf{u}} = \mathbf{1}_N \otimes \overline{\mathrm{d}u}$, $\underline{\delta\mathbf{u}} = \mathbf{1}_N \otimes \underline{\delta u}$, $\overline{\delta\mathbf{u}} = \mathbf{1}_N \otimes \overline{\delta u}$, $\underline{u} = [\underline{u}_1, \ldots, \underline{u}_m]^{\mathrm{T}}$, $\overline{u} = [\overline{u}_1, \ldots, \overline{u}_m]^{\mathrm{T}}$, $\overline{\delta u} = [\overline{\delta u}_1, \ldots, \overline{\delta u}_m]^{\mathrm{T}}$, $\underline{\delta u} = [\underline{\delta u}_1, \ldots, \underline{\delta u}_m]^{\mathrm{T}}$, $\underline{\mathrm{d}u} = [\underline{\mathrm{d}u}_1, \ldots, \underline{\mathrm{d}u}_m]^{\mathrm{T}}$, and $\overline{\mathrm{d}u} = [\overline{\mathrm{d}u}_1, \ldots, \overline{\mathrm{d}u}_m]^{\mathrm{T}}$. Note that $\Lambda_k \geq 0$, which indicates that (21) always holds with $\Delta\mathbf{u}_k = 0$.

B. Basic Results

1) Optimal ILC Strategy: We suppose that Assumptions 1–3 are satisfied. The idea is to optimize $\Delta\mathbf{u}_k$ during the ILC design, which does not take the model uncertainty into consideration. At each iteration, the following quadratic performance index is minimized to obtain the cyclewise control input:

$$\min_{\Delta\mathbf{u}_k} \ \|\tilde{\mathbf{e}}_k\|_{\mathbf{Q}}^2 + \|\Delta\mathbf{u}_k\|_{\mathbf{R}}^2 \tag{22a}$$

$$\text{s.t.} \ \ \tilde{\mathbf{e}}_k = \mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k \tag{22b}$$

$$\Gamma\Delta\mathbf{u}_k \leq \Lambda_k. \tag{22c}$$

After solving the optimization problem (22), we obtain the optimal solution, denoted as $\Delta\mathbf{u}_k^{\circ}$, and the optimized state $\tilde{\mathbf{e}}_i^{\circ} = \mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k$. Then, we apply the following ILC algorithm:

$$\mathbf{u}_k^{\circ} = \mathbf{u}_{k-1} + \Delta\mathbf{u}_k^{\circ} \tag{23}$$

to the system (1) and (2) at the kth iteration. For more details about convergence analysis of this kind of optimal ILC strategies, one can refer to [34] and [35].

2) Robust Optimal ILC Strategy: We suppose that Assumptions 1–3 are satisfied. In order to incorporate model uncertainty information explicitly into the ILC design, the worst case performance index is considered. At each iteration, the following quadratic performance index is minimized to obtain the cyclewise robust control input:

$$\min_{\Delta\mathbf{u}_k}\max_{\Delta\mathcal{H}} \ \|\tilde{\mathbf{e}}_k\|_{\mathbf{Q}}^2 + \|\Delta\mathbf{u}_k\|_{\mathbf{R}}^2 \tag{24a}$$

$$\text{s.t.} \ \ \tilde{\mathbf{e}}_k = \mathbf{e}_{k-1} - (\mathcal{H}+\Delta\mathcal{H})\Delta\mathbf{u}_k \tag{24b}$$

$$\Gamma\Delta\mathbf{u}_k \leq \Lambda_k. \tag{24c}$$

After solving the optimization problem (24), the robust optimal ILC algorithm given by

$$\mathbf{u}_k^{\diamond} = \mathbf{u}_{k-1} + \Delta\mathbf{u}_k^{\diamond} \tag{25}$$

is applied to the system (1) and (2) at the kth iteration. For more details about convergence analysis of this kind of robust optimal ILC strategies, one can refer to [36] and [37].

Remark 1: Note that the above two strategies are one-stage approaches, which only need to optimize the input vector $\Delta\mathbf{u}_k$ given in (19) for each iteration. Then, the control inputs $\mathbf{u}_k$ of the kth iteration will be implemented. Therefore, they are run-to-run ILC strategies since they can calculate the updating laws at the beginning of each iteration [2], [24]. Different from the run-to-run ILC strategies, the real-time feedback-based ILC strategies consider the optimization of timewise input vectors (see, e.g., [40] and [41]). However, if the optimization horizon is equal to the trail length of each iteration, they can be modified to the problem formulations given by (22) or (24).

III. CIM-Based Optimal ILC Design

For the uncertain system given by (19), the process data, including the historical input and output sequences, constrain the information, which interacts with the uncertainties. Thus, incorporating these data into the optimization-based ILC can reduce the errors of the current optimization. To that end, the CIM design methodology is presented using convex set and cone theory in this section. Next, the CIM-based optimal ILC strategy is developed and the theoretical analysis of the strategy is presented.

A. CIM Methodology

In convex geometry [42], a *convex combination* is a linear combination of distinct points with the form $\sum_{i=1}^{r} \alpha_i s_i$, where $\sum_{i=1}^{r} \alpha_i = 1$, $\alpha_i \geq 0$, $i = 1, \ldots, r$, and s_i can be vector, scalar, or more general point in an affine space. A set is convex if it contains all convex combinations of its points. As pointed out in [43], the *convex hull* of a set is the set of all convex combinations of points in the set.

In cone theory [43], a set $\mathbb{S}$ is called a *cone*, if for every $s \in \mathbb{S}$ and $\alpha \geq 0$, we have $\alpha s \in \mathbb{S}$. A set $\mathbb{S}$ is a *convex cone* if it is convex and a cone, that is, for any $s_1, s_2 \in \mathbb{S}$ and $\alpha_1, \alpha_2 \geq 0$, we have $\alpha_1 s_2 + \alpha_2 s_2 \in \mathbb{S}$. A *conic combination* is a linear combination of distinct points provided that all coefficients are non-negative. The *conic hull* of a set is the set of all conic combinations of points in the set. More formally, given a finite set $\mathbb{S} = \{s_1, s_2, \ldots, s_n\}$, where $s_i \in \mathbb{R}^q$, $i = 1, 2, \ldots, r$, the *conic hull* of $\mathbb{S}$ is given by

$$\mathcal{CH}(\mathbb{S}) \triangleq \left\{ s \in \mathbb{R}^q : s = \sum_{i=1}^{r} \alpha_i s_i : \forall s_i \in \mathbb{S} \right\} \tag{26}$$

where $\alpha_i \geq 0$, $i = 1, 2, \ldots, r$. It follows from [43] that $\mathcal{CH}(\mathbb{S})$ is the smallest convex cone that contains $\mathbb{S}$. In addition, the convex cone of a given set of points is identical to the set of all their conic combinations.

To incorporate input updates of previous trails into the current optimization, we let $\mathbb{S}_k$ denote the historical control input sequences as

$$\mathbb{S}_k = \{\Delta \mathbf{u}_1, \ldots, \Delta \mathbf{u}_{k-1}\}. \tag{27}$$

In terms of (26), we have $s_i = \Delta \mathbf{u}_i \in \mathbb{S}_k \subset \mathbb{R}^{Nm}$ with $q = Nm$ and $r = k - 1$ and using (19), $\Delta \mathbf{u}_i$, $i = 1, \ldots, k-1$, satisfies the real uncertain dynamics, i.e.,

$$\mathbf{e}_i = \mathbf{e}_{i-1} - \mathcal{H} \Delta \mathbf{u}_i - \Delta \mathcal{H} \Delta \mathbf{u}_i. \tag{28}$$

Then, using the conic hull (26) and the data set (27), we design the cyclewise control input as

$$\Delta \mathbf{u}_k = \mathcal{CH}(\mathbb{S}_k) + \zeta_k \tag{29}$$

where

$$\mathcal{CH}(\mathbb{S}_k) = \sum_{i=1}^{k-1} \alpha_i \Delta \mathbf{u}_i \tag{30}$$

and $\alpha_i \geq 0$, $i = 1, \ldots, k-1$; the vector $\zeta_k \in \mathbb{R}^{Nm}$ denotes an auxiliary control input vector, which can be seen as a projection of $\Delta \mathbf{u}_k$ on the conic hull $\mathcal{CH}(\mathbb{S}_k)$ given by (26) using (27). Besides, the term ζ_k in (29) is an additive term to the conic hull $\mathcal{CH}(\mathbb{S}_k)$ for feasibility consideration, but in practical implementation, it can be set to 0.

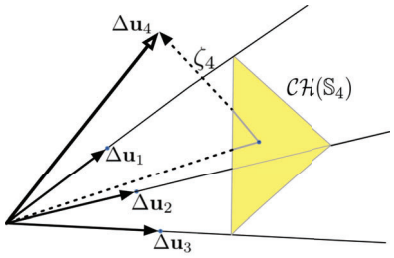


Fig. 1. Illustration of $\Delta \mathbf{u}_4 = \mathcal{CH}(\mathbb{S}_4) + \zeta_4 \in \mathbb{R}^3$, where $\mathcal{CH}(\mathbb{S}_4)$ denotes the conic hull of the vector set $\{\Delta \mathbf{u}_1, \Delta \mathbf{u}_2, \Delta \mathbf{u}_3\}$.

It is important to note that the basic results presented in Section II-B only optimize the current control input $\Delta \mathbf{u}_k$, whereas using (29), we can map the current control input $\Delta \mathbf{u}_k$ into the conic combination of the historical control input data $\{\Delta \mathbf{u}_1, \Delta \mathbf{u}_2, \ldots, \Delta \mathbf{u}_{k-1}\}$ and optimize the coefficients α_i associated with each data $\Delta \mathbf{u}_i$, $i = 1, 2, \ldots, k-1$. It is this change that allows us to enforce the constraints for the current control input $\Delta \mathbf{u}_k$ and reduce the error of current trail optimization. An example of $\Delta \mathbf{u}_4 \in \mathbb{R}^3$ using (29) with $k = 4$ is illustrated in Fig. 1.

B. CIM-Based Optimal ILC Design

In this section, we incorporate the historical input data into the CIM-based optimal ILC design. For the optimization formulation (22) presented in Section II-B1, using (26) and (30) and substituting (30) into (22b) yields

$$\begin{aligned} \hat{\mathbf{e}}_k &= \mathbf{e}_{k-1} - \mathcal{H}\left(\sum_{i=1}^{k-1} \alpha_i \Delta \mathbf{u}_i + \zeta_k \right) \\ &= \mathbf{e}_{k-1} - \left(\sum_{i=1}^{k-1} \alpha_i \mathcal{H} \Delta \mathbf{u}_i \right) - \mathcal{H} \zeta_k. \end{aligned} \tag{31}$$

Then, the minimization of the performance index (22a) with respect to $\Delta \mathbf{u}_k$ is transferred into the minimization of the index with respect to the coefficients $\{\alpha_1, \ldots, \alpha_{k-1}\}$.

Finally, using (31), a newly developed optimization problem that utilizes historical input data by the following conic hull theory can be formulated as:

$$\min_{\alpha_1, \ldots, \alpha_{k-1}, \zeta_k} \quad \left\| \hat{\mathbf{e}}_k \right\|_{\mathbf{Q}}^2 + \left\| \Delta \mathbf{u}_k \right\|_{\mathbf{R}}^2 \tag{32a}$$

$$\text{s.t.} \quad (31) \tag{32b}$$

$$\Delta \mathbf{u}_k = \sum_{i=1}^{k-1} \alpha_i \Delta \mathbf{u}_i + \zeta_k \tag{32c}$$

$$\alpha_i \geq 0, \ \ i = 1, \ldots, k-1 \tag{32d}$$

$$\Gamma \Delta \mathbf{u}_k \leq \Lambda_k \tag{32e}$$

where the performance index given in (32a) is used to minimize the tracking error and the cyclewise control input; $\mathbf{Q} \succcurlyeq 0$ and $\mathbf{R} \succ 0$ are symmetric sign-definite weighting matrices

with compatible dimensions. Note that the weighting matrices $\{\mathbf{Q}, \mathbf{R}\}$ serve as tuning knobs to achieve a proper tradeoff between the tracking error and the control input.

For the optimization problem (32), we highlight that it employs cone theory to incorporate error updates of previous trails of solving (22) into the current trail optimization. To be more specific, using (28) and (23), we obtain $\mathcal{H}\Delta\mathbf{u}_i^\circ = \mathbf{e}_{i-1} - \mathbf{e}_i - \Delta\mathcal{H}\Delta\mathbf{u}_i^\circ$ and substituting it into (32b) yields

$$\hat{\mathbf{e}}_k = \mathbf{e}_{k-1} + \left[\sum_{i=1}^{k-1} \alpha_i\big(\mathbf{e}_i - \mathbf{e}_{i-1} + \Delta\mathcal{H}\Delta\mathbf{u}_i^\circ\big)\right] - \mathcal{H}\zeta_k. \quad (33)$$

For the optimization problem (22), we obtain that for each $i = 1, \ldots, k-1$, the real error $\mathbf{e}_i$ in (19) satisfies $\mathbf{e}_i = \tilde{\mathbf{e}}_i^\circ - \Delta\mathcal{H}\Delta\mathbf{u}_i^\circ$, where $\tilde{\mathbf{e}}_i^\circ$ is the optimized state after solving (22). Then, we obtain $\tilde{\mathbf{e}}_i^\circ - \mathbf{e}_{i-1} = \mathbf{e}_i - \mathbf{e}_{i-1} + \Delta\mathcal{H}\Delta\mathbf{u}_i^\circ$. Thus, (33) implies that

$$\hat{\mathbf{e}}_k = \mathbf{e}_{k-1} + \left[\sum_{i=1}^{k-1} \alpha_i\big(\tilde{\mathbf{e}}_i^\circ - \mathbf{e}_{i-1}\big)\right] - \mathcal{H}\zeta_k. \quad (34)$$

Note that when $\zeta_k = 0$, the current to-be-optimized state $\hat{\mathbf{e}}_k$ satisfies $\hat{\mathbf{e}}_k - \mathbf{e}_{k-1} \in \mathcal{CH}(\{\tilde{\mathbf{e}}_i^\circ - \mathbf{e}_{i-1}\}_{i=1}^{k-1})$, which confines the error $\hat{\mathbf{e}}_k - \mathbf{e}_{k-1}$ to be the minimal *conic hull* of the optimized error data set $\{\tilde{\mathbf{e}}_i^\circ - \mathbf{e}_{i-1}\}_{i=1}^{k-1}$. This indicates that the optimized error data of the optimization problem (22) is adopted to compensate for the to-be-optimized state of the optimization problem (32). To guarantee the feasibility of the optimization problem (32), we do not force ζ_k to be 0. By using the cone theory, the error of the current trail optimization problem (32) is reduced compared with the problem (22). Thus, the optimization problem (32) can improve the control performance using (30) with the CIM methodology.

Remark 2 [Implementation for Solving (32)]: For the developed optimization problem (32), it is a general QP problem. To solve the problem, *fmincon*, a nonlinear programming solver in MATLAB, can be adopted.

C. CIM-Based Optimal ILC Algorithm

After solving the optimization problem (32), we obtain the optimal solution, denoted as $\{\alpha_1^\star, \ldots, \alpha_{k-1}^\star\}$ and $\zeta_k^\star$. Then, using (32c), the CIM-based optimal input is given by

$$\Delta\mathbf{u}_k^\star = \sum_{i=1}^{k-1} \alpha_i^\star \Delta\mathbf{u}_i + \zeta_k^\star. \quad (35)$$

Next, we apply the optimized control input given by

$$\mathbf{u}_k^\star = \mathbf{u}_{k-1} + \sum_{i=1}^{k-1} \alpha_i^\star \Delta\mathbf{u}_i + \zeta_k^\star \quad (36)$$

to the system at the kth iteration and repeat the procedure at the next iteration. The CIM-based optimal ILC strategy can be summarized as the following Algorithm 1.

Remark 3: Note that all the parameters $\{\alpha_1, \ldots, \alpha_{k-1}, \zeta_k\}$ in (30) are the design variables that are to be optimized in (32). In particular, the parameters $\{\alpha_1, \ldots, \alpha_{k-1}\}$ are the non-negative coefficients in (26) used to construct the *conic hull* of the set $\mathbb{S}_k$, which contains its historical input sequences $\{\Delta\mathbf{u}_1, \ldots, \Delta\mathbf{u}_{k-1}\}$. The parameter ζ_k given in (32c) is a free vector in the admissible input space within $\mathbb{R}^{Nm}$, which can be designed as 0 for the conic hull $\mathcal{CH}(\mathbb{S}_k)$ in a practical implementation. Designing the control input by following the CIM methodology given by (30) will provide constraints in the domain of the admissible input variables for data-driven ILC strategies.

Algorithm 1 CIM-Based Optimal ILC Algorithm

0) Form the lifted matrix $\mathcal{H}$ given by (10) and select weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ to satisfy (38).
1) Set iteration index $k = 1$ and initialize $\mathbf{u}_0$.
2) Apply $\mathbf{u}_k$ to the system, measure output $\mathbf{y}_k$, and compute tracking error $\mathbf{e}_k$.
3) Obtain the optimal parameters $\{\alpha_1^\star, \ldots, \alpha_{k-1}^\star\}$ and vector $\zeta_k^\star$ by solving the optimization problem (32).
4) Apply (36) to the system and update the data set (27).
5) Set $k \leftarrow k+1$ and go back to step 2).

Remark 4 (Comparison of Computational Complexity): The computation complexity of Algorithm 1 is compared with the optimal ILC strategy presented in Section II-B1. In the optimal ILC strategy, the input vector $\Delta\mathbf{u}_k$ for each iteration is obtained by solving the optimization problem (22), which is a general QP problem. In general, a QP problem with ρ variables has computational complexity $\mathcal{Q}(\rho^2)$. For the QP problem given by (22), the total number of variables at each update is mN. Consequently, the computational complexity for the optimal ILC strategy is $\mathcal{Q}((mN)^2)$. Next, for Algorithm 1, the optimal solution is obtained by solving the new QP problem (32). Since the total number of variables at each update for solving the QP problem (32) is $(k-1+m)$, the computational complexity of Algorithm 1 is $\mathcal{Q}((k-1+m)^2)$. Note that the number of variables increases slowly in the size of the QP problem (32), indicating that the computing time for executing Algorithm 1 increases slowly with more data collecting during the process running. Thus, since the trail length N of each iteration is much larger than the dimension m of the input variable and if the iteration index $k \ll (m(N-1)+1)$, Algorithm 1 achieves better computational efficiency than the optimal ILC strategy.

D. Theoretical Analysis

This section analyzes the feasibility and monotonic convergence of Algorithm 1.

Lemma 1: Consider the system given by (1) and (2) subject to (3)–(5) and controlled by Algorithm 1. If Assumptions 1–3 hold and the optimization problem given by (32) is feasible at iteration k, then Algorithm 1 is always feasible.

Proof: Consider any iteration k such that the problem given by (32) has a feasible solution, denoted by $\{\alpha_1^\star, \ldots, \alpha_{k-1}^\star, \zeta_k^\star\}$. Then, the optimal control input $\mathbf{u}_k^\star$ given by (36) satisfies all the constraints and is implemented for the iteration k. Next, at the end of iteration k, we update (32c) as

$$\Delta\mathbf{u}_{k+1} = \sum_{i=1}^{k-1} \alpha_i \Delta\mathbf{u}_i + \alpha_k \Delta\mathbf{u}_k + \zeta_{k+1} \quad (37)$$

where $\alpha_i = \alpha_i^\star$, $i = 1, \ldots, k-1$, $\alpha_k = 0$, and $\zeta_{k+1} = \zeta_k^\star$. In this case, for iteration $k+1$ with the constructed solution given by (37), we obtain that all the constraints given by (32b)–(32e) are satisfied except for the minimization of the performance measure (32a). Thus, there is at least one admissible solution for the problem given by (32) at iteration $k+1$; that is, it is feasible at iteration $k+1$. Finally, by mathematical induction, we obtain that Algorithm 1 is always feasible. This completes the proof. ■

Lemma 1 shows that constructing the to-be-optimized control input by utilizing the conic hull of its historical input sequences will not affect the solvability of the optimization problem; that is, the optimization problem (32) is still feasible for each iteration. Next, the following proposition shows that the actual tracking error $\|\mathbf{e}_k\|_{\mathbf{Q}}^2$ under Algorithm 1 is upper bounded.

Proposition 1: Consider the system given by (1) and (2) and assume that Assumptions 1–3 hold. If the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ in (32) are designed to satisfy the following condition:

$$\beta_{\Delta\mathcal{H}}^2 \|\mathbf{Q}\| \leq \|\mathbf{R}\|. \tag{38}$$

Then, the actual tracking error $\|\mathbf{e}_k\|_{\mathbf{Q}}^2$ under Algorithm 1 is upper bounded by

$$\left\|\mathbf{e}_k\right\|_{\mathbf{Q}}^2 \leq J_1(\Delta\mathbf{u}_k) \tag{39}$$

for any k, where $J_1(\Delta\mathbf{u}_k)$ is defined as

$$J_1(\Delta\mathbf{u}_k) \triangleq \left\|\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k\right\|_{\mathbf{Q}}^2 + \left\|\Delta\mathbf{u}_k\right\|_{\mathbf{R}}^2. \tag{40}$$

Proof: To execute Algorithm 1, we need to solve the optimization problem (32). For (32), we note that $\hat{\mathbf{e}}_k = \mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k$. But, using the actual model (19), it follows that:

$$\begin{aligned}
\left\|\mathbf{e}_k\right\|_{\mathbf{Q}}^2 &= \left\|\mathbf{e}_{k-1} - (\mathcal{H}+\Delta\mathcal{H})\Delta\mathbf{u}_k\right\|_{\mathbf{Q}}^2 \\
&\leq \max_{\Delta\mathcal{H}} \left\|\mathbf{e}_{k-1} - (\mathcal{H}+\Delta\mathcal{H})\Delta\mathbf{u}_k\right\|_{\mathbf{Q}}^2 \\
&= \max_{\Delta\mathcal{H}} \big[(\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k)^{\mathrm{T}}\mathbf{Q}(\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k) \\
&\qquad + \Delta\mathbf{u}_k^{\mathrm{T}}\Delta\mathcal{H}^{\mathrm{T}}\mathbf{Q}\Delta\mathcal{H}\Delta\mathbf{u}_k \\
&\qquad - \Delta\mathbf{u}_k^{\mathrm{T}}\Delta\mathcal{H}^{\mathrm{T}}\mathbf{Q}(\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k) \\
&\qquad - (\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k)^{\mathrm{T}}\mathbf{Q}\Delta\mathcal{H}\Delta\mathbf{u}_k\big] \\
&= \left\|\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k\right\|_{\mathbf{Q}}^2 \\
&\quad + \max_{\Delta\mathcal{H}} \left(\Delta\mathbf{u}_k^{\mathrm{T}}\Delta\mathcal{H}^{\mathrm{T}}\mathbf{Q}\Delta\mathcal{H}\Delta\mathbf{u}_k\right) \\
&\quad + \max_{\Delta\mathcal{H}} \big[-\Delta\mathbf{u}_k^{\mathrm{T}}\Delta\mathcal{H}^{\mathrm{T}}\mathbf{Q}(\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k) \\
&\qquad - (\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k)^{\mathrm{T}}\mathbf{Q}\Delta\mathcal{H}\Delta\mathbf{u}_k\big].
\end{aligned} \tag{41}$$

Note that for any $\Delta\mathcal{H}$, (17) indicates $\|\Delta\mathcal{H}\| \leq \beta_{\Delta\mathcal{H}}$. Then, using (38), we obtain

$$\max_{\Delta\mathcal{H}} \|\Delta\mathcal{H}^{\mathrm{T}}\mathbf{Q}\Delta\mathcal{H}\| \leq \beta_{\Delta\mathcal{H}}^2\|\mathbf{Q}\| \leq \|\mathbf{R}\|. \tag{42}$$

It further implies that

$$\max_{\Delta\mathcal{H}} \left(\Delta\mathbf{u}_k^{\mathrm{T}}\Delta\mathcal{H}^{\mathrm{T}}\mathbf{Q}\Delta\mathcal{H}\Delta\mathbf{u}_k\right) \leq \left\|\Delta\mathbf{u}_k\right\|_{\mathbf{R}}^2. \tag{43}$$

Now, substituting (43) into (41) yields

$$\begin{aligned}
\left\|\mathbf{e}_k\right\|_{\mathbf{Q}}^2 &\leq \left\|\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k\right\|_{\mathbf{Q}}^2 + \left\|\Delta\mathbf{u}_k\right\|_{\mathbf{R}}^2 \\
&\quad + \max_{\Delta\mathcal{H}} \big[-\Delta\mathbf{u}_k^{\mathrm{T}}\Delta\mathcal{H}^{\mathrm{T}}\mathbf{Q}(\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k) \\
&\qquad - (\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k)^{\mathrm{T}}\mathbf{Q}\Delta\mathcal{H}\Delta\mathbf{u}_k\big] \\
&\leq \left\|\mathbf{e}_{k-1} - \mathcal{H}\Delta\mathbf{u}_k\right\|_{\mathbf{Q}}^2 + \left\|\Delta\mathbf{u}_k\right\|_{\mathbf{R}}^2 \\
&\quad + 2\max_{\Delta\mathcal{H}} \big[(\mathcal{H}\Delta\mathbf{u}_k - \mathbf{e}_{k-1})^{\mathrm{T}}\mathbf{Q}\Delta\mathcal{H}\Delta\mathbf{u}_k\big].
\end{aligned} \tag{44}$$

Next, we consider the function $\Phi(\Delta\mathcal{H})$ defined as

$$\Phi(\Delta\mathcal{H}) \triangleq \max_{\Delta\mathcal{H}} \big[(\mathcal{H}\Delta\mathbf{u}_k - \mathbf{e}_{k-1})^{\mathrm{T}}\mathbf{Q}\Delta\mathcal{H}\Delta\mathbf{u}_k\big] \tag{45}$$

and note that $\Phi(0) = 0$.

Finally, using (40), (44), and (45), it follows that:

$$\left\|\mathbf{e}_k\right\|_{\mathbf{Q}}^2 - 2\Phi(\Delta\mathcal{H}) \leq J_1(\Delta\mathbf{u}_k). \tag{46}$$

Since $J_1(\Delta\mathbf{u}_k)$ given by (40) is independent of $\Delta\mathcal{H}$, letting $\Delta\mathcal{H} = 0$ in (46) yields (39). This completes the proof. ■

Theorem 1: Consider the system given by (1)–(2) subject to (3)–(5) and controlled by Algorithm 1. If Assumptions 1–3 hold and the problem given by (32) is feasible at the first iteration, then Algorithm 1 guarantees a monotonic decrease on the tracking error $\mathbf{e}_k$ given by (18), i.e.,

$$\|\mathbf{e}_k\|_{\mathbf{Q}}^2 \leq \|\mathbf{e}_{k-1}\|_{\mathbf{Q}}^2$$

for any $k \in \mathbb{I}_{[1,\infty)}$. Furthermore, $\lim_{k\to\infty}\|\mathbf{e}_k\|$ exists.

Proof: First, using Proposition 1, we obtain that $J_1(\Delta\mathbf{u}_k^\star)$ is an upper bound of $\|\mathbf{e}_k\|_{\mathbf{Q}}^2$, where $\Delta\mathbf{u}_k^\star$ is given by (35), i.e.,

$$\|\mathbf{e}_k\|_{\mathbf{Q}}^2 \leq J_1(\Delta\mathbf{u}_k^\star). \tag{47}$$

Next, we note that (32e) holds when $\Delta\mathbf{u}_k = 0$. It can be verified that all the constraints given by (32b)–(32d) is satisfied when $\alpha_i = 0$, $i = 1, \ldots, k-1$ and $\zeta_k = 0$. This indicates that $\Delta\mathbf{u}_k = 0$ is a feasible solution to the optimization problem (32). After solving the optimization problem given by (32), using optimality of the solution $\Delta\mathbf{u}_k^\star$, we have

$$J_1\left(\Delta\mathbf{u}_k^\star\right) \leq J_1(0). \tag{48}$$

Then, substituting $\Delta\mathbf{u}_k = 0$ into (40) yields

$$J_1(0) = \|\mathbf{e}_{k-1}\|_{\mathbf{Q}}^2. \tag{49}$$

Finally, combining (47)–(49) yields

$$\|\mathbf{e}_k\|_{\mathbf{Q}}^2 \leq \|\mathbf{e}_{k-1}\|_{\mathbf{Q}}^2. \tag{50}$$

This means that $\|\mathbf{e}_k\|_{\mathbf{Q}}^2$ is monotonically decreasing with the iteration k. Since $\|\mathbf{e}_k\|_{\mathbf{Q}}^2 \geq 0$, we obtain that $\lim_{k\to\infty}\|\mathbf{e}_k\|$ exists. This completes the proof. ■

IV. CIM-Based Robust Optimal ILC Design

In this section, we develop a CIM-based robust optimal ILC design strategy that utilizes the CIM methodology and adopts the min–max optimization method by maximizing the bounded model uncertainties. To that end, we combine the proposed Algorithm 1 with the presented results in Section II-B2 and redesign the optimization problem (32) to achieve better control performance.

A. CIM-Based Robust Optimal ILC Design

For the developed optimization problem (32), we note that the uncertainty matrix $\Delta\mathcal{H}$ is not explicitly incorporated into the design. Thus, the optimization problem (32) can be designed by combining the robust optimal ILC strategy presented in Section II-B2.

First, for the optimization problem (24), using CIM methodology and substituting (30) into (24b) yields

$$\begin{aligned}\hat{\mathbf{e}}_k &= \mathbf{e}_{k-1} - (\mathcal{H}+\Delta\mathcal{H})\left(\sum_{i=1}^{k-1}\alpha_i\Delta\mathbf{u}_i + \zeta_k\right)\\ &= \mathbf{e}_{k-1} - \sum_{i=1}^{k-1}\alpha_i(\mathcal{H}+\Delta\mathcal{H})\Delta\mathbf{u}_i - (\mathcal{H}+\Delta\mathcal{H})\zeta_k. \quad (51)\end{aligned}$$

Since for the previous iteration i, $i = 1, \ldots, k-1$, using (28) we obtain $(\mathcal{H}+\Delta\mathcal{H})\Delta\mathbf{u}_i = \mathbf{e}_{i-1} - \mathbf{e}_i$ and substituting it into (51) yields

$$\hat{\mathbf{e}}_k = \mathbf{e}_{k-1} + \left[\sum_{i=1}^{k-1}\alpha_i(\mathbf{e}_i - \mathbf{e}_{i-1})\right] - (\mathcal{H}+\Delta\mathcal{H})\zeta_k. \quad (52)$$

Next, substituting (52) into the optimization formulation (24) and taking the model uncertainty information into consideration, a new optimization problem can be formulated as

$$\min_{\alpha_1,\ldots,\alpha_{k-1},\zeta_k}\ \max_{\Delta\mathcal{H}}\ \left\|\hat{\mathbf{e}}_k\right\|_{\mathbf{Q}}^2 + \left\|\Delta\mathbf{u}_k\right\|_{\mathbf{R}}^2 \quad (53a)$$

$$\text{s.t.}\quad (52) \quad (53b)$$

$$\Delta\mathbf{u}_k = \sum_{i=1}^{k-1}\alpha_i\Delta\mathbf{u}_i + \zeta_k \quad (53c)$$

$$\alpha_i \geq 0,\ \ i = 1, \ldots, k-1 \quad (53d)$$

$$\Gamma\Delta\mathbf{u}_k \leq \Lambda_k \quad (53e)$$

where $\mathbf{Q} \succcurlyeq 0$ and $\mathbf{R} \succ 0$ are symmetric sign-definite weighting matrices with compatible dimensions and designed to satisfy (38). It is obvious from (53b) and (53c) that it incorporates both previous input and output error data.

For the optimization problem (53), by comparing (53b) with (34), we obtain that it incorporates the real input and error data of previous trails into the current trail optimization. When $\zeta_k = 0$, the to-be-optimized state $\hat{\mathbf{e}}_k$ satisfies $\hat{\mathbf{e}}_k - \mathbf{e}_{k-1} \in \mathcal{CH}(\{\mathbf{e}_i - \mathbf{e}_{i-1}\}_{i=1}^{k-1})$, which confines the error $\hat{\mathbf{e}}_k - \mathbf{e}_{k-1}$ to be the minimal *conic hull* of the error data set $\{\mathbf{e}_i - \mathbf{e}_{i-1}\}_{i=1}^{k-1}$. This guarantees that the current to-be-optimized state $\mathbf{e}_k$ is always restrained by the real error data of previous iterations. Since we use the real data to compensate for the to-be-optimized state of the model given by (24), the error of the current trail optimization problem (53) is reduced compared with (24) and the control performance using (30) with the CIM methodology is thus improved.

Remark 5: For the optimization problem (53), it is a min–max QP problem based on the worst case performance measure. To solve this problem, we rewrite it in the form of

$$\begin{aligned}\min_{\alpha_1,\ldots,\alpha_{k-1},\zeta_k}\ & \eta\\ \text{s.t.}\ & \left\|\hat{\mathbf{e}}_k\right\|_{\mathbf{Q}}^2 + \left\|\sum_{i=1}^{k-1}\alpha_i\Delta\mathbf{u}_i + \zeta_k\right\|_{\mathbf{R}}^2 \leq \eta\\ & (52)\quad \forall\ \Delta\mathcal{H}\\ & \alpha_i \geq 0,\ \ i = 1, \ldots, k-1\\ & \Gamma\Delta\mathbf{u}_k \leq \Lambda_k.\end{aligned}$$

Then, similar to solving (32), using the *fmincon* function, we can solve the problem.

Algorithm 2 CIM-Based Robust Optimal ILC Algorithm

0) Form the lifted matrix $\mathcal{H}$ given by (10) and select weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ to satisfy (38).
1) Set iteration index $k = 1$ and initialize $\mathbf{u}_0$.
2) Apply $\mathbf{u}_k$ to the system, measure output $\mathbf{y}_k$, and compute tracking error $\mathbf{e}_k$.
3) Obtain the optimal parameters $\{\alpha_1^\star, \ldots, \alpha_{k-1}^\star\}$ and vector $\zeta_k^\star$ by solving the optimization problem (53).
4) Apply (55) to the system and update the data set.
5) Set $k \leftarrow k+1$ and go back to step 2).

B. CIM-Based Robust Optimal ILC Algorithm

We let $\{\alpha_1^\star, \ldots, \alpha_{k-1}^\star, \zeta_k^\star\}$ denote the optimal solutions to the optimization problem (53), then using (53c), the CIM-based robust optimal control input can be represented as

$$\Delta\mathbf{u}_k^\star = \sum_{i=1}^{k-1}\alpha_i^\star\Delta\mathbf{u}_i + \zeta_k^\star. \quad (54)$$

Next, we apply the optimized control input given by

$$\mathbf{u}_k^\star = \mathbf{u}_{k-1} + \sum_{i=1}^{k-1}\alpha_i^\star\Delta\mathbf{u}_i + \zeta_k^\star \quad (55)$$

to the system at the kth iteration and repeat the procedure at the next iteration. We denote the CIM-based robust optimal ILC algorithm as Algorithm 2, which only needs to revise Algorithm 1 from solving (32) in step 3) to solving (53). The detailed description of Algorithm 2 is given as follows.

Remark 6 (Comparison of Computational Complexity): We consider the computation complexity of Algorithm 2 compared with the robust optimal ILC strategy presented in Section II-B2. The total number of variables at each update of the QP problem given by (24) is mN and the computational complexity for the robust optimal ILC strategy is $\mathcal{Q}((mN)^2)$. For Algorithm 2, the total number of variables at each update for solving the QP problem (53) is $(k-1+m)$ and the computational complexity is $\mathcal{Q}((k-1+m)^2)$. Thus, if the iteration index $k \ll (m(N-1)+1)$, Algorithm 2 achieves better computational efficiency than the robust optimal ILC strategy.

C. Theoretical Analysis

In this section, we discuss the feasibility and monotonic convergence of Algorithm 2.

Lemma 2: Consider the system given by (1) and (2) subject to (3)–(5) and controlled by Algorithm 2. If Assumptions 1–3 hold and the problem given by (53) is feasible at iteration k, then Algorithm 1 is always feasible.

Proof: The proof follows similar arguments as the proof of Lemma 1. First, we construct a feasible solution as (37) with $\alpha_i = \alpha_i^\star$, $i = 1, \ldots, k-1$, $\alpha_k = 0$, and $\zeta_{k+1} = \zeta_k^\star$. Then, it can be easily verified that all the constraints given by (53b)–(53e) are satisfied except for the minimization of the performance measure (53a). Finally, we conclude that there is at least one admissible solution for the problem given by (53) at iteration $k+1$, indicating that Algorithm 2 is feasible. This completes the proof. ∎

Proposition 2: Consider the system given by (1) and (2) and assume that Assumptions 1–3 hold. Let the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ in (53) are designed to satisfy (38). Then, the actual tracking error $\|\mathbf{e}_k\|_{\mathbf{Q}}^2$ under Algorithm 2 is upper bounded by

$$\left\|\mathbf{e}_k\right\|_{\mathbf{Q}}^2 \leq J_2(\Delta \mathbf{u}_k) \tag{56}$$

for any k, where $J_2(\Delta \mathbf{u}_k)$ is defined as

$$J_2(\Delta \mathbf{u}_k) \triangleq \max_{\Delta \mathcal{H}} \ \left\|\mathbf{e}_{k-1} - (\mathcal{H} + \Delta \mathcal{H})\Delta \mathbf{u}_k\right\|_{\mathbf{Q}}^2 + \left\|\Delta \mathbf{u}_k\right\|_{\mathbf{R}}^2.$$

Proof: Using similar arguments used to prove Proposition 1, we obtain

$$\begin{aligned}\left\|\mathbf{e}_k\right\|_{\mathbf{Q}}^2 &\leq \max_{\Delta \mathcal{H}} \ \left\|\mathbf{e}_{k-1} - (\mathcal{H} + \Delta \mathcal{H})\Delta \mathbf{u}_k\right\|_{\mathbf{Q}}^2 \\ &\leq \max_{\Delta \mathcal{H}} \ \left\|\mathbf{e}_{k-1} - (\mathcal{H} + \Delta \mathcal{H})\Delta \mathbf{u}_k\right\|_{\mathbf{Q}}^2 + \left\|\Delta \mathbf{u}_k\right\|_{\mathbf{R}}^2 \\ &= J_2(\Delta \mathbf{u}_k). \end{aligned} \tag{57}$$

This completes the proof. ∎

Theorem 2: Consider the system given by (1) and (2) subject to (3)–(5) and controlled by Algorithm 2. If Assumptions 1–3 hold and the problem given by (53) is feasible at the first iteration, then Algorithm 2 guarantees a monotonic decrease on the tracking error $\mathbf{e}_k$ given by (18), i.e.,

$$\|\mathbf{e}_k\|_{\mathbf{Q}}^2 \leq \|\mathbf{e}_{k-1}\|_{\mathbf{Q}}^2$$

for any $k \in \mathbb{I}_{[1,\infty)}$. Furthermore, $\lim_{k\to\infty} \|\mathbf{e}_k\|$ exists.

Proof: The proof is similar to that for Theorem 1. Thus, we give sketch proof here. First, using Proposition 2, we obtain

$$\|\mathbf{e}_k\|_{\mathbf{Q}}^2 \leq J_2\big(\Delta \mathbf{u}_k^\star\big) \tag{58}$$

where $\Delta \mathbf{u}_k^\star$ is the optimal solution to (53) given by (54).

Next, it can be verified that $\Delta \mathbf{u}_k = 0$ is a feasible solution to the optimization problem (53) with $\alpha_i = 0$, $i = 1, \ldots, k-1$ and $\zeta_k = 0$. Then, using optimality of the solution $\Delta \mathbf{u}_k^\star$, it follows that:

$$J_2\big(\Delta \mathbf{u}_k^\star\big) \leq J_2(0) = \|\mathbf{e}_{k-1}\|_{\mathbf{Q}}^2. \tag{59}$$

Finally, combining (58) and (59) yields

$$\|\mathbf{e}_k\|_{\mathbf{Q}}^2 \leq \|\mathbf{e}_{k-1}\|_{\mathbf{Q}}^2. \tag{60}$$

This indicates that $\|\mathbf{e}_k\|_{\mathbf{Q}}^2$ is monotonically decreasing with the iteration k. Note that $\|\mathbf{e}_k\|_{\mathbf{Q}}^2 \geq 0$. Finally, we obtain that $\lim_{k\to\infty} \|\mathbf{e}_k\|$ exists. This completes the proof. ∎

Remark 7: For Algorithms 1 and 2, we have the following observations.

1) They are both data-driven learning methods that utilize the CIM methodology to reduce the errors of the current trail optimization. Both Algorithms are optimal strategies with an integration of the *conic hull* of the data set $\mathbb{S}_k$.
2) The CIM methodology utilizes the historical process data to compensate for the current trail optimization, thus achieving smaller tracking errors and better control performance for uncertain systems. In particular, Algorithm 1 is expected to achieve better control performance than the strategy presented in Section II-B1; Algorithm 2 can achieve better control performance than the strategy presented in Section II-B2 and Algorithm 1.

V. Illustrative Numerical Examples

In this section, two illustrative numerical examples are provided to illustrate the key ideas presented in this article. The first example is a benchmark example in the presence of uncertainties. The second example is an injection molding process in the presence of uncertain perturbations.

A. Example 1: Benchmark

Consider the benchmark example adapted from [44] given by (1) and (2) with

$$A = \begin{bmatrix} 0.8 & -0.4 & 0.2 \\ 0 & 0.3 & -0.5 \\ 0 & 0 & 0.5 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & -0.6 \\ 0.5 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Delta_a(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.15\exp(-0.1t) \\ 0 & 0 & 0 \end{bmatrix}, \Delta_b(t) = 0$$

which is a MIMO model with $m = p = 2$. Note that $\|\Delta_a(t)\| \leq \beta_a = 0.15$ and $\beta_b = 0$. The desired reference trajectory is set as $y_{\mathrm{d}}(t) = [0.5\sin(2\pi t/N), \sin(2\pi t/N)]^{\mathrm{T}}$, where $N = 50$ s is the trail length. The constraints are set in the form of (3)–(5) with $\overline{u}_i = -\underline{u}_i = 0.6$, $\overline{\delta u}_i = -\underline{\delta u}_i = 0.5$, and $\overline{\mathrm{d}u}_i = -\underline{\mathrm{d}u}_i = 0.5$, $i = 1, 2$.

To illustrate the performance results of Algorithms 1 and 2, the weighting matrices are set as $\mathbf{Q} = I$ and $\mathbf{R} = 0.5I$. Fig. 2 shows the control inputs of the system in the presence of uncertainties with Algorithm 1 from eight iterations. Fig. 3 shows the same graph for the system with Algorithm 2. It can be seen from Figs. 2 and 3 that all the constraints are satisfied for all iterations. To compare the results with the existing methods without using the CIM methodology, we adopt the optimal ILC strategy developed in [34] and the robust optimal ILC strategy developed in [36] to realize the same control objective for the system. To make the comparisons more clear,

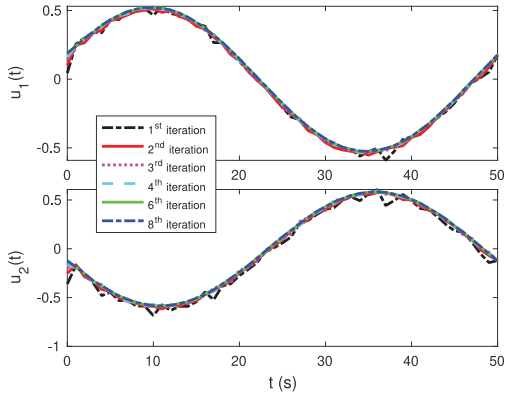


Fig. 2. Control inputs of the system in the presence of uncertainties with Algorithm 1 from eight iterations.

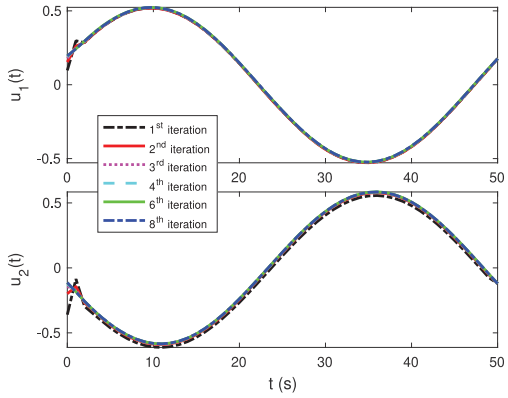


Fig. 3. Control inputs of the system in the presence of uncertainties with Algorithm 2 from eight iterations.

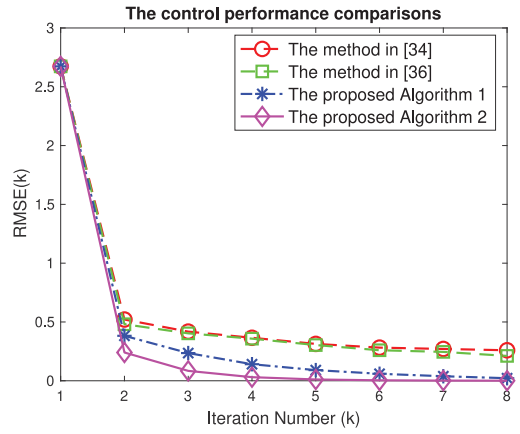


Fig. 4. Comparison results of RMSE of the system in the presence of uncertainties from eight iterations with the control methods in [34] and [36], Algorithms 1 and 2.

we introduce the index of root mean squared error (RMSE) as

$$\mathrm{RMSE}(k) = \sqrt{\sum_{i=1}^{N}(y_{\mathrm{d}}(t) - y_k(t))^2} = \|\mathbf{y}_{\mathrm{d}} - \mathbf{y}_k\|_2. \quad (61)$$

The use of RMSE is very common, and it is considered an excellent general-purpose error metric for numerical predictions. Comparisons of RMSE of the system from eight iterations with the control methods in [34] and [36] and Algorithms 1 and 2 are shown in Fig. 4. It can be seen from

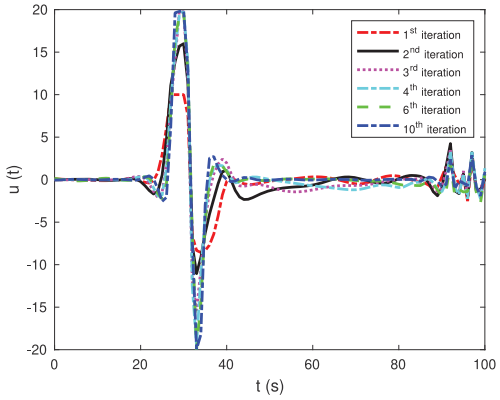


Fig. 5. Control inputs of the injection molding process in the presence of uncertain perturbations with Algorithm 1 from ten iterations.

Fig. 4 that the system with Algorithm 2 converges faster than Algorithm 1.

B. Example 2: Injection Molding Process

Injection molding, which is a typical batch process, consists of three main stages: 1) filling; 2) packing; and 3) cooling [45]. For the packing stage, the nozzle pressure, which is a key process variable, should be controlled to follow a given profile to ensure product quality. Based on the least-square regression method using historical process data, the nozzle packing pressure response to the hydraulic control valve opening can be identified as an uncertain state-space form presented in [46]. This model has been widely studied in [47] and [48]. The detailed model adopted from [46] is given by

$$\begin{aligned} x_k(t+1) &= \left(\begin{bmatrix} 1.607 & 1 \\ -0.6086 & 0 \end{bmatrix} + \Delta_a(t)\right)x_k(t) \\ &\quad + \left(\begin{bmatrix} 1.2390 \\ -0.9282 \end{bmatrix} + \Delta_b(t)\right)u_k(t), \quad 0 \le t \le 100 \\ y_k(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_k(t) \end{aligned}$$

where the uncertain perturbations $\Delta_a(t)$ and $\Delta_b(t)$ are calculated by using statistical learning methods based on a confidence level from the data, which are given by

$$\Delta_a(t) = \begin{bmatrix} 0.08\delta_1(t) & 0 \\ 0.08\delta_2(t) & 0 \end{bmatrix}, \quad \Delta_b(t) = \begin{bmatrix} 0.1\delta_3(t) \\ 0.1\delta_4(t) \end{bmatrix}$$

and $\delta_i(t) \in \{\delta_i : |\delta_i| < 0.5\}$, $i \in \{1, 2, 3, 4\}$, denote independently and uniformly distributed random variables in the interval $(0, 1)$. Note that $\|\Delta_a(t)\| \le \beta_a = 0.08$ and $\|\Delta_b(t)\| \le \beta_b = 0.1$. The desired reference trajectory is set as

$$y_{\mathrm{d}}(t) = \begin{cases} 0, & 0 \le t < 20 \\ 150\tanh(t-30) + 150, & 20 \le t \le 40 \\ 300, & 40 < t \le 100 \end{cases}$$

which corresponds to $N = 100$ in (1) and (2) for the injection molding process. Here, we use the function $\tanh(\cdot)$ as a smoother for practical implementation. Furthermore, we consider the constraints in (3)–(5) for the process as $\overline{u}_i = -\underline{u}_i = 20$, $\overline{\delta u}_i = -\underline{\delta u}_i = 10$, and $\overline{\mathrm{d}u}_i = -\underline{\mathrm{d}u}_i = 20$.

To illustrate the system performance of Algorithms 1 and 2, the weighting matrices are set as $\mathbf{Q} = I$ and $\mathbf{R} = 10I$. In Fig. 5,

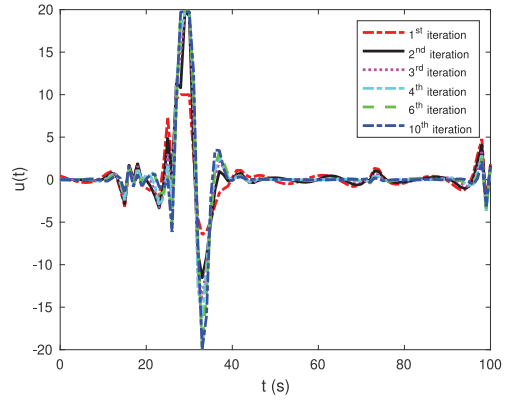


Fig. 6. Control inputs of the injection molding process in the presence of uncertain perturbations with Algorithm 2 from ten iterations.

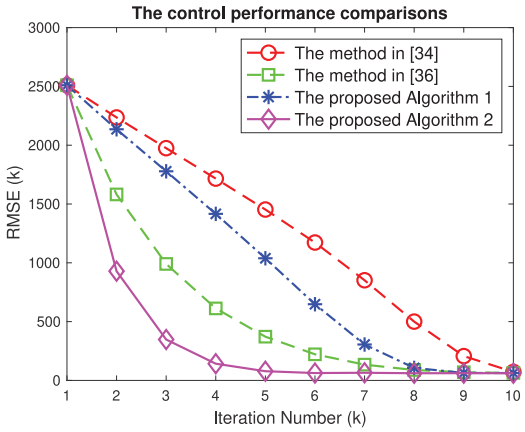


Fig. 7. Comparison results of RMSE of the injection molding process in the presence of uncertain perturbations from ten iterations with the control methods in [34] and [36], and Algorithms 1 and 2.

we show the control inputs of the process in the presence of uncertain perturbations with Algorithm 1 from ten iterations. In Fig. 6, we show the same graph for the process with Algorithm 2. It can be seen from Figs. 5 and 6 that the constraints are satisfied for all iterations. Next, we compare the control performance of Algorithms 1 and 2 with the basic optimal ILC strategy developed in [34] and the robust optimal ILC strategy developed in [36] for the same control objective. To make the comparisons more clear, we adopt the RMSE given by (61) as a comparison index. Fig. 7 plots the comparison results of RMSE of the process from ten iterations with the control methods in [34] and [36] and Algorithms 1 and 2. It can be seen from Fig. 7 that the CIM-based optimal ILC strategies have improved the control performance for the system. In particular, Algorithm 1 achieves better control performance and faster convergence rate than the optimal ILC method in [34]; Algorithm 2 achieves better control performance and faster convergence rate than the robust optimal ILC method in [36] and Algorithm 1.

VI. Conclusion

In this article, we proposed a data-driven design methodology called CIM. Then, CIM-based optimal ILC and CIM-based robust optimal ILC are developed for uncertain systems with constraints. Theoretical analyses for the two strategies are presented, respectively, demonstrating the feasibility of the algorithms and the monotonic convergence of the systems. The key idea behind our CIM design methodology is that there was available process data that interacted with uncertainties and thus, incorporating these data into the optimization-based ILC design reveals the potentials to reduce the errors of the current trail optimization, thereby improving control performance for uncertain systems. The illustrative numerical examples on the benchmark example and the injection molding process have shown that the proposed optimal ILC methods have some clear advantages over the existing optimal ILC methods. Future research will focus on exploiting the potential of the CIM methodology for systems with passive incomplete information, such as random data dropouts, faded measurements, etc.

References

[1] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.

[2] Y. Wang, F. Gao, and F. J. Doyle III, "Survey on iterative learning control, repetitive control, and run-to-run control," *J. Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.

[3] J.-X. Xu and Y. Tan, *Linear and Nonlinear Iterative Learning Control*. Berlin, Germany: Springer, 2003.

[4] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1099–1121, Nov. 2007.

[5] L. Wang, R. Zhang, and F. Gao, *Iterative Learning Stabilization and Fault-Tolerant Control for Batch Processes*. Singapore: Springer, 2020.

[6] J. H. Lee and K. S. Lee, "Iterative learning control applied to batch processes: An overview," *Control Eng. Pract.*, vol. 15, no. 10, pp. 1306–1318, 2007.

[7] D. Shen, "Iterative learning control with incomplete information: A survey," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 5, pp. 885–901, Sep. 2018.

[8] 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.

[9] J. Lu, Z. Cao, R. Zhang, and F. Gao, "Nonlinear monotonically convergent iterative learning control for batch processes," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5826–5836, Jul. 2018.

[10] X. Jin, "Fault tolerant nonrepetitive trajectory tracking for MIMO output constrained nonlinear systems using iterative learning control," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 3180–3190, Aug. 2019.

[11] W. He, T. Meng, X. He, and C. Sun, "Iterative learning control for a flapping wing micro aerial vehicle under distributed disturbances," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1524–1535, Apr. 2019.

[12] X. Jin, "Iterative learning control for MIMO nonlinear systems with iteration-varying trial lengths using modified composite energy function analysis," *IEEE Trans. Cybern.*, vol. 51, no. 12, pp. 6080–6090, Dec. 2021.

[13] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control for discrete-time systems with exponential rate of convergence," *IEE Proc. Control Theory Appl.*, vol. 143, no. 2, pp. 217–224, 1996.

[14] C. Freeman and T. V. Dinh, "Experimentally verified point-to-point iterative learning control for highly coupled systems," *Int. J. Adapt. Control Signal Process.*, vol. 29, no. 3, pp. 302–324, 2015.

[15] P.-C. Lu, J. Chen, and L. Xie, "Iterative learning control (ILC))-based economic optimization for batch processes using helpful disturbance information," *Ind. Eng. Chem. Res.*, vol. 57, no. 10, pp. 3717–3731, 2018.

[16] B. Srinivasan and D. Bonvin, "Real-time optimization of batch processes by tracking the necessary conditions of optimality," *Ind. Eng. Chem. Res.*, vol. 46, no. 2, pp. 492–504, 2007.

[17] Q. Ai *et al.*, "High-order model-free adaptive iterative learning control of pneumatic artificial muscle with enhanced convergence," *IEEE Trans. Ind. Electron.*, vol. 67, no. 11, pp. 9548–9559, Nov. 2020.

[18] R. Chi, Z. Hou, S. Jin, and B. Huang, "Computationally efficient data-driven higher order optimal iterative learning control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 5971–5980, Dec. 2018.

[19] R. Chi, H. Zhang, B. Huang, and Z. Hou, "Quantitative data-driven adaptive iterative learning control: From trajectory tracking to point-to-point tracking," *IEEE Trans. Cybern.*, early access, Oct. 23, 2020, doi: 10.1109/TCYB.2020.3015233.

[20] D. Li *et al.*, "Synthesis of ILC–MPC controller with data-driven approach for constrained batch processes," *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 3116–3125, Apr. 2020.

[21] J. Shi, F. Gao, and T.-J. Wu, "Single-cycle and multi-cycle generalized 2D model predictive iterative learning control (2D-GPILC) schemes for batch processes," *J. Process Control*, vol. 17, no. 9, pp. 715–727, 2007.

[22] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Eng. Pract.*, vol. 11, no. 7, pp. 733–764, 2003.

[23] Y. Zhou, D. Li, J. Lu, Y. Xi, and L. Cen, "Networked and distributed predictive control of non-linear systems subject to asynchronous communication," *IET Control Theory Appl.*, vol. 12, no. 4, pp. 504–514, 2017.

[24] D. Li, Y. Xi, J. Lu, and F. Gao, "Synthesis of real-time-feedback-based 2D iterative learning control–model predictive control for constrained batch processes with unknown input nonlinearity," *Ind. Eng. Chem. Res.*, vol. 55, no. 51, pp. 13074–13084, 2016.

[25] X. Liu, L. Ma, X. Kong, and K. Y. Lee, "Robust model predictive iterative learning control for iteration-varying-reference batch processes," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 7, pp. 4238–4250, Jul. 2021.

[26] J. Shi, F. Gao, and T.-J. Wu, "Robust design of integrated feedback and iterative learning control of a batch process based on a 2D Roesser system," *J. Process Control*, vol. 15, no. 8, pp. 907–924, 2005.

[27] P. Dabkowski, K. Galkowskiy, E. Rogers, Z. Cai, C. T. Freeman, and P. L. Lewin, "Iterative learning control based on relaxed 2-D systems stability criteria," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 1016–1023, May 2013.

[28] 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.

[29] Z. Hou, H. Gao, and F. L. Lewis, "Data-driven control and learning systems," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4070–4075, May 2017.

[30] J. Lu, Z. Cao, C. Zhao, and F. Gao, "110th anniversary: An overview on learning-based model predictive control for batch processes," *Ind. Eng. Chem. Res.*, vol. 58, no. 37, pp. 17164–17173, 2019.

[31] J. Liu, X. Ruan, and Y. Zheng, "Iterative learning control for discrete-time systems with full learnability," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 629–643, Feb. 2022.

[32] S. He, W. Chen, D. Li, Y. Xi, Y. Xu, and P. Zheng, "Iterative learning control with data-driven-based compensation," *IEEE Trans. Cybern.*, early access, Jan. 5, 2021, doi: 10.1109/TCYB.2020.3041705.

[33] X. Yu, Z. Hou, M. M. Polycarpou, and L. Duan, "Data-driven iterative learning control for nonlinear discrete-time MIMO systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1136–1148, Mar. 2021.

[34] Z. Xu, J. Zhao, Y. Yang, Z. Shao, and F. Gao, "Optimal iterative learning control based on a time-parametrized linear time-varying model for batch processes," *Ind. Eng. Chem. Res.*, vol. 52, no. 18, pp. 6182–6192, 2013.

[35] H. Sun and A. G. Alleyne, "A computationally efficient norm optimal iterative learning control approach for LTV systems," *Automatica*, vol. 50, no. 1, pp. 141–148, 2014.

[36] Z. Xu, J. Zhao, Y. Yang, Z. Shao, and F. Gao, "Robust iterative learning control with quadratic performance index," *Ind. Eng. Chem. Res.*, vol. 51, no. 2, pp. 872–881, 2012.

[37] J. Lu, Z. Cao, Q. Hu, Z. Xu, W. Du, and F. Gao, "Optimal iterative learning control for batch processes in the presence of time-varying dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 680–692, Jan. 2022.

[38] Y. Zhou, D. Li, Y. Xi, and Z. Gan, "Synthesis of model predictive control based on data-driven learning," *Sci. China Inf. Sci.*, vol. 63, pp. 1–3, Mar. 2020.

[39] Y. Zhou, K. G. Vamvoudakis, W. M. Haddad, and Z.-P. Jiang, "A secure control learning framework for cyber-physical systems under sensor and actuator attacks," *IEEE Trans. Cybern.*, vol. 51, no. 9, pp. 4648–4660, Sep. 2021.

[40] S.-K. Oh and J. M. Lee, "Iterative learning model predictive control for constrained multivariable control of batch processes," *Comput. Chem. Eng.*, vol. 93, pp. 284–292, Oct. 2016.

[41] K. S. Lee, I.-S. Chin, H. J. Lee, and J. H. Lee, "Model predictive control technique combined with iterative learning for batch processes," *AIChE J.*, vol. 45, no. 10, pp. 2175–2187, 1999.

[42] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1970.

[43] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[44] J. Hou, F. Chen, P. Li, and Z. Zhu, "Prior-knowledge-based subspace identification for batch processes," *J. Process Control*, vol. 82, pp. 22–30, Oct. 2019.

[45] F. Gao, Y. Yang, and C. Shao, "Robust iterative learning control with applications to injection molding process," *Chem. Eng. Sci.*, vol. 56, no. 24, pp. 7025–7034, 2001.

[46] J. Shi, F. Gao, and T.-J. Wu, "Robust iterative learning control design for batch processes with uncertain perturbations and initialization," *AIChE J.*, vol. 52, no. 6, pp. 2171–2187, 2006.

[47] D. Meng and K. L. Moore, "Convergence of iterative learning control for SISO nonrepetitive systems subject to iteration-dependent uncertainties," *Automatica*, vol. 79, pp. 167–177, May 2017.

[48] S. Hao, T. Liu, and E. Rogers, "Extended state observer based indirect-type ILC for single-input single-output batch processes with time- and batch-varying uncertainties," *Automatica*, vol. 112, Feb. 2020, Art. no. 108673.

Yuanqiang Zhou received the B.S. degree in mathematics and applied mathematics and the M.S. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2013 and 2015, respectively, and the Ph.D. degree in control science and engineering from the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, in 2020.

From 2017 to 2019, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, New York University, New York, NY, USA. He is currently a Research Associate with the Department of Chemical and Biological Engineering, Hong Kong University of Science and Technology, Hong Kong. His research interests include model predictive control, iterative learning control, data-driven control, optimization, and reinforcement learning.

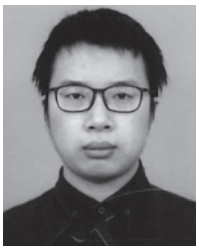
Kaihua Gao received the B.S. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Chemical and Biological Engineering, Hong Kong University of Science and Technology, Hong Kong.

His current research interests include batch process learning-based control and dynamic modeling.

Xiaopeng Tang received the B.S. degree in automation from the University of Science and Technology of China, Hefei, China, in 2015, and the Ph.D. degree in chemical and biological engineering from the Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2020.

He is currently a Postdoctoral Researcher with HKUST. His research interests include modeling and control of energy storage systems.

Dr. Tang was a recipient of the National scholarship in 2015, the Hong Kong Ph.D. Fellowship Scheme in 2017, and the RGC Postdoctoral Fellowship Award in 2021. He is on editorial boards of some journals of his area, including *Transactions on the Institute of Measurement and Control* and *Frontiers in Energy Research*.



Huanjia Hu received the B.S. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Chemical and Biological Engineering, Hong Kong University of Science and Technology, Hong Kong.

His current research interests include quality prediction and soft sensor of injection molding process.

Dewei Li received the B.S. and the Ph.D. degrees in automation from Shanghai Jiao Tong University, Shanghai, China, in 1993 and 2009, respectively.

He worked as a Postdoctoral Researcher with Shanghai Jiao Tong University, from 2009 to 2010, where he is a Professor with the Department of Automation. His research interests include predictive control, robust control, and the related applications.

Dr. Li is an Associate Editor of the IFAC journal and *Control Engineering Practice*.

Furong Gao received the bachelor's degree in automation from the East China Institute of Petroleum, Beijing, China, in 1985, and the master's and Ph.D. degrees in chemical engineering from McGill University, Montreal, QC, Canada, in 1989 and 1993, respectively.

He is a Chair Professor of Chemical and Biological Engineering with the Hong Kong University of Science and Technology (HKUST), Hong Kong, and also serves HKUST as the Division Director of Advanced Manufacturing and Automation with the HKUST's Fok Ying Tung Graduate School. His research interests include batch process modeling, monitoring, optimization, and control.